IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4587

Basic Conceptualisation of a Resource Modelling Language

Dipl.-Inform. Juergen Jung
University of Koblenz, Universitaetsstrasse 1, 56070 Koblenz, Germany
phone: +49 261 287 2554, fax: +49 261 287 2521, e-mail: jjung@uni-koblenz.de

ABSTRACT

This paper presents the basic conceptualisation of a resource modelling language for business process modelling. This language is an extension to an existing process modelling language - MEMO-OrgML. The basic conceptualisation comprises basic resource types and their mutual relations. The resource modelling language aims at satisfying the needs of different user types. The language will be applicable by domain experts and offer domain specific resources and resource types for various domains by offering adequate abstraction on resources. Potential benefits of such a resource modelling language are outlined by an example.

1 MOTIVATION

The analysis, representation and management of knowledge related to an organisation and its processes have always been very important (Koubarakis and Plexousakis 2000). A lot of work has been done on the development and evaluation of ontologies for process modelling (Wand and Weber1989, Wand and Weber 1990a, Wand and Weber 1990b, Wand and Weber 1993, Weber 1997, Green and Rosemann 1999), the specification of process modelling languages (Eertink et al. 1999, Oberweis 1996, Sutton and Osterweil 1997, van der Aalst and van Hee 2002) as well as on business process modelling methods and concepts (Herbst 1997, Österle 1995). Business process models can be used for different purposes:

- documentation of processes of an organisation to foster communication (Frank 1999, Oberweis 1996)
- analysis of business processes (Eertink et al. 1999, Bergholtz and Johanneson 2001, Scheer 1999)
- simulation of processes (Baumgarten 1996)
- support for business process re-engineering (Curtis et al. 1992, Oberweis 1996)
- generation of workflow schemata (Curtis et al. 1992, Oberweis 1996)
- software development of process-oriented applications (Frank 1999, Scheer 1998, Scheer 1992, Österle 1995, Curtis et al. 1992)

The documentation of processes found in organisations (as well as other organisational aspects like structure or strategy) enables communication with recently hired employees or external consultants (Frank 1999, Oberweis 1996). An analysis of business processes relies on well formulated process descriptions. An analysis might help to find weaknesses in traditional processes (Eertink et al. 1999, Bergholtz and Johanneson 2000, Scheer 1999) and a business process re-engineering might follow (Curtis et al. 1992, Oberweis 1996). Simulation also supports the detection of weaknesses (Baumgarten 1996). In contrast to analysis, simulation works on prototypical instantiations of a process model. Typical instances of objects and their values (e.g. a specific processing time or accounting information) are added to the model and the execution of a process can be observed. Business process re-engineering supports the redesign of processes with respect to weaknesses identified by an analysis or a simulation (Curtis et al. 1992, Oberweis 1996). Business process models may also be a preliminary stage for an information system (IS) design. A workflow-management-system (WfMS) or proprietary corporate applications are alternatives for such an IS (Curtis et al. 1992, Frank 1999, Österle 1995, Scheer 1992, Scheer 1998).

Resources are essential for the modelling of processes (Podorzhny et al. 1999). Processes and their relationships only describe *what* has to be done. Resources assigned to processes specify *who* has to work on the process and *what* will be needed. Usually resources are not available in an unlimited quantity (Nübel 2001, Podorzhny et al. 1999). Modelling resources offers the opportunity to determinate the economic efficiency of a process. Hence, the usage of scarce resources has to be taken into account during the analysis or simulation of processes as well as the development of a workflow application or an information system. One application for the "rich in content" modelling of resources is business continuity planning (BCP). Bottlenecks resulting from scarce resources can be identified and supported by alternative resources which may replace the original resources in case of a failure.

The quality of analysis, simulation and system development depends on the conceptual power of the resource modelling language. Such a language should offer:

- domain specific concepts to be (re-)used by domain experts
- semantically rich resource types including integrity constraints
- support for analysis and simulation
- mappings to information systems

Many process modelling languages have been developed in the last years. Many of them also include the specification of resources on different levels of abstraction. Petri-nets offer a formal language for the specification of processes (Baumgarten 1996). In classical Place/Transition-Nets, resources (i.e. their states) can only be modelled by a subnet. Higher-level Petri-nets offer an extended expressiveness by using tuples as markings (instead of anonymous markings like classical Petri-nets). Oberweis (1996) developed a language based on Petri-nets but using complex relations as markings. Nonetheless these approaches only offer formal language features for the descriptions of resources. The term *resource* does not exist explicitly. Also Scheer's event-driven process chains (EPC) do not include dedicated resources (Scheer 1992, Scheer 1998, Scheer 1999). All entities participating at a business process have to be modelled by an entity-relationship-diagram (ERD). The current version of the Unified Modelling Language (UML) provides features for business process modelling (Bennet et al. 1999, Marshall 2000, Eriksson and Penker 2000). Like in all previously presented approaches, resources are not a dedicated language feature and have to be added by the user.

