Enterprise Modelling in the Context of Manufacturing

Sprenger, Jonas; Jung, Jürgen

In: ICB Research Reports - Forschungsberichte des ICB / 2008

DOI: https://doi.org/10.17185/duepublico/47115

URN: urn:nbn:de:hbz:464-20180920-084409-0

Link: https://duepublico.uni-duisburg-essen.de/servlets/DocumentServlet?id=47115

Source: ICB-Resarch Report No. 23, Januar 2008

ICB
Institut für Informatik und Wirtschaftsinformatik

Jonas Sprenger
Jürgen Jung

23

ICB-RESEARCH REPORT

Enterprise Modelling in the Context of Manufacturing

Outline of an Approach Supporting Production Planning

UNIVERSITÄT DUISBURG ESSEN

ICB-Research Report No.23
Januar 2008

Authors' Address:

Jonas Sprenger
Jürgen Jung

Institut für Informatik und
Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen
45117 Essen
Germany

jonas.sprenger@uni-duisburg-essen.de
juergen.jung@uni-duisburg-essen.de

Managing Assistant and Contact:

Jonas Sprenger

Institut für Informatik und
Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen
Universitätsstr. 9
45141 Essen
Germany

Email: icb@uni-duisburg-essen.de

ISSN 1860-2770

Abstract

Manufacturing enterprises are faced with an increasing global competition and dynamic markets. This development results in a need for more flexible manufacturing structures and the ability for global collaboration. In the curse of this, cross-functional information systems turn into an essential competitive factor. However, the analysis and the integration of an enterprise's core functions is a major challenge. A model-centric approach, namely, multi-perspective enterprise modelling, is an appropriate conceptual instrument to overcome those gaps. Enterprise modelling is widely regarded as an appropriate tool to support analysing, designing and managing corporate information systems. In the paper at hand, we propose an approach fostering the applicability of enterprise models by offering domain-specific concepts. These modelling concepts reflect domain-specific concepts, which are well-known to manufacturing people. Hence, they are particularly suitable for supporting specific analyses, as well as information systems' development.

Contents

List of Figures

1 Motivation

Manufacturing enterprises (ME) are affected by dynamic markets and increased global competition. Such a development requires flexible manufacturing structures that support the production of customised products. At the same time, costs resulting from manufacturing processes have to be reduced (ElMa06). New manufacturing strategies promise the production of customised products, while having low cost production structures (Mass Customisation). Further emphasis is on global collaboration with customers, suppliers and even with competitors in virtual enterprise networks (Giac99; PiSK+04). In the course of this, cross-functional information systems (IS) are recognised as an essential factor in order to support and integrate functions, "such as product / process modeling, process planning, process and production control strategies and logistics" (ElMa06, p. 2).

However, the status quo of ME usually shows a multiplicity of badly integrated applications. Reasons for this are a historically grown separation of production planning and control systems (PPC) and the so called CAx[1] systems. A popular approach emphasising the integration of PPC and CAx is the Computer integrated Manufacturing (CIM). Nevertheless, the initial euphoria has been replaced by the perception that CIM is too complex, unwieldy and cost intensive (cf. Cors00, pp. 546f.).

Beyond that, manufacturing systems are usually considered from actors with different professional backgrounds, e. g., business economics and the engineering. The various actors have got specific expectations and views on manufacturing systems, as well as terminologies of their own. Such differing views and the inherent complexity of manufacturing systems have lead to a "diversity of models used in the different stages of the life cycle of a product" (SeTa97, p. 733). Expected consequences of this diversity are frictions that hamper the communication caused by synonyms and homonyms. Further problems may arise from redundant information represented in different models. A particular bill of material (BOM), for instance, might be considered from different views: The purchase department usually focuses on an analytical view (what do the products consist of?). The construction department generally has a synthetic view (in what do the products go into?). Despite obvious corresponding concepts, the BOM is usually stored and maintained separately. This redundantly stored information often causes extra work, as well as an increased error-proneness.

We assume that enterprise modelling (EM), and in particular multi-perspective enterprise modelling is an adequate tool for bridging those gaps. It provides mechanisms for handling the complexity of the overall enterprise system (Fran94). EM is widely regarded as an appropriate conceptual instrument to support analysing, designing and managing corporate information systems. It further supports the design and management of an enterprise's strategy and organisation. The transparency created by such an approach enables a strategical, tactical, and operational decision support. Examples for concrete decision scenarios are business planning, restructuring, specialisation, and subcontracting. A multi-perspective enterprise model consists of various models of an information system (e. g., class diagram, message flow diagram) and of the *action system* it serves to support (e. g. business process model). However, the description of an enterprise's processes is a challenging task, especially for people without a modelling background. Some researchers refer to conceptual limitations of EM approaches, which do not cover manufacturing-specific aspects in an appropriate manner. Especially resources and their allocation to processes are usually not considered at all (DeWW+03; ChAW07).

From an academic and an industrial point of view, providing manufacturing-specific concepts and their integration with further enterprise modelling concepts are a matter of particular interest. It is assumed that a general purpose modelling language (GPL) is not satisfactory for this purpose.

[1] *CA* stands for computer aided systems. The *X* is a place holder for further systems encompassing Design (CAD), Engineering (CAE), Planning (CAP), etc.

Examples for such GPL's are the *Unified Modelling Language* UML (RuJB99), the *Entity Relationship Model* ERM (Chen76), or *Event-driven Process Chains* EPC (KeNS92). A GPL does not provide domain-specific concepts for representing and analysing manufacturing systems. Instead, it requires for the reconstruction of common concepts by using primitive generic terms, such as "class", "attribute", or "activity" (cf. Fran99, p. 7). Such a challenging and time-consuming task can hardly be done by an unexperienced modeller, since it requires special diligence and competencies. As a consequence, industry has a critical attitude to modelling (FrRi98).

A domain specific language (DSL) can overcome these gaps by providing concepts that reconstruct the professional terminology, i. e., concepts that are analogue to the concepts of the domain experts. Domain-specific concepts increase the comprehensibility and the clearness of models. It further contributes to the documentation and reuse of knowledge, thereby fostering modelling productivity and integrity of models. Domain-specific concepts may foster the acceptance and application of such a language (GrPR06; Kirc07). Re-use of such predefined concepts might pinpoint possible sources of modelling errors by a well-defined syntax and semantics, which increases the quality of models. Providing an adequate graphical notation improves the intelligibility of models. Analysing models and transformations to more sophisticated representations can be supported (e. g., process plans, scheduling models). In conclusion, the design of a DSL demands to consider the specific requirements and terminology of a domain. A major challenge that usually force the researcher to consider the industrial grasps, as well as different scientific research fields.

The report at hand outlines a domain-specific extension to a given enterprise modelling method supporting the analysis, design and management of manufacturing processes. The remainder of the report is structured as follows: A major characterisation of *enterprise modelling* and *manufacturing system* is given in Section 2. The enterprise modelling characterisation considers five major aspects, which constitute our notion of this term. An overview on existing approaches for manufacturing process modelling is discussed in Section 3. A draft meta-model of the manufacturing DSL is introduced in Section 4. This presentation of the DSL comprises its abstract syntax, as well as its usage in the context of manufacturing process configuration using reference models. Section 5 consists of a case study involving a manufacturing SME and a tool developed within a research project. The report closes with a summary and an outlook on future research in Section 6.

2 Basic Terminology

Manufacturing Systems are a classical subject matter of various academical fields, such as production management, business economics, engineering, Wirtschaftsinformatik[2], and computer science. For this reason, basic terminology has to be drew upon different sources, especially academical literature of production management and Wirtschaftsinformatik. Our notion of *enterprise modelling* is presented in Section 2.1, which is supplemented by well-known enterprise modelling approaches. Manufacturing related terminology is presented in Section 2.2.

2.1 Enterprise Modelling

Goal of the section at hand is to clarify our understanding of the term *enterprise modelling*. This characterisation focuses on relevant properties of *enterprise modelling* and therefore mentions aspects that separate common models from enterprise models. Enterprise modelling aims at describing enterprises (or organisations in general) using different perspectives and views. The basic idea of multiple views has already been proposed and elaborated in the context of several EM approaches. Some well-known examples are listed in this introductory section.

2.1.1 Main Characteristics

The term *enterprise modelling* is widely used and usually considers several abstractions. Consequently, a set of characteristics is given in order to provide a foundation for its understanding:

Conceptual Modelling Enterprise models usually are conceptual models. Conceptual models aim at representing major concepts of a given application domain. They focus aspects that are relevant over a long period of time. Specific instances are usually ignored.

Levels of Abstraction Information on an organisation can be presented on different levels of abstraction. Examples for different abstraction levels are diverse elementary process types (including concrete information), as well as aggregated processes (containing information on a cumulation of business processes). An enterprise model should consist of several levels of abstraction. A dedicated modelling tool can support the navigation between these levels.

Multiple Paradigms An enterprise model usually takes account of varying modelling paradigms. There is a major distinction between static and dynamic models. Static models might represent the organisational structure, as well as information processed within a process. Dynamic models reflect the behaviour of a system, e. g., (business) process models, state charts and data flow diagrams.

Interdisciplinarity Enterprise models tend to cover various aspects of an enterprise. Examples for such aspects are economic restrictions, processes, relevant data or production planning. Hence, multiple aspects and academic disciplines have to be taken into account. An enterprise model is therefore a combination of partial models, each representing a specific aspect of an enterprise.

Views and Integration Enterprise models can be structured by several dimensions. Such dimensions might be the *levels of abstraction*, *paradigms* and *domains*. A view represents an unique combination of facets of each of these dimensions. An enterprise model therefore

[2] The German speaking world's counterpart to Informations Systems (IS).

consists of different views and the consistency of all views have to be maintained. An integration of these views can be realised by common meta-model concepts and a common meta-meta-model.

Despite the existence of some common concepts and approaches, there is no generally accepted interpretation of the term *enterprise modelling*. There might be a consensus in some modelling communities. However, current research on enterprise modelling, as well as according publications are affected by different scientific disciplines. There are some approaches, which – at least in part – meet our requirements on enterprise modelling. These approaches are introduced in the remainder of this section.

2.1.2 Zachman Framework

Zachman presented his framework in the IBM Systems Journal in 1987 (Zach87). It has been extended by Sowa and Zachman in 1992 (SoZa92). The extended framework consists of two dimensions: The first dimension represents several aspects: data, function, network, people, time, motivation. The second dimension reflects different levels of abstraction. Each level is associated with a typical role of a participant in enterprise modelling. The levels of abstraction range from a planner's perspective (*Scope*) to the implementation of an information system (*Components*). The levels of abstraction are called *Perspectives*, each of which represents a strategic point-of-view, the operational level or the implementation of an information system. However, modelling languages that represent the particular views are not introduced.

2.1.3 GERAM Framework

The abbreviation GERAM stands for *Generalized Enterprise Reference Architecture and Methodology*. It has been developed by the IFAC/IFIP[3] task force formed at the *IFAC World Congress* in 1990. It is a framework for the unification of various methods of several domains to support enterprise engineering and integration. Hence, it comprises – among others – methods from industrial engineering, control engineering, and information systems. The idea is to generalise the contributions of existing enterprise architecture frameworks. More information on GERAM can be found in (BeNS03, pp. 22 et seq.). It represents an framework rather than a modelling method comprising process models and modelling languages.

The GERAM components describe requirements concerning reference architecture, modelling language, process model, tools, and enterprise models. The Enterprise Engineering Methodology (EEM) is in charge of describing the process of enterprise engineering and integration through process models or structured procedures with detailed instructions for each activity. The major part is the *Human Factor*, which defines the phases of the enterprise engineering and integration project, as well as the way of involving as much as possible people of the enterprise into the analysis and design of the manufacturing and service systems. Aspects concerning project management and economic issues have to be considered too.

Enterprise Modelling Languages (EMLs) have to encompass “various forms of descriptions and models of the target enterprise” (cf. BeNS03, p. 53). Thereby, various viewpoints have to be considered according to the views of the GERA modelling framework. Bernus et al. mention that a set of consistent modelling languages has to be selected or developed, such as the CIMOSA

[3] IFIP stands for International Federation on Information Processing; IFAC stands for International Federation of Automatic Control.

(cf. Section 2.1.5) or IDEF languages[4]. Such a language should be defined by a *Glossary*, a *Meta-model* or an *Ontology*. The language definitions are part of the so called *Generic Enterprise Modelling Concepts* (GEMC) component.

Partial Enterprise Models (PEM) provide a set of predefined reusable reference models. Further components are the Enterprise Engineering Tool (EET), which supports the design and the management of enterprise models. The Enterprise Module (EMO) provides already implemented building blocks or systems. The Enterprise Models (EM) component represents the designed enterprise models developed using elements of the predecessor components. The operational system is called Enterprise Operational System (EOS). It is in charge of the fulfillment of the enterprise objectives.

2.1.4 ARIS

The *Architecture of integrated Information Systems* (ARIS) is the most popular EM representative in academia and business practice. It is defined by the so called ARIS-House (Sche99), which comprises five different views. The views are given as follows:

Data View The data view comprises static aspects of an enterprise model. It represents data types, messages and events.

Function View Corporate functions are represented within the function view. It also contains goals and software applications.

Organization View The notion of organisation in ARIS comprises varying kinds of concepts needed for the execution of business processes. These might be organisational units, as well as resources (e. g., machines, human resources, computer hardware).

Output View All products, which are needed by or the result of a business process are situated in the *output view*. Examples for such products are information, services, physical goods or money.

Control View The afore-mentioned views are integrated by the control view. The dominating language is the EPC, which is used to describe the control flow between corporate functions. An EPC consists of functions (Function View) and events (Data View) and also integrates organisational units, as well as resources.

The process model of ARIS prescribes three phases for each of these views. Requirements are determined during the phase called *requirements definition*. Requirements of the application domain are recognised and documented for further development. The *design specification* serves as an interface between requirements definition and implementation. It maps the concepts documented in the requirements to implementation-specific concepts. These concepts are further described in the *implementation description* and mapped to concrete information technology components.

The ARIS-House is popular in the area of *Wirtschaftsinformatik* and enterprise modelling in Germany. However, the basic conceptualisation of the ARIS-House is ambiguous. Physical and human resources are situated in the organisation view. Software resources (i. e., application software) are part of the function view. Hence, resources are distributed over different views. Data and message

[4]The purpose of IDEF is to provide a powerful but usable modelling languages to support system design (cf. MeMa98, p. 209) or NIST. See publications of the National Institute of Standards and Technology for more information visit http://www.nist.gov/ (last visit 05.04.2007). IDEF stands for *ICAM DEFintion*. The former abbreviation stands for Integrated Computer Aided Manufacturing, which is initiative commissioned by the US Air Force in the 1970s. The main goal of IDEF is to leverage computer technology to increase manufacturing productivity. This improvement is should be provided by a set of modelling languages for system design and analysis.

types seem to be reasonable candidates for static concepts. But, events are also categorised as static objects. This categorisation is adequate as far as events do not contain activities and control flows. Events are notwithstanding an elementary concept of the control view, which is in turns described by EPC. Further applications of EPC discuss the feasibility of applying workflow modelling techniques within production planning and control (BeNH02). However, no specific modelling languages are published that provide concepts for modelling manufacturing processes and resources appropriately.

2.1.5 CIMOSA

The Open Systems Architecture for Computer Integrated Manufacturing (CIMOSA) is an EM approach that originates from a manufacturing context. It has been developed over a period of ten years in a number of industrial projects starting in 1984. The goal was to provide an integrated enterprise model, which helps enterprises in integrating facilities and operations. Its core components are:

1. The Enterprise Modelling Framework (EMF) for business operations representation, analysis, and design.
2. An integration infrastructure to support integration, execution, as well as monitoring and control of business and applications.
3. A process model for supporting the application of CIMOSA.

The process model of CIMOSA prescribes the phase's *identification, conception, requirements definition, design specification, implementation description, operation, maintenance* and *decommissioning* (Vern93; Vern96; Vern98). A reference architecture, the so called CIMOSA Cube, guides the user in applying CIMOSA. This cube can be used as starting point from which a particular enterprise architecture can be derived. Four views are provided, namely, the *Information View, Organisation View, Function View* and *Resource View*. CIMOSA distinguishes between three kinds of flows in an enterprise control flow, material flow and information flow. The former is defined as a workflow defining the enterprise behaviour. The material flow describes the transformation of physical goods. The information flow can be decomposed into data, document and decision flow. The different types of flows can be modelled separately, as well as in one single model.

- Information View
  - Enterprise Object: This element represents real-world entities of an enterprise. It comprises *Information Elements* and *Abstraction Mechanisms* (generalisation, aggregation).
  - Object View: The *Object View* is a manifestation or state of an enterprise object. It can either represent information entities or material entities (i. e., references to concrete real-world objects).
- Organisation View
  - Organisation Unit: An *Organisation Unit* is an organisational element defined by its list of skills, responsibilities and authorities within an organisation.
  - Organisation Cell: This element is an aggregation of *Organisation Units* and/or other *Organisation Cells*.
- Function View

- Event: *Events* represent relevant happenings of an enterprise. Examples are arrival of customer order, orders given by mangers, etc.
- Domain: A *Domain* represents a functional area of an enterprise encompassing a defined set of domain processes.
- Domain Process: A *Domain Process* represents existing business processes from its start to its end. It can functional decomposed into lower-level *Business Processes*. The processes on the lowest level are called *Enterprise Activities*.

- Resource View
  - Resource: Resources are grouped into active resources (Functional Entities) and passive resources (Components). The latter one comprises objects that need to be used or manipulated by a *Functional Entity*.
  - Capability Set: This element consists of various *Capability Elements*, which concern the functionality of an enterprise activity or resource.
  - Allocation Mode: The *Allocation Mode* expresses the reservation of resources at runtime. Possible values are first-in-first-out, last-in-last-out or any other algorithm for the assignment. No information could be found that express how to model further aspects of the reservation resources, e. g., capacity consumption.
  - Assignment Mode: The *Assignment Mode* can express an acquisition strategy between various jobs, i. e., priority.-based resource assignment

CIMOSA is sometimes propagated as the leading EM approach within the manufacturing domain (WoHG01; RaWe05; ChAW07). It prefers a non-graphical modelling approach. Semi-formal graphical modelling approaches are described as not useful for detailed analysis of behavioural properties, performance evaluation and resource management specification. The usage of Petri nets is recommended to formally describe business processes and enterprise activities (Vern96). We assume that the lack of semi-formal graphical modelling concepts hampers the understanding of the models. Especially stakeholders with non-technical backgrounds usually struggling in understanding formal models, since they lack semantics. Beyond that, CIMOSA does not provide appropriate semantics in terms of specific process and resource types.

2.2 Manufacturing Systems

A *manufacturing system* is a sub-system of the overall system *Enterprise* (cf. Figure 1). It uses or consumes scarce resources (Input), such as machines, material, humans in order to transform goods (Throughput) to a state of completion (Product/Output).
The optimal determination of these production factors is a classical task of production management (PM). Corsten suggests a conceptional framework for production management in order to reduce its inherent complexity (cf. Cors00, pp. 23f.). It presents three levels, namely, product and manufacturing program planning (Output), capacity planning (Input), and process planning (Throughput). These levels are interrelated to each other, i. e., program planning determines basic conditions for capacity planning. Manufacturing program planning addresses all questions around the current and future range of products encompassing research and development, as well as product design. Capacity planning is focused on the harmonisation of all applied resources, to avoid capacity shortages and accompanied idle time costs. Process planning is aiming at the optimal executing of production processes. It comprises the design of the production site to , for instance, reduce transportation costs.

