

An Architecture for Collaborative Scenarios applying a common BPMN-Repository

Thomas Theling¹, Jörg Zwicker¹, Peter Loos¹, Dominik Vanderhaeghen²

¹ Johannes-Gutenberg University Mainz, Information Systems and Management,
55099 Mainz, Germany
{Theling, Zwicker, Loos}@isym.bwl.uni-mainz.de
<http://www.isym.bwl.uni-mainz.de>

² Institute for Information Systems at the German Research Center for Artificial Intelligence
(DFKI), 66123 Saarbruecken, Germany
vanderhaeghen@iwi.uni-sb.de
<http://www.iwi.uni-sb.de/>

Abstract. Collaborative business scenarios require a flexible integration of enterprises. To manage inter-organizational business processes, existing concepts for business process management (BPM) need to be adopted and extended. This paper presents a conceptual architecture for cross-enterprise processes' planning, implementation, and controlling. Core component of the architecture is a distributed repository managing all required data and information, which especially obtains a process-oriented view on collaboration networks. Cross-organizational processes need close coordination among networking partners. This is achieved through the integration of business process models. Thus we further propose a procedure model of an XML-based model transformation, which reduces complexity of the modeling-task and enables model integration on a conceptual level.

1 Collaborative Business Process Management Field of Research

Innovation pressure arises in interaction relations like supply chains and in networks with complementary core competence partners. Thus the concept of borderless enterprises [1] has been discussed for years and the collaborative production of goods and services has been established. The opening of organizational borders is no longer regarded as a necessary evil, but rather as a chance with strategic importance. Therefore inter-organizational applications and the transfer of business process management aims into collaborative IT systems are considered.

An architecture for managing cooperative and collaborative business processes requires functions which support all phases of collaborations [2]. Existing architecture concepts are discussed e. g. by Gronau, Scheer, van der Aalst, or Wild et al. [3-6]. They stress various single aspects and tasks within collaborations. Workflow-functionalities [7] for the IT-supported execution of processes was developed, while EAI-functionalities [8] ensure data integration and process integration. Business process management methods were steadily developed and enhanced. Web Service tech-

nologies [9] are proposed to facilitate process management based on application components. Service-oriented Architectures (SOA) [10] are established for composing a process-oriented, flexible and reactive information system architecture.

To exchange data between architecture components, data and especially business process models have to be stored in unique data formats. Transformation of models is addressed in the field of software development with the use of approaches like the model-driven architecture (MDA) [11]. Furthermore existing business process knowledge should be reused in collaborations. Notation-specific XML representations like the Event-driven Process Chain Markup Language (EPML) [12] focus on business process model transformations. Techniques as Extensible Stylesheet Language Transformations (XSLT) [13], support the automated transformation of XML-based models between different, heterogeneous formats in general. Our approach designs a framework for the transformation into a collaboration-wide business process description format.

This contribution depicts the concept of an Architecture for Collaborative Scenarios and to exemplify a procedure model enabling syntactic business process model interoperability. The paper unfolds as follows: After this introduction, we address the Architecture for Collaborative Scenarios in the following chapter. Our approach considers organizational aspects and assimilates existing concepts of workflow management and BPM frameworks [4, 5, 7]. Further on, to fulfill the aforementioned requirement of a collaboration-wide unique syntax of business process models BPMN is suggested in chapter 3 for storing of these models within the introduced architecture. For solving the problem of heterogeneity in business process modeling, a generic, adaptable procedure model to gain syntactic model interoperability is developed – showing the steps for the transformation in a collaboration-wide BPMN syntax. Thus, we provide the preparing methods for the integration of process models in collaborations in a holistic approach. Finally, conclusions of our approach are discussed. Also, we point to some further research directions.

2 Architecture for Dynamic Collaborative Scenarios

Dealing with collaborations and information systems, the term dynamic concerns the change of people, the change of technologies and the change of processes [14]. Further dynamic aspects are e. g. the time lag between IT-investments and IT-payoffs [15]. Main requirement resulting from dynamic factors is flexibility. In order to reconfigure or to resolve a collaboration, to integrate a new partner into collaborations or to plan new collaborative strategies, indicators have to be defined and depicted. Further on, changing processes require an ad-hoc-execution and adaption of business processes and structures. This implies a flexible integration of various operational IT systems.

