Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers and Chemical Engineering 29 (2004) 65–82

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Workflow and information centered support of design processes—the IMPROVE perspective

Wolfgang Marquardt[a,*], Manfred Nagl[b]

[a] *Lehrstuhl für Prozesstechnik, RWTH Aachen University, D-52056 Aachen, Germany*
[b] *Lehrstuhl für Informatik III (Software Engineering), RWTH Aachen University, D-52056 Aachen, Germany*

Received 16 June 2004; accepted 20 July 2004
Available online 22 September 2004

Abstract

Design process excellence is considered a major differentiating factor between competing enterprises since it determines the constraints within which plant operation and supply chain management are confined. The most important prerequisite to establish such design process excellence is a proper management of all the design process activities and the associated information. Starting from an analysis of the characteristics of chemical engineering design processes, some important open research issues are identified. They include the development of an integrated information model of the design process, a number of innovative functionalities to support collaborative design, and the a-posteriori integration of existing software tools to an integrated design support environment. Some of the results obtained and experiences gained in the last years in the collaborative research center IMPROVE at RWTH Aachen University are presented.

Keywords: Computer-aided design; Information modeling; Software engineering; Tool integration; Business processes; Workflow; Work process

1. Manufacturing and design in the 21st century

The markets and hence the requirements on manufacturing in the process industries have been changing tremendously in the last decades. Growing market volume and limited, often largely local competition have been dominating manufacturing in the seventies and eighties. Today, the process industry is facing largely saturated markets in many geographical regions of the world. Internet technology has been successfully used in e-commerce solutions to achieve almost complete market transparency. Engineering and manufacturing skills are available globally. At the same time, transportation cost have been decreasing significantly. Hence, every manufacturer is facing truly global competition. Economic success is only possible, if new ideas can be quickly transformed into new marketable products or if the production cost of established products can be diminished substantially to counteract decreasing profit margins. Product innovation, process design as well as manufacturing processes have to be continuously improved to reduce time to market of a new product, to minimize manufacturing cost and to establish a high level of customer satisfaction by offering the right product at the right time and location.

1.1. Two business processes

The *value chain* in any manufacturing oriented industry comprises *two major business processes*—manufacturing and design—which are highly interrelated (Schuler, 1998). These business processes are constrained by the socio-economic environment, in particular, the market, the legislation and the available process technologies (Fig. 1).

Value creation happens in the *manufacturing process* (Fig. 1, top), which is part of a supply chain including warehouses, distribution and procurement in addition to the production plants. Excellence in manufacturing is not possible without explicit consideration of the constraints

---

[*] Corresponding author. Tel.: +49 241 809 6712; fax: +49 241 809 3226.
E-mail address: marquardt@lpt.rwth-aachen.de (W. Marquardt).

0098-1354/$ – see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2004.07.018

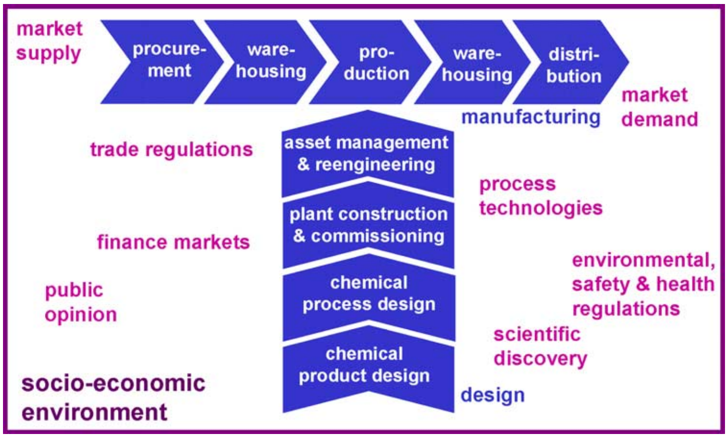

Fig. 1. The two major business processes in the process industries: manufacturing and design.

and potentials resulting from interaction between the plant and the supply chain it is embedded into. The influencing factors from the supply chain on plant operation have to be exploited rather than rejected by model-based plant management considering all the manufacturing business processes across the whole supply chain (Backx, Bosgra, & Marquardt, 1998). The changing business environment can be addressed on a short time scale by adapting supply chain management and plant operation strategies for a fixed design.

The manufacturing process is largely determined by the second business process, the *design process*, which comprises all the activities related to the design of a new product and the associated production plant including the process and control equipment as well as all operation and management support systems (Fig. 1, bottom). This business process starts with an idea on a new product and subsequent product design. Conceptual design, basic and detail engineering of the production plant are the major activities which follow, before the plant can be built and commissioned. Excellence in design requires consideration of the complete design lifecycle (Marquardt, Wedel, & Bayer, 2000). In particular, the interactions between different design lifecycle phases focusing on different aspects such as the chemical product, the process concept, equipment design, plant layout, or control structure selection need to be exploited. Only an integrated consideration facilitates the realization of synergies and the achievement of the true economical potential. The plant and the supply chain have to be continuously reengineered during their lifetime in order to adjust manufacturing to major changes in the market conditions and legislation, to adopt new process technologies and to profit from accumulated operational experience. Asset management is increasingly established to make best use of existing facilities and to support preventive maintenance and benchmarking activities. Plant reengineering is only possible on a longer time scale as compared to an adaptation of the manufacturing process for a given plant and supply chain design.

1.2. Value creation

The economic performance of an enterprise heavily relies on the quality of the products of these two business processes. Typically, the *major focus* is on the product of the *manufacturing process*, namely the chemicals, which are sold to customers and therefore are considered to generate the revenue to the enterprise. The manufacturing process and its associated supply chain, however, are considered as the cost generators. Profit can be increased on the short time scale with limited investment, if the manufacturing cost can be reduced by optimized strategies for plant operation and supply chain management. It is therefore not surprising, that the current industrial focus is on the reduction of manufacturing cost in order to counteract decreasing profit margins.

This strategy does not seem to be sustainable in the long run, since cost reduction by means of better supply chain management and plant operation using existing assets is largely independent of a certain product portfolio and does not contribute to a fundamental understanding of the processing technology and its impact on chemical product characteristics. The employed operations research techniques apply to many businesses and may therefore evolve in a technological commodity. After a transition period during which these technologies are adopted, the differentiation between competitors with respect to manufacturing excellence vanishes.

Hence, at least at this point in time, there is no adequate appreciation of the contribution of design excellence to the overall success of an enterprise. It is the design *process* which *determines* the design of a *manufacturing* plant. This design is largely responsible for the achievable quality of the chemical product and for the order of magnitude of the production cost. The design also constrains the operational envelope and hence the flexibility to react to changing market conditions. Ideally, an integrated consideration of plant and supply chain design on the one and supply chain and plant management on the other hand should be addressed (Backx et al., 1998). However, such an approach would have to generalize and extend the problem of an integrated design and control of a single plant, which itself has not yet been solved satisfactorily.

We hypothesize that *design excellence* is becoming a *major differentiating asset* in the future which, to a large extent, will decide on the economical success of an enterprise. Of course, for this hypothesis to be true, design has to be interpreted in a broader than the traditional sense. In particular, not only the process flowsheet and equipment, but also the operation support system as well as the chemical product itself have to be considered part of the integrated design business process. The quality of the design process is strongly depending on the available knowledge about the chemical process and products and its long-term management. We claim that design excellence in addition requires a profound understanding of the integrated design process itself. Design excellence has to be based on a systematic acquisition, management and reuse of such knowledge. It forms the basis for identifying shortcomings in available knowledge and established work processes. It is therefore a prerequisite for design process reengineering to establish better process design practices. Clearly, information technology support and model-based design process integration are key enablers. Together with a deep understanding of the design process, they are the major pre-requisites for the implementation of a suitable software environment to support the activities in the design process in an integrated manner.

This perspective of the design process is not entirely new. It has been stressed in a similar way by a few other research groups, most notably those at Carnegie Mellon University (Subrahmanian, Westerberg, & Podnar, 1991; Konda, Monarch, Sargent, & Subrahmanian, 1992; Finger, Konda, & Subrahmanian, 1995; Westerberg, Subrahmanian, Reich, Konda, & the n-dim group, 1997; Davis et al., 2001) and at the University of Edinburgh (Banares-Alcantara, 1991, 1995; Banares-Alcantara & Lababidi, 1995; Costello et al., 1996).

1.3. Overview on the paper

In the following we *focus* in this paper only on a *part of the design process*, namely to the early phases of the chemical process design lifecycle, *the conceptual design and front-end engineering*, for pragmatic reasons to avoid excessive complexity. Further, we believe that many of our findings will carry over to the more complicated integrated design and manufacturing problem. Certainly, this problem is much more complex and presents additional requirements and challenges for information technology support. However, the key issues in the chemical process design process as discussed in Section 2 are also relevant for the integrated design and manufacturing problem. Current chemical process design shares a lot of commonalities with the design practice in other industrial domains. The assessment of the chemical process design process in the next section holds almost equally well for other engineering design processes. In that sense, our findings seem to be relevant not only for chemical engineering design and manufacturing. On the basis of the assessment in Section 2, key research questions are formulated and the interdisciplinary research center IMPROVE is introduced subsequently in Section 3. Sections 4 to 6 present major results of the research work of IMPROVE and the experience made in the areas of information modeling, design environment architecture and tools supporting collaborative work processes.

2. The character of chemical process design processes

The plant lifecycle can be subdivided into *six major phases* which comprise conceptual design, basic engineering, detail engineering, construction and commissioning as well as asset management, maintenance and continuous reengineering (Fig. 1). Conceptual design and front end engineering (the early phase of basic engineering) constitute those parts of the lifecycle with the most significant impact on the lifecycle cost. In this early design phase, almost all of the conceptual decisions on the raw materials and the reactions, the process, the equipment, the plant and even on control and plant operation are taken. Though, only a small fraction of the total investment cost of the plant is spent in these early lifecycle phases, the consequences on the total cost of ownership of the plant are most significant. The results of this early lifecycle phase form the basis for the subsequent refinement during basic and detail engineering. These *early phases of the design lifecycle* constitute the *focus of this contribution* due to their significance for the whole plant lifecycle.

