VTT PUBLICATIONS 272

Enchancing communication of plant design knowledge

Kari Kaarela

VTT Electronics

Academic dissertation for degree of Doctor of Technology to be presented, with the permission of the department of Process Engineering of the University of Oulu, for public discussion in the auditorium L10, Linnanmaa, on May 10th, 1996, at 12.00 o'clock noon.

TECHNICAL RESEARCH CENTRE OF FINLAND
ESPOO 1996

ISBN 951-38-4930-9
ISSN 1235-0621

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 Elektroniikka, Sulautetut ohjelmistot, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 509 111, faksi (08) 551 2320

VTT Elektronik, Inbyggd programvara, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 509 111, fax (08) 551 2320

VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland
phone internat. + 358 8 509 111, fax + 358 8 551 2320

Technical editing Leena Ukskoski

LIBELLA PAINOPALVELU OY, ESPOO 1996

Dedicated to my children, Terhi, Ilari, Taneli, and Katja.

Kaarela, Kari. Enhancing communication of plant design knowledge. Espoo 1996, Technical Research Centre of Finland. VTT Publications 272. 110 p. + app. 81 p.

UDC 338.45.006.3:658.512.2:681.3

Keywords industrial plants, design, planning, computers, computer programs, information, transferring, models, computer interfaces, communicating, conceptual process design, multilevel flow modelling, design knowledge, operator support systems, on-line documentation

ABSTRACT

In industrial plant projects much of the design knowledge is lost. Design documents typically concentrate on implementation-oriented structural information that is required to construct the plant. The documents to a great extent lack the functional knowledge representing the original design intentions, formed mainly during the conceptual design stage, and information on the design process. As a result, the design documents do not contain answers to questions about the purpose of the design. One typically cannot find any justifications for the design decisions, e.g. the selection of one of several apparently sufficiently good alternatives. Yet both kinds of information have proven to be important for effective co-operation of designers within a project, for succeeding projects, and for the operation and maintenance of the resulting plant. This thesis addresses the problem of structuring, recording, and transferring the design knowledge between designers and from the designers to the end users.

The proposed solution is to introduce formal knowledge representations powerful enough to express the relevant concepts of the domain into the design process. With these representations and supporting tools we aim to structure and record the functional design knowledge at design time. Multilevel flow modelling is used to capture the functional knowledge of the plant. The resulting conceptual model is utilised for structuring the informal design knowledge related to both the designed artefact and the design process. Based on two validating industrial cases, the approach seems able to capture this design knowledge in industrial plant projects.

On-line presentation of the functional knowledge to the plant operators is discussed. Our work emphasises the consistent presentation of the conceptual model created by the designers to the operators during the training sessions, in the user interfaces and alarm systems of the plant's automation systems, and in the on-line documentation system coupled to the automation systems. Two operator support system prototypes, implemented using commercial state-of-the-art automation systems and recent information technology developments such as the World Wide Web, show that it is feasible to implement useful MFM-based sources of information for the operators of industrial plants, based on the information gathered while applying the proposed approach during the design phase.

PREFACE

This thesis is based on the research work performed at VTT Electronics in the TIESU and CICERO projects during the period 1992 through 1995. TIESU (a Finnish acronym for "Integrating and Embedding Design Knowledge in Process Management") was part of a Finnish national process technology research program funded mainly by the Technology Development Centre in Finland (TEKES). It was run jointly by the Department of Process Engineering at the University of Oulu and VTT Electronics. CICERO ("Control Information system Concepts based on Encapsulated Real-time Objects") was an international collaboration, organised and managed by CERN.

I wish to express my deepest gratitude to my closest colleagues in both of the projects, Mr. Pertti Huuskonen and Mr. Jaakko Oksanen, for their friendship and co-operation during all these years. Not only was it a period of hard labour but also of great fun (Motto: poor humour is better than no humour?). I am most grateful to all my fellow researchers who participated in the TIESU project: Mr. Pekka Ala-Siuru, Mr. Juha Jaako, Mr. Jarmo Kalaoja, Mr. Pasi Kinnunen, Mr. Kimmo Lukkari, Ms. Agne Malmström, Ms. Elina Savolainen, and Ms. Anne Väisänen. Mr. Juha Jaako is especially acknowledged for providing his process expertise and creating the MFM models. Mr. Juha Takalo implemented the on-line documentation system for the CICERO prototype with flying colours.

This research would not have been possible without the industrial interest and involvement of Imatran Voima, Inc.; Valmet Automation, Inc.; and Tampella Power, Inc., in the projects. I wish to express my sincere gratitude to all these organisations for enabling this research by contributing to the required financial, human, and material resources. Imatran Voima, Inc., provided the TIESU project with a case study and a lot of expertise. Tampella Power, Inc., allowed us to use their process design tool in the TIESU project. Valmet Automation, Inc., provided both projects with state-of-the-art automation tools and their expertise on automation design. I am grateful to Dr. Jean-Marie Le Goff, the co-ordinator of the CICERO project at CERN, for providing us with a good opportunity to test and further develop the solutions generated in the TIESU project in an interesting and demanding environment.

I also want to express my gratitude to all the members of the participating companies who have contributed to this work. The staff of the Haapavesi peat power plant is acknowledged for teaching us some of their experiential knowledge on running and maintaining industrial plants. Mr. Erkki Rautio and Mr. Osmo Katainen explained us the essentials of industrial plant design projects. Ms. Tiina Rossi both taught us how to use the process design tool and created the original P&I diagrams. The active participation of Mr. Hannu Paunonen and Mr. Heikki Pernu in both the management and technical work of the TIESU project is appreciated. Mr. Martti Meri and Mr. Esa Salonen are acknowledged for providing great visions during our fruitful high-level discussions in France.

I wish to thank Prof. Kauko Leiviskä, the supervisor of this thesis at the University of Oulu, for his encouragement and guidance throughout this work. The reviewers of this dissertation, Dr. Raimo Korhonen and Prof. Asko Riitahuhta, deserve my deepest gratitude for their comments that helped me to improve both the content and readability of this thesis. Thanks are also due to Prof. Veikko Seppänen who unselfishly interrupted his studies on Päätalo's Iijoki series to provide insightful critiques of several versions of this thesis on an extremely short notice. Dr. Kari Leppälä's expert comments helped me to improve the organisation of the thesis. Mr. Douglas Foxvog is acknowledged for finding the time to proofread the thesis as well as several of the included papers.

The supervisor of my earlier postgraduate studies, Prof. Matti Pietikäinen, deserves a special mention. His memorable words concerning the *drive* required to accomplish doctoral studies, uttered during my Lic. Tech. party, were a great inspiration during this work.

I wish to thank all my colleagues at VTT Electronics in general and in the embedded knowledge-based systems research group in particular for creating a stimulating working atmosphere. The recent theses of my colleagues, Dr. Pekka Isomursu and Dr. Matti Kurki, showed me a path to follow and gave me the strength to carry on with this work, which sometimes felt never ending. I hope my thesis will similarly inspire my younger colleagues.

The financial support provided by VTT, the above mentioned organisations, and Jenny and Antti Wihurin Säätiö is gratefully acknowledged.

Finally, I wish to express my sincere gratitude to my beloved for their support and understanding during the course of this work. Despite the fact that this thesis has mainly been written in the loneliness of the night, far too many otherwise shared moments have been sacrificed. Terhi, Ilari, and Taneli gave me the joy of life required to counteract the fatigue of writing the included papers. Katja, my little associate researcher, taught me that it is often better to give things a second thought than to write them down at once. Maybe writing the thesis took a little longer this way, but obviously the result is somewhat better. My companion in life, Elina, gave me inspiration, support, and loving care that enabled me to get the work done.

Oulu, May 1996
Kari Kaarela

CONTENTS

ABSTRACT 5

PREFACE..6

CONTENTS 8

LIST OF ORIGINAL PUBLICATIONS 10

LIST OF SYMBOLS AND ABBREVIATIONS 12

1 INTRODUCTION 14
 1.1 Industrial plant design...14
 1.2 Multilevel flow modelling 17
 1.3 Research problem 18
 1.4 Research assumptions...19
 1.5 Hypothesis 19
 1.6 Research methodology..20
  1.6.1 Research strategy 20
  1.6.2 Scope of applicability..21
 1.7 Results...23
 1.8 Outline of the dissertation...24

2 ENGINEERING DESIGN 25
 2.1 Design knowledge...27
  2.1.1 Artefact-related design knowledge 28
  2.1.2 Design-related design knowledge 30
 2.2 Interdisciplinary design projects...35

3 CURRENT DESIGN TOOLS AND REPRESENTATIONS 40
 3.1 Tools 40
 3.2 Design representations..42
  3.2.1. Drawings and diagrams...43
  3.2.2. Textual design documents...45
 3.3 Standards...48
 3.4 Discussion...49

4 MFM AS A FRAMEWORK FOR CAPTURING PLANT DESIGN KNOWLEDGE...50
 4.1 Multilevel flow modelling 50
  4.1.1 Whole-part and means-end dimensions 51
  4.1.2 Flows..52
 4.2 Representing artefact-related design knowledge with MFM.......55
  4.2.1 Representing structural and functional knowledge.................55
  4.2.2 Representing problem decomposition knowledge..................57
 4.3 Using MFM for structuring design-related knowledge 58
 4.4 Discussion...59

5 A DESIGN ENVIRONMENT PROTOTYPE 61
5.1 Representing the design at multiple levels of abstraction......................63
5.2 Design-related knowledge 63
5.3 Structured documentation...65
5.4 Support for reuse...67
5.4.1 Reuse of complete or partial MFM models 68
5.4.2 Explaining existing designs 69
5.5 Conversions 71
5.6 Discussion...71
6 PRESENTING DESIGN KNOWLEDGE TO INDUSTRIAL PLANT OPERATORS...74
6.1 User interfaces 76
6.1.1 Problems with current user interfaces.......................................76
6.1.2 MFM-based user interfaces...77
6.2 Alarm processing systems 79
6.2.1 Problems with current alarm processing systems 79
6.2.2 MFM-based alarm processing systems.....................................80
6.3 On-line documentation 82
6.3.1 Problems with current on-line documentation systems 82
6.3.2 An approach for improving on-line documentation...................83
6.3.3 Storing experiential knowledge 87
6.4 Training...88
6.5 Explanation mechanism..89
6.6 Discussion...90
7 INTRODUCTION TO THE PAPERS 91
7.1 Paper I, An Overview (ICARCV '92)...91
7.2 Paper II, Needs analysis (ICO '93)...91
7.3 Paper III, Organising design knowledge (IFIP '94) 92
7.4 Paper IV, Recognition of safety goals (SAFEPROCESS '94)...............93
7.5 Paper V, Explaining design knowledge (HCI '95)................................93
7.6 Paper VI, Principles for representing MFM information to operators (HCI '93) 93
7.7 Paper VII, An operator support system based on MFM information (HICS '94)...94
7.8 Paper VIII, On-line documentation 94
7.9 Paper IX, Explaining design knowledge to operators (IEA&AIE '95)..95
8 CONCLUSIONS 96
REFERENCES 99
APPENDICES (PAPERS I TO IX)

Appendices of this publication are not included in the PDF version.
Please order the printed version to get the complete publication
(http://www.inf.vtt.fi/pdf/publications/1996)

LIST OF ORIGINAL PUBLICATIONS

This dissertation includes the following nine original publications (Appendices I to IX):

I Kaarela, K., Huuskonen, P. & Kurki, M. 1992. Systematic Knowledge Acquisition in Process Automation Design. In: Proceedings of the Second International Conference on Automation, Robotics, and Computer Vision (ICARCV '92), Singapore, September 16 - 18, 1992. Vol. 2. Singapore: Nanyang Technological University and the Institution of Engineers. Pp. IA-6.5.1 - 5.

II Kaarela, K., Huuskonen, P. & Leiviskä, K. 1993. The Role of Design Knowledge in Industrial Plant Projects. In: Proceedings of the 4th International Conference on Cognitive and Computer Sciences for Organizations (ICO '93), Montreal, Canada, May 4 - 7, 1993. Montreal: Girico. Pp. 173 - 183. ISBN 2-7624-0555-6.

III Kaarela, K. & Oksanen, J. 1994. Structuring and Recording Plant Design Knowledge. Proceedings of the IFIP International Conference on Feature Modelling and Recognition in Advanced CAD/CAM Systems, Valenciennes, France, May 24 - 26, 1994, Vol. 2. Valenciennes, France: University of Valenciennes. Pp. 853 - 866.

IV Kaarela, K. & Oksanen, J. 1994. Communicating the Safety Aspects of Industrial Plants. In: Ruokonen, T. (ed.) Preprints of IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS '94), Espoo, Finland, June 13 - 16, 1994. Vol. 1. Oxford: Elsevier Science Ltd. Pp. 195 - 200. ISBN 951-96042-6-X.

V Huuskonen, P. & Kaarela, K. 1995. Explaining Plant Design Knowledge through Means-End Modelling. In: Anzai, Y., Ogawa, K. & Mori, H. (eds.) Symbiosis of Human and Artifact: Human and Social Aspects of Human-Computer Interaction. Proceedings of the 6th International Conference on Human-Computer Interaction (HCI International '95), Tokyo, Japan, July 9 - 14, 1995. Vol. 20B. Amsterdam: Elsevier. Pp. 417 - 422. ISBN 0-444-817956. ISSN 0921-2647.

VI Kaarela, K., Huuskonen, P. & Jaako, J. 1993. Providing Plant Design Knowledge to the Operators, In: Smith, M. J. & Salvendy, G. (eds.) Human-Computer Interaction: Applications and Case Studies. Proceedings of the Fifth International Conference on Human Computer Interaction (HCI International '93), Orlando, Florida, August 8 - 13,

1993. Vol. 19A. Amsterdam: Elsevier. Pp. 546 - 551. ISBN 0-444-89540-X. ISSN 0921-2647.

VII Kaarela, K. & Oksanen, J. 1994. Operator Support System Based on an Information Model. In: Proceedings of the '94 Symposium on Human Interaction With Complex Systems (HICS '94), Greensboro, NC, USA, September 18 - 20, 1994. Greensboro, NC: NASA-CORE, North Carolina Agricultural & Technical State University, and SPIE. Pp. 156 - 165.

VIII Kaarela, K., Oksanen, J. & Takalo, J. 1995. An Information Model as a Basis for Hypermedia-Based Plant Documentation. Computer Networks and ISDN Systems, Vol. 27, pp. 751 - 764.

IX Huuskonen, P., Kaarela, K., Okkonen, J. & Väisänen, A. 1995. Explaining Control Logic to Process Operators. In: Forsyth, G. F. & Moonis, A. (eds.) Proceedings of the Eighth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE '95), Melbourne, Australia, June 5 - 9, 1995. Amsterdam: Gordon and Breach Publishers. Pp. 203 - 211. ISBN 2-88449-198-8.

The papers will be referred to in the text by the corresponding Roman numerals (I - IX).

The author is the main author of papers I - IV and VI - VIII. The original research ideas came from the author for all the papers. The ideas are based mainly on the work done by the author, Mr. Pertti Huuskonen, Mr. Jaakko Oksanen, and Prof. Kauko Leiviskä. The role of the co-authors has mainly been to implement and test the ideas in practice, and to comment on the early versions of the papers.

In Papers V and IX, Mr. Pertti Huuskonen has been the principal author and proposed the application of explanation techniques in clarifying design knowledge. The author has significantly contributed to the development of the approach and its application to the case studies, and the writing of the papers.

LIST OF SYMBOLS AND ABBREVIATIONS

AHP	Analytic Hierarchy Process
CAD	Computer Aided Design
CERN	Conseil Europeen pour la Recherche Nucleaire (European Laboratory for Particle Physics)
CICERO	Control Information system Concepts based on Encapsulated Real-time Objects
DR	Design Rationale
DTD	Document Type Definition
DXF	Data eXchange file Format
EDI	Electronic Data Interchange
EDIFACT	Electronic Data Interchange For Administration, Commerce and Transport
FEA	Finite Element Analysis
IGES	Initial Graphics Exchange Specification
ISO	International Organization for Standardization
KADS	Knowledge Analysis and Design Structure
KEE	Knowledge Engineering Environment (IntelliCorp, Inc.)
L3	A large electron positron collider experiment at CERN
MFM	Multilevel Flow Modeling
ODA	Open Document Architecture
QFD	Quality Function Deployment
SGML	Standard Generalized Markup Language
STEP	STandard for the Exchange of Product model data
VDI	Verein Deutscher Ingenieure

"This is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning."

Sir Winston Churchill, 1942.

1 INTRODUCTION

1.1 INDUSTRIAL PLANT DESIGN

The design phase of an industrial plant is extremely important: the designers have to make decisions that affect all the phases of the plant's lifecycle. This is illustrated in Figure 1 (VDI 2221 1987). Like any man-made system, an industrial plant is designed to fulfil specific human needs. During the preliminary study these needs, as well as other requirements such as the official regulations, are clarified to represent the design goals of the plant. The initial design phases deal with the most crucial design decisions such as the location and operating principles of the plant. Typically, the overall structure of the plant and its most important components (e.g. the turbine and the boiler in a power plant) are also fixed at this point. Design decisions in general and those during the initial design phases in particular have far-reaching consequences throughout the lifecycle of the plant. In general, a major part of the total construction costs of any product (75 - 85%) is typically committed after the initial design phases even though only some 5% have actually been spent (Nichols 1990, Sheldon *et al.* 1990). Moreover, more than half of the total lifecycle costs are committed and the product quality is determined at this stage.

Design to a great extent concerns creating models of the artefact to be built. The designers' task is to turn the requirements already set for the artefact into a design that can be implemented. Consequently, the design process produces a variety of design documents that describe the structure of the artefact and how the artefact and its parts should be manufactured and assembled. As such, the design documents most often describe only the final outcome of the design process omitting the design knowledge that led to the specific design. The functional knowledge representing the original design intentions mainly generated during the initial design phases is either vaguely and informally presented or totally missing in the documents. This kind of knowledge includes both the goals the plant was designed fulfil and its intended function, i.e. how the designers meant it to work.

Information on the design process -- design decisions, the rationale behind them, and the design history -- is usually missing in the documents. The decisions describe design issues, i.e. the design problems and the solutions that were considered. The rationale covers the knowledge concerning why each specific design alternative was chosen and why the others were not. Design histories capture sequences of design selections together with the associated rationales. In succeeding projects, this information would be

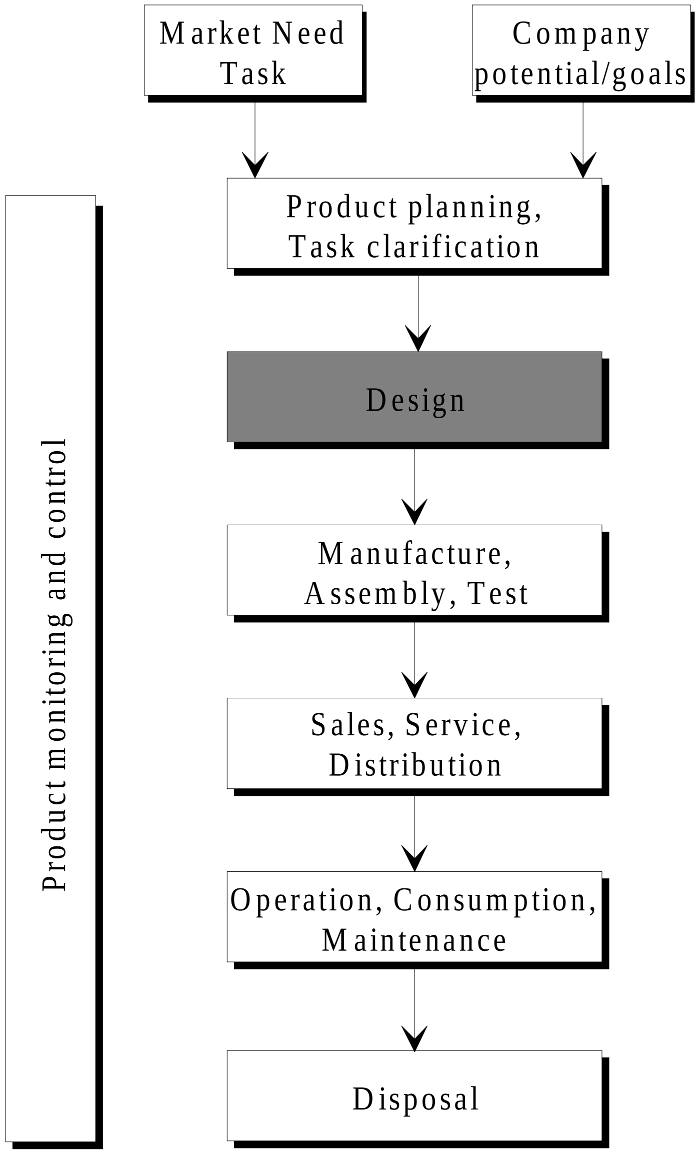


Figure 1. The significance of design in a product's lifecycle. Design produces important information for all the succeeding phases (VDI 2221 1987).

useful to see how the design evolved, to be able to reuse the best choices and to avoid poor solutions that have been examined or tried earlier.

In a study on co-ordination problems in engineering design, Crabtree *et al.* (1993) found that 24% of project delays were due to poorly documented design knowledge. In general, industrial plants are not designed from scratch every time a new plant is constructed. Substantial parts of old designs can be reused at new plants. However, reuse of old designs is not always straightforward due to missing information about why the design was produced in a specific way. Designers, especially novices, spend a lot of time searching for the design rationale of earlier designs. To put it in other words: to be able to make changes one should understand the original design. The maintenance staff of a plant meets similar challenges in their work, especially during revisions.

Design knowledge, especially that describing the intended functionality of the product, is of great importance in multidisciplinary design projects. In this research, we concentrated on the co-operation between process and automation design. The automation designers found difficulties in understanding the intended functionality of a process depicted in the diagrams produced by the process designers. This was emphasised by the lack of textual documents, such as system descriptions, during the automation design.

Functional design knowledge would also be useful for plant operators. Understanding the intended functionality of a process helps the operators to run the process more optimally and to manage process disturbances more quickly and safely. Regrettably, this information is sparsely shown in current control rooms. The procedural knowledge on how to run the process and its sub-processes that is provided in the operating manuals usually does not explain the purpose of the actions.

In this thesis, we are especially interested in plant design knowledge that can be used as a source for explanations in response to questions about the design, i.e. to provide design rationale. The primary goal of the thesis is to enhance the communication of that knowledge throughout the lifecycle of the plant and in succeeding plant design projects. A primary prerequisite is that the design knowledge be recorded in the design phase for later use[1]. This necessitates the introduction of representations capable of capturing the abstract design knowledge born in the initial phases of plant design. Another requirement is computer support for the modelling effort to allow reuse of previous designs and the associated design knowledge. The representations should be understandable to both humans and computers. In this work, we utilised a special modelling approach, Multilevel Flow Modelling (MFM), for representing formal functional design knowledge and for structuring informal design information related to both the artefact and the design process. A final point in getting plant design knowledge into effective use concentrates on the end-user's point of view: how design knowledge can be displayed to the users of the plant in order to assist them in their work.

---

[1]We want to emphasize that despite the inadequacies in recording design knowledge industrial plants are successfully designed, constructed, maintained, and operated. The ideas and solutions presented in this thesis are aimed at improving the current state of affairs.

1.2 MULTILEVEL FLOW MODELLING

The key idea of MFM is to model the plant as a man-made artefact designed for specific purposes (Lind, 1982). In MFM, these purposes are modelled with the concept of *goals*. The goals state *why* the system is being used. There are three types of goal: *production, economy, and safety*.

A second important concept of MFM is *function*. In MFM, functions are seen as the means to *achieve* the goals. The functions of the plant are represented by a set of mass, energy, and information flow structures at several levels of abstraction. The physical components of the system, the *devices,* are used to *realise* one or several functions.

MFM combines *whole-part* and *means-end* abstraction hierarchies for structuring systems (Lind, 1990). The whole-part hierarchy decomposes a plant into parts using the part-of relationship. It can be used to describe the relationship between a goal and its subgoals, between a function and its subfunctions, and between a physical system, its subsystems, and components. The means-end hierarchy divides the plant into levels of abstraction, with abstract concepts on the top and concrete concepts at the bottom. Means-end relations -- such as achieved-by, achieved-by-control, condition, and realised-by -- are used to represent the relationship between objects at different levels of abstraction. In process engineering, the lower levels tend to represent the physical components of a plant and the higher levels their abstract purposes.

MFM derives its power largely from means-end information, and enhances it by the concept of flows and their balances. At several levels of abstraction, energy, material, and information flows are recognised. A number of balances are defined for the mass and energy flows, allowing one to reason about the behaviour of the system by looking at the balances. The information flows provide a basis for designing the automation system as well as the operators' tasks.

MFM can be utilised for capturing functional design knowledge. It offers a formal language for *explicitly* representing the goals of the system, the functions required to achieve the goals, the devices needed to realise the functions, and their relationships. An MFM model represents the design as a meaningful artefact, where every single function and device has a purpose or role in the achievement of the overall goals of the designed system. Such a presentation provides a valuable description of the whole that is typically missing in current design documents. The benefit is obvious: the model can be used to explain the intended operation of the modelled system and the purpose of its components. It is important to note that MFM models are normative, describing how the system was intended to work, not how it is actually working. Such models can be used to detect deviations from the intended operation but not to predict the future state of the system.

1.3 RESEARCH PROBLEM

Design knowledge -- the intentions behind a design -- tends to be lost because it is seldom recorded in everyday projects. There are many reasons why design knowledge is not recorded:

1) ***Current design practice.*** In several companies, documentation is seen as the 'inevitable evil' that *has to* be done with additional costs. The documents are hastily written just in time before (or even after) the delivery deadline, long after the actual design work. The content and quality requirements of the documents are often neither specified nor controlled in any way.
2) ***Limited expressive power of current design representations.*** Current design representations are limited to describe the end result of the design process -- the artefact -- in terms of implementation details only. Most of the representations are informal in the sense that their semantic content is not understandable by computers and thus do not allow for automated reasoning on the included knowledge.
3) ***Lack of supporting tools.*** Current design tools do not support recording (or reusing) design knowledge.
4) ***Lack of motivation.*** Designers are generally not motivated to record design knowledge: they regard it as extra work that is of no use for themselves (which is not true; the design knowledge would help them in subsequent projects) or they may even fear for losing their expert status.
5) ***Lack of valuation.*** The value of proper documentation (containing design knowledge) as a corporate knowledge base is not often seen. Furthermore, the value of proper documentation as a factor of the quality of a product (from the customer's point of view) is not always acknowledged. The importance of design knowledge in the operation and maintenance of the plant is seldom recognised.

In this research we develop representations that make plant design knowledge explicit, tools supporting capture of design knowledge during the design process, and ways to use the design knowledge both in interdisciplinary design projects and during the use of industrial plants. In our research we have adopted the use of Multilevel Flow Modelling as a framework for structuring and recording design knowledge. The models form the backbone of our research on presenting the knowledge to the designers, maintainers, and operators of industrial plants.

From the discussion above we derive our research problem:

> *How can plant design knowledge be structured and recorded in order to enhance its communication in design projects and at industrial plants?*

This problem definition is rather broad and therefore needs to be further refined into the following research questions:

> *Q1. Is it feasible to use MFM for structuring plant design knowledge?*
>
> *Q2. What kind of techniques are required for presenting this knowledge to plant operators?*

1.4 RESEARCH ASSUMPTIONS

In this thesis, we are dealing with design knowledge -- "the knowledge that led to a specific design (solution)". It is obvious that the subject, the designer, has a major role in producing this information. After all, it is most often the designer who makes the design decisions and is responsible for documenting them. This notion leads us to the following assumptions:

1) ***Designers act rationally.*** This implies that designers should be aware of design goals, able to identify the design alternatives and criteria, and able to evaluate the alternatives using the selected criteria.
2) ***Motivation.*** It is possible to motivate designers to record design knowledge for later use, even though they are generally not willing to change their working habits, especially when the change seems to imply more work.