Further demands for an integrative architecture are e. g. graphical, multi-perspective modeling while supporting different modeling methods, integrated collaboration management and controlling, integrated process execution based upon the

stored business process models, and a concept for integration of new network partners. At the same time existing solutions have to be considered and assimilated.

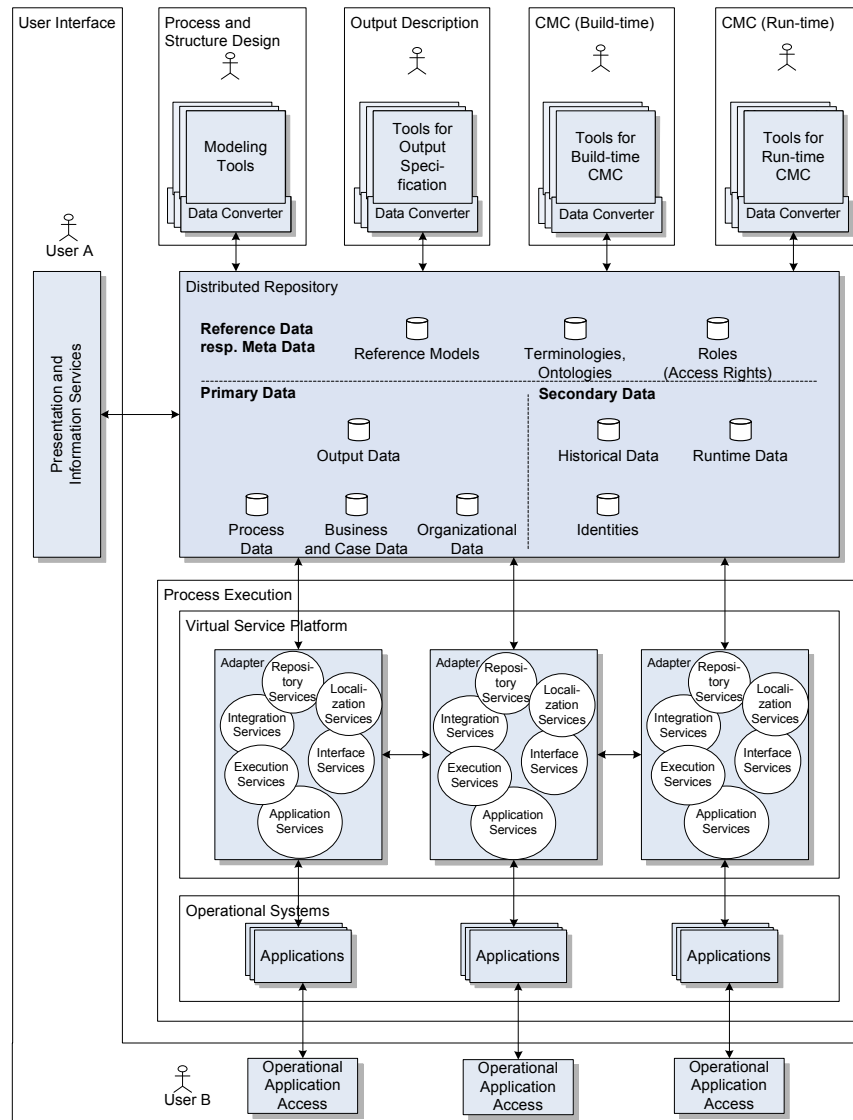


Fig. 1. Architecture for Collaborative Scenarios

Based on these requirements the Architecture for Collaborative Scenarios has been developed (cf. figure 1). Basic information for the process execution is visualized in business process models, output models, and organization models. These are created

by modeling tools and stored in a distributed repository. Depending on process- and organization models the virtual service platform executes processes and integrates different operational systems of the collaboration partners. A life-cycle-model [2] serves as a manual for the process-oriented establishment, the process integration, and operation of collaborations towards a common production of goods and services. The following sections describe the components and characteristics of the architecture.