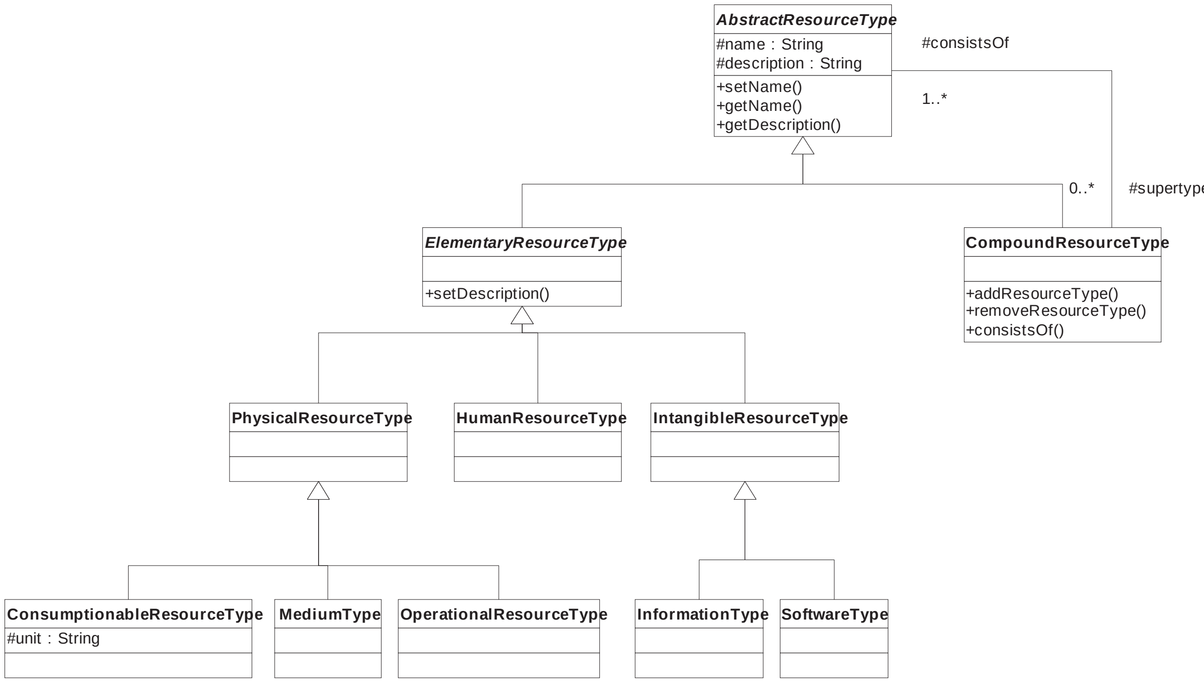
This paper represents a current work on the development of a resource modelling language for business processes. This language aims to satisfy the needs of different user types. The resource modelling language is an extension to an existing process modelling language - OrgML (Organisation Modelling Language) - which is part of an integrated enterprise modelling method called MEMO (Multi-perspective Enterprise MOdelling) (cf. Frank 1999).

2 DIFFERENT LEVELS OF ABSTRACTION

Many everyday terms are ambiguous with respect to their level of abstraction and can be interpreted accordingly. The term *resource* might be interpreted as *resource type* (i.e. a *resource class*) or just as likely as an instance of resource type (e.g. an existing database server in the corporate network). Ambiguities of interpretations should be avoided in the context of business process modelling to foster formal analysis, simulation or software development. Furthermore, concepts of the hereby described resource modelling language will be integrated into the existing process modelling language MEMO-OrgML.

Language features of a resource modelling language are specified by a meta-model. Instances of types in the meta-model are resource types such as a resource type representing a *database server*. The resource type *database server* may be classified from a technical or economical point of view. Technical

Figure 1: Part of the Meta-Model of Basic Resource Types



aspects are computer models, processor speed and maximum number of transactions per minute. These aspects determine the restrictions of usage for such a resource type within a certain context. Economic properties of a *database server* resource type address cost of operation and maintenance of a specific instance in a business process. Those aspects are described at different levels of abstraction by the same resource type.