1.5 HYPOTHESIS

The design, maintenance, and operation of a complex system, such as an industrial plant, are demanding tasks that could be assisted by providing the people involved with design knowledge in addition to documents describing only implementation details. We propose the use of multilevel flow modelling for organising the design knowledge. We advocate the importance of representations that enable the use of the design knowledge in various phases of a plant's lifecycle.

We state this as our research hypothesis:

> *Multilevel flow modelling can be utilised for effectively structuring and recording plant design knowledge during the plant design process, and for presenting the design knowledge to operators of the plant in order to improve the understandability of the designs.*

We can specify the work needed to validate the hypothesis as follows:

1) ***Problem analysis:*** Identify design knowledge that is not recorded in current design practice but could be acquired during the design process.
2) ***Design knowledge representation:*** Develop MFM-based representations for plant design knowledge.
3) ***Tools supporting design knowledge capture:*** Develop MFM-based tools to support the structuring and recording of design knowledge.
4) ***Evaluation:*** Apply the newly developed representations and supporting tools to real-life test cases in order to evaluate their usefulness.
5) ***Operator support systems:*** Develop and demonstrate new ways to present design knowledge to plant operators.

1.6 RESEARCH METHODOLOGY

1.6.1 Research strategy

The approach of this research is that of constructive research. As depicted in Figure 2, it can be divided into the following three stages:

1) ***Problem Analysis:*** We analyse the operation and maintenance of industrial plants, concentrating on the information needs of the plant operators when performing their tasks. We analyse the design process of an industrial plant in order to determine how communication of design knowledge could be improved from current design practices and what additional methods, tools, and representations are needed for this purpose.
2) ***Construction:*** Based on the results of the problem analysis and a study of modelling techniques, knowledge representation techniques, and existing approaches, we develop an approach to the structuring and recording of design knowledge as a part of a designer's routine work. This approach includes the utilisation of MFM as a framework for recording design knowledge and tools supporting the modelling effort.

3) ***Demonstration:*** We validate our solutions in two industrial case studies. We propose new ways to represent the design knowledge to the plant staff in order to support them in their tasks.

As shown in Figure 2, the methods used for acquiring knowledge of the current design practices (Problem analysis) were eliciting theoretical background information on plant design from textbooks, a literature review, a case study consisting of unstructured interviews of the plant staff and process and automation designers, and analysis of the plant design documentation augmented with walkthroughs with the designers.

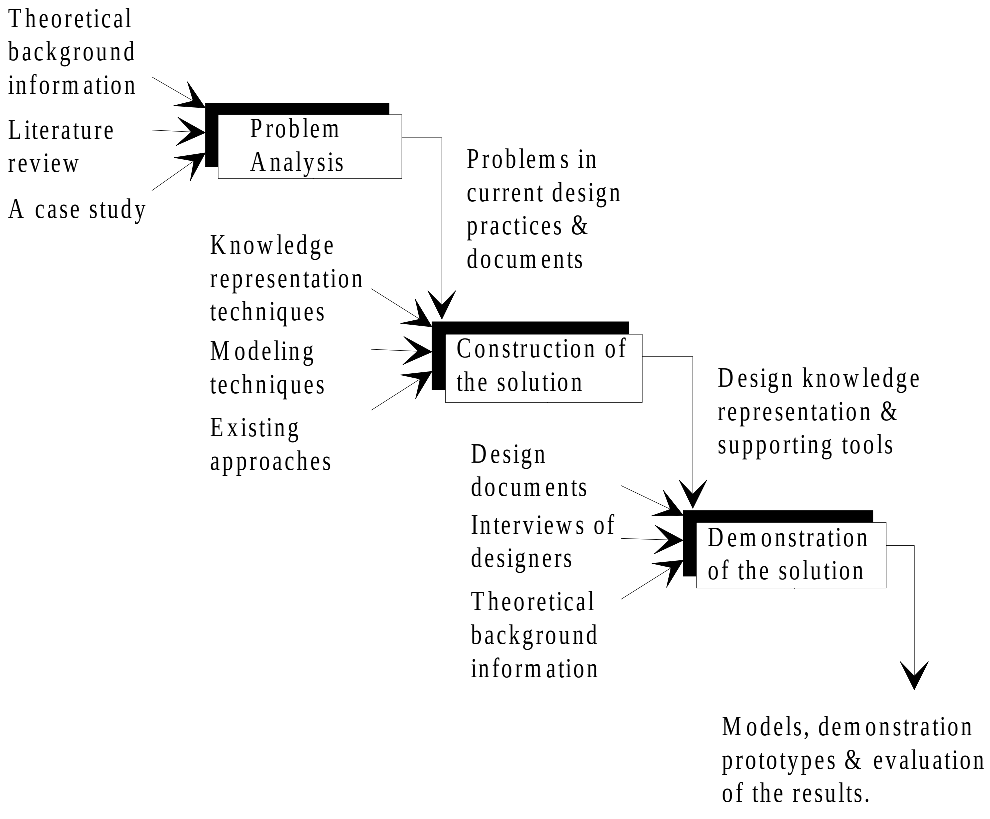


Figure 2. The research strategy followed.

1.6.2 Scope of applicability

This research is related to design of industrial plants. It is, however, not targeted at developing the design process, per se. The scope of the research is to develop and apply representations expressive enough to present the concepts of the domain of plant design in order to make the design knowledge explicit and to improve its communication to other people involved during the life cycle of the plant.

Currently, the design phase of an industrial plant produces a variety of documents describing the plant and its subsystems. These documents are most valuable in describing implementation details. The idea of this research is *not* to replace those representations but to extend the documents with the *means-end* aspect, in order to improve the quality and expressive power of design documentation in general and the *management of the whole* in particular. This is necessary from the viewpoint of making design knowledge usable during the lifecycle of the plant.

We concentrate on the technical aspects of organising design knowledge during the plant design process. Our research should be judged with the following restrictions in mind:

1) ***Modelling technique.*** In this work, we are not developing the modelling technique. MFM is used on an "as-is" principle. We have mainly utilised the two hierarchies provided by the modelling technique: whole-part and means-end. Modelling the normative behaviour of the plant using the mass and energy balances of the flows and using those models to detect process deviations is not specifically addressed in this thesis.
2) ***Modelling method.*** Multilevel flow modelling provides a way to represent an artefact but it is not a modelling *method* in the sense that it would support or guide the modelling process. A guideline for modelling is, however, presented by Lind (1990). Regrettably, it does not provide a clear procedure on how to recognise the goals. Developing such a method is out of the scope of this research, even though we make some suggestions on this topic in Paper IV.
3) ***Application field.*** Multilevel flow models are well suited for representing *process and control engineering* aspects of *continuous* processes due to their ability to describe flows. The flow functions are meant for representing mass, energy, and information flows in the system. This limits the scope of applicability of the modelling technique: it is most suitable for representing processes dealing with real substances. In this sense, our case studies were optimal targets for MFM, the first one representing an energy process and the second one a process with gas flows.
4) ***Experiential knowledge.*** As stated in Paper I, the authors are fully aware that design knowledge is not enough for the plant operators to perform their work. This is because the actual operation of the system cannot always be predicted at design time and may differ from the designed operation. Experiential knowledge of the operation of the plant is thus required. This research is limited to storing and representing design knowledge. In Chapter 6, however, we show how the structure imposed by the multilevel flow model of the system can be used as a framework to effectively store and retrieve experiential knowledge.

5) ***Managerial aspect of design knowledge.*** We do not address the managerial aspect on how to control the timescale, costs, and quality aspects of design projects even though we are aware of its necessity in real-life projects. However, improved design documentation will serve the needs of the management by providing information on what tasks have been performed.

6) ***The organisational aspect.*** We concentrate on the technical aspects of recording and utilising design knowledge. We do *not* address the problem of how to introduce into an organisation a novel design approach that inevitably requires changes in the current practices both at the individual and organisational levels.

1.7 RESULTS

This work contributes to improving the design, use, and maintenance of industrial plants through recording and representing design knowledge in a systematic way.

The results can be summarised as follows:

❑ Needs analysis. An analysis of the operation and maintenance of industrial plants concentrating on the information needs of the plant operators when performing their tasks. An analysis of the design process of an industrial plant, in order to find out what should be improved in current design practices and what additional methods, tools or representations are needed.

❑ An approach and supporting tools for structuring and recording design knowledge as a part of the designers' work using MFM as a framework.

❑ Design and implementation of a design environment that makes the design knowledge explicit and assists the designer in reusing existing designs.

❑ Demonstration of the use of the approach and the supporting tools in two case studies.

❑ Design of new techniques for presenting the design knowledge to the operators of industrial plants. Implementation of two operator support system prototypes based on the knowledge captured in the case studies.

The results can be viewed as a framework for systematically recording design knowledge at design time in order to improve the understandability of the designs both from a functional and a (design) decision-making point of view at reuse (redesign of similar artefacts), maintenance, and utilisation phases. We demonstrate the use of the approach with two case studies to validate its applicability. In both cases, a subsystem of an industrial plant was modelled using MFM and the tools developed in this work. The first case was a feedwater system of a power plant, and the second a gas system of a detector experiment at CERN (from the point of view of complexity the detector experiment can be regarded as an industrial plant). The emphasis was on capturing functional design knowledge.

The knowledge acquired through modelling was used as a major information source when implementing the operator support systems. The design knowledge stored in the object hierarchy of the design tool was automatically converted to a representation required by the operator support systems, using the facilities implemented in the design environment.

The results are discussed in more detail in the following chapters and in the original Papers I to IX included in the appendices.

1.8 OUTLINE OF THE DISSERTATION

This thesis concerns enhancing the communication of design knowledge, i.e. how to capture such knowledge during the design process and how to present it not only to other designers but to operators of the designed artefact. The structure of this dissertation is the following:

- Chapter 1 states the research problem and gives a brief introduction to its background.
- Chapter 2 introduces the concepts related to design that are used in this thesis. The essence of design knowledge is discussed.
- Chapter 3 contains a brief survey of design representations and design tools. Problems of current design practices are discussed.
- Chapter 4 gives an introduction to multilevel flow modelling and explains how design knowledge can be captured using MFM as a framework.
- Chapter 5 shows the features of a design environment that is capable of capturing the design knowledge targeted in this research. Reuse of the design knowledge is discussed from the designers' point of view.
- Chapter 6 discusses how to present design knowledge to the operators of industrial plants. The chapter describes the principles and implementation of the operator support systems built in the two case studies.
- Chapter 7 gives an introduction to the original Papers I - IX that form the basis of this dissertation. The papers are included as Appendices 1 - 9.
- Chapter 8 sums up the work done in this research and draws conclusions about its relevance. Directions for further research are presented.

2 ENGINEERING DESIGN

"Engineering design is a purposeful activity directed toward the goal of fulfilling human needs, particularly those which can be met by the technological factors of our culture." (Asimov 1974)

M. Asimov states that "A designer does not usually produce the goods or services which immediately satisfy a consumer's needs. Rather, he produces the model which is used as a template for replicating the particular good or service as many times as required" (Asimov 1974). Hence, design appears to be about the creation of descriptions (i.e. models) of physical artefacts that perform the desired functionality. The design process produces several kinds of models of the artefact for different purposes such as analysing the design (e.g. simulation models) and manufacturing the artefact proposed by the design (e.g. CAD drawings). Unfortunately, these models generally represent only the outcome of the design process. However, there is a lot more information related to design that both could and should be captured and utilised.

Design can be regarded as a transformation from the functional domain to the physical domain (Mostow 1985). Thus, design is to a great extent reasoning about function; *how* to achieve the desired functionality. Yet, these considerations are seldom represented in design descriptions that depict only *what* is in the design.

A design problem can formally be regarded as a search problem in a large space for objects satisfying multiple constraints. Only a limited number of objects in the problem space provide satisfying solutions to the problem (Newell & Simon 1972, Brown & Chandrasekaran 1989). Given an initial state, the problem is to achieve a goal state, i.e. to find a path of design moves (transitions) in the space leading from an initial state to a goal state. Generally it is not feasible to perform a design task through exhaustive search. Domain knowledge and heuristics can be used for drastically reducing the search space. In this research, we view design as a process comprising of successive elaborations of the design descriptions. Each elaboration refines the design from the abstract to more concrete or from the general to more detailed level until the final design is achieved. The information related to this process -- what decisions were made, *why* and under what circumstances they were made, and what alternatives were considered -- provides a sound basis for understanding design decisions.

Design is often divided into three classes depending on the degree of novelty present in the design (Pahl & Beitz 1988, Brown & Chandrasekaran 1989):

- ❑ Original design, where the designer has to elaborate an original principle for implementing a system.
- ❑ Adaptive design, which involves adapting a known system to a changed task while the principle of solution remains the same.
- ❑ Variant design, which involves varying the size or arrangement of a system.

According to Brown and Chandrasekaran, original design is open-ended, *creative* design that calls for major new inventions. The average designer seldom faces tasks requiring original design (Brown & Chandrasekaran 1989). The tasks of designers most often fall into the categories of adaptive and variant design, the distinction between which cannot be precisely fixed. In these categories a powerful problem decomposition already exists, and due to the stable technology basis in process industries only a limited number of alternative solutions are available to the subproblems. In adaptive and variant design, prior designs or versions of a design form a major source of knowledge. In addition to problem decomposition they can provide optimal, necessary, or acceptable solutions to the subproblems and information on previously tried "false turns", since they describe how the design evolved over time.

In fact, a great part of all industrial engineering can be regarded as redesign, i.e. making changes or improvements to earlier designs. In the process industry, these changes primarily take place when a new plant is being designed or during plant revisions. Regularly, earlier designs are used as a basis for new or modified plants. In many cases, essential improvements in design quality and productivity can be obtained. Reuse of earlier designs also allows novice designers to apply proven solutions produced by design experts in their work.

Reuse is often limited to the reuse of existing designs - not the knowledge behind them. This is a natural consequence of the fact that design knowledge is generally not recorded. This can be dangerous for two reasons. First, there is the problem of how to recognise a similar case among earlier designs without proper information. Applying old solutions to implicitly similar problems is much based on the art of good guessing. Second, two different phases should be distinguished in the revision of earlier designs: understanding and making changes (Fischer *et al.* 1991). Without understanding the original design, the designer is most probably unable to estimate all the effects of a change.

We are interested in the essence of this design knowledge, both the hows and whys, that leads to a specific design. The next subchapter is devoted to discussing the concept of design knowledge.

2.1 DESIGN KNOWLEDGE

Design knowledge is often referred to as 'the knowledge that led to a specific design' or as 'the designer's intentions behind a design'. Briefly, design knowledge is the expertise designers apply while making design and implementation decisions. We share the view of Gruber and Russell (1990) that design knowledge can be utilised as a source for explanations in response to questions about the design, providing design rationale (DR).

In this research, design knowledge is classified into two categories: *artefact-related* and *design-related* (Gruber & Russell 1992, Chandrasekaran *et al.* 1993). It is essential to note the distinction between these two knowledge categories: the former explains a design (i.e. the artefact) through its functionality whereas the latter describes the design *process* that led to the design. As a source for design rationale this approach is broader than the one generally applied within the DR research community where most researchers concentrate solely on the design process (Lee & Lai 1991, Conklin & Yakemovic 1991, Fischer *et al.* 1991).

We acknowledge the existence and necessity of the managerial aspect of controlling design projects but it is outside the scope of this research. However, the managerial aspect does have some influence on the other knowledge categories. The management of a design project deals with controlling the timescale, costs, and quality of design projects. Quality assurance has a close relationship with the work done in this thesis. The quality of design work is typically assured by following guidelines that specify how the design work should proceed. The guidelines should explicitly identify both the phases of the design process and the results of each phase. Typically the results are reviewed in technical meetings as a part of the project co-ordination activities. The work presented in this thesis could support the managerial aspect in the sense that improved documentation enhances communication of the project's status to the management.

Another distinction is made regarding the formality of the design knowledge. In this research, design knowledge is considered formal if it is represented in a way that allows a computer to reason using it. It is important to note that a typical CAD diagram does not fulfil this requirement. Even though the diagram is syntactically formal in the sense that it includes a set of unambiguous symbols that can be connected according to a predefined syntax, its semantics is not interpretable by the computer. Another informal representation for design knowledge is written text. However, text can be made semi-formal, i.e. partly processable by the computer, by introducing a structure into design documents.

2.1.1 Artefact-related design knowledge

Initially, any man-made system is designed with one or many specific purposes in mind, typically to fulfil specific needs. The system to be designed has specific functional and structural requirements. The designer's work is to turn these requirements into a design that can be implemented. The end result of the designer's work, the design documents, thus typically reflects the implementation issues such as the manufacture and assembly of the system. The design documents do not lay out a trail from the design to the requirements showing why the system was designed in a specific way.

To define artefact-related design knowledge we have to distinguish between *function design* and *form design*. This dichotomy is recognised in the work of several researchers, especially those emphasising a systematic approach to design (Pahl & Beitz 1988, VDI 2221 1987, Suh 1990).

Pahl and Beitz describe function as "the general I/O relationship of a system whose purpose it is to perform a task" on the basis of its *inputs* and *outputs* that can be *material*, *energy,* and *signals* (i.e. information). The definition given by Suh (1990) states that "function is something we want to achieve" in design while form (i.e. the physical solution) is "how we want to achieve it".

The concept of form should be understood in a wide sense as the physical implementation of the artefact. As such, it includes the structural description of the solution, topology, geometry, and material. This knowledge is often referred to as *structural* knowledge.

Systematic approaches to design generally propose a phased design process. They all consist of similar design phases even though the number and naming of phases are somewhat different. Figure 3 shows the approach proposed by Pahl & Beitz consisting of clarification of the task, conceptual design, embodiment design, and detail design. The early design phases (especially conceptual design) are more associated with function design while embodiment and detail design mainly deal with form design.

In task clarification, the requirements of the system and the design constraints should be identified. We emphasise that the requirements should be quantified, if possible, and further refined to explicit goals of the system. Conceptual design generates a function structure that is refined into a detailed design during the embodiment and detail design phases. Deriving a function structure from the requirements necessitates design knowledge: the designer has to apply underlying knowledge of whether the proposed functions are realisable within a specific technology (Leppälä 1995).

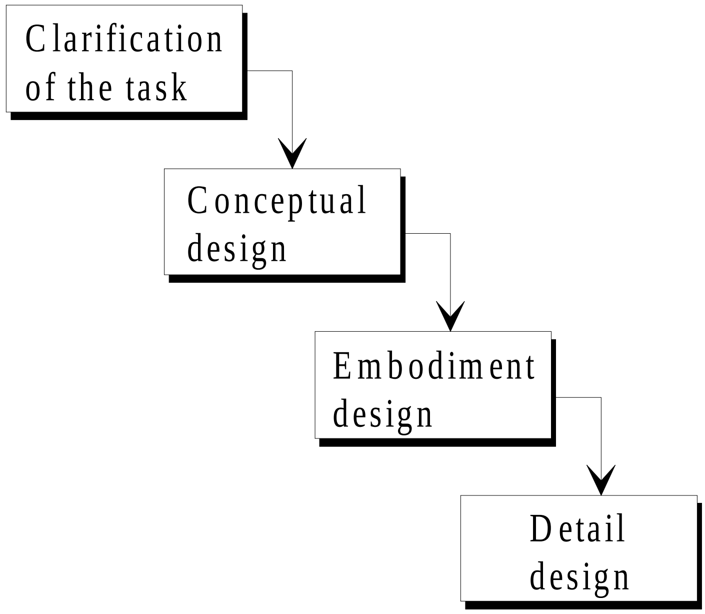


Figure 3. The main phases of design (Pahl & Beitz 1988).

Now we can define another class of artefact-related knowledge, *functional* knowledge, as a *mapping* among the functional requirements or rather the design goals of the system, the overall function structure, and the detailed design representing the form aspect. Functional knowledge explains a design through its intended function, i.e. how the designer meant it to work. The meaning or purpose of any part in a system can be explained via the function it was designed to accomplish in the system.

Structural and functional knowledge represent *declarative* knowledge about an artefact. Another important class of design knowledge is *behavioural* knowledge. The system, its subsystems and components can be in various *states* representing their behaviour. For an industrial plant such principal states are stand by, stopped, startup, normal operation, shutdown, and disturbance[2]. These states may have different goals or at least the priority of the goals can be different in the various states. This implies that the role of the functions (and the devices) may become dependent on the state of the system. Behavioural knowledge, consisting of the states of the process, the conditions for state transitions, the actions together with their conditions and their order of execution, timing, and possible disturbances, forms a basis for specifying how to operate the process. This information is extremely important for the automation designer.

[2]Due to their complexity and relatively long duration startup and shutdown are often regarded as states of their own even though they essentially are state transitions.

Decomposing a design problem too complex to be solved by an individual designer into smaller subproblems is a common way of managing complexity in engineering. Decomposition implies that the functional (and other) requirements of the original problem need to be parcelled out to the specifications of the subproblems. A number of alternative decompositions for a problem may exist, in which case a selection has to be made. *Decomposition* knowledge -- D -> D1, D2, D3, ... Dn, where D is the given problem and Di's denote the smaller subproblems -- represents a kind of design knowledge since it can be reused during future design of similar artefacts where it may help reduce the size of the search space (Brown & Chandrasekaran 1989, Chandrasekaran 1992). A typical form of decomposition knowledge is present when functional specifications can be decomposed into a set of subfunctions (Freeman & Newell 1971). *Design plans* are a special case of decomposition knowledge; in addition to the decomposition they specify a sequence of design actions to take for producing a piece of abstract or concrete design (Brown & Chandrasekaran 1989, Chandrasekaran 1992).

2.1.2 Design-related design knowledge

Designers are usually preoccupied with creating an artefact that achieves a desired functionality. If the same functionality can be achieved using various alternatives, design involves selecting one of several functionally equivalent design alternatives (that already exist at some level of description), i.e. making design decisions (Kannapan & Marshek 1992). While making the decisions, the designers have to consider many other aspects related to the artefact such as its costs, manufacture, assembly, testing, service, and operation.

The design process involves a series of interdependent design decisions that contribute to the development of the final design. Design-related knowledge deals with the design process itself: both design decisions and the underlying knowledge. The knowledge underlying these decisions is a major source of design rationale answering the question of why a specific alternative was preferred or rejected.

Before having a look at design as a decision-making process and the related knowledge, it is useful to state the concepts related to the design process which are used in this work. The concepts presented in Table 1 originate (although slightly modified) from the ontology of design defined by Gruber and Russell (1990).

Table 1. Characterisation of the concepts related to the design process.

Artefact:	A man-made system that has been designed with a specific purpose in mind.
Design:	An intermediate or end result of the design process; a description of a designed artefact.
Design alternative:	A proposed design (solution) meeting the design constraints and requirements of the design issue at hand.
Design constraint:	A limiting factor that has to be taken into account in the design work. Constraints may pertain to the parameters (e.g. physical properties such as weight), to the design process (maximum duration, cost-efficiency, etc.) or to manufacturing, assembling, testing, and using the artefact.
Design criterion:	A feature of the design alternatives that can be used in their evaluation in order to choose the best fit.
Design decision:	A choice among design alternatives. A description of a design decision should specify the design issue, the set of alternatives considered, relevant criteria, the evaluation function, and the outcome of the decision.
Design goal:	A goal or objective that the artefact under design has to achieve.
Design intention:	A typically tacit design goal of the designer.
Design history:	A record of design decisions.
Design issue:	A description of a design problem for which a solution is searched among design alternatives in order to make a design decision.
Design justification:	An explanation of why a specific alternative was chosen with respect to a set of criteria.
Design knowledge:	Knowledge underlying the design process that led to a specific design.
Design move:	An action in the design process that produces a change in the design.
Design process:	The process of creating and refining a design.
Design rationale:	An explanation (based on design knowledge) of how and why an artefact is designed the way it is.
Design requirement:	A functional or structural requirement regarding the artefact.
Evaluation function:	A function used for evaluation of design alternatives with respect to a set of design criteria.

Figure 4 shows a simplified model of design as a decision-making process.

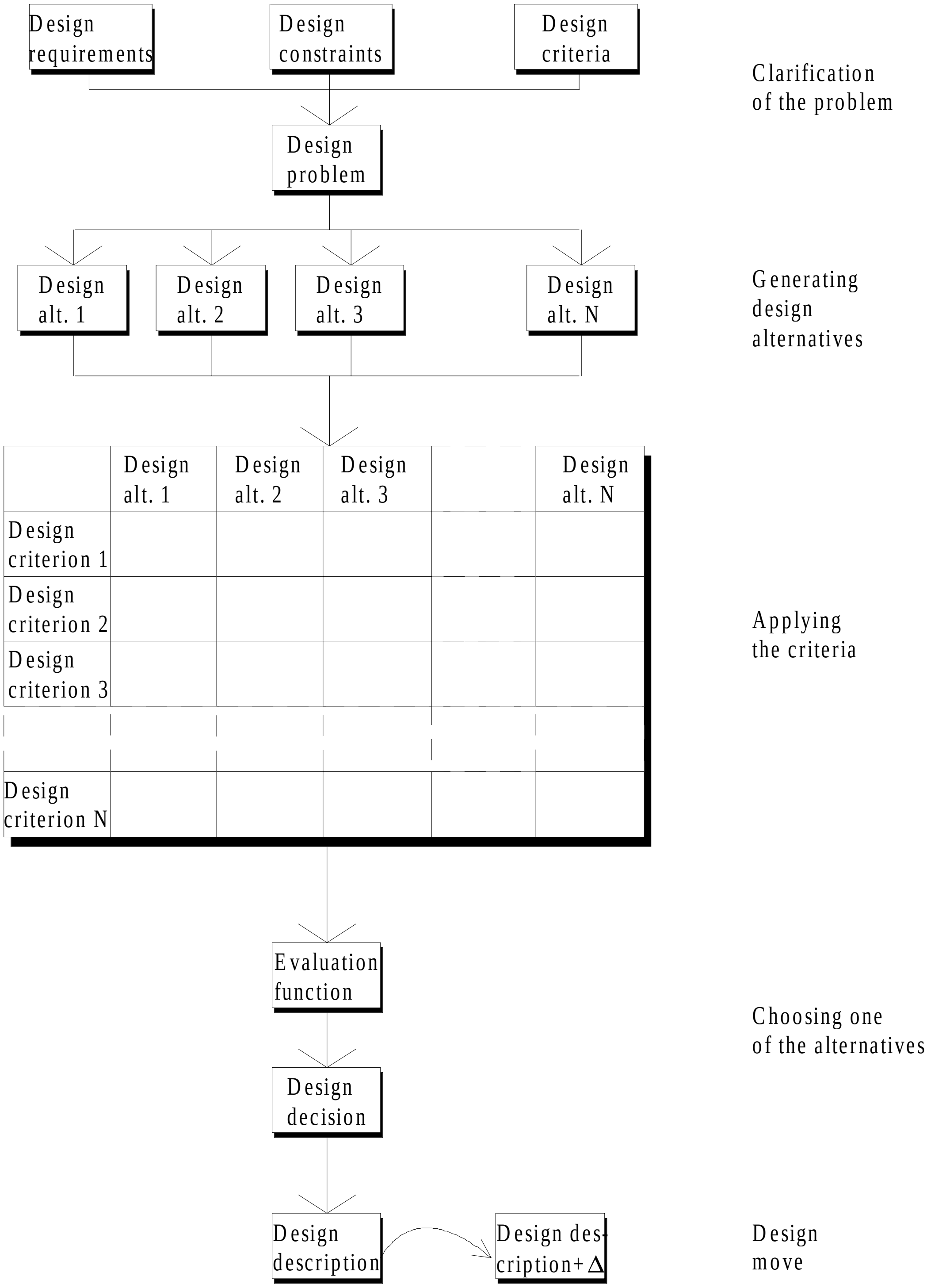


Figure 4. Design problem solving.

