

A PROCESS DATA WAREHOUSE FOR TRACING AND REUSE OF ENGINEERING DESIGN PROCESSES

Sebastian C. Brandt

Informatik V, RWTH Aachen University, Ahornstraße 55, 52056 Aachen, Germany

Marcus Schlüter

Fraunhofer FIT, Schloss Birlinghoven, 53754 Sankt Augustin, Germany

Matthias Jarke

Informatik V, RWTH Aachen University and Fraunhofer FIT, Sankt Augustin

Correspondence Email: sbrandt@cs.rwth-aachen.de

ABSTRACT

The design and development processes of complex technical systems are of crucial importance to the competitiveness of an enterprise. These processes are characterised by high creativity and strong non-deterministic dynamics. Traditional information science methods, however, are intended for more deterministic work processes. They cannot be effectively applied to support creative activities like conceptual synthesis, analysis, and decision-making. Therefore methods of experience management need to be exploited here. This paper presents a new integrated approach to such design process guidance based on capturing the process traces in a *Process Data Warehouse* (PDW). Both the products to be designed and the process steps corresponding, are structured and stored as extended method traces. This trace capture facilitates the processing and subsequent reuse of the information through a process-integrated development environment. The concept of the PDW has been evaluated in an engineering design case study which focuses on the phase of basic engineering in designing a chemical production plant.

Keywords: Decision Support Systems, Knowledge Management, Software Engineering

1. INTRODUCTION

In designing and developing complex technical systems, the design engineers are confronted with the high reactivity of the technical structures to be designed, and the many degrees of freedom in designing such structures. The complexity of their tasks results from the use of utilities like different design and simulation tools, and the planning and distribution of resources, e.g. personnel. In such a creative engineering process, any restriction of the experts in their options would greatly affect their productivity and might even lead to project failure. Today, these tasks frequently involve experts from several disparate disciplines, e.g. computer science, electrical and mechanical engineering, and control engineering. Their complex and *non-deterministic* work processes need to be supported by thorough and integrated information science systems. Presently, the possibilities of such support are widely being researched in computer science.

Solutions have been established so far mainly for *deterministic* work processes. They are often found in the domains of business, economics or management. They cannot be directly transferred due to the non-determinism found in complex technical design processes. These design processes involve the elicitation of complex requirements, their analysis and later realisation. Furthermore they involve the intricate interplay of *synthesising* the requirements or solution possibilities, *analysing* the alternative models by simulations and other methods, and *deciding* on alternatives by discussing the analysis results. The software environments currently found in these areas are usually very heterogeneous. Each tool has been developed for a separate step of these processes, yet these software tools are seldom able to cooperate. While they are able to support the evolution of *product* data, only a few tools also comprise *process* support. It is important not to separate the products or artefacts created as part of these design processes – documents, diagrams and other products – from the processes themselves. Any kind of integrated support needs to take both aspects of product and process into account.

On this background, the aim of the research presented here is to improve the work situation of the system developers participating in these design and development processes. Hence the authors of this paper have developed the concept of the *Process Data Warehouse* to allow the tracing and reuse of complex engineering design processes.

The paper is organised as follows. The next section will investigate the possibilities of supporting technical design processes by experience management. In section three, the concept of the Process Data Warehouse for trace capture and experience reuse will be explained in more detail. Section four will illustrate how these concepts have been applied in a case study in the domain of chemical engineering. Finally, the conclusion is provided including further on-going and future research.

2. SUPPORT FOR DYNAMIC DESIGN PROCESSES

The inherent dynamics of the work processes pose one of the main problems in engineering design and development. In these processes, the requirements and other parameters change from one project to the next, and can also evolve during the lifetime of a single project. As no methods in the sense of 'best practice' are known, the driving influence is the personal experience of the system designers. Transferring this tacit knowledge from one expert to another is often needed. This transfer normally requires a long-term process which prominently consists of more or less successful trials and errors.

A common approach to knowledge management is based on offering information to the expert which has been recorded in earlier executions of the same or a similar task. In this way, each expert can use his own experience and that of his colleagues to improve both his work situation and the quality of his work, while also enhancing his autonomy. This information allows the construction of some kind of best practice rules by analysing the completed steps.

