IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP5029

Configurative Process Modeling - Outlining an Approach to Increased Business Process Model Usability

Jörg Becker
University of Münster, Dept. of Information Systems, Leonardo-Campus 3, 48149 Muenster, Germany, isjobe@wi.uni-muenster.de
Phone: +49 251 8338100

Patrick Delfmann
University of Münster, Dept. of Information Systems, Leonardo-Campus 3, 48149 Muenster, Germany, ispade@wi.uni-muenster.de
Phone: +49 251 8338083

Alexander Dreiling
University of Münster, Dept. of Information Systems, Leonardo-Campus 3, 48149 Muenster, Germany, isaldr@wi.uni-muenster.de
Phone: +49 251 8338070

Ralf Knackstedt
University of Münster, Dept. of Information Systems, Leonardo-Campus 3, 48149 Muenster, Germany, israkn@wi.uni-muenster.de
Phone: +49 251 8338094

Dominik Kuropka
University of Münster, Dept. of Information Systems, Leonardo-Campus 3, 48149 Muenster, Germany, isdoku@wi.uni-muenster.de
Phone: +49 251 8338079

ABSTRACT

Business processes comprise an indispensable element of the behavior of contemporary organizations rendering Business Process Management essential. Information models are a valuable tool for providing information on business processes. As with any model, business process models serve a specific purpose. This purpose makes it nearly impossible to adequately communicate them to all user groups within an organization, because each user group has a specific perspective of business processes rendering certain aspects irrelevant. To address this problem, configurative process models can be used to provide information on business processes to different user groups in an appropriate and efficient manner. In this paper, we will outline a methodology, which facilitates the creation of configurative process models.

INTRODUCTION

The concept of Business Process Management (BPM) has been discussed for decades. Early examples of scientific work in the area of organizational science go back to 1934, where Nordsieck pointed out the necessity of a process oriented corporate design (Nordsieck, 1934). In 1972 he stated (translated from German):

"Business activity is a continuous process, an uninterrupted value chain […]. Actually, the structure of a company is like that of a stream. It cyclically and continually creates and distributes new products and services based on unaltered or only slightly altered tasks. Thus, how could […] a company be differently organized than according to its natural and technical processes?" (Nordsieck, 1972, p. 9)

Despite the early scientific discussions of BPM, process orientation was effectively implemented in organizations only in the late 80's and early 90's, after other researchers such as Gaitanides (Gaitanides, 1983), Porter (Porter, 1985), Davenport (Davenport, 1993), Scheer (Scheer, 1990), and Hammer & Champy (Hammer, Champy, 1993) published their work.

Information Systems (IS) are essential for enacting business processes efficiently within and between organizations. Of late, they have been more often recognized as a vital backbone of an organization, rather than only as a simple business support tool (Henderson, Venkatraman, 1999; Li, Chen, 2001; Venkatraman, 1994). Furthermore, Information Technology (IT) and IS play important roles in creating competitive advantages, making IT and IS essential for companies acting in highly competitive markets (Johnston, Vitale, 1988). Hence, IS can be seen as an enabler for increasing business process efficiency, as Davenport stated in 1993 (Davenport, 1993).

Information models are the basis of successful information systems engineering (Karimi, 1988; Kottemann, Konsynski, 1984). They enable members of an interdisciplinary project team to communicate with each other. The model should have a degree of formality in order to clearly specify information systems. Furthermore a degree of intuitive understanding must be conveyed so that it is understood by the targeted users. Enhanced means of communication explicitly allow the inclusion of targeted users, management, application designers, and programmers within the development process of an information system and therefore addresses management support and user involvement (Becker et al., 2003; Holten, 2003; Holten, Dreiling, Becker, to be published in 2004). In order to develop high quality IT solutions, business requirements need to be identified and modeled from a business perspective. After having defined the business requirements, an information system needs to be specified, which can subsequently be implemented.