Design requirements and constraints are refined into a problem definition explicitly expressing the design goal and constraints. Based on design knowledge, design alternatives fulfilling functional requirements are generated and evaluated against a set of design criteria. Design constraints limit the space of possible solutions by ruling out certain choices. A design decision is made upon the evaluation of the alternatives using an evaluation function. Finally, a design move causing a change (Δ) in the design descriptions is performed.

In a (design) decision situation, designers may have two or more alternatives to choose from. To make a rational decision, they have to be able to evaluate and put the alternatives into an order of preference using a set of design criteria. The concept of utility is often mentioned in this connection if a numeric value can be applied when comparing the alternatives. A decision that maximises the utility is regarded as rational (Luce & Raiffa 1957, Gruber & Russell 1992).

Usually the initial requirements of a system will contain *design constraints* in addition to the functional and structural requirements. Design is all about trying to create an artefact that satisfies this set of functional requirements and constraints. Since some of the constraints may relate to the properties of the artefact (such as portability), the difference between a constraint and a requirement is sometimes hard to distinguish. We make the following distinction between constraints and requirements (functional or structural) as proposed by Chandrasekaran (1992):

- ❑ Functional requirements refer to the fulfilment of the system's technical function, and are the primary reasons why the artefact is being designed and implemented.
- ❑ Design constraints limit the space of potentially functionally adequate solutions. They can refer to the properties of the artefact, its manufacturing or testing, its use, or the design process itself.

Functional requirements and design constraints also differ, in that inability to achieve the requirements could lead to abandoning the design endeavour. Often all the constraints can not be met, and the designer has to make selections between constraints to be satisfied. This requires relaxing some of the constraints and making compromises.

Decomposition seldom leads to independent subproblems. Interactions occur between alternative design choices for the subsystems. Selections made in one subsystem may restrict the choices in another. Sriram *et al.* (1992) have separated these constraints into *causal* and *interaction* constraints. Causal constraints stem from the physics of the system and describe the physical interaction between the subsystems. Interaction constraints include constraints on information flow, compatibility of materials, and geometric alignment constraints.

According to Seppänen (1990) three levels of recording design-related knowledge may be used, as depicted in Figure 5. At the first level, only the initial requirements and constraints (C_{init}) as well as the intermediate and final designs (D_i) are recorded. The second level recognises the design process; it records a *design history* comprising a sequence of design decisions (S_i) that led to the final result. The third level captures not only the successful design sequence (the shaded area in the figure) but also the alternative designs at any stage. Such an *extended design history* captures a design process as a sequence of design decisions (S_i) with the underlying knowledge: the design rationale (R_i). An extended design history may also include information on dead-ends due to poor design decisions, and the respective backtrackings.

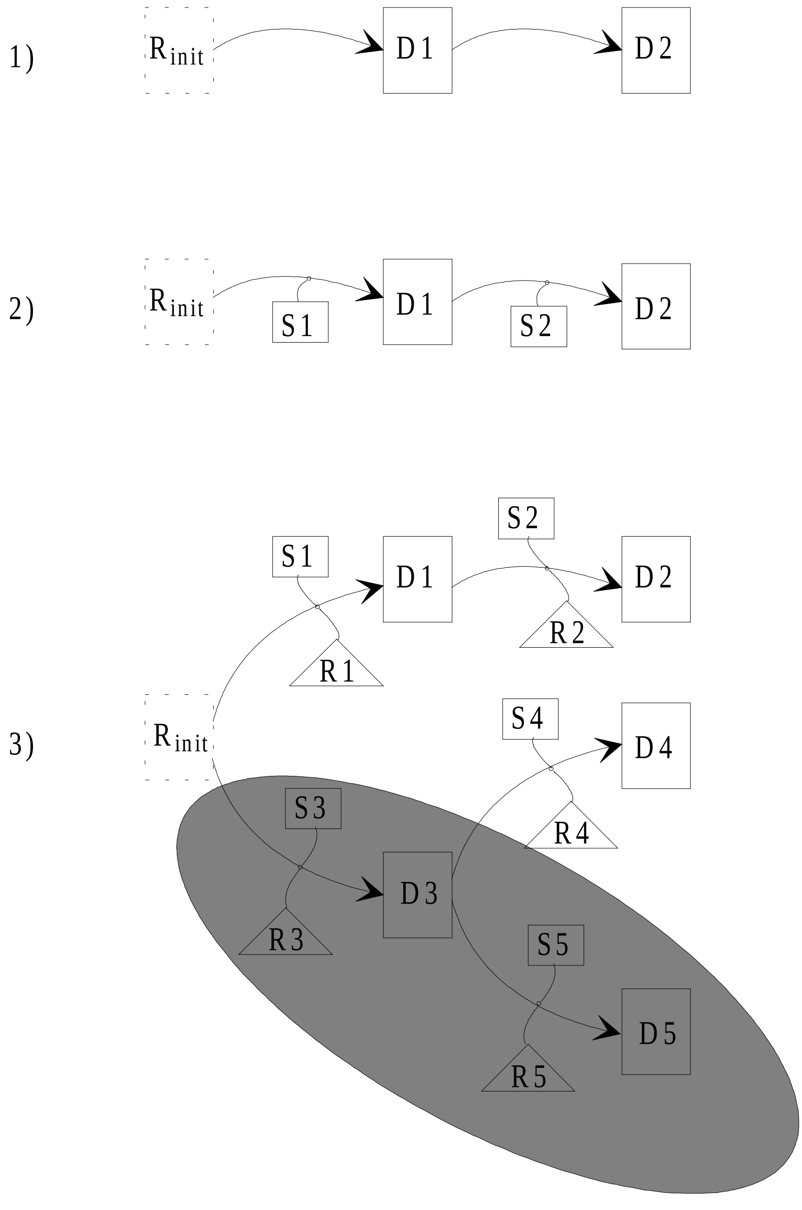


Figure 5. Three levels of recording design-related knowledge.

The model of the design process presented in this section is idealistic. It assumes that designers have all the required information at hand when making decisions. Holsapple and Whinston (1992) conclude that decisions made by engineers and administrators range from structured to unstructured. Structured decisions require that all the knowledge needed to make a decision is available in an easy-to-use form. However, this is not always the case in reality. Some issues pertinent to producing a decision may be unclear or unknown to the decision maker which makes the decisions unstructured. The alternatives may be vague, difficult to compare, or difficult to evaluate with respect to the organisation's goals.

Protocol studies on designers' work conducted by several researchers have indicated that designers usually pursue a single alternative at a time instead of systematically considering several solutions (Akin 1979, French *et al.* 1993). Sriram *et al.* (1992) have concluded in their studies that this results from the inability of humans to retain several alternatives in their memory, and that the situation could be aided with proper computer support. According to French *et al.* (1993), designers do not on the whole use such systematic approaches but act rather opportunistically. Designers use their wealth of experience, and in many cases are able to go directly to the results. Prins & Olthoff (1993) state that designers get and reject a lot of ideas while performing a design task. To document all the rejected ideas -- the false turns -- would require too much work and is seldom done.

Another limitation of the presented simple model of the design process is that is does not explicitly consider the iterative nature of design: design is a stagewise process progressing iteratively through conceptual design, embodiment design, and detail design. In general, design is not a straightforward process starting from some set of initial requirements and constraints and progressing step by step to a final solution. In reality, design involves evolution of the requirements, constraints, and the design descriptions, and even discovery of new goals.

2.2 INTERDISCIPLINARY DESIGN PROJECTS

So far we have discussed design as a task of mapping functional requirements to the overall function structure and, ultimately, the implementation and also as a process of sequential design decisions. In this section, we enlarge our view to large design projects and the problems due to the involvement of a number of designers simultaneously working in the same project.

The design of a complex system, such as an industrial plant, is a huge effort requiring hundreds of person-years of work. To manage the complexity, the design problem is typically decomposed into subproblems to be allocated to individual designers, teams, divisions, and organisations. Such design projects are also interdisciplinary, i.e. they necessitate expertise in various fields of engineering: process design, electrical design, automation design, organisational design, safety design, etc. Other critical elements for a successful design project are co-ordination of design activities among the groups of designers working on different subproblems of the design and communication between designers.

Decomposition is clearly a practical means for successfully designing a complex system. However, problems related to the management of the whole still remain (Korhonen 1991, Riitahuhta 1988). For example, the total optimisation of a boiler plant is still at the same state as it was in engineering design in the 1960's (Riitahuhta 1988). Design typically begins with a vague notion of the desired functionality and general requirements (Green 1992).

In many fields of process engineering there are a limited number of established, alternative ways to implement a plant, i.e. to decompose a plant into its rather standardised subsystems. The functional structure of the plant is therefore fixed in the early stages of the design process and designers are allocated to work on its subsystems. However, information on either the goals of the plant as a whole or the rationale behind the functional decomposition is hard to find in the design documentation. Such information would be valuable for reminding the designers where the requirements come from (Korhonen 1991). Thus, designers generally tend to ignore the global effect of their design and concentrate on searching for optimal solutions to local design problems. However, the global solution is often more than the sum of the local solutions.

Early work on design assumed that design is decomposable with little or no interaction between the subproblems (Simon 1981). Real plant design projects are seldom this simple. The assumption of non-interacting or weakly interacting subproblems rarely holds. In reality, work in any design discipline is strongly dependent on the results of other design disciplines: a solution in one part of the design often constrains the selection of feasible design alternatives in the other parts. As a result, a change in one part may necessitate changes on several other portions of the design. Green calls this the design-in-the-large syndrome (Green 1992). Even the design goals may alter or new goals may be discovered in the course of a design project.

The design-in-the-large view of the design process emphasises the role of co-ordination and communication in large design projects. Co-ordination is required to ensure that subsolutions produced by individual designers or design teams can be integrated into a whole that fulfils the requirements set for the system. Lack of co-ordination can cause significant useless iterative work and costs, and generally results in a suboptimal solution. Co-ordination activities include design reviews where both managerial (milestones, costs, resources) and technical issues are covered. Design documents have an important role in co-ordination: they communicate the status of design to the management and other interested parties.

Ideal co-operation is difficult to achieve in real plant design projects and several kinds of co-ordination problems exist (Crabtree *et al.* 1993). Activity management problems stem from the difficulty of keeping a large project on schedule. However, a schedule is always based on assumptions of the workload of the tasks and availability of resources. In reality, tasks may be more demanding than expected and designers may be unavailable. The different design disciplines generally do not advance at the same pace. Due to the interdependence of the design tasks, a delay in one part of the project may cause delays in others as well. Inability to make major design decisions due to either unavailability of the key designers or a delay in one part of the design task may leave a large group of engineers idle. Some of these problems are caused by poor design discipline: schedules are not met; the design documents contain premature information that is bound to change; they do not contain the required information or they are not delivered at all; etc.

A primary prerequisite for co-operation is that designers can effectively utilise results obtained by other designers within the same or other fields of expertise in their work. Design documents are the main vehicle for carrying this information. Figure 6 portrays some of the communication between some of the disciplines in a plant design project.

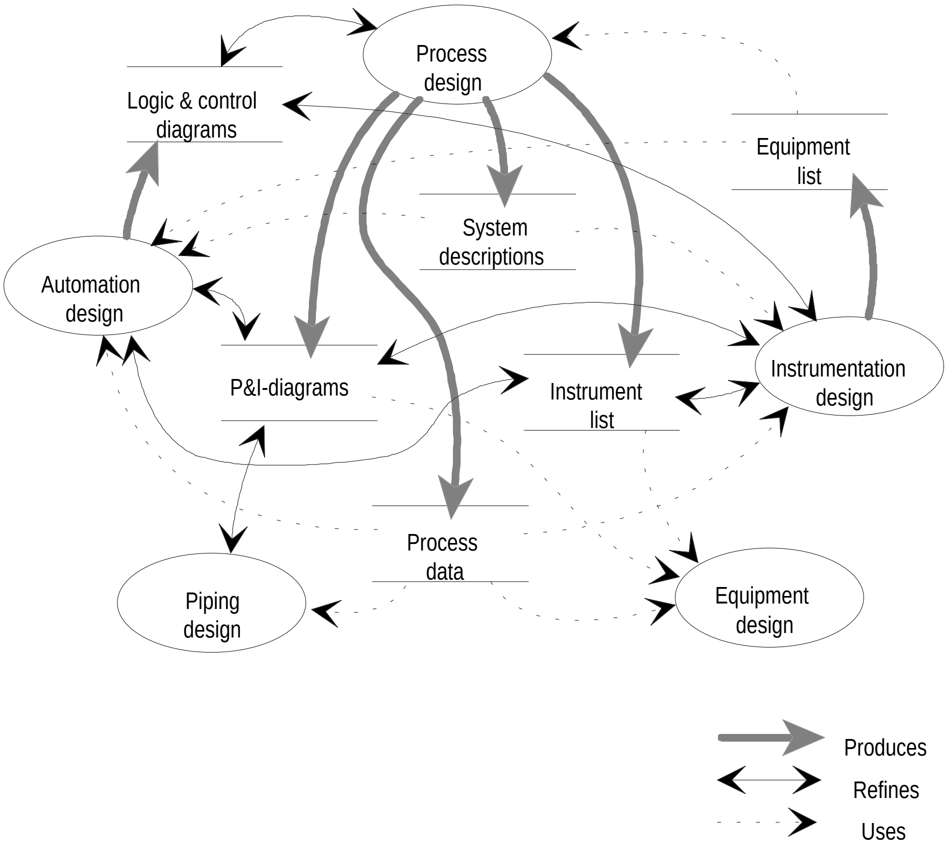


Figure 6. Dependencies between some design areas in an industrial plant project (adapted from Tommila et al. 1990).

The significance of functional design knowledge is emphasised in multidisciplinary design projects. According to several studies automation designers often find it difficult to "decode" the intended functionality of the process based on the structural descriptions of the process (e.g. P&I diagrams) (Tommila *et al.* 1990, Korhonen 1991, Tommila & Viitamäki 1991, Anon. 1992, Huuskonen & Jaako 1992, Kaarela *et al.* 1992). Moreover, the control principles, i.e. how the system should be controlled to achieve the desired functionality, are missing from the process descriptions (Anon. 1992). Textual documents, for example system descriptions, are typically missing during the design project. In many cases, they are written after the design has been completed or the plant has been commissioned. Especially in the case of an unfamiliar process, the automation designer may misunderstand the diagrams due to inadequate explanatory information. The growing complexity of processes makes the situation worse (Korhonen 1991). In many projects and companies discussions between process and automation designers have been introduced to transfer design knowledge. However, such discussions are not a panacea. The concepts and terminology used in the various fields of engineering are not commonly agreed on. This may leave a possibility for misunderstandings. Moreover, the knowledge transferred in these discussions is very seldom written down for somebody else to see.

Information acquisition problems stem from the fact that design documents do not contain all the information related to earlier designs (Crabtree *et al.* 1993). A typical example of this kind of information is missing design rationale. Novice designers frequently need to ask for advice, for example a justification for a specific design decision in an old design to be reused. The expertise in companies is typically possessed by experienced senior engineers. Knowledge access problems can be caused by inaccessibility of the experts; they may be too busy to reply to questions.

Other kinds of information access problems are due to difficulties in accessing written information such as standards, specifications, requirements, and other documents. These problems are mainly related to finding and retrieving a specific piece of information within a set of distributedly stored documents.

3 CURRENT DESIGN TOOLS AND REPRESENTATIONS

Cost is one of the most important competitive factors of virtually any product. In spite of its rather small share of the total costs of the product, the design phase has also been put under scrutiny. Many kinds of computer-supported tools have been introduced to assist the designer and to enhance the design work, in order to cut the costs and improve the quality of the design. However, the quality of the design documents according to their readers leaves a lot to desire. This can in part be explained by the generally disparaging attitude toward documentation work in some companies: it is regarded as unproductive work. Even though a complex system could not be operated without the manuals, only the product itself seems to matter and the value of the design documents as a competitive factor is not acknowledged (Bock 1992). Too little time and resources are allocated for documentation in design projects. Another explanation for documentation problems is that current computer tools are mainly restricted to the final stages of the design process which deal with physical descriptions required for manufacturing and assembling the designed artefact (Green 1992). The design representations supported by these tools cannot capture the design knowledge that led to a specific design. In this chapter, we discuss the inadequacies of the current design tools and representations.

3.1 TOOLS

Current computer-aided design (CAD) tools help engineers in design work by providing assistance for drafting and analysing solutions. These tools focus on assisting drafting of diagrams and drawings that describe implementional issues of the design. The addition of 3D wireframe, surface and solid modelling, and feature-based modelling have made CAD tools powerful in modelling the geometric form of artefacts (Green 1992, Zeid *et al.* 1993). These systems typically represent the design as a set of graphical elements that have a set of static attributes attached to them. These attributes define how (i.e. line style and thickness, colour, etc.) and where (i.e. co-ordinates) the element is displayed in the drawing (Warman 1993). The semantic content of the diagrams or drawings is generally not considered in these systems. The representations do not describe what semantic objects are in a diagram or how they are connected to each other.

A variety of analysis packages, such as finite element analysis (FEA), kinematic and dynamic analysis, thermal analysis, etc. have been developed to assist in the design of complex systems. However, the use of such analysis packages requires the existence of readily produced solutions as their input data. Typically, these programs do not operate on descriptions produced by the above CAD tools, but require the user to create additional mathematical models.

In most CAD tools the supported representations are limited to those depicting the end result of design. Thus the tools are able to record information about the product geometry, materials, and topology for the needs of manufacturing and assembly of the artefact. Similarly, corporate design databases are mostly limited to storing numerical and device data.

Current CAD tools do not provide much for capturing design knowledge in the sense meant in this research. Neither the designer's intentions nor design process itself can be captured (Andersson 1993). The produced design documents contain no information about the functionality of the design or the design process itself. Therefore the tools are not appropriate for answering questions about the purpose of the design or how the design evolved. The tools do not support the early stages of plant design (e.g. conceptual design) where the most important and expensive design decisions are made. In fact, designers use CAD tools only after the major design decisions have been made (Green 1992). This is due to the fact that the tools cannot represent the design at multiple levels of abstraction. They can only represent it at the implementation detail level. In conceptual design, however, implementation details are suppressed and the attention is on the functions of the artefact. Representing this functional knowledge, much less its mapping to the implementation, is not supported.

Current CAD tools do not support systematically recording knowledge related to the design process. The designer is thus required to turn to some other tool to analytically compare design alternatives or record the information related to a design decision. This surely does not attract the designer to do either. Another disadvantage of many CAD tools is that they do not support textual documentation. The textual documentation is most often produced using standard word processors. The separation of the environments contributes to the fact that the textual documents are typically written at the last possible moment or later in many projects.

A major problem with current design tools is that the volumes of data they generate are stored in a fragmented manner. Each tool has dedicated representations and file formats that are isolated at tool boundaries. There is usually no way of propagating a change or indicating a need for a change in other related design documents when a design is changed. Remembering the nature of an interdisciplinary project, this is a true challenge for those responsible for project co-ordination.

The limitations of the representations supported by current CAD tools also have some implications on the reuse of earlier designs: only the resulting device-level solutions can be utilised in succeeding projects. CAD tools allow the reuse of drawings from earlier projects as a basis for new drawings, but with a little if any background information. In other words, reuse is based on low-level details rather than more abstract concepts of the artefact or the design process itself.

Knowledge-based design tools are emerging for industrial use. These tools aim at disseminating design knowledge by allowing for its storage and reuse. Designers' expertise is stored as design rules that can be automatically applied in future projects. However, such systems typically fail to communicate the design knowledge hidden in their internal representations. The knowledge is automatically applied but remains implicit to the users. The explanation facilities in these systems that are intended to explain why certain design decisions were made, often fail to make the knowledge explicit. Often, the explanations only provide a commented list of the rules that have been fired. These design tools are aimed at *routine* design tasks where several successful implementations have been reported (McDermott 1982, Riitahuhta 1988). The designer is still responsible for *creative* design. The tools cannot justify or explain the intention behind a design simply because they do not possess any knowledge as regards creative design.

3.2 DESIGN REPRESENTATIONS

The design phase of an industrial plant produces a variety of design documents that describe the specific decisions taken. Examples of these are

- ❑ diagrams: process flow diagrams, piping and instrumentation diagrams (P&I diagrams), plant layouts, logic diagrams, control diagrams, sequence diagrams
- ❑ device data: component data sheets, technical specifications
- ❑ numeric data: energy and mass balances, stress calculations, design parameters
- ❑ textual documents: system descriptions, operation and maintenance manuals
- ❑ project management data: schedules, deadlines, bids, contracts, reports.

Design documents are the main vehicle for communicating design knowledge to other people (Suitiala 1993). Leppälä (1995) has identified three purposes for communication via design documents. First, the evolving design documents act as a working platform for a group of designers. Second, the documents communicate the status of the design to the project management. Finally, the documents transfer the content of the design to later phases of the lifecycle of the designed artefact. The last point includes the use of the design documents as a corporate knowledge base. However, design documents typically concentrate on implementation details, omitting the more abstract design knowledge (Korhonen 1991, Tommila *et al.* 1990). It has been claimed that this is due to the fact that designers avoid abstraction and use physical concepts in their communication. Suitiala (1993) has explained this tendency as the designer's intention to shorten the gap between design and implementation. While this claim may in many cases hold true, another possible reason is that in spite of its importance, there are neither good tools nor adequate representations for the information produced in the early design stages (Riitahuhta 1988). In the following subchapters we study graphical and textual design representations and consider what is missing in them.

3.2.1. Drawings and diagrams

Graphical design representations, drawings and diagrams, are used extensively in engineering design and generally form the core of design documentation. The drawings and diagrams used in various fields of engineering typically concentrate on implementation details. They are oriented to representing the *structure* and the *physical form* of the design: the subsystems and parts that the system is made of. These representations are usually meant for manufacturing and assembling the artefact. Such a graphical representation shows only the end result of the design process. Current design representations cannot *explicitly* capture or represent the artefact-related design knowledge referred to in Chapter 2. The documents do not contain answers to questions about purpose. Yet, this information is often believed to be *implicitly* transferred via design diagrams.

Figure 7 shows an example of a P&I diagram. Such diagrams are regarded as the most common representations of process knowledge. Even though P&I diagrams only describe the process equipment, their physical interconnections, and some of their physical properties, the diagrams are believed to convey functional knowledge about the process. Of course, this may be true for experienced designers who share the same *background knowledge*. The diagrams imply *tacit*[3] knowledge available only to those possessing general knowledge of and previous experience on similar processes. Other persons such as novices, designers working in different design disciplines, and the operating personnel of the plant encounter difficulties in understanding the diagrams -- and the functionality of the depicted process.

[3]The Oxford English Dictionary defines tacit as "Not openly expressed or stated, but implied, understood, inferred".

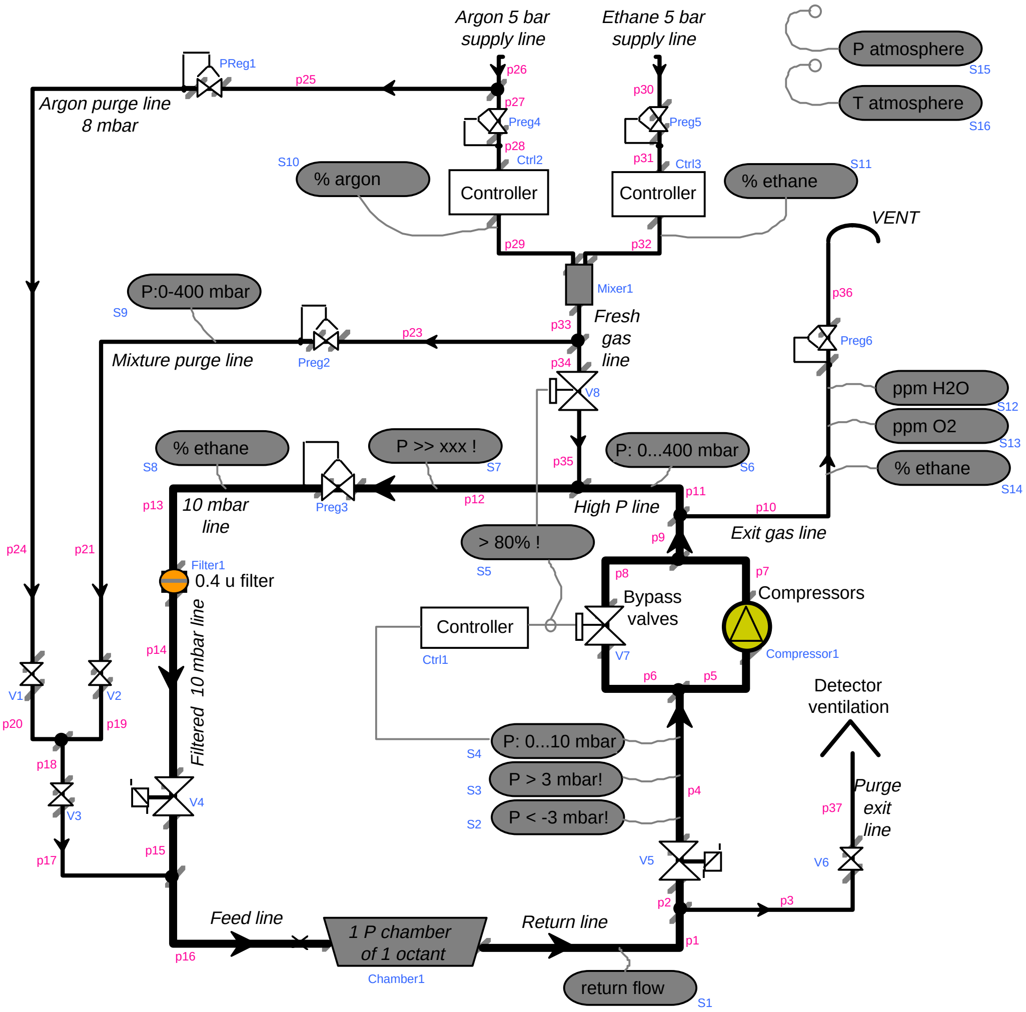


Figure 7. A portion of a P&I diagram of a gas system in a detector at the L3 experiment[4]. (Reprinted with permission from CERN)

Examples of typical questions that the tacit knowledge could answer include

- ❑ goals: "What are the goals of the system? What purpose was it designed to serve?"
- ❑ the relationship between the form and function: "How is function F implemented?"
- ❑ purpose: "What is the purpose of the device Preg1 in the system?"
- ❑ design alternatives: "What other alternatives were considered? Why was a PI controller not used instead of a PID controller?"

[4] The example diagram successfully depicts a common problem: the use of non-standard symbols. This makes the understanding of the diagrams even harder. Working out the intended functionality of the depicted system is left as an exercise to the reader.

- ❑ causal relationships: "What happens if Compressor 1 breaks down? What side-effects does it have?"

A diagram depicting the structure of an artefact is not able to capture the design process, but represents only the outcome of the process. The diagram does not tell anything about the design decisions that have been made to accomplish the diagram. The information regarding each design decision could, of course, be written into textual documents, but it seldom is. Another related problem with graphical design representations is that they do not include the design history, how the design evolved. Even between experts this form of tacit knowledge causes problems; design histories may not be communicated in the diagrams and the same errors may be repeated in succeeding projects.

3.2.2. Textual design documents

Design documents communicate design knowledge to other designers in a project, the users of the designed artefact, and succeeding projects. Here, textual design documents, which form an essential part of the documentation of any ordinary industrial project, have an important role. Written descriptions can be used to explain the information in diagrams and drawings. Some of the tacit knowledge in the diagrams can be made explicit using textual descriptions. Procedural information concerning the designed artefact that is typically presented in operation and maintenance manuals, e.g. procedures required to operate or maintain it, is best expressed textually even though sequences of actions can also be presented by graphic diagrams.