For different steps or phases of a design project, different specialised software tools need to be employed. A similar situation arises if the project needs the synergetic cooperation of experts from different domains. Each of these tools is normally based on its own proprietary model and contains only some generic import and/or export functions. Possibilities of converting or integrating the data directly between the different internal tool models rarely exist. Their hard-wired usage processes are enacted independently of each other.

Because of these problems, it would be helpful to offer to the experts fine-grained support for these non-deterministic processes. For some time it has been known that experience and understanding of one's own work is necessary to enable process evolution and improvement ([7]). This insight has resulted in several approaches based on the basic concept of experience reuse. For example, the designers of the TAME project (see [3]) propose a process model for supporting creative software development processes which is based on their own experiences in software requirements engineering. Their approach focuses strongly on quantitative and metrics-based method evaluation for the later steps of software engineering. In the *Experience Factory* approach, an independent logical organisation is responsible for gathering the knowledge and core competencies of a development group and offering this information for reuse ([1]).

Some other research approaches exist in the area of engineering design processes. In [6], a knowledge-based approach for product design is examined, based on integrating partial domain models and using patterns to represent the non-deterministic behaviour of design processes. Another project is developing a process platform which supports the experience-based management and reuse of coarse-grained aspects of software development processes ([12]). A different approach to reuse the experience of product development is shown in [17]. This case stresses the manual processing of expert knowledge and its reuse by less experienced colleagues, by storing the knowledge and the contact information of experts who know more about it, in a corporation-wide experience portal.

3. DATA WAREHOUSES FOR PROCESS TRACING AND DECISION SUPPORT

3.1 Process Tracing and Decision Logging

Beyond the approaches described so far, there is the need of *integrated* technical support for the early phases of creative system design, specifically concerning complex technical systems. Therefore, the authors' research group has examined the possibilities offered by recording and reusing the traces of work *processes* in technical design. The projects have been performed as part of the DFG-funded Research Centre on Cooperative Computer-Aided Process Engineering (SFB 476 IMPROVE) which started in 1997. The research deals with the product-based view as well as the concepts of direct process support, already investigated in previous projects. It has led to supporting creative design and development processes by integrated method guidance (see [14]). These views have been extended and adapted to the domain of process engineering. This strategy resulted in the model of the *Process Data Warehouse* (PDW). It captures and analyses the traces of design processes: products, process instantiations and their interdependencies (see [8]). The main concept has been to capture the artefacts (the technical system) to be designed and modified during the processes, and to relate them to the processes which perform these modifications. From these semantically structured product and process traces, the relevant information can be extracted in an *analysis* step, and then *reused* in further process executions. This information can be presented to the experts as experience knowledge in order to solve the problems of later development cycles more easily, efficiently, and autonomously.

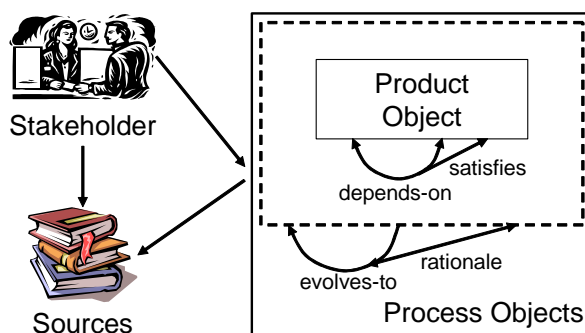


Figure 1: Traceability reference model from [15]

The central problem of this approach is that of supporting traceability. To enable traceability, first of all the conceptual relations between products, processes and their dependencies need to be examined. Therefore, in [15] the *traceability reference model* shown in Fig. 1 was abstracted from a large number of industrial case studies. This model distinguishes between product-oriented and process-oriented trace objects.

The *product-oriented* traces describe the properties and relationships of concrete design objects. A high-level object defines some goal or constraint that needs to be *satisfied* by a number of *product objects* on a more fine-grained level of modelling. This implies dependencies (*depends-on*) between these lower-level objects which also comprise the special cases of generalisation and aggregation.

