

Information modelling using product life cycle views

R. Anderl, B. Daum, H. John, C. Pütter
Fachgebiet Datenverarbeitung in der Konstruktion
Darmstadt University of Technology
Petersenstraße 30, D-64287 Darmstadt
Tel. +49 (0)6151/166001, Fax +49 (0)6151/166854
{anderl,daum,john,puetter}@dik.tu-darmstadt.de

Abstract

Industrial enterprises are increasingly forced to design their products for the environment. As environmental effects result from processes throughout the product life cycle, product and process information must be integrated in a conceptual model to realise systems of information technology that support product design and manufacturing. This paper presents a new method to derive an object-oriented information model from a life cycle oriented modelling approach. The information model consisting of a product data model and several partial models representing life cycle phases can be implemented easily, e.g. as an object-oriented database, which is a basis for life cycle impact assessment and its integration with a design system environment.

Keywords

Information modelling, process modelling, object-oriented modelling, design system environment, environmentally sound product

1 INTRODUCTION

Industrial enterprises are increasingly forced to develop and manufacture environmentally sound products. To meet this requirement it is necessary to analyse processes in all life cycle phases regarding environmental impacts. Important is the relationship of these process effects to product properties in order to optimize products and production during development. A powerful software support for this optimization can be developed if a formal information model is created that represents the process information and its relationship to a product data model as developed in ISO 10303 (STEP).

On the one hand, for an efficient implementation of IT Systems an object-oriented information model is required. On the other hand, the knowledge resulting from analysis of production processes can be represented more easily in a process oriented manner. As behaviour description in object-oriented modelling languages is complex, simple process modelling languages would be preferred, arising the need for supporting a transformation to a modelling language that is computer accessible.

As production processes within a particular enterprise are supplied with raw materials and semi-products and followed by product use, recycling and disposal, the analysis must be extended over all participating enterprise departments and their suppliers. Therefore a support for co-operation during information model development is necessary.

Contemporary methods of information modelling do not offer a special support for transforming process models into object-oriented models as a basis for IT systems. A new information modelling technique that includes such a transformation has been developed within the research project "SFB 392: Development of Environmentally Sound Products"¹ at Darmstadt University of Technology. This approach of an interdisciplinary group of scientists develops a design system environment that includes an holistic assessment of product and process alternatives depending on environmental but also technical and economical properties. An important task within the research project is the analysis of processes in product life cycle and the representation of this knowledge in a formal information model. Core of this model is a product data model based on ISO 10303. Several partial models represent views of particular product life cycle phases.

The following chapter will discuss the need for process oriented modelling techniques despite an object-oriented implementation. Known approaches to transform process models towards implementation will be analysed. The

¹granted by the German research association (DFG)

developed modelling technique and its tool support will be described with focus on the transformation process.

2 COMPARISON TO OTHER APPROACHES

For the development of IT systems supporting design for environment, process modelling can be used in two ways: For analysis of the product development process in order to define the tools supporting the development of environmentally sound products, or for identification of processes from life cycle phases with relevant impact on environment. Especially the first case is comparable to ISO 10303 Application Protocol Development: First of all, application and modelling experts describe the application in terms of processes to define the scope and to identify the information flows that must be supported by data exchange, in an activity model using the process modelling language IDEF0.

The information flows are detailed in the reference model using the more object-oriented description language EXPRESS-G. A transition of activity model to reference model is based on a clustering of information flows in the activity model. The reference model is structured in *Units of Functionality* each describing the information requirements of a restricted group of application functions or processes in the activity model (Mohrmann, Katzenmaier, 1997).

As a basis for the implementation of data exchange processors, the information requirements specified in the reference model are mapped to suitable constructs of the application independent *integrated resources* of ISO 10303 to ensure consistency between multiple application protocols (Anderl, Wasmer, 1996). This task is not comparable to the integration of partial models with a core model in the research project described - whereas an application protocol uses constructs of the integrated resources of ISO 10303 in a special application context, the partial models detail and reference core model constructs to specify environmental impacts in all life cycle phases.

In the area of business process reengineering an integration of process and product models is performed by various methods, but without any transformation of the process model (Kosanke, 1996).