2.1 Distributed repository

The architecture is realized upon a physical **distributed repository** managing all data. It enables business process management, common work on the underlying models, and cross-enterprise process execution. Individual knowledge of each partner is stored within decentralized parts of the repository. A logical centralization of the repository represents the knowledge of the overall network.

The repository contains following data categories: *Reference data resp. Meta data* provides a basis for the design of process models and organizational models: *Reference models* support the construction of individual models and improve the design of specific organizational structures and processes of the enterprise network [16, 17]. *Ontologies* unify differing vocabulary of concepts and meanings regarding contents and model semantics [18]. Ontologies are conceptual systems of a domain, which obtain knowledge transfer between applications and users. Within the architecture, ontologies integrate different language formats of applications. For the design of models they establish a common conceptual understanding. *Roles* are parts of the security concept of the architecture and define templates, which describe requirements on persons within the network. A role bundles access rights on resources and can be assigned to several persons.

Following the distinction of primary and secondary functions in a value chain [19], also the used data can be distinguished into primary and secondary data. *Primary data* support the business process execution and are related with the output of a collaboration: *Output data* describes results of collaborative business processes [4] and is required as input for creating process data and organizational data. *Process data* describe process models of the value-added network. In order to keep business secrets, these models are distinguished into local and global models. Local processes are intra-organizational processes of particular network enterprises, which belong to functions, roles and resources within the enterprises, though they have interfaces to the enterprise-external processes and resources. Global processes form the process structure of the overall network by aggregating the local processes at corresponding interfaces. *Organizational data* describe organization structures of the value-added network and can be distinguished into global and local structure models. Global structure models represent enterprises' relationships in the value-added network, whereas local models represent intra-organizational structures. *Business and case data* are, on the one hand, task-oriented resource data of the value-added network, which pass through the processes and will be processed to stand-alone products [7], e. g. documents or technical drawings. On the other hand, business data describe the task and network itself, e. g.

data about partner enterprises or cooperation structure. In a dynamic collaboration these data can continually change

Secondary data support process management and controlling: *Historical data* and *runtime data* collect defined execution data and possible exceptions due to process execution [20]. Historical data are about former processes and can be used to optimize processes of an actual network. Runtime data provides information about the current processes of the value-added network. Both the historical data and the runtime data support collaboration management and controlling (CMC). Finally, *identities* are basic elements for access control on data of the repository. They are digital features of individuals used for their identification at information systems.

2.2 Decentralized tools for accessing the repository

Output Description is realized by distributed available tools for designing output data. Depending on the output, different enterprises with individual processes are brought together to determine the output. The output description is also used for operational tasks like cost calculation, requests for quotation or accounting.

Process and Structure Design (modeling of the value-added net) is realized by tools for modeling and graphical visualization of processes and organization structures. Global models are stored in an abstract level and are centrally available, while local models are more detailed and only available for corresponding organizations. This can be realized either by decentralized storage of these models or by assigning roles and access rights. The created models are stored in the repository. Reference models can be imported from the repository into the modeling tools. Furthermore, ontologies enable consistent semantic modeling.

Collaboration Management and Controlling (CMC) is distinguished into Build-time-CMC and Run-time-CMC. Referring to a Collaborative Business Process Management Lifecycle [2], Build-time-CMC includes several steps of collaboration establishment, while Run-time-CMC encompasses collaboration execution. Following the distinction into a procedural and an organizational view of collaborations, the methods of CMC are divided into process-oriented and organization-oriented tasks.

Organizational tasks of *Build-time-CMC* are for example portfolio analysis, due diligence or boundary management for collaboration-, enterprise- and department-borders. Concerning collaborative processes, a unique project controlling has to be defined. This contains monetary methods like cost planning, revenue planning, budgeting and calculation as well as planning of collaboration-wide processes. For a pre-evaluation of procedural and organizational behavior of the collaboration we suggest a Petrinet-based simulation. Herewith all process models and organizational models can be validated and verified. The simulation can also estimate lead-times, costs and capacity utilizations, and it can deliver useful data to optimize processes and organizational structures a priori.