The *process-oriented* traces represent the history of actions that led to the creation of the product objects. Two link types exist between those process objects: *evolves-to* which describes the temporal evolution of a lower-level design object towards a higher level, and *rationale* which captures the reason for this evolution. The integrated presentation of the product and process traces in this 'onion-shell' meta model symbolises the fact that they cannot be reasonably separated as one strongly depends on the other.

As visible in the left part of Fig. 1, the role of the *stakeholder* during product creation or documentation is of importance as well. It is also necessary to record and connect the *sources* which contain and display the information.

The description of this reference model shows that recording the process traces needs to include all related influence factors, like the actual problem situation, the resulting artefacts, and the decisions that led to the final results.

From these traces the semantically relevant information can be extracted in an *analysis* step. Due to the complexity of the traces, automated analysis is impossible in most cases. When working on complex processes with only few repetitions and few concrete product instances, this analysis step can often be left out. The decision between the available information can be done in the moment of reuse. If there are too many data to be retraced this way, a so-called *method engineer* is responsible for extracting and explicitly modelling method fragments and situations, often supported by methods of data mining.

When an expert needs to solve a certain problem, the current process and product situation is analysed by the PDW to find matching solutions from the recorded (and analysed) traces. If an adequate method or product fragment is found, it can be offered to the expert for *reuse* through a *guidance* mechanism. Yet it is his own decision whether to adapt and use this information, to request more details, or to discard it. In many cases a small hint should suffice that the step currently enacted, conforms to the experience gathered up to now or, even more important, conflicts with it.

It also has to be recorded whether the problem was successfully solved, and how far the process support provided was appropriate, as a final *feedback* information. By using this information the system and the support it offers, can be evaluated and improved.

3.2 The Process Data Warehouse

The Process Data Warehouse (PDW) has been defined as a knowledge-based metadata repository for tracing and driving heterogeneous work processes in engineering. It stores the history of processes and products to enable experience reuse with the aid of situated process support.

Based on the well-known concept of Data Warehousing (see [9]), the PDW has been developed to incorporate the *process* aspect. Beyond this integration of product and process data, the necessary flexibility has been achieved by building the system on top of loosely connected *partial models*. Thus the conceptual model of the PDW is composed of a number of models which are held together by a central model, the so-called *Core Ontology*.

The recording of *product* information is normally realised inside the tools contained in a development environment. Special steps need to be taken for recording and relating *process* traces and their supplementary information like arguments and decisions. It is necessary to enrich the application environment with tools to enable a consistent automated and partially manual trace capture. Over the last decade, the authors' research group has developed the process-integrated development environment PRIME which implements the vision of *integrating* development tools, product data, process guidance and trace capture. The concept of process integration was derived from the idea of process-centred environments (PCE, see [4]) which are based on the three domains of modelling, enactment and performance. More information about this topic can be found in [14].

The PDW forms a central part of this process-integrated environment. Thus, it records the development traces based on the concepts offered by the partial models for processes, products and application domains. In this way the information from diverse tools, workplaces and disciplines is integrated into one central storage. Other external data sources and repositories are indirectly integrated, with the PDW taking the role of a semantic data mediator. The models are based on ontological concepts and modern ontology- and logic-based languages. Thus it is possible to use semantic search mechanisms on this storage. This enables the retrieval of objects and their attributes based on relationships, attribute values, object inheritance, and other constraints. The current process state, the situation of the selected products and the attributes of other objects in their vicinity are the parameters to those queries. Thus experience traces can be found and then reused.

3.3 The Conceptual Framework of the Process Data Warehouse

As already mentioned, the conceptual model of the Process Data Warehouse is based on a set of loosely connected partial models. These are interconnected through the *Core Ontology* which will be introduced in this paragraph. It comprises the process models, the product and dependency models, models for decision support and documentation, for the description of content and categorisations, and other integration models. Around these fundamental and domain-independent models, *extension points* are placed that can be used to add the models of a specific application domain or other specialisations. These models are explicitly implemented in a modelling language similar to the realisation of ontologies as part of the semantic web effort (RDF, OWL). The concrete data is then stored as instances of the appropriate concepts. This allows modifications and extensions of the partial models used, even during the runtime of the PDW.