Unfortunately, poor quality often plagues design documents and deteriorates their usability both during design projects and in the operation and maintenance phases. We judge the quality of technical documentation by customer satisfaction: does the documentation contain all the relevant information, are the documents easy to understand, and can the users quickly and easily find the information they are looking for? These questions refer to information content, writing style, and accessibility.

Using a natural language, textual documents should not suffer from the limitations caused by the lack of expressive power of a representation. However in practice, textual design documents seem to lack higher level concepts regarding the design. Designers seem to avoid abstractions in documents and concentrate on the physical implementation of the system. Yet, according to interviews with experienced process designers, they do consider abstract concepts, such as goals, during the early stages of plant design, but the information is not systematically recorded in the documents. Korhonen (1991) has pointed out that designers concentrate on the normal state of the process in their documentation. The exceptional states of the process and how to recover from them are seldom explained. The rationale for design decisions, such as selecting a specific process component, are often written in the designers' personal notebooks but regrettably are seldom found in the design documents (Marsh & Wallace 1995). Due to the delay between the actual design work and the writing of the documentation, some information is usually lost. The designers cannot remember all the details that affected their solutions.

The problems related to the missing information content of the documents may be a consequence of the fact that in many organisations the documents have not been designed to serve their users. The company standards strictly define the layout (format) of the documents whereas the information content of the documents is only vaguely specified. The designers have been given freedom to document what they want to. Often, they decide to document the matters they find easiest to document: the implementation. Thus, the documents describe the implementation of the system informally.

Technical documents are usually produced by a number of experts of various design disciplines each writing in their own style. This considerably affects the understandability of the documents. As an example, inconsistent use of technical terms and jargon make the documents hard for the end user to understand.

To understand the problems of plant design documentation accessibility we must discuss the use of technical documentation. According to Steehouder (1994), technical documentation can be used for a tutorial function, i.e. to help novices master a system, and for a reference function, i.e. to help users to deal with the complex functions of the system or to solve unexpected problems. Typical use of technical documents, using them as reference, differs remarkably from reading an ordinary textbook. Technical documents are usually not read from cover to cover. They may be accessed only when specific information is required to solve a problem for. Steehouder has identified situations where a user of a technical system needs to look for information in the documentation (Steehouder 1994):

- ❑ An impasse: The users cannot proceed because they do not know which functions, procedures, or commands should be used.
- ❑ An error situation: Something unfamiliar and unexpected happens. The users needs information to be able to diagnose the error and plan corrective actions.
- ❑ Uncertainty: The users think they know how the system works or how to act but want to verify the ideas by consulting the documentation.
- ❑ Dis-coordination: The users may be confused by detailed information and look for information that provides an overall picture.

Readers of technical documents try to find a piece of information relevant to their actual problem at any one time. This makes accessibility one of the most important requirements of technical documentation. Regrettably, current documentation systems do not support easy access to the information. For example, the documentation of a modern industrial plant typically consists of thousands of pages of more or less detailed technical information about the plant's subsystems stored in tens of folders. Traditionally, the documentation is divided into different technology areas, such as process technology, electrical systems, automation, etc. Each set of documentation provides a different view of the same plant. Much of the information is repeated in a number of documents. The documentation is usually organised according to the process hierarchy: each subsystem has its own system description, for example. The search mechanisms provided by the paper documents, tables of contents and keyword indexes, are not adequate; it is still difficult to find relevant information. Additional problems are caused by the high number of cross-references between documents and by the fact that information regarding a topic tends to be scattered around the documentation.

An obvious solution is to provide the documents on line. However, this is not very helpful: it neither improves their information content nor provides any great improvement to their accessibility. The documentation is most often authored, maintained, and managed at the document level, i.e. a document stored in the file system of the computer is the smallest separately manipulated entity. Such a document is informal; the computer does not understand anything about its semantic content and can only handle its content as a single text string. Poor document structure makes free text search almost useless. A search even with a well-selected search term may produce too many hits to be helpful. The key point here is that the information presented in the documents is not structured -- the information is not explicitly divided into semantic elements that could easily be manipulated (e.g. searched) as separate units.

The maintenance of technical documentation is laborious and therefore often neglected despite of its obvious importance. Marsh & Wallace (1995) have studied the role of design documents for storing corporate design knowledge. Their results indicate that the relevance of the stored design knowledge is lost if its proper maintenance is neglected. The information does not reflect current knowledge, state of the art technology, and "best practices". Thus the documents do not provide certain information relevant to designers' needs.

The difficulty of maintaining the design documents can be partially explained by the poor structure of the documentation. Due to the multiple occurrences of the same and related information it is difficult to make certain that all the affected points in the documentation have been updated after a design change. For the same reason, any update often requires delivering new versions of the whole paper documentation sets, which is expensive. This may lead to a situation where several inconsistent and partly outdated documentation sets coexist.

3.3 STANDARDS

Within all design disciplines CAD tools are used to enhance the efficiency and quality of work. The result of the work is typically stored in a number of data files obeying various tool-specific file formats causing difficulties in exchanging data between different systems. A great number of direct translators between the various CAD formats have been developed. However, the problem of direct translators is the need for a specific translator for each pair of systems, leading to a huge number of translators. Obviously, a generic file format would reduce this number because only two translators would be required for each system. This has led to the development of international standards to allow easier data exchange: Initial Graphics Exchange Specification (IGES) (IGES 1988) and Standard for the Exchange of Product Model Data (STEP) (ISO 10303-1 1994). IGES is a neutral file format that can represent the geometric information in CAD drawings. However, IGES suffers from difficulties in translation, especially when the functionalities of the sending and the receiving CAD systems are dissimilar (Schumann-Hindenberg 1993). Another problem with IGES is that it is not widely supported; in fact the data exchange file format (DXF) of the most popular CAD tool, AutoCAD[5], has gained a position of a de facto standard to exchange two-dimensional data (Teeuw *et al.* 1995).

Since IGES is limited to representing geometric data, it does not aim at solving the problems presented in the previous section. The objective of the STEP standard is to provide a way to transfer product data throughout the life cycle of a product in a system-independent manner. This implies that information going far beyond pure geometric data representing part and product models or design rationale could be represented and transferred between systems. In theory, the representations used in STEP, such as the EXPRESS language do not set any limits on the kind of information to be transferred (ISO 10303-11 1994). However, current application protocols in various fields of engineering are mostly limited to support drafting and detailed design. No models for data of a more abstract nature representing, for example, functional properties of the product have yet been defined.

---

[5]AutoCAD is a registered trademark of the Autodesk Company.

There are also several standards that are related to text. The electronic data interchange for administration (EDI) between organisations is quite well standardised, especially for commercial data (e.g. orders, invoices, etc.). Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) is a standard that has been developed for EDI purposes (ISO 9735 1987). The main goal of EDI (and EDIFACT) is to structure the data in a standardised way so that receivers can recognise the parts of the message and use them as such in their own information systems. The layout of the documents is ignored in EDI. Another standard that has been developed for transferring electronic documents between organisations is Open Document Architecture (ODA) (ISO 8613-1 1989). ODA preserves the layout of the transferred documents; in addition to defining the logical content of the documents, the layout is also defined in ODA. ODA is meant to be a kind of a neutral format between various word processing systems. ODA presupposes that special conversion programs are used to convert the ODA messages to and from the internal representation of the word processing system.

The most important standard related to documents is however Standard Generalized Markup Language (SGML) (ISO 8879 1986). SGML is a meta-language for defining the structure of documents. It is based on logical markup of the documents; the logical parts of the document are tagged so that they can be recognised from the text. An SGML document is attached to a document type definition (DTD) that defines its structure, the character set, and the tags used in the markup. A related standard is Hypermedia/Time-Based Structuring Language (HyTime) that offers a standardised way of expressing external links between information elements (ISO/IEC 10744 1992).

3.4 DISCUSSION

Current CAD tools are not designed to cover conceptual design but are restricted to detailed design. The diagrams or drawings represent the design as a set of graphical elements (e.g. circles and lines) while the semantic content of the diagrams or drawings is generally not considered in these systems. Consequently, the supported representations are limited to those depicting the end result of design, mainly for manufacturing and assembly of the artefact, ignoring both the functional design knowledge and that related to the design process.

New standards for both CAD (STEP) and textual documents (SGML) provide representations that could capture the more abstract information which has previously been ignored. However, current applications of these standards do not generally utilise this ability.

4 MFM AS A FRAMEWORK FOR CAPTURING PLANT DESIGN KNOWLEDGE

This chapter gives a brief introduction to multilevel flow modelling and presents how the design knowledge identified in Chapter 2 can be captured using multilevel flow modelling as a framework.

In Chapter 2, two important classes of design knowledge were identified: one related to artefacts and another related to the design process. In Chapter 3, the general drawback of current design representations and tools was recognised: they are mainly limited to structural models that describe the structure of the process. They only can capture the implementational issues of the designed artefacts while omitting the higher levels of abstraction typically considered in the conceptual design phase. Since designers' work is to a great extent *reasoning about function,* i.e. how to achieve specific functional requirements using available technology, such design descriptions alone seem to be inadequate. Acquiring knowledge related to the design process itself is poorly supported.

Multilevel flow modelling combines structural and functional descriptions of a process. It provides a mechanism for representing the correspondence between the goal structure and the resulting product structure. MFM does not provide any support for recording design-related knowledge. However, the formal representation of the domain concepts can be used for this purpose.

A multilevel flow model is not a *behavioural* model in the sense that it could be used for predicting the behaviour of the process it describes. However, MFM is a normative model that describes the intended behaviour of the system that can be used for detecting deviations from the normal behaviour (Lind 1990). For this purpose, it utilises conservation laws for mass and energy flows.

4.1 MULTILEVEL FLOW MODELLING

This section presents a brief description of MFM, relying heavily on the work of Lind (1982, 1990). Originally, MFM was created for the design of man-machine systems but has later been found useful for several other purposes such as measurement validation, diagnosis, alarm analysis, and plant state identification (Lind 1982, Larsson 1992, Sassen 1993, Van de Ree 1994).

A multilevel flow model describes a system as a man-made artefact designed for specific purposes (Lind 1982). MFM models are normative, focusing on the role of the various parts of the system in fulfilling the purposes of the system as a whole (Lind 1990). In MFM, purposes are modelled with the concept of *goals* denoting the objectives for using the system. There are three types of goal: *production, economy, and safety*. The second important concept of MFM is a *function*. In MFM, functions are seen as the means to *achieve* the goals. The functions of the plant are represented by a set of mass, energy, and information flow structures at several levels of abstraction. The flow structures are composed using a set of elementary flow functions. The physical components of the system, the *devices,* are used to *realise* one or several functions.

MFM provides a formal language for representing the functional structure of a plant at several levels of abstraction. The language consists of a grammar based on mass and energy conservation laws, a set of concepts, and their relationships. MFM provides a set of graphical symbols that allows models to be created and represented in an easily understandable graphical format.

4.1.1 Whole-part and means-end dimensions

For managing complexity, MFM utilises hierarchical abstraction in two dimensions: the *whole-part* dimension and the *means-end* dimension, as depicted in Figure 8. The whole-part dimension, which decomposes a system into its subsystems and into parts, is well known and commonly used in the engineering of many kinds of man-made systems. The whole-part decomposition utilises the *is-part-of* relationship.

The means-end dimension, based on means-end analysis introduced by Rasmussen (1986), is the far more distinctive dimension of MFM. Even though such abstraction is implicitly used in the design of almost any kind of system, it is rarely supported in design tools. Means-end relations are used to represent relationships between a goal and the functions required for its achievement (*achieved-by*), and between a function and the physical components required to realise it (*realised-by*). In addition to mandating a set of flow functions, the achievement of a goal may require active control or management of the flow functions involved. This implies a need for an external controller. In MFM, such a controller is represented by a *manager* function. The resulting tripartite relations between a goal, the flow structure required for its achievement, and the manager function are described via *achieved-by-control* relations. These relations are important because they provide a way to model the control functions performed by either human operators or automatic control systems as an integral part of the process.

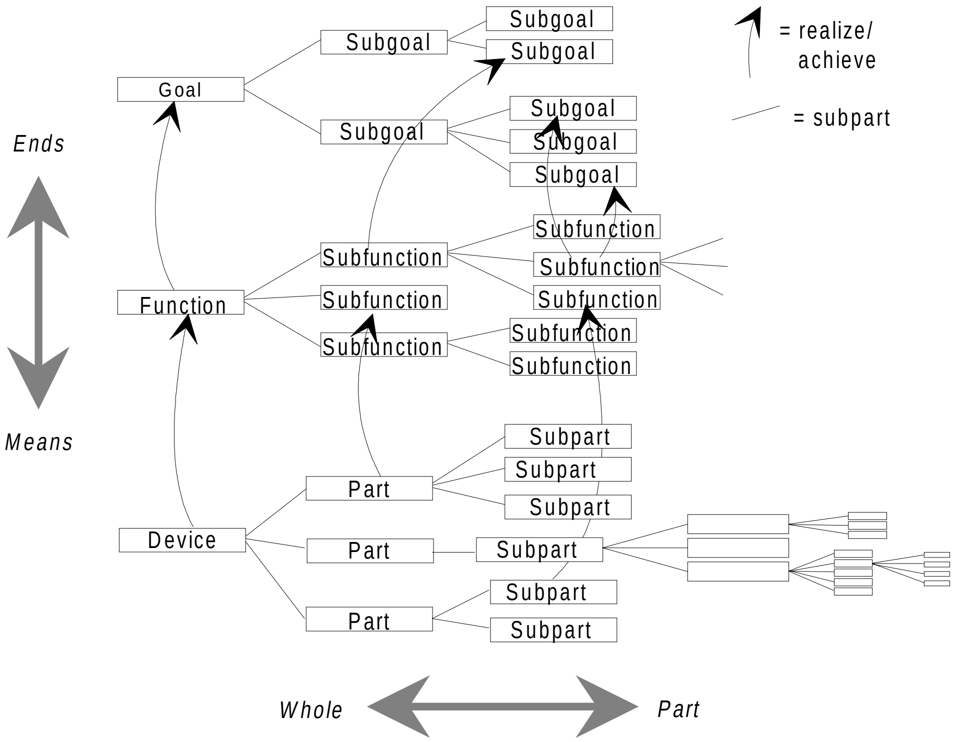


Figure 8. The dimensions of a Multilevel Flow Model.

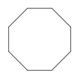
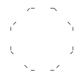
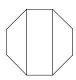
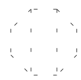
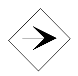
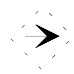
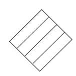
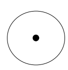
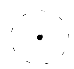
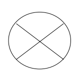
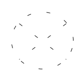
A *condition* relation is used to express a dependency of a function on the achievement of a goal. This means that the function will only be available while the supporting goal is achieved. Examples of the above relations are shown in Figure 9.

An MFM model is a hierarchical functional model which basically describes the plant's intended function. It is important to note that the functional structure depicted in the MFM model does not necessarily follow the physical structure. This is evident because all the relations of MFM models are usually not one-to-one mappings; one-to-many, many-to-one, and many-to-many mappings frequently appear. For example, several functions may be required to achieve one goal, one function may serve in the achievement of several goals, and one physical device may be involved in realising several functions. Moreover, the functional structure of a plant is different in its various operational states (e.g. start-up, normal operation, shutdown). Not all the physical devices used during start-up may be needed at all during the normal operation of the plant. This indicates a need for a dedicated MFM model for each operational state of the plant (Van de Ree 1994).

4.1.2 Flows

The concept of *flow* is important in MFM. Each function is described by a flow structure that depicts the flow of mass, energy, or information through (part of) an industrial plant. The most important reason for the choice of using flow (especially mass and energy) concepts in MFM is that "the interchange of energy and mass is responsible for the interactions between plant components, subsystems, and the control systems" (Lind, 1990). A flow structure consists of basic flow functions that are meaningfully connected according to certain rules resulting from the mass and energy conservation laws. Table 2 shows the mass and energy flow functions of MFM, their explanations, and the graphical symbols used in modelling work. Means-end relations are used to connect the flow structures at various level of abstraction to form an MFM model.

Table 2. The basic flow functions of MFM (Lind 1990).

Flow function	Description	Symbol	
		mass	energy
Storage	A property of a system to function as an accumulator of mass or energy		
Balance	A property of a system to provide a balance between incoming and outgoing flows		
Transport	A property of a system to provide transfer of material or energy between two systems		
Barrier	A property of a system to prevent the transfer of material or energy between two systems		
Source	A property of a system to act as an infinite reservoir of mass or energy		
Sink	A property of a system to act as an infinite drain of mass or energy		

A multilevel flow model does not describe the behaviour of the modelled system either accurately using a mathematical model or approximately using, for example, a quantitative model (Kuipers 1986). Thus, it cannot be used for predicting the system's future state. However, its normative description of a process acts as a reference model of the plant's intended operation. In MFM, the intended operation is described as a set of flow structures each of which should obey the mass and energy conservation laws. Thus, modelling a system with MFM yields a set of simple flow balances describing the normal operation. Comparing the actual behaviour of the process to the reference model offers a way to detect abnormal process conditions. Loss of mass or energy is an indication of a process fault whereas the accumulation of mass or energy indicates a potential risk: plant emergencies are usually caused by violation of these balances, e.g. by leakage of dangerous materials.

Figure 9 shows a fraction of the MFM model of a feedwater system. According to a standard MFM practice, the devices are not shown in the model.

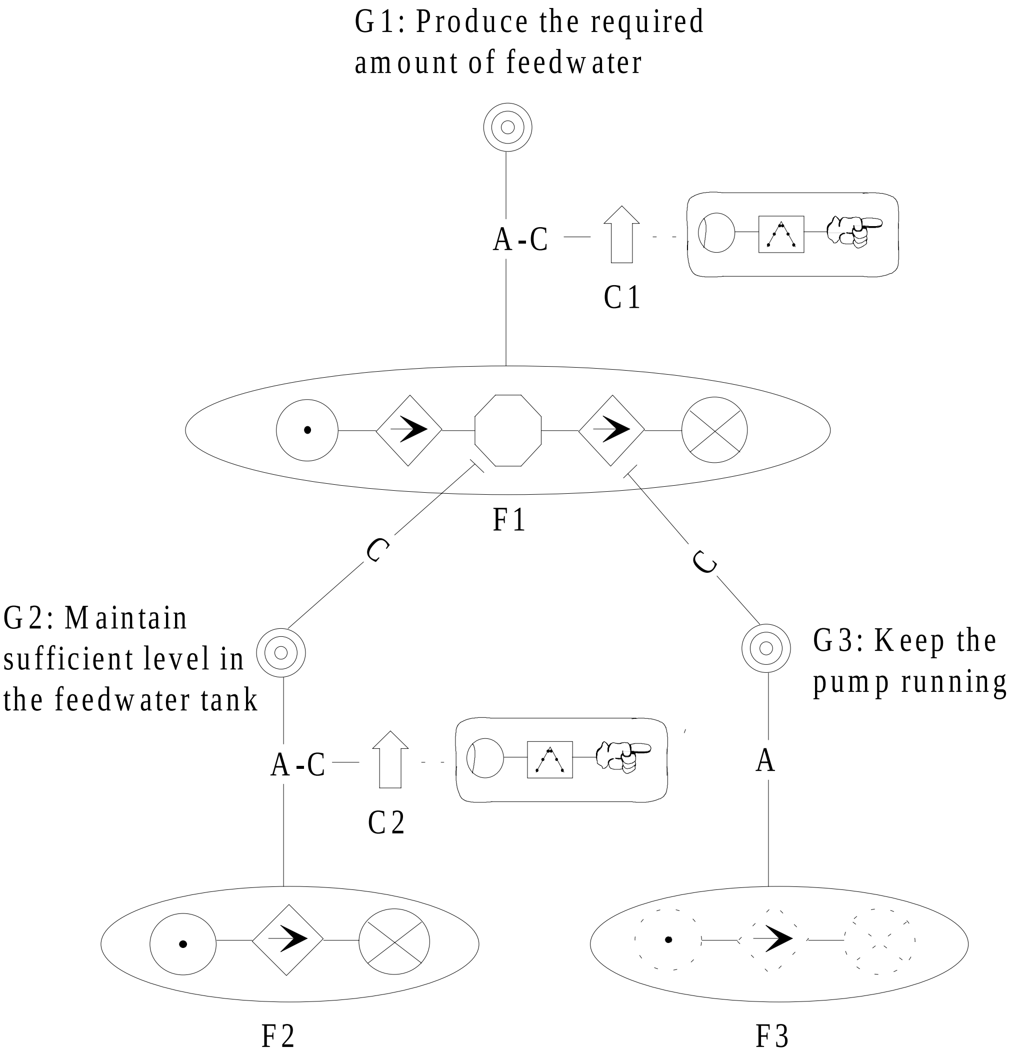


Figure 9. An example of a multilevel flow model.

The uppermost goal in the figure, *G1: Produce the required amount of feedwater*, is to achieved through an aggregation of flow functions labelled F1 in the figure. However, presence of external control is also required to achieve the goal. Thus, an achieve-by-control relationship is applied in the model. The controller is represented with a manager function C1 in the figure depicting the information flow related to the control task. Following

Larsson's MFM improvements (1992), the manager function is further divided into the observation, decision, and actor functions.

Figure 9 also shows an example of the condition relationships that are utilised in MFM models to depict the dependency of a flow function on the achievement of another goal. For example, the storage function in F1 (depicting the feedwater tank) is conditioned on the goal *G2: Maintain sufficient level in the feedwater tank*. The goal G2 is achieved by controlling the water supply of the feedwater tank.

The goal *G3: Keep pump running* is achieved by the set of flow functions F3 that depicts the energy supply to the feedwater pump.

4.2 REPRESENTING ARTEFACT-RELATED DESIGN KNOWLEDGE WITH MFM

In Chapter 2.1.1 we defined artefact-related design knowledge as consisting of structural, functional, and problem decomposition knowledge. Functional knowledge was defined as a *mapping* among the functional requirements (i.e. the design goals of the system), the overall function structure, and the detailed design representing the structural (i.e. form) aspect. Problem decomposition knowledge in turn provides information on how a complex design problem can be divided into more manageable subproblems.

4.2.1 Representing structural and functional knowledge

Let us now consider how multilevel flow modelling could be utilised for capturing functional and structural knowledge in a systematic design process. The design process proposed by Pahl & Beitz (1988) consists of four phases, clarification of the task, conceptual design, embodiment design, and detail design. The phases are not strictly successive due to the iterative nature of design; the latter phases may produce useful feedback to the previous ones -- design may even involve discovery of new goals. This iterative process has been acknowledged by the MFM community (Lind 1990).

The functional requirements of a system usually state the purposes for which it is being designed. In MFM, the purposes (i.e. the objectives of running a system, for example) are modelled with the concept of *goals*. However, it is common in engineering design that the requirements are rather vaguely expressed, often in qualitative terms. Such requirements are a poor starting point for design: they do not provide the designer with any means of deciding whether the goal is satisfied or not (Lind 1990). Therefore, the requirements of the system should be quantified, if possible, and further refined to explicit goals of the system in the task clarification phase.

MFM does not provide any support on how to recognise the various goals of an artefact to be modelled -- it only provides a means of representing the goals and how they were meant to be achieved. Paper IV presents some elementary ideas on how to recognise the safety goals of an industrial plant. We utilise checklists for recognising known potential hazards -- a well-known method within safety analysis research (Toola 1992). Safety goals are formulated to prevent or mitigate the potential hazards.

According to Pahl & Beitz (1988), one of the major tasks of conceptual design is establishing alternative function structures fulfilling the goal of the system[6]. The function structures show the combination of individual subfunctions required to perform the overall function. Since the functions are generally based on the flow of energy, mass, and information, MFM seems a good alternative for representing such function structures, especially in the case of continuous processes. In MFM, the functions can be described as an interrelated set of mass, energy and information flows using the concept of *flow functions*. Modelling the information flows deserves a special mention. The information flows are related to cases where the achievement of a goal requires the active operation of an external controller. The information flow models the external controller in terms of the process variable(s), the control algorithm, and the actuator. It is important to note that a control algorithm may be implemented by either an automation system or a human operator. In this sense, the model provides a basis for allocation of control tasks between the automation system and the human operators.

In Pahl and Beitz's approach (1988), design issues related to form are mainly taken care of in embodiment and detail design. In MFM, the lowest level of the means-end hierarchy, the devices, represents a part of the structural knowledge of the artefact reflecting the functional structure of the system. However, it is important to note that MFM models the functional structure of the process which may differ from its physical structure. MFM does not provide any specific graphical presentation of the devices, and it is common to leave the device-level solutions out of a multilevel flow model for reasons of clarity. Furthermore, current domain-dependent representations (e.g. P&I-diagrams in process engineering) are both well-suited and commonly used for this purpose. However, the *realise*

[6]Pahl & Beitz use the term "technical function of a system" to define the objective of the system. We use the term "goal" to avoid confusion with the concept of function in MFM.

relationships between the function and device level of the model should be recorded since they provide an effective way to structure the information.

4.2.2 Representing problem decomposition knowledge

There are two widely used approaches to managing complexity in engineering design: abstraction and decomposition. When using abstraction, complexity is reduced by hiding the details and concentrating on the essentials. Decomposition utilises the divide-and-conquer principle: the problem is divided into smaller subproblems that are easier to solve. Most systematic design approaches generally emphasise the importance of problem decomposition (Pahl & Beitz 1988, VDI 2221 1987).

In many domains of industrial engineering (e.g. design of a power plant), the overall decomposition has remained relatively invariant for several decades (Riitahuhta 1988). Introduction of new technologies does not generally change the decomposition but affects the way the subproblems are solved. This implies that decompositions that are found effective can be reused.

MFM's *whole-part* relationships can be effectively used to describe problem decomposition at various levels of abstraction: goal, function, and device. An MFM model can be regarded as representing a kind of a design sequence (i.e. plan) that is structured according to the design goals (for achieving <goal>, use <plan>). In MFM, goals are typically divided into subgoals to be achieved by a number functions and their subfunctions. The model thus represents valuable problem decomposition knowledge showing a clear trail from the original design problem to its final solution. Each subgoal (and a function or subfunction, respectively) may have a number of design alternatives to choose among. A design decision describing the selection of one of the alternatives represents a *step* in the design sequence.

Decomposition does not generally result in a set of independent subproblems. Respectively, the solutions -- e.g. the subsystems of an industrial plant -- do interact. For example, a small drop in the temperature of live steam in the boiler of a power plant will lower the efficiency of the turbine. On the other hand, too high a temperature of the steam could damage the turbine. In an industrial process, these causal interactions between subsystems are typically based on the interchange of mass and energy. In MFM, such a process is described as an interrelated set of mass, energy and information flow structures. These structures are composed of flow functions that are connected together at *boundaries*. A boundary is characterised by a set of variables shared by the connected functions. Conservation laws for mass and energy are applied to model the interactions between the flow functions through the boundaries. A violation of the conservation laws indicates a disturbance or a failure. Furthermore, accumulation of mass and energy indicates a potential source of risk in plant operation (Lind 1990, van de Ree 1994).

In addition to process interaction via boundaries, a flow function can be conditioned by another goal. This means that the function will not be available if any supporting goal is not yet achieved. MFM has a special relationship for representing these interactions; the *condition* relationship (Lind 1990).

4.3 USING MFM FOR STRUCTURING DESIGN-RELATED KNOWLEDGE

A design process can be regarded as a sequence of interrelated design decisions. Each design decision causes a design move, i.e. a change towards the final design is made to the design descriptions[7]. MFM does not directly support the recording of design-related knowledge, i.e. descriptions of the design process itself. However, an MFM model of the final design of an artefact in essence represents a sequence of design decisions that led to this specific design. In this work, we use MFM for modelling artefacts during their design which generally proceeds from abstract to concrete and from general to detail. A transition between the levels in either of the two hierarchies is essentially a design move preceded by a well thought out design decision. For example, choosing to achieve the goal "raise enthalpy" via two abstract functions "raise pressure" and "raise temperature", is a design decision. Thus each object in the model represents a design decision.