A quite different approach transforming process models directly to object-oriented structures has been implemented for IDEF0 to EXPRESS in the software tool *XP-IDEF* by *CSTB*. It is based on a reference schema for the process modelling language and can be used to produce a neutral STEP version of IDEF0 models or to transform them into different reasoning frameworks (CSTB, 1998).

Of course, process models themselves can be specified formally without any transformation, e.g. using a description language with dynamic constructs as UML (Booch, Rumbaugh, 1997) or EXPRESS-C (Staub et al., 1994). But this object oriented information modelling not only requires understanding of its philosophy, but also programming experience.

To support an analytic approach and active participation of non-modelling experts in an information modelling project, the use of an informal modelling language is advantageous. In later phases, the information model must be formalised in order to become unambiguous and computer accessible supporting its fast implementation. As a result, an information model - in particular for product and environmental information - should be formalised step by step.

3 METHOD FOR TRANSFORMATION OF PROCESS MODELS

As an example for the transformation of process models in order to create an information model for environmentally sound products, the IDEF0 language is chosen as a process modelling language. The object-oriented language to be transformed has been developed based on EXPRESS-G in order to achieve a fully graphical representation, easier to use as UML but with more capabilities than EXPRESS (Anderl et al, 1997). This approach facilitates reuse of ISO 10303 models, but also considers the requirement to represent all environmental product information, e.g. ecological assessment data based on functional and rule-based dependencies.

Concerning the process modelling language, IDEF0 has some advantages against others such as an easy structure and a support for analysis of processes by detailing them. The transition concept of *Petri nets* as another kind of process modelling rather aims at simulation of processes, not analysis and identification of process properties. In the area of life cycle assessment, a framework based on IDEF0 has already been defined (Tipnis, 1995). The usage of IDEF0 within the research project SFB 392 for the identification of environmental impacts and parameters with significant influence on environmental product and process properties is shown in figure 1.

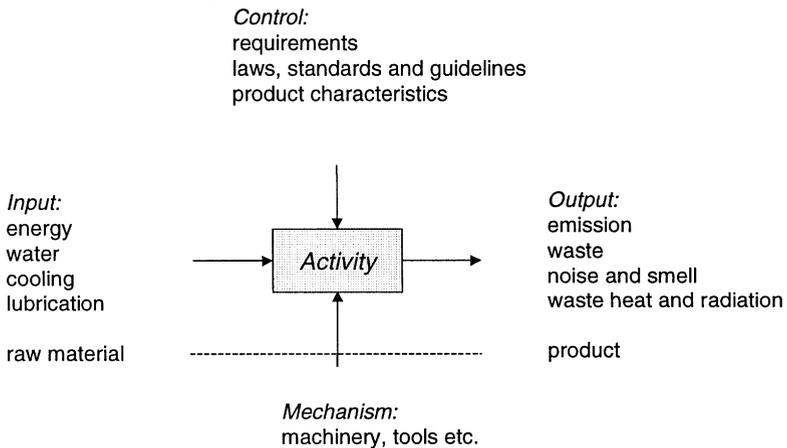


Figure 1: Usage of IDEF0 for environmental process analysis

As the main difference between mechanism and input, it has been defined that mechanisms are nearly independent from processing a particular product. Both inputs and mechanisms can result from other process chains not included in the process model - the distinction between various levels of process chains is important for defining the scope of life cycle inventory calculation.

The transformation of such IDEF0 process models into object-oriented models needs to be realized by an incremental transformation technique. The steps of this technique are shown in figure 2.

