BAMBERGER BEITRÄGE

ZUR WIRTSCHAFTSINFORMATIK UND ANGEWANDTEN INFORMATIK

ISSN 0937-3349

Nr. 76

Applying Business Process Management Systems – a Case Study –

Gregor Scheithauer and Guido Wirtz

May 2008

FAKULTÄT WIRTSCHAFTSINFORMATIK UND ANGEWANDTE INFORMATIK

OTTO-FRIEDRICH-UNIVERSITÄT BAMBERG

Distributed and Mobile Systems Group

Otto-Friedrich Universität Bamberg

Feldkirchenstr. 21, 96052 Bamberg, GERMANY

Prof. Dr. rer. nat. Guido Wirtz

http://www.uni-bamberg.de/pi/

Due to hardware developments, strong application needs and the overwhelming influence of the net in almost all areas, distributed and mobile systems, especially software systems, have become one of the most important topics for nowadays software industry. Unfortunately, distribution adds its share to the problems of developing complex software systems. Heterogeneity in both, hardware and software, concurrency, distribution of components and the need for interoperability between different systems complicate matters. Moreover, new technical aspects like resource management, load balancing and deadlock handling put an additional burden onto the developer. Although subject to permanent changes, distributed systems have high requirements w.r.t. dependability, robustness and performance.

The long-term common goal of our research efforts is the development, implementation and evaluation of methods helpful for the development of robust and easy-to-use software for complex systems in general while putting a focus on the problems and issues regarding the software development for distributed as well as mobile systems on all levels. Our current research activities are focussed on different aspects centered around that theme:

- *Robust and adaptive Service-oriented Architectures*: Development of design methods, languages and middleware to ease the development of SOAs with an emphasis on provable correct systems that allow for early design-evaluation due to rigorous development methods and tools. Additionally, we work on approaches to autonomic components and container-support for such components in order to ensure robustness also at runtime.
- *Agent and Multi-Agent (MAS) Technology:* Development of new approaches to use Multi-Agent-Systems and negotiation techniques, for designing, organizing and optimizing complex distributed systems, esp. service-based architectures.
- *Context-Models and Context-Support for small mobile devices*: Investigation of techniques for providing, representing and exchanging context information in networks of small mobile devices like, e.g. PDAs or smart phones. The focus is on the development of a truly distributed context model taking care of information reliability as well as privacy issues.
- *Peer-to-Peer Systems*: Development of algorithms, techniques and middleware suitable for building applications based on unstructured as well as structured P2P systems. A specific focus is put on privacy as well as anonymity issues.
- *Visual Programming- and Design-Languages*: The goal of this long-term effort is the utilitization of visual metaphores and languages as well as visualization techniques to make design- and programming languages more understandable and, hence, easy-to-use.

More information about our work, i.e., projects, papers and software, is available at our homepage. If you have any questions or suggestions regarding this report or our work in general, don't hesitate to contact me at guido.wirtz@uni-bamberg.de

Guido Wirtz

Bamberg, April 2008

Applying Business Process Management Systems – a Case Study –

Gregor Scheithauer and Guido Wirtz

Lehrstuhl für Praktische Informatik, Fakultät WIAI

Abstract Business Process Management Systems (BPMS) aim to support the Business Process Management paradigm and to ease legacy application integration. Often, they rely on Service-oriented Architecture (SoA). However, do such systems really meet real-world requirements? This paper introduces and discusses a set of criteria which are important for BPMS and applies these criteria in comparing tools from three important vendors, namely *IDS Scheer*, *Oracle* and *Intalio* based on a case study.

Keywords BPMS, usability criteria, Webservice, BPM, SoA

Note: This technical report is an extended version of the paper *Case Study: Applying Business Process Management Systems* that is published in the Proceedings of SEKE'08 – the 2008 International Conference on Software Engineering and Knowledge Engineering, to be held from 1.–3. July 2008 in Redwood City, CA (USA).

Contact: gregor.scheithauer@gmail.com, guido.wirtz@uni-bamberg.de

Contents

1 Introduction

Business Process Management Systems (BPMS) [1] are sets of tools to support the Business Process Management (BPM) life-cycle [2] that are either offered by one vendor, or multiple vendors. Smith [3] sees a list of key advantages in using a modern BPMS: it bridges heterogeneous application environments, includes human activity by incorporating workflow, allows web service orchestration, provides the opportunity to customize the whole process for specific customers and partners, offers an integrated user interface through a single portal and back-end integration, and monitors process instances. Rather than introducing new technology or replacing existing business applications, BPMS integrate existing technologies and existing applications in a process-oriented fashion. Based on this notion of BPMS, Smith and Fingar [1] describe requirements for a BPMS as follows: a BPMS should be able to support modeling, deploying, and monitoring business processes, as well as to support integration of heterogeneous processes, automation, and collaboration.