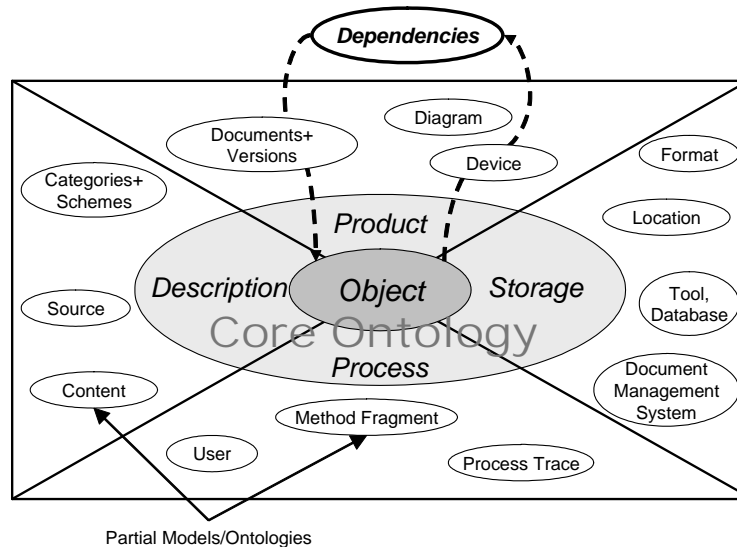


Figure 2: Overview of the four modelling areas

Four prominent areas of conceptualisation are arranged around the *object* as the abstract central concept. They are shown in Fig. 2:

- The *product* area (top) contains basic models for the artefacts created or modified during the design processes – documents, document versions and their structural composition.
- The *descriptive* area (left) contains basic concepts for describing the content or role of documents and products on a high semantic level. This includes content descriptions, sources, and categorisations which are grouped into categorisation schemes.
- The *process* area (bottom) contains the concepts needed to describe the process steps which modify the artefacts. This comprises process definitions which can be enacted (method fragments), process traces resulting from enactment, and users which guide the enactment.
- In the *storage* area (right), external stores and repositories are integrated into the PDW. This applies to document management systems, databases, external tools, and others.

Dependencies have been introduced as a global concept to enable specialised relations between elements independent of their concrete relationships. They are also described in the traceability reference model in section 3.1. This hierarchy of relationship types is modelled inside an additional area which can be seen as orthogonal to the four areas.

All areas offer the *extension points* which have been mentioned above. Here, the partial models of an application domain or other specialisations can be added. For example, the storage area offers the basic models for file storage inside a document management system, relating file-based documents with their conceptual representation inside the Process Data Warehouse. This allows

accessing the documents' contents and their physical storage places, including the visualisation or modification inside appropriate tools. To apply these concepts, an integration of the EMC Documentum® system ([5]) is being developed by extending the concepts, and implementing specialised functionality.

So far, the Process Data Warehouse has been described as it has been developed by the authors' research group. In the following section, this approach is applied to the process of designing a chemical production plant.

4. A CASE STUDY IN CHEMICAL ENGINEERING

4.1 Designing a Chemical Plant

The sample scenario described in this section is a simplified version of designing a plant for polyamide 6 production. The scenario was developed in an industrial workshop with several German partners (e.g. Bayer AG). The elaboration of the cooperative scenario, the requirements for the productions processes and the product itself, are described in [13].

At the beginning of plant design, various constraints are written down concerning the chemical process itself. For example, the residue for some of the input components must not rise above a certain limit, or the finished and cooled product must have a certain size. The complete recycling of any catalyst or input component is required. During the phase of basic engineering, an initial process concept is stepwise refined into more detailed functional elements, the so-called 'process steps' and 'unit operations'.

4.2 The Process Data Warehouse for Designing the Plant

In Fig. 3, the Process Data Warehouse is shown as it has been used in the application scenario of chemical plant design. A brief overview of the PDW for this specific application domain follows here.