The Object Management Group (OMG) addresses the problem of information systems engineering by proposing the so-called Model Driven Architecture (MDA) (Soley, the OMG Staff Strategy Group, 2000). Various modeling techniques are used to develop vendor- and middleware-neutral information models. In a second step, these information models are used to design middleware concepts. After selecting a language, the implementation of information systems can be initiated based on the middleware design. The Architecture of Integrated Information Systems (ARIS) presented by Scheer, is another approach to specifying information systems (Scheer, 2000). The four different views: data, functions, organization, and control, each consisting of the three layers of conceptual model, technical model and implementation, can be used to model different aspects of a software system from both a business perspective and an IT perspective. All of these models correspond to each other. Language constructs from one view can be integrated into models from a different view which ensures that the information models are highly integrated.

Several approaches support the modeling of business processes. Examples are Petri Nets (Petri, 1962), State Activity Charts (Booch, Rumbaugh, Jacobson, 1999), and Event Driven Process Chains (Scheer, 2000). The Event Driven Process Chain (EPC) in particular is a valuable tool for modeling business processes conceptually. EPCs can very easily be understood by business users, which allows their integration into the development process of an information system.

In order to enhance the communication potential of information models, different perspectives on these models need to be defined, because users or user groups focus on process information relevant for them. Configurative process modeling facilitates the creation of process models which can be adapted automatically to provide information on the respective processes to different users or user groups in an appropriate manner. The presented meta model-based approach to configurative process modeling has been developed in a research project focusing on method building (for a detailed discussion on the research finding refer to Becker et al., 2002; Becker et al., 2001). We will present the research findings by first introducing relevant related work. This is followed by a section describing state-of-the art process modeling. In order to address the current limitations with which process modeling is faced, we will introduce an approach to configurative process modeling.

PROCESS MODELING – STATE-OF-THE-ART

According to the Total Quality Management approach, the quality of a product is determined by it's fitness-for-use for the consumer and his requirements (Ishikawa, 1985). When transferred to process models, their quality depends on the fitness-for-use concerning the requirements of particular users or user groups. User requirements result from their different perspectives on business processes (Rosemann, 2003). The perspective of a software engineer on a core process such as invoicing will be different from the perspective of a customer sales representative or manager. This results from the fact, that a software engineer mainly uses information models of business processes for software implementation and maintenance, whereas a customer sales representative focuses on enacting them. The manager, on the other hand, could use the business process model to influence the structure of the business process. Perspectives result from the deliberately specific use of a business process model, the organizational role of a user, as well as individual preferences on the conceptual and representational design of business process models (Becker et al., 2002).

The more effectively the process model meets the requirements of a particular perspective, the higher its quality. Ideally, each identified perspective should be provided with a tailor-made version of a process model. This approach is called "multi-perspective process modeling" (Becker et al., 2002; Darke, Shanks, 1996; Rosemann, 1998). The most significant problem resulting from a multiplicity of perspective specific models is the need to manage possible redundancies inside the model itself. This leads to increased modeling and maintenance costs and the danger of inconsistencies within the model base.

In order to enable an efficient multi-perspective process modeling, redundancies have to be overcome. A modeling methodology which enables the user to avoid redundancies, and to consider multiple perspectives within the model base is called configurative process modeling (Becker et al., 2002). This concept will be introduced in the next section.

CONFIGURATIVE PROCESS MODELING

Configurative process modeling implies the use of integrated information models that contain all relevant and specific variations without redundancies of a particular domain. To enable automatic transformation of these integrated information models into perspective-specific models, all models must be instances of a formalized meta model. A meta model is a model of a specific class of models. It specifies syntactic constraints for the models within the class. Meta modeling is a popular approach to analyzing information system methods. Based on models related to real-world objects, meta models are used to specify modeling languages (Holten, 2000; Nissen et al., 1996; Strahringer, 1996). Recently, Rosemann and Green presented a meta model of the Bunge-Wand-Weber ontology (Rosemann, Green, 2002). Meta models have also been developed within the field of decision support systems (van Hee, Somers, Voorhoeve, 1991). In van Hee et al.'s approach, both the meta model and language are specified by formal expressions. Modeling techniques with user-appropriate concepts and representations, simplify the modeling process and thus help to align further business and IT objectives (Reich, Benbasat, 2000).