Business process design includes process documentation with a process notation, such as Event-driven Process Chain (EPC) [4] notation and Business Process Modeling Notation (BPMN) [5]. Configuration includes the transformation [6] from process models into formal languages such as the Business Process Execution Language (BPEL) [7]. Integration facilitates better reuse of existing applications. BPMS allows easy deployment of configured process models, and to execute them.

This paper summarizes the results of a case study to find out if prominent existing BPMS meet real-world expectations. The rest of this paper is structured as follows: Section 2 and 3 introduce stakeholder roles and a life-cycle. Both concepts are used to further explain BPMS and to implement the case study. Section 4 offers a detailed list of evaluation criteria covering all steps relevant for BPMS. A real-life scenario is introduced in section 5. Section 6 shows the case study implementation with two different BPMS - a multi-vendor system based on tools from *IDS Scheer* and *Oracle* as well as a single vendor system provided by *Intalio*. The results are discussed in section 7. Section 8 concludes this work.

2 BPM Stakeholder

This section explains different roles which take part in the BPMS life cycle. Six roles are identified and described in the following paragraphs. It is likely that a person has more than one role, especially in small companies, or that complete teams represent one role [8, 9]. Four categories are consulted to explain stakeholder roles. The *background* of a person refers to the kind and to the level of education, and the nature of operating experience the person has adapted. *Knowledge* covers the knowledge a person has in form of concepts, techniques, and languages. *Function* refers to responsibilities, involvements and sponsorship of a person. *Relationship* highlights the relationship between stakeholder roles.

The role of a *business owner* represents the CEO of a company or head of an independent business unit. The role comprises domain knowledge, such as knowledge about the market and the company's product as well as high level business knowledge. In general, this person owns a university degree in business studies and has years of experience in the company's business domain. The person employs high level business concepts, such as strategic management. Business owners use a language which is filled with terms like business strategies, business goals, and business rules. No technical knowledge regarding software applications or process tools is present. However, the person might be familiar with business intelligence tools such as OLAP (On Line Analytical Processing) tools. The role's functionality is to discover strategies, to have visions, to determine business rules, and business goals. The role is involved in the process discovery step, the process monitoring step, and the process improvement step of the process management life cycle. Business owners work with process owners close together in order to create high level business strategies and to break them down to strategies on a business department level, to observe process performance, and to improve process performance.

The role of a *process owner* represents a department head, or someone who is responsible for a complete business process. The role comprises a high level of domain knowledge and years of experience in the business domain. Process owners own a business degree in the profession of the business domain. They think in terms of product and price strategies, value chain configuration and profit, and are able to use different kind of general business information technology to achieve business objectives, such as planning tools, financial tools, and process monitoring tools. Process owners use a language which is filled with the terms of business goals, tasks, business functions, and business objects. The role's functionality is to break high level business strategies and business goals down to strategies which can be applied to the business department. The person is involved in the following steps of the process management life cycle: Process discovery, process design, process execution, process monitoring, and process improvement. Process owners work with business owners, business analysts, system analysts, and process participants.

Business analysts' role represents an information manager. Analysts might be in an functional department, but might also work for a cross-functional division. Business analysts have a general business degree, but also some technical experience with knowledge management tools. Business analysts possess experience in understanding non-formalized process sequences. Business analysts think in terms of business processes, in business functions, in business rules, in business objects, in input and output, in dependencies, and business partner integration. They use process design notations to expose and document processes for communication, but are also capable of transferring business strategies and business goals into process diagrams. The business analysts' functions are both, to document business processes and to transform business strategies and business goals into business processes as a way to implement the business strategy. Furthermore, setting up metrics and performance measure concepts, such as balance score card. They design process documentation on different levels, by decomposing high level processes for business owners and process owners into low level processes for process participants. The end of the decomposition is so called *atomic business tasks*, which are business tasks which cannot be decomposed anymore without loosing business relevance. Business analysts work close together with process owners, system analysts, but may also work with software developers at times.

System analysts' role is not yet easily found in companies. System analysts may work in IT departments or information offices. Another possibility would be, that system analysts are part of a software development project. System analysts definitely have a technology oriented degree, and are familiar with middleware technology and concepts. The person might come from a developer position, rather than from a business operational profession. Though system analysts have a more technology oriented background, they do understand concepts regarding business processes. They think more in terms of services, conditions, input and output, and message types. Analysts are very familiar with information and knowledge tools, such as process modeling tools, development environments, middleware technology, and integration technology. They do understand business language to some degree. Though they do not have detailed insight in complex business strategies and high level business processes as a whole, they do understand low level business tasks. The semantic of atomic business tasks is self contained and to understand without knowledge of the processes which use these atomic business tasks. In addition, they do understand also process design notations, execution languages, and message and integration concepts, such as web services and message types. System analysts are the link between process documentation and process execution. Their function is to transform process documentation into executable processes. To do that, they do not use classical programming languages, but utilize execution languages, message protocols, and middleware solutions. System analyst interact rarely with process owners, but very much with business analysts, and with software developers.