An important organizational task of *Run-time-CMC* is the steering of the collaboration partners' behavior. Used methods are e. g. collaboration-intern transfer prices or a repertory-grid-based soft-fact analysis to identify cultural weaknesses of individual enterprises concerning their cultural fit with the collaboration. Process oriented Run-

time-CMC-tasks are e. g. integrated progress controls, which include process monitoring, capacity control or performance measurement [4]. Concerning CMC in general, the repository can consist of collaboration-internal data as well as external data. The CMC-results can on the one hand be used to optimize the collaborative processes. On the other hand the results are used as a knowledge-base for the modeling of succeeding collaborations.

2.3 Process Execution

A **Virtual Service Platform** enables process execution and integrates different applications of the collaboration partners. Therefore process data and organizational models of the repository are used. By using local adapters, operational systems interact with each other without implementing a central coordinating instance. Corresponding applications are requested using the location service. The repository service reads relevant process parts and further data like business- or output data. Execution services use these process definitions to execute processes using application services provided by distributed operational systems. Hereby integration services convert different data formats. If necessary, integration services access ontologies. When the process execution in an adapter is ended, the next adapter is triggered to start the next process part. Therefore interface services arrange the interaction between the adapters.

Dynamic process changes can be realized by using a predefined concept for handling different disturbance types. This is stored within the repository and is triggered e. g. by an event-handler within CMC. Based on information stored in runtime data, responsible adapters and persons are determined. Finally, several possible ad-hoc-solutions are enabled, like defining an additional process or reconfigure the current process. The amended process can be continued within the adapter.

As a technical realization of the adapters, web service technologies are suitable [9]. Services are available by using Web Service Description Language (WSDL). Web services can interact with Simple Object Access Protocol (SOAP), based on internet protocols like Hypertext Transfer Protocol (HTTP) [21]. Business Process Execution Language for Web Services (BPEL4WS or BPEL) allows to define and execute web service-based workflows within the execution service [22], while Web Services Choreography Description Language (WS-CDL) defines collaboration between adapters [23].

A further part used for process execution is the **user interface**. A user in this case is a person or an organizational unit which fulfils functions and processes within the collaboration. Two different types of users can be distinguished. On the one hand they use presentation services as well as information services to get access to the repository (e. g. technical drawings, process definitions or visualizations of other data). On the other hand they use “traditional” user interfaces of operational systems to fulfill their tasks.

All data and models are stored in repository-wide unique data formats. By using converters different software can be integrated, although it does not support these standards. In particular data formats of process models and organization models have to fulfill several requirements: on the one hand different modeling tools with different

modeling languages can be used. On the other hand inter- and intra-organizational models may use different modeling languages. A uni-directional transformation of models not supporting the common process model format will be described in the following section.

3 Transformation of Business Process Models into BPMN

The presented architecture establishes the general framework towards the management and execution of collaborative scenarios. From a conceptual point of view, business processes have proven to be ideal design items in conjunction with the use of graphical methods and tools [24]. Using heterogeneous process notations within collaborative networks, the need for a central process notation standard emerges to reduce communication and transformation complexity. In the architecture (cf. chapter 2) these aspects are especially relevant to the process and structure design of networks with modeling tools and their connection to the distributed repository. While the architecture abstracts from model heterogeneity, the modeling notations, used in the collaboration, may differ from each other. The following solution presents an adaptable approach for transforming enterprise-specific process models into a collaboration-wide notation standard. The Business Process Modeling Notation (BPMN) specification developed by the Business Process Management Initiative (BPMI) provides an appropriate, standardized notation for representing business processes on a conceptual, near-business level [25].