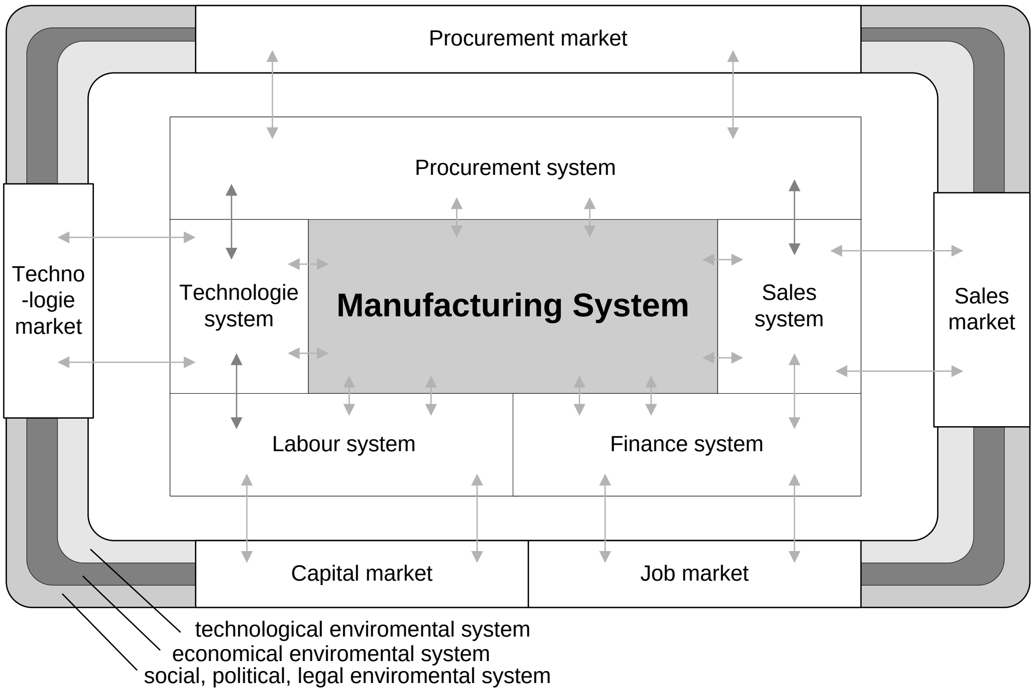

Figure 1: The Manufacturing System as Sub-System (Cors00)

Three time horizons are distinguished, i. e., strategical, tactical and operational. On strategical level, long-term decisions are made. Thus, knowledge about the overall situation of the enterprise is needed, which usually comprises information about the current product range, processes, resources and their capabilities, future demands, etc. The tactical level concretises the strategical specifications, e. g., outsourcing of processes or purchase strategy. Operational decisions are aimed at the optimal execution of the manufacturing system, with a given set of resources and orders. It becomes obvious that each decision level considers manufacturing systems on its own level of abstraction. Various modelling approaches have been applied to plan and control, or simulate manufacturing processes. This encompasses various forms of project plans, *Leitstand*, which are not considered further, since they lack appropriate modelling concepts.

This following paragraphs characterise fundamental terms of the manufacturing domain, which are relevant for the paper at hand. The terms business process (BP) and manufacturing process (MP) are introduced and discussed in terms of their characteristics.

Manufacturing Process The conceptual separation of business processes and manufacturing processes (synonyms: production / transformation process, throughput) is recommended, since they differ in terms of their process-subject, resources, and analyses. Both are usually not differentiated in literature in an appropriate manner. It is assumed that business processes surround manufacturing processes, i. e., both processes are deeply interconnected (ScBa95). Business processes represent administrative processes usually executed in the office encompassing finances, customer and supplier relationships, and more. By contrast, production processes are executed on the shop floor aiming at the transformation of goods. It is obvious that the subsequent contribution of the produced goods can be allocated to the business process area. The distribution of goods, for instance, is classically described as business, as well as manufacturing process. The latter implies a wide understanding of the term transformation, since goods are transformed in terms of their location and state (e. g., sold). Such state is in accordance to the Association for Operations Management (American

Production and Inventory Control Society; APICS) a state of further completion:

> A manufacturing process is a series of operations performed upon a material to convert it from the raw material or a semi-finished state to a state of further completion. (cf. CoBl98, p. 54).

Jiao et al. describe manufacturing processes as a combination of BOM, process and resources (i. e., workcenters as capacity units) (JiLP07). A process, in this context, is understood as set of activities carried out by human and technical resources. During its execution geometry of materials is changed, measured or formed material is transported, stored, assembled or quality checked. The goal is to realise products and services for customers, as well as values for stakeholders (BoBH01; RaWe05). In contrast to this specific a "business process is a recurring sequence of activities following a more or less rigid pattern of rules. A business process is goal-oriented and is directly related to the market oriented value creation of an enterprise. Executing business processes requires the use of scarce resources", which "frequently - but not necessarily - accompanied by a graphical illustration" (FrLa03). In the domain of manufacturing a more abstract definition of the notion of business process is given by APICS. A BP is "a set of logically related tasks or activities performed to achieve a defined business outcome" (cf. CoBl98, p. 11).

The following characteristics of manufacturing processes can be synthesised from existing literature regarding business and manufacturing processes:

- Manufacturing processes are surrounded by business processes.
- A MP is a sequence of activities (process steps) transforming a good from its raw or semi-finished state to a state of further completion.
- The process subject has to be described in a structured manner usually in a BOM.
- Within manufacturing processes specific resources are utilised or consumed. Such resources usually have relationships between each other comprising informational and material trade-offs (cf. Zele97, p. 1).

Process Plan A process plan (synonyms routing, routing sheet, bill-of-activities, bill-of-operations, operation list, routing sheet, etc.) is the counterpart to the German notion *Arbeitsplan*. It is commonly used to describe information "detailing the method of manufacture of a particular item" (CoBl98, p. 84) and can be labelled as the core information carrier, combining product/material features, processes, and further manufacturing resources, such as machinery and humans (Wien97; ElMa06). Thus, a process plan is a description of a manufacturing process on a low level of abstraction. It abstracts from alternative resources and process flows. In other words, processes are described in a strictly sequential manner. They can be either order-dependent or order-independent. The former necessitates the annotation of quantities and due-dates, taken from particular customer or work orders (Wien97). Process plans are used within various contexts, namely, material, and tool management, scheduling, material provision, calculation, and so on. A process plan usually comprises information about (Loos96; Wien97; CoBl98; JiTM00):

- applied material, parts and sub-assemblies
- the operations to be performed and their sequence
- involved working centers,
- standards for set-up and run time
- additional information: tooling, operator skills, inspection operations and quality control

Process Planning *Process Planning* is the activity of generating process plans, which "seeks to define all necessary steps required to execute a manufacturing process" (ElMa06, p. 2). The activity comprises the generation of the process description, so called process plans, numeric control (NC) programming and the organisation of quality control. The total generative process planning (i. e., full automated) of process plans is yet to be realised. The major challenge is the formal modelling of all relevant aspects and their interdependencies (cf. ElMa06, p. 4). Consequently, a couple of process planning approaches are published differing in their degree of automation and their level of granularity.

Resource Resources encompass "anything" that adds value to a good (CoBl98). Borja et al. mention that manufacturing resources enable the realisation of physical products. Thereby, an element can be a machine, a tool, a human resource (e. g., workers) or a material (BoBH01). Manufacturing processes require specific resources, which are commonly distinguished into consumable and performing resources. The former comprises consumable resources, such as raw-material and components. The latter is characterised by heavy machinery, ranging from simple machines, to complex computerised machines. A detailed discussion regarding resources in the context of business processes modelling can be found in (Jung07).

Bill of Material A bill of material is a "listing of all the sub-assemblies, intermediates, parts and raw materials that go into a parent assembly showing the quantity of each required to make an assembly." (CoBl98, p. 8). Various formats of BOM can be found in literature, e. g., one-dimensional and multi-dimensional BOM. The latter one allows the generation of variants through the use of assigned varying parameters, such as colour, material and shape. The connection of the product and its elements, assemblies and parts, are described by quantitative relationships (cf. Cors00, pp. 424f.)). It is quite common to group products and goods in so called (product) families. Product families are groups of products with similar characteristics (cf. CoBl98, p. 73). Grouping is usually done by similar parameters, functionality, or usage.

Production Planning and Control Production Planning and Control consists of a function setting the "overall level of manufacturing output (production plan)" to meet the planned levels of sales (sales plan). The control function is further responsible for managing (directing, regulating) "the movement of goods through the entire manufacturing life cycle from raw-material requisition to the final delivery of the finished product" (CoBl98, pp. 73f.). Thus, a PPC system is responsible for the timing and execution of particular operations with a given set of resources and orders.

3 Manufacturing-Specific Modelling: State of the Art

The paper art hand is motivated by the perception that manufacturing-specific concepts foster the applicability of EM concepts within the manufacturing domain. Existing enterprise modelling approaches, namely, frameworks (Zachman, GERAM) and methods (ARIS, CIMOSA), are only suitable to a limited extent, since they lack domain-specific semantics. This comprises manufacturing-specific modelling concepts regarding resources and their allocation to processes (e. g., goods and resource capabilities (ChAW07)). This section glance at modelling approaches that address particular aspects of manufacturing systems. The following criteria supports the comparison of the approaches from an EM perspective:

1. Community and Context: This criterion introduces the community, and in particular the context of the approaches. It also outlines topicality and regularity of publications published by the research groups.
2. Goal and Purpose: What are the goal and the purpose of the approach?
3. Object of Research: Describing the research object of an approach allows assessing the presented modelling concepts.
   a) Process Model: Is a process model provided that guide the user in applying the concepts?
   b) Modelling: Does the approach introduce modelling concepts?
   c) Modelling Language: If so, what kind of modelling language is introduced and how is it specified? Formal, semi-formal, and informal language specification are distinguished (FrLa03). A language specification, and in particular a meta-model based approach can be integrated into an existing EM framework conveniently. Provided that the modelling languages of a certain EM approach are specified adequately. In general, a formal language specification restricts the usage and – in part – the semantics of the language. It serves as a starting point for the implementation of a dedicated modelling tool.
   d) Reference Models: Does the approach provide reference models?
   e) Tool: Is a modelling tool developed and presented?
   f) Concrete Models/Case Studies: Does the approach provide concrete models, derived from case studies.
4. Multi-Perspective: A core criterion is the support of multi-perspective views, which is a fundamental requirement for enterprise modelling (FrLa03).
5. Integration of Modelling Languages: If an approach provides various views and languages, the question arises: How to integrate these differing views and languages, e. g., static and dynamic aspects?
6. Domain-Specific Concepts: This criterion addresses the support of domain-specific concepts. Another question is: Does the approach differ between business and manufacturing aspects?
   a) Manufacturing Concepts: Does the approach introduce manufacturing-specific concepts? Specific concepts like process, resource, and material (Wien97; Cors00; Chat03; ChAW07).
   b) Business Concepts: Business concepts encompass strategies, costs, or performance metrics (FrLa03)

3.1 Autonomous Logistic Processes

The approach "Modelling and Analysis of Dynamics of Autonomous Logistics Processes" aims at optimising material flow by innovative logistical control concepts. Logistic within manufacturing encompasses the planning and the control of production, transport, storage processes (ScWK+04; ScKH06b; ScKH+06a; ScHK06). The approach is issue of an ongoing research project funded by the German Research Foundation (DFG) (cf. project flyer[5]). Autonomous in this context means that decisions are made by interacting system elements, which can react autonomously on changing environment, e. g., exceptions like machine breakdowns. Each system element has to know its particular processing, from it's beginning to the point of it's finishing. This knowledge has to be described formally.

The authors introduce the following *core requirements* that the approach has to fulfill: decision making, interaction and self-monitoring. The decision making demands an explicit annotation of goals, parameters and input quantities. The notion of interaction emphasises the required ability of interacting with further system elements. Each element determines the behaviour of the entire system. The derived modelling concepts are introduced by a meta-model-driven approach enhancing the UML. The so called *factor X* represents possible conceptual enrichments – an issue of future research tasks. A UML class diagram is presented, which introduces manufacturing-specific concepts. The class diagram encompass the type *Resource* and *Good*. The former represents physical elements of a manufacturing system. It is specialised into the sub-classes *Machine, Tool, Employee*, as well as *Store* and *Conveyors*. The *Good* is the base class for physical goods of a certain type. It shows relations to other classes, e. g., *BOM, Activity, Customer Order Line item, Purchase Line item* and *Transportation Line Item*.

Static and dynamic modelling aspects are distinguished. The static models encompass the views *Structure*, *Knowledge* and *Ability*. The dynamic models describe the dynamic behaviour of the system. They encompass the views *Process* and *Communication*. Each aspect is captured by a certain UML diagram:

- Structure: The *Structure* is responsible for the depiction of the structure and organisation of the logistic system. Aggregation and composition can be used to model organisational units. UML model: Class diagrams
- Knowledge: This view describes the knowledge that has to be available for decision making. Dedicated goals have to be described and assigned to the particular logistic objects. UML model: Class diagrams; for more complex coherences a specific knowledge representation language has to be developed
- Ability: The *Ability* view provides concepts for the modelling of real life problems. The Mapping between the abilities and logistic objects is another issue of this view. UML model: Class diagram
- Process: The process plays an important role in connecting the static models with its dynamic context (behaviour). The description of time-based sequences of activities and states is focused. UML model: State machines and Activity diagrams
- Communication: This view describes the message exchange between the logistic objects. UML model: Sequence diagram

[5]Project homepage university Bremen: http://www.ips.biba.uni-bremen.de/projekt.html, last time visited: 15/02/2007

Recapitulating, the approach is in an too early stage to assess it exhaustively. The domain-specific extension of the UML seem to be a promising task. However, the manufacturing-specific extensions are not published so far.

3.2 Data Model Driven Manufacturing

The data model-driven approach introduces data models for supporting computer aided engineering. These models are used as foundation for database representations of products, processes and resources. The aim is to provide data for decision making during design and manufacturing (MoBe99). To provide this data, a data model for manufacturing the so called *Manufacturing Model* (MM) is presented, which should be applicable for any kind of manufacturing enterprise. The MM represents an information model that captures "data, information and knowledge" describing manufacturing resources, processes and strategies (MoBe99, p. 226).

Further publications integrate the MM with a specific product data model, the so called Product Model (PM) (BoHT00; BoBH01; BoHB01). The integration is introduced on a conceptual level using the Booch modelling notation. It is used to describe the information models, i. e., their certain elements and their relationships. Thus, the structure of the PM is described by a conceptual Booch model, the so called *Product Data Model* (PDM), which provides the value types and structure of a product. A model for the MM, manufacture data model (MDM) is introduced, as well. Both schemes use corresponding concepts, namely, the *manufacturing Resource* and the *operation*. A PDM *operation* is decomposed into MDM *manufacturing processes* illustrated by the arrows (cf. Figure 2). Tables have to be specified in order to describe more abstract strategies, e. g., technology, capacity strategies. Such a table encompass the elements *Cell, Strategic, Operational rules,* and *Performance measures*. The description of these aspects is not formalised.

Product Data Model A certain database for keeping the design data is used as a central information repository. The schema of the database is defined by the PDM. The core concepts of the PDM are product and manufacturing related aspects. The former one encompasses information about the product structure, as well as information about required resources and operations. It comprises certain types, such as *part, assembly, component, product* and *design entity*. The latter one is the core element of the PDM. It represents a physical object that satisfies a set of requirements. It possesses *specifications, actuals* and *definitions*. The definition type describes physical aspects (geometry, dimensions, tolerances, surface, etc.), performed functions and functional structure.
The manufacturing related aspects of the PM comprise elements for defining the physical description of the certain entities and feasible process routes. Whereby, a PM comprises (*operation*), *manufacturing resource* and geometrical changes of a design entity. A special entity for capturing manufacturing information is presented, namely, the *manufacturing information*. This class should comprise information about the raw material, the number of operations, total manufacturing costs and the time required to perform the operations. An *operation* can have two relations, so called kind_of relationships to *manufacturing resources*. The *secondary resource* link combines the *operation* with resources, which are required to perform the Operation. Resources linked with the *secondary resource* link represent facilities that have to be available for the execution of the operation, e. g., tools.

Manufacturing Data Model The MM represents information about the manufacturing system presenting the types *manufacturing processes, manufacturing resources, production tool, machine tool, facility, station* and *cell*. More specific types are *cutter, turning center, etc.*. The manufacturing resources are part of the PM. These abstract entities are decomposed in the MM by particular

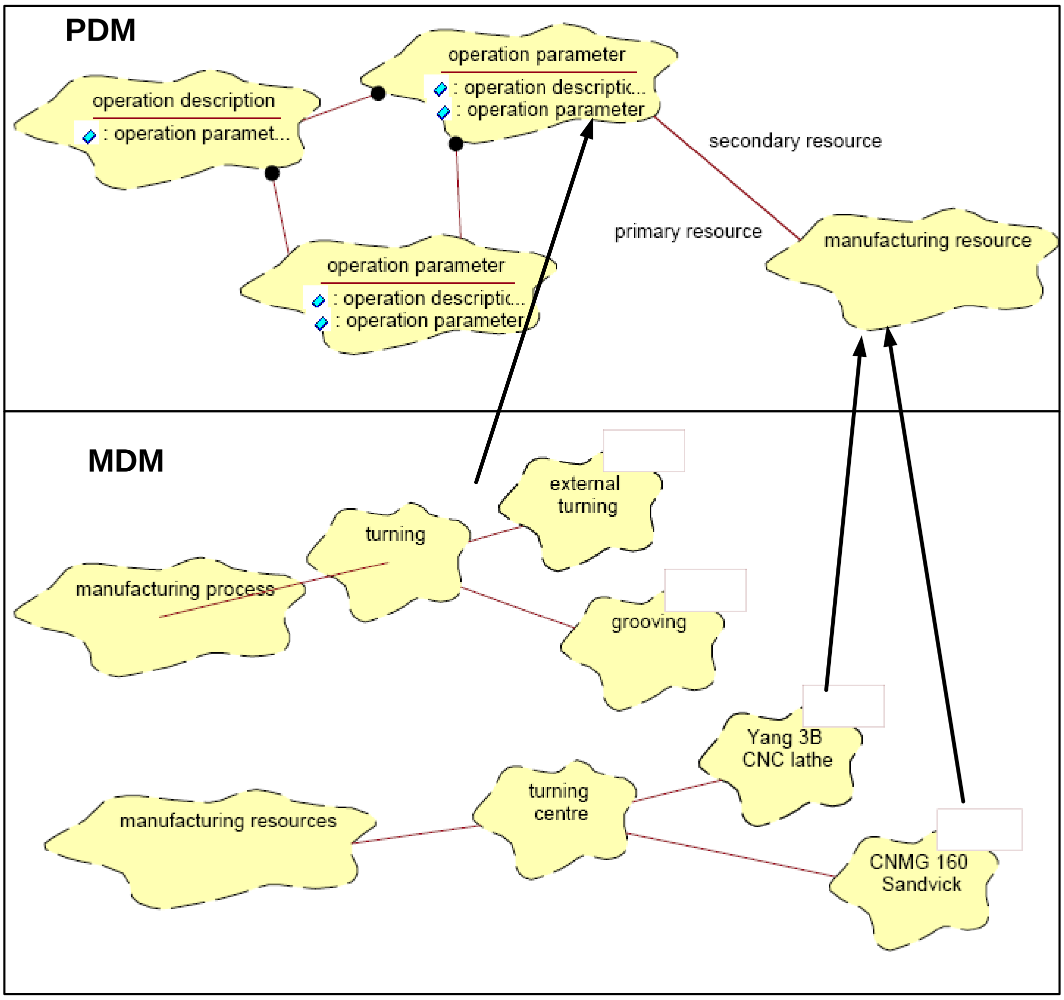

Figure 2: Relationship between PDM and MDM (cf. BoBH01, p. 1764)

resource instances. In other words, the MM describes specialisations of the type *manufacturing resource*. Figure 2 depict the relation of both models by directed arrows. PM operations, for instance, refer to more specific manufacturing processes in the MM, e. g., *external turning* and *growing*.