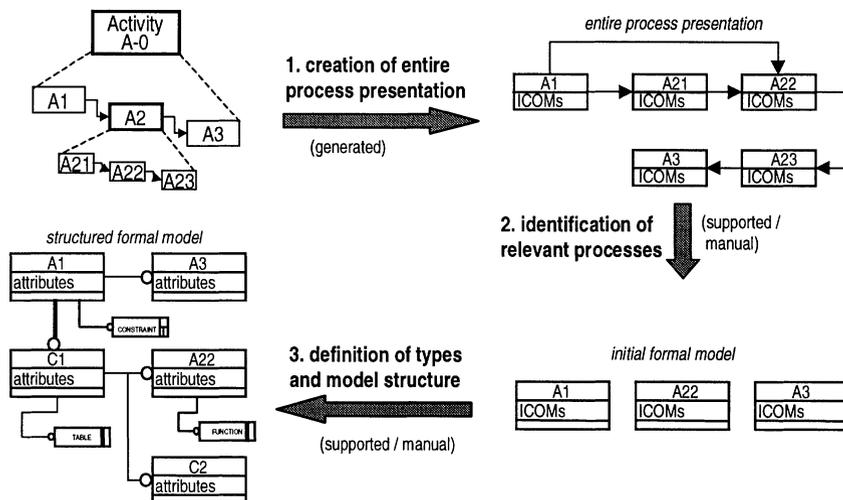


Figure 2: Transformation of life cycle models into product data models

The main aspect is, that activities are grouped to classes and their ICOMs (input, output, control and mechanism) become static or dynamic attributes of these classes. A reverse approach to create classes from the ICOMs and to represent the activities by dynamic constructs did not prove useful in the given context.

The technique is based on the assumption that the lowest level activities of an IDEF0 model contain all aspects of the process. These activities are used for a coherent representation of the real world process. From this representation, an initial class diagram is developed which contains classes representing activities relevant for the given context. ICOMs from the process model start as attributes without types. In the following step, the initial class diagram is transformed into a structured model. All classes have to be checked for aggregation or inheritance relationships between them, and correct types of class attributes must be identified distinguishing the following:

1. The class attribute represents a single atomic value or a set of atomic values. In this case, the attribute has to be defined as a *Simple Type* (e.g. REAL, FUZZY_SET etc.).
2. The class attribute has a structure important for the given context. Then the attribute is deleted and a new class has to be created containing the relevant properties.
3. The attribute belongs to a different class. It is deleted from this class and defined in another one.
4. The value of the attribute can be calculated by a function or a rule. The attribute is deleted, and it must be determined which dynamic construct is used to represent it.
5. The attribute is unimportant and can be deleted without any other actions.

To support processing of case 4, the modelling language developed on the basis of EXPRESS-G includes dynamic constructs. These extensions to EXPRESS-G can be divided into constructs for representing functions, rules, constraints and informal dependencies, but use a common logical structure for modelling and follow object-oriented principles.

Mathematical functions are represented either as a *function* or a *table* construct. A *function* consists of an arithmetical combination of numerical or string attributes. In the context of environmental information, there are a lot of relationships between product and process parameters, which cannot be expressed by a simple formula. To represent such relationships (i.e. measuring data), the *table* construct is used. A Table maps multiple columns - probably containing numerical attribute values as well as alphanumerical - to a result in the last column.

For representing rules, constructs for condition and implication are defined and can be combined by Boolean operators. The implication contains a list of statements which change the attribute values of a class.

Constraints can also be represented graphically to specify - for example - restrictions of attribute values for instantiation or modification of objects in a database. Objects of a class are valid, if all *constraints* are satisfied.

To represent information hard or senseless to formalise, the element *document* is defined. A *document* references class attributes where external (possibly multimedia) documents e.g. describing guidelines for product design are assigned to.

The resulting model can be further processed by representing additional static and dynamic aspects. Then an integration with an existing product data model can be performed by links between the partial and core models. These links are defined as supertype or attribute relations or as references for use of class attributes in dynamic constructs.

4 IMPLEMENTATION OF MODEL TRANSFORMATION

The method to transform process models into object-oriented models and its implementation in software systems is realised with a suitable modelling tool developed within the research project. It supports the entire model development as a basis for providing transitions between the modelling steps (figure 3).

To support a co-operation between partial model developers, all modelling steps can be automatically documented as HTML including image maps of the graphical model representations.

Special functions have been implemented for deriving a consistent process chain out of an IDEF0 process model. To solve the hierarchy of IDEF0, any process is decomposed into its sub-activities until an elementary activity is found. Then a class is generated for that activity and all ICOMs of this process are added as attributes automatically. By browsing through all levels, the order and the flows between elementary activities are determined, so an entire representation of the process model without redundancies can be created.