Following, the Event-driven Process Chain (EPC) is chosen as an exemplary process notation for enterprise-specific process models. Other heterogeneous process notations which are applied in collaboration networks may be transformed in the same way as the EPC models in the example. The suggested procedure model for transforming process models into BPMN can be applied to a unidirectional, horizontal mapping of modeling notations. Underlying models ought to describe things at the same conceptual level to permit a horizontal integration. However, the existence of appropriate XML-representations of the involved notations is essential due to the need of formal process representations. This enables automatic interpretation and transformation by the modeling tools. We aim at a smooth transformation process with minor human interaction.

3.1 Meta Model Mapping

Achieving a collaboration-wide **agreement on the meta-models** of each of the process-modeling methods used by the partners, forms the **first step** towards the model transformation. Meta-models are documented for a majority of notations, but are often altered or enhanced by company-specific definitions. The meta-model for the common repository process format has to be defined - due to its creative nature - manually by modeling experts of all partners. Furthermore a common meta-model for modeling methods (e. g. EPC) should be defined in a unique way to ease later transformation and to disregard enterprise-specific adjustments. Thus, the resulting meta-

models help to harmonize the vocabulary of the notation constructs (cf. step two) and are a prerequisite for extracting mapping rules, which is done by defining corresponding process objects (cf. step three).

Second, the **usage of terms** has to be **unified** in order to reach a certain degree of semantic interoperability – by implementing semantic comparability and correctness – and to achieve a high model quality. This is realized by using ontologies within the repository of the presented architecture (cf. chapter 2).

The **third step** comprises the **meta-model mapping** as the conceptual transformation specification, realized by the design of object relations between the meta-model of the process notation for the repository and meta-models of process notations used in modeling tools. Therefore elements of one notation are related manually to corresponding elements of another notation. Future transformation rules can be extracted from these relations.

Table 1 presents an extract of the mapping of EPC to BPMN. Non-ambiguous, bidirectional relations are visualized with double-headed arrows, ambiguous, unidirectional ones use simple arrows. If exception rules are necessary for the mapping because of, e.g. the lack of syntactical and semantical unambiguousness, arrows are put in brackets.

Table 1. Object mapping between EPC and BPMN

EPC	uni- / bidirectional mapping	BPMN
Function	↔	Activity
Aggregated Function	↔	SubProcess
Basic Function	↔	Task
Event	← (→)	Start Event
Event	← (→)	Intermediate Event
...

3.2 Model Transformation

Process models, which have to be stored within the repository, are exported in a **fourth step** of the procedure model to a standardized exchange format – in this example from EPC to the Event-driven Process Chain Markup Language (EPML) [12]. The following code shows a part of the resulting, formal EPML-representation of the process, which is later transformed via meta-model mapping into BPMN.

Example of a formal EPML-representation

```
<?xml version="1.0" encoding="UTF-8"?>
...
<definition>
...
</definition>
<directory name="Root">
<epc epcId="1" name="business_process">
```



```

        <event id="1">
            <name>start_event</name>
            <description>start_event</description>
            <graphics>...</graphics>
        </event>
        <arc id="2" defRef="1" from="1"
            to="3"/>
        <function id="3">
            <name>function_one</name>
            <description>function_one</description>
            <graphics>...</graphics>
        </function>
        ...
    </epc>
</directory>
</epml:epml>

```

To export models via XML, also tool-specific interface languages as – for instance – ARIS Markup Language (AML) may be applied, but the complexity of the transformation process rises within networks with the use of proprietary formats. Thus, standardized instruments should be applied to achieve a simple and effective model transformation.

After the export, the **mapping** between two **XML-methods** is executed in a **fifth step**. Based on the rules predefined in step three, the XML-model (EPML) is transformed into another XML-based process markup-language as – for instance – Petri Net Markup Language (PNML) for Petri Nets or Business Process Modeling Language (BPML) for BPMN. In our example, the BPML is used as the target format [26].

As an example for the underlying analysis of transformation execution, the **task** sequence has to be extracted from the EPML-document by the analysis of relations between events (<event/>), arc relations (<arc/>) and functions (<function/>) and has to be transformed into the corresponding BPML-code. The sequence of EPML functions is transformed to the sequential <bpml:sequence>-form with the EPC starting event triggering the BPML sequence. Further corresponding objects, whose mapping was defined in step three, are converted into the new format. The definitions of exceptions and ambiguousness have to be considered within the XML-transformation. The results of the transformation are shown in the following code.