A manufacturing enterprise is decomposed into four layers *Factory, Shop, Cell* and *Station*. Each of them possesses its own level of abstraction in terms of strategic decisions, operational rules and performance. On the shop level, for instance, information are aggregated over certain shops, while the station level provide information about specific machines. Industrial case studies are presented in (MoBe99; BoBH01).

Summarising it can be said that the approach contributes to the integration of traditional separated product and process data by providing conceptual data models. In doing so, two conceptual models sharing corresponding concepts are introduced. However, the integration of MDM and PDM is not explicitly described. It remains unclear if both model share common concepts and how they reference each other. Figure 2 shows that MDM and PDM share the concept of manufacturing resource but the operation is not part of the MDM, which uses the term processes for describing operations. The decomposition of process and operation is not described in a comprehensible manner. Furthermore, the dynamic modelling of processes is not addressed at all.

3.3 Enriched Multi-Process Modelling

Enriched Multi-Process Modelling (E-MPM) is aiming at the combined application of EM, and in particular complementary simulation and workflow modelling. It is an approach enhancing CIMOSA and emphasising the need for manufacturing-specific modelling concepts and transformations. The approach is introduced by the enterprise modelling group of the research institute for manufacturing systems integration MSI[6].

The underlying assessment is that CIMOSA "[. . .] lacks sufficient modelling concepts to capture time-dependent (dynamic) attributes for an enterprise." (ChWe05, p. 115). The transforming of EM models (regarding process and resource) into simulation and workflow models is assessed as a crucial issue of EM (RaWe05, p. 2). Nevertheless, Chatha et al. assess the application of common enterprise modelling approaches as a time consuming and high skill requiring tasks, which would hamper the practical acceptance of EM approaches. In order to proof this assumption a literature review has been done, comprising selected EM approaches, namely, CIMOSA, RPM, IDEF3, IEM, (ChWe05; ChAW07). The conclusion is drawn that common EM approaches lack "coherent sets of process and resource systems", as well as the ability of modelling dependencies between "material, information, knowledge, control and exception flows." (ChAW07, p. 114). Unfortunately, the literature review in (Chat03; ChAW07) excluded ARIS without giving a reason. Further EM approaches like SOM and MEMO were not mentioned at all (Fran99; FeLZ06; Kirc07).

A further publication (RaWe05) reports on an ongoing research project, named "Study of the Interplay between Role Dynamics and Organisation Performance" (EPSRC). The goal of the project is to analyse and decompose the manufacturing enterprise processes and resources using an EM approach. However, the constitutional project publication describes neither the transformation nor the time-dependent properties in a comprehensible manner.

The E-MPM approach enhances CIMOSA by specific modelling concepts and process models (Chat03; ChAW07). The process model introduces seven phases:

1. Capture *as is* situation, and in particular the data collection from engineering partners.

2. Create and validate *as is* process models.

[6]see http://www.lboro.ac.uk/departments/mm/research/manufacturing-systems/, last time visited 22/03/2007

3. Create and validate dynamic models. These models are developed by enhancing the previously developed models, i. e., instance information, such as capacities.

4. Create static *to be* processes.

5. Create and validate dynamic *to be* processes by using scenarios and simulation techniques.

6. Modularising *to be* process using organisation boundaries and roles.

7. Transformation of process models to workflow models.

A so called "semantically rich process specification language" is introduced, which presents concepts for Activity, Events, Information, Human/Physical Resource, Finance and External links. It allows to describe parallel *resource* and *control* flow by assigning resources to processes. Its essential language concepts are outlined below, which are specified by textual descriptions (Chat04; ChAW07).

- resources: A resource possesses functional provisions, capability performance levels and constraints.
  - information resource: Depicted by a circle.
  - material resource: Depicted by an ellipse.
  - active resource: Depicted by a triangle; Performs actions in order to produce a good. The types machines, humans and (software) applications are distinguished.
  - passive resource: Depicted by an ellipse; Passive resource do not perform any actions. they take the form of two types namely, *processed-passive* and *used-passive* resources.
- enterprise activity or business process: Such an element describes functional requirements, performance requirements and constraints.
- Manual Enterprise Activity: Depicted by a rectangle; Operations and decision are carried out manually.
- Automated Enterprise Activity: Depicted by a rectangle; Operations and decision are carried out automatically with little human involvement.
- connectors:
  - resource transfer: Depicted by a dashed arrow.
  - resource transformation: Depicted by an arrow.

In summary it can be said that E-MPM is a promising approach for modelling manufacturing systems taking the need for domain-specific concepts and operationalisability into account. Several language concepts are introduced, as well as a process model for generating simulation and workflow models. The importance of time-dependent aspects for modelling manufacturing systems is emphasised. Process decomposition is supported, as well as the description of start and end events, so called *Triggering/Completion Rules*. Different process and resource types are distinguished. Resources can be assigned to processes. However, this assignment cannot be described in a quantitative or qualitative manner, as introduced by Jung (Jung03; Jung06; Jung07). Business concepts encompassing IT resources or business strategies are neglected. Business and manufacturing processes are not differentiated explicitly. Beyond this, neither the presented language concepts nor the transformations are specified comprehensibly.

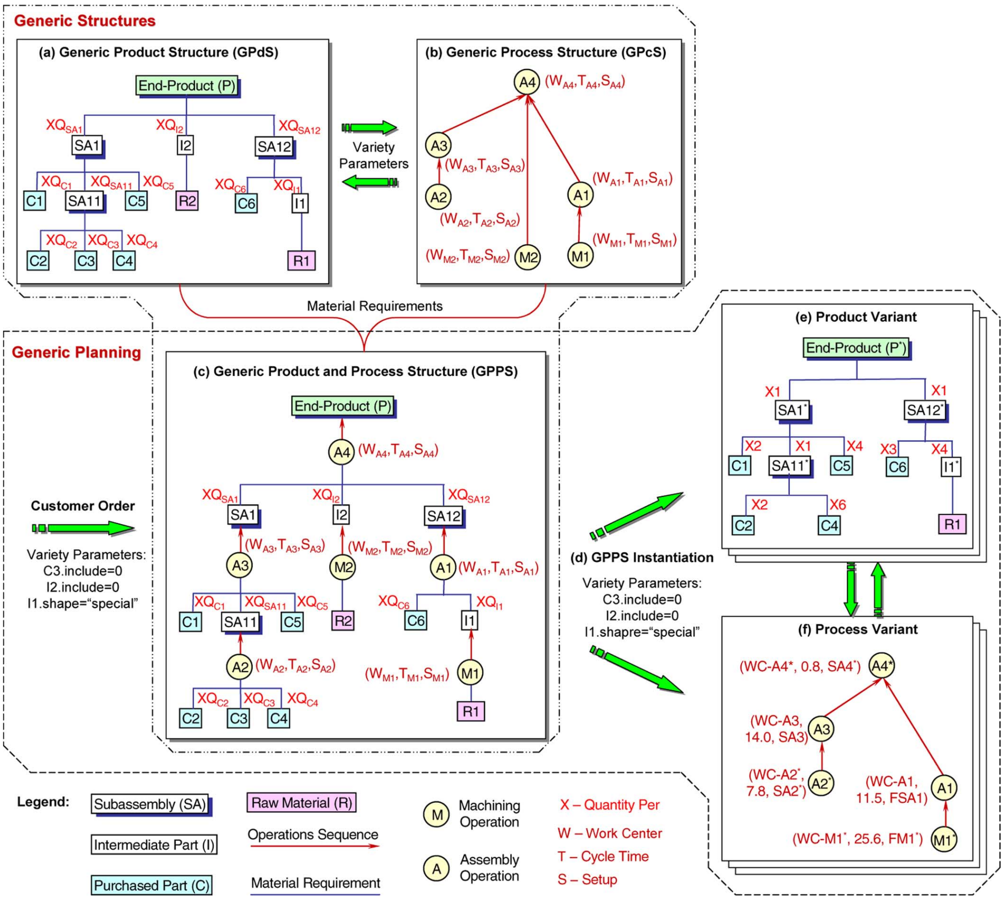

Figure 3: Structure of a Generic Process Platform (cf. JiHe06, p. 31)

3.4 Generic Product and Process Structure

The data integration of traditional separated areas of product design and manufacturing process is a central theme of several articles published in manufacturing-journals, such as Hastings, Yeh and Jiao et al. (HaYe92; Yeh95; JiTM00). Concepts are introduced that should support managing variety on the base of generic Bill-of-Materials-and-Operations (BOMO). A BOMO describes the dependencies between products and their production processes, i. e., to provide an integrated view on BOM and manufacturing processes. The idea is to handle increasing product variety, caused by mass customisation "with a coherent and effective mechanism." (JiLP07, 113).

Generic structures The generic structures describe product and process types, so called families, encompassing a set of products respectively processes. They can be used as starting point or blueprints for deriving particular variants,. They describe the constituent elements of product and process types, as well as relationships within and between product and process structures. The relationships between a product structure (BOM), for instance particular material, and its relevant processes is called *Material Requirement Link* (cf. Figure 3). A specific view, the *Generic product and Process Structure* (GPPS), integrates the two generic structures, namely, *Generic Product Structure* (GPdS) and *Generic Process Structure* (GPcS). The former provides a template for deriving product variants and the latter for specific process variants. A GPcS includes information about workcenters,

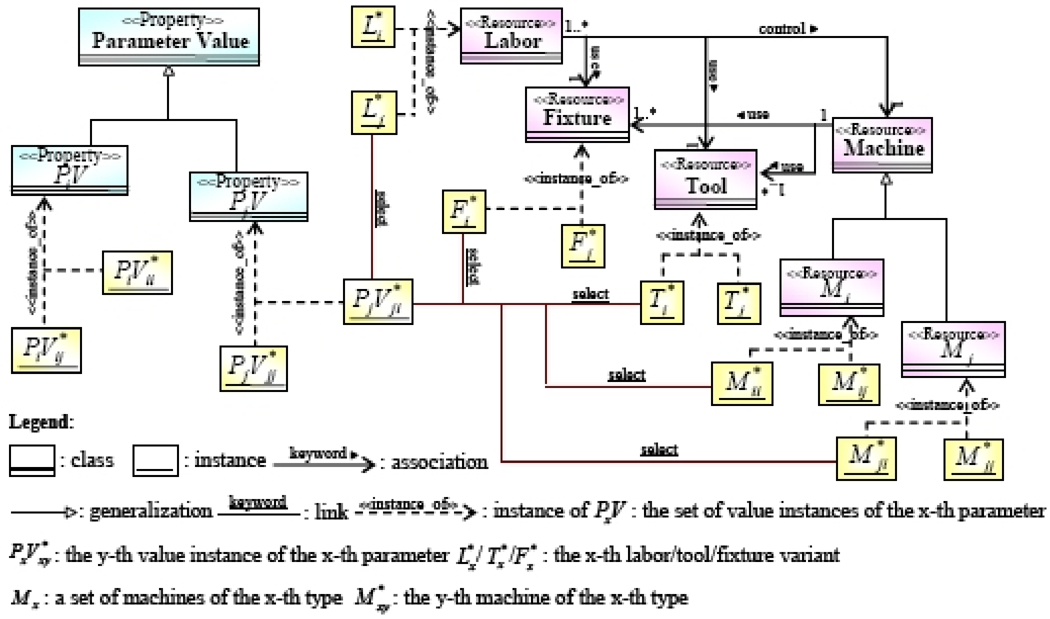

Figure 4: Links between Product and Process (cf. JiHe06, p. 31)

cycle times and set-ups assigned to each operation. A workcenter combines different types of machines or work stations, to be used by particular processes.

Jiao et al. present UML class diagrams to model the generic structures. Thereby, the levels of abstraction between classes and instances are mixed within the models. The idea is to describe product/process families (classes) and their family members/variants, which are modelled as instances. Classes and instances of the GPPS and the GPdS can be assigned to each other. This allows to describe existing relationships/dependencies between product (elements) and processes. In other words, a particular BOM object (e. g., raw material, finished product) can be assigned with specific process objects (e. g., activity, machines or tools). Instances are linked with its class by *Instance of* relationships. They can be linked by specific *associations*, namely, *select* and *specify* (cf. Figure 4). These types of associations are subsumed under the notion *Selection Rules* describing product constraints and *Planning Rules* describing process constraints. Such rules express that two objects are either compatible (AND) or incompatible (XOR) e. g., a certain shape should not be combined with a certain material or a certain type of paint. Shape, material and paint are examples of instances of the generic product class *Parameter*. Values, such as "round", "steel", "red" represent instances of the class *Parameter Value*. If an instance has a *XOR* relationship to an other instance on the same level[7], both are not allowed to be assembled or machined to create the upper (parent) item. The derivation of a particular variant is done by selecting predefined objects. For example: A generic product structure defines the class *Table Top*. Its instance comprises the assigned parameters "shape", "material" and "paint" and values "round", "steel" and "red". The following list outlines the presented classes for the representation of GPdS, as well as GPcS.

- Process
  - Operation: A set of aggregated sequenced manufacturing steps.
  - Material Handling system (MHS) used for material transportation.

[7] Level in this context has to be understood as an level of an bill of material. The highest level is the root of the BOM. The next levels encompasses further parts that have to be assembled or machined to form the root part.

  - Cycle time: Time from the start of the operation to its end.
  - Machine: A machine is a performing physical resource used by an operation.
  - Labor: Either a human or a robot carrying out a part of a particular operation using tools and fixtures. Set-up activities are included as well.
  - Set-up: Set of activities that has to be performed before the operation begins.
  - Fixture: Auxiliary device of a machine or labour.
  - Tool: Tool for executing an operation shared by a machine, as well as labour.

- Product
  - Item: Represents a physical product or a non-physical concept. An item can be an end product, a part, raw material or an assembly at any arbitrary level.
  - End product: A set of one ore more product variants at the root of the generic BOM.
  - Assembly: An Assembly has at least two child parts, as well as other assemblies. They are assembled together to form the end product.
  - Part: A part is either produced in-house or purchased. It consists of at least one raw material.
  - Raw material: A raw material variant is either produced or purchased.
  - Parameter: Parameter of an item, which may have a set of values.
  - Value: Represent an assignment of and value to an parameter.
  - Parameter Value: Defines the relationship between an entity and its parameter.

Summarizing it can be said that Jiao et al. present a concept for two classical separated areas, namely, design and production. The authors mention that a typical product configuration would involve thousands of products. This high variety increases the number of rules. The authors highlight the importance of a graphical modelling language in order to handle the resulting complexity. It is also stated that the UML would be unsuitable for supporting the variant derivation process. This dynamic modelling aspect should be a future research task. So far the UML would provide a "solution for graphical user interfaces of production and process modelling" (JiHe06, p. 24). It has been stated before that the application of the general purpose modelling language UML is not satisfactory for modelling manufacturing systems. Indeed, the application of UML presents an attempt to reduce complexity using conceptual modelling concepts. However, it has to be mentioned that class diagrams are not appropriate in terms of modelling dynamic aspects of manufacturing enterprises.

3.5 MFert

A popular method for modelling manufacturing processes is the MFert (abbreviation for *Modell der Fertigung*) modelling method. "It enables a model-oriented construction of manufacturing control systems" (DaKL97, p. 273). The modelling is done by directed bipartite graphs, similar to Petri nets. Additional manufacturing-specific concepts are introduced, with the aim to support computer-aided production planning and control. The original method has been enhanced by various concepts (DaWi93; DaWa97): Kuhn, for instance, developed a set of reference models for the analysis and design of PPC systems (Kuhn97). Langemann introduces concepts for assigning performance metrics to support analysis and evaluation of manufacturing systems (Lang99). Schmidtmann presents a specification language for the production control (Schm99). Geck-Müge defines a language using

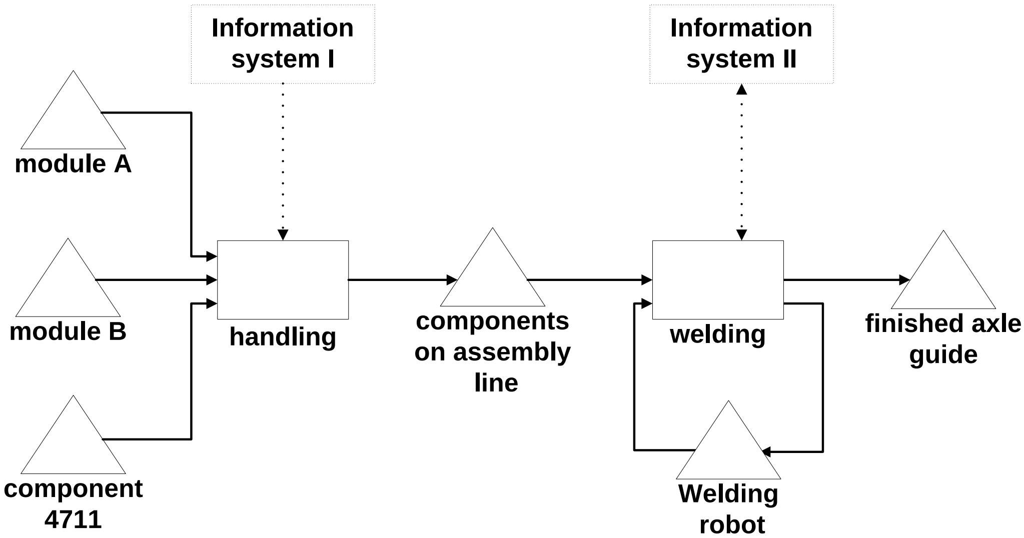

Figure 5: Exemplary MFert Model (cf. Lang99, p. 120)

static models in EXPRESS-G[8] to allow a conceptual modelling of information (GM99). A formal language specification of the fundamental concepts can be found in (Schn96). A tool implementation, named OOPUS is documented in (Holt97).