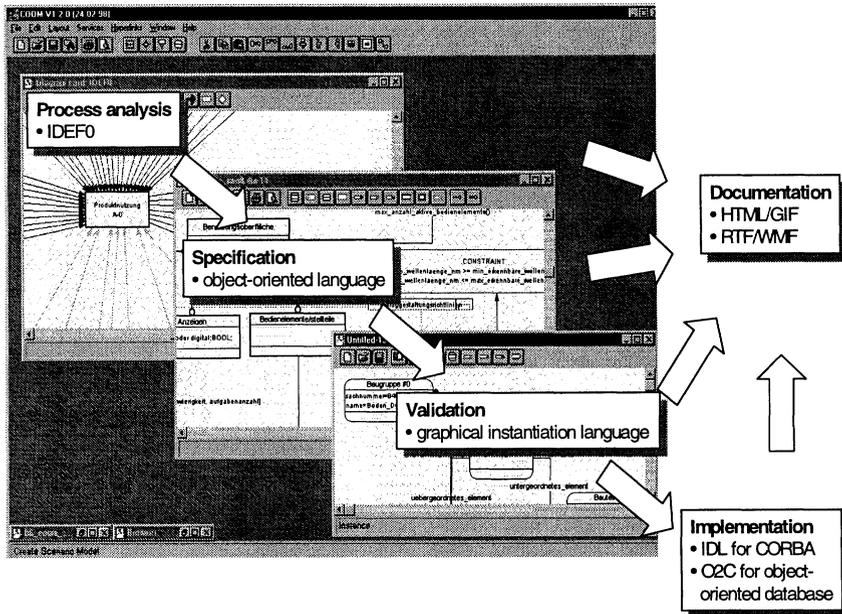


Figure 3: Tool support for modelling process

Another tool feature important for modifying the initial class diagram is a fast redefinition of element types to replace simple class attributes efficiently by more sophisticated constructs such as functions, rules, constraints or classes as described.

Validation of developed partial models is performed using a graphical instantiation language. This language provides constructs to instantiate classes, their attributes and relations.

Then dynamic elements derived from attributes within the object-oriented model must be calculated and visualised by functions of the modelling tool.

Any partial model can be integrated with a product data model by importing it into the partial model as a *read only* component. Finally, all partial models developed to represent product life cycle phases are combined on the basis of the product data model. A check for redundancy between the partial models in the entire information model can be performed with some tool support. Concerning the model contents, an analysis by a modelling expert in co-operation with the partial model developers is required to ensure a consistent information model.

The modelling tool also includes a compiler to create a database schema directly from the information model. The schema clearly corresponds to the information model which simplifies initial database instantiation as well as keeping the information up-to-date. Another important aspect is the derivation of *Interface Definition Language (IDL)* from the information model to specify the interfaces between database and application system.

5 IMPLEMENTATION OF INFORMATION MODEL

The object-oriented approach for the information model including dynamic properties of a product and its processes enables a fast implementation in an object-oriented database system. The database is the main component for the design system environment currently being developed in the research project. Its system architecture is based on the *Common Object Request Broker Architecture CORBA* (OMG, 1996). Additional systems required for the development of environmentally sound products are integrated or implemented respectively using the interface description specified in *IDL*:

A parametric 3D-CAD system is used to instantiate the core model part of the information model with product data. A so-called life cycle modeller is used to select processes represented by partial models, if the process chain cannot be derived directly from product properties (Schott et al., 1997). This life cycle modeller includes components for each life cycle phase that are used to add product specific process information during product development, as well as enterprise specific information during system configuration. Direct interfaces to the database are used to instantiate all data which are independent from a specific product or enterprise. Based on this information provided by CAD and the life cycle modeller or instantiated initially, an assessment system for environmental and also technical and economical properties of products accesses the assessment data calculated in the partial models and performs a life cycle inventory and impact assessment that is presented to product developers in a suitable manner for optimising the product's environmental properties.