Transformation Result

```

<bpml:process name="business_process">
    <bpml:documentation>
        ...
    </bpml:documentation>
    <bpml:sequence>
        <bpml:event activity="function_one"
            name="start_event"/>
        <bpml:action name="function_one"
            operation="request"/>
        <bpml:action name="function_two"/>
    </bpml:sequence>
</bpml:process>

```

```

        <bpmml:action name="function_three"/>
        ...
    </bpmml:sequence>
</bpmml:process>

```

With the use of XML-data formats to exchange process model data, an eXtensible Stylesheet Language Transformations (XSLT) script, which transforms XML-documents from one format into another, can be implemented.

For the **sixth** and **final step** towards saving processes in the standardized BPMN format, the resulting XML-model (BPML) is **transferred to the repository**. Process data is now stored in a BPMN-compliant format. The information can be – for instance – exchanged between networking partners in a unified way with reasonable efforts towards communication complexity. The collaboration is executed on a common denominator for the conceptual description of business processes within collaborative scenarios.

4 Conclusions and Further Research

Based on the requirements for IT based support of collaboration networks an architecture for cross-enterprise business processes is given. Moreover, the prerequisites for the unique storage of Business Process Models are described with a generic procedure model for business process model transformations. The presented transformation concept provides an approach to

- solve the problem of heterogeneity in business process modeling by presenting a procedure model to gain syntactic model interoperability.
- consider current research efforts towards XML-based representations of business process models, as – for instance – it is done with EPML and BPML.
- take care of forward-looking standardization approaches, as they were presented by the BPMN – and consider at the same time well-known, established modeling techniques as the EPC to decrease investment risk for enterprises by merging “new” with established models.
- describe business model integration efforts on a conceptual level to get an open reference solution, independent of any fixed connection to certain methods. The approach may be adapted to other modeling notations, such as Petri-Nets or Activity Diagrams as far as a corresponding XML-representation is available and transformation makes sense in a syntactical and semantical way.

The approach does not claim completeness in terms of semantic integration and syntactic mapping covered due to the lack of an adequate formal XML-representation of BPMN and further essential research. It focuses rather on a general procedure model that shows how transformation in a unidirectional way can be conducted in order to facilitate the exchange of process models in heterogeneous environments. The scope of human interaction is minimized because it is only essential to those preliminary steps which can't be formalized like the agreements on the meta-model (cf. first step) or the mapping of meta-models (cf. third step). The model transformation itself may be executed smoothly with appropriate IT solutions (modelling tools, parser etc.).

Furthermore, problems as, ambiguity or other textual model defects may be avoided, which leads to a significant reduction in complexity and enables a more efficient planning- and design-task concerning BPM.

Further research is to enhance XML-based representations of standardized modeling notations. Thus, related representation and transformation approaches as, for instance, XML Metadata Interchange [12] have to be analyzed to improve the presented solution. Also the development of supporting tools, which ease the task of exchanging process models between different enterprises, i. e. to automate all possible mapping tasks by adequate rule-based systems is to be enforced. Furthermore, the vertical integration of process information through transformation and mapping of business concepts into IT-interpretable, formal process specifications is another field for further research. This comprises on the one hand a network-wide controlling for tracing processes over several interaction tiers. On the other hand reference models should be designed to facilitate the establishment of cooperation. At the same time a generic connection to infrastructures built on the augmented Service Oriented Architecture paradigm should be created. The presented approach and further research questions are discussed within the background of the research project ArKoS, raised by the German Ministry of Education and Research.