MFert claims to describe operable models for computer aided planning and control of manufacturing systems (GM99). A concrete *MFert* model has to cover all relevant data for production control systems. Two essential types of nodes are distinguished, the *Operation* (Transition) and *Element* (Place). Figure 5 shows an example *MFert* model depicting an multi-part component, which consists of module A and B, and component 4711. The welding operation is executed by an *Element*, called welding robot. The *element* expresses that one particular element is in a certain state, i. e., the static representation of material, resource or information. The term state refers to relevant states of resource. A transportation resource, for instance, might have the states *loaded* and *unloaded*. An *Element* can be an aggregate of further *Element* types. The aggregate is called *element_in_state_category*. Product *A*, for instance, is assembled by component *B* and component *C*, which represent the sub-categories of *A*. The tokens represent the selected/produced elements. Further concepts of the *Element* type are:

- time constraints: Timescale representation of availability and non-availability.
- capacity: The type capacity_of_element represents the capacity provided by a certain element.
- events: Events allow the annotation of time data to elements, such as inventory increases or decreases.

An operation represents a certain step of the production process. It describes certain input and output elements, as well as a certain operation time. They can be aggregated to *operation_category* or (*operation_method_class*).

- in-/output of operation: This type is used to describe the input and out in a quantitative manner.
- in-/output of operation class: This type is an aggregation of in/output operation types.

[8]Express-G is a graphical modelling language using a subset of the data modelling language EXPRESS (more information about Express (BeMS98)).

- role of involved element: Each element plays an particular role, namely,: object, physical mean, informational mean.
- duration of operation: This type describes the duration of an operation.
- time delay of in/output: This type describes the time between start and end of an operation.
- operation constraints: This type is used to annotate capacity constraints, which could not be assigned directly to a particular element.

In summary it can be said that the approach presents comprehensive modelling concepts taking various aspects of manufacturing systems into account. It presents process models and various language extensions, such as performance metrics. Nevertheless, due to the origin in engineering science the approach lacks business semantics. Beyond that, it is questionable that the provided graphical notation is clear for stakeholders with non-engineering backgrounds.

3.6 OMEGA

OMEGA (Objektorientierte Methode zur Geschäftsprozess-modellierung und -analyse) is an object-oriented method for modelling and analysis of business processes. OMEGA provides objects that are transformed by processes. Events provided by common business modelling languages, such as ARIS-EPC or MEMO-OrgML, are not supported. An informal specification of the OMEGA method can be found in (Fahr95). Further publications apply OMEGA within manufacturing domain, to integrate product and process modelling (GaGP+97). A more recent publication introduces OMEGA for modelling product design processes (GaMO+04).

The user is supported by a modelling tool named *OMEGAPrestige*, which should support the design of workflow management (WfM) or engineering data management (EDM) systems. A process model introduces four phases: An *analysis* phase (1), a *forecasting and goal definition* phase (2), a development of *to-be models* phase (3) and an *implementation* phase (4). The latter phase is encompasses three-steps. Firstly, the existing processes have to be modelled in close collaboration with the end-user. Secondly, the derivation of the *Processing Objects* has to be done, which serve as a starting point for designing product structures. Thirdly, workflow and data models have to be derived in order to support EDM or WfM systems. A task that has to be executed by an expert for EDM or WfM systems.

OMEGA Language Concepts OMEGA supports the decomposition of the processes, as well as the view concept. The former one is realised through aggregation relationships between the various objects, which allows a consideration of the models on various levels of abstraction. Fahrwinkel suggests building process aggregation levels in accordance to the levels of the organisational structure (Fahr95). It is possible to define specific views by selecting or hiding particular elements. This view configuration is supported by a specific query mechanism, which allows the combination or hiding of model elements. Views on the models can be customised using a query language.

- Business Process: Processes are represented by arrow boxes. It has an incoming and an outgoing processing object.
- Organisational Unit: This element is represented by an organisational unit, which is responsible for the process.
- Processing Objects
  - IT-Object: Information, Data, etc.

  - Paper-Object: Notes, drawings, etc.
  - Oral Information: Information that is not set out in writing
  - Material-Object: Raw-material, parts, end-products are taken as an example for this type of object. Hierarchic decomposition is provided.
  - Group of Information: Group of various processing objects.

- Other Objects
  - External-Objects: This element is used for all external objects, such as customers, suppliers, etc.
  - Communication-Relations: Linking business processes, technical resources and external objects.

Summarising it can be said that OMEGA supports the decomposition of processes, as well as the explicit exchange of objects. In contrast to other business process modelling languages (ARIS-EPC, MEMO-OrgML, etc.) objects instead of events are used to trigger processes (e. g., IT-Objects, Material-Objects). This is feasible for modelling material flows of manufacturing systems. Nevertheless, no particular case is documented that shows such material flow. Business processes and manufacturing processes are not differentiated explicitly. In addition, the presented language concepts (e. g., product, process) are not specified appropriately.

3.7 Related Work

Petri Nets Petri nets (PN) allow the formal modelling (semantic and syntax) of dynamic discrete systems. They support the analysis of time and dynamic system relationships. Various publications discuss PN in the context of manufacturing processes modelling (RuHo96; Vern96; Zimm97; ChPo04). The Deutsche Forschungsgemeinschaft (DFG) work group "Fuzzy-Petri-Netze" published a report on modelling manufacturing systems using PN (DFG97). Nevertheless, PN show some disadvantages: No integration with static models, such as Entity-Relationship-Diagrams, is provided. This lack of integration makes it difficult to describe the information requirements of enterprises, which is a core characteristic of EM. In contrast to semi-formal modelling languages PN lack semantics, since the modelling has to be done by reconstructing *Transitions* and *Places*. Unexperienced modellers or modellers with non-technical backgrounds may struggle in understanding PN models (Fran94; Lang99; ChPo04).

Variant Enterprise Modelling Wortmann et al. present an approach titled "understanding enterprise modelling from product modelling" (WoHG01). The comparison of enterprise modelling and product modelling result in the insight that "the notion of versions and progeny, the idea of domains and views, the notion of abstraction" show similarities (cf. WoHG01, p. 240). It is pointed out that variants are "not covered explicitly in enterprise modelling theory" (cf. WoHG01, p. 241). Thus, a generic concept for EM is introduced in order to support parametrised enterprise modelling. Up to now, no further publications could be found that would allow a detailed discussion.

Approach / Criteria	Auotnomous Logisitc Process Modelling	Data Model Driven Approach	Enriched Multi-Process Modelling	Generic Product and Process Structure	Mfert	OMEGA
Community	Manufacturing, Wirtschafts-informatik	Manufacturing	Manufacturing	Manufacturing	Manufacturing, Wirtschafts-informatik	Manufacturing, Wirtschafts-informatik
Goal/Purpose	Modelling and analysis of autonomous logistics processes.	Provide data models for design and manufacturing decision making.	Enhance EM by exectuable concepts for simualtion and workflow management.	Manage product and process variety.	Method for the planning and control of manufactuing systems.	Method for the modelling and analyis of business processes.
Procedure Model	No	No	Yes	No	Yes	Yes
Process Modelling	Yes	No	Yes	Yes	Yes	Yes
Modelling Language	UML, domain-specific concepts can be included (future task)	Booch class diagram	Graphical modelling	UML (Classdiagramm, Stereotypes)	Specific Modelling Language (similiar to Petri nets)	Semi-formal, graphical modelling
Language Specification	Semi-formal	Semi-formal	Textual	Semi-formal	Formal, except placement rules	No
Reference Models	No	No	No information	No	Yes	No
Concrete Models	No	Yes	Yes	Yes	Yes	Yes
Tool Implementation	No	No	Yes	No	Yes	Yes
Multi-Perspective Views	Yes	Yes	Yes	Yes	No	Yes
Manufacturing Concepts	Good (BOM, Line item), Resource (Machine, Employee, Tool, Store, Conveyors)	Product Model(Part, Assembly, Component, Product, Operation), Manufacturing Model (Process, Resource)	Resource (Material, Information, Machines, Humans), Process, Constraints, Events	BOM (Endproduct, Assembly, Part, Raw-material), Process (Operation. MHS, Cycle time, Machine, Labor, Setup, Fixture, Tool)	Element (Material, Machine and Information), Operation, further concepts Time, Capacity, Constraints	Not explicitly
Economical Concepts	No	No	No	No	Yes	No

Figure 6: Summary of the Literature Review

4 Manufacturing-Specific Modelling Concepts

The aim of our research is designing a manufacturing-specific modelling method. This aim is based on the perception that existing modelling approaches are only suitable to a limited extent, since they do not cover specific concepts appropriately. Thus, the section at hand presents modelling concepts that should be used to extend an existing EM approach, namely, multi-perspective enterprise modelling (MEMO) (Fran94). This enables retaining well-proven modelling concepts specified within the MEMO framework. A perspective in this context is a methodically reconstruction of a professional view on an enterprise, e. g., managerial, technical, dynamic and static. Complexity can be reduced by offering specific views hiding – for particular stakeholders – unnecessary information.

4.1 The MEMO-Framework

MEMO is a modelling method "that offers a set of specialised visual modelling languages together with a process model, as well as techniques and heuristics to support problem specific analysis and design." (cf. Fran99, p. 5). A multi-perspective enterprise model consists of various models of a corporate information system (e. g., class diagram, message flow diagram) and of the *action system* it serves to support (e. g., business process model). These models are integrated in order to foster collaboration of stakeholders with different professional backgrounds and corresponding technical languages, and to contribute to the integrity of the overall enterprise model.

MEMO allows to adapt and integrate modelling concepts comfortably by a dedicated language architecture (Fran98; Fran99). This architecture provides the integration of modelling language on a high semantic level. It comprises four levels of abstraction: Meta-Meta, Meta, Type and Instance. MEMO languages are specified by the same Meta-meta-modelling language, i. e., they are instances of the Meta-Meta-model (Meta-Meta-level). On Meta-level the different modelling languages are specified by instantiating the meta-meta language. They can be integrated by common concepts, as well as relationships between each other. The modelling languages are applied on type-level, i. e., the particular models designed by the certain language (e. g., object model). The MEMO language architecture supports the integrated usage of different language elements in one model (Fran99). So far, the MEMO framework has been extended by several modelling language addressing specific aspects of enterprises, such as strategy, IT-landscapes, and resources. Within the paper at hand, two particular languages (i. e., OrgML, ResML) are extended, since they already provide basic concepts for modelling manufacturing processes. Figure 7 shows the MEMO architecture and highlights the two selected languages.

- **MEMO-OrgML** – Organisation Modelling Language: This modelling language is focused on the operational and organisational structure of enterprises. The latter can be described by dedicated modelling concepts. Event based modelling concepts are introduced to describe the operational structure. These concepts (mainly process and event types) should be extended by dedicated manufacturing concepts.

- MEMO-SML – Strategy Modelling Language: A modelling language for the description of business strategy. The core mean of representation is the strategy network.

- **MEMO-ResML** – Resource Modelling Language: The ResML has been originally developed in the context of business process modelling. Therefore, concepts for extending the ResML are required to provide its domain-specific applicability for the manufacturing domain (cf. Section 4.2).

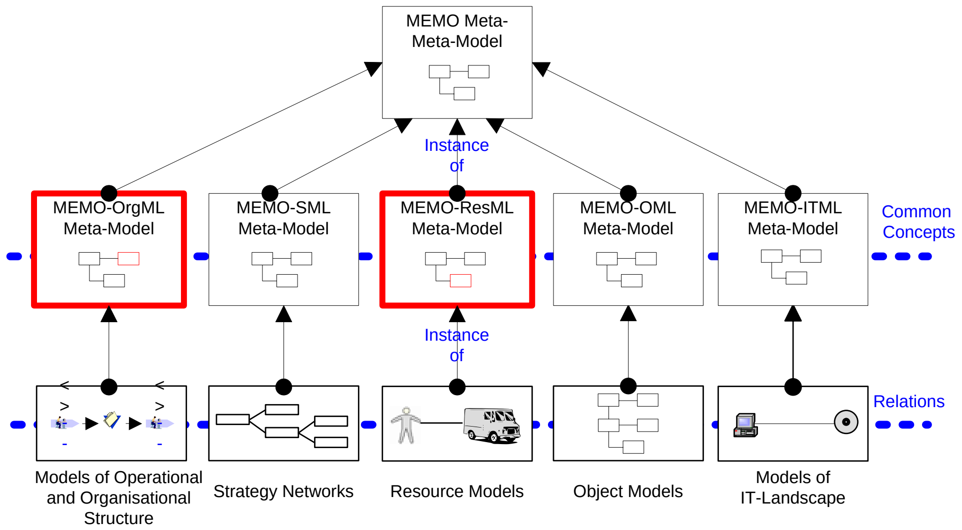

Figure 7: Integration of MEMO languages (cf. Kirc07, p. 35)

- MEMO-OML – Object Modelling Language: This language can be used to design object-models similar to the UML. Common concepts like *class, attribute, association*, etc. are provided.
- MEMO-ITML – Information Technology Modelling Language: This language allows the modelling of IT-Landscapes, which encompasses software engineering and hardware-specific concepts (*Scanner, Printer*, *Data*, or *License*).

4.2 Language Concepts

In this section we present a meta-model representing manufacturing-specific language concepts qualified for extending two MEMO modelling languages, namely, MEMO-OrgML and MEMO-ResML. A meta-model restricts the usage and – in part – the semantics of the language. It serves as a starting point for the implementation of a dedicated modelling tool.

4.2.1 Process Related Concepts

A manufacturing system transforms a given input (set of resources) to a specified output. Normally, such a transformation requires several process steps (also called activities or operations). A process usually requires at least one *Resource* to transform a *Good* from one state to a state of further completion. *Physical* and *Human* resources are applied during this transformation. Thus, each process requires resources, which are occupied or consumed. Jung introduces concepts for the description of quantitative or qualitative resource allocations (Jung03; Jung06; Jung07). These concepts have been extended by manufacturing-specific concepts (cf. Figure 8).

Process The central concept of the process meta-model extraction is the *Process*. A *Process* allows to describe all relevant activities on the shop floor, as well as manufacturing processes that are not executed by the enterprise itself. The aggregation of processes is provided by the association *comprises*. The definition of aggregated process types can be depicted with a specific symbol in a decomposition diagram. A *Process* provides six properties:

- *id*: This property provides an unique identifier.

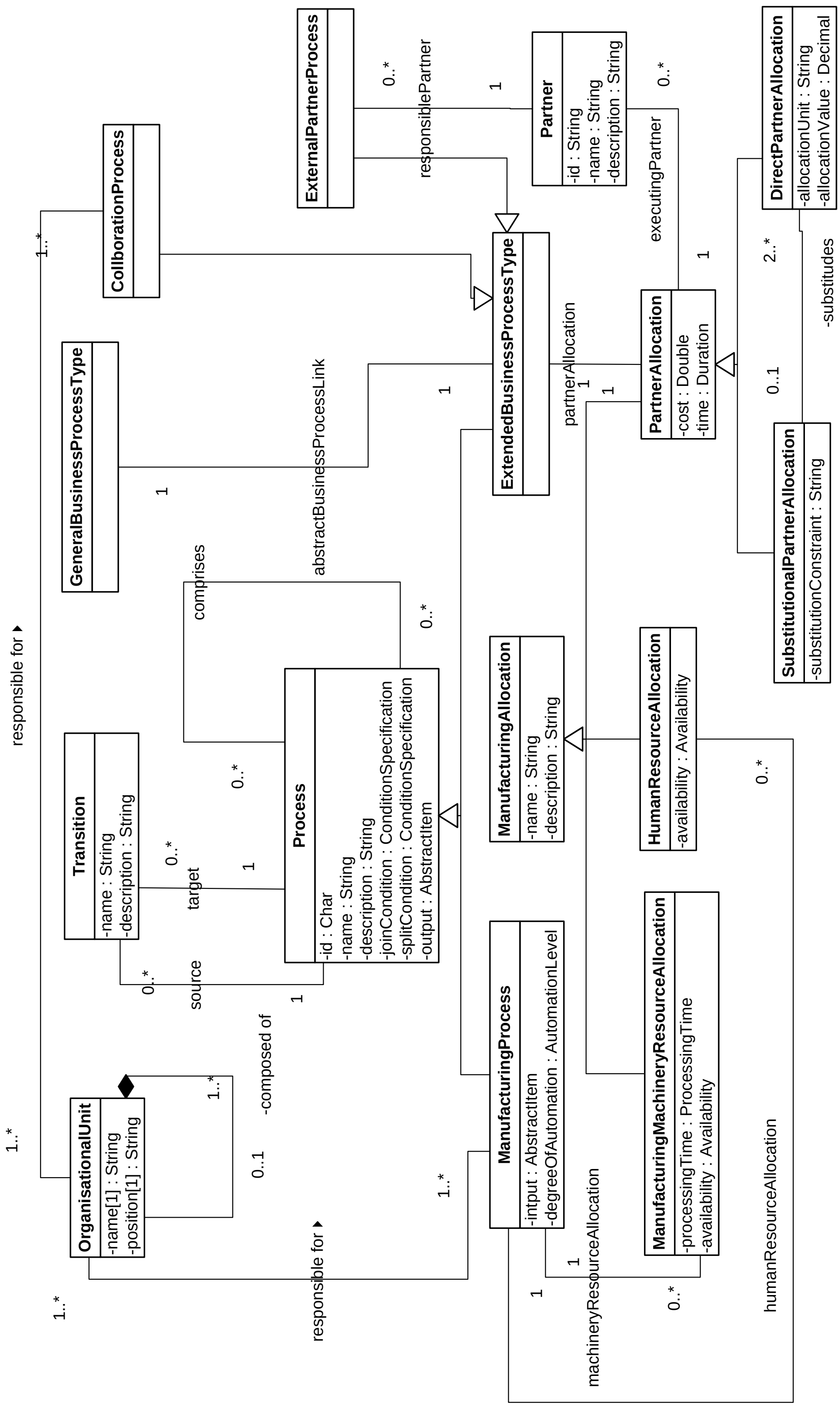

Figure 8: Process Meta-model

- *name*: Label for naming the process.
- *description*: This attribute allows storing a description of the process.
- *joinCondition*: This attribute is used to define the join condition of this type. Its default value is the sequence flow, which expresses that the process has one predecessor only. If a process has more than one predecessor, the condition attribute has to be either exclusive *XOR* or merge *AND*.
- *splitCondition*: This attribute is used to define the split condition of this type. It offers three different values: The *Sequence* that describes the link to one successor process. If a process has more than one successors, the condition attribute has to be either exclusive *XOR* or merge *AND*.
- *output*: This attribute is used to assign one or more output items. The outputs item type is the *AbstractItem*. Note, the concept of input (goods) is relevant for manufacturing process executed on the shop floor, only. Sub-contracted processes have an monetarily input (costs), which is of minor relevance for the paper at hand. Thus, the process type *ExtendedBusinessProcessType* does not possess an input.