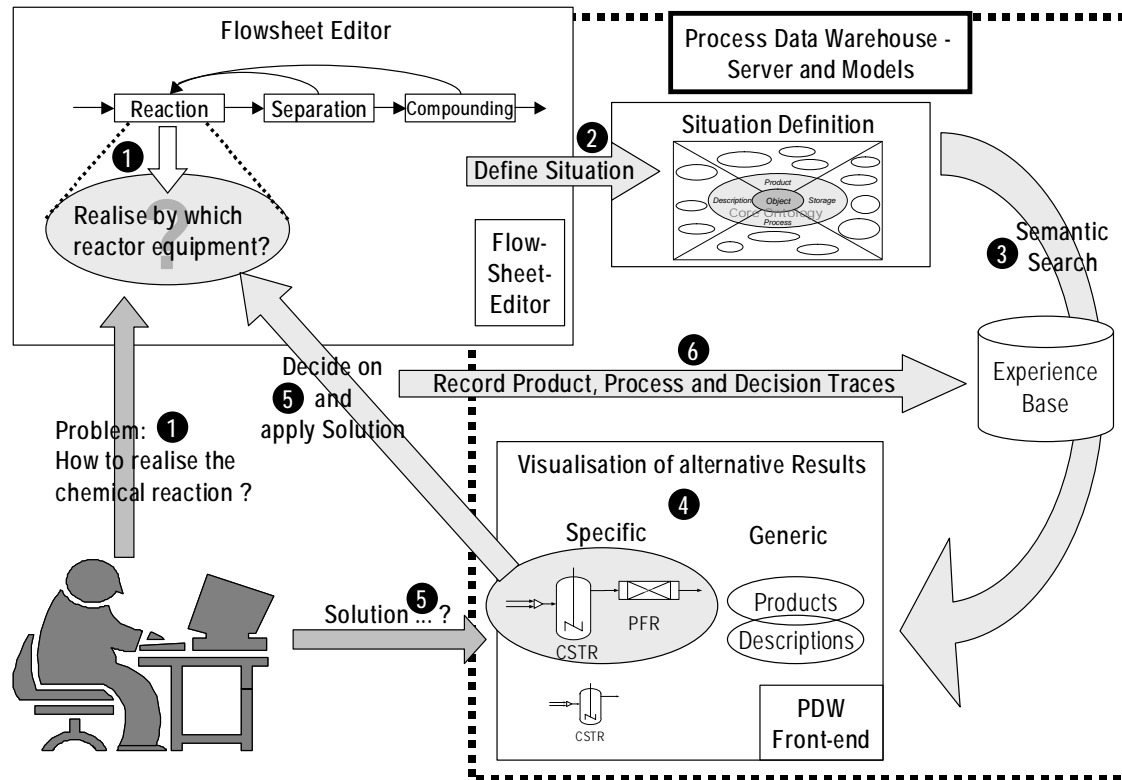


Figure 3: Process Data Warehouse – the application scenario of chemical plant design

The partial models integrated into this PDW are based on the Conceptual Life Cycle Process model CLiP (see [2]) which contains all fundamental concepts of the domain of basic (chemical)

engineering. This model is constructed out of four modelling levels, as usual in this kind of information models. The meta meta model mainly consists of the concepts System and its Aspects whereas the concrete instances created and/or recorded in the development process can be found three levels below. The role of a central connecting concept is taken by the flowsheet which forms the centre of the design process, both as a diagram and as a data model. The CLiP model has been transformed into an ontology-based version (see [18]), realised in the Ontology Web Language (OWL, see [16]). This allowed the direct integration into the PDW. The structural components of the chemical plant (like diagrams, devices and streams) have been derived from the product and document concepts. The functional aspects extend the descriptive area by adding content and source concepts, e.g. the mathematical or physical model which generated a certain simulation result, the kind of reaction it models, or the chemical components which react with each other.