Our configurative process modeling approach is based on the idea of *meta model projection*. Using meta model projection, perspective-specific information models are derived in two steps: In the first step, the meta model of the set of integrated information models is reduced with respect to the perspective-specific needs. In the second step, the set of integrated information models, which has already been configured partially in the first step, is projected to the new meta model, thus satisfying the syntactic constraints of the reduced meta model. Explicitly defined parameter-based rules for the reduction of meta models specify how models are adapted automatically to provide appropriate information for a perspective (Becker et al., 2002).

For a fine granular configuration, each process element can be enriched with information specifying the perspective to which it belongs. This defines how an element reacts to automatic adaptation. However, even if high flexibility is achieved by providing such detailed information, the corresponding high modeling cost render such an approach impractical. Hence, it is reasonable to provide fine granular configuration mechanisms only in case the process model contains a limited amount of elements to be configured. Coarse granular configuration mechanisms on the other hand, should be provided in order to enable configurations operating on groups of several model elements to be configured similarly (Becker et al., 2002).

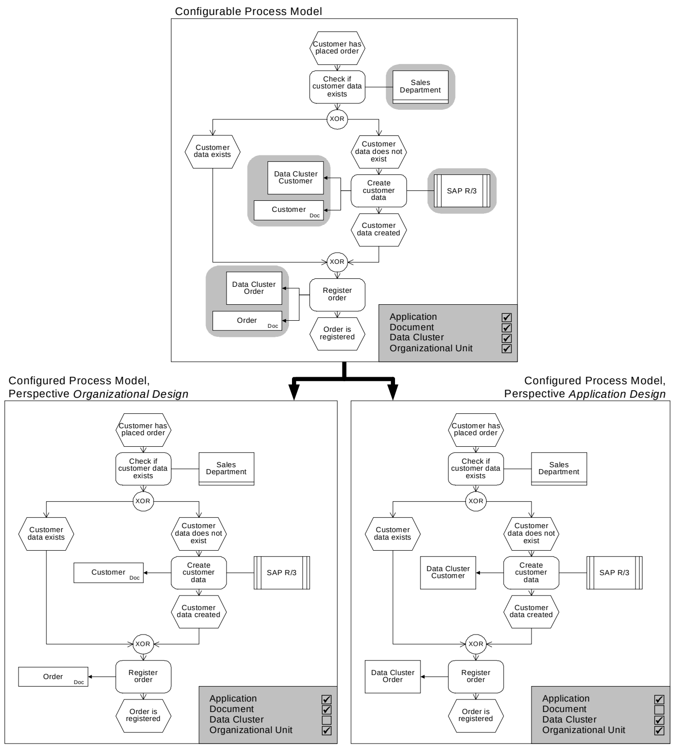
In this paper, we focus explicitly on how the process modeling technique Event Driven Process Chain (EPC) can be made configurable (A detailed discussion on the methodological foundation of the used configuration mechanisms can be found in Becker et al., 2002). An EPC consists of interrelated instances of the following process object types: *function*, *event*, *connector*, and several *resource types* (e.g. document, employee, application). For the automatic configuration of an EPC, logically, we can define only mechanisms which either address object types, their instances, or relationships between these instances. We regard configurations at the type level as coarse granular and those at the instance level (instances as well as their relationships) as fine granular.

Subject to these assumptions, there are only two possible configuration mechanisms at a coarse granular level. The first is referred to as *Object Type Selection*:

- The use of optional object types in EPCs (e.g. the annotation of organizational units to process functions so as to demonstrate their specific responsibility) depends on the particular perspective. The configuration mechanism *Object Type Selection* enables the model designer to assign object types within the EPC to perspectives. By applying *Object Type Selection*, a model designer is automatically able to hide all instances of one object type which is irrelevant for the considered perspective. *Object Type Selection* is achieved by a projection of the EPC's underlying meta model.

An example of a coarse granular configuration is shown in Figure 1. This is based on a configurable process model where the two perspectives *Organizational Design* and *Application Design* determine which object types are to be hidden while configuring the process model. In order to hide data clusters, the first perspective *Organizational Design* requires representative data that is closely related to the implementation of the business process. On the other hand, the second perspective *Application Design* requires that precisely these data clusters remain in the model. Documents which are also data, but describe the used constructs semantically and pragmatically, will be hidden from

Figure 1: Object Type Selection



Source: (Becker et al., 2002, p. 94)

the perspective *Application Design* in a configured model, but remain visible for the perspective *Organizational Design*.