A specific resource 'Oracle9i-server' is an instance of the resource type *database server* and can in turn also have instances (i.e. concrete installations of Oracle corresponding to a license). Consequently we have a multi-level type-instance-relationship for the description of database servers in a resource model. This relationship however is hard to handle when using business process models as the basis for information systems development.

Conceptual modelling usually prescinds from concrete objects (i.e. instances) and changeable aspects. Hence, only types are allowed in conceptual models. Nevertheless modelling of instances might be appropriate in certain situations:

- The modelling of - anonymous - instances is used to formulate relationships between resources in different processes. Such a relationship may be of the kind: The resource (instance) used in process A is also required in process B.
- A prototypical instance in a process model reflects the average property-values of a resource type. Costs for the usage of a resource are usually attached to the concrete resource instance and differ between instances. Hence, costs can not be assigned to resource types in conceptual modelling. But average costs of a certain resource type might be assigned to a prototypal instance which is used for formal analysis and simulation.
- Concrete instances are required to describe currently used resources. The first step of business process re-engineering is the modelling of existing business processes. This model includes processes and resources as they exist at the time being. This might also include dedicated resources like special servers, relevant machinery or technical engineers.

Hence, types and instances have to be modelled by a resource modelling language. This language should also allow different kinds of abstraction (technical and economical). Furthermore different levels of abstraction have to be available with respect to type-instance-relationships. We will present our first approach for resource modelling in the following section. This approach will cover some of the discussed aspects.

3 RESOURCE TYPES

The resource modelling language is specified by a meta-model. Such a meta-model defines basic concepts of the language, relationships between language elements and integrity constraints. An extract of the meta-model of the resource specification language is presented by the UML class-diagram in Figure 1.

3.1 Basic Meta-Model

At the top level of the type hierarchy we distinguish between compound and elementary resource types. A compound resource type is an abstract resource type, which is composed by other abstract resource types. Hence, a compound resource type may consist of several elementary or compound resource types. The design of compound resource types follows the *composite pattern* by (Gamma et al. 1998). Elementary resource types are specialised to human, physical and intangible resource types. A human resource type corresponds to an organisational unit or a role filled by an employee. Physical resource types comprise all tangible objects used within a business process. The distinction between human and physical resource types bases on the differentiation of human work (labour) on the one hand and man-made aids to further production on the other hand (Diederich 1992, Chrystal and Lipsey 1997). Subtypes of physical resource types are

- used for the completion of a process (OperationalResourceType),
- consumed by the completion of a process (ConsumptionableResourceType) or
- able to store information (MediumType).

Operational resource types are physical resource types, which are used within a process and are still available after its completion (Diederich 1992). Examples are machinery, tools and vehicles. They are all used for processing but remain available. In contrast to this, consumable resource types are a prerequisite for a process and are transformed during the execution of a process. Raw material as well as spare parts (both consumptionable resource types) are used by a business process and are transformed to a (partial) product (Chrystal and Lipsey 1997). They (individually) will not be available for other processes. Information containing media does not fit the differentiation between operational and consumable resource types. A medium or its information might be out-of-date after the completion of a process or it might be a prerequisite for subsequent processes.

Intangible resource types represent all resource types, which are neither physical nor human. Two subtypes have been identified up to now - information and software. They can not exist on their own, because they need a medium for representation - persistent or for transmission. The classification of software as an intangible resource is controversial and therefore problematic. Software is usually not regarded as an intangible resource or asset like human qualification, knowledge or business relationships. But from a computer scientist's point of view or in the context of project-management, software is no physical object; it is intangible. Hence, the current classification has only been done for evaluation purposes and will be changed whenever it seems not to be appropriate.

3.2 Information Types

Information types correspond to some kind of knowledge of an organisation (Chrystal and Lipsey 1997). Information types might be order, assembly list or invoice (Jung and Fraunholz 2002). Furthermore an information type reflects explicit knowledge, guidelines and regulations for the execution of business processes. All these kinds of information are covered by InformationType in the meta-model in Figure 2. An information type is characterised by its structure, as described by the following examples:

- An invoice consists of a date, sender, receiver and several positions.
- A guideline is structured by its format, the corresponding applicability and its contents.