As shown in Fig. 3, this case study deals with the situation where a certain process step, the *Reaction*, is to be realised by one or more different reactor equipments. This step can be seen as the starting point in Fig. 3, item (1).

The experience reuse framework consists of the Process Data Warehouse, the process-integrated development environment PRIME and the flowsheet editor which has been integrated into this environment. This framework is able to specify the current problem situation based on the integrated rules, and tries to find a matching process trace or a recurring method fragment in the experience base of the PDW (Fig. 3, item (2)).

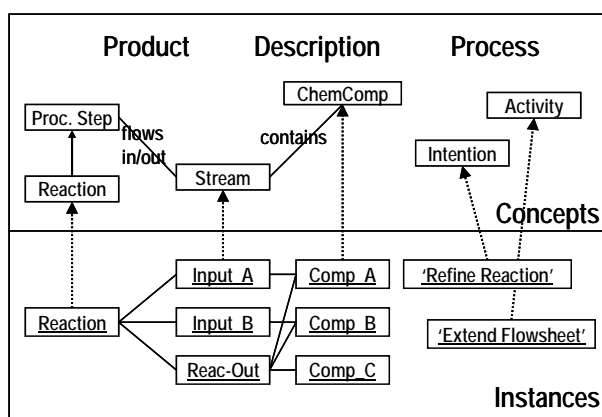


Figure 4: The instances of the situation definition and their concepts

The required problem definition consists of elements of several of the four areas introduced in section 3.3, each of the elements being the concrete instance of one of the semantic concepts contained in the PDW's ontologies. This can also be seen in Fig. 4:

- The *product* part of the situation is composed of the design elements displayed in the flowsheet editor, e.g. the reaction, the streams of chemical elements flowing into and out of it, their relationships, and attributes like temperature or pressure. Their states – e.g. being selected in the graphical user interface – also need to be taken into consideration.
- Some of the relations and dependencies reach into the *description* area, as chemical components, reaction types and other categorisations can be found here.
- The most important *process* element is the user's intention in this situation, i.e. 'Refine the selected Reaction'. This is normally determined by a user interface element being activated, e.g. a menu item being clicked. The current activity type of 'Extend the Flowsheet with Reaction Alternatives' – synthesising a model, in contrast to analysing a number of alternatives or deciding on one of them – may also be used here.

This situation definition can then be passed on to the Process Data Warehouse to search for matching experience information (item (3) in Fig. 3). In the example scenario, several different realisations are found and returned. These include using or combining two different kinds of

polymer reactors (Plug Flow Reactor (PFR), Continuous Stirred Tank Reactor (CSTR)). This information is then presented to the expert in the PDW client front-end, as visible in item (4). Two different visualisations can be applied here. A generic representation shows the concept instances, their attributes and relations in UML instance notation, while the specific representation in this case shows a graphical snapshot of the reactor equipments and their streams (see Fig. 5).

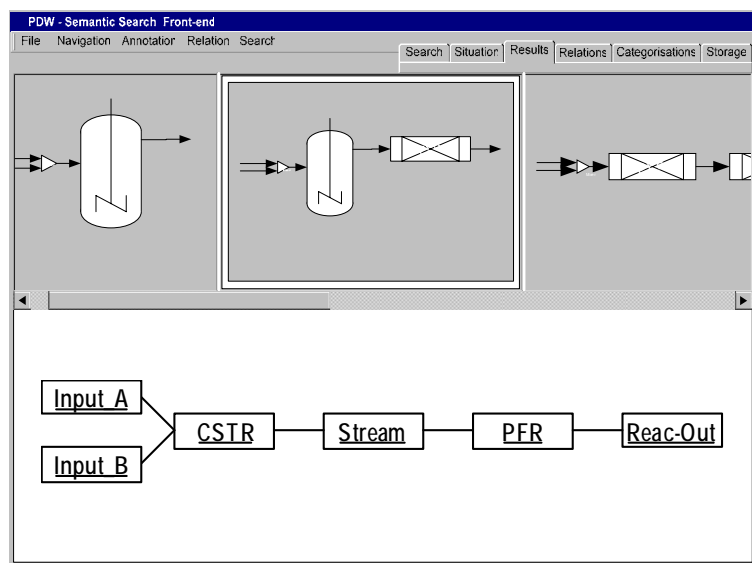


Figure 5: The specific (top) and the generic visualisations of the search results

Now the expert needs to decide which realisation to use. He may decide on reusing one of the alternatives offered. Then a method fragment inside PRIME is activated that directly executes the steps in the flowsheet editor which are needed to add the new refinement and to insert the selected alternative (see item (5)). If he wants to create a new realisation, this can be done using the normal functionality of the flowsheet editor.