2.1. Status of industrial design processes

The design process is carried out by a *team* of multidisciplinary experts from different organizational units within the same or different companies. The team is formed to carry out a dedicated *project*, it is directed by a project manager. Usually, a number of consultants are contributing to the design activities in addition to the team members. All team members are typically part of more than one team at the same time. Often, the team operates at different, geographically distributed sites. The duration of a single project may range from weeks to years with varying levels of activity at a certain point in time. Hence, the team and the status and assignments of its members may change with time in particular in case of long project duration. Inevitably, there is no common understanding about the design problem in the beginning of the project. Such a common understanding, called shared memory by Konda et al. (1992), has to evolve during collaborative work.

The *design process* constitutes of all the related activities carried out by the team members while they work on the design problem (Westerberg et al., 1997). This multidisciplinary process shows an immense complexity. It has to deal with the culture and paradigms from different domains. Complicated multi-objective *decision making* processes *under uncertainty* are incorporated in the design. They rely on the typically incomplete information produced in the current and previous design activities. In particular, conceptual design processes show a *high degree of creativity*, they are of an inventive nature and do not just apply existing solutions.

Creative conceptual design processes are hardly predictable and can therefore only be pre-planned on a coarse-grained level. A work process definition—even coarse-grained—is mandatory to establish simultaneous and concurrent engineering to reduce the total time spent on a design. The lack of precise planning on a medium-grained level inevitably results in *highly dynamic* work processes. They show *branches* to deal with the assessment of alternatives and to allow for a simultaneous work on only loosely related subtasks. *Iterations* occur to deal with the necessary *revision of previous decisions and solutions*. In the first place, they are due to inevitable uncertainties during decision making because of lacking or incomplete information. While the design process is carried out, this uncertainty can be continuously reduced because of the additional information becoming available, it is either collected from various available but not yet exploited resources or it is generated while the design process progresses. Additional information always gives rise to new insight to either address a problem which has not yet been recognized, to exploit an identified potential for improving an existing solution, or to even evolve the design requirements. A strict definition of the work process in conceptual design (as accomplished in many administrative business processes (Fisher, 2000)) is not only impossible but also highly undesirable. It would largely constrain the creativity of the designer with obviously undesirable consequences for the quality of the resulting design.

The team of experts typically uses a *multitude of resources* in the various phases of the design process. For example, web-based *text retrieval and browsing systems* are used to search the scientific and patent literature or internal archives for information on the materials or processing technologies. *Lab-scale or pilot-scale experiments* allow the investigation of specific questions related to physical properties, kinetics, scale-up of equipment or the accumulation of impurities in recycles and their impact on the process behavior. All kinds of *software tools* with diverse and often overlapping functionality have been increasingly used in the last two decades to support different design activities.

First, there are *standard software tools* such as word processing, spreadsheet or groupware systems, which are completely independent of a specific application domain and hence are established in all industrial segments. Second, there are *domain specific tools* which support specific chemical process design activities. Such tools include, for example, block or equation oriented process modeling environments, equipment rating and design or cost estimation software. Often, different tools are in use for the same or similar tasks within a typically globally acting enterprise. This diversity and heterogeneity of software tools may even show up in a geographically distributed design team. Often, these tools rely on some *mathematical model* of the chemical process to perform a synthesis or analysis step in a model-based fashion. These models are of differing coverage and rigor, but contain a lot of process knowledge in a formalized and structured manner.

In the course of the design process, a *complex configuration of different types of information* is created. This information appears in multiple ways. There are, for example, standardized documents including equipment specification sheets or design reports, informal texts like e-mail or telephone notes, or input or output files of certain software tools containing problem specifications or result summaries in a formal syntax. More recently, audio and video clips may be included in addition. This information is typically held in a decentralized manner in the local data stores of the individual software tools, in document management systems or in project databases. Typically, the relationship between the various information units is not explicitly held in the data stores. Information is exchanged in the design team by means of *documents*, which aggregate selected data relevant to a certain work process context.

Though a large amount of information is created and archived in some data store during the design process, there is typically no complete *documentation* of all the alternatives considered during the design. However, a full documentation of the final conceptual design has to be compiled from the information created during the design process. Typically, this documentation is handed over to an engineering contractor and to the operating company. The contractor employs this design documentation to continue the design process during basic and detail engineering, whereas the operating company uses the conceptual design package to prepare maintenance and asset management procedures.

2.2. Analysis of current design practice and supporting software tools

The analysis of current *design practice* reveals a number of *weaknesses* which have to be overcome to successfully establish design process excellence. The most import issues are the following:

There is no common understanding and terminology related to the design process and its results.

- Creative design processes are not properly understood. There is no systematic reengineering and continuous improvement process in place.
- Design processes and their results are not sufficiently well documented. This lack of documentation prevents the tracing (i) of ideas which have not been pursued further for one or the other reason, (ii) of all the alternatives studied, (iii) of the decision making processes and (iv) of the design rationale.
- Reuse of previous solutions and experiences at a later time in the same or similar design projects is not supported.
- The creation of knowledge through learning from previous experience is not systematically supported by information technologies.

- There is no systematic evolution of requirements and no assessment of design objectives with respect to the requirements.
- A coherent configuration of all the design data in the context of the work process is not available. Time spent for searching and interpreting information on a certain design in the course of the plant lifecycle is enormous. Often, it is less effort to repeat a task. There is no systematic management of conflicts between design information or change propagation mechanism between design documents.
- There is no systematic integration of design methodologies based on mathematical models of the chemical processes with the overall design work process.

In addition to these work process oriented deficiencies, there are also serious *shortcomings* with respect to the *software tools* supporting the design process. Some important considerations are the following:

- Tools are determining the design practice significantly, because there has been largely a technology push and not a market pull situation in the past. Tool functionality has been constrained by technology, often preventing a proper tailoring to the requirements of the design process. Usually, the tools are providing support functionality for only a part of a design task or a set of design tasks.
- There is a limited integration between tools largely focusing on those of a single vendor or its collaborating partners. The integration of legacy tools into such an environment or the integration of the software infrastructure of a company is costly.
- The heterogeneity of the software environment impedes cooperation between organizations.
- Design data are represented differently in the various tools. There are not only technical, but also syntactic and semantic mismatches which prevent integration.
- There is a lack of managing relations between data and documents produced by different tools in different design activities.
- Project management and administration software is not at all integrated with engineering design support software. Hence, proper planning and controlling of creative design processes is difficult.
- Tool integration is largely accomplished by data transfer or data integration via a central data store, neglecting the requirements of the work processes.
- Communication in the design team is only supported by generic tools like e-mail, video conferences, etc., which are not integrated with engineering design tools.
- The management of creative design processes is not supported by means of domain specific tools.

These two lists clearly reveal *high correlation* of the *work processes* itself and the *supporting software tools*. Both have to be synergistically improved and tailored to reflect the needs of the design process in a holistic manner. We believe that a work process oriented view on design and the required information technology support is a major prerequisite to achieve design process excellence. In addition, a further development of model-based chemical process design methodologies, algorithms and tools has to take place.

3. The collaborative research center improve

About 6 years ago, the interdisciplinary collaborative research center (CRC) 476 (IMPROVE) has been established at RWTH Aachen University. It is funded by Deutsche Forschungs*g*emeinschaft (DFG, the German Science Foundation) to address some of the issues identified in the last section. Computer scientists and engineers from six disciplines are collaborating with substantial financial and human resources in this long-term research effort. The focus is on new concepts and software engineering solutions to support collaborative engineering design processes (Nagl & Westfechtel, 1999). Research is concentrated on the early phases of the design lifecycle due to its significant impact on total cost of ownership and due to the challenges resulting from the creative and highly dynamic nature of the work process.

A scenario-based research approach has been used in IMPROVE in order to identify the requirements based on a concrete chemical process design case study. The selected scenario comprises the conceptual design of a polymerization process for the production of polyamide-6 from caprolactam (Eggersmann, Hackenberg, Marquardt, & Cameron, 2002a; Eggersmann, Schneider, & Marquardt, 2002b). This process is well documented in the literature and of significant industrial relevance. The polymerization domain has been chosen because there are much less mature design support tools as compared to petrochemical processes. Therefore, tool integration and work process support are of considerable interest in both, the end user as well as the software vendor industries.

The process consists of a number of polymerization reactors followed by a number of units to separate water and monomer from the reaction products and a compounding extruder. The extruder is not only used for compounding but also for degassing of the monomer remaining in the melt. Typically, polymerization, separation, and extrusion are designed in different organizational units of the same or even different corporations using different approaches and supporting software tools. An integrated solution of this problem has to overcome the traditional gap between polymer reaction engineering, separation process engineering and polymer processing with their different cultures as well as the problem of incompatible data and software tools. Hence, the scenario poses a challenge for any integrated conceptual design process and its supporting software environment.

The design support software tools employed in the scenario are of a completely different nature. They include commercial as well as legacy tools. Examples are Microsoft Excel, various simulators such as Polymers Plus from Aspen Technology, gPROMS from PSE, Morex, BEMflow and BEMview from Institut für Kunststoffverarbeitung at RWTH

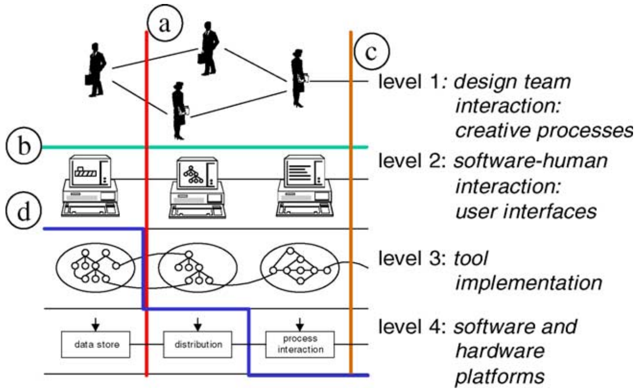

Fig. 2. Perspectives and levels of integration: The thick lines indicate necessary integration efforts on four different levels of interaction between software and humans.