Software developers work in software development project groups or for the IT department. They have a technology oriented degree, e.g. information system studies. Developers have experience with programming languages, such as Java, C++, and .NET. They are familiar with XML standards, such as WSDL [10], SOAP, XSD, and BPEL. Software developers use integrated development environments (IDE) during development. Their language is filled with terms like, software patterns, application functions, loops and cases, and data types. As requested by system analysts and business analysts, software developers develop new application functions and wrap them for examples as web services. They are also involved in the process deployment.

Process participants are involved in the process execution part. They are employees and take over operational business tasks. Participants are familiar with business applications.

3 BPM Life Cycle

Figure 1 [2] shows the BPM life cycle which is used for the case study implementation. The life cycle is introduced below.

Process discovery refers to the detection of business goals and strategies in order to conduct a business. A way to formalize goals and strategies are either ontologies or a business model [11].

Process design refers to the transformation of goals and strategies into a process diagram on the business level. In order to perform this step, two intermediate steps are required: first, business analysts will need a business process design notation. In this context, this might either be the BPMN or the EPC notation. Subsequent, business analysts will use their experience, design paradigms, and the purpose of the documentation for transforming the informal goals,

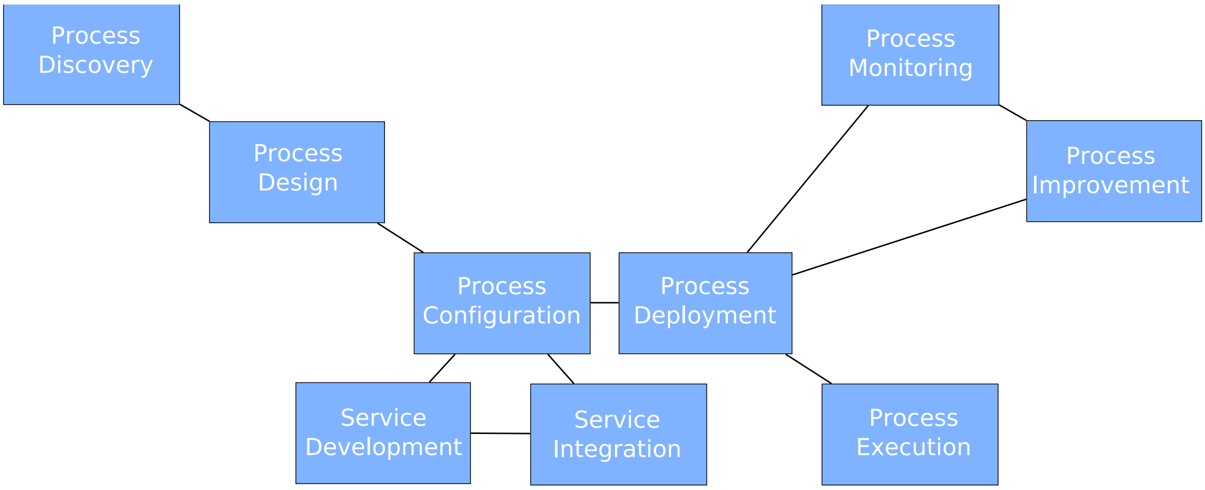

Figure 1: Nine Steps of BPM Life Cycle.

strategies, and rules into a process documentation. The design paradigm is process-oriented. Business analysts decompose tasks and objects to substantiate business processes. Second, they need to simulate the process in the diagram to verify the semantical behavior of the process.

Process configuration refers to the transformation of process documentation into a platform independent process configuration. It implies the mapping strategy on the integration level. It is intended to transfer the process documentation as complete as possible, not to change any business logic, and to reuse existing implementations. To achieve this step, system analysts need to perform four steps: First, the process documentation is transformed into a technical representation. At this point, a switch in notation is not necessary. The configuration tool offers additional constructs to enrich the process documentation with technical details. Hence, the flow of business tasks (service composition) can be derived from the business process diagram. Second, web services are mapped to atomic business tasks. The integration is is done by using middleware technology, such as web services. Third, message descriptions need to be applied to web services. They also need to be maintained and integrated by the system analyst. Fourth, system analysts need to validate the work and build the process configuration. In case no service for an atomic business task is available, two possibilities are conceivable: consultation of a software developer, who develops an application function to address the business task's requirements, or to integrate a