References

- [1] Picot, A., Reichwald, R., Wigand, R. T.: Information, Organization and Management: Expanding Markets and Corporate Boundaries. John Wiley & Sons, New York (1999)
- [2] Adam, O., Hofer, A., Zang, S.: Cross-enterprise Process Orchestration - Framework for Collaborative Process Automation. In: Proc. 1st International Workshop on Computer Supported Activity Coordination, CSAC 2004, In conjunction with ICEIS 2004 (2004) 185-197
- [3] Gronau, N.: Collaborative Engineering Communities - Architecture and Integration Approaches. In: Proc. of the 2003 Information Resources Management Association International Conference (2003) 792-795
- [4] Scheer, A.-W.: ARIS, Business Process Frameworks. 3 edn. Springer-Verlag (1999)
- [5] van der Aalst, W. M. P.: Making Work Flow: On the Application of Petri Nets to Business Process Management. In: Proc. Applications and Theory of Petri Nets 2002: Proc. of 23rd International Conference, ICATPN 2002 (2002) 1-22
- [6] Wild, R. H., Griggs, K. A., Li, E. Y.: A Web Portal/Simulation Architecture to Support Collaborative Policy and Planning Decision Making. In: Proc. Ninth Americas Conference on Information Systems (2003) 2400-2411
- [7] Hollingsworth, D. The Workflow Reference Model. Workflow Management Coalition [Online]. Available: <http://www.wfmc.org/standards/docs/tc003v11.pdf>
- [8] Linthicum, D. S.: Enterprise Application Integration. 4 edn. Addison-Wesley, Boston et al. (2003)
- [9] Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services - Concepts, Architectures and Applications. Springer, Berlin et al. (2004)
- [10] Pallos, M. S.: Service-Oriented Architecture: A Primer. EAIJournal 2001 (2001) 32-35

- [11] Object Management Group. OMG Model Driven Architecture.[Online]. Available: <http://www.omg.org/mda/>
- [12] Mendling, J., Nüttgens, M.: Exchanging EPC Business Process Models with EPML. In: Proc. 1st GI Workshop "XML4BPM - XML Interchange Formats for Business Process Management" at Modellierung 2004 (2004)
- [13] W3C. XSL Transformations (XSLT). W3C [Online]. Available: <http://www.w3.org/TR/xslt>
- [14] Kanungo, S.: Using Systems Dynamics to Operationalize Process Theory in Information Systems Research. In: Proc. Twenty-Fourth International Conference on Information Systems (2003) 450-463
- [15] Dedrick, J., Gurbaxani, V., Kraemer, K. L.: Information Technology and Economic Performance: A Critical Review of the Empirical Evidence. *ACM Computing Surveys* 35 (2003) 1-28
- [16] Mertins, K., Bernus, P.: Reference models. In: P. Bernus, K. Mertins, and G. Schmidt, (eds.): *Handbook on Architectures of Information Systems*. Springer (1998)
- [17] Fettke, P., Loos, P.: Classification of reference models - a methodology and its application. *Information Systems and e-Business Management* 1 (2003) 35-53
- [18] Kishore, R., Sharman, R., Ramesh, R.: Computational Ontologies and Information Systems: I. Foundations. *CAIS - Communications of the AIS* 14 (2004) 158-183
- [19] Porter, M. E.: *Competitive Advantage - Creating and Sustaining Superior Performance*. The Free Press, New York (1985)
- [20] Neumann, S., Probst, C., Wernsmann, C.: Kontinuierliches Prozessmanagement. In: J. Becker, M. Kugeler, and M. Rosemann, (eds.): *Prozessmanagement - Ein Leitfaden zur prozessorientierten Organisationsgestaltung*. Springer-Verlag (2003) 309-335
- [21] Blake, M. B.: Coordinating Multiple agents for workflow-oriented process orchestration. *ISEB - Information Systems and e-Business Management* 1 (2003) 387-404
- [22] Shapiro, R.: A Comparison of XPDL, BPML and BPEL4WS. *ebPML.org*, (2002)
- [23] W3C: Web Services Choreography Description Language Version 1.0. W3C., W3C Working Draft (2004)
- [24] Champy, J.: *X-Engineering the corporation : reinvent your business in the digital age*. Hodder & Stoughton, London (2002)
- [25] Owen, M., Raj, J.: *BPMN and Business Process Management*. (2003)
- [26] Arkin, A.: *Business Process Modeling Language*. BPML.org, (2002)