Aachen, the project database Comos PT from Innotec, the document management system Documentum as well as the platform Cheops for run-time integration of heterogeneous simulators, the repository ROME for archiving mathematical models and the modeling tool ModKit, all of Lehrstuhl für Prozesstechnik at RWTH Aachen.

A prerequisite for IT support of chemical process design processes requires integration on a number of levels and from different perspectives as illustrated schematically in Fig. 2. There are four different levels of interaction. On the top level 1, interaction occurs in the design team, level 2 is addressing interaction between the human designer and various application software modules, level 3 relates to the interaction between different application programs and level 4 refers to the interaction between the application programs and the software and hardware platforms. The integration problems on these four levels are indicated by four bold lines in Fig. 2. They relate (a) to integration of the human work processes in geographically and institutionally distributed design teams, which has be accomplished to facilitate collaboration, (b) to the mismatch between design tool functionality, cognitive processes and the activities of the human during design, (c) to the different styles of conceptualization and information modeling applied in a different style on the four levels, and (d) to the a-posteriori integration of existing software tools—either standard (office) or domain specific applications—among each other and with the software and hardware platforms on levels 3 and 4 follow different styles of implementation which poses severe software engineering problems if their a-posteriori integration into a design support environment is envisioned. This understanding of the integration problems in supporting chemical process design processes has been the basis to shape the major research areas to be considered in IMPROVE. In particular, they include.

- the *modeling, analysis and reengineering design processes* by either integrating yet largely isolated design activities or by defining innovative design processes,
- the development of an *integrated information model* of the complete design process in the sense of an ontology,
- the development of *novel computer science concepts* and their prototypical implementation for information and collaborative work process management in engineering design processes,
- the implementation of a *demonstrator of an integrated design support system* to illustrate the synergy of integration and to prove the additional benefit to the end user by means of an industrially relevant and realistic design scenario, and
- the development of *software technologies* for the *a-posteriori integration of existing tools* and their functional extensions with an emphasis on the automatic generation of wrappers to homogenize interfaces.

Some results of IMPROVE will be presented in the remainder of this contribution. More detailed information with numerous references to publications originating from IMPROVE can be found at http://www-i3.informatik.rwth-aachen.de/research/sfb476/.

4. Modeling of design work processes and their products

A major objective of our research in IMPROVE is the development of an *integrated information model* which covers the work processes, the resources employed, and the resulting design (or product) data, which are typically organized in documents reflecting the context of a certain activity during the design process. Such a modeling activity is not self-sufficient. The resulting model can be used in a number of ways.

For example, *deficiencies of established design processes* may be identified as a prerequisite for their improvement and reengineering. Further, *new innovative work processes* may be developed from an analysis of existing approaches in order to better integrate traditionally separated activities. Examples include the tighter integration of mathematical modeling and cost estimation with the increasing refinement of the design in a continuous manner, despite the constraints imposed by current tool function. Another example relates to the improved integration of different design domains such as polymer reaction, monomer separation and polymer processing.

Besides these engineering related use cases, the information model is the basis for a *model-based top-down design of new software tools* with innovative functionality and for the integration of these new and of existing tools to a design support environment. The envisioned information model not only has to cover work processes and the information generated and used, but has also to describe the design process and the associated information from various perspectives with differing levels of detail.

Fig. 3 shows some *relevant perspectives* on the *information managed during the design process* on various levels of detail and with various degrees of formalism (Bayer, 2003).

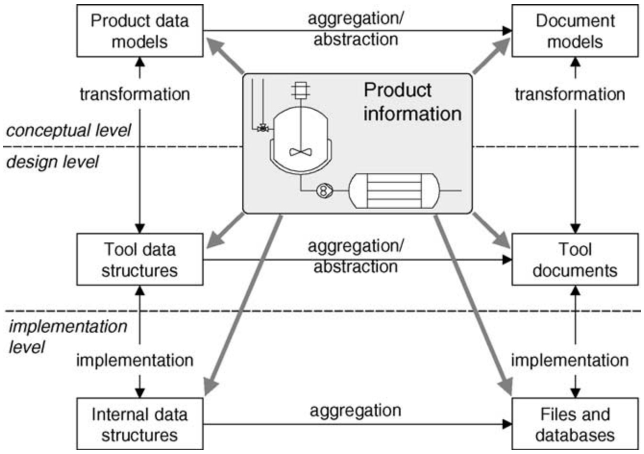

Fig. 3. Different perspectives on an integrated information model for the representation of design process information.

First of all, the major process design concepts have to be represented on a *conceptual level* (Fig. 3, top) to address the needs of the designers in the team. For example, such a conceptual model facilitates a common understanding of the design process and its results, a prerequisite for improving the design process or for formulating requirements on appropriate design support software tools.

The conceptual information model can be transformed into a *design model* (Fig. 3, middle). It serves the needs of the software engineer during software development and also determines the user interface of tools. Finally, the design model is implemented by means of some technology resulting in the *implementation model* of the design support software (Fig. 3, bottom). In addition to these levels of detail and degrees of formalization, we also distinguish between the data itself (Fig. 3, left) and the documents as carriers of data related by a certain design context (Fig. 3, right). Hence, documents link contextual design data to the work process.

In the sequel, we will discuss some of the information models developed and their relation. For the sake of clarity, the focus will be largely on the conceptual level. Besides such a conceptual model, various more refined and strongly formalized implementation models have been derived from or related to the conceptual model in IMPROVE.

4.1. *Work process modeling during empirical studies*

The investigation of existing work processes in empirical studies is supported by means of the *work process model* C3. It is a semi-formal model which aims at a coarse representation of the work process. C3 is based on the Unified Modeling Language, UML (Rumbaugh, Jacobson, & Booch, 1999), but includes a number of specific extensions (Foltz, Killich, Wolf, Schmidt, & Luczak, 2001). It supports work process modeling in a hierarchical manner on an arbitrary level of granularity. It covers the *roles* of the *members* of the design team, the *order of activities* carried out in a certain role, the *information* used, modified or generated, as well as the *resources* (software tools, in particular) employed during an activity. C3, implemented by the Workflow Modeling System (WOMS), facilitates the acquisition and documentation of actual work processes by industrial designers with little extra effort due to its easily accessible and illustrative graphical notation (Schneider & Gerhards, 2003). The *weak degree of formalization* is considered a strength of C3. It minimizes the modeling effort to a minimum which is essential for being accepted by always time constrained industrial designers. The following questions can be answered after an empirical study of an existing work process:

- Which design process step is carried out by which team member in which role and in which organizational unit?
- Which resources (tools, etc.) have been used?
- Which information is being exchanged between tools?
- Which documents are exchanged between team members?
- Which information has to be stored for documentation and later reuse ?
- Which relation exists between data and documents ?

The understanding obtained in the study is a good starting point for the improvement and reengineering of the work processes. For example, recommended work processes can be defined in C3 and documented by means of WOMS. Further, the C3 work process model can form the starting point for further extension and refinement to a conceptual information model of the work process (Eggersmann et al., 2003) which itself can further be transformed in the sense of Fig. 3 in order to assist the development of software supporting the design process in geographically and institutionally distributed teams (Eggersmann et al., 2002).

4.2. *The conceptual information model CLiP and its applications*

The conceptual information model CLiP has been developed to clarify the most important concepts and their relations for the description of chemical process design processes in the sense of an ontology (Uschold & Gruber, 1996). The design of CLiP is based on ideas from general systems theory (van Gigch, 1991), which have been successfully applied to represent complex structured systems in various domains. Its design philosophy is detailed by Bayer and Marquardt (2004).

The development of CLiP aims at a *well structured* and therefore *extensible information model*, which ultimately covers all the design data produced during the design process, the mathematical models used in the various model-based design activities, the documents for archiving and exchanging data between designers, collaborating institutions, or software tools, as well as the design activities with the resources they use.

CLiP is not planned as an information model which fixes all the details of the universe of chemical process design in a comprehensive manner. Rather, it is understood as a *modeling framework* in the first place to provide a coarse structure for the very diverse types of data occurring in the design process.

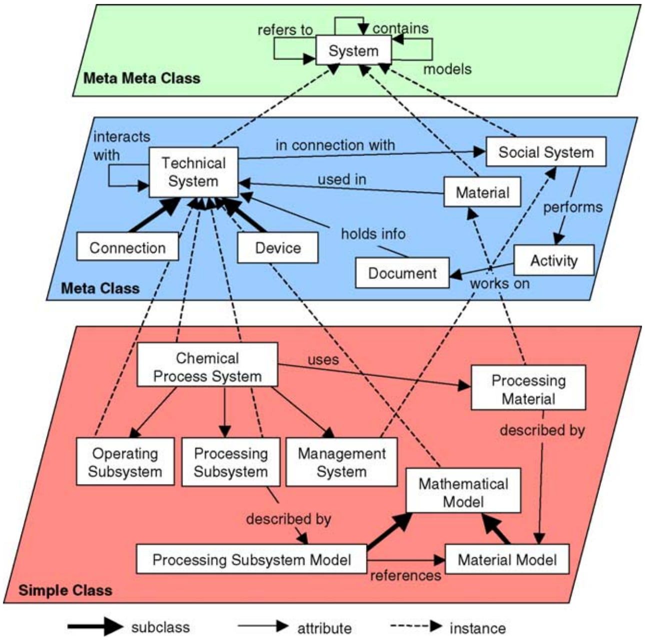

Fig. 4. The conceptual information model CLiP: Meta model and partial model structures.

Such a model framework has to be open for extensions driven by the requirements of a certain application. Further, it has to been designed to allow for an *integration of already existing data models*. Fig. 4 gives an overview on the structure of CLiP emphasizing the results of the design process, the so-called *product data*. A more detailed description can be found elsewhere (Bayer, 2003; Bayer & Marquardt, 2004a, 2004b; Eggersmann et al., 2003).