The provided *Process* types are *ManufacturingProcess* and *ExtendedBusinessProcess* (business processes, e. g., purchase or external manufacturing processes executed by a external partner). Their major difference is the requirement of internal resources, i. e., an external process does not consume internal manufacturing resources. In doing so, in-house and cross company processes are distinguished. They represent fundamental decisions made within a product life cycle (cf. WiLu02, p.7). This decision might depends on the current state (utilisation) of the manufacturing system.

Manufacturing Process The *ManufacturingProcess* type represents transformation process. It consumes and produces *AbstractItem* types. The resources can be allocated *directly* or as a set of *substitutional* resource allocations. The latter describes a selection of different substitutional resources. They may differ in terms of their processing time. A *ManufacturingProcess* has to have a responsible *OrganisationalUnit*. The provided property is the *input*. This attribute is used to assign one or more items, which are transformed to an output. The latter can be modelled by the inherited attribute *output*.

General Business Process The *GeneralBusinessProcessType* represents the interface to the BP modelling language. It is associated to the *ExtendedBusinessProcessType* to express the relationship between both *worlds*. The design decision is motivated by the sub-system character of manufacturing systems. This means that manufacturing processes can trigger or can be triggered by surrounding business processes, such as purchase, invoicing, shipment processes. As introduced in an foregoing section, a manufacturing system is a *sub-system* of the overall system *Enterprise*, which is strongly related to its environment by further sub-systems, e. g., finance, marketing, sales, etc. (cf. Cors00, pp. 2f.). Therefore, the communication with its environment is usually not done by itself, i. e., other sub-systems are in charge of customers and suppliers (cf. Figure 1). It is recommended to describe these surrounding processes by business process languages.

Extended Business Process type The *ExtendedBusinessProcessType* is a sub-type of *Process*. The description of material flows between various processes is supported by the *input* and the inherited *output* attributes. The difference to the *ManufacturingProcess* is that it does not consume any internal resources. It is specialised into the types *CollaborationProcess* and *ExternalProcess*.

The *CollaborationProcess* is executed by an partner in a supply chain or a network. The responsibility is shared between the enterprise and the executing partner. Therefore, an responsible *OrganisationalUnit* has to be assigned. The *ExternalProcess* is executed by an external partner. This partner is in charge to perform a specific service. Thus, the responsibility for the execution of the process is up to the him. This is usually stipulated in a contract document. A *PartnerAllocation* is provided, which allows the annotation of partners that might perform the same service.

Partner Allocation The *PartnerAllocation* is a sub-type of *ManufacturingAllocation*, which is an element of the ResML. It provides two sub-types, namely, the *DirectPartnerAllocation* and the *SubstitutionalPartnerAllocation*. The former one allows allocating partners directly to a process. This implicates that no further partner is known that is able to execute the process. The *SubstitutionalPartnerAllocation* type provides a set of *DirectExternalAllocation* types. These types may differ in terms of its process parameters:

- *costs*: This property is used to annotate the costs of the sub-contraction.
- *duration*: This attribute is used to store the time that is required to produce a particular output. This marks the period from the process start to the point when the output is available for successor processes.

Partner A *Partner* type subsumes all external partners that are able to perform or deliver goods. Partners might be sub-contractors within a supply-chain or an enterprise network. Its properties are:

- *id*: This property provides an unique identifier.
- *name*: Label for naming the external partner.
- *description*: This attribute provides a description of the partner.

Transition The type *Transition* represents the link between two processes. Thereby, two different roles are distinguished, namely, *source* and *target*. A process can be source, target or both. The latter case is used to describe recursions, which requires a break condition. The type attributes are:

- *name*: Label for naming the transition.
- *description*: This attribute allows describing the transition.

Organisational Unit *OrganisationalUnit* is a common modelling concept, especially in the context of business process modelling. Its core function is to describe the structure of the organisation and to obtain responsibilities.

- *name*: Label for naming the transition.
- *description*: This attribute allows to store a description of the organisational unit.

Property The *Property* type is a general element to allow the modelling of further attributes. It encompass the *propertyName* and the its *propertyValue*. Properties can be individually specified for *Process, ExternalPartner* and *AbstractItem* types.

4.2.2 Product Related Concepts

In our understanding a product is any good produced for sale. Any *Item* can be sellable, consequently it can be a product with product-specific attributes, e. g., price. Products can consist of parts that may be other products or parts. Products and items possess a structure that describes their build, commonly called BOM. A BOM can consist of *Product, CompositeItem*, and *ElementaryItem*.

AbstractItem The *AbstractItem* serves for describing the output of processes, as well as the input of *ManufacturingProcesses*. It encompasses all elements usually depicted in a BOM. *AbstractItem* and its sub-types provide a basis for modelling BOM. The attributes of *AbstractItem* are:

- *id*: This property provides an unique identifier.
- *name*: Label for naming the type.
- *description*: This attribute provides describing the item.

Product A *Product* is usually but not necessarily the root of a BOM. A *Product* type is mainly characterised by its attribute *Price*. It possess an one-to-one relationship to the *Item* type in order to describe the product's structure.

- *value*: The calculated price of an certain product type.
- *priceUnit*: The unit of the price.
- *priceDeterminationMethod*: This attribute describes the method of price calculation, which might change from product to product. The method is defined by the type *DeterminationMethod*.
- *minimumPrice*: This attribute is *true* if the price is a minimum (production) price. Thus, the sales prices should not fall short of its *value*.

ProductItemQuantification A *Product* can consist of one ore more *Item* types. The type *ProductItemQuantification* describes the amount of items required to build a particular product. This type provides the attributes:

- *quantity*: This attribute stores the quantity of items.
- *quantityUnit*: The quantification unit may change from industry to industry or from enterprise to enterprise.

ProductFamily A *ProductFamily* can be assigned to a *Product*. This allows to describe product variants. As a *ProductFamily* type combines various similar product types, it necessitates a generic BOM structure combining the various similar product types and their parameters.

- *name*: Label for naming the type.
- *description*: This attribute allows to store a textual description of the product family.

Parameter and ParameterValue To provide the ability of modelling generic BOM, the BOM elements have to be able to define sets of parameters and its values. The parameters can be defined by the type *Parameter* and its particular *ParameterValue*. The former one provides the attribute *parameterName*. The latter one provides the attribute *parameterValue*. The definition of more than one *ParameterValues* is optional to describe the generic item structures, which may have more than one parameters. This allows a rudimentary description of generic BOM.

Item The type *Item* can be used to describe process in- and outputs (cf. Figure 9). It can be specialised into the types *ElementaryItem* and *CompositeItem*. The latter one can either represent an assembly, an intermediate or a part. If it consists of other *Items*, the *CompositeItem* holds an *ItemCompositeQuantification* that expresses the quantification of the ingoing lower level items. An *ElementaryItem* is a raw-material, which does not consists of other items. Raw material has to be purchased at least from one *ExternalPartner*.

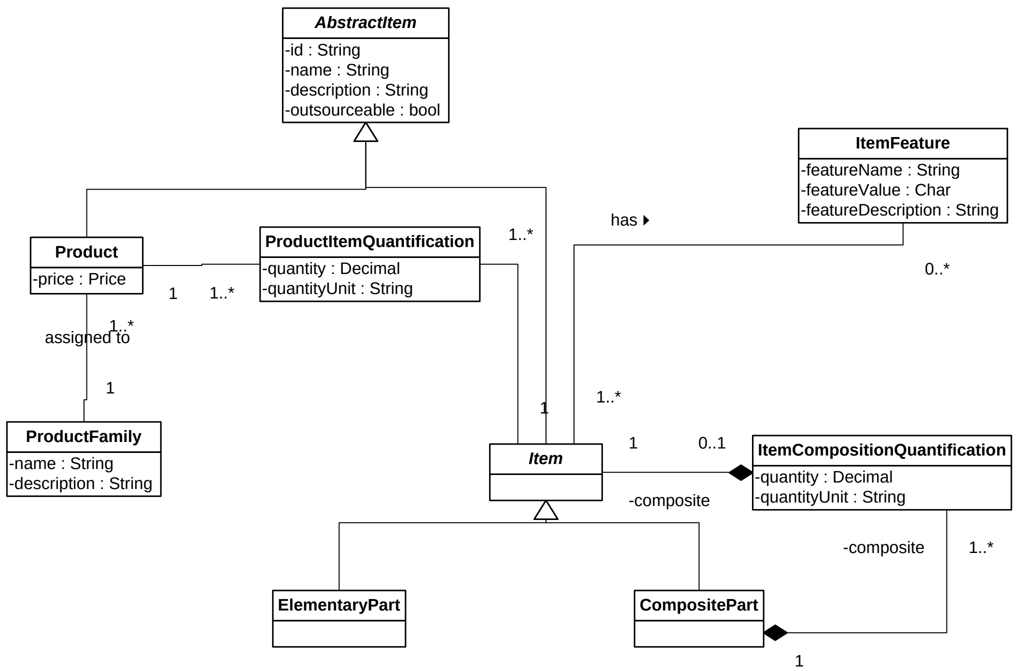

Figure 9: Material and Product Meta-model

ItemCompositionQuantification A *CompositePart* consists of a specific amount of other *Item* types. This relationship of items is described by the type *ItemCompositionQuantification*. The provided attributes are:

- *quantity*: This attribute stores the quantity of ingoing items.
- *quantityUnit*: This attribute allows to annotate the unit of the quantity.

ElementaryItem *ElementaryItem* types represent atomic *Item* types, i. e., they do not consist of other *Item* types. They are usually labelled as raw material, what means that they a normally delivered by *ExternalPartner*(s).

CompositeItem The *CompositeItem* can either represent an assembly, an intermediate, or a part. If it consists of other *Items*, the *CompositeItem* holds an *ItemCompositeQuantification*. A *ItemCompositeQuantification* describes the quantification of the ingoing lower level items.

DeterminationMethod The provided methods for price determination are defined as its sub-types. The provided attributes are:

- *name*: Name of the determination method.
- *description*: The description of the determination method.

The sub-types are:

- *Estimation*: Estimated value determination.
- *Calculation*: The value is calculated.
- *Predetermined*: The value is predetermined.
- *Measured*: The value is measured.

4.2.3 Resource Related Concepts

As an integral characteristic every manufacturing process uses resources for the transformation of goods. Processes and their relationships describe *what* has to be done. Resources assigned to processes specify *who* has to work on the process and *what* is needed. Resources are usually not available in an unlimited manner (PoLO99). Hence, scarce resources have to be taken into account for modelling, analysing, and optimising manufacturing processes.

The ResML is a language for modelling resources in the context of business process modelling. It consists of different resource types, resource relationships, as well as the concept of *resource allocation* assigning resources to business processes. Resource types comprise human, physical, and intangible resources. The latter ones are specialised to software, information, patent and beneficial interest. Physical resources are data media, machinery, transportation resources, as well as computing and communication devices. The main concepts of the ResML used in the approach presented in the paper at hand are shown in Figure 10.

Resource types A human resource is usually characterised by its skills. We differentiate between *skills* and *soft skills*. Both have a name and a description in natural language. Skills usually are capabilities, which can be learned in a course or training (e. g., foreign language, programming language, modelling language, mathematics, and welding). Skills are often documented by a certificate. In contrast to that, soft skills can hardly be proved by an official document. A soft skill is rather given by a distinguishing quality of a person (e. g., convincingness).

The resource type *machinery* is mainly described by its *capacity specification*. Such a specification distinguishes between a *minimal*, *optimal*, and a *maximal* capacity. The maximal capacity generally represents the upper limit of a machine, which must not be exceeded. The minimal capacity is the machine's lower limit. Producing at a lower rate (quantity per time unit) is not possible. The optimal capacity reflects an optimal quantity-time-ration from an economic point of view.

Both resource types are sub-types of `ElementaryResource`, which in turn is a sub-type of the `AbstractResource` type. Attributes inherited from `AbstractResource` are `name` and `description`. The first one denotes an unique name for a resource type; the latter one is a textual description of it. A *potential unit of allocation* is defined within `ElementaryResource`. This attributes refers to a

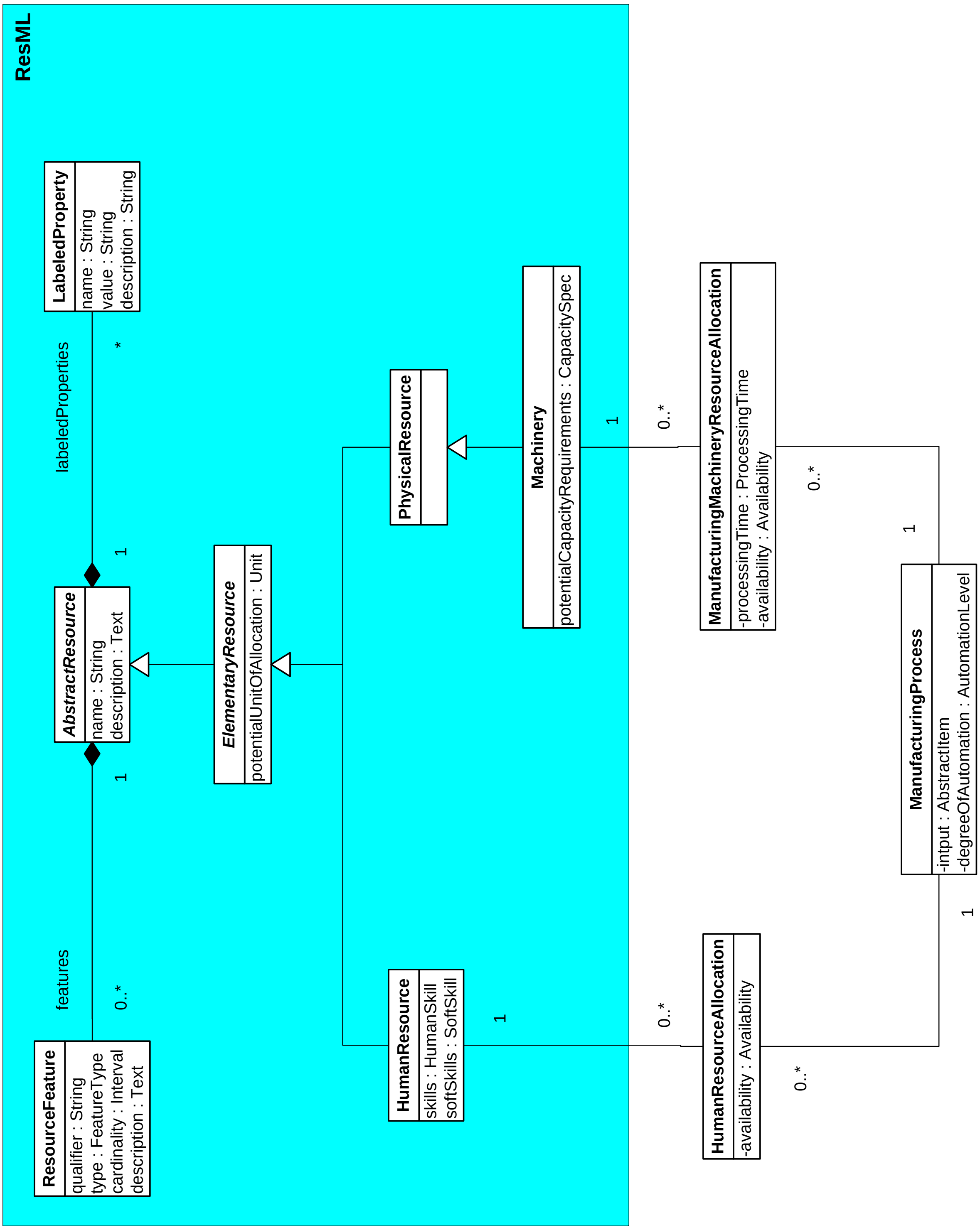

Figure 10: Resource Type Meta-model

typical allocation unit of a resource type. Human resources are typically allocated on a time base (e. g., hours worked on a project) and the consumption of goods on a quantity basis (e. g., quantity per item produced). Such an allocation unit does not have to be valid for all resources of a given type. However, it represents a typical allocation unit, which has to be applicable for most resource instances of that type.

Resource Allocation Resources are needed for the execution of business and manufacturing processes. However, the quantification of a resource's share regarding a process' execution depends on both, the process, as well as the resource assigned to it. Different resource types allocated to one single process might result in distinctive allocations and a given resource might affect the execution of different process types. A vineyard can, for example, be reaped faster using harvesting machine instead of people. But, a human resource is better suited for identifying grapes of good quality. Hence, the concept of *allocation* is used for modelling the usage of a resource during the execution of a given process.

Manufacturing processes are allocated to resources using human or manufacturing resource allocation. Either resource type is represented by a corresponding allocation type. Both types (human and manufacturing resources) determine the general availability of a resource type (`availability` defined in types `HumanResourceAllocation` and `ManufacturingResourceAllocation`). The description of manufacturing resource's allocation additionally requires the specification of different kinds of processing times[9]:

- The *transportation time* represents the time required for moving a single product from one machine to another.
- The time required for setting up a machine for a specific product is called *set-up time*.
- The plain time required for processing an item is called *run time*. It does not include any time for setting up a machine or transferring goods to another machine.
- Sometimes, goods produced in a single step have to rest for a given time, which is called *idle time*.

The partial meta-model for allocating resources to manufacturing processes is shown in Figures 10 and 8. The allocation of manufacturing machinery is described using the attribute `processingTime` as given above. Hence, the type `ProcessingTime` encapsulates transportation, set-up, idle and run time. Furthermore, the allocation of both, human and machine resources is determined by their availability. The general availability of such resources is documented using respective attributes (`availability`) in the resource allocation model.

4.3 Using Reference Models for Manufacturing Process Configuration

As explained in Section 3, the manufacturing domain is insufficiently supported by enterprise modelling approaches. Hence, the approach presented in the paper at hand aims at modelling, optimising, and implementing manufacturing processes in the context of enterprise modelling. It therefore distinguishes between business and manufacturing process models. A manufacturing process model is embedded in a business processes model. The latter one is part of an enterprise model and explicitly supports the achievement of an enterprise's goals. An embedded manufacturing process focuses on the production of goods and its goal is determined by the surrounding business process' goal.

[9]Those kinds of time data usually varies between different products and production plans. Hence, absolute times represent typical values

4.3.1 Overview on Manufacturing Process Configuration

As manufacturing experts are usually not familiar with enterprise models, a corresponding approach should meet their needs by providing adequate concepts for a rapid – and at the same time correct – modelling of manufacturing processes. This requirement is achieved by providing a domain-specific manufacturing modelling language (cf. Section 4.2) and elaborate generic models. An overview on the generic models used within our approach is given in the upper part of Figure 11. For every product family exists a generic BOM, as well as an associated manufacturing process reference model. Such a process model describes the manufacturing tasks needed for the production of a product family on an abstract level. This model has to be refined for any concrete product belonging to the corresponding product family. Analogously, the generic BOM specifies the structure of a product on an abstract level. Deriving concrete models from generic BOM and manufacturing process reference models works as follows:

1. Select a generic BOM representing the product family of a given product. This generic BOM has to be instantiated in the following way:
   - Properties of a BOM part specified by attributes, represented by name-type-pairs. Hence every attribute has a locally unique name and a type.
   - There is a range for possible values of a property (e. g., length between 25 and 35 centimetres).
   - For every property, a value satisfying the property's type and range has to be defined.