This structure is encapsulated by the attribute structureDefinition. This definition is independent of a special data definition or medium. Information might be structured by a formal data definition but does not have to be. Consequently the structure definition of an information type is independent of a formal data definition. Such a data definition might be an XML-DTD, an SQL table definition or another structure definition language. If there is a formal structure definition, it might be assigned to the information type. Formal data-format type definitions are covered by DataFormatType in Figure 2. Every data-format type consists of the name of the format (formatType), the format's version (formatTypeVersion) and its formal specification (formatTypeDefinition).

3.3 Media Types

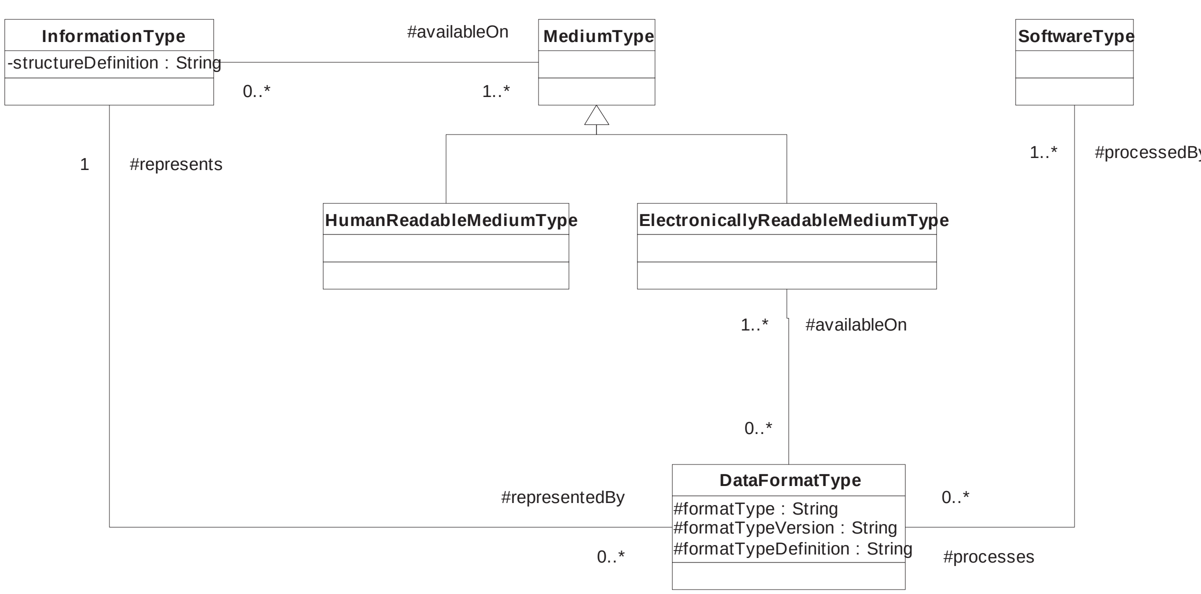
Information and data-format types represent a non-physical kind of resource. Every information type has to be stored on a medium representing the according information. We distinguish between human and electronically readable media in this paper. The distinction between those kinds of media is of special importance for business process modelling. Human readable media correspond to all representations of information to be interpretable by human readers. Examples for such kinds of documents are a printed invoice, a paper-based product catalogue or any kind of paper-based documents. Information is represented in a form that humans are able to further process them. Electronically readable media are those which are based on an IT-system and allow a further processing by an information system. The conceptualisation of media types for the resource modelling language for the MEMO-OrgML is displayed in Figure 2. The class HumanReadableMediumType comprises all media readable by humans but not necessarily by automated information systems. This includes media-types like printed documents. ElectronicallyReadableMediumType comprises all kinds of media which are readable by information system components. Storage of information on an electronically readable medium is only allowed if a data-format for this information exists. Such a differentiation between human and electronically readable media will not be unambiguous for the future specification of resources. Some media are interpretable by human readers and IT-systems at the same time. Future research will prove or disprove adequacy of this classification.

Generally speaking, an information type does not change its readability with respect to human readers or information systems. All kind of readability might be reduced to the media containing information. The medium type determines the kind of reader: human readable media are restricted to the processing by humans and electronically readable media to information systems. Examples for human readable media are every kind of representation which maps formal structures of information types to a representation adequate to a human readable medium. Electronically readable media correspond to an information system containing a structured representation of information. Both representations base on the same type of information and its corresponding structure as presented in paragraph 3.2. The same kind of information might be represented on a human or electronically readable medium. Differences usually are based on the kind of media. Main goal of the differentiation of human and electronically readable media is the recognition of media clashes.