Meta modeling has been used as a first structuring mechanism, in order to allow for an efficient representation of symmetric and recurrent model structures. This way the coarse structure of the information model can be fixed and a simple categorization of the most important modeling concepts becomes feasible. We distinguish the meta meta class level, which only defines the concept of a general system, the meta class level, which holds the major categories of concepts for our domain, and the simple class level, which defines concepts related to different tasks in the design process. The meta class level comprises a *technical system* with its constituents *device* and *connection*, the *material*, the *social system* consisting of the members of the design team, the *activities* carried out during the design process and the *documents* associated to the various activities. Hence, CLiP integrates the product data resulting from the design process with the information model of the design process itself by means of the *activity* and *document* concepts in Fig. 4 (meta class level). The activity concept is referring to the individual steps in the design process. The product data which is associated with a particular design activity is typically concatenated in some *document*. Documents are either passed between humans, between humans and software or between software to communicate on the status of the design from the perspective of a certain design activity. Hence, a document provides a certain view on the product data and explicitly links the design process to the design product as indicated in Fig. 4.

The *open model structure* of CLiP is achieved by grouping the concepts on the simple class level to related logical units. The resulting *partial models* relate to design tasks which are typically addressed independently in parallel or in sequence during the design process. The concepts in the partial models can be introduced and maintained largely independently from each other. However, since the same real object is often referred to in different design tasks from different perspectives with differing degree of detail, overlap, partial redundancy, conflicts, and even inconsistency can hardly be avoided. Existing relationships between concepts are explicitly captured by means of association links. These links are defined by means of integration classes to specify relations not only between concepts in different partial models but also between the associated data. To reduce the specification effort and the complexity of the resulting information model, only those relations are represented which are of any relevance in the course of the design process. This principle of systematic, task-oriented decomposition and subsequent selective reintegration is considered an essential prerequisite to successfully deal with the inherent complexity of an integrated information model covering the whole design lifecycle.

CLiP is implemented by means of *different modeling formalisms*. The meta model and some of the concepts of the simple class level have been implemented in ConceptBase (Jeusfeld, Jarke, Nissen, & Staudt, 1998). This system nicely supports meta modeling and offers a sound logical foundation with basic deductive reasoning capabilities to assist schema development and maintenance. All the partial models of the simple class level are represented by means of UML. This formalism is well-suited for large data models due to its graphical notation. The contents of documents are represented by means of XML (W3C, 2004). The information units within documents are linked to CLiP classes and their attributes by means of associations. This link is the prerequisite for explicitly relating information stored in a project database to that contained in design documents, typically stored in a document management system.

Currently, CLiP is being enhanced by *additional formal semantics* for various reasons. First, the associations between partial models can only be specified if a precise meaning of the concepts and attributes is established. Second, model development is facilitated and third, the model can be directly used by a reasoner based on description logics. This way, new data and concepts can be classified and introduced in an existing database. Also, browsing and retrieval of data can be assisted across heterogeneous data sources, if the semantically enriched data model is used as a homogenization layer. Still, a coarse conceptualization by means of UML is accomplished first, before the refinement and further formalization of the UML concepts is addressed by means of some *ontol-*

ogy language, such as DAML + OIL or OWL (Gomez-Perez & Corcho, 2002).

4.3. Application of CLiP—From conceptualization to implementation

The software implementation of design support functionality requires a *refinement and transformation* of this conceptual information model according to Fig. 3. This refinement may be organized by means of various horizontal layers on the simple class level. Such layers serve as an additional structuring mechanism to maintain transparency and to support extensibility. The specific refinement of the model is determined by the envisioned application and the target software platform. There may be more than one refined model, if different tools for the same or similar tasks are being used in an integrated software environment. Often, available data models are subject to reuse and integration. These data models can either be those used in the tools to be integrated, or some standardized data model such as PDXI (Book et al., 1994) or the application protocols of STEP (Yang & McGreavy, 1996) which have been developed for data exchange between the software environments of different organizations. Different ways of integrating existing data models with the information model framework CLiP have been discussed by Bayer, Schneider, & Marquardt (2000).

There have been a number of data modeling activities in the process engineering domain (see Bayer & Marquardt, 2003, for a critical review) without a proper validation. *Information model validation* is difficult in principle since there is very little theoretical foundation to decide upon the quality of a certain information model. Validation is only feasible if such a data model is implemented in a variety of different ways. Such an implementation requires a refinement of a conceptual model into an implementation model (see Fig. 3) which is more precise and more detailed. Such a refinement step with objectives of different target applications in mind often reveals shortcomings in the (more abstract) conceptual model with respect to generality and expressiveness. Hence, different implementations provide feedback to conceptual modeling and contribute to its improvement. If finally all the requirements of a potentially large number of implementation efforts can be met, the model has been proven to be generic on (a limited) set of test cases. Implementation of the data model requires the software engineer to capture its basic ideas and understand its underlying construction principles in a reasonable amount of time. A good model has to be sufficiently transparent in order to facilitate this process. Still, the ultimate test for the data model is only possible after implementation and testing of the software by a larger user community. The data model should only be considered valid if the resulting software matches the cognitive model of the user which is a prerequisite for an easy to use software tool. CLiP has been forming the basis for various software development projects in IMPROVE in order to contribute to assess and validate its expressiveness, transparency and generality.

In particular, CLiP has been used for example to extend the *database schema* of the *project database Comos PT* of Innotec to also cover conceptual design data (Bayer, 2003). Originally, the database schema of Comos PT has been focusing on detail engineering and maintenance only. The case study revealed the versatility of CLiP and its simple integration with existing data models.

Another case study carried out in IMPROVE is related to the *integration of different software tools* (Bayer, Becker, & Nagl, 2003a). CLiP is refined into the data model for the specification of so-called integration documents which explicitly model the relations between the schema and the data of the implementation models of different tools. This way, an integration of tools is facilitated by a selective data homogenization approach without the need for defining and implementing a centralized data store (see Section 5). Such an approach avoids the problems of data centered tool integration as often practiced in the software industries, which in particular relate to the maintenance and the implementation of the necessarily complex data model of the complete design process.

In contrast to this tool-to-tool data integration, CLiP is also being used in IMPROVE as a basis for the implementation of a *data warehouse* to integrate heterogeneous data sources such as tool specific file systems or databases which inevitably occur in an integrated design support environment. Such a process data warehouse not only archives all data generated, but also the work processes operating on these data (see Section 5).

Besides the application of the product data model of CLiP for the implementation of information management functionality, e.g. for archiving of the design data generated during a design project and for the exchange of data between tools, the integrated data model can also be used as a starting point for the implementation of tools which support the *execution of work processes* during design. Such tools can be considered generalized workflow systems which, in contrast to existing workflow systems, satisfy the needs of highly dynamic and creative engineering design processes. At least in the medium time range, such systems are considered of high industrial relevance. The focus will shift from mere information management to an efficient support of the execution of high-quality design processes. Two work process support approaches are pursued in IMPROVE (see Section 6). They aim on the one hand at the guidance of an individual designer during the execution of unstructured and not properly planned personal work process, and on the other hand on the administration and management of the complete design process carried out by the design team.

4.4. Some lessons learned and future challenges in information modeling

Four major and largely independent issues will be briefly sketched in the sequel. They relate to empirical studies of design processes, the integrated modeling of data, documents and work processes, the structuring of an integrated information model and its application.

Work process oriented information modeling has to rely at least in part on *empirical studies* of real industrial design processes. These empirical studies, however, should not only be confined to clarify the social context of a design process (Bucciarelli, 1994). Rather, they should be related to the concrete engineering domain and to the information technology support of design, either desired or actually used. According to our experience, empirical studies have to be goal oriented towards an in-depth understanding of the design process relating organization, management, resources, requirements, tasks, and results produced. The understanding is at best cast into an information model. Since it is impossible to completely formalize (on a fine-grained level) creative conceptual process design, the information model has to remain coarse-grained (and hence vague) in parts. Such a focus on understanding and modeling is comparable to inductive (empirical) mathematical modeling of chemical processes. Acquisition of real work process data is most effective if it is carried out by the designers themselves. WOMS has proved to be a useful tool to support such work process data acquisition. As in mathematical modeling, this *bottom-up approach* of empirical studies has to be complemented by some deductive component as in fundamental chemical process modeling. Obviously, such a *top-down* component of modeling a design process requires a “design theory” or, more pragmatically, a good understanding of current design practice or preferred design processes. A meaningful combination of both approaches remains a challenge for future research (Foss, Lohmann, & Marquardt, 1998). As soon as an information model of the existing design process is available, techniques from business process engineering may be applied to improve the design process and to formulate requirements for computer-aided support (Hammer & Champy, 1993).

The integrated consideration of *data, documents and work processes* together with the resources used and the organizational structures involved seems to be appropriate. Still, a lot of conceptual as well as technical issues of developing and validating such an integrated information model have to be addressed by future research work. A much better capturing of the real design process seems to be possible, if documents of all kinds are systematically considered to link design data and work processes. Documents are not only used to archive a design, but they also play a dominant role in the exchange of design results between people across organizational boundaries. Hence, they are closely related to the human part of the work process. Further, documents can be interpreted as input files and they are the result of the execution of some software tools. Therefore, documents relate to the computer-assisted part of the work process. Documents always define the context of a work process and provide a situated view on the design data. Separated documents, however, do not allow a comprehensive and consistent presentation of the whole configuration of the design data. Hence, more work has to be done to clarify the conceptual relations between different types of documents and their data items.

An integrated information model of the design process lifecycle has an immense *inherent complexity*. An appropriate structure of a multi-faceted information model is crucial to facilitate transparency, extensibility, and maintainability. The continuously evolving formalisms and languages for information model representation are further complicating the problem. Just in the last 15 years, we have seen entity-relationship, frames, object-oriented, description logic and ontological representation paradigms. Last but not least, the collaborative work process of information modeling has to be properly defined, managed, and supported by suitable tools.