   This task is represented by *Specify Values* in Figure 11.
2. Every product family is associated with a corresponding manufacturing process reference model. Such a reference model describes the process needed for the production of such a product on an abstract level. It has to be configured with respect to the production of a concrete product. Hence, additional processes have to be introduced, which are unique to the manufacturing process of one single product. However, the concrete part's structure has to fit the one of the generic part. This task is represented by *Selection and Configuration* in Figure 11.
3. Typical resources for the execution, given manufacturing processes are allocated to the manufacturing process reference models. Throughout the configuration of a reference model (as described in Figure 11), concrete resource's allocations have to be refined on the basis on the reference manufacturing process models.

4.3.2 General Manufacturing Process Configuration

The approach presented in the paper at hand addresses a domain, which usually does not use enterprise models for analysis and planning purposes. Hence, our approach broadens the usage of models within the manufacturing domain by integrating enterprise models and manufacturing process models. The models used for the determination of concrete manufacturing process models consist of the following parts:

- A *generic BOM* reflects the typical structure of all products representing a given family. Properties in the generic BOM are given on a type-level (attributes represented by name-type-pairs). Every attribute specifies the type of a – named – property and its range might be restricted. The length of a part might for example be restricted to a range between 5 and 8 meters.

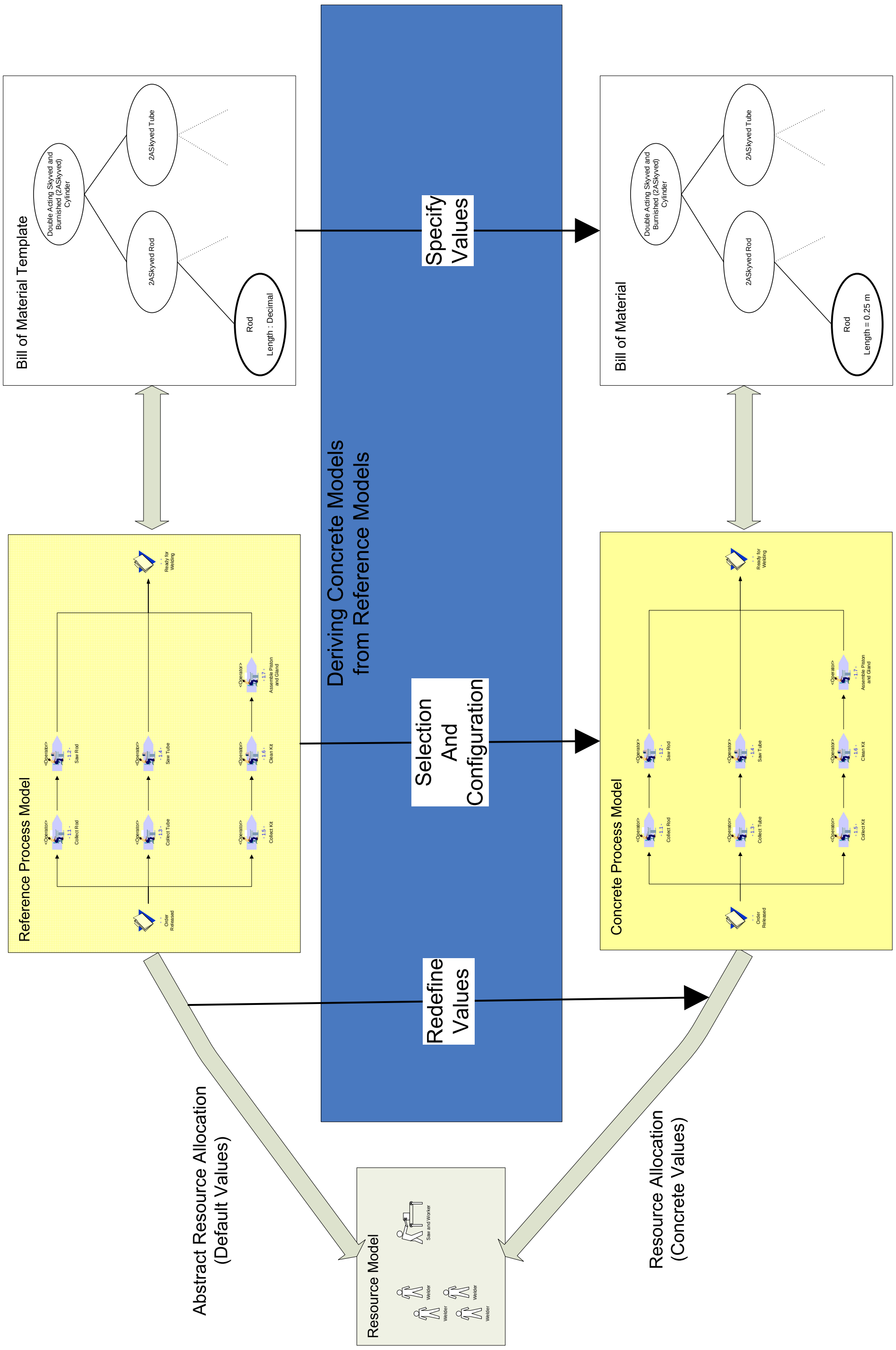

Figure 11: Usage of Generic Models for Manufacturing Process Modelling

- Reference process models assigned to product families represent typical manufacturing processes for concrete products belonging to the given family. However, there are many differences in processes for the production of concrete products. Hence, every reference process model has to be adapted to a concrete product's manufacturing process model.
- Reference manufacturing process models usually refer to resources, which can be utilised or consumed. The different abstraction levels are depicted in Figure 11. Resources are generally allocated to reference models (*Abstract Resource Allocation (Default Values)*). Such an allocation can be refined during the configuration of manufacturing process reference models. Default allocation specifications and alternative resources are restricted to concrete resources (*Redefine Values* in Figure 11).

4.3.3 Example Application of the Approach

We aim at providing different reference models so that concrete models of products can be derived in a comfortable manner. A typical scenario within this context is characterised as follows:

- A manufacturing planning and management system contains several reference models, each representing a single product family.
- After the arrival of a new inquiry for a specific product, the production manager acts as follows:
  - Select a product family representing the product
  - Configure the reference models with respect to the values of the concrete product
  - Release the configured model to the production planning system and determine concrete product times
- If the product can be produced within the given manufacturing process, the order can be accepted. Otherwise it is declined.

Every order accepted is transferred to the production planning an manufacturing system. This comprises all information regarding the client, the product type and product itself. Product information consists of a bill of material (BOM), the control flow of an items manufacturing process, as well as required resources.

Reference Model Derivation within Manufacturing We recommend that a generic BOM, a reference process model representing a typical control flow, as well as a prototypical resource allocation is assigned to each product family. The configuration of a new order relating to product families is shown in Figure 11. The procedure is as follows. After selecting a product family, the user has to refine the according models. A generic BOM consists of kinds of values associated with a product. The user has to provide concrete values in order to generate a concrete BOM. A generic BOM, for instance, specifies physical length and width for a certain product. during the configuration concrete values for length and width have to be attached to describe a particular product. Beyond that, the control flow and the resources of the abstract process model have to be refined.

4.4 Language Transformations

Manufacturing models usually show a set of instance-oriented models, e. g., scheduling models. Therefore, a draft approach of a transformation is introduced in the section at hand. This transformation bridges the intended gap between conceptual process models and instance oriented scheduling models. Figure 12 shows the core elements required for the transformation. The dashed arrow represents the requirement of integration between the enterprise model types and the information held by the information system on the operational level (**Operational System**).

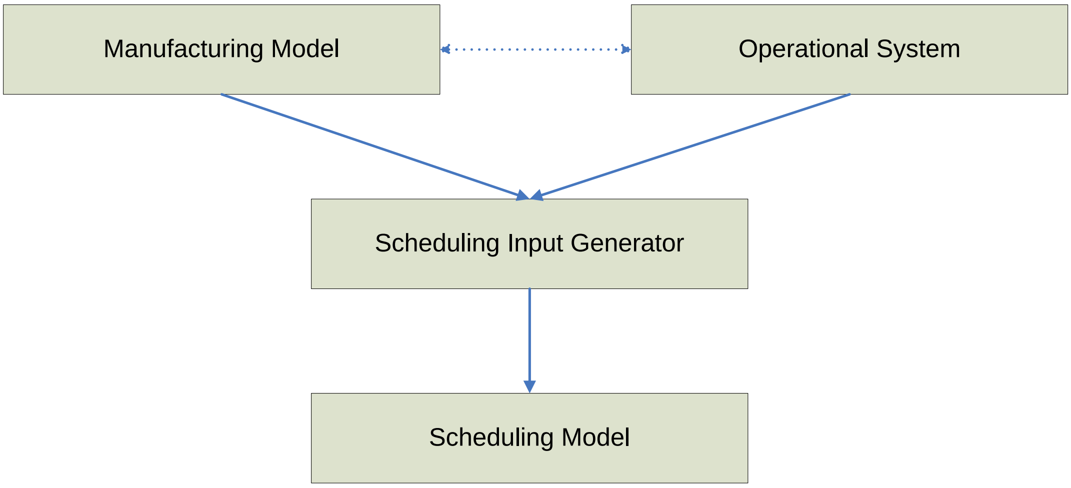

Figure 12: Transformation of Conceptual Models

The aimed model is a scheduling model for optimising the production plan for a given time period. The solver is able to handle alternative process flows and alternative resources. Both modelled on conceptual level as exclusive process flows and substitutional resources. The conceptual manufacturing process model is the base for the generation of the solver input. As type information is included only, a *Scheduling Input Generator* is in charge of combining the conceptual models with the instance data. This data is taken from the **Operational System**. Table 1 shows the information provided by the particular source. It encompasses further concepts that are out of scope of the work at hand. Future research tasks will examine additional applications and interfaces in order to support further manufacturing models.

Manufacturing Model	Operational System
Process models (processing times) Product types Allocated resource types	Orders and amount of particular products Resource instances (e. g., Machines, Humans)

Table 1: Manufacturing Model vs. Operational System

5 Case Study

The approach presented within this paper has been applied to the business processes of an enterprise dealing with the construction and manufacturing of hydraulic cylinders. This firm produces individual cylinders basing on its customer's requirements. Cylinders usually vary in length, width and the position of valves. There are some prototypical cylinder types, each representing one single product family. Such a family corresponds to a prototypical design (i. e., single or double-acting cylinders) each being specialised by individual requirements. The case study firm employs about 40 people – mainly shop floor operators.

5.1 Requirements

Capturing and describing manufacturing processes should be supported by an according modelling tool, which can provide automated model maintenance, validation, analyses and transformation. This includes the identification of manufacturing processes, as well as their formal definition. However, the effort for building production process models should be leveraged as much as possible. Core competencies of manufacturing firms are situated in the execution of manufacturing processes. Hence, the approach presented within the paper at hand basses on reference models. Every reference model reflects the activities needed for the production of a part family. A part family represents a set of similar products, which can be produced by a given generic manufacturing process. Concrete requirements on the tool support for manufacturing planning and optimisation are given as follows:

Usability The tool should be usable by people without elaborate modelling skills. Many domain experts are skilled with respect to their application domain. However, they usually have no modelling experience. Hence, the modelling language should contain concepts, which are common to manufacturing people.

Reusability Modelling usually demands (scarce) resources. Consequently, a tool for modelling manufacturing processes should allow for reusing (parts of) models in different contexts. Reference models are a common abstraction for reusing knowledge within a given domain.

Optimisation Goal of our approach is the support of optimisation of manufacturing processes. Since optimisation is not an inherent part of our tool, it should allow for the generation of a representation, which can be used as an input for optimisation tools.

Basing on the concepts presented in Section 4.3, we aimed at developing a modelling tool supporting the following procedure: After the arrival of a new customer order, the production manager selects an according part family out of a repository. This repository has been created during a preliminary customisation phase. The production manager then adjusts the reference model to the product ordered by the customer, as he insert concrete values and selects appropriate resources. The control flow of the process model can additionally be adapted in a limited manner. Finally, a textual representation can be generated as an input for an optimisation tool.

5.2 Determination of Reference Models

Reference models have been designed in close collaboration of several representatives from the cylinder manufacturing firm and researchers from different fields. Research fields covered are *manufacturing*, *constraint programming*, *enterprise modelling*, *computer science*, and *information systems*. In the course of the workshop the manufacturing plant has been presented to all participants, as well

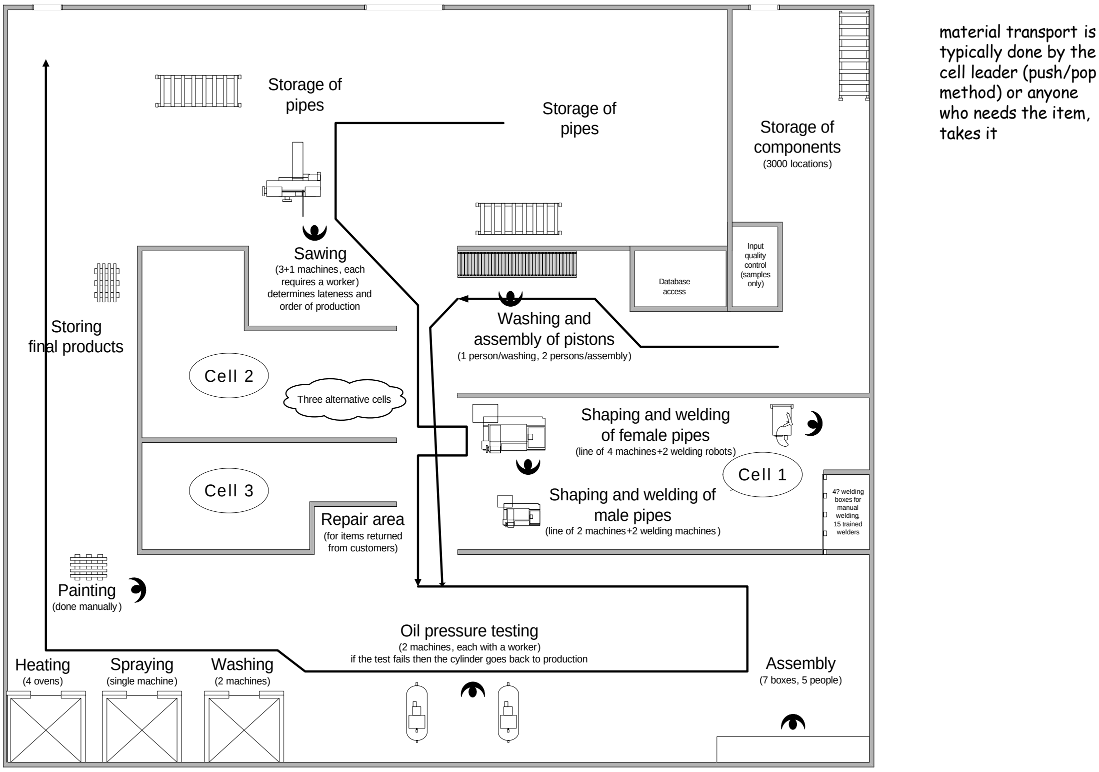

Figure 13: Drawing of the Manufacturing Plant

as the already used information systems. An outline of the manufacturing plant is given in Figure 13. In a conclusive session, two major cylinder types have been identified; the so called *Double Acting Skived and Burnished* cylinder (2ASkived) and the *Double Acting Smooth Bore* cylinder (2ASmooth), both based on the double acting cylinder (cf. Figure 14).

A reference model covering general aspects of both kinds of cylinders is given in Figure 15. Every cylinder mainly consists of a tube, a rod, and additional parts (represented by a *kit*). Those parts (like valves and glands) are welded to the tube and the rod. Both parts (tube and rod) are assembled afterward. Subsequently, the entire product is cleaned and optionally painted. Reference process models covering all manufacturing steps (including quality assurance and delivery) have been created in the context of the project.

Additionally an exchange format has be defined in order to support optimisation models, namely, an ILOG[10] (BaCS07).

5.3 Modelling Tool

A modelling tool has been developed based on the concepts described in Section 4.3, the requirements given in Section 5.1, and the reference models mentioned in Section 5.2. We used the Eclipse platform as an implementation basis, as well as several Eclipse-based frameworks. One of these frameworks is the *Graphical Modeling Framework* (**GMF**), which supports the creation

[10]ILOG is an popular software developer for optimisation solutions, more information http://www.ilog.com/, last time visited: 27.10.2007

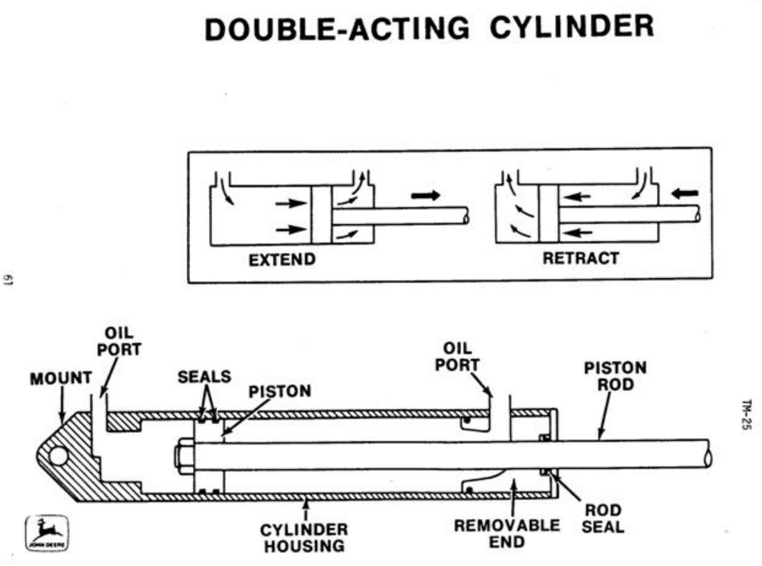

Figure 14: Double Acting Cylinder (http://cast.csufresno.edu, last time visited: 03.01.2008)

of graphical modelling tools. It therefore uses two other frameworks out of the Eclipse-context: The *Eclipse Modelling Framework* (EMF) allows for the definition of concepts using a graphical modelling language and the *Graphical Editor Frameworks* (GEF) provides concepts for defining two-dimensional graphical objects and editors. Despite the fact that the EMF is often used within the context of the development of modelling tools, its application is not only restricted to the description of meta-models.

The modelling tool for the case-study firm has been developed as an Eclipse-Plug-In. A plug-in is a software module offering a standardised interface and can therefore be integrated into the Eclipse platform. The tool allows the selection of a reference model on the basis of a part family. Figure 16 shows a screen-shot of the process model editor allowing the manipulation of concrete diagrams derived from reference process models. The modelling of product families, and sub-contractors, as well as resources and their allocation can be done using a textual editor (cf. Figure 19). The models created with the modelling tool can be used for the optimisation of the corresponding manufacturing processes. We provide an interchange format, which can be imported by two different optimisation tools. This format is based on a XML-DTD covering information from the manufacturing process models, as well as instance information required by the optimisation tools.