4 FURTHER CONCEPTS

This paper only presents a small subset of the meta-model of the resource modelling language of MEMO-OrgML. Further language features concern concrete resources related to types, economic aspects of resources and the allocation of resources to business processes (Jung and Fraunholz 2002). Concrete resources differ from general resource types (e.g. a database-management-system (DBMS)). They rather describe concrete instances of resource types. A DBMS developed by a specific manufacturer has as type the DBMS resource type but includes additional properties: product name, product version and manufacturer as well as system requirements. These properties do not necessarily apply to a general resource type DBMS. For example DB/2 version

Figure 2: Part of the Meta-Model of Information and Media Resource Types



7.1 by IBM is a concrete resource of the type DBMS.

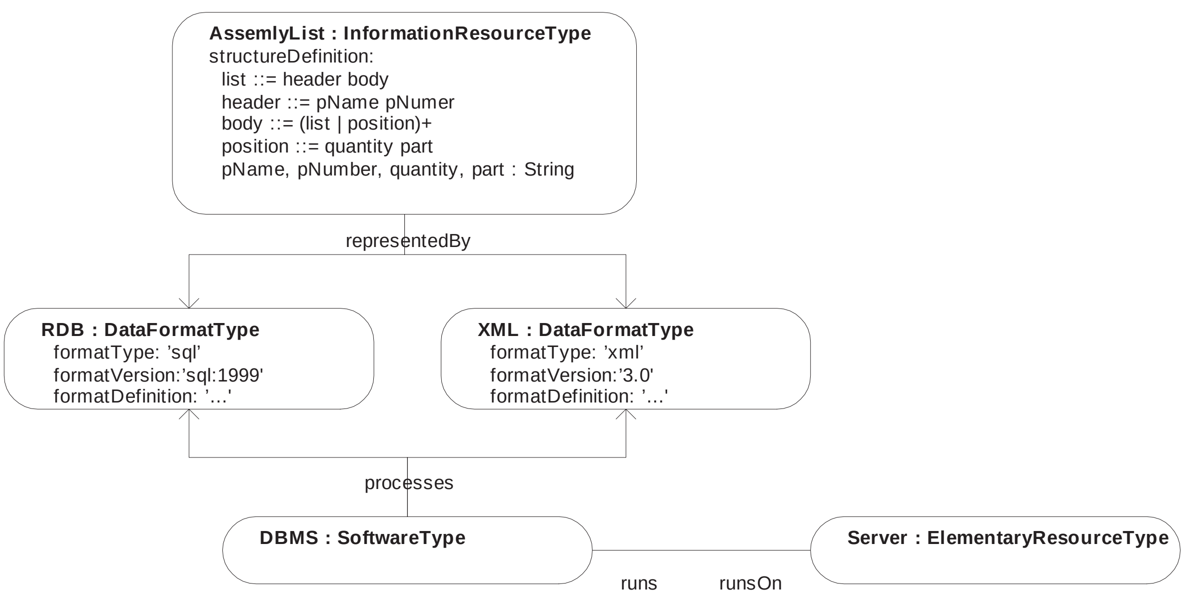
Economic aspects are very important properties of resources and resource types for the analysis and simulation of different process alternatives. Hence, the resource modelling language of the MEMO-OrgML includes the specification of costs related to the usage of a resource. This supports the mapping of different alternatives to monetary values. Beside other aspects like customer satisfaction, costs for the processing of tasks determine the quality of processes for an enterprise (Österle 1995). Additionally the allocation of resources to processes has to be taken into account. The allocation of resources determines the extent of a resource's use. The resource modelling language of MEMO-OrgML also includes this feature.

5 EXAMPLE

The application of the resource modelling language is demonstrated by an example given in Figure 3. An assembly list of a specific product type and its according data format types are presented. Additionally a database server running a DBMS managing the assembly list is modelled. An assembly list is a list of all parts of a physical product. Each part may either be an elementary part or a complex part specified by its own assembly list. The assembly list in Figure 3 is of the type InformationResourceType and therefore specified by a structure definition. This definition (structureDefinition in the rounded box labelled AssemblyList:InformationResourceType) is formulated in an abstract manner using the Extended Bacchus-Naur-Form (EBNF). The assembly list is stored in a corporate information system using a relational database and an XML document. The relational schema is represented by the data format type RDB and the document type definition expressed in XML. Both electronic incarnations of an assembly list are processed by a DBMS running on a dedicated server of the kind ElementaryResourceType. Benefits of such a modelling of the assembly list are the documentation of document structures, association to required resources and the specification of exchange formats.