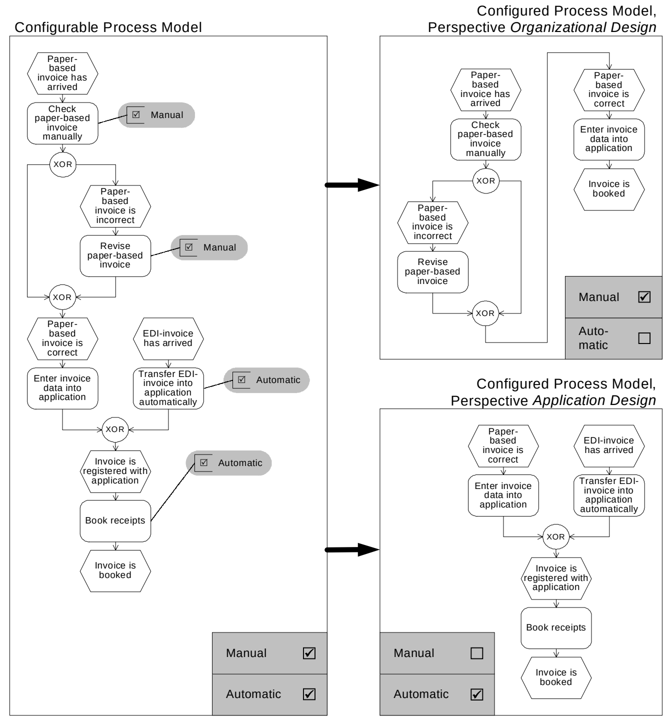
The second coarse granular configuration mechanism is referred to as *Representation Variation*:

- While *Object Type Selection* substantially changes the process model's language and content by hiding several object types and thus all instances within the configured process model, *Representation Variation* replaces the symbols used within the configurable process model. On the one hand, *Representation Variation* facilitates the replacement of object type symbols, because images may be more intuitive in particular perspectives, which focus on presenting a process model, than vector-based identifiers for more technical perspectives. On the other hand, different topologies of models have been established which are also addressed by *Representation Variation* (e.g. arrangement of process logic from left to right in the area of *Knowledge Management* vs. column-based EPCs for the purpose of *Organizational Design*).

Apart from the coarse granular configuration mechanisms, we can define several fine granular configuration mechanisms which are based on *Object Selection*. There are several sub-configuration mechanisms from which we wish to introduce the three most important, starting with *Attribute-Based Object Selection*:

- Objects within process models possess certain attributes, whose values render them relevant or irrelevant for particular perspectives. For instance, there are process events within EPCs which deliver no additional semantic or pragmatic value such as the event "payment booked" which follows the activity "book payment". This event is trivial and can be hidden, if the model has a certain degree of complexity. Such examples can often be found in process streams which have no branches. If a Boolean-attribute "trivial object" is assigned the value "true", then *Attribute-Based Object Selection* hiding all objects with the attribute "trivial object" being "true", leads to a configured process model without trivial objects. *Attribute-Based Object Selection* can also be applied to objects other than events.

Figure 2: Attribute-Based Object Selection



Source: (Becker et al., 2002, p. 111)

An example focusing on hiding automatic or manual objects within a business process model is shown in Figure 2, which illustrates the concepts of *Attribute-Based Object Selection*. The perspective *Organizational Design* hides all automatic functions of the original process model within the configured process model. The second configured model from the perspective *Application Design*, in turn, hides all manual functions of the original process model.