Generally, the tool supports three different diagram types:

Process Diagram A *Process Diagram* represents (elementary) processes building a process, as well as the control flow between them. An example is given in Figure 15. Beside processes and control flows, a process model might also contain events.

Process Decomposition Diagram A *Process Decomposition Diagram* describes the composition of aggregated processes on the basis of other processes. An example for a process decomposition is given in Figure 17. A manufacturing process is decomposed into the design and production of a cylinder, as well as administrative processes like accepting and processing a customer order.

Organisational Chart An organisational charts reflects the structure of an organisation. The organisational structure of the case study firm is presented in Figure 18. The managing director

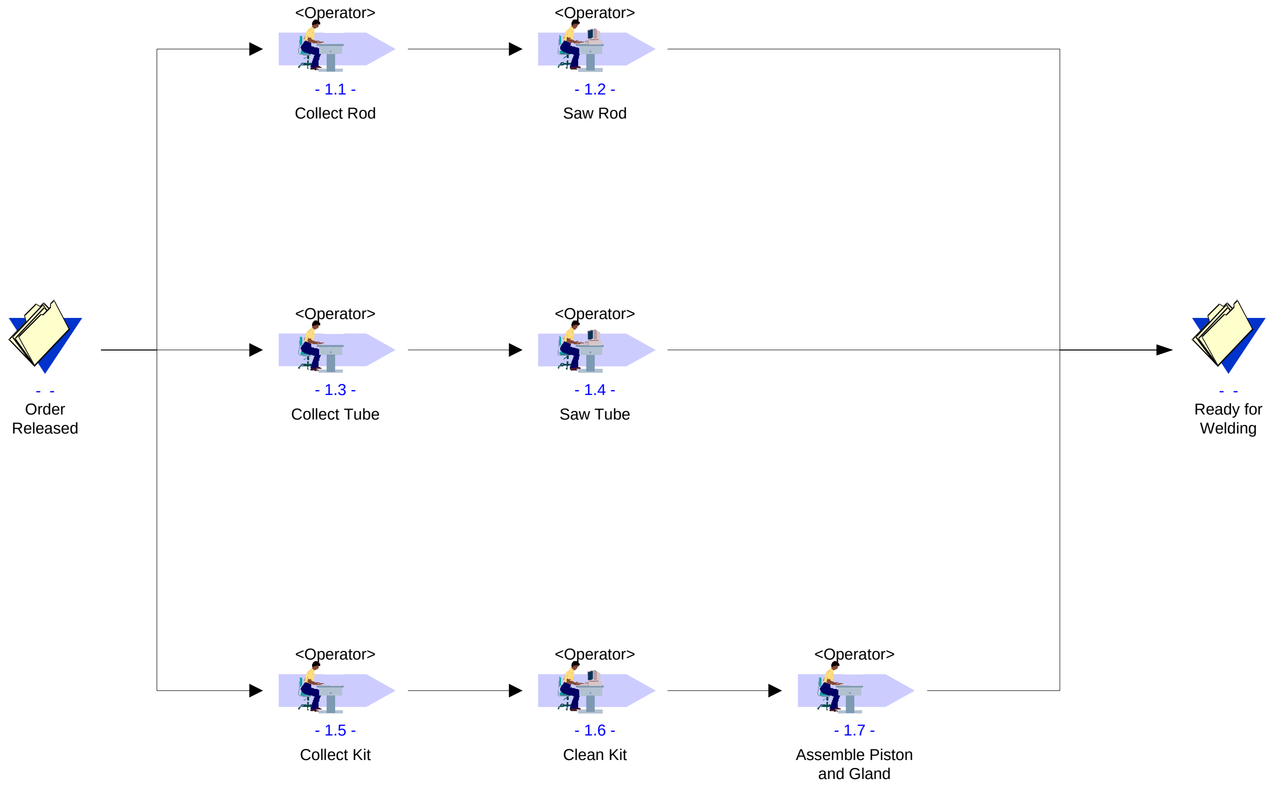

Figure 15: General Reference Model Regarding Cylinder Production

is the supervisor of the purchasing, sales and production department. He is assisted by the finance department an additional administrative staff. The production department is headed by a production manager who is in charge of supervising a designer and several cell leaders. A cell leader is responsible for one given production cell (cf. Figure 13).

BOM Diagram The *bill of material* is modelled using a tree-like structure. Such a structure reflects the composition of a product/component by other items. A generic BOM has been created for each part family.

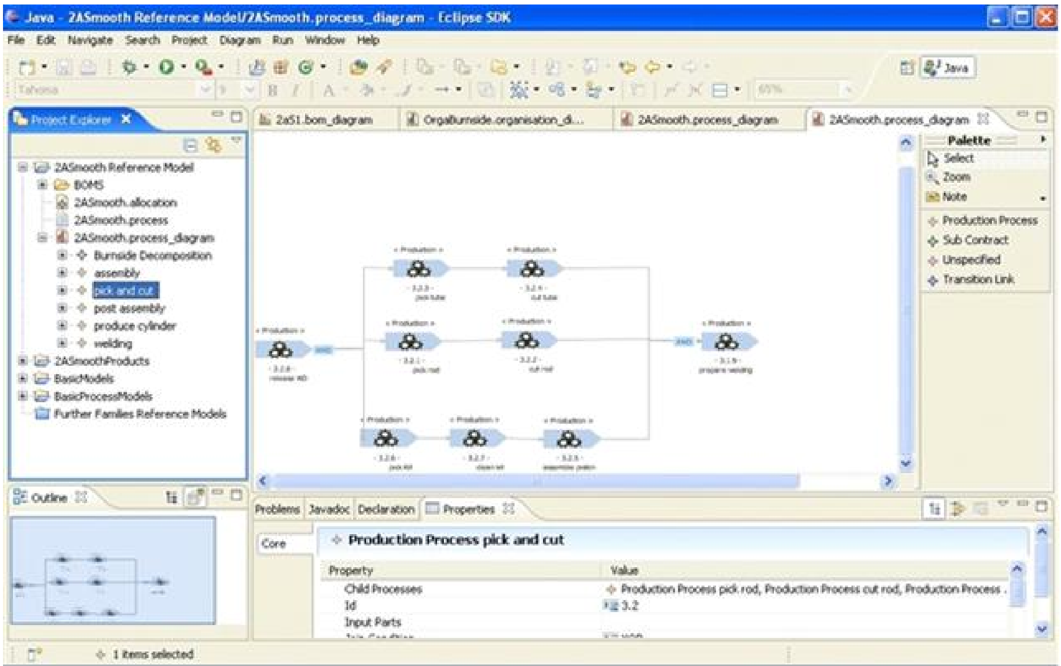

Figure 16: Modelling Tool - Process Diagram

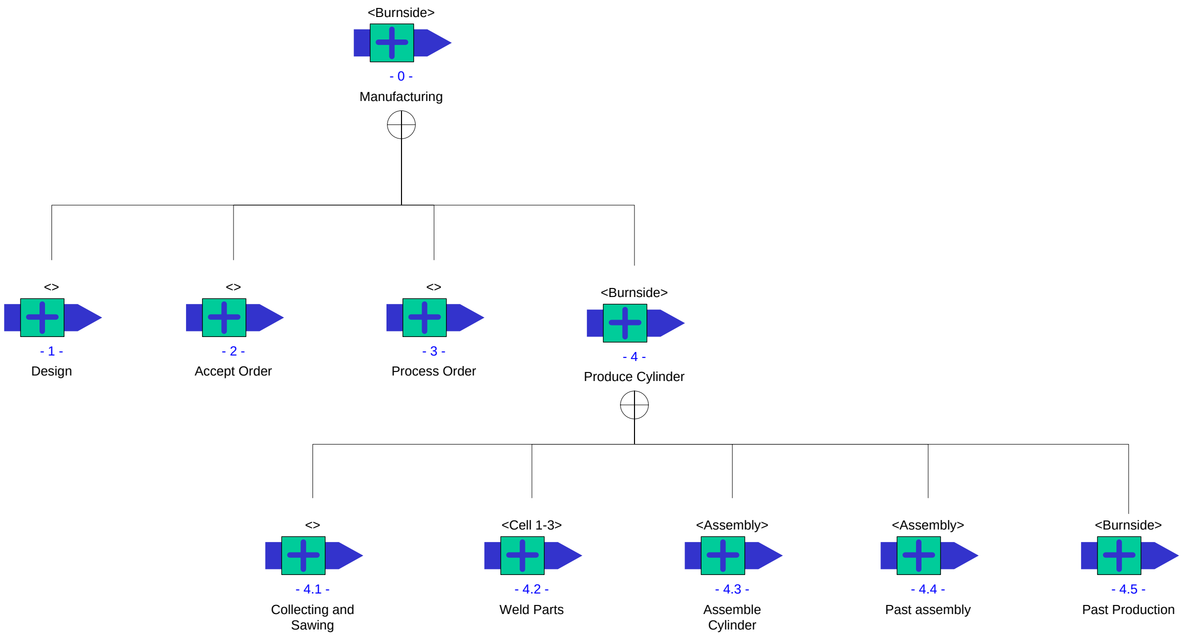

Figure 17: Exemplary Process Decomposition

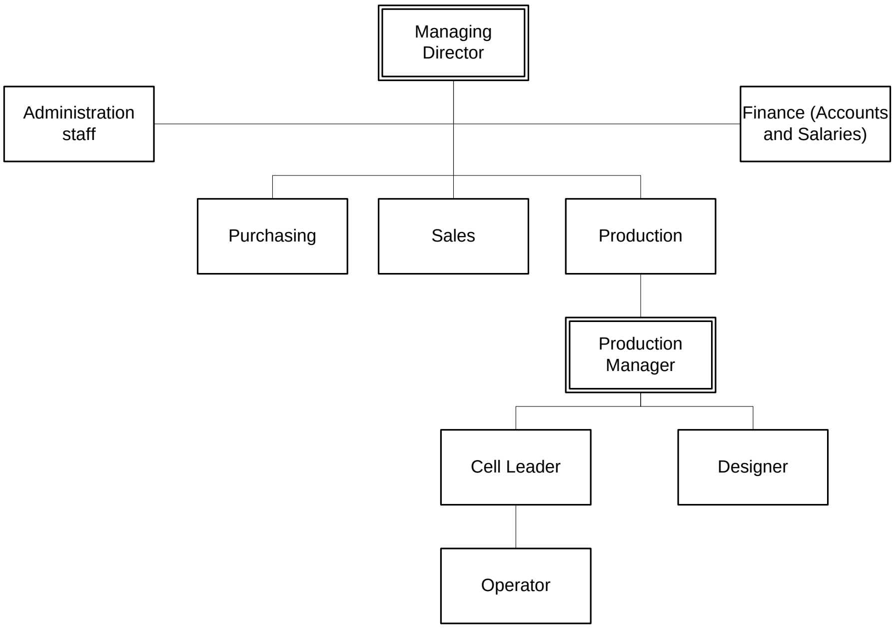

Figure 18: Organisational Chart of the Case Study Firm

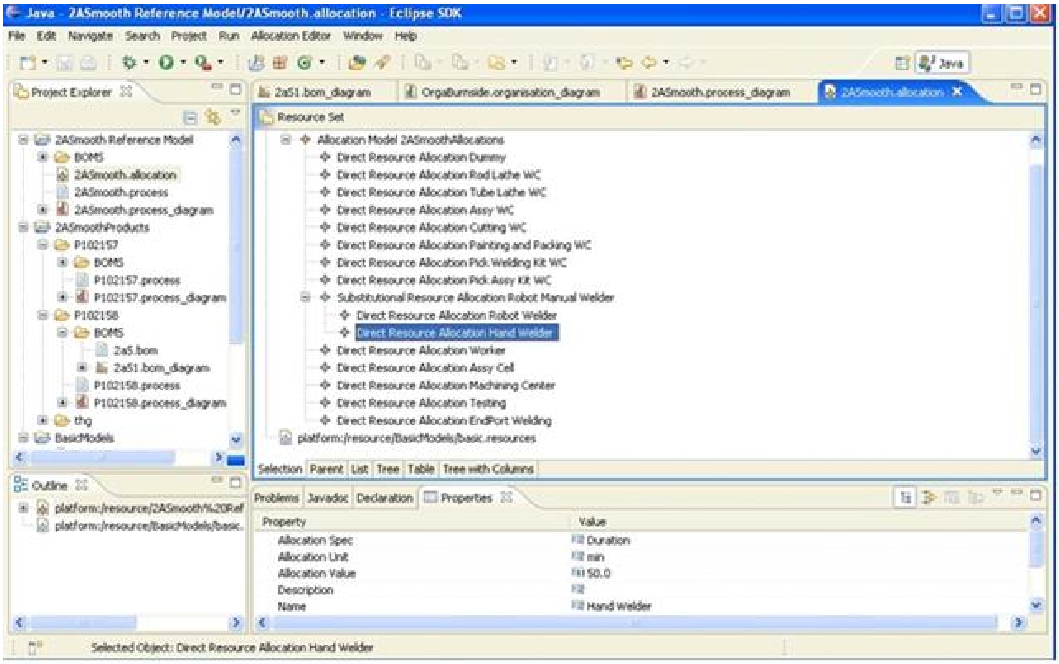

Figure 19: Modelling Tool - Resource Allocation Table

6 Conclusion and Future Research

The report at hand introduces a multi-perspective enterprise modelling method aiming at modelling manufacturing systems. It is motivated by the perception that a model-driven approach and in particular multi-perspective enterprise modelling, is a promising way to handle todays challenges of manufacturing enterprises. Examples for application domains of enterprise models are the integration of heterogeneous software systems, the elimination of redundancies, and in particular fostering communication, as well as analysis and optimisation of manufacturing processes. Domain-specific modelling concepts are introduced, which represent manufacturing-specific concepts for extending an existing modelling language. Furthermore, a reference model-based approach is presented to support production planning and optimisation. Additionally, a prototypical modeling tool supporting manufacturing-specific concepts has been implemented. The development and the evaluation of that tool has been done in the context of a EU funded project supporting the optimisation of manufacturing processes using enterprise models and key performance indicators. The modelling tool proved to be a reasonable tool for modelling manufacturing processes within the project. These models have been further analysed an optimised using external tools. A model interchange format has been defined for transferring graphical models to such external tools. Additionally, our reference model based approach for creating concrete manufacturing process models turned out to be a valuable help especially for people without modelling backgrounds. However, there is a need for further enhancements in order to develop an appropriate manufacturing modelling method:

Language Concepts The introduced concepts for modelling manufacturing systems provide two process types, i. e., manufacturing process for describing in-house manufacturing processes and extended business process for describing sub-contracted processes. Nevertheless, there is a multiplicity of different types of operations within manufacturing, e. g., cutting, welding, turning. Obviously, theses types of operations differ in terms of their applied resources, in particular machines, which play an major role within manufacturing. Thus, further manufacturing process and resource types might be useful to provide more semantics. An issue that will be addressed within future research tasks. Further neglected aspects are quality control and waste, which are of particular relevance within the domain (Cors00; Chat03). Beyond that, events have to be considered in future research. As already indicated in Section 2.1, events (e. g., machine breakdowns) are an essential element for describing enterprises from a dynamic perspective. Our perception is that this is also true in the context of manufacturing processes. Thus, further works have to obtain appropriate event types.

Process Model A modelling method – in short – usually consists of modelling languages and a description of a process model defining activities for creating models. Such a modelling process model is mandatory to guide the modeller through the modelling process by using given modelling languages. A process model also includes concepts supporting the operationalisation of the approach, e. g., transformation, analysis, and process monitoring. An initial step has been done by supporting the generation of a scheduling model for manufacturing enterprises. However, further analyses (e. g., resource untilisation) and transformations (e. g., simulation) have to be researched in the future. This research may contribute to the major challenge of "the complete mathematical modeling of the various processes and their characteristics" (ElMa06, p. 4). Another promising application domain is the model-driven support of PPC systems applying workflow management techniques. Current works addressing this issue seem to lack specific language concepts (e. g., (BeBH00)).

References

[BaCS07] *Barták, R.; Cepek, O.; Surynek, P.*: Modelling Alternatives in Temporal Networks. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007), IEEE Press 2007, S. 129–136.

[BeBH00] *Becker, J.; Bergerfurth, J.; Hansmann, H.*: Methoden zur Einführung Workflow-gestützter Architekturen von PPS-Systemen. Arbeitsbericht, Arbeitsbericht des Instituts für Wirtschaftsinformatik, Nr. 73. Münster 2000.

[BeMS98] *Bernus, P.; Mertins, K.; Schmidt, G*: Handbook on Architectures of Information Systems - International Handbooks on Information Systems. Springer-Verlag, Heidelberg 1998.

[BeNH02] *Becker, J.; Neumann, S.; Hansmann, H.*: Workflow-integrierte Produktionsplanung und -steuerung: Ein Architekturmodell für die Koordination von Prozessen der industriellen Auftragsabwicklung. Arbeitsbericht, Arbeitsbericht des Instituts für Wirtschaftsinformatik, Nr. 85, Münster 2002.

[BeNS03] *Bernus, P.; Nemes, L.; Schmidt, G*: Handbook on Enterprise Architecture – International Handbooks on Information Systems. Springer-Verlag, Heidelberg 2003.

[BoBH01] *Borja, V.; Bell, R.; Harding, J.A.*: Assisting design for manufacture using the data model driven approach. In: Proceedings of the I MECH E Part B Journal of Engineering Manufacture, 215 (2001) 12, S. 1757–1771.

[BoHB01] *Borja, V.; Harding, J.A.; Bell, R.*: A conceptual view on data-model driven reverse engineering. In: Industrial and Production Engineering and Engineering 39 (2001) 4, S. 667–687.

[BoHT00] *Borja, V.; Harding, J.A.; Toh, K.T.K.*: Product re-engineering process using an enterprise modelling architecture. In: International Journal of Agile Management Systems 2 (2000) 3, S. 214–225.

[Chat03] *Chatha, K. A.*: Multi-Process Modelling Approach to Complex Organisation Design. Dissertation, Loughborough University 2003.

[Chat04] *Chatha, K.A.*: Multi-process modelling approach to complex organisation design. Dissertation, Loughborough university 2004.

[ChAW07] *Chatha, K. A.; Ajaefobi, J. O.; Weston, R. H.*: Enriched multi-process modelling in support of the life cycle engineering of Business Processes. In: International Journal of Production Research 1 (2007), S. 103–141.

[Chen76] *Chen, P.*: The Entity-Relationship Model – Toward a Unified View of Data. In: ACM Transactions on Database Systems 1 (1976), S. 9–36.

[ChPo04] *Cheng, K.; Popov, Y.*: Internet-enabled modelling of extended manufacturing enterprises using process-based techniques. In: Journal The International Journal of Advanced Manufacturing Technology 23 (2004), S. 148–153.

[ChWe05] *Chatha, K.A.; Weston, R.H.*: Combined Enterprise and Simulation Modelling in Support of Process Engineering. In: International Journal of Computer Integrated Manufacturing 18 (2005) 8, S. 652–670.

[CoBl98] *Cox, J. F.; Blackstone, J.H.*: APICS Dictionary. American Production and Inventory Control Society (APICS), Ninth Edition, Falls Church 1998.