In any case, the solution is recorded in the Process Data Warehouse (item (6)). Apart from the product information (the selected refinement alternative), the intermediate process steps and situations are traced and related to the decision that led to this alternative, including some additional arguments the expert may want to enter manually.

This is only a preliminary decision, as the chosen realisation still has to be analysed. The process-integrated flowsheet editor and the PDW also serve as the starting and integration points for simulation runs in the appropriate tool(s). After the simulation, the same cycle will have to be repeated for other, alternative refinements. This is also supported by the flowsheet editor mentioned above. In the end, the expert can decide – and document – which of the alternatives should be kept for further steps in the development process. This might also include a cooperative discussion. In any case, the arguments entered earlier and the simulation results related to them will be needed for supporting the decision process here. This information is only kept in the PDW.

The repeated cycles of choosing a reactor realisation, furthermore simulating the realisation, documenting the results and entering some notes or arguments, and the concluding decision-making may be recognised as a single recurring method fragment, through analysing the process traces. In this analysis phase, the method engineer may decide to add a loosely modelled method fragment into the experience base that can be activated by the expert when facing this or a similar problem situation again, to guide him or her efficiently through these tasks.

A similar product reuse solution has already been realised in the area of plastics engineering. The 3-dimensional simulation of a section of a compounding extruder has been process-integrated to support the expert in his decisions. Other aspects, like the visualisation and handling of multimedia-based simulation results and cross-organisational cooperation, have been treated there as well ([10]).

CONCLUSION

The concept described here deals with capturing product and process traces in a Process Data Warehouse (PDW). This facilitates the support of creative design processes. The concept has been applied in the case study of designing a plant for the production of polyamide 6 (nylon) by hydrolytic polymerisation. The case study was successfully concluded as part of the Cooperative Research Centre IMPROVE. Its results are presently being transferred by the authors into several other industrial application areas, as follows.

In plastics engineering, it has been found that certain kinds of *continuous production processes* for rubber profiles show characteristics similar to those of *design processes*. An operator is only able to efficiently start such a production line after having gained a lot of personal experience. For this research, these processes offer the advantage that process parameters and production quality can be automatically registered and measured quantitatively. Using the concepts and models of the PDW described in this paper, a system for supporting production operators driving such a production line is being installed. It is being evaluated at the site of the cooperation partner, Meteor rubber productions ([11]). Apart from improving the work situation, the aim is to reduce the time spent in producing rejects, thus saving money and reducing environmental impact.

A further project is being initiated to transfer the often vast amounts of printed documentation resulting from the design and construction of a chemical plant, into digitised form. After scanning and text recognition, document mining tools are applied to visualise and structure the information. The Process Data Warehouse is used to semantically enrich and categorise these documents and their content. The resulting information repository can be used to enhance document access during plant operation, repair and modification.

Two other research areas are being pursued based on applying this experience-based approach. The application scenario will be extended to include cross-organisational cooperation processes, positioned in the intersection of chemical engineering and plastics engineering design. As the protection of intellectual property is of utmost importance in any industrial application domain, a major topic is controlling the selective transfer of information following the principle of need-to-know.

Finally, in the domain of chemical engineering, the workflows of the full plant life cycle are going to be examined, from the design phase through plant operation, modifications and redesign or reengineering. The 'freezing' and later restarting or continuing of design processes is also to be researched.