Another sub-configuration mechanism of Object Selection is *Term-based Object Selection*:

- *Term-based Object Selection* is more flexible than *Attribute-Based Object Selection*. Terms represent identifiers of perspectives or expressions containing identifiers of perspectives and logical connectors. These expressions are formulated in a context-free grammar (Hopcroft, Motwani, Ullman, 2000):

The terms can be parsed in order to derive the relevant objects for a perspective or set of perspectives. This type of configuration mechanism leads to very complex process modeling and should therefore only be used if no other configuration mechanisms are suitable. This may occur if the definition of a new attribute does not pay off, because of an inadequately low number of applicable objects (for example, process variants due to differences in process execution within different departments). E. g., a retailing company which performs different business activities such as *warehousing*, *third-party deals* and *central settlement*,

Figure 3: Context-free grammar for the specification of configuration terms (segment above from Becker et al., 2002; Schwegmann, 1999).

<Term>	::= <Expression> {<Operator> <Expression>}
<Expression>	::= <Prefix> "Perspective" <Perspective_Value_List>
<Perspective_Value_List>	::= "(" <Prefix> <Perspective_Value_List> { <Operator> <Prefix> <Perspective_Value_List> } ")"
<Perspective_Value_List>	::= <Perspective>
<Operator>	::= "|" | "+"
<Prefix>	::= "NOT" | <empty>

has to perform different process variants of invoice auditing which can be configured for the responsible departments. The following term assigns an element of an invoice, auditing EPC directly to the perspectives "Warehousing", "Third-Party Deal" and combinations:

NOT Perspective (Central_Settlement)

The third sub-configuration mechanism of Object Selection is referred to as *Relationship Type Selection*:

- Some perspectives may require that certain relationship types between objects be hidden. If, for example, a process contains a detailed description of responsibilities for process functions, such as "performs", "is responsible for", "participates" etc., then users from certain perspectives may be distracted from the actual content of the process model, i.e., its process logic. For instance, this is the case when employees are to be instructed in new tasks by use of process models. As a consequence, these models must contain only information relating to the duties of the *performing* employees to be instructed. This information is provided by organizational elements annotated to functions with the relationship type "performs". All other relationship types must be hidden. *Relationship Type Selection* allows the assigning of perspectives to relationship types, which leads to their hiding in case a perspective identifies them as irrelevant. This may lead to the hiding of other objects if their link to the rest of the model has been lost, causing lost semantics and pragmatics.

There are several other configuration mechanisms depending on the modeling technique used or the setting in which model configuration is applied. The introduced three fine granular configuration mechanisms present a sufficient choice for the configuration of EPCs.

The introduced configuration mechanisms are based on hiding and exposing modeling constructs. The specification of configuration mechanisms is defined within the language-based meta model, which also defines the modeling technique itself. In order to be able to allow type-based configuration (e.g. Object Type Selection), it is necessary to design a flexible meta model. A flexible meta model leads to a modifiable meta language, which is specified at the meta meta level (for a detailed meta model-based specification of the proposed modeling technique refer to Becker et al., 2002). Regarding the configurable model, the meta meta model is the language-based meta model (for a detailed discussion on the metaization process refer to Holten, Dreiling, Becker, to be published in 2004).

RELATED WORK

The approach presented in this paper differs from what is commonly known as model transformation, which focuses of transforming source models into target models. Both languages in model transformation, source and target language, may differ radically from each other. The transformation itself is achieved by transformation rules which map source and target language (Engels et al., 1997). For instance, EPCs can be transformed to Petri Nets (Moldt, Rodenhagen, 2000). Model transformation mechanisms usually feature a high degree of abstraction and expressive power (e.g. combinations of generic transformation operations such as *create new*, *create link*, *delete*, *refer else create*, *create inside*, *refer to*. These are used to map elements of the source language to those of the target language). Due to the generality of transformation operations, utilization of the expressive power and flexibility of transformation mechanisms is only possible if experienced and highly skilled experts are available. Each new transformation requirement, such as hiding a process branch, leads to the construction or extension of a transformation mechanism, which must be specified with the mentioned generic transformation operations.

The proposed approach is based on more specialized configuration mechanisms that are more intuitive for non-skilled users. This is mainly due to the simplicity and higher domain specificity of the introduced configuration mechanisms. In terms of domain-specific modeling (Nordstrom et al., 1998), configuration mechanisms are modeling technique extensions that meet the special requirements of multi-perspective process modeling.