An abstract definition of the structure of an information type fosters communication on this information type. Different communication partners can base their discussion on information types on a common definition. The specification of the assembly list might act as a reference for engineers and business administrators. It is also a specification for external partners producing subparts. The association with data formats like RDB or XML leads to a correspondence of an assembly list to computerised formats. These formats are not only used within a process but might also be used for the interaction with other information systems. Such information systems include other corporate information systems and systems of external partners as well. Data format type definitions are, thus, used as an inter-organisational reference system. They specify interfaces for the exchange of data.

Figure 3: Example for the Usage of Information Resource Types



Associating resource types to an information type supports the allocation of resources to a process. Some processes rely on the availability of information. Information classified as an intangible resource will not be available directly. There are other resources providing the required information. These might be media containing information or an elementary resource type providing information. Information systems in a broader sense fulfil this requirement. Hence, systems allowing the access to information are an important resource. Modelling these resources specifies the way of accessing the needed information.

6 FUTURE WORK

This paper presents the basic conceptualisation of a resource modelling language for an existing business process language. Core features of the meta-model of this language have been described and some other features have been mentioned. In the future the resource modelling language will be evaluated - in cooperation with other research groups and projects - fostered by modelling resources of different domains. Goals of this evaluation are the verification of the specified concepts and the determination of possible ambiguities in the language specification. Those ambiguities will be fixed in future versions of the resource modelling language. Furthermore the modelled resources and resource types will be integrated into the resource modelling language. They will be grouped by several domain-specific packages and made available to the user. We are planning to develop resource-packages for logistical processes, project management and IT-resources.

Up to now, tool-support for the modelling of resources using the resource modelling language is not available. An appropriate resource modelling tool will be developed after the end of the evaluation phase. This tool will be integrated into the process modelling framework of the MEMO-OrgML. It will support the definition of resources and resource types as well as the management of existing resources and resource packages.

REFERENCES

Baumgarten, B. (1996) *Petri-Netze: Grundlagen und Anwendungen*, 2nd edition, Spektrum Akademischer Verlag, Heidelberg (Germany)

Bennett, S., McRobb, S. and Farmer, R. (1999) *Object-Oriented Systems Analysis and Design using UML*, McGraw-Hill, London (UK)

Bergholtz, M. and Johannesson, P. (2000) 'Validating Conceptual Models - Utilising Analysis Patterns as an Instrument for Explanation Genertion', In: Bouzeghoub, M., Kedad, Z. and Métais, E. (eds) 5th International Conference on Applications of Natural Language to Information Systems, Versailles, France, June 28-30 2000, pp. 325-339

Chrystal, K. and Lipsey, R. (1997) *Economics for Business and Management*, Oxford University Press, Oxford (UK)

Curtis, B., Kellner, M. and Over, J. (1992) "Process Modeling" Communications of the ACM, Vol.35, No.9, September 1992, pp. 75-90

Diederich, H. (1992) *Allgemeine Betriebswirtschaftslehre*, 7th edition, Kohlhammer, Stuttgart (Germany)

Eertink, H., Janssen, W., Luttighuis, P.O., Teeuw, W. and Vissers, C. (1999) 'A Business Process Design Language', In: Wing, J., Woodcock, J. and Davies, J. (eds): Proceedings of the World Congress on Formal Methods in the Development of Computing Systems, Toulouse, France, September 20-24 1999, Volume I, pp. 76-95

Eriksson, H.-E. and Penker, M. (2000) *Business Modeling with UML - Business Patterns at Work* Wiley, New York (NY)

Frank, U. (1999) *MEMO: Visual Languages for Enterprise Modelling*, Research Report of the IS Research Institute, University of Koblenz, No. 18

Frank, U. (2001) *Organising the Corporation: Research Perspectives, Concepts and Diagrams*, Research Report of the IS Research Institute, University of Koblenz, No. 25

Gamma, E., Helm, R., Johnson, R. and J. Vlissides (1998) *Design Patterns: Elements of Reusable Object-Oriented Software* Addison-Wesley, Reading (Massachusetts)