6 CONCLUSIONS AND FUTURE WORK

The presented modelling transformation simplifies the modelling process in the research project SFB 392 and facilitates a rapid prototyping of a design system. The methodology seems to be transferable to the design of other IT systems if they are based on process analysis. A transformation of knowledge represented in other forms than an IDEF0 process model is also planned in order to create object-oriented models even from informal enterprise knowledge.

To ensure completeness and correctness of the information model for environmentally sound product design, a sophisticated collaboration support during information modelling will be developed. For integrating external information sources with this information model it is planned to generate mappings automatically.

An interface for enterprise configuration and instantiation of the information model will provide future application of the described design system basing upon information model in enterprises. The first prototype of the design system is currently being implemented.

7 REFERENCES

- Anderl, R.; Wasmer, A. (1996) Integration of product life cycle views on the basis of a shared conceptual information model. Proceedings of the 2nd International Conference on The Design of Information Infrastructure Systems for Manufacturing DIISM '96. Kaatsveuhel, The Netherlands 1996.
- Anderl, R.; Daum, B.; John, H.; Pütter, C. (1997) Cooperative Product Data Modelling in Life Cycle Networks. 4th International Seminar on Life Cycle Engineering, Berlin 1997
- Booch, G.; Rumbaugh, J. (1997) Unified Method for Object-Oriented Development. Documentation Jet Version 1.0, Rational.
- CSTB (1998) XP-IDEF: <http://cic.sop.cstb.fr/ILC/products/xp-idef/xp-idef.htm>
- ISO STC1/SC7/WG11 (1997) High-level Petri-Nets-Concepts, Definitions and Graphical Notations, ISO 1997
- Kosanke, K. (1996) Comparison of enterprise modelling methodologies. Proceedings of the 2nd International Conference on The Design of Information Infrastructure Systems for Manufacturing DIISM '96. Kaatsveuhel, The Netherlands 1996
- Mohrmann, J.; Katzenmaier, J. (1997) ISO TC184/SC4/WG3/N578, Product Data Representation and Exchange Part 214 -- Application Protocol "Core Data for Automotive Mechanical Design Processes" .
- OMG (1995) The Common Object Request Broker: Architecture and Specification, Revision 2.0, OMG

- Schott, H.; Grüner, C.; Büttner, K.; Dannheim, F.; Birkhofer, H. (1997) Design for Environment - Computer Based Product and Process Development. 4th International Seminar on Life Cycle Engineering, Berlin 1997
- Staub, G. et al. (1994) EXPRESS-C Language Reference Manual. ESPRIT III Project No. 6876: PISA.
- Tipnis, V. A. (1995) Towards a comprehensive life cycle modelling for innovative strategy, systems, processes and product/services. PROLAMAT '95, Berlin.

8 BIOGRAPHY

Prof. Dr.-Ing. Reiner Anderl studied mechanical engineering at the University of Karlsruhe, where he received his doctor degree in 1984 after working as a research assistant. In 1985 he took the position as a chief engineer and received the habilitation in 1991 in Karlsruhe at the faculty of mechanical engineering. In April 1993 he received the professorship for computer integrated design at Darmstadt University of Technology. For several years he has participated in research on product development and standardisation activities on international and national level.

Dipl.-Ing. Bernd Daum studied mechanical engineering at Darmstadt University of Technology. Since January 1996 he has worked as a research assistant at the Fachgebiet Datenverarbeitung in der Konstruktion (computer integrated design). Within the research project "SFB 392: Development of Environmentally Sound Products", his subproject aims at a suitable design environment including co-operative working techniques.

Dipl.-Ing. Harald John also studied mechanical engineering at Darmstadt University of Technology and has worked as a research assistant at the Fachgebiet Datenverarbeitung in der Konstruktion (computer integrated design) since January 1996. Within the research project "SFB 392: Development of Environmentally Sound Products", he co-ordinates the development of partial information models representing the design relevant knowledge of all product life cycle phases.

Dipl.-Inform. Christian Pütter studied natural computer science at the University of Bielefeld. Since May 1996 he has worked as a research assistant at the Fachgebiet Datenverarbeitung in der Konstruktion (computer integrated design). Within the research project "SFB 392: Development of Environmentally Sound Products", he is focusing on methods and tools to support the development of an information model.