The MFM model of an artefact represents only a part of the design process: the sequence of successful design decisions that led to the design. The MFM model, per se, does not contain any information about alternative solutions that were considered or the design rationale. However, the model provides a framework to structure and store the otherwise typically informal design knowledge: the model identifies the design decisions, each of which involves both related design knowledge such as other alternatives that were considered and design rationale. Linking the informal design knowledge to the formal representation provided by the MFM model makes the information semi-formal: parts of the representations can be interpreted by computers which makes the use of the information a lot easier. This can be augmented with the introduction of templates or forms indicating what information is expected to be filled in by the designer.

---

[7]As stated earlier in Chapter 2, a real design process is seldom if ever that simple; it also includes poor design decisions that have to be reconsidered and often reversed.

Figure 10 shows schematically how design-related knowledge can be linked to an object of the model.

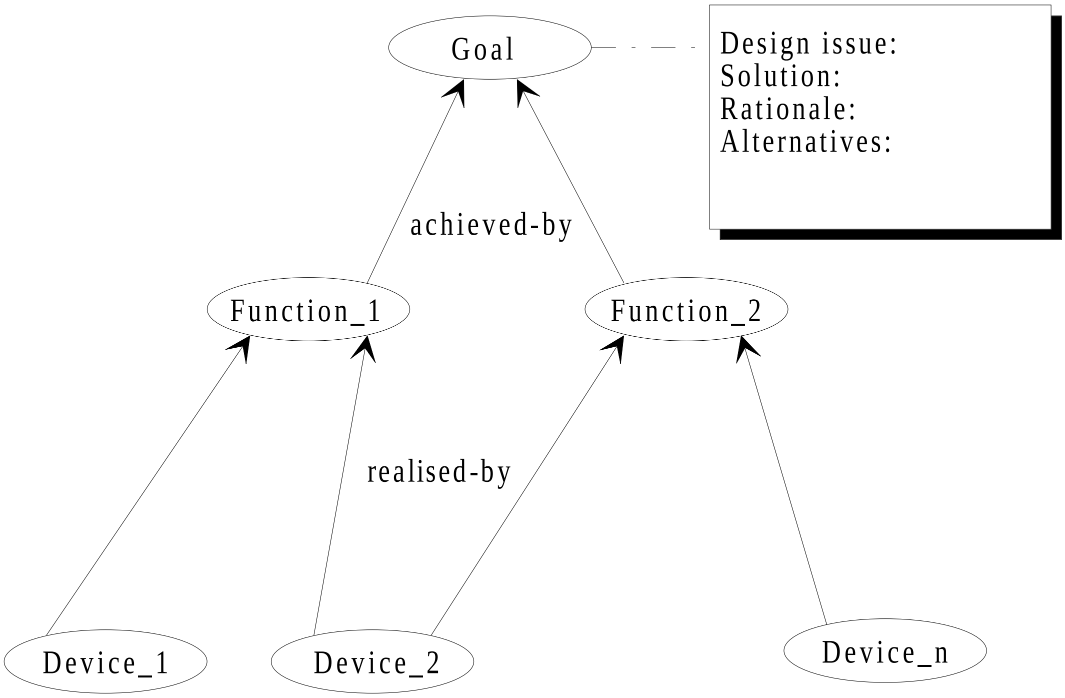


Figure 10. Design-related knowledge in an MFM framework.

4.4 DISCUSSION

The importance of functional design knowledge in industrial plant projects has been recognised by several researchers. The bottleneck between process and automation design caused by missing information about the plant's or its sub-system's functionality and control principles has been a prime inspiration for many techniques. For example, improved content models of process descriptions have been proposed (Tommila *et al.* 1990, Tommila & Viitamäki 1991, Korhonen 1991, Anon. 1992).

There is a large variety of representations for modelling the functionality of systems. Keuneke (1991) has presented a device representation language in which the functions of the device represent its purpose. In her model, the functions are achieved by behaviours that are represented as causal sequences of states. MODEL.LA is a modelling methodology for interactive or automatic definition of chemical processes (Stephanopoulos *et al.* 1990a, 1990b). It is based on a declarative object-oriented approach. It allows multi-faceted modelling by providing multiple views of the process in terms of structure, topology, and behaviour. Padalkar *et al.* (1991) have presented a system which performs an alarm analysis of the target process. The process is described by hierarchical tree structures with a functional representation of functions and subfunctions, and a representation of the system's structure. Constraints are used for searching the faulty nodes. Lower level causes are searched by propagating the information downward in the hierarchies.

The foundations of this thesis lie on the MFM modelling technique (Lind 1982, 1990) that is used to record the conceptual knowledge of the plant in a formal manner. An MFM model can be used to represent a plant design resulting from the work of several design disciplines including process, automation, instrumentation, electrical, and organisation design. Thus, an MFM model provides a common conceptual model for several design disciplines. As such, it provides a basis for implementing a shared design database enabling the concurrent engineering approach (Reddy *et al.* 1992).

This thesis introduces the idea of using MFM for structuring the informal design knowledge related both to the artefact and to the design process. As for recording design-related design knowledge, our work utilises the research results obtained within the design rationale community. Our approach for describing the design decisions is similar to the ISAAC structure used in Design Journal (Conklin 1987). However, the formal representation of the design provided by the MFM model is used as a framework to identify the design decisions and to structure the knowledge related to those decisions. A similar approach has been presented by Prins & Olthoff (1993) who use a morphological design approach to structure the design process instead of MFM. Malmqvist (1995) uses a function-means tree for the same purpose. The advantage of having such an abstract design representation as a basis for storing the design-related information is that many design decisions in general, and those in the conceptual design in particular (e.g. concerning functionality), are more naturally connected to the more abstract objects of the domain (Malmqvist 1995).

5 A DESIGN ENVIRONMENT PROTOTYPE

In this research, a prototype of a design environment capable of structuring and recording design knowledge was implemented. The main point of the design environment is to make both artefact-related and design-related design knowledge explicit. For this purpose, the environment should be able to

- ❑ represent the design at multiple levels of abstraction
- ❑ capture design decisions and their rationale
- ❑ support structured documentation
- ❑ support efficient reuse of earlier designs and design knowledge
- ❑ export the design knowledge to other applications.

Undoubtedly, the most important requirement is the first one: to be able to represent the design at multiple levels of abstraction. The representation should be formal since several tasks call for computer-supported reasoning about the knowledge.

A multilevel flow model is in essence a semantic network that describes a system as objects at various levels of abstraction and their relationships. According to the KADS[8] terminology, it represents the *domain knowledge*, or conceptualisation of the domain (Wielinga & Breuker 1986). We are not acquiring the domain knowledge with a specific task in mind. Rather, we share the view that a considerable part of the domain knowledge is useful in a number of tasks (Abu-Hanna & Jansweijer 1994).

Our goal was not to build a knowledge-based design tool that would automate the use of the domain knowledge. Our tool was not designed to capture the knowledge-use level of expertise: the strategic, task, and inference layers in KADS.

Traditional CAD tools are constructed on the basis of graphical entities (e.g. lines, circles, etc.) rather than on the basis of objects. The entities may have attributes but they have mainly been used for defining graphical properties of the elements (e.g. thickness, line style, etc.) (Warman 1990). Such an approach is not adequate for representing the design as an object hierarchy. The conceptual design phase especially suffers from this inadequacy.

---

[8]KADS is an approach to knowledge engineering developed in the KADS project (ESPRIT-I) P1098.

Object-oriented databases are well-suited for engineering applications (Warman 1990). In addition to the static knowledge contained in a semantic network, they can incorporate the behaviour of the objects. One of the strengths of the object-oriented approach is that the objects identified in conceptual modelling can be seen in the implementation (Kim 1991). In multidisciplinary projects, an object-oriented database which stores all the design data as objects and their attributes and relationships can be seen as a gateway between different design disciplines. A design change in one discipline can immediately be seen by designers in the other disciplines (Paasiala *et al.* 1994).

A prototype of the design environment was built on top of a commercial knowledge-based object-oriented design tool, Design++ from Design Power, Inc. (Harmon 1990, Katajamäki 1991), and one of its applications, P&ID. The most important reason for selecting Design++ was the object-oriented facilities provided by the tool. Furthermore, our industrial partners also had interest in Design++, and one of them had already developed a successful design system using the same environment.

The object system of Design++, based on the KEE environment by Intellicorp (KEE 1986), is used to store the knowledge about the product and the design[9]. Geometry tasks, such as drafting and three-dimensional visualisation, are carried out by the AutoCAD system (Autodesk, Inc.), which is transparently linked to Design++. There is also an interface to a relational database which contains component data and allows communication with other design environments.

P&ID is a knowledge-based design tool for supporting the initial phases of boiler design from boiler balance calculations to process dimensioning. It has been developed on top of Design++ by Tampella Power, Inc., a Finnish boiler manufacturer. The aim of the system is to capture some of the designers' know-how in the system's design rules and libraries and to reduce the time spent in the boiler design phase. With the tool, final P&I diagrams of the boiler can be produced more quickly. The diagrams contain automatically generated component labels and more information about the process variables. The system's knowledge base can be used to produce various component data listings and specifications automatically.

[9]We used an early version of the software. In more recent versions, KEE has been replaced with a C++-based repository.

5.1 REPRESENTING THE DESIGN AT MULTIPLE LEVELS OF ABSTRACTION

P&ID was a good basis for our design environment because it readily featured a way to represent structural product knowledge. In the tool, the concept of product structure is essential: components are arranged in a tree hierarchy in which more abstract components reside at higher levels and more detailed ones at lower levels. The product structure thus provides a way of representing the whole-part relationships of the product. The knowledge about the components' properties is stored in the objects' attributes. In addition to static properties, each attribute may have an associated design rule that can be used to infer the value of the attribute. The design rules are coded in Common Lisp.

However, the product structure was limited to representing the form of the design (the topology of the system and the properties of its individual components). We expanded the representation capabilities of the system by introducing two new hierarchies for representing the function and goal levels in the MFM models (Figure 11). Using the extensions, it is possible to create the higher levels of abstraction in the MFM models and the means-end relationships between the levels. The relationships between the levels of hierarchy have been omitted for clarity in Figure 11.

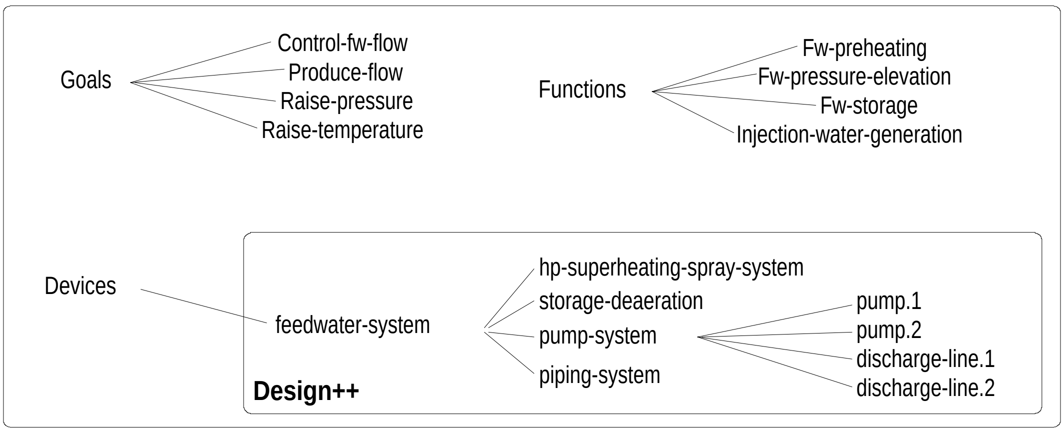


Figure 11. Extensions made to the P&ID-tool.

5.2 DESIGN-RELATED KNOWLEDGE

The design process of a large industrial plant can be regarded as a series of design decisions. These decisions range from fixing the type, size, location, and layout of the plant at the early design stages to selecting among a set of alternative components during detailed design. The importance of these decisions cannot be overemphasised, keeping in mind that some 75 - 85% of the total costs may have been fixed after the design stage (Nichols 1990, Sheldon *et al.* 1990). It is clear that design knowledge behind these decisions is of great interest during plant operation and maintenance (especially during revisions) and when designing similar plants. However, the information is generally not available because it is not systematically recorded. Even though designers may use analytical methods while making the decisions (which is not always the case) they usually only make informal personal notes. Such information is not available later for other people or even for the designer.

Clearly, the design environment should support analytical decision making and systematic recording of knowledge related to the design process. The original intention was to integrate a tool supporting analytical decision making into the design environment and to import the information generated by the tool (e.g a matrix describing the decision process) into the design documentation. For this purpose, a tool developed at VTT Electronics named Requex, based on the Quality Function Deployment method (QFD) (Akao 1990) and the Analytic Hierarchy Process (AHP) (Saaty 1983), was evaluated and several matrices supporting various kinds of design decisions were created (Väisänen 1994). Due to the use of different platforms the workload and the cost were estimated as too high and the software was not integrated into the design environment. Moreover, the feasibility of such an approach in general and the availability of all the data required for the matrices in particular was heavily questioned by the professional designers. This implies that design decisions are often not structured enough to allow the use of such an approach (Holsapple & Winston 1992).

A semi-formal approach was chosen for capturing the design rationale, based on the use of forms indicating the expected information regarding each design decision. The use of such semi-formal structures serves two purposes: they assist the designer to concentrate on essential information regarding the decisions and they make the informal information partly processable by computers (Conklin 1987). As suggested in Section 5.3 design decisions have a close relationship with MFM modelling: refining the MFM model to a more detailed level in any of its two hierarchies (means-end or whole-part) should be preceded by a well thought out decision. During the design work, the designer is requested to fill in a form (a document template) regarding each object in the model. The forms include slots for

- ❏ design issue
- ❏ alternatives that were considered
- ❏ solution: which alternative was chosen
- ❏ design rationale: why one alternative was chosen and why the others were not.

The design issue is a description of an object to be decomposed into parts belonging to a more detailed level. It denotes the function of the object and it is typically written when the object is created. Therefore, it is typically not written when the decision is made but automatically retrieved from the database of the design environment. The other slots (alternatives, solution, design rationale) are filled in when the object is further decomposed. Concerning rationale slots of such forms, it is possible to provide a menu selection of possible rationales for the designer to choose from. At the device level, such a function seems necessary: decisions are numerous and the designer does not have time to justify each decision with written text. Some possible rationales for such choices are technical performance, cost, reliability, safety, and durability.

Technically, the support for recording the design-related knowledge was implemented using the features of the design environment supporting structured documentation described in the next section.

5.3 STRUCTURED DOCUMENTATION

In Section 3.2.2 we recognised problems related to technical documentation. During the projects, the late creation and delivery of documents to other parties forms a severe hindrance for effective co-operation. The poor quality of documents, in terms of their information content and semantic structure, reduces their usability during both the design projects and the operation and maintenance phase.

To overcome the above problems with plant documentation, the design documents should be structured according to the information they contain, not their physical layout. In other words, a document should not be seen as a bunch of paper obeying some predefined formatting rules (Figure 12 a) but as a collection of interrelated pieces of information. We propose that plant documentation should be structured according to the MFM model depicting the plant (Figure 12 b). At each level, the information is further broken down into a set of information elements. The information elements are connected to the objects of the model. Introducing a semantic structure into design documentation makes the documents semi-formal. This implies that the documents are partly interpretable by computers which allows their efficient computer-supported use. Moreover, such a model -- the explicit definition of the goals of the plant and the relationships between the goals, functions, and devices -- defines both the issues to be documented and their relationships. The relationships can be used as navigational links when searching for a specific piece of information.

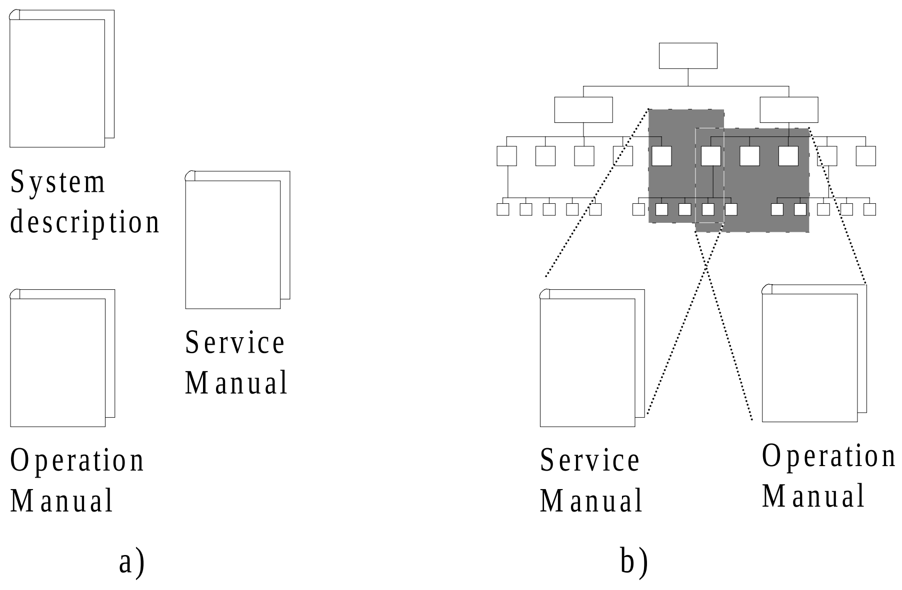


Figure 12. Two ways of managing the information in the documentation: a) documentation is managed at the level of documents, and b) the information is explicitly divided into semantic elements from which the documents can be generated. (Adapted from Weich 1992)

We have implemented a documentation support system to demonstrate the use of structured documentation during plant design. The idea is to allow the designer to write small fragments of structured text regarding each object in the MFM model as the design work progresses. The support system features a pop-up editor for writing these texts. The editor is automatically invoked whenever the designer creates a new object in the MFM model. For each type of object, the editor automatically imports a template showing the designer what to document. The text elements are explicitly marked up using SGML (ISO 8879 1986) and stored in a database. When applicable, a code for the object generated by the design tool is used as a unique identifier for each text element. The markup facilitates their identification and easy retrieval for different purposes, e.g. for producing printed or electronic documents or for explaining the design. The approach for supporting documentation resembles that of Concordia (Walker 1988, 1989) and RADIO (Arango *et al.* 1993).

The database containing the text elements can be regarded as a set of information depicting the plant (Smith 1992). Moreover, together with the other representations of the design, the text database could be seen as a corporate knowledge base. Ideally, both on-line documentation and traditional paper documents could be automatically configured from the text elements much in the same way that software is linked from the program modules. This is of great importance since several documents contain redundant information. For example, operation and maintenance manuals derive descriptions of the functionality and structure of the subsystems of a plant from the system descriptions written by the designers. Even though all the material for all the documents is very unlikely to be created during the design phase, the material produced will greatly assist in creating the final documentation.

Let us consider the task of building a hypertext-based on-line documentation system. There are two important concepts in hypertext: nodes and links. The nodes represent the information elements and the links represent the relationships between them. The problem of creating hypertext is identifying the information elements needed by the intended user and recognising their relationships. These tasks are far from trivial, and current design documents hardly provide any assistance. The notion of *markup* plays an important role here: the document markup most often depicts only the layout structure of the document, i.e. how the document is formatted. This indicates that the markup in the document can be used for recognising parts of texts that have a specific graphical appearance (e.g. font type, font size, etc.) (Coombs *et al.* 1987). Such a markup does not assist in the task of recognising the essential -- the information elements. Conversion of text with a layout markup into hypertext only yields an electronic copy of the manual or design document in question.

The other major task in hypertext creation is that of generating the links between the information nodes. Conversion based on the layout markup is missing the point: the resulting links do not represent semantic interrelationships between the nodes. Entering the links manually, on the other hand, is very laborious and error prone. Furthermore, without any systematic approach for creating the links, they easily become somewhat arbitrary, reflecting the author's conceptions and even misconceptions about the target system.

In our approach the information elements are readily recognised and stored in the database. Moreover, the MFM model can be utilised to automatically incorporate the semantic links between the elements in a hypertext documentation. In our view, utilising an information model for generating hypertext links yields good results. First, the process produces links that have a semantic meaning. Second, the process is systematic, leading to a consistent structure of the documentation. Third, this approach improves maintainability: changes to the information model can be reflected in the documentation by simply relinking the documents.

5.4 SUPPORT FOR REUSE

Reuse is often limited to the reuse of existing designs - not the knowledge behind them (Figure 13 a). We propose that reuse of earlier designs could be made more dependable by systematically storing the related design knowledge in addition to the earlier designs and providing the designers an easy access to this knowledge (Figure 13 b).

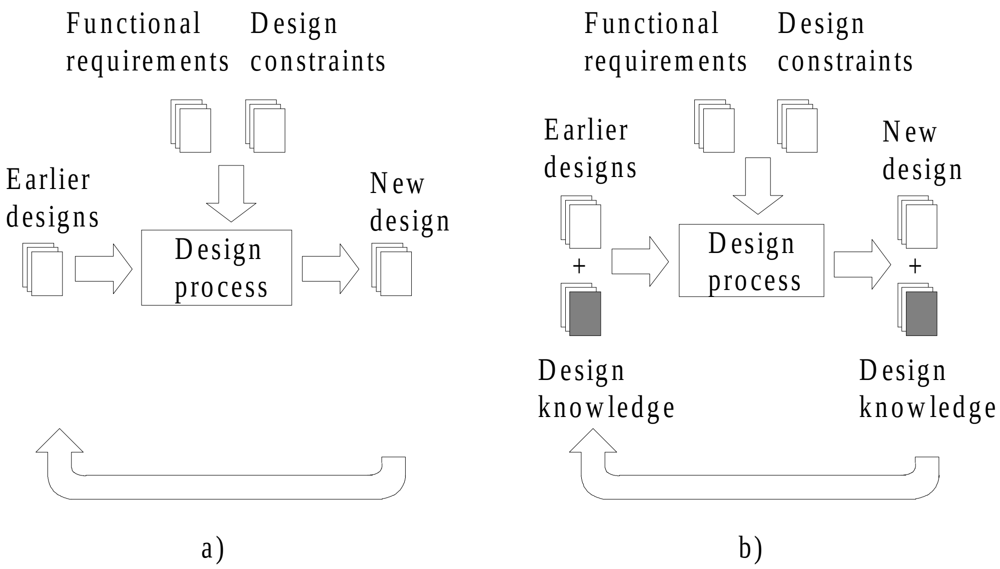


*Figure 13. Reuse of designs vs. reuse of designs **and** design knowledge.*

Our design environment supports reuse in two ways,

❑ reuse of complete or partial MFM models

❑ explaining existing designs.

The first alternative supports top-down design. Complete MFM models or individual higher level objects (goals or functions) can be reused. The model and its objects store decomposition knowledge, e.g. how the system can be decomposed into subsystems. In bottom-up design, the design knowledge stored in the environment has mainly a supportive role. It can be used to explain the existing designs and to assist the designer in estimating the effects of a change made at the implementation level. The latter alternative would serve the needs of the maintenance staff of industrial plants.

5.4.1 Reuse of complete or partial MFM models

MFM starts by recognising the goals of the plant to be built. For example, a model of a previously constructed power plant could be used as a basis for a new plant. This is due to the fact that the goals of power plants tend to be the same: producing electricity economically and safely. However, the priorities of these goals can vary a great deal. As an example, environmental safety goals are today much more important than a few decades ago. Furthermore, the goals and functions of a power plant intended to run only during energy consumption peaks are somewhat different from those intended to run constantly at the same nominal power. The function level of the model explicitly records the designer's intention: how the plant is meant to reach these goals. Due to the stable technology base in the process industry and the limited number of alternative designs this information is very likely to be reusable in succeeding projects.

An MFM model or any of its higher level objects store reusable decomposition knowledge both in graphical and object-oriented representations. The object-oriented representation of the model allows us to record several attributes describing the model's objects and their interrelationships. The tools that we have developed make it possible to reuse the information in the models. Both complete MFM models and their individual objects are stored in libraries where they can be retrieved for reuse. The structured documentation stored in the database is linked to these objects and can therefore be easily accessed.

Generally, a reuse support system should address

❏ design representation

❏ design storage and retrieval

❏ design adaptation and reuse.

Our design environment does not go that far. It does not feature any facilities for automatic retrieval of similar cases or their adaptation to fit the current problem. The main goal of our design environment was to make both artefact-related and design-related design knowledge explicit -- not to automate its use. This information is believed to assist the designer both in the retrieval and the adaptation of designs.

5.4.2 Explaining existing designs

The design knowledge stored in the environment can be utilised in explaining the implementation issues of a design. This is important because the designer may want to consider changes directly at this level. Since all the effects of a change may be hard to predict, the functional knowledge and the interactions between the subsystems provided by the MFM models can assist the designer in this task. Justifications regarding design decisions offer information that can help in *not* repeating previous errors.

We view explanation as navigation in the two hierarchies of a multilevel flow model (Huuskonen 1992). This view is based on the assumption that the upper level object in a multilevel flow model can give a purpose to the connected objects at the lower levels. Explanations can be found either explicitly, as structured text referring to the item under consideration, or indirectly, through its relations to other items. We define two ways of explaining designs: *explicit* explanations, referring to general design knowledge, and *derived* explanations, referring to related design choices. This is a straightforward approach to modelling design knowledge that does not allow for automatic understanding of the design, but will suffice for the purpose of explaining the knowledge.

Explicit explanations are structured text that can be attached to objects, their attributes, and to design rules. They contain a number of predefined keywords giving rationale for individual items, and freeform text for less formal definitions. A justification can be derived by traversing the network formed by the means-end relations. Indirect explanations are derived from other items through relations as shown in Figure 14. An item may possess explicit justifications of design knowledge, in which case the justifications are simply shown to the user. If an item does not have an explicit purpose, its neighbouring objects are searched recursively until an explicit purpose is found.

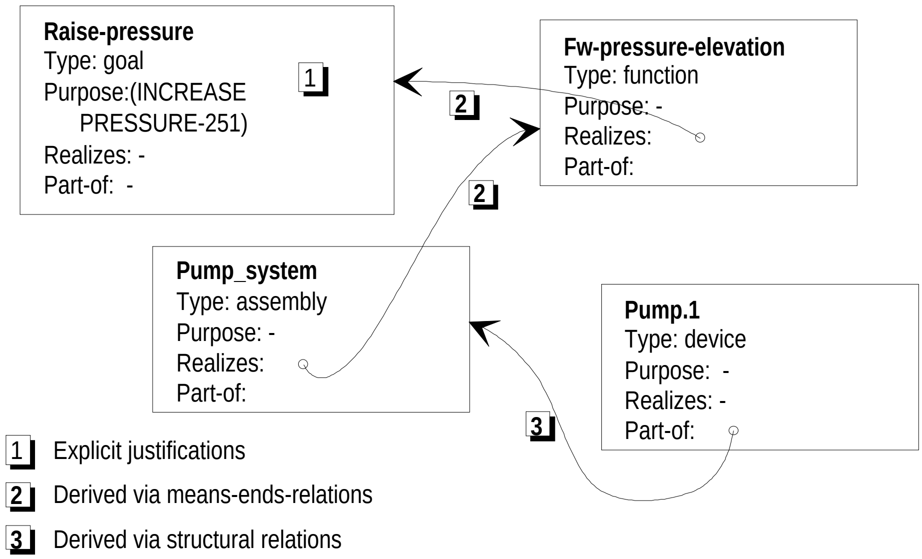


Figure 14. Explanations derived through relationships of an MFM model.

If means-end links (‘realises’ relation in Figure 14) are not defined for an item, the purposes can be derived through structural decomposition (part/subpart relations). The search continues until an explicit justification is found. Means-end relations are given the priority and, if they are not explicitly defined, an implicit relation is derived through part-of hierarchies. The number of potential relations is decreased, since several items can inherit a common purpose from their superiors.

5.5 CONVERSIONS

To enable the use of the knowledge in the MFM models in other applications, we developed conversion programs that convert the object-based representations in the design tool to data structures in C++ , Prolog, and ASCII files using special delimiters.

Building the on-line documentation system presents an interesting example of how the conceptual model created in the design phase (the MFM model) can be utilised at the later stages. For automatic generation of hypertext links into the on-line documentation, a *hypertext linker* was implemented (Paper VIII). The linker requires the conceptual model of the target system to be input in Prolog. Hypertext links are generated according to this information model to depict both the concepts of the domain and their interrelationships.

5.6 DISCUSSION

The need for support for conceptual design phase is apparent and several kinds of tools have been proposed. However, the proposed tools have remained at the level of research prototypes. The tools built so far vary along two axes. First, the systems vary in respect to the intended degree of automation of the design tasks. The systems range from the design expert, which uses a "traditional" expert system approach in which the tool takes care of the design task after the user has input the requirements, to "passive" storage of design information. The tools also differ in respect to the knowledge they represent, store, and handle. At one extreme the systems model only the designed artefact, at the other the design process.

Several automatic tools have been built for the mechanical engineering field (Taleb-Bendiab 1993, Zeid *et al.* 1993, Iivonen & Riitahuhta 1994). In process engineering, a good example is the Design-Kit tool (Stephanopoulos *et al.* 1987). It was built for the design of preliminary process flowsheets, the synthesis of plantwide control configurations, planning process operations, and the analysis and diagnosis of real-time operations. In addition to providing both a graphic and an object-oriented representation of the hierarchical models of processes, the system also features production rules and reasoning mechanisms for building knowledge-based applications.

A more recent example of a tool for conceptual design of chemical design processes is ConceptDesigner (Han *et al.* 1995). ConceptDesigner captures and utilises both declarative knowledge of the domain and procedural knowledge of the design process. The procedural design knowledge is used by a set of manager and design agents that are used to co-ordinate or solve the subtasks of the design problem. The human designer is allowed to define the scope of design, guide the direction of design evolution, and select among competing designs.

Andersson (1993) proposes the use of a design language called CANDLE for supporting conceptual design. The language is formulated upon the concepts that the designers use for describing and discussing the design tasks and solutions. The vocabulary of the language, together with its design grammar, is expected to supply designers with guidelines for how the design solutions should be created.

Bañares-Alcántara *et al.* (1994) emphasise the importance of modelling the process of design. They have developed an experimental design support system, KBDS, that models the design process as the histories of design alternatives and design objectives coupled with the evolving description of the plant.

Larsson has developed a graphical tool for building expert system applications based on MFM models (1992). The aim of his tool is thus clearly different from that of ours -- recording the design knowledge behind the designs. The tool, called the MFM toolbox, has been built on top of the commercial expert system development environment G2, and utilises its graphic features for the modelling. It also includes a rulebase for checking the syntax of the created MFM models and a set of rulebases allowing diagnostic reasoning once a model has been completed.

Our design environment is not intended for automating any design tasks. Its only purposes are to make artefact-related and design-related design knowledge explicit and store it for later use. Reuse of the stored designs together with the associated knowledge is manual: no support for retrieving similar cases is provided. However, reuse of earlier designs is supported by providing explanations based on the stored design knowledge.

The design environment built in this research does not explicitly model the design process, e.g. the design phases and their results. Therefore it does not provide guidelines for the designer on how to proceed in a design task in order to accomplish a specific design phase. Similarly, the tool does not provide direct means for project management. However, the document templates utilised by the tool supporting structured documentation do provide guidance on what to document.

Recording informal design knowledge using structured documentation provides a way to close the gap between the actual design work and the documentation. The idea of structuring the documentation of a technical system according to the product model is not new (Weich 1992, Magnusson 1994). However, the product models applied so far have been lacking the higher levels of abstraction present in the MFM models. As to recording the design history, we utilise the form-based approach of Design Journal (Conklin 1987). Our approach is different from theirs in the sense that design knowledge capture is integrated in the design environment and not done using a separate tool.

6 PRESENTING DESIGN KNOWLEDGE TO INDUSTRIAL PLANT OPERATORS

According to a literature review and interviews with plant operators performed during the TIESU project, industrial plant operators face many kinds of problems in their work, especially during process disturbances and other unexpected events. Several of these problems seem to be due to inadequate functional knowledge of the plant possessed by the operators (Paper II, Rasmussen 1985, Rasmussen 1986, Stassen *et al.* 1988, Huuskonen & Jaako 1992, Kaarela *et al.* 1992). The problems will be discussed in detail in the following sections.

This chapter describes our experience on how design knowledge could be presented to the users of an industrial plant to assist them in accomplishing their tasks. We present an approach that was utilised for implementing an operator support system prototype for two industrial cases. The first case was a feedwater system of a power plant (Paper VII, Leiviskä *et al.* 1994), and the second a gas system of a detector experiment at CERN (Paper VIII). The prototypes were built using mainly the information that could be obtained during the design phase using the approach for capturing design knowledge presented in this thesis. Capturing a major part of the knowledge required for such a system at design time is of primary importance, because the knowledge acquisition bottleneck has generally been accepted as the biggest obstacle to implementing operator support systems from scratch (Johannsen & Alty 1991).

A factor obstructing large-scale introduction of such systems has been the use of unfamiliar or exotic tools in the implementation of laboratory prototypes. The resulting systems in many cases are hard to integrate into an industrial automation system. Our prototypes were built upon a new generation automation system using mainly its standard features.

The prototypes of the operator support systems were not intended to take over some of the operators' tasks, which is the goal of several knowledge-based support systems (Hollnagel 1991). In our approach, the support system has a *supportive* role making some of the design knowledge explicit and easily available while human operators make the decisions. The emphasis was on functional knowledge that portrays how the plant was intended to work. Such knowledge is important in a great variety of operator's tasks that call for understanding the functional properties of the system as well as prediction of the consequences of the planned actions. Such tasks include fault diagnosis, planning of future activities, and explaining events in the process. The tasks often deal with unfamiliar situations, where learned skills or rules do not exist or do not suffice.

Problem solving is goal-oriented and often includes generation of alternative plans and selection among them. According to Rasmussen (1985, 1986), functional reasoning based on a *mental model* representing the internal structure of the system is typically required to perform these tasks. Stassen *et al.* (1988) share this view and state that an operator must possess a mental model[10] of the plant, the tasks, and the statistics of disturbances to successfully operate a complex plant. Such a conceptualisation allows the user not only to plan actions for novel situations but also to explain why a particular action produces the result it does (Carroll & Olson 1988).

Norman (1983) has presented a framework including a *target system*, a *conceptual model* about the target system, a *mental model*, and a *scientist's conceptualisation of the mental model*. This framework can be applied in transferring design knowledge to the operators. Designers have a mental model of the system they are designing which contains higher levels of abstraction even though their work most often results in documents describing only the implementation aspects of the target system. The model augmented with actual information about the target system could be expressed as a conceptual model using proper modelling techniques. The conceptual model created by the designer could then be used by other people (designers, operators, maintenance staff, etc.) when forming their own mental model of the system. However, in reality the situation is not that simple. The formation of a mental model is a complicated process that takes place whenever the user is interacting with the system. A mental model can not be transferred from one person to another but its formation can be affected by providing proper information that is learnable, understandable, and usable (Norman 1983).

According to Norman (1983), building the entire human interface on the same conceptual model would support the users' formation of mental models that are consistent with that conceptual model[11]. Our approach in presenting design knowledge to the operators is based on this notion. The information in a multilevel flow model (especially that represented in the means-end hierarchy) forms a conceptual model that has been created by the designer. To efficiently support operators in adopting this conceptual model while forming, complementing, or replacing their own mental models of the plant, the information in the conceptual model is consistently presented in the display hierarchies, alarm systems, and easily accessible on-line documentation, all of which are structured upon the same model.

---

[10]Stassen *et al.* use the term internal representation instead of a mental model.

[11]Norman calls a user interface based on the conceptual model a *system image*.

6.1 USER INTERFACES

6.1.1 Problems with current user interfaces

The operating personnel of an industrial plant are responsible for the achievement of the goals of their plant. Ironically, these goals are seldom explicitly presented in the control room. Information representing the higher levels of abstraction in terms of goals, functions, and equipment seems to be missing or sparsely shown (Rasmussen 1985). The user interfaces of the process automation systems have typically concentrated on the presentation of measured process data depending on the one sensor-one indicator principle (Rasmussen 1985). They mostly reflect structural design knowledge (Korhonen 1991).

The automation systems rely heavily on the concept of direct manipulation -- objects of interest (sensors, controllers, etc.) are continuously represented in the displays and they can be manipulated directly using a pointing device (such as a mouse or a trackball) and menus instead of commands obeying a complex syntax (Shneiderman 1987). Unfortunately, direct manipulation systems that are built based on device-level descriptions of the process (e.g. by dividing the P&I diagram into pieces of proper size) lack the semantics: it is the responsibility of the user to learn the purpose of the components of a graphic representation. For example, the operator must know what the effect of closing a valve will be. Thus, the advantage of direct manipulation without the semantic knowledge is limited to reducing the need for syntactic knowledge on how to manipulate the components.

In current automation systems, the evident complexity resulting from a huge number of indicators and alarms has been attacked by designing a series of hierarchical displays each of which represent a limited amount of information related to a specific process area. The information includes both continuous analog measurements (direct or abstracted) and discrete data about the states of various devices (e.g. the on/off state of a motor). Trend displays may be used to represent the time dependency of certain variables. The displays often are mimic diagrams depicting the physical or logical layout of the process. Such display hierarchies typically reflect the process hierarchy. Each of the lower level displays represents a specific part of the process while the upper level displays act mainly as an index for the lower level displays (Korhonen 1991). Such a hierarchy is based on aggregation along the whole-part decomposition of the process and can be used for identifying the state of the process in a bottom-up manner (Rasmussen 1985, 1986).

Aggregation alone cannot solve the problems resulting from complexity; it can not be used for reducing complexity -- it can only be used to group specific measurements together, for example according to a hierarchy, usually the process hierarchy. Without abstraction the number of measurements remains the same. Operators are in fact expected to infer the state of the process -- in relation to vaguely specified goals -- on the basis of a set of measurements. They monitor values of process variables that they think best reflect the status of the process and look for indications of familiar states of the system. In fact, such an abstraction represents a part of their mental model. While this kind of a strategy may be effective during normal operation, it will fall short during unfamiliar plant states (Rasmussen 1986). Moreover, differences between the mental models of individual operators lead to different ways of controlling the process. This implies less-than-optimum performance during some shifts.

6.1.2 MFM-based user interfaces

The user interfaces of automation systems so far have primarily depicted the structure or topology of the process. Since several of the operator's tasks, especially those which relate to knowledge-based behaviour, require functional knowledge of the process, it is reasonable to assume that not all these tasks can be optimally supported by providing only a single view of the process. In our work, we concentrated on providing another view of the process by explicitly displaying the means-end hierarchy to the operators and using it for abstracting the status of the process on the basis of the process measurements.

The user interfaces of the two prototypes facilitate representation of both functional and structural knowledge of the plant. In the prototypes, the means-end information is represented as a set of hierarchical displays (Figure 15). The highest level of abstraction represents the safety, economy, and production goals of the system (A). At the next level, there is a display for each goal category (safety, economy, and production) showing a detailed goal hierarchy (B). The next hierarchical level represents the abstract functions required to achieve a goal (C). Such a display consists of a graph of connected symbols that denote the functions. Finally, conventional displays based on device or process hierarchy (whole-part) represent the actual devices (e.g. pumps, valves, pipelines) that implement the abstract functions (D).

Traversal up and down the display hierarchy is performed using links corresponding the relationships of the means-end hierarchy depicting the system. Such a feature is useful in a user interface because the relevant level of consideration will vary among different tasks requiring knowledge-based behaviour such as diagnosis or planning of future actions (Rasmussen 1986). Switching between the levels is done simply by positioning the cursor on the object and clicking the activation key. Alternatively, the

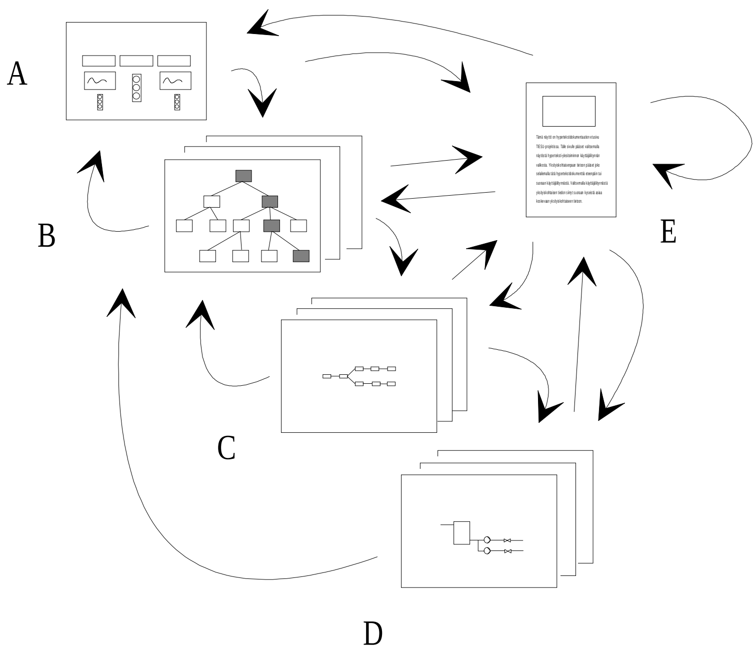


Figure 15. A display hierarchy built upon an MFM model.

operator may choose to have a simultaneous view of the various levels of abstraction by opening multiple windows.

Conventional displays representing the process or device hierarchy act as the basic displays of the system. These displays are basically piping and instrumentation diagrams of which the details have been omitted for clarity. All the direct manipulation commands (e.g closing a valve or changing a setpoint) are performed using these displays. The means-end information is presented to the operators using multiple windows providing additional information that the operator can turn to at will.

The key point of display hierarchies built upon a MFM model is that they raise the level of abstraction of the operators' work. The number of indicators to be monitored by the operator can be greatly reduced through abstraction in the means-end dimension. The goal level display makes it possible for the operator to monitor the process at the level of these explicit goals. Colour codes are used to indicate the achievement of goals both at the highest level (A) and at the level of the detailed goal hierarchy (B).

The introduction of such higher level displays presupposes that information processes to abstract the measured process parameters into higher levels of abstraction are defined (Lind 1991). This implies a clear definition exists for the working condition of every higher level node in the MFM graph which specifies via which measurements and calculations the achievement of a goal can be verified. The MFM model hierarchy ultimately shows *what* measurements are related which goal -- the information processes describe *how* they are related. The working condition of each node is checked at predetermined intervals, and the availability of the node is expressed with the colour coding of the respective object in the display (see the next section).

The strength of such a systematic approach is that the information processes become explicit and can thus be discussed and developed to indicate the state of the process in the best possible way. Another advantage is that the information processes become common to all operators.

6.2 ALARM PROCESSING SYSTEMS

6.2.1 Problems with current alarm processing systems

Alarms form an essential part of the user interface of a process automation system. Their purpose is to notify the operator of unexpected events related to the process itself, the process devices, or the control systems. The alarms are typically indicated on the mimic displays of the automation system screen. Automation systems often keep a log of the alarms, possibly with additional information on the related process area, priority, or time of occurrence. Dedicated alarm list displays can be used to browse the alarms grouped according to their chronological order, priority, or the process area they belong to. This information often proves valuable when trying to find the cause of an alarm or a burst of alarms.

Alarms cause several kinds of problems for the operators. First, it is often difficult for the operator to determine what the actual purpose of the alarm is and what to do with it. This indicates a need for additional information about the alarm: why the system is alarming, the severity of the alarm, the consequences of the alarm, the side effects, and the correct recovery procedure. Second, during abnormal conditions alarms often come up in bursts making it difficult to differentiate between the primary and secondary, i.e. consequential, alarms.

Despite the large body of research done within the field of alarm processing and the technical facilities provided by modern automation systems, systematic approaches for alarm design seem to be missing and the alarms are designed in a bottom-up fashion. For example, alarms based on measurements of analog process variables are generally built using the one-sensor-one-indicator principle in which each measurement is equipped with a number of predefined alarm limits. The limits may refer to either the absolute value of the measurement, or its first or second derivative. Generally, the limits are fixed and do not take into account the varying situations in the process (e.g. operational states of the process). An alarm is raised whenever a process variable exceeds such a limit. Of course, current automation systems do provide means for coping with alarms that are dependent on the operational state of the process, subprocess, or device. The alarms can be either conditioned (i.e. they are generated only in a specific state) or suppressed (alarms are generated and collected into an alarm log but not displayed) using the features of the automation systems. Another approach is to set the alarm limits individually for each operational state to avoid (state-dependent) false alarms (Doyle *et al.* 1989). Summary alarms combining several more detailed alarms are often used for reducing the number of alarms to be displayed to the operator. Corsberg has proposed an approach for alarm filtering that uses a network of connected alarms indicating their causal relations (Corsberg 1987, Bray & Corsberg 1994). Using four types of causal relations (level precursors, direct precursors, required actions and blocking conditions) it is possible to dramatically suppress the number of alarms displayed to the operator. Model-based alarming is based on several process variables and a model of their interaction. A potential fault is detected and an alarm is given if the actual process values differ from those predicted by the model.

6.2.2 MFM-based alarm processing systems

Multilevel flow models provide two important means to enhance current alarm processing systems. First, the hierarchical nature of the means-end model facilitates a way to manage complexity. Alarms can be generated and displayed at several levels of abstraction. At the goal level, alarms are given to inform the operator if the system fails to achieve its goals -- not when any individual measurement exceeds a predefined alarm limit. Of course, the operator is able to view the device-level alarms at will. Since alarms are basically always based on measurements of process variables higher level alarms require processing of the measurements (Lind 1991). Second, the mass and energy balances of the individual flow functions provide a reference model of the plant's intended behaviour. This information can be used for detecting deviations from the normal operation of the plant and recognising the primary and secondary alarms in case of an alarm burst (Larsson 1992, Sassen 1993, Van de Ree 1994).

The means-end hierarchy also provides a guideline for a more comprehensive alarm design. Instead of designing an alarm and alarm limits independently for each control circuit one could take the approach of checking the achievement of goals as a starting point. An MFM model explicitly states what functions are required to achieve a goal, and what devices (including the control circuits and the measurements) implement each function. This means that the availability of all the functions attached to a goal are a precondition of its achievement. The availability of the functions depends on device-level objects that can be monitored using process measurements.

In the prototypes, we demonstrated the use of higher-level alarms based on the means-end hierarchy. For each object in the higher-level displays, a set of alarm limits (low alarm limit, low notification limit, high notification limit, and high alarm limit) was defined. The value of the object was calculated using the definition of its working condition referred to in the previous section. Crossing an alarm limit is indicated with a change in the colour of the object in the display (green indicating the normal state, yellow indicating a minor deviation from the normal state, and red a disturbance). In the prototypes, only the safety-related alarms were implemented. In the cases considered, a failure in any of the related functions caused an alarm at the goal level. The information processes for abstracting the alarms were in this case simply logical OR-functions.

Such an alarm system using the MFM-based displays supports the operator in diagnosing process disturbances. If something goes wrong, the operator can find the cause and effects of a disturbance or device failure on the overall goals by following the display hierarchy based on the relationships in the MFM model from the abstract goal level displays to the more concrete levels. In an alarm situation, the hierarchy supports the operator in the analysis of the cause of the problem in question, using the traversal of the alarm tree indicated on the display. Figure 16 shows an example of a detailed goal hierarchy where the sub-goals that are not achieved are shown in a different colour on the automation system screen.

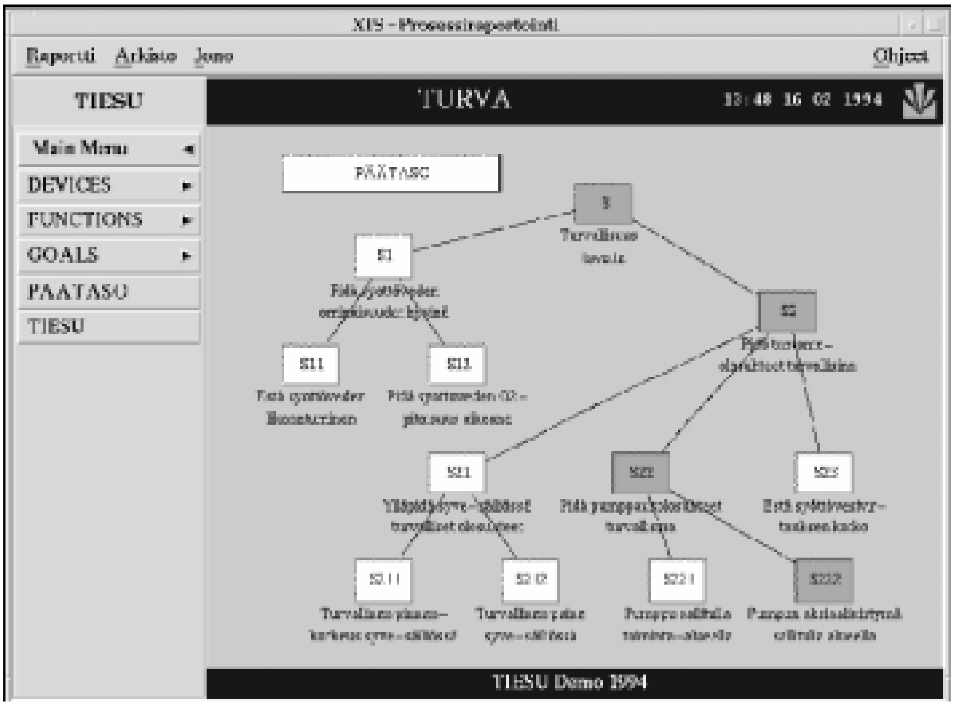


Figure 16. A display depicting the traversal of an alarm in the goal hierarchy.

6.3 ON-LINE DOCUMENTATION

6.3.1 Problems with current on-line documentation systems

On-line documentation systems, especially hypertext-based ones, have been regarded as promising alternatives in improving the accessibility of plant documentation. However, such systems are not yet very common in industry. There are two main reasons for this. First, the systems are expensive to build and maintain using the existing paper documents as the primary source of information. Second, most on-line documentation systems so far have not been accepted by the users due to difficulties of their use.

Most on-line documentation systems built so far have been separate systems with dedicated user interfaces. The operator has to turn away from the control system to obtain information from another system. This is not desirable, especially during disturbance management when the need for information would be most urgent. The second user interface adds to the operators' cognitive load: they have to learn how to use it. For these reasons, separate systems with dedicated user interfaces are doomed to be neglected by the operators no matter how useful they could be (Pernu 1992).

Typically, a hypertext system is used by browsing through the document following the links between its nodes. The purpose is to search for relevant information without knowing what exactly should be searched for or where it is located. User disorientation -- getting lost in a hypertext document -- is a commonly recognised problem related to browsing. A related problem is cognitive overhead resulting from the need to make decisions of which link to follow (Conklin 1987). Both problems are important for technical documentation for industrial plants, due to the huge amount and complexity of documentation.

The disorientation problem is closely related to the problem that the readers do not possess a sufficient understanding of the domain -- the process they are assigned to. Their mental model of the system does not cover all the concepts of the domain, not to mention their interrelationships. This holds true especially in the case of complex technical systems (Tyrväinen 1994). Inadequate knowledge of the domain concepts and terminology limits the usability of the search mechanisms provided by several hypertext systems. Badly selected keywords or index terms in user-generated queries often result to either no or too many links to follow.

In industrial plants, documentation is typically needed to solve technical problems. Operators need the information quickly, especially during emergencies: there is no time for timely seeking operations. Clearly, neither browsing nor keyword or index-based search mechanisms sufficiently support a user's retrieving information from technical documentation. A promising approach is to support the user with a model representing the domain. There is a well-known analogy between hypertext and semantic networks with both representing concepts and their interrelationships (Conklin 1987, Carlson 1989, Snaprud & Kaindl 1992, Rada 1992, Jonassen 1993, Tyrväinen 1994). This provides a way to support the user in navigating in the hypertext. A semantic network portraying the domain knowledge can be represented to the user as a graph to support finding a relevant node (Halasz 1988).

6.3.2 An approach for improving on-line documentation

We propose the introduction of hypertext-based on-line documentation systems into the control rooms to improve the accessibility of the plant documentation. To overcome the operator's problems of using an on-line documentation system that is separate from the primary tool -- the automation system -- we propose that the documentation be displayed on the automation system display.

Our work differs from prior mainstream development efforts in several aspects. A key point of our approach is to utilise the results of the design phase. Using the approach and tools discussed in the previous chapters, much of the material to construct a hypertext on-line documentation system is readily available after the design phase. The conceptual model born in the design phase is utilised for structuring the hypertext, creating semantic links between the nodes of the hypertext, and assisting the user in the task of searching relevant information.

The approach taken in this work emphasises the opportunities permitted by the similarity of semantic structures of the hypertext documentation and the user interfaces. In our work, the semantic structure is that of the means-end hierarchy of the multilevel flow model representing the target process. The hierarchy depicts *functional* design knowledge of the process. The purpose of displaying this semantic structure in the user interfaces is to provide assistance for the users when navigating in the information space: it supports them in finding the desired pieces of information. It is believed to lessen the cognitive overload of the operators: they do not have to memorise the relationships between the concepts in the documentation. The developed prototypes pursue this goal by two means,

- ❑ the use of the automation system's displays as graphical maps to the documentation
- ❑ the presentation of typed links (based on the information model) to the user.

Presenting the same conceptual model in both the user interface of the automation system and in the structure of the on-line documentation system allows the use of the automation system displays as graphical maps to the documentation: for each object in the display hierarchy there is a node (or a set of interrelated nodes) in the documentation. The document can be accessed by simply pointing and clicking on the object in the display as depicted in Figure 17.

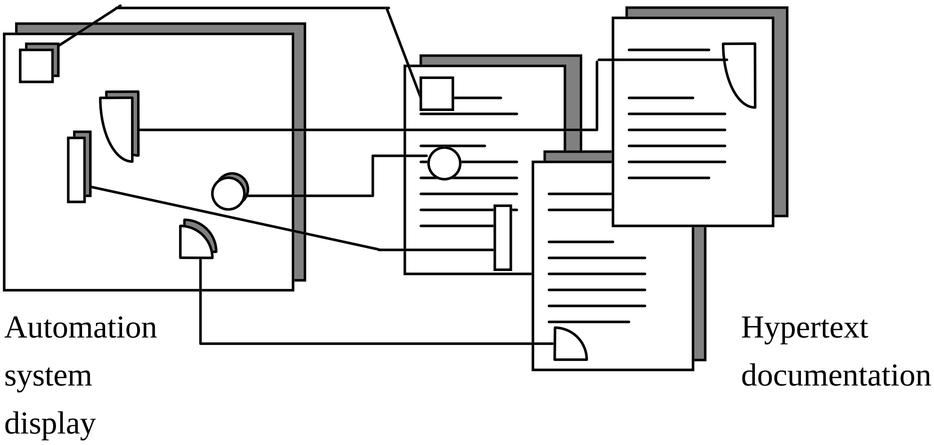


Figure 17. An automation system display as a graphical map to the on-line documentation.

Structuring the hypertext according to a conceptual model corresponds to providing it with a high-level semantic structure. Displaying the type of the semantic link to the user is essential because it can provide the reader with information about the content of the destination node. According to Halasz (1988), such links support the reader in navigation. The items of the bulleted lists in Figure 18 represent some examples of such typed links[12]. The bulleted lists have been automatically generated using the hypertext linker based on the means-end hierarchy of the multilevel flow model. The items 'ACHIEVES' and 'HAS-SUBPARTS' denote the links starting from the function 'PREVENT OVERPRESSURE'. Thus, 'PREVENT OVER-PRESSURE' has a subpart 'ADD FRESH GAS' and achieves the goal 'PROTECT CHAMBERS'.

For clarity, only the hypertext window is displayed in Figure 18. In the prototypes, the hypertext window would typically be shown on the

[12]The example is taken from the gas system documentation of the L3 experiment at CERN authored by David Peach (1992). It was used as the source material for developing the demonstration system for the CICERO project at CERN. Published with permission from CERN.

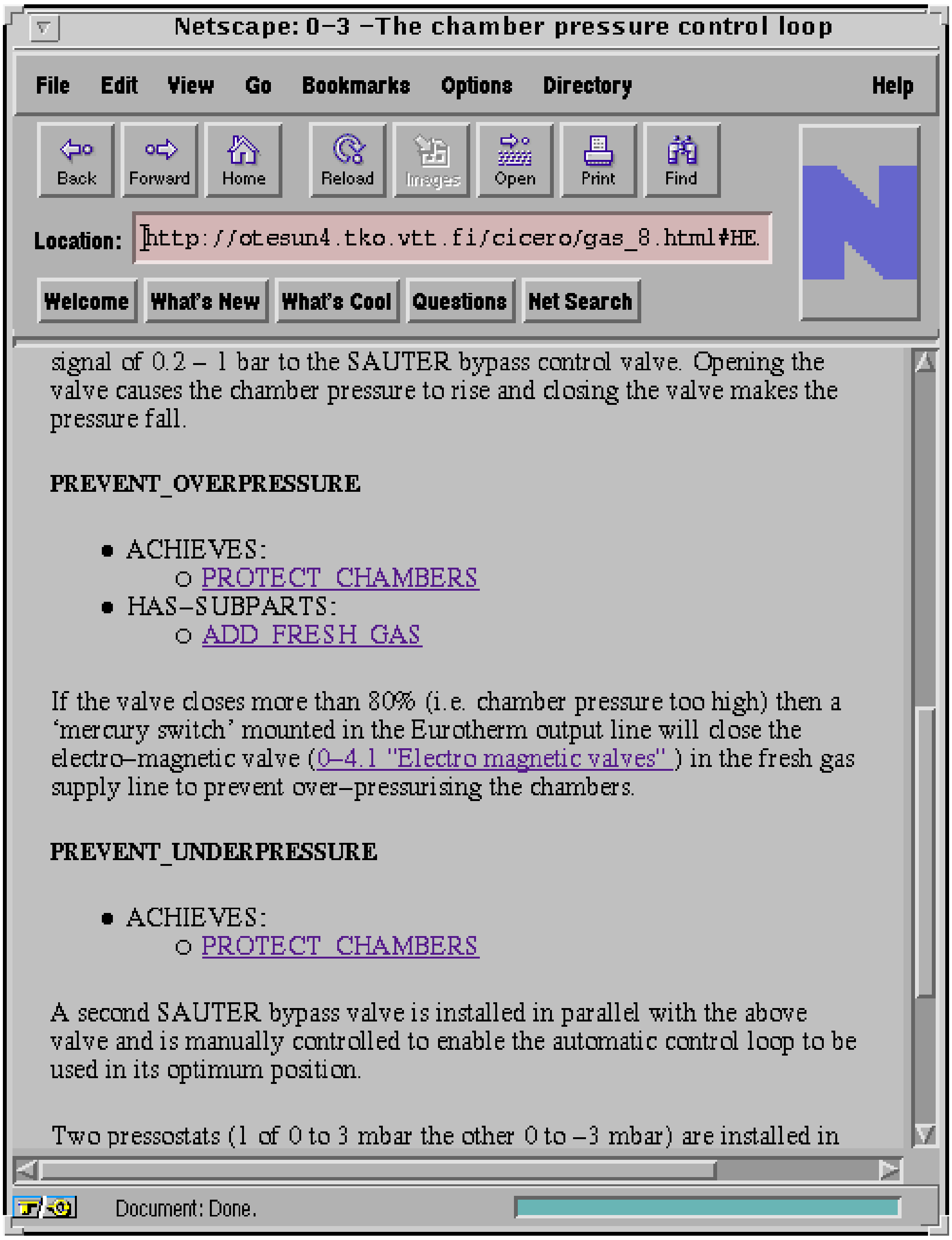


Figure 18. A view of automatically generated hypertext links in the L3 Gas System Documentation.

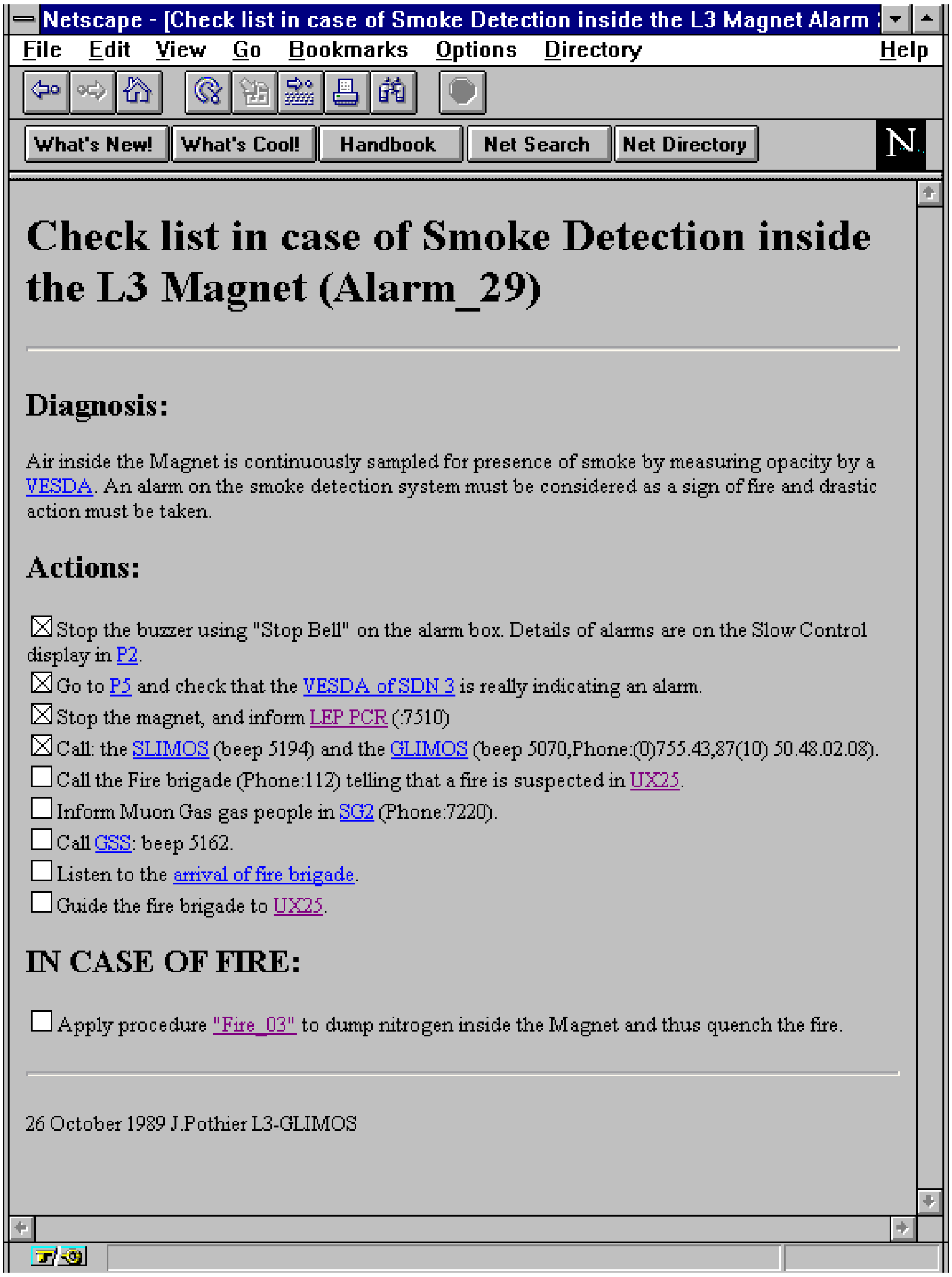
automation system monitor. However, the use of general purpose software for the implementation of the on-line documentation system allows easy access to plant documentation from any computer connected to the plant network. The implementation of the prototype systems is described in Paper VII.

Functional knowledge about the plant alone is not enough for the operators. They need procedural information about their tasks. As described in Chapter 4, an MFM model can be used as a basis for allocating the control tasks between the automation system and the human operators. The developed on-line documentation system provides a structured way to store procedural knowledge on how to perform the tasks of the human operator. The tasks often consist of a set of actions that have to be executed sequentially. Such sequences of actions can be easily represented as structured text. The on-line documentation system provides ready access to the information. For example, in the case of an alarm the respective alarm procedure is just one mouse click away -- the procedure can be accessed by clicking the symbol of the alarming measurement.

Figure 19 shows an example of an alarm procedure giving detailed instructions to the operator of the L3 experiment at CERN in case smoke is detected within the L3 magnet[13].

---

[13]The example is based on the paper version of the alarm procedure authored by Mr. J. Pothier of CERN. Published with permission from CERN.



Netscape - [Check list in case of Smoke Detection inside the L3 Magnet Alarm

File Edit View Go Bookmarks Options Directory Help

What's New! What's Cool! Handbook Net Search Net Directory

Check list in case of Smoke Detection inside the L3 Magnet (Alarm_29)

Diagnosis:

Air inside the Magnet is continuously sampled for presence of smoke by measuring opacity by a VESDA. An alarm on the smoke detection system must be considered as a sign of fire and drastic action must be taken.

Actions:

- ☒ Stop the buzzer using "Stop Bell" on the alarm box. Details of alarms are on the Slow Control display in P2.
- ☒ Go to P5 and check that the VESDA of SDN 3 is really indicating an alarm.
- ☒ Stop the magnet, and inform LEP PCR (:7510)
- ☒ Call: the SLIMOS (beep 5194) and the GLIMOS (beep 5070,Phone:(0)755.43,87(10) 50.48.02.08).
- ☐ Call the Fire brigade (Phone:112) telling that a fire is suspected in UX25.
- ☐ Inform Muon Gas gas people in SG2 (Phone:7220).
- ☐ Call GSS: beep 5162.
- ☐ Listen to the arrival of fire brigade.
- ☐ Guide the fire brigade to UX25.

IN CASE OF FIRE:

- ☐ Apply procedure "Fire_03" to dump nitrogen inside the Magnet and thus quench the fire.

26 October 1989 J.Pothier L3-GLIMOS

Figure 19. An example of task-related knowledge: an alarm procedure.

6.3.3 Storing experiential knowledge

This research focuses on design knowledge. However, not all the knowledge related to a plant is available as a result of the design phase. Operators and other staff in the process industry gain a great amount of experiential knowledge, especially related to disturbances of the process, malfunctions of its devices, or ways of recovering from them (Stephanopoulos 1987). Up until now, there has been no structured way to store such information. The knowledge mainly remains the property of individuals. Some of it may be communicated orally, some written on notes or in computer-supported logs. Because of the importance of this kind of knowledge the above means of recording it seem insufficient. Information told by one operator to another may easily be misunderstood and is bound to be forgotten. Notes written on paper tend to get lost and, even if properly stored, are hard to access as their number increases. Computer logs are typically poorly organised -- in some cases they are just sequential files where each new note is added at the end. Search for the notes relevant to the current situation easily becomes a major task.

An MFM model provides a structured means of collecting and disseminating experiential knowledge emerging during the use of the plant. By structuring the documentation of the plant according to the MFM model, a framework for recording some of the experiential knowledge in a structured manner is created. Such an on-line documentation system is extensible: users can attach annotations to the documentation at will. With the semantic structure of the documentation, the annotations are systematically linked to the concepts of the MFM model. It is therefore possible for the users to get a view only of those notes related to a specific valve, for example. The approach is especially suitable for recording disturbances, their causes, and procedures taken to recover from them.

6.4 TRAINING

Newcomers at an industrial plant receive training for their new position. The purpose of training is to give them sufficient knowledge of the plant to be able to successfully operate it. Especially in the case of new plants the operators and other staff attend courses where the operating principles of the plant and its subsystems are taught. Process simulators are frequently used for training, especially in safety-critical, e.g. nuclear power, plants. With new plants, the commissioning process has been found to be an effective time for training the operating and maintenance personnel at site. During the commissioning the plant and its operating procedures are thoroughly tested. Commissioning generally includes running the plant in production use before its approval by the customer. Commissioning provides a good opportunity for the operators to get acquainted with their plant and get expert advice from the representatives of the vendors.

Such training often concentrates on teaching procedures of how to act in different situations. For example, the startup and shutdown procedures are thoroughly clarified to the operators. This kind of training is particularly useful for teaching the normal operation of the plant. The trainees can familiarise themselves with the *expected* operational sequences required in the operation of the plant. However, the operators may later encounter difficulties when meeting unexpected situations where the learned rules do not apply and knowledge-based behaviour is required. To reduce these difficulties, training should concentrate on better understanding of the plant, i.e. improving the operator's mental model of the plant (Stassen *et al.* 1988).

Norman (1983) emphasises the relationship between a conceptual model of a system and the mental model of its user. Training can be regarded as teaching the conceptual model to the user who will form a mental model that hopefully is consistent with the conceptual model. This can be best achieved if all the training material and instruction is based on that conceptual model. From the point of view of the formation of a mental model, the training phase is of great importance, especially if the trainee has only little prior knowledge of similar plants.

Training the operators of a complex plant is not a one-shot effort. The operators may not be able to assimilate all the information at once and they may forget it if they do not need it for some time. Their mental model evolves as they operate the plant. Much of the knowledge of the plant comes only with its operation and maintenance. Therefore, training should be supported throughout the plant's operation. The on-line documentation system presented in Section 6.3 provides easy access to plant documentation and an effective way to disseminate the experiential knowledge gained during the plant's operation among the people involved.

6.5 EXPLANATION MECHANISM

Chapter 5 presented a facility of the design environment for explaining existing designs to the designers. The system is capable of producing explanations of the plant functionality, its structure, and the design decisions based on the MFM hierarchies, through structured text. Since maintenance includes making changes to existing designs, such an explanation facility embedded in an automation system would be useful for the maintenance staff of industrial plants, to provide them with currently missing information on the original design intention, purposes of process and automation equipment, and justifications of design decisions.

Such a system could be embedded in an automation system fairly easily. The MFM model hierarchy can be converted into a C++ class hierarchy using the conversion tools of the design environment. The algorithms originally written in Lisp could be coded in C or C++ to allow embedding within the automation system.

6.6 DISCUSSION

Representation of functional knowledge to plant operators was one of the main reasons for developing the MFM modelling technique. Vicente (1992a) presents a comprehensive study of the large body of work done with MFM-based user interfaces and alarm processing systems. Several researchers have proposed the presentation of means-end information to the operators in addition to the information provided by the conventional displays (Rasmussen 1985, Rytoft *et al.* 1990, Lind 1991, Duncan & Praetorius 1992, Easter 1992, Larsson 1992, Sassen 1993, Vicente 1992a, Vicente 1992b, Bisantz & Vicente 1994). Larsson (1992) and Sassen (1993) have used the MFM models as a basis for diagnostic reasoning about the process. Bisantz & Vicente (1994) have applied the abstraction hierarchy to provide the user with explanations regarding the diagnostic reasoning.

Our approach for presenting the information to the users was similar to the earlier work. However, the case applications were implemented in a state-of-the-art automation system using its standard features such as support for multiple windows. This allows the use of simultaneous multiple views of the process at several levels of abstraction relieving the operator from having to switch back and forth between the levels. It also allows the operator to have a constant view of the conventional displays and to get supportive information from the higher level displays.

Compared to other work developing MFM-based user interfaces, this research introduces an on-line documentation system structured according to the same conceptual model as the automation system's displays. The approach was evaluated in a test that was performed as a part of the CICERO collaboration (Takalo 1995). The evaluation concentrated on the efficiency of use of the documentation, the criteria being the average time required to perform a set of information retrieval tasks and the correctness of the answers to the questions presented in the tasks. The results showed an improvement both in the access time and correctness of the answers by the testees using the system developed according to the principles developed in this research as compared to paper documentation and a hypertext version following the layout structure of the paper documentation. However, due to the limited number of testees the results can only be regarded as suggestive.

The use of on-line documentation in automation systems has also been studied by Vanhainen & Paunonen (1994) and Rytoft *et al.* (1990). However, their approaches neither emphasise the structure of the hypertext nor its systematic creation during the design phase. Tyrväinen (1994) has proposed the use of domain and task models to assist the user in finding the relevant documentation. His work does not stress the creation of the models or structuring the documentation accordingly at design time.

7 INTRODUCTION TO THE PAPERS

This chapter gives an overview of the papers that form the basis of this thesis. In the following sections, the content of each paper is briefly discussed.

7.1 PAPER I, AN OVERVIEW (ICARCV '92)

Paper I is an introductory paper reflecting some initial ideas behind this research. Prior to this research, the author's main interest was that of building embedded knowledge-based systems. This can be seen in the problem setting: the difficulty of knowledge acquisition is highly emphasised in the paper. The knowledge acquisition bottleneck led the author to consider improving the recording of design knowledge at design time. The main points in Paper I are:

- ❑ the need for systematic knowledge acquisition during the design process
- ❑ goal-oriented definition of the tasks of the automation system and the operator
- ❑ Structured knowledge representations (e.g. object databases) allow the use and refinement of the design knowledge throughout the design process and even at the plant during its utilisation phase.

7.2 PAPER II, NEEDS ANALYSIS (ICO '93)

Paper II presents the results of a study conducted at the beginning of this research. The study concentrated on the information needs of designers in a multidisciplinary design project, and the operators and maintenance staff of an industrial plant The study was performed via unstructured interviews of the process and automation designers, the operators, and maintenance staff of a 150 MW peat power plant built jointly by our industrial partners a few years earlier.

The main points found in the study are:

- ❑ Poor design practice (discipline) leads to delayed, false, or missing information.
- ❑ Incompatibility of concepts, representations, and design tools existed between organisations participating in a joint project.
- ❑ Work distributed among several organisations causes consistency problems in the absence of a common design database.

- ❑ Designers concentrate on implementation details when writing the documentation.
- ❑ Current representations for design knowledge are not expressive enough. They cannot convey higher level concepts regarding the design, such as the purposes of the process or its subsystems or interdependencies between various parts of the process.

Paper II proposes a systematic knowledge acquisition procedure for the design process to remedy some of the problems found in the study. The use of multilevel flow modelling is proposed for capturing the higher level knowledge behind a device-level design. The supporting tools developed in this research are also described. Our ideas continued to develop since the writing of Paper II, so that Paper III describes our current approach to design knowledge more accurately.

7.3 PAPER III, ORGANISING DESIGN KNOWLEDGE (IFIP '94)

Paper III sums up our work on capturing design knowledge using MFM as a framework. The most important notion in this paper is that in addition to representing *artefact-related* or functional knowledge (i.e. how an artefact was meant to work) MFM can also be utilised to store *design-related* knowledge, i.e. knowledge about the design process: design issues, alternatives considered, design constraints, and evaluation criteria.

Paper III describes the tool developed during this research to support the modelling effort. The tool is based on a commercial knowledge-based object-oriented design tool. Our tool features

- ❑ storage of the model in an object hierarchy
- ❑ a graphical tool for drafting MFM models
- ❑ structured documentation of the model's objects
- ❑ versatile conversions of the model information to other languages such as C++ and Prolog to facilitate its use in other applications.

The results of a case study in which the tool was applied are reported.

Paper III also briefly discusses the role of design knowledge in the reuse of existing designs.

7.4 PAPER IV, RECOGNITION OF SAFETY GOALS (SAFEPROCESS '94)

Paper IV proposes the use of MFM to represent the safety-related information during the course of the design process to enhance its communication among designers, safety analysts, and the operating personnel of the plant. The explicit documentation of the safety-related aspects (possible hazards, safety goals, and measures taken to achieve the goals by preventing or limiting specified hazards) makes the designs understandable and appropriate for open discussion and constructive criticism.

Unfortunately, MFM does not provide any support for recognising the goals of an artefact to be modelled -- it only provides a means of representing the goals and how they were meant to be achieved. In Paper IV, we present some elementary ideas on how to recognise safety goals of a system, in this case an industrial plant. Even though these ideas are well-known within safety analysis research, representing safety-related information in the MFM framework throughout the design process and during the operation and maintenance of the plant differs from the approach generally applied within the safety community.

7.5 PAPER V, EXPLAINING DESIGN KNOWLEDGE (HCI '95)

Reuse of earlier designs is common in industrial design. To improve efficiency and quality of design work, earlier designs are regularly taken as a basis for new ones. To be able to *safely* modify a design, one should be aware of the original intentions behind the design, its design knowledge. This is regrettably not always the case: novice designers reuse and make changes to earlier designs with "the art of good guessing".

Paper V describes a possible way to explain design knowledge through the hierarchies of a multilevel flow model. An explanation facility was integrated into a design tool to allow the designers to ask for explanations about the design. Such an approach could have a major impact on the nature of reuse: reuse would be based on design knowledge -- not just the end results of the design work.

7.6 PAPER VI, PRINCIPLES FOR REPRESENTING MFM INFORMATION TO OPERATORS (HCI '93)

Paper VI forms the backbone of our ideas of how to represent design knowledge to the users of an artefact. The principal idea of Paper VI is to build the entire human interface of a system based on a conceptual model to support the users of the system in forming a consistent mental model of the system. In Paper VI, the concept of a user interface consists of

- ❑ the displays of the automation system
- ❑ the on-line documentation of the system
- ❑ a mechanism capable of explaining the system's functionality based on the design knowledge.

7.7 PAPER VII, AN OPERATOR SUPPORT SYSTEM BASED ON MFM INFORMATION (HICS '94)

In Paper VII, a demonstration system based on the approach represented in Paper VI (except for the explanation mechanism) is presented. The demonstration system features:

- ❑ an MFM-based display hierarchy
- ❑ an MFM-based alarm system
- ❑ an integrated on-line documentation system.

The demonstration system was implemented in an industrial automation system environment and the principles developed are thus easily *applicable* in similar applications.

7.8 PAPER VIII, ON-LINE DOCUMENTATION

Paper VIII presents a demonstration prototype developed for CERN, emphasising the work done within the field of structured documentation. The main point is the use of the multilevel flow model of a system for

- ❑ structuring the documentation of an industrial plant
- ❑ generating semantic links between the various parts of the documentation
- ❑ using the displays of the automation system as a means of graphically accessing the documentation.

7.9 PAPER IX, EXPLAINING DESIGN KNOWLEDGE TO OPERATORS (IEA&AIE '95)

In Paper IX, the explanation of control logic in industrial plants to plant operators is discussed. Explanations are considered to fall into three main categories: explaining the meaning, explaining the behaviour, and providing speculative help (what-if analysis). Of these, the first category is closely related to the topic of this thesis. The paper proposes the use of the means-end information of a multilevel flow model to explain the meaning or purpose of control logic. This is derived from the assumption that the purpose of any part of a system can be explained through the function it was designed to serve in the system. Through the use of a simple algorithm, the relationships in a multilevel flow model can be followed to answer such queries.

8 CONCLUSIONS

Industrial plant projects suffer from problems in communication among designers. The differences in the expertise, concepts, vocabulary, and representations of the various design disciplines often lead to problems in understanding and misconceptions of each other's design documents. The representations and tools used in industry normally allow for representing merely the end result of the designers' work, i.e. the physical implementation of the artefact, ignoring much of the design knowledge that led to the specific design. In this research, design knowledge was divided into two categories: artefact-related, depicting the intended functionality of the resulting design, and design-related, recording the design decisions and their rationale. The goal of this research was to improve the recording and expand the utilisation of design knowledge.

We hypothesised that multilevel flow models can be used for structuring, recording, and presenting design knowledge. The given hypothesis is supported by our developing an approach and supporting tools for recording and structuring design knowledge using an MFM framework. Multilevel flow models allow formal representation of the functional knowledge about the artefacts via means-end hierarchies. Our work emphasises that the models should be created during the design process as a systematic part of designers' routine work. The thesis shows that a multilevel flow model can be used for structuring and recording the typically informal knowledge regarding the design decisions by attaching the knowledge to the objects of the model as structured text elements.

A prototype design environment supporting the developed framework was built as an extension of a process design application implemented on top of a commercial software package. The process design application utilised a product model stored in an object-oriented database to record design information. The product model was, however, restricted to representing the physical structure of the artefact, i.e. the *devices*. Our prototype adds the *goal* and *function* levels of MFM, allowing the representation of conceptual functional design knowledge. To shorten the gap between the design work and the writing of design documents, a structured documentation system for recording descriptions of the concepts of the domain, design decisions, and their rationale was implemented in the environment.

A primary goal of our design environment was to make design knowledge explicit. The design environment was not intended to automate design, but to make reuse of earlier designs more efficient and dependable by providing the designers with relevant information about both the original design intent and the design history. For this purpose, the design environment was facilitated with a mechanism capable of providing explanations based on the

MFM models of the designs. The explanations assist the designer in recognising similar cases, altering a design to meet the new situation, and predicting the effects of a proposed change.

The approach and tools presented in this thesis have so far been used in two industrial cases which concentrated on subsystems representing limited, yet significant, parts of whole plants. The resulting models consisted of 1600 and 400 objects, respectively. As the plants already existed, the MFM models were reverse engineered using design documents and interviews with designers as information sources, and validated via walkthroughs with the designers. Such situations can be regarded as typical in several fields of process engineering since a high percentage of new projects aim at modernising existing plants.

According to the experience gained, modelling systems with MFM requires thorough expertise in the application domain, which should not pose a problem for professional designers. Problems could, however, be caused by having to learn the new modelling technique and the absence of guidelines of applying it. Consequently, the introduction of the approach presented in this thesis into an organisation would require both education of designers and the creation of MFM guidelines.

The case studies indicate that the approach presented in this thesis is able to capture design knowledge in industrial plant projects. To obtain more accurate information on the applicability of the approach for comprehensive industrial design projects in general, and for modelling the interactions between the subsystems of a plant in particular, would require the application of the approach in a large-scale industrial project.

The design environment developed in this research concentrated on process design. Since a multilevel flow model of a plant can also represent concepts related to several other design disciplines, MFM could provide a basis for concurrent engineering of industrial plants. Developing a system for supporting concurrent engineering remains an interesting topic for further research.

Two operator support system prototypes were developed, using the approach reported in this dissertation, to show the usability of MFM information at the plants. The value of presenting the means-end information to the plant staff in the displays of the automation system has been shown by several studies (Duncan & Praetorius 1992, Sassen 1993). As a new feature, our research introduced a hypertext-based on-line documentation system structured according to the MFM model of the plant. The system facilitated direct access to the documentation from the automation system's displays and semantic hypertext links based on the MFM model of the plant. Our study of providing on-line documentation reflecting the MFM structure of the system to the users gave encouraging results (Takalo 1995). The testees using our system could find more correct answers to the test problems in less time than those using a paper version of the documentation or a conventional hypertext version.

The prototypes showed that it is feasible to implement MFM-based sources of information for the operators of industrial plants using commercial state-of-the-art automation systems and recent information technology developments such as the World Wide Web. The resulting solutions are directly applicable in other applications. More work remains to be done on how and when to represent MFM-related information to plant operators. The user interfaces demonstrated in the prototypes only provide a good starting point.

Our development of the two operator support system prototypes indicates that the cost of developing such systems, which is generally regarded as the biggest obstacle to their construction, can be significantly reduced when a systematic design approach is followed in the plant design project. The design knowledge recorded using our approach constitutes a major part of the information required for supporting many of the operators' tasks, especially those which relate to knowledge-based behaviour. The knowledge was readily obtained for the prototypes from the object-oriented database of the design environment presented in this thesis. The automatic generation of semantic hypertext links based on the MFM model of the plant was found especially useful.

The research described in this thesis has already found some practical applications. We are currently re-implementing three on-line documentation systems for industrial organisations representing three quite different application fields: telecommunications, heavy machinery, and process industry. The on-line documentation systems to be found in these applications rely heavily on the information model concept described in this work. The concepts of goals, functions, devices, and their relationships have been found useful in the fault diagnosis and maintenance applications. The integration of control systems and on-line documentation that forms the cornerstone of our approach has been found essential in such applications.

REFERENCES

Abu-Hanna, A. & Jansweijer, W. 1994. Modelling Domain Knowledge Using Explicit Conceptualization. IEEE Expert, Vol. 9, No. 5, pp. 53 - 64.

Akao, Y. 1990. Quality Function Deployment. Integrating Customer Requirements into Product Design. Cambridge, Mass: Productivity Press, Inc. 369 p. ISBN 0-915299-41-0.

Akin, O. 1979. An Exploration of the Design Process. Design Methods and Theory, Vol. 13, No. 3 - 4, pp. 115 - 119.

Andersson, K. 1993. CANDLE - A New Tool for Conceptual Design. In: Roozenburg, N. F. M. (ed.) Proceedings of the 9th International Conference on Engineering Design (ICED'93), The Hague, August 17 - 19, 1993. Zürich: Heurista. Vol. 1, pp. 45 - 52. ISBN 3-85693-0272.

Anon. 1992. Prosessin hallinta -- Automaation tehtäväkuvaus. (Process Management -- Automation Task Description). Helsinki: Suomen Automaation Tuki Oy. 128 p. ISBN 951-96567-0-7. (In Finnish.)

Arango, G., Schoen, E. & Pettengill, R. 1993. A Process for Consolidating and Reusing Design Knowledge. In: Proceedings of the 15th Conference on Software Engineering, Baltimore, May 17 - 21, 1993. Los Alamitos, CA: IEEE Computer Society Press. Pp. 231 - 242. ISBN 0-8186-3700-5. ISSN 0270-5257.

Asimov, M. 1974. A Philosophy of Engineering Design. In: Rapp, F. (ed.) Contributions to a Philosophy of Technology. Dordrecht, Holland: D. Reidel. Pp. 150 - 157. ISBN 90-277-0433-3.

Bañares-Alcántara, R., Lababidi, H. M. S., Ballinger, G. & King, J. M. P. 1994. KBDS: A Support System for Chemical Engineering Conceptual Design. In: Proceedings of the Fourth European Symposium on Computer Aided Process Engineering (ESCAPE 4), Dublin, March 28 - 30, 1994. Rugby, UK: Inst. Chem. Engng. Pp. 419 - 426. ISBN 0-85295-321-6.

Bisantz, A. & Vicente, K. J. 1994. Making the Abstraction Hierarchy Concrete. Int. J. Human-Computer Studies, Vol. 40, pp. 83 - 117.

Bock, G. 1992. Unappreciated Writers: A Survey on Technical Writing in Germany. In: Steehouder, M., Jansen, C., van der Poort, P. & Verheijen, R. (eds.) Quality of Technical Documentation. Amsterdam: Editions Rodopi B.V. Pp. 249 - 257. ISBN 90-5183-627-9.

Bray, M. & Corsberg, D. 1994. Practical Alarm Filtering. InTech (The International Journal of Instrumentation and Control), Vol. 41, No. 2, pp. 34 - 36.

Brown, C. & Chandrasekaran, B. 1989. Design Problem Solving. Knowledge Structures and Control Strategies. London: Pitman Publishing. Pp. 32 - 35. ISBN 0-273-08766-5. ISSN 0268-7526.

Carlson, P. A. 1989. Hypertext and Intelligent Interfaces for Text. In: Barrett, E. (ed.) Society of Text. Cambridge, Mass: The MIT Press. Pp. 59 - 76. ISBN 0-262-02291-5.

Carroll, J. M. & Olson, J. R. 1988. Mental Models in Human-Computer Interaction. In: Helander, M. (ed.) Handbook of Human-Computer Interaction. Amsterdam: Elsevier Science Publisher B.V. Pp. 45 - 65. ISBN 0-444-88673-7.

Chandrasekaran, B. 1992. Design Problem Solving: A Task Analysis. In: Green, M. (ed.) Knowledge Aided Design. San Diego: Academic Press Inc. Pp. 25 - 46. ISBN 0-12-298250-9.

Chandrasekaran, B., Goel, A. & Iwasaki, Y. 1993. Functional Representation as Design Rationale. IEEE Computer, Vol. 26, No. 1, pp. 48 - 56.

Conklin, E. J. 1987. Hypertext: An Introduction and Survey. IEEE Computer, Vol. 20, No. 9, pp. 17 - 41.

Conklin, E. J. & Yakemovic, K. C. B. 1991. A Process-Oriented Approach to Design Rationale. Human-Computer Interaction, Vol. 6, No. 3&4, pp. 357 - 391.

Coombs J. H., Renear, A. H. & DeRose, S. J. 1987. Markup Systems and the Future of Scholarly Text Processing. Communications of the ACM, Vol. 30, No. 11, pp. 933 - 947.

Corsberg, D. 1987. Alarm Filtering: Practical Control Room Upgrade Using Expert Systems Concepts. InTech (The International Journal of Instrumentation and Control), Vol. 34, No. 4, pp. 39 - 42.

Crabtree, R. A, Baid, N. K. & Fox, M. S. 1993. An Analysis of Coordination Problems in Design Engineering. In: Roozenburg, N. F. M. (ed.) Proceedings of the 9th International Conference on Engineering Design (ICED'93), The Hague, August 17 - 19, 1993. Vol. 1. Zürich: Heurista. Pp. 285 - 292. ISBN 3-85693-0272.

Doyle, R. J., Sellers, S. M. & Atkinson, D. J. 1989. A Focused Context-Sensitive Approach to Monitoring. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-89), Detroit, August 20 - 25, 1989. Vol. 2. San Mateo, CA: Morgan-Kaufmann. Pp. 1231 - 1237.

Duncan, K. D. & Praetorius, N. 1992. Flow Displays Representing Complex Plant for Diagnosis and Process Control. Reliability Engineering and System Safety, Vol. 36, pp. 239 - 244.

Easter, J. R. 1992. Engineering Human Factors into the Westinghouse Advanced Control Room. Nuclear Engineering International, Vol. 22, No. 4, pp. 35 - 38.

Fischer, G., Lemke, A. C. & McCall, R. 1991. Making Argumentation Serve Design. Human-Computer Interaction. Vol. 6, No. 3&4, pp. 393 - 419.

Freeman, P. & Newell, A. 1971. A Model for Functional Reasoning in Design. In: Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI-71), London, September 1 - 3, 1971. London: The British Computer Society. Pp. 621 - 633.

French, M., Chaplin, R.V. & Langdon, P. M. 1993. A Creativity Aid for Designers. In: Roozenburg, N. F. M. (ed.) Proceedings of the 9th International Conference on Engineering Design (ICED'93), The Hague, August 17 - 19, 1993. Vol. 1. Zürich: Heurista. Pp. 53 - 59. ISBN 3-85693-0272.

Green, M. 1992. Conceptions and Misconceptions of Knowledge Aided Design. In: Green, M. (ed.) Knowledge Aided Design. San Diego: Academic Press Inc. Pp. 1 - 24. ISBN 0-12-298250-9.

Gruber, T. R. & Russell, D. M. 1990. Design Knowledge and Design Rationale: A Framework for Representation, Capture, and Use. Technical Report KSL 90-45. Knowledge Systems Laboratory, Computer Science Department, Stanford University. Stanford, CA: Stanford University. 40 p.

Gruber, T. R. & Russell, D. M. 1992. Beyond the Record and Replay Paradigm for Design Rationale Support. In: Working Notes of the AAAI '92 Workshop on Design Rationale Capture and Use, San Jose, CA, July 15 1992. American Association for Artificial Intelligence. Pp. 111 - 118.

Halasz, F. 1988. Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems. Communications of the ACM, Vol. 31 No. 7, pp. 836 - 852.

Han, C., Douglas, J. M. & Stephanopoulos, G. 1995. Agent-Based Approach to a Design Support System for the Synthesis of Continuous Chemical Processes. Computers Chem. Engng, Vol. 19, Suppl., pp. S63 - S69.

Harmon, P. 1990. DESIGN++: An Expert System for Automating the Engineering Design Process. Expert System Strategies, Vol. 6, No. 6, pp. 9 - 12.

Hollnagel, E. 1991. GRADIENT Technical Overview. Birkerød, Denmark: Computer Resources International A/S. 75 p.

Holsapple, C. W. & Whinston, A. B. 1992. Decision Support Systems. In: Salvendy, G. (ed.) Handbook of Industrial Engineering. New York: John Wiley & Sons. Pp. 109 - 141. ISBN 0-471-50276-6.

Huuskonen, P. 1992. Knowledge-Based Support for the Process Operator. Oulu, Finland: University of Oulu, Department of Electrical Engineering. 84 p. + app. 9 p.

Huuskonen, P. & Jaako, J. 1992. Analyysi Haapaveden turvevoimalaitoksen ongelmakohdista ja kehittämistarpeista. Internal project report (not published). Oulu, Finland: VTT Electronics & University of Oulu. 24 p. (In Finnish.)

IGES. 1988. Initial Graphic Exchange Specification IGES Version 4.0. Washington DC: US Department of Commerce, National Bureau of Standards. 515 p.

Iivonen, H. & Riitahuhta, A. 1994. Case-Based Reasoning in Conceptual Design. In: Knudsen, J.K.H., MacConail, P.A. & Bastos, J. (eds.) Sharing CIM Solutions. Proceedings of the 10th CIM-Europe Annual Conference, Copenhagen, Denmark, October 5 - 7, 1994. Amsterdam: IOS Press. Pp. 307 - 314. ISBN 90-5199-194-0.

ISO 8613-1. 1989. Information Processing - Text and Office Systems. Open Document Architecture (ODA) and Interchange Format - Part 1: Introduction and General Principles. Geneve: International Organization for Standardization. 38 p.

ISO 8879. 1986. Information Processing - Text and Office Systems. Standard Generalized Markup Language (SGML). Geneve: International Organization for Standardization. 155 p.

ISO 9735. 1987. Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT). Geneve: International Organization for Standardization. 19 p.

ISO 10303-1. 1994. Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 1: Overview and Fundamental Principles. Geneve: International Organization for Standardization. 17 p.

ISO 10303-11. 1994. Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 11: Description Methods: The EXPRESS Language Reference Manual. Geneve: International Organization for Standardization. 208 p.

ISO/IEC 10744. 1992. Information Technology - Hypermedia/Time-Based Structuring Language (HyTime). Geneve: International Organization for Standardization. 125 p.

Johannsen, G. & Alty, J. L. 1991. Knowledge Engineering for Industrial Expert Systems. Automatica, Vol. 27, No. 1, pp. 97 - 114.

Jonassen, D. H. 1993. Effects of Semantically Structured Hypertext Knowledge Bases on User's Knowledge Structures. In: McKnight, C., Dillon, A. & Richardson, J. (eds.) Hypertext, A Psychological Perspective. Chichester, England: Ellis Horwood Limited. Pp. 153 - 168. ISBN 0-13-441650-3.

Kaarela, K., Oksanen, J. & Lukkari, K. 1992. Laitosprojektin toteuttamisen aikana ilmenevät tiedonsiirto-ongelmat. Internal project report (not published). Oulu, Finland: VTT Electronics & University of Oulu. 36 p. (In Finnish.)

Kannapan, S. M. & Marshek, K. M. 1992. Engineering Design Methodologies: A New Perspective. In: Kusiak, A. (ed.) Intelligent Design and Manufacturing. New York: John Wiley & Sons. Pp. 3 - 38. ISBN 0-471-53473-0.

Katajamäki, M. 1991. Knowledge-Based CAD. Expert Systems with Applications, Vol. 3, pp. 277 - 287.

KEE. 1986. KEE Software Development System User's Manual. Mountain View, CA: IntelliCorp, Inc.

Keuneke, A. 1991. Device Representation: The Significance of Functional Knowledge. IEEE Expert, Vol. 6, No. 2, pp. 22 - 25.

Kim, W. 1991. Object-Oriented Database Systems: Strengths and Weaknesses. Journal of Object-Oriented Programming, Vol. 4, No. 4, pp. 21 - 29.

Korhonen, R. 1991. Framework for Improving Quality and Efficiency in Automation Design. Tampere, Finland: Tampere University of Technology, Publications 85. 113 p. ISBN 951-721-761-7. ISSN 0356-4940.

Kuipers, B. J. 1986. Qualitative Simulation. Artificial Intelligence, Vol. 29, pp. 289 - 338.

Larsson, J.-E. (1992). Knowledge-based Methods for Control Systems. Doctoral thesis. Lund, Sweden: Lund Institute of Technology. 236 p. ISSN 0280-5316.

Lee, J. & Lai, K.-Y. 1991. What's in Design Rationale? Human-Computer Interaction, Vol. 6, No. 3&4, pp. 251 - 280.

Leiviskä, K., Kaarela, K. & Oksanen, J. 1994. TIESU-projektin loppuraportti. Oulu, Finland: University of Oulu. 136 p. + app. 27 p. (In Finnish.)

Leppälä, K. 1995. Inside a Contract Research Laboratory. A Study of Concepts, Methods, and Performance. Espoo, Finland: VTT. 250 p. + app. 22 p. VTT Publications 227. ISBN 951-38-4765-9. ISSN 1235-0621.

Lind, M. 1982. Multilevel Flow Modelling of Process Plant for Diagnosis and Control. Risø-M-2357. Denmark: Risø National Laboratory.

Lind, M. 1990. Representing Goals and Functions of Complex Systems - An Introduction to Multilevel Flow Modeling. Institute of Automatic Control Systems, Technical University of Denmark, Report No. 90-D-38. 86 p. ISBN 87-87950-52-9.

Lind, M. 1991. Representations and Abstractions for Interface Design Using Multilevel Flow Modelling. In: Human-Computer Interaction and Complex Systems. London: Academic Press Limited. Pp. 223 - 243. ISBN 0-12-742660-0.

Luce, R. D. & Raiffa, H. 1957. Games and Decisions. New York: John Wiley & Sons. 509 p.

Magnusson, J. S. 1994. Defining Information Objects against a Product Data Model. In: The SGML Horizon -- A Standard in Practice. Proceedings of the SGML Europe '94 Conference, Montreux, Switzerland, May 16 - 19, 1994. Alexandria, VA: Graphic Communications Association. Pp. 253 - 258.

Malmqvist, J. 1995. Improved Function-Means Trees by Inclusion of Design History Information. In: Hubka, V. *et al.* (eds.) Proceedings of the 10th International Conference on Engineering Design (ICED'95), Praha, August 22 - 24, 1995. Vol. 4. Zürich: Heurista. Pp. 1449 - 1454. ISBN 3-85693-028.0.

Marsh, R. & Wallace, K. 1995. Integrity of Design Information. In: Hubka, V. *et al.* (eds.) Proceedings of the 10th International Conference on Engineering Design (ICED'95), Praha, August 22 - 24, 1995. Vol. 4. Zürich: Heurista. Pp. 1449 - 1454. ISBN 3-85693-028.0.

McDermott, J. 1982. R1: A Rule-Based Configurer of Computer Systems. Artificial Intelligence, Vol. 19, No. 1, pp. 39 - 88.

Mostow, J. 1985. Toward Better Models of the Design Process. AI Magazine, Vol. 6, pp. 44 - 57.

Newell, A. & Simon, H. A. 1972. Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall. 920 p. ISBN 0-13-445403-0.

Nichols, K. 1990. Getting Engineering Changes under Control. Journal of Engineering Design, Vol. 1, No. 1, pp. 5 - 15.

Norman, D. A. 1983. Some Observations on Mental Models. In: Gentner, D. & Stevens, A. (eds.) Mental Models. Hillsdale, NJ: Lawrence Erlbaum Associates. Pp. 7 - 14. ISBN 0-89859-242-9.

Paasiala, P., Aaltonen, A., Tanskanen, K, & Riitahuhta, A. 1994. Managing Multidisciplinary Design Projects. In: Proceedings of the International Conference on Machine Automation, Mechatronics Spells Profitability, Tampere, Finland, February 15 - 18, 1994. Vol. 2. Tampere: Tampere University of Technology. Pp. 659 - 670. ISBN 951-722-107-X.

Padalkar, S., Karsai, G., Biegl, C. & Sztipanovits, J. 1991. Real-Time Fault Diagnostics. IEEE Expert, Vol. 6, No. 3, pp. 75 - 85.

Pahl, G. & Beitz, W. 1988. Engineering Design. A Systematic Approach. 2nd edition. London: The Design Council. 450 p. ISBN 0-85072-239-X.

Peach, D. 1992. The L3 Gas Systems Operators Manual. Geneve: CERN. Not published.

Pernu, H. 1992. Personal Communication.

Prins, J. A. A. & Olthoff, M. M. 1993. Morphology as a Documentation Framework. In: Roozenburg, N. F. M. (ed.) Proceedings of the 9th International Conference on Engineering Design (ICED'93), The Hague, August 17 - 19, 1993. Vol. 1. Zürich: Heurista. Pp. 158 - 164. ISBN 3-85693-0272.

Rada, R. 1992. Converting a Textbook to Hypertext. ACM Transactions on Information Systems, Vol. 10, No. 3, pp. 294 - 315.

Rasmussen, J. 1985. The Role of Hierarchical Knowledge Representation in Decisionmaking and System Management. IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC - 15, No. 2, pp. 234 - 243.

Rasmussen, J. 1986. Information Processing and Human-Machine Interaction. Amsterdam: North-Holland. 215 p. ISBN 0-444-00987-6.

Reddy, Y. V., Wood, R. T. & Cleetus, K. J. 1992. The DARPA Initiative in Concurrent Engineering. Concurrent Engineering Research in Review. Vol. 1, Winter 1991/1992, pp. 2 - 10.

Riitahuhta, A. 1988. Enhancement of the Boiler Design Process by the Use of Expert System Technology. Acta Polytechnica Scandinavica, Mechanical Engineering Series No. 92. Helsinki: Finnish Academy of Technology. 122 p. ISBN-951-666-272-2. ISSN 001-687X.

Rytoft, C., Johansson, B., Bladh, K. & Hoggard, N. 1990. Aspects of Future Process Control Systems. Control Engineering, Vol. 37, No. 9, pp. 135 - 138.

Saaty, T. 1983. Priority Setting in Complex Problems. IEEE Transactions on Engineering Management, Vol. EM - 30, No. 3, pp. 140 - 155.

Sassen, J. A. M. (1993). Design Issues of Human Operator Support Systems. Doctoral thesis. Delft, The Netherlands: Delft University of Technology. 226 p. ISBN 90-370-0090-8.

Schumann-Hindenberg, U. 1993. The EDI Experience in Airbus Development and Manufacturing. In: Proceedings of the 4th Annual International Conference on Computer-aided Acquisition and Logistic Support (CALS) and Information Management in Europe. Berlin, Germany, September 22 - 24, 1993. Falls Church, VA: World Computer Graphics Foundation. Pp. 39 - 48.

Seppänen, V. 1990. Acquisition and Reuse of Knowledge to Design Embedded Software. Espoo, Finland: VTT. 218 p. + app. 10 p. VTT Publications 66. ISBN 951-3579-0. ISSN 0358-5069.

Sheldon, D. F., Perks, R., Jackson, M., Miles, B. L. & Holland, J. 1990. Designing for Whole Life Costs at the Concept Stage. Journal of Engineering Design, Vol. 1, No. 2, pp. 131 - 145.

Shneiderman, B. 1987. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, Massachusetts: Addison-Wesley. Pp. 179 - 223. ISBN 0-201-16505-8.

Simon, H. A. 1981. Sciences of the Artificial. Second Edition. Cambridge, Mass: MIT Press. 247 p. ISBN 0-262-19193-8.

Smith, J. 1992. SGML and Related Standards. Document Description and Processing Languages. Chichester, England: Ellis Horwood Limited. 151 p. ISBN 0-13-806506-3.

Snaprud, M. & Kaindl, H. 1992. Knowledge Acquisition Using Hypertext. Expert Systems With Applications, Vol. 5, No. 3/4, pp. 369 - 375.

Sriram, D., Cheong, K. & Kumar, M. L. 1992. Engineering Design Cycle. In: Green, M. (ed.) Knowledge Aided Design. San Diego: Academic Press Inc. Pp. 117 - 156. ISBN 0-12-298250-9.

Stassen, H. G., Johannsen, G. & Moray, N. 1988. Internal Representation, Internal Model, Human Performance Model and Mental Workload. In: Preprints of the IFAC/IFIP/IEA/IFORS Conference on Man-Machine Systems, Oulu, Finland, June 14 - 16, 1988. Vol. 1. Finnish Society of Automatic Control and International Federation of Automatic Control. Pp. 252 - 261.

Steehouder, M. 1994. The Quality of Access: Helping Users Find Information in Documentation. In: Steehouder, M., Jansen, C., van der Poort, P. & Verheijen, R. (eds.) Quality of Technical Documentation. Amsterdam: Editions Rodopi B.V. Pp. 131 - 143. ISBN 90-5183-627-9.

Stephanopoulos, G. 1987. Intelligent Systems for Process Operations Overview. In: Proceedings of the First International Conference on Foundations of Computer Aided Process Operations, Park City, Utah, July 5 - 10, 1987. Pp. 503 - 555.

Stephanopoulos, G., Johnston, J., Kriticos, T., Lakshmanan, R., Mavrovouniotis, M. & Siletti, C. 1987. Design-Kit: An Object-Oriented Environment for Process Engineering. Comput. Chem. Engng, Vol. 11, No. 6, pp. 655 - 674.

Stephanopoulos, G., Henning, G. & Leone, H. 1990a. MODEL.LA. A Modeling Language for Process Engineering - I. The Formal Framework. Comput. Chem. Engng, Vol. 14, No. 8, pp. 813 - 846.

Stephanopoulos, G., Henning, G. & Leone, H. 1990b. MODEL.LA. A Modeling Language for Process Engineering - II. Multifaceted Modeling of Processing Systems. Comput. Chem. Engng, Vol. 14, No. 8, pp. 847 - 869.

Suh, N. P. 1990. The Principles of Design. New York: Oxford University Press. 401 p.

Suitiala, R. 1993. Work-oriented Development of Interactive Software Tools. Understanding the Work of Software Maintainers and Making an Interactive Software Tool for Them. Espoo, Finland: VTT. 176 p. + app. 10 p. VTT Publications 139. ISBN 951-38-4257-9. ISSN 1235-0621.

Takalo, J. 1995. Enhancing Access to Technical On-line Documentation Through Information Modelling and Retrieval Techniques. Master's Thesis. Oulu, Finland: University of Oulu, Department of Information Processing Science. 81 p.

Taleb-Bendiab, A. 1993. ConceptDesigner: A Knowledge-Based System for Conceptual Engineering Design. In: Roozenburg, N. F. M. (ed.) Proceedings of the 9th International Conference on Engineering Design (ICED'93), The Hague, August 17 - 19, 1993. Vol. 3. Zürich: Heurista. Pp. 1303 - 1310. ISBN 3-85693-0272.

Teeuw, W. B., Liefting, J. R., Demkes, R. H. J. & Houtsma, M. A. W. 1995. Product Data Interchange in Practice: Experiences, Problems, and Opportunities. In: Paul, A. J. & Sobolewski, M. (eds.) Proceedings of the Concurrent Engineering 1995 Conference, McLean, Virginia, August 23 - 25, 1995. Johnstown, PA: Concurrent Technologies Corporation. Pp. 1 - 11. ISBN 0-9642449-1-8.

Tommila, T., Toola, A. & Viitamäki, P. 1990. Prosessin mallintaminen ohjausjärjestelmän suunnittelun lähtökohtana. Espoo, Finland: VTT. 71 p. VTT Research Notes 1099. ISBN 951-38-3669-X. ISSN 0358-5085. (In Finnish.)

Tommila, T. & Viitamäki, P. 1991. Vaatimusmäärittely prosessi-automaatiossa. Lähestymistapoja esi- ja perussuunnitteluun. Espoo, Finland: VTT. 98 p. + app. 20 p. VTT Research Notes 1292. ISBN 951-38-4025-5. ISSN 0358-5085. (In Finnish.)

Toola, A. 1992. Safety Analysis in Conceptual Design of Process Control. Espoo, Finland: VTT. 100 p. VTT Publications 117. ISBN 951-38-4235-5. ISSN 1235-0621.

Tyrväinen, P. 1994. Domain Modelling for Technical Documentation Retrieval. Acta Polytechnica Scandinavica, Mathematics and Computing in Engineering Series No. 64. Helsinki: Finnish Academy of Technology. 171 p. ISBN 951-666-406-7. ISSN 1237-2404.

Van de Ree, R. V. 1994. SCWERE, Supervisory Control Systems. Doctoral thesis. Delft, The Netherlands: Delft University of Technology. 214 p. ISBN 90-900-7028-1.

Vanhainen, J. & Paunonen, H. 1994. Experiences and Future Trends on Developments of Process Control Tools on a Greenfield Coated Woodfree Art Printing Paper Mill. In: Preprints of The Conference on Control Systems in the Pulp and Paper Industry, Stockholm, Sweden, May 31 - June 2, 1994. Swedish Pulp and Paper Research Institute and the Swedish Association of Pulp and Paper Engineers. 7 p.

VDI 2221. 1987. VDI Guidelines 2221. Systematic Approach to the Design of Technical Systems and Products. Düsseldorf: VDI-Verlag. 34 p.

Vicente, K. J. 1992a. Multilevel Interfaces for Power Plant Control Rooms I: An Integrative Review. Nuclear Safety, Vol. 33, No. 3, pp. 381 - 397.

Vicente, K. J. 1992b. Multilevel Interfaces for Power Plant Control Rooms II: A Preliminary Design Space. Nuclear Safety, Vol. 33, No. 4, pp. 543 - 548.

Väisänen, A. 1994. Järjestelmällinen tavoiteasetanta prosessilaitoksen suunnittelun lähtökohtana. Oulu, Finland: University of Oulu. 78 p. (In Finnish.)

Walker, J. H. 1988. Supporting Document Development with Concordia. IEEE Computer, Vol. 21, No. 1, pp. 48 - 59.

Walker, J. H. 1989. Authoring Tools for Complex Documentation Sets. In: Barrett, E. (ed.) Society of Text. Cambridge, Mass: The MIT Press. Pp. 132 - 147. ISBN 0-262-02291-5.

Warman, E. A. 1990. Object-Oriented Programming and CAD. Journal of Engineering Design, Vol. 1, No. 1, pp. 37 - 46.

Weich, R. E. 1992. Utilizing CALS Standards to Achieve Effective Information Management. CALS Journal, Vol. 1, No. 4 (Winter 1992), pp. 47 - 52.

Wielinga, B. J. & Breuker, J. A. 1986. Models of Expertise. In: Proceedings of the 7th European Conference on Artificial Intelligence, Brighton, UK, July 21 - 25, 1986. Vol. 1, pp. 306 - 318.

Zeid, I., Bardasz, T., Bhaskara, S. & Gao, Y. 1993. Design Plan Systems for Case-based Mechanical Design: A Survey and Open Issues. In: Roozenburg, N. F. M. (ed.) Proceedings of the 9th International Conference on Engineering Design (ICED'93), The Hague, August 17 - 19, 1993. Vol. 3. Zürich: Heurista. Pp. 1369 - 1376. ISBN 3-85693-0272.

Appendices of this publication are not included in the PDF version.
Please order the printed version to get the complete publication
(http://www.inf.vtt.fi/pdf/publications/1996)