Green, P. and Rosemann, M. (1999): 'An Ontological Analysis of Integrated Process Modelling', In: Jarke, M. and Oberweis, A. (eds): Proceedings of the 11th International Conference CAiSE'99, Heidelberg, Germany, June 14-18 1999, pp. 225-240

Herbst, H. (1997) *Business Rule-Oriented Conceptual Modeling*, Physica-Verlag, Heidelberg (Germany)

Jung, J. and Fraunholz, B. (2002) 'Resource Modelling in an object-oriented Process Modelling Language', In: Cybulsky, J.; Nguyen, L.; Lamp, J. and Smith, R. (eds): Proceedings of the Australian Workshop on Requirements Engineering, AWRE2002, Melbourne (Australia), 2-3 December 2002, CD-ROM edition

Koubarakis, M. and Plexousakis, D. (2000) 'A Formal Model for Business Process Modeling and Design', In: Wangler, B. and Bergman, L. (eds): Proceedings of the 12th International Conference CAiSE 2000, Stockholm, Sweden, June 5-9 2000, pp. 142-156

Marshall, C. (2000) *Enterprise Modeling with UML - Designing Successful Software through Business Analysis*, Addison-Wesley, Reading (MA)

Miller, B.B. (1988) Managing Information as a Resource, in: Rabin, J. and Jackowsky, E.M. (eds): Handbook of Information Resource Management, Marcel Dekker, New York (NY)

Nübel, H. (2001) "The resource renting problem subject to temporal constraints", OR Spektrum, Vol. 23, No. 3, 2001, pp. 359-381

Oberweis, A. (1996) *Modellierung und Ausführung von Workflows mit Petri-Netzen*, Teubner, Stuttgart (Germany)

Österle, H. (1995) *Business Engineering: Prozess- und Systementwicklung*, Springer, Berlin (Germany)

Podorzhny, R.M., Staudt Lerner, B. and Osterweil, L.J. (1999) 'Modeling Resources for Activity Coordination and Scheduling', In: Ciancarini, P. and Wolf, A.L. (eds): Proceedings of the ThirdInternational Conference, COORDINATION '99, Amsterdam, The Netherlands, April 26-28 1999, pp. 307-322

Scheer, A.-W. (1992) *Architecture of Integrated Information Systems - Foundations of Enterprise-Modelling*, Springer, Berlin (Germany)

Scheer, A.-W. (1998) *Business Process Engineering - Reference Models for Industrial Enterprises*, Springer, Berlin (Germany)

Scheer, A.-W. (1999) *ARIS - Business Process Modeling* 2^{nd} edition, Springer, Berlin (Germany)

Sutton, S.M. and Osterweil, L.J. (1997) 'The Design of a Next-Generation Process Language', In: Jazayeri, M. and Schauer, H. (eds): Proceedings of the 6th European Software Engineering Conference Zürich, Switzerland, September 22-25 1997, pp. 142-158

van der Aalst, W. and van Hee, K. (2002) *Workflow Management - Models, Methods and Systems*, MIT Press, Cambridge (MA)

Wand, Y. and Weber, R. (1989) An Ontological Evaluation of Systems Analysis and Design Methods, in: Falkenberg, E.D. and Lindgreen, P. (eds): Information Systems Concepts: An In-depth Analysis, North-Holland

Wand, Y. and Weber, R. (1990a) "An Ontological Model of an Information System", IEEE Transactions on Software Engineering, Vol. 16, No. 11, pp. 1281-1291

Wand, Y. and Weber, R. (1990b) Mario Bunge's Ontology as a Formal Foundation for Information System Concepts, in: Weingartner, P. and Dorn, G.J.W. (eds): Studies on Mario Bunge's Treatise, Rodopy, Atlanta

Wand, Y. and Weber, R. (1993): "On the Ontological exspressivness of Information Systems Analysis and Design Grammar", Journal of Information Systems, Vol. 3, No. 2, pp. 217-237

Weber, R. (1997) *Ontological Foundations of Information Systems*, Coopers and Lybrand Accounting Methodology, Monograph No. 4, Melbourne, Australia

zur Muehlen, M. (1999) 'Resource Modeling in Workflow Applications', In: Becker, J., zur Mühlen, M. and Rosemann, M. (eds) Proceedings of the 1999 Workflow Management Conference, November 9 1999, Münster, Germany, pp. 137-153