[Cors00] *Corsten, H.*: Produktionswirtschaft. Oldenbourg, 9. Edition 2000.

[DaKL97] *Dangelmaier, W.; Kuh, A.; Langemann, T.*: Reference models as an approach for the construction of flexible production planning and control systems. In: Proceedings of International Conference on Manufacturing Automation, Hong Kong, 23–30 April 1997 1997.

[DaWa97] *Dangelmaier, W.; Warnecke, H.-J.*: Fertigungslenkung: Planung und Steuerung des Ablaufs der diskreten Fertigung. Springer 1997.

[DaWi93] *Dangelmaier, W.; Wiedenmann, H.*: Modell der Fertigungssteuerung. In: Entwicklungen zur Normung von CIM (1993).

[DeWW+03] *Derks, W.; Weston, R.; West, A. et al.*: Role of workflow management systems in product engineering. In: International Journal of Production Research 41 (2003), S. 652–670.

[DFG97] *DFG*: Arbeitsgruppe Fuzzy Petri Netze – Modellierungstechniken für Fertigungssysteme auf der Grundlage von Petri-Netzen. Gemeinsamer Bericht der Arbeitsgruppe Fuzzy-Petri-Netze im DFG-Schwerpunktprogramm Modellierung der Produktion. Arbeitsbericht, DFG 1997.

[ElMa06] *ElMaraghy, H.A.*: Reconfigurable Process Plans for Reconfigurable Manufacturing. In: 3rd International Conference on Digital Enterprise Technology (DET), Setúbal, Portugal, 18th to 20th, September 2006.

[Fahr95] *Fahrwinkel, U.*: Methoden zur Modellierung und Analyse von Geschäftsprozessen zur Unterstützung des Business Process Reengineering. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn 1995.

[FeLZ06] *Fettke, P.; Loos, P.; Zwicker, J.*: Business Process Reference Models: Survey and Classification. Lecture Notes in Computer Science, Springer Berlin, Heidelberg 2006 S. 469–483.

[Fran94] *Frank, U.*: Multiperspektivische Unternehmensmodellierung 1994.

[Fran98] *Frank, U.*: The MEMO Meta-Metamodel. Arbeitsbericht, Arbeitsberichte des Institut für Wirtschaftsinformatik der Universität Koblenz-Landau, Nr. 9 1998.

[Fran99] *Frank, U.*: Visual Languages for Enterprise Modelling. Arbeitsbericht, Arbeitsberichte des Instituts für Wirtschafts- und Verwaltungsinformatik, Universität Koblenz-Landau, Nr. 18 1999.

[FrLa03] *Frank, U.; van Laak, B.*: Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. Arbeitsbericht, Arbeitsberichte des Instituts für Wirtschaftsinformatik, Universität Koblenz-Landau, Nr. 34 2003.

[FrRi98] *Frank, Ulrich; Rittgen, Peter*: Zur Anreicherung von Modellierungsmethoden mit domänenspezifischem Wissen: Chancen und Herausforderungen der Unternehmensmodellierung. In: *Pohl, Klaus; Schürr, Andy; Vossen, Gottfried (Hrsg.)*: Modellierung, Bd. 9 von *CEUR Workshop Proceedings*, CEUR-WS.org 1998.

[GaGP+97] *Gausemeier, J.; Grasmann, M.; Pusch, R. et al.*: Integrierte Produkt- und Prozessmodellierung. In: VDI-Berichte 1357 (1997).

[GaMO+04] *Gausemeier, J.; Michels, J. S.; Orlik, L. et al.*: Modellierung und Planung von Produktentstehungsprozessen. In: Mechatronischer Systementwurf, VDI-Berichte 1842 (2004).

[Giac99] *Giachetti, R.E.*: A standard manufacturing information model to support design for manufacturing in virtual enterprises. In: Journal of Intelligent Manufacturing 10 (1999), S. 49–60.

[GM99] *Geck-Mügge, K.*: Herleitung und Spezifikation generischer Bausteine zur einheitlichen Modellierung von Fertigungsinformationen für die Fertigungssteuerung. Dissertation, Universität Paderborn, Heinz Nixdorf Institut, Wirtschaftsinformatik, insbesondere CIM, HNI-Verlagsschriftenreihe, Paderborn, Band 48 1999.

[GrPR06] *Gruhn, V.; Pieper, D.; Röttgers, C.*: MDA – Effektives Softwareengineering mit UML2 und Eclipse. Xpert.press 2006.

[HaYe92] *Hastings, N.A.J.; Yeh, C.-H.*: Bill of manufacture. In: Production and Inventory Management Journal 33 (1992) 4, S. 27–31.

[Holt97] *Holtkamp, R.*: Modellierungswerkzeug OOPUS: Modellausführung OOPUS/X. Beuth, Berlin 1997.

[JiHe06] *Jiao, J.X.; Helo, P.T.*: Process Platform Representation based on Unified Modeling Language. In: International Journal of Production Research 17 (2006) 6, S. 323–350.

[JiLP07] *Jiao, J.X.; Liafeng, Z.; Pokharel, S.*: Process Platform Planning for Variety Coordination from Design to Production in Mass Customization Manufacturing. In: IEEE Transactions on Engineering Management 54 (2007) 1.

[JiTM00] *Jiao, J.X.; Tseng, M.M.; Ma, Q. Zou, Y.*: Generic Bill-of-Materials-and-Operations for High-Variety Production Management. In: Concurrent Engineering 8 (2000) 4, S. 291–322.

[Jung03] *Jung, Jürgen*: Basic Conceptualisation of a Resource Modelling Language. In: Information Technology and Organisations – Proceedings of the 2003 Information Resource Management Association International Conference, Philadelphia (PA) (Mai 2003), S. 827–831.

[Jung06] *Jung, J.*: Supply Chains in the Context of Resource Modelling. Arbeitsbericht, ICB Research Report, Nr. 5 2006.

[Jung07] *Jung, J.*: Entwurf einer Sprache für die Modellierung von Ressourcen im Kontext der Geschäftsprozessmodellierung. Dissertation, Universitaet Duisburg Essen 2007.

[KeNS92] *Keller, G.; Nüttgens, M.; Scheer, A. W.*: Semantische Prozessmodellierung auf der Grundlage" ereignisgesteuerter Prozessketten (epk). Iwi 1992.

[Kirc07] *Kirchner, L.*: Entwurf einer Modellierungssprache zur Unterstützung der Aufgaben des IT-Managements: Grundlagen, Anforderungen und Metamodell. Arbeitsbericht, ICB Research Report, Nr. 11 2007.

[Kuhn97] *Kuhn, A.*: Referenzmodelle für Produktionsprozesse zur Untersuchung und Gestaltung von PPS-Aufgaben. Dissertation, Universität GH Paderborn 1997.

[Lang99] *Langemann, T.*: Modellierung als Kernfunktion einer systemorientierten Analyse und Bewertung der diskreten Produktion. Dissertation, Universität-GH Paderborn 1999.

[Loos96] *Loos, P.*: Workflow und industrielle Produktionsprozesse. Arbeitsbericht, Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 123 1996.

[MeMa98] *Menzel, M.; Mayer, R.J.*: The IDEF Family of Languages. In: Handbook on Architectures of Information Systems – International Handbooks on Information Systems, P. Bernus et al., Springer-Verlag, Heidelberg (1998), S. 209–242.

[MoBe99] *Molina, A.; Bell, R.*: A manufacturing model representation of a flexible manufacturing facility. In: Proceedings of the I MECH E Part B Journal of Engineering Manufacture 213 (1999) 3, S. 225–246.

[PiSK+04] *Piller, F.; Schubert, P.; Koch, M. et al.*: From Mass Customization to Collaborative Customer Co-Design. In: Proceedings of the European Conference on Information Systems (ECIS 2004), June 14–16, Turku, Finland (2004).

[PoLO99] *Podorozhny, Rodion M.; Lerner, Barbara Staudt; Osterweil, Leon J.*: Modeling Resources for Activity Coordination and Scheduling. In: *Ciancarini, P.; Wolf, A. L. (Hrsg.)*: Proceedings of the 3rd International Conference on Coordination Models and Languages (Coordination 1999), Nr. 1594 in Lecture Notes in Computer Science, Springer, Berlin, Heidelberg 1999, S. 307–322.

[RaWe05] *Rahimifard, A.; Weston, R.H.*: Integrated Use of Enterprise and Simulation Modelling to Achieve Improved Planning and Control of Multiple Product Flows. In: CIRP 3rd International Conference on Reconfigurable Manufacturing, (RMS 2005) Ann Arbor, MI, USA 2005.

[RuHo96] *Rudas, I.J.; Horvath, L.*: Modelling of Manufacturing Processes Using a Petri-net Representation. In: Engineering Applications of Artificial Intelligence 10 (1996) 3, S. 243–255.

[RuJB99] *Rumbaugh, J.; Jacobson, I.; Booch, G.*: The Unified Modeling Language Reference Manual. Addsion Wesley 1999.

[ScBa95] *Scholz-Reiter, B.; Bastian, d.*: Kooperative Unternehmensmodellierung – Ein Groupeware-basierter Ansatz zur Integration von Geschäftsprozess- und Produktionsmodellierung. In: Zeitschrift für wirtschaftlichen Fabrikbetrieb 8 (1995), S. 385–386.

[Sche99] *Scheer, August-Wilhelm*: ARIS: Business Process Modeling. 2. Aufl., Springer, Berlin et al. 1999.

[ScHK06] *Scholz-Reiter, B.; Hildebrandt, T.; Kolditz, J.*: A Modelling Methodology for Control Processes of Autonomous Production Systems. In: Proceedings of the 8th International Conference on The Modern Information Technology in the Innovation Processes of the Enterprises Budapest, Hungary (2006), S. 317–322.

[Schm99] *Schmidtmann, A.*: Eine Spezifikationssprache für die Fertigungslenkung. Dissertation, Universität-GH Paderborn 1999.

[Schn96] *Schneider, U.*: Ein formales Modell und eine Klassifikation für die Fertigungssteuerung – Ein Beitrag zur Systematisierung der Fertigungssteuerung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn 1996.

[ScKH+06a] *Scholz-Reiter, B.; Kolditz, J; Hildebrandt, T. et al.*: Modellierung selbststeuerender logistischer Prozesse in der Produktion In: Herausforderungen in der Wirtschaftsinformatik. Festschrift für Hermann Krallmann. Springer Berlin 2006 S. 221–236.

[ScKH06b] *Scholz-Reiter, B.; Kolditz, J.; Hildebrandt, T.*: UML as a Basis to Model Autonomous Production Systems. In: Proceedings of the 3rd CIRP Sponsored Conference on Digital Enterprise Technology Hrsg.: Cunha, P. F. and Maropoulos, P. G. Setúbal, Portugal (2006), S. 1–8.

[ScWK+04] *Scholz-Reiter, B.; Windt, K.; Kolditz, J. et al.*: New Concepts of Modelling and Evaluating Autonomous Logistic Processes. In: Proceedings of IFAC Conference on Manufacturing, Modelling, Management and Control, Athens, Greece 2004.

[SeTa97] *Senechal, O.; Tahon, C.*: A modelling approach for production costing and continuous improvement of manufacturing processes. In: Production Planning and Control 8 (1997) 8, S. 731–742.

[SoZa92] *Sowa, J.F.; Zachman, John A.*: Extending and formalizing the framework for information systems architecture. In: IBM Systems Journal 31 (1992) 3, S. 590–616.

[Vern93] *Vernadat, F.*: CIMOSA: Enterprise modelling and enterprise integration using process-based approach. In: H. Yoshikawa, J. Goosseneaerts (eds.), Information Infrastructure Systems for Manufacturing, Amsterdam 1 (1993) 1, S. 65–84.

[Vern96] *Vernadat, F.*: Enterprise Modelling and Integration: principles and Application. Chapman & Hall, London 1996.

[Vern98] *Vernadat, F.*: The CIMOSA Languages. In: Handbook on Architectures of Information Systems – International Handbooks on Information Systems, P. Bernus et al., Springer-Verlag, Heidelberg (1998), S. 243–263.

[Wien97] *Wiendahl, H.P.*: Betriebsorganisation für Ingenieure. Hanser Verlag, Wien 1997.

[WiLu02] *Wiendahl, H.P.; Lutz, S.*: Production in Networks. In: Annals of CIRP 51 (2002) 2, S. 1–14.

[WoHG01] *Wortmann, J.C.; Hegge, H.M.H.; Goossenaerts, J.B.M.*: Understanding enterprise modeling from product modeling. In: Production Planning and Control 12 (2001), S. 234–244.

[Yeh95] *Yeh, C.-H.*: Production data modelling: an integrated approach. In: International Journal of Operations and Production Management 15 (1995) 8, S. 52–62.

[Zach87] *Zachman, John A.*: A framework for information system architecture. In: IBM Systems Journal 26 (1987) 3, S. 277–293.

[Zele97] *Zelewski, S.*: Elektronische Maerkte zur Prozesskoordinierung in Produktionsnetzwerken. In: Wirtschaftsinformatik 39 (1997), S. 231–243.

[Zimm97] *Zimmermann, A.*: Modellierung und Bewertung von Fertigungssystemen mit Petri-Netzen. Dissertation, Technischen Universität Berlin 1997.

Previously published ICB - Research Reports

2008

No 22 (Januar 2008)
Heymans, Patrick; Kang, Kyo-Chul; Metzger, Andreas, Pohl, Klaus (Eds.): "Second International Workshop on Variability Modelling of Software-intensive Systems"

2007

No 21 (September 2007)
Eicker, Stefan; Annett Nagel; Peter M. Schuler: "Flexibilität im Geschäftsprozess-management-Kreislauf"

No 20 (August 2007)
Blau, Holger; Eicker, Stefan; Spies, Thorsten: "Reifegradüberwachung von Software"

No 19 (June 2007)
Schauer, Carola: "Relevance and Success of IS Teaching and Research: An Analysis of the ‚Relevance Debate'

No 18 (May 2007)
Schauer, Carola: "Rekonstruktion der historischen Entwicklung der Wirtschaftsinformatik: Schritte der Institutionalisierung, Diskussion zum Status, Rahmenempfehlungen für die Lehre"

No 17 (May 2007)
Schauer, Carola; Schmeing, Tobias: "Development of IS Teaching in North-America: An Analysis of Model Curricula"

No 16 (May 2007)
Müller-Clostermann, Bruno; Tilev, Milen: "Using G/G/m-Models for Multi-Server and Mainframe Capacity Planning"

No 15 (April 2007)
Heise, David; Schauer, Carola; Strecker, Stefan: "Informationsquellen für IT-Professionals – Analyse und Bewertung der Fachpresse aus Sicht der Wirtschaftsinformatik"

No 14 (March 2007)
Eicker, Stefan; Hegmanns, Christian; Malich, Stefan: "Auswahl von Bewertungsmethoden für Softwarearchitekturen"

No 13 (February 2007)
Eicker, Stefan; Spies, Thorsten; Kahl, Christian: "Softwarevisualisierung im Kontext serviceorientierter Architekturen"

No 12 (February 2007)
Brenner, Freimut: "Cumulative Measures of Absorbing Joint Markov Chains and an Application to Markovian Process Algebras"

No 11 (February 2007)
Kirchner, Lutz: "Entwurf einer Modellierungssprache zur Unterstützung der Aufgaben des IT-Managements – Grundlagen, Anforderungen und Metamodell"

No 10 (February 2007)
Schauer, Carola; Strecker, Stefan: "Vergleichende Literaturstudie aktueller einführender Lehrbücher der Wirtschaftsinformatik: Bezugsrahmen und Auswertung"

No 9 (February 2007)
Strecker, Stefan; Kuckertz, Andreas; Pawlowski, Jan M.: "Überlegungen zur Qualifizierung des wissenschaftlichen Nachwuchses: Ein Diskussionsbeitrag zur (kumulativen) Habilitation"

No 8 (February 2007)
Frank, Ulrich; Strecker, Stefan; Koch, Stefan: "Open Model - Ein Vorschlag für ein Forschungsprogramm der Wirtschaftsinformatik (Langfassung)"

2006

No 7 (December 2006)
Frank, Ulrich: "Towards a Pluralistic Conception of Research Methods in Information Systems Research"

No 6 (April 2006)
Frank, Ulrich: "Evaluation von Forschung und Lehre an Universitäten – Ein Diskussionsbeitrag"

No 5 (April 2006)
Jung, Jürgen: "Supply Chains in the Context of Resource Modelling"

No 4 (February 2006)
Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part III – Results Wirtschaftsinformatik Discipline"

2005

No 3 (December 2005)
Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part II – Results Information Systems Discipline"

No 2 (December 2005)
Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part I – Research Objectives and Method"

No 1 (August 2005)
Lange, Carola: „Ein Bezugsrahmen zur Beschreibung von Forschungsgegenständen und -methoden in Wirtschaftsinformatik und Information Systems"

Research Group	Core Research Topics
Prof. Dr. H. H. Adelsberger Information Systems for Production and Operations Management	E-Learning, Knowledge Management, Skill-Management, Simulation, Artificial Intelligence
Prof. Dr. P. Chamoni MIS and Management Science / Operations Research	Information Systems and Operations Research, Business Intelligence, Data Warehousing
Prof. Dr. F.-D. Dorloff Procurement, Logistics and Information Management	E-Business, E-Procurement, E-Government
Prof. Dr. K. Echtle Dependability of Computing Systems	Dependability of Computing Systems
Prof. Dr. S. Eicker Information Systems and Software Engineering	Process Models, Software-Architectures
Prof. Dr. U. Frank Information Systems and Enterprise Modelling	Enterprise Modelling, Enterprise Application Integration, IT Management, Knowledge Management
Prof. Dr. M. Goedicke Specification of Software Systems	Distributed Systems, Software Components, CSCW
Prof. Dr. R. Jung Information Systems and Enterprise Communication Systems	Process, Data and Integration Management, Customer Relationship Management
Prof. Dr. T. Kollmann E-Business and E-Entrepreneurship	E-Business and Information Management,E-Entrepreneurship/E-Venture, Virtual Marketplaces and Mobile Commerce, Online-Marketing
Prof. Dr. B. Müller-Clostermann Systems Modelling	Performance Evaluation of Computer and Communication Systems, Modelling and Simulation
Prof. Dr. K. Pohl Software Systems Engineering	Requirements Engineering, Software Quality Assurance, Software-Architectures, Evaluation of COTS/Open Source-Components
Prof. Dr.-Ing. E. Rathgeb Computer Networking Technology	Computer Networking Technology
Prof. Dr. A. Schmidt Pervasive Computing	Pervasive Computing, Uniquitous Computing, Automotive User Interfaces, Novel Interaction Technologies, Context-Aware Computing
Prof. Dr. R. Unland Data Management Systems and Knowledge Representation	Data Management, Artificial Intelligence, Software Engineering, Internet Based Teaching
Prof. Dr. S. Zelewski Institute of Production and Industrial Information Mangement	Industrial Business Processes, Innovation Management, Information Management, Economic Analyses

For more information visit us on the Web: **http://www.icb.uni-due.de**

ISSN 1860-2770