SUMMARY AND OUTLOOK

Configurative process modeling is necessary to maintain multiple perspectives on business processes. The latter are necessary to address the critical success factors of information systems development. We illustrated this with the example of the two perspectives Organizational Design and Application Design which supplied different information about the same process to different model users, in order to provide them only with relevant information. Generally, configurative process modeling facilitates the inclusion of business users, management, application designers, and programmers, because process information can be provided adequately to these different user groups for validation, discussion, and implementation.

We introduced a set of coarse and fine granular configuration mechanisms, outlined the implications for process modeling, and the benefits that can be achieved by using these configuration mechanisms. The proposed configuration mechanisms allow the hiding of entire object types including all of their instances, as well as particular instances and their relationships. Furthermore, we are able to influence the layout by changing the representation of model constructs.

Our further work will concentrate on implementing the proposed method and configuration process, thus extending a popular process-design tool. Support for the proposed configuration mechanisms can be provided, because the meta model projection methodology creates views of total process models. Thus, it can be implemented efficiently using database technology and SQL queries. The next step will be a validation of the proposed method by means of a business case using the process-design tool. We will also add a component-orientated extension to the proposed methodology.

REFERENCES

Becker, J., Delfmann, P., Knackstedt, R., & Kuropka, D. (2002). Konfigurative Referenzmodellierung. In J. Becker & R. Knackstedt (Eds.), *Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem- und Organisationsgestaltung* (pp. 25-144). Heidelberg.

Becker, J., Dreiling, A., Holten, R., & Ribbert, M. (2003). Specifying Information Systems for Business Process Integration - A Management Perspective. *Information Systems and e-Business Management, 1*(3), 231-263.

Becker, J., Knackstedt, R., Kuropka, D., & Delfmann, P. (2001). *Subjektivitätsmanagement für die Referenzmodellierung: Vorgehensmodell und Werkzeugkonzept*. Paper presented at the KnowTech 2001, Dresden.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *Unified Modelling Language User Guide*. Boston, MA, USA et al.: Addison-Wesley Professional.

Darke, P., & Shanks, G. (1996). Stakeholder Viewpoints in Requirements Definition. *Requirements Engineering, 1*(1), 88-105.

Davenport, T. H. (1993). *Process Innovation: Reengineering Work Through Information Technology*. Boston, MA, USA: Harvard Business School Press.

Engels, G., Heckel, R., Taentzer, G., & Ehrig, H. (1997). A View-Oriented Approach to System Modelling Based on Graph Transformation. *ACM SIGSOFT Software Engineering Notes, 22*(6), 327-343.

Gaitanides, M. (1983). *Prozessorganisation. Entwicklung, Ansätze und Programme prozessorientierter Organisationsgestaltung*. München: Vahlen.

Hammer, M., & Champy, J. (1993). *Reengineering the Corporation. A Manifesto for Business Revolution*. New York, NY, USA: HarperBusiness.

Henderson, J. C., & Venkatraman, N. (1999). Strategic Alignment: Leveraging Information Technology for Transforming Organizations. *IBM Systems Journal, 38*(2/3), 472-484.

Holten, R. (2000). Framework and Method for Information Warehouse Development Processes. In R. Jung & R. Winter (Eds.), *Data Warehousing 2000 - Methoden, Anwendungen, Strategien* (pp. 135-

163). Heidelberg, Germany.

Holten, R. (2003). Specification of Management Views in Information Warehouse Projects. *Information Systems, 28*(7), 709-751.

Holten, R., Dreiling, A., & Becker, J. (to be published in 2004). Ontology-Driven Method Engineering for Information Systems Development. In M. Rosemann & P. Green (Eds.), *Ontological Analysis, Evaluation, and Engineering of Business Systems Analysis Methods [Working Title]*: Publisher in negotiation.

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2000). *Introduction to Automata Theory, Languages, and Computation* (2 ed.). Boston, MA, USA: Addison-Wesley Longman Publishing.

Ishikawa, K. (1985). *What is Total Quality Control? The Japanese Way*. Englewood Cliffs, NJ, USA: Prentice Hall.