ACKNOWLEDGEMENT

This work was supported by the German National Science Foundation (Deutsche Forschungsgemeinschaft, DFG) within the Collaborative Research Centre CRC/SFB 476 'Informatics Support for Cooperative Chemical Engineering – IMPROVE'. Thanks are due to our colleagues from the project, and to our students who implemented the Process Data Warehouse and PRIME.

REFERENCES

- [1] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. The Experience Factory. In John J. Marciniak (ed), *Encyclopedia of Software Engineering, Volume 1*, pp. 469-476, John Wiley & Sons, 1994
- [2] Birgit Bayer and Wolfgang Marquardt. Towards Integrated Information Models for Data and Documents. *Computers and Chemical Engineering*, 28:1249-1266, 2004.
- [3] Victor R. Basili and H. Dieter Rombach. The TAME project: Towards improvement-oriented software environments. *IEEE Trans. Software Eng.*, 146:758-773, 1988.

- [4] Mark Dowson and Christer Fernström. Towards Requirements for Enactment Mechanisms. In *EWSPT*, pp. 90-106, 1994.
- [5] Documentum. Enterprise Content Management, ©EMC Corporation. <http://www.documentum.com/>, access date: Feb. 2005.
- [6] Hans Grabowski, Ralf Lossack, and Marc Leutsch. A Design Process Model. In *Proc. of the 3rd International Workshop on Strategic Knowledge and Concept Formation*, Sydney, Australia, 2001.
- [7] Watts S. Humphrey. *Managing the Software Process*. Addison-Wesley, 1990.
- [8] Matthias Jarke, Thomas List, and Jörg Köller. The Challenge of Process Data Warehousing. In *Proceedings of the 26th International Conference on Very Large Databases – VLDB*, Cairo, Egypt, 2000.
- [9] Matthias Jarke, Maurizio Lenzerini, Yannis Vassiliou, and Panos Vassiliadis. *Fundamentals of Data Warehouses*. Springer-Verlag, 1999.
- [10] Matthias Jarke, Michalis Miatidis, Marcus Schlüter, and Sebastian Brandt. Media-Assisted Product and Process Traceability in Supply Chain Engineering. In *37th Hawaii International Conference on System Sciences – HICSS*, Big Island, HI, USA, January 2004.
- [11] Meteor Gummiwerke K.H. Bädje GmbH&Co.KG, 31167 Bockenem, Germany, <http://www.meteor.de>, access date: May 2005
- [12] Jürgen Münch and Dieter Rombach. *Modelle, Werkzeuge und Infrastrukturen zur Unterstützung von Entwicklungsprozessen*, chapter Eine Prozessplattform zur erfahrungsbasierten Softwareentwicklung, pp. 93-106. Wiley-VCH, 2003.
- [13] Manfred Nagl, Bernhard Westfechtel, and Ralph Schneider. Tool Support for the Management of Design Processes in Chemical Engineering. *Computers and Chemical Engineering*, 27(2):175-197, 2003.
- [14] Klaus Pohl, Klaus Weidenhaupt, Ralf Dömges, Peter Haumer, Matthias Jarke, and Ralf Klamma. PRIME: Towards Process-Integrated Environments. *ACM Transactions on Software Engineering and Methodology*, 8(4):343-410, 1999.
- [15] Balasubramaniam Ramesh and Matthias Jarke. Toward Reference Models for Requirements Traceability. *IEEE Transactions on Software Engineering*, 27(1):58-93, January 2001.
- [16] D.L. McGuinness and F. van Harmelen. *Web Ontology Language (OWL) Overview*. <http://www.w3.org/TR/owl-features/>, W3C Recommendation, Feb 2004.
- [17] Kristina Wagner and Stephanie Aslanidis. Prozessorientierte Nutzung von Erfahrungswissen in den frühen Phasen der Produktentstehung. In *4th Conference on Application of Knowledge Management in Industry and Public Administrations – Knowtech*, München, Germany, 2002.
- [18] Aidong Yang and Wolfgang Marquardt. An Ontology-based Approach to Conceptual Process Modelling. In A. Barbarosa-Póvoa, H. Matos (Eds.), *European Symposium on Computer Aided Process Engineering – 14*, Elsevier, 2004, pp. 1159-1164