The resulting information model not only provides a *common understanding* of the domain of interest within the design team. It is also mandatory for a fully *model-based top-down design of design support software* systems. There are many applications which can benefit from the same integrated information model such as tool development, integration of existing tools, data exchange between tools and organizations, homogenization of heterogeneous data sources, or the realization of the semantic web to create the knowledge base of an enterprise. Ideally, all this software should be generated automatically from a formal specification. There is obviously a long way to go due to the complexity of the design domain.

5. Architecture of a future integrated design environment

The information models introduced in the previous section are indispensable for a top-down design and for the implementation of integrated design environments. Before we discuss advanced cooperative design support under development in IMPROVE in Section 6, we present and discuss a coarse software architecture which is suitable for the work process oriented integration of existing and novel software tools.

5.1. An example architecture

Fig. 5 depicts a sketch of a software architecture of a future design support environment. A prototype of such an environment with partial functionality has been implemented and evaluated in IMPROVE.

The environment comprises *existing tools* typically employed in industrial practice which stem from different sources, either commercial or in-house. Tool integration is still and will remain of substantial interest for the operating companies despite the substantial effort of major vendors to integrate their own tools with each other and with those of selected collaborating partners. The end users in the operating companies are interested in customizing their design support environments by integrating additional tools and data bases provided by other vendors or in-house development groups in order to differentiate their technology from that of their com-

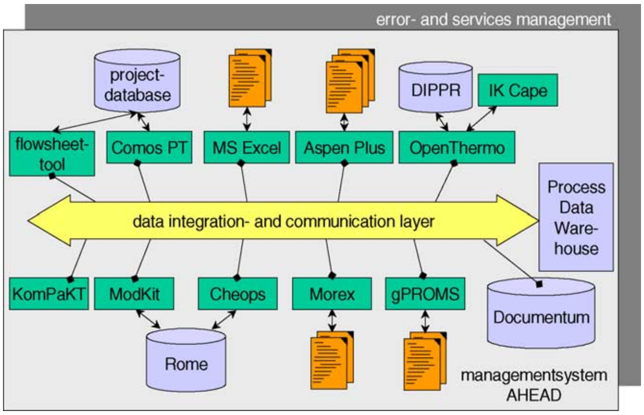

Fig. 5. A coarse sketch of a software architecture of a future integrated design environment as partially implemented in the CRC IMPROVE.

petitors. The software to be integrated can therefore be either "complete" design environments from some major vendor or highly specialized tools or data bases from niche providers. The tools are wrapped by thin software layers to provide standardized interfaces for data exchange and method invocation employing state of the art middleware technology (Adler, 1995). The interface definition is guided by the conceptual information model of the design process discussed in the previous section. The design documents and their evolution during the work processes determine the interface definition to a large extent, since they are providing the context for tool interoperation in a natural manner.

The architecture in Fig. 5 suggests *interoperation of very different types of software modules* in an integrated design support environment. There are, for example, general purpose process modeling environments (e.g. Aspen Plus from Aspen Technology or gPROMS from Process Systems Enterprise) as well as dedicated simulation tools (e.g. Morex for the simulation of extrusion processes). In addition to the various simulation capabilities various data bases need to be integrated. For example, a project database (e.g. Comos PT from Innotec) is required to store the major product data during a design project. Such a project database may offer a flowsheet centered graphically supported portal to access the design data stored as well as interfaces to a limited number of design tools. Alternatively, a separate flowsheet tool with extended functionality (Bayer, Weidenhaupt, Jarke, & Marquardt, 2001) could be integrated in order to serve the needs of other tools integrated in the environment. In addition to the project database, a physical property database (e.g. DIPPR) with raw experimental data as well as parameters for physical property correlations and a repository for storing mathematical models of different kinds (such as ROME (von Wedel & Marquardt, 2000)) are part of the integrated environment. A commercial document management system is used to serve as an archive for all design documents. A process data warehouse captures the design data in the context of the work process (Jarke, M., List, T., Köller, J., 2000).

In order to support the execution of distributed design process, the management system AHEAD (Nagl, Westfechtel, & Schneider, 2003) is integrated. It assists the project manager in allocating and monitoring the resources (e.g. the members of the design team and the tools they use), in providing a consistent set of documents produced during the design project, and in keeping track of all the activities carried out during the design process on a medium-grained level. An extended middleware platform developed as part of CRC IMPROVE provides load balancing, error handling and service management for the integrated design environment which is typically operated in a distributed wide area network.

5.2. Integration approach

The software *integration* approach chosen is *driven* by the characteristics of *actual design processes*, the resulting product data distributed in documents of various kinds and the relations between those documents, or the data items they contain. It is not intended to extract the design data, completely or in parts, from the native data stores of tools in order to duplicate them for example in a central data warehouse and store them together with the relevant associations existing across the various tools. Rather, in contrast to such a data centered integration approach followed by all commercial integration solutions, we preserve the native data stores of the tools to be integrated.

Hence, integration is achieved by means of *a-posteriori homogenization* of *heterogeneous data sources*. For this purpose, the data and communication layer of the architecture (see Fig. 3) is equipped with dedicated mediators (Wiederhold & Genesereth, 1997) which map the data instances between data sources and sinks. The process data warehouse stores the meta data which are required to trace work processes and the resulting product data for documentation purposes and to facilitate later reuse in the same or in a different project (Jarke et al., 2000). Such an integration approach has been advocated by a requirements analysis of a number of German operating companies (Klein, Anhäuser, Burmeister, & Lamers, 2002). If integration considers both, the work processes as well as the data handled in a particular design context, the implementation and maintenance effort of integrated solutions is limited.

5.3. Providing new functionality for collaborative design

New design support functionality has to be provided by means of a *functional extension of existing software tools* (e.g. a simulator or a project database). These extensions have to be accomplished without reengineering existing tools which is typically not feasible because of commercial as well as technological constraints. Hence, the functional extensions of existing tools are implemented as separate and self-contained software components. These software components are subsequently wrapped by a thin software layer to implement logically as well as technically matching interfaces to facilitate integration with existing tools. Examples of such new functionality under development in IMPROVE will be discussed in Section 6.

In many cases, some desired functionality is already available as part of an existing tool. Often, the level of sophistication of the available implementation is too limited in order to apply it for a related purpose for which it has not been designed originally. In such cases, it would be desirable to *isolate* and *extract* the *available generic functionality* from the existing tool in order to offer its service to other tools in the integrated environment after the required extensions and modifications. For example, most computer-aided process engineering tools include some software module for the specification, representation and visualization of flowsheets. Typically, the level of abstraction and the information content covered is determined by the specific task addressed by the tool in the design process. It is obviously preferable from a usability as well as from a maintenance point of view to centralize all the flowsheet functionality in a single dedicated tool. Such an advanced flowsheet tool (see Fig. 3) is designed to fulfill all the requirements for managing flowsheet representations on various levels of granularity and for browsing and retrieving flowsheet related design data (Bayer et al., 2001).

In practice, the extraction of some functionality from existing code may not be possible. There are at least two reasons: the source code may not be available, or the functionality to be extracted may be tightly linked to other tool functions such that the extraction is impossible without complete reimplemention of the tool. In those cases, the functionality is not extracted, but it is bypassed instead. An extended functionality superseding the existing capabilities is provided by a new dedicated tool as part of the integrated design support environment.

5.4. *Some lessons learned and future challenges in tool integration*

A number of challenging issues have come up during our studies on the development of integrated design support environments. Some of them are briefly sketched in the sequel.

The *a-posteriori integration of existing tools* into an open integrated design support environment is meeting the expectations of the end users but is, at least to some extent, contradicting the objectives of the software vendors. The latter want to offer their own integrated solutions to extend coverage and market share. Especially, their tools do not offer transparent interfaces which easily allow tool integration. The data structures may not be documented or the data can not be exported. Existing tools often combine too much functionality in a single software systems due to historical reasons. Typically, the tools have not been designed for integration. Rather, they have been created in an evolutionary extension process which steadily extended the functionality of a monolithic tool. Obviously, a redesign and modularization of the tools would not only facilitate integration into open environments but would also reduce software maintenance cost. Both issues, the lack of transparent interfaces and appropriate modularization are hard problems for tool integration.

Middleware and wrapper technology has come a long way and is nicely supporting the control and platform integration aspect of tool integration (Wasserman, 1990) on a technical level. However, the interfaces are only standardized on a syntactic level, which is not sufficient for tool integration. Rather, *standardization on a semantic level* is required to ensure proper function and meaningful data exchange between tools. Such a semantic standard may be accomplished by ontologies which are tremendously pushed by semantic web approaches (Fensel, Hendler, Liebermann, & Wahlster, 2002). Ultimately, the classical tool integration dimensions (Wasserman, 1990) have to be extended by a work process dimension to provide context to the integration exercise. If such a work process orientation is lacking, tool integration is unnecessarily complex and costly to develop and maintain.

Hardware and software platforms are rapidly changing. The *technological progress in information technology* is driven by the mass consumer markets and not by the requirements of engineering design applications. The level of sophistication and functionality of the service layer on top of traditional operating systems is steadily increasing. Improved services simplify the implementation of integrated design environments and allow more advanced functionality. For example, multimedia services can be used for advanced communication between design team members. However, a careful modularization of the application becomes crucial to allow the absorption of consolidated new software technologies.

In summary, the integration of tools into useful design support environments at reasonable cost requires *careful architectural considerations*. Both, the integration of existing commercial as well as in-house legacy software and the absorption of evolving software technologies have to be accommodated. Vendors have to design their tools systematically for a-posteriori integration to satisfy the needs of their customers and to reduce their own development and maintenance cost.

6. New design support functionality

A work process oriented integration of existing design support software tools requires novel functionality if a new quality of support for collaborative design is aimed at. Subsequently, a selection of such novel support functions are discussed.

6.1. *Semantic support of individual designers*

A designer has accumulated a substantial amount of experience during previous design projects. The quality of the design processes can be improved tremendously if this implicit knowledge can be converted into explicit knowledge which

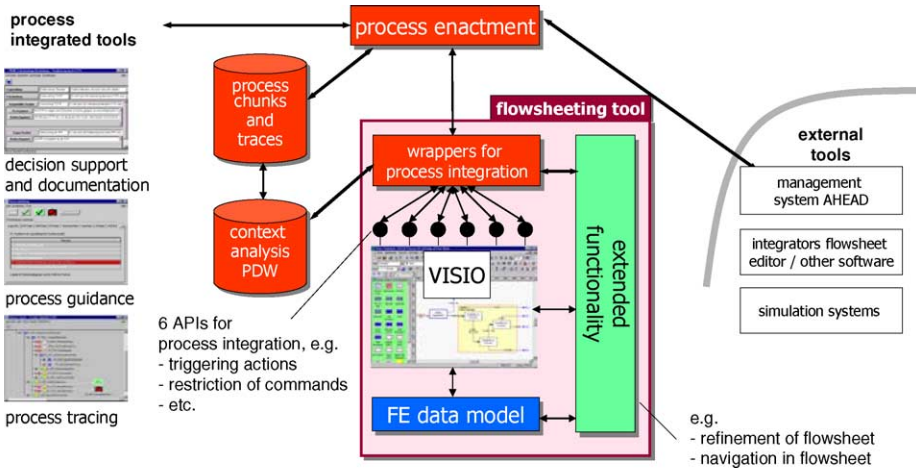

Fig. 6. The PRIME environment for supporting individual designers during their work processes.

is amenable to a later reuse by himself or by a colleague in a completely different context either within this or another design process. There have been numerous attempts to acquire implicit knowledge from experts by means of formal techniques in artificial intelligence. These techniques typically require a basic *understanding* of the business processes of interest. Since *creative design processes* are, at least in part, not sufficiently well understood to effectively guide such knowledge acquisition processes and since experts are not always cooperating well, a new approach formerly suggested in the context of requirements engineering was adopted to apply to engineering design processes in IMPROVE. We briefly sketch the idea in the following with reference to the architecture of a PRIME in Fig. 6, an environment for supporting individual design processes, and refer for details to the work of Pohl et al. (1999).

Instead of acquiring knowledge a-posteriori by means of structured interviews, reviews of past design processes, etc. the design process is recorded automatically by a *process tracing tool* during its execution. The *recording* results in so-called *process traces* which capture all the major steps carried out during the design process together with the data and documents which have been handled. These traces are stored in a database (Fig. 6), which is part of the process data warehouse of the integrated design support environment (see Fig. 5). The traces are not only used to document the work processes in detail. Rather, they provide the basis for *design context analysis* and for interactively extracting repetitively occurring *process chunks* applicable in a certain design context. Again, chunks and design context are stored in databases (Fig. 6). As in the area of mathematical process modeling, such an identification task can be supported if the purely data driven identification is complemented by some a-priori knowledge. While such knowledge is comprised by model structures derived from the fundamental laws of physics in mathematical modeling, it is not that obvious what kind of a-priori knowledge can assist the discovery of design process chunks. We are currently investigating to what extent specific parts of a design process can be modeled on an abstract level in order to provide parameterized chunks which could guide the discovery process based on process traces.

The process chunks and the design contexts are supposed to be employed by the *enactment tool* of the PRIME environment to assist the individual designer during repetitive activities. The enactment tool has to analyze the current context of the design process first. Next, it has to match it with similar contexts stored in the context database. If a matching context has been found, applicable process chunks are retrieved from the process chunks database and suggested to the designer. *Decision and documentation support* as well as *guidance* are provided to the designer who interacts with via an integrated flowsheeting tool. After his approval and after providing lacking context data, the process chunk is enacted. The enactment of a process chunk typically requires the invocation of external applications such as a process simulator.

6.2. Administration and coordination of the complete design process

Individual designers are typically contributing to different design processes simultaneously. All these processes are administrated and coordinated by a chief design engineer, the manager for short. Obviously, the individual design processes are not independent but highly interrelated by the documents they work with and by the resources they share. The resources include time and budget, team members, experimental facilities and available software tools. Inevitably, the inherent complexity of the design processes requires *management support* to effectively *monitor and coordinate the design processes* and the associated activities, to keep track of the resulting design documents and their relationships, and to administrate and allocate the available resources. The strong relation between resources, activities, and documents has to be taken into account for a proper allocation of resources to specific design tasks as well as for consistency management of the documents.

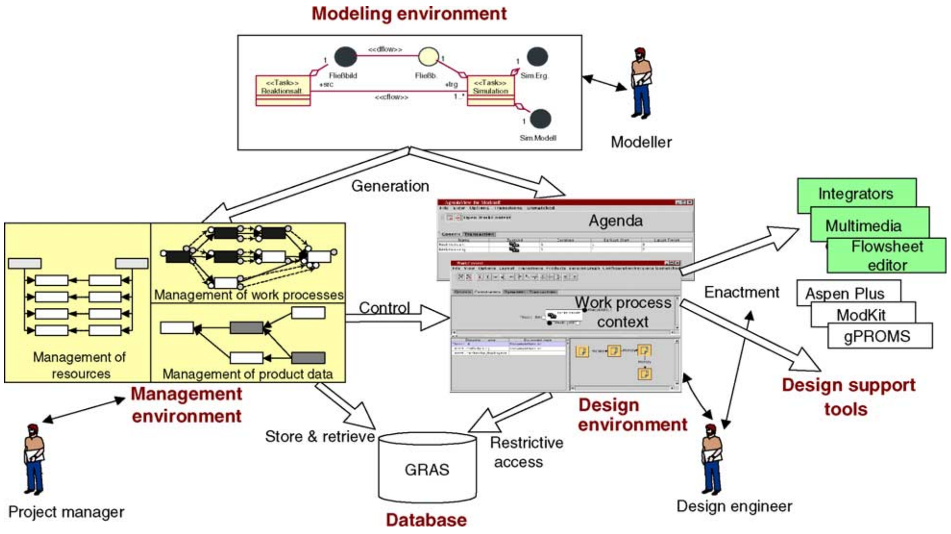

Fig. 7. The AHEAD environment for supporting the management and the execution of design processes in a team of designers.

AHEAD, a software tool to *support the management of cooperative design processes and their interdependencies on a coarse-grained level*, provides functionality for two different kinds of users, the manager and the designer (Nagl et al., 2003). A rough sketch of the architecture of AHEAD is shown in Fig. 7. The *manager* is supported by a *management tool* consisting of three fully integrated tool sets. Dynamic task networks with control and data flow interrelations are provided to implement *work process management*. Version control, configuration management and an explicit notion of the dependencies between documents are provided to facilitate *management of product data*. The *resource management* allows for the definition of the organizational structure of the design teams working on the various design processes. The *designer* is supported by a *design environment* which comprises an *agenda tool* to display the upcoming tasks to be carried out by the design team members, and of a *work process context tool* to manage the documents and the software tools required to carry out a certain design task. The latter links with existing software applications to invoke their context driven execution. The design and implementation of AHEAD directly addresses the *inherent dynamics of a design process.* In particular, the task networks in the management environment can be modified at any time during project execution to reflect changes in the design process as consequences of emerging insight into the design problem or handling of problems and mistakes. Further, an adaptation of the functionality to the peculiarities of a given domain of application is possible by means of a *modeling environment* which facilitates the representation of domain specific knowledge, for example, related to the capabilities of the tools employed. Domain specific code is generated to customize the management tool to the domain. This facilitates customizing to the peculiarities in the design process of a certain company or even to the requirements of some industrial domain.

The design support offered by AHEAD is purposely limited to coarse-grained activities in order to facilitate the link between the actual design work carried out by the design teams and the management of related design processes. Hence, it differs in scope from the work process support offered by the PRIME environment which focuses on guiding and supporting activities of an individual designer on a fine-grained level.

6.3. Multimedia communication in distributed design teams

Geographically distributed design teams already use a multitude of services including e-mail, groupware systems, joint workspaces or even video conference systems in order to facilitate *synchronous and asynchronous communication*. Typically, these services are not integrated among each other, and more importantly, with the engineering design tools of a given domain. Hence, the available communication support systems do not offer sufficient functionality to effectively assist the members of distributed engineering design teams.

For example, during the design of an extruder as part of a polymer production process, the potential separation of remaining monomer from the polymer melt during polymer processing in the extruder has to be assessed in order to decide on the degree of monomer separation in the evaporation unit following the polymer reactors. This question can only be resolved effectively, if the chief engineer, the extrusion expert and the separation expert—all working at different locations and in part in different institutions—can easily *communicate via multimedia services* which are *seamlessly integrated with the design support environment*. Only then, all the team members have access to the same set of currently valid design documents and to all the required software tools to jointly carry out the necessary design studies during their virtual conference. For example, they may carry out a CFD simulation of the degassing melt flow in the extruder and a process simulation to study the effect of shifting the monomer separation partly from the evaporator to the extruder. The results of the simulations have to be discussed immediately to decide on the required equipment design modifications of the extruder given the multiple domain specific requirements.

In order to support such a scenario effectively, the system KomPaKT has been developed in the CRC IMPROVE and evaluated on the basis of the polyamide-6 design case study (Schüppen, Trossen, & Wallbaum, 2001). KomPaKT offers a set of modular services in a homogeneous environment to support the needs of *multimedia conferencing in engineering design applications*. Communication is supported asynchronously, for example by e-mail and audio messages, and synchronously by means of a whiteboard and video streams. Floor control and conference management functions are also provided. KomPaKT is integrated with AHEAD in order to support spontaneous as well as planned conferences. AHEAD provides information on the organizational data of the project, the tools and the documents of a design context of interest. Communication on design issues is supported by *application and event sharing mechanisms*. In application sharing, the output of a design tool residing on the computer of one designer is presented to all participants of a multimedia conference. Often, communication bandwidth is not sufficient if 3D images or movies have to be transmitted. In those cases, event sharing is more appropriate. An instance of the design tool is then available on every team member's computer and only control information is communicated to synchronize the different instances of the tool during communication in the multimedia conference.

6.4. Document oriented tool integration

Tool integration is always possible via input and output data which form a certain configuration of the product data denoted as documents, if the data contained in the documents of two different tools can be mapped to each other in a consistent manner at any time during the design process. Despite the independent creation and incremental revision of such documents by individual design tools, there exist a large number of fine-grained dependencies between the data contained in different documents. For example, the abstraction of the process flowsheet used to define the steady-state simulation problem has to match the real flowsheet stored in the project database. Inconsistencies between the various documents are unavoidable. However, a certain level of consistency has to be established as soon as two tools of a design support environment are used in a cooperative manner.

The manual reconciliation of the content of associated documents is time-consuming and error-prone. Hence, *integration tools* are preferable which *automate such a reconciliation process to the extent possible*. It should be noted that a fully automated integration is not feasible in many cases because of a potential semantic mismatch between the data models employed by the tools to be integrated. This mismatch can only be resolved manually. Obviously, document oriented integration tools are crucial for the implementation of design support environments (as suggested in Fig. 5) which do not rely on integration via a centralized design data store.

Document oriented integration functionality is subject to research and development in IMPROVE (Bayer et al., 2003a). The integration tools developed assist the user in consistency analysis of two documents, in browsing document content and in the necessary transformations between documents. They operate in an incremental manner and propagate only the increments between documents in a bi-directional manner. They are interactively used by the designer in order to control the transformation process. The reconciliation of the documents is automatic if possible, it can also be assisted by manual interaction of the designer in those cases, where the integration mechanisms fail. The reconciliation is rule based. The rules build on an information model of the documents to be integrated. The objects of the two models are related to each other by means of an integration document, which holds the links between the data items in the two documents. These links are derived by refining the associations between concepts in different parts of the conceptual information model defined in CLiP. Because of a model-based design, the integration tool can be customized to the peculiarities of the tool documents to be reconciled, if the conceptual information model covers the data objects in the documents semantically.

Various integrators between different tools have been developed and tested as part of the activities in IMPROVE by employing a common reference architecture. Fig. 8 shows the architecture of document-oriented integration of the process simulator Aspen Plus and the design process database Comos PT. The data models of both tools are represented in the *Aspen Plus and Comos PT documents*. The integration document reconciles the two proprietary data models. Data integration is accomplished during execution by the *integrator* which is relying on a rule base derived from the correspondences in the integration document or on interactive input from the designer, for example to fill in missing information or to support conflict resolution during integration. The integrator can be generated from a formal specification provided in the modeling environment.

6.5. Advanced tools for mathematical model development, maintenance and reuse

Chemical process design has been quickly moving towards solutions which heavily rely on *mathematical models*. Process simulation is used on a routine basis during conceptual design today assisting the analysis of design alternatives. Tomorrow, the generation of a design alternative itself is routinely supported by short-cut methods and partly automated by rigorous structural optimization employing a multitude of tailored mathematical models.

The variety of mathematical models requires their management *across the design process lifecycle* (Marquardt et al., 2000). Two objectives can be distinguished, namely the inte-

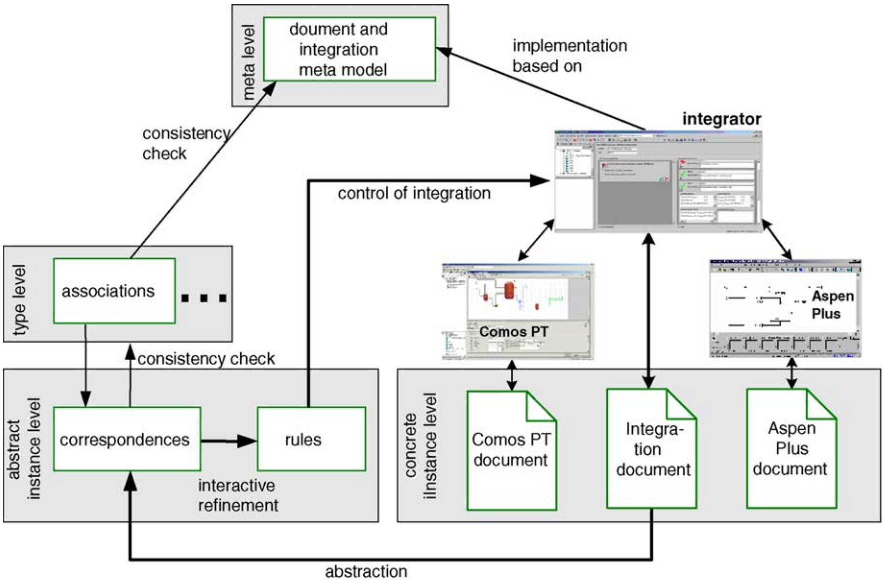

Fig. 8. Document-oriented integration of two software tools, a case study for the integration of a process simulator with a process design database.

gration across the process of mathematical modeling to reuse existing model knowledge downstream in the design process and to integrate existing models at runtime to facilitate multi-model, multi-method and multi-platform integration of simulation and optimization tools. Until recently, traditional heuristics and experienced based design have been largely separated from model-based design. Consequently, the software environments used in both areas are not integrated, neither conceptually nor technically.

Both issues, the *management and integration of mathematical models* across the lifecycle as well as the integration of design data, mathematical models and the results produced during simulation experiments are addressed as part of the IMPROVE project. For the support of mathematical modeling, three complementary software systems are under development. ModKit (Bogusch, Lohmann, & Marquardt, 2001) supports the *generation* of tailored *mathematical models* which cannot be found in the library of a simulator. The model can either be exported into the proprietary format of a commercial process modeling environment or in a neutral format derived from Modelica (Mattson, Elmqvist, & Otter, 1998) to facilitate *model exchange* between applications. Models generated by either ModKit or any other commercial modeling environment can be *stored* in their native form in the *model repository* ROME (von Wedel & Marquardt, 2001). Hence, ROME stores symbolic models in a neutral format or in any proprietary format of a commercial simulator, declarative equation-based models as well as executable block-oriented models. Links between models in a flowsheet or between models from different sources are kept at this point on a coarse-grained level only in the database schema which derives from the appropriate partial model in CLiP. Models can be checked out in their native form to be processed by the appropriate simulation tool. However, models from different sources can be linked to a single flowsheet and *integrated during runtime* by means of Cheops (von Wedel & Marquardt, 2000). Cheops allows steady-state as well as dynamic equation-oriented and modular simulation using existing dedicated simulators which have been developed for specific parts of a process. For example, in the polyamide-6 case study, Polymers Plus may have been used for polymer reactor modeling, gProms for monomer separation from polymer melt in a wiped-film evaporator, and the legacy tool Morex for the modeling of the extrusion process. These simulators are wrapped by standard interfaces and integrated with a configurable simulation strategy (modular, simultaneous, or mixes thereof) to form a simulator of the complete flowsheet showing a recycle of the unconverted monomer. This reuse of individual models is possible without the need for a costly and error-prone reimplementation in a single process modeling environment.

Mathematical models and their results have to be related to the design process and in particular to the design data. However, *mathematical models and design data* are kept in different tools without explicitly accounting for relations between them. Obviously, there is a significant overlap and the risk of inconsistencies in these two data sets. Further, tracing of the design process and its rationale requires an explicit relation between design data and mathematical models (Bayer, von Wedel, & Marquardt, 2003b). Such an integration is currently being carried out using ROME as a model repository to archive models from various simulators in a coherent manner in the first place and Comos PT which serves as the project database storing relevant design data. This kind of integration may be considered a special case of the homogenization of related data from different sources as discussed already above.

6.6. Discussion

The advanced functionality discussed in the previous subsections is not meant to be the only necessary to effectively upgrade current design environments for collaborative and geographically as well as organizationally distributed conceptual design processes in the process industries. Many other support functions to improve the efficiency of collaborative design are conceivable. We have limited our attention on those activities which are currently being studied in IMPROVE.

There is yet very little experience with those functionalities which impact the way a designer works. This is not just a matter of human-computer interaction which is essential for both, acceptance and high productivity. An interesting question also concerns the *social implications of such an extended design functionality* (see Brown & Duguid (2000) for a general discussion). More and more activities are becoming computer-based, the interaction between humans is changing in quality with unforeseen consequences, for both, the quality of the design and the satisfaction of the designer. Further, the *full transparency* of the design process results in an almost complete assessment of the performance of a designer. Any inefficiency or any mistake is documented. Obviously, such transparency has to be handled with care by management.

7. Conclusions

This contribution has attempted to show that information technology support of engineering design processes (not only in the chemical process domain) is a complex and far reaching problem. It goes well beyond the classical problem of data exchange or of data centered integration of tools to a design environment. IMPROVE addresses this problem area in a long-term project. The objective of the research work is the clarification of work process oriented support of engineering design by means of information technologies. This objective is considered to be the guiding paradigm of the research work and determines the concrete research projects in the center to a large extent.

Some of these research issues together with results obtained and experience gained have been summarized in this contribution. Despite the long-term and fundamental research focus of IMPROVE, some of the concepts and technologies have already reached a level of maturity which is sufficient to start transfer into industrial practice in focused joint research and development work with the software and end user industries.

Acknowledgements

The authors thank the German Research Foundation (DFG) for the financial support of the Collaborative Research Center CRC 476 (Sonderforschungsbereich SFB 476) and all members of the CRC for their fruitful collaboration, without which the results presented in this paper would not have been possible.

References

Adler, R. M. (1995). Emerging standards for component software. *IEEE Computer*, *28*(3), 68–77.

Backx, T., Bosgra, O., Marquardt, W. (1998). *Towards intentional dynamics in supply chain conscious process operations*. FOCAPO ’98, Snowbird, Utah, 1998. Online available from http://www.lfpt.rwthaniraachen.de/Publication/Techreport/1998/LPTanir1998anir25.html. edu.

Banares-Alcantara, R. (1991). Representing the engineering design process—two hypotheses. *Computer Aided Design*, *23*, 595–603.

Banares-Alcantara, R. (1995). Design support systems for process engineering. Part I. Requirements and proposed solutions for a design process representation. *Computers and Chemical Engineering*, *19*, 267–277.

Banares-Alcantara, R., & Labadibi, H. (1995). Design support systems for process engineering. Part II. KBDS: an experimental prototype. *Computers and Chemical Engineering*, *19*, 279–301.

Bayer, B. (2003). *Conceptual information modeling for computer-aided support of chemical process design*. Fortschritt-Berichte VDI: Reihe 3, Nr. 787. VDI-Verlag, Düsseldorf.

Bayer, B., & Marquardt, W. (2003). A comparison of data models in chemical engineering. *Concurrent Engineering: Research and Applications*, *11*(2), 129–138.

Bayer, B., Schneider, R., & Marquardt, W. (2000). Integration of data models for process design—first steps and experiences. *Computers and Chemical Engineering*, *24*, 599–605.

Bayer, B., Weidenhaupt, K., Jarke, M., & Marquardt, W. (2001). A flowsheet centered architecture for conceptual design. In R. Gani & S. B. Jørgensen (Eds.), *European Symposium on computer aided process engineering: Vol. 1* (pp. 345–350). Amsterdam: Elsevier.

Bayer, B., Becker, S., Nagl, M. (2003a). Integration tools for supporting incremental modifications within design processes in chemical engineering. In: Chen, B., Westerberg, A.W. (Eds.), *Process systems engineering* (pp. 1256–1261). Elsevier.

Bayer, B., von Wedel, L., Marquardt, W. (2003b). An integration of design data and mathematical models in chemical process design. In A. Kraslawski & I. Turunen (Eds.), *Proceedings of the European Symposium on computer aided process engineering* (Vol. 13, pp. 29–34). Elsevier.

Bayer, B., & Marquardt, W. (2004). Towards integrated information models for data and documents. *Computers Chemical Engineering*, *28*, 1249–1266.

Bayer, B., Marquardt, W., 2004b. A conceptual information model for the chemical process design lifecycle. In: Jarke, M., Jeusfeld, M. A., & Mylopoulos, J. (Eds.), *Meta modeling and method engineering*. MIT Press.

Bogusch, R., Lohmann, B., & Marquardt, W. (2001). Computer-aided process modeling with ModKit. *Computers and Chemical Engineering*, *25*, 963–995.

Book, N., Sitton, O., Motard, R., Blaha, M., Maia-Goldstein, B., Hedrick, J., & Fielding, J. (1994). The Road to a common byte. *Chemical Engineering*, *101*(9), 98.

Brown, J. S., & Duguid, P. (2000). *The social life of information*. Boston, MA: Harvard Business School Press.

Bucciarelli, L. L. (1994). *Designing engineers*. MIT Press.

Costello, D. J., Fraga, E. S., Skilling, N., Ballinger, G. H., Banares-Alcantara, R., Krabbe, J., Laing, D. M., McKinnel, R. C., Ponton,

J. W., & Spenceley, M. W. (1996). epee: A support environment for process engineering software. *Computers and Chemical Engineering*, *20*, 1399–1412.

Davis, J. G., Subrahmanian, E., Konda, S., Granger, H., Collins, M., & Westerberg, A. W. (2001). Creating shared information spaces to support collaborative design work. *Information Systems Frontiers*, *3*(3), 377–392.

Eggersmann, M., Hackenberg, J., Marquardt, W., & Cameron, I. (2002). Applications of modelling—a case study from process design. In B. Braunschweig & R. Gani (Eds.), *Software Architectures and Tools for Computer Aided Process Engineering* (pp. 335–372). Amsterdam: Elsevier.

Eggersmann, M., Schneider, R., & Marquardt, W. (2002). Modeling work processes in chemical engineering—from recording to supporting. In J. Grievink & V. Schijndel (Eds.), *European Symposium on computer aided process engineering—12* (pp. 871–876). Amsterdam: Elsevier.

Eggersmann, M., Gonnet, S., Henning, G., Krobb, C., Leone, H., & Marquardt, W. (2003). Modeling and understanding process design activities—different types of activities. *Latin American Applied Research*, *33*, 167–176.

Fensel, D., Hendler, J., Liebermann, H., & Wahlster, W. (2002). *Creating the semantic web*. Cambridge: MIT Press.

Finger, S., Konda, S., & Subrahmanian, E. (1995). Concurrent design happens at the interfaces. *AI EDAM*, *9*(2), 89–99.

Fisher, L. (2000). *Workflow Handbook*, 2001. Lighthouse Point. Future Strategies Inc.

Foltz, C., Killich, S., Wolf, M., Schmidt, L., & Luczak, H. (2001). Task and information modeling for cooperative work. In M. J. Smith & G. Salvendy (Eds.), *Systems, social and internationalization design aspects of human-computer interaction*. Proceedings of HCI International (Vol. 2, pp. 172–176). Mahwah: Lawrence Erlbaum Associates, 2.

Foss, B. A., Lohmann, B., & Marquardt, W. (1998). A field study of the industrial modeling process. *Journal of Process Control*, *8*, 325–337.

Gomez-Perez, A., & Corcho, O. (2002). Onotology languages for the semantic web. *IEEE Intelligent Systems, January–February*, 54–60.

Hammer, M., & Champy, J. (1993). Business reengineering the corporation. In *A manifesto for business revolution*. New York: Harper.

Jarke, M., List, T., Köller, J. (2000). The challenge of process data warehousing. In *Proceedings of the 26th International Conference on Very Large Data Bases VLDB 2000* (pp. 473–483).

Jeusfeld, M. A., Jarke, M., Nissen, H. W., Staudt, M. (1998). ConceptBase—managing conceptual models about information systems. In P. Bernus & G. Schmidt (Eds.), *Handbook on Architectures of Information Systems* (pp. 265–285). Springer, Berlin.

Klein, R., Anhäuser, F., Burmeister, M., & Lamers, J. (2002). Planungswerkzeuge aus Sicht eines Inhouse-Planers. *Automatisierungstechnische Praxis*, *44*(1), 46–50.

Konda, S., Monarch, I., Sargent, P., & Subrahmanian, E. (1992). Shared memory in design: a unifying theme for research and practice. *Research in Engineering Design*, *4*, 23–42.

Marquardt, W., Wedel, L. V., & Bayer, B. (2000). Perspectives on lifecycle process modeling. In M. F. Malone, J. A. Trainham, & B. Carnahan (Eds.), *Foundations of computer-aided process design: Vol. 96* (pp. 192–214).

Mattson, S. E., Elmqvist, H., & Otter, M. (1998). Pysical system modelling with Modelica. *Control Engineering Practice*, *6*(4), 501–510.

Nagl, M., & Westfechtel, B. (1999). *Integration von Entwicklungsumgebungen in Ingenieuranwendungen*. Berlin: Springer-Verlag.

Nagl, M., Westfechtel, B., & Schneider, R. (2003). Tool support for the management of design processes in chemical engineering. *Computers and Chemical Engineering*, *27*, 175–197.

Pohl, K., Weidenhaupt, K., Dömges, R., Haumer, P., Jarke, M., & Klamma, R. (1999). PRIME: towards process-integrated environments. *ACM Transactions on Software Engineering and Methodology*, *8*(4), 343–410.

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The Unified Modeling Language Reference Manual*. MA: Addison-Wesley, Reading.

Schneider, R., & Gerhards, S. (2003). WOMS—Ein Werkzeug zur Modellierung von Arbeitsabläufen. In M. Nagl & B. Westfechtel (Eds.), *Modelle, Werkzeuge und Infrastrukturen zur Unterstützung von Entwicklungsprozessen* (pp. 375–376). Weinheim: Wiley, VCH.

Schüppen, A., Trossen, D., & Wallbaum, M. (2001). Shared workspaces for collaborative engineering. *Annals of Cases on Information Technology*, *IV*(S), 119–130.

Schuler, H. (1998). *Prozessführung Chemie Ingenieur Technik*, *70*, 1249–1264.

Subrahmanian, E., Westerberg, A. W., & Podnar, G. (1991). Towards a shared computational environment for engineering design. *Lecture Notes in Computer Science*, *492*, 200–228.

Uschold, M., & Gruber, T. R. (1996). Ontologies: principles, methods and applications. *The Knoweldge Engineering Review*, *11*, 93–136.

van Gigch, J. P. (1991). *System design modeling and metamodeling*. New York: Plenum Press.

von Wedel, L., Marquardt, W. (2000). ROME: A repository to support the integration of models over the lifecycle of model-based engineering. In S. Pierucci (Ed.), *European Symposium on Computer Aided Process Engineering* (Vol. 10, pp. 535–540), Elsevier.

von Wedel, L., Marquardt, W. (2001). Cheops: A case study in component-based process simulation. In: M.F. Malone, J.A. Trainham, B. Carnahan (Eds.), *Foundations of computer-aided process design* (Vol. 96, pp. 494–497), AIChE Symposium Series 323.

Wasserman, A. (1990). Tool integration in software engineering environments. In F. Long (Ed.), *Proceedings of the International Workshop on envrionments LNCS 467 of software engineering environments* (pp. 137–149). Berlin: Springer.

Westerberg, A. W., Subrahmanian, E., Reich, Y., Konda, S., & the n-dim group. (1997). Designing the process design process. *Computers and Chemical Engineering*, *21*(Suppl), S1–S19.

Wiederhold, G., & Genesereth, M. (1997). The conceptual basis for mediation service. *IEEE Expert, September–October*, 38–47.

W3C. Extensible Markup Language (XML). Online available from http://www.w3.org/XML/.

Yang, X., & McGreavy, C. (1996). Requirements for sharing data in the life cycle of process plants. *Computers and Chemical Engineering*, *20*(Suppl), S363–S368.