Johnston, H. R., & Vitale, M. R. (1988). Creating Competitive Advantages with Interorganizational Information Systems. *MIS Quarterly, 12*(2), 153-165.

Karimi, J. (1988). Strategic Planning for Information Systems: Requirements and Information Engineering Methods. *Journal of Management Information Systems, 4*(4), 5-24.

Kottemann, J. E., & Konsynski, B. R. (1984). Information Systems Planning and Development: Strategic Postures and Methodologies. *Journal of Management Information Systems, 1*(2), 45-63.

Li, E. Y., & Chen, H.-G. (2001). Output-driven information system planning: a case study. *Information & Management, 2001*(38), 185-199.

Moldt, D., & Rodenhagen, J. (2000). *Ereignisgesteuerte Prozeßketten und Petrinetze zur Modellierung von Workflows.* Paper presented at the Visuelle Verhaltensmodellierung verteilter und nebenläufiger Software-Systeme. 8. Workshop des Arbeitskreises GROOM der GI Fachgruppe 2.1.9 Objektorientierte Software-Entwicklung, Münster, Germany.

Nissen, H. W., Jeusfeld, M., Jarke, M., Zemanek, G. V., & Huber, H. (1996). Managing Multiple Requirements Perspectives with Metamodels. *IEEE Software, 13*(3), 37-48.

Nordsieck, F. (1934). *Grundlagen der Organisationslehre*. Stuttgart: Poeschel.

Nordsieck, F. (1972). *Betriebsorganisation. Lehre und Technik*. Stuttgart, Germany: Schäffer Verlag.

Nordstrom, G., Sztipanovits, J., Karsai, G., & Ledeczi, A. (1998). *Metamodeling - Rapid Design and Evolution of Domain-Specific Modeling Environments.* Paper presented at the IEEE ECBS'98.

Petri, C. A. (1962). *Kommunikation mit Automaten.* Unpublished Dissertation, University of Bonn, Bonn.

Porter, M. E. (1985). *Competitive Advantage*. New York, NY, USA: The Free Press.

Reich, B. H., & Benbasat, I. (2000). Factors That Influence the Social Dimension of Alignment between Business and Information Technology Objectives. *MIS Quarterly, 24*(1), 81-113.

Rosemann, M. (1998). *Managing the Complexity of Multiperspective Information Models using the Guidelines of Modelling.* Paper presented at the 3rd Australian Conference on Requirements Engineering, Geelong, Australia.

Rosemann, M. (2003). Preparation of Process Modeling. In J. Becker, M. Kugeler & M. Rosemann (Eds.), *Process Management. A Guide for the Design of Business Processes* (pp. 41-78). Berlin et al.: Springer.

Rosemann, M., & Green, P. (2002). Developing a meta model for the Bunge-Wand-Weber ontological constructs. *Information Systems, 27*(2), 75-91.

Scheer, A.-W. (1990). *EDV-orientierte Betriebswirtschaftslehre. Grundlagen für ein effizientes Informationsmanagement*. Berlin et al.: Springer.

Scheer, A.-W. (2000). *ARIS - Business Process Modeling* (3 ed.). Berlin: Springer.

Schwegmann, A. (1999). *Objektorientierte Referenzmodellierung. Theoretische Grundlagen und praktische Anwendung*. Wiesbaden: Gabler.

Soley, R., & the OMG Staff Strategy Group. (2000). *Model Driven Architecture* (White Paper). Needham, MA, USA: Object Management Group.

Strahringer, S. (1996). *Metamodellierung als Instrument des Methodenvergleichs. Eine Evaluierung am Beispiel objektorientierter Analysemethoden*. Herzogenrath, Germany: Shaker.

van Hee, K. M., Somers, L. J., & Voorhoeve, M. (1991). A modeling environment for decision support systems. *Decision Support Systems, 7*(3), 241-251.

Venkatraman, N. (1994). IT-Enabled Business Transformation: From Automation to Business Scope Redefinition. *Sloan Management Review*, 73-87.

FOOTNOTES

1 This work has been funded by the German Research Association (Deutsche Forschungsgemeinschaft DFG), record number Be 1422/5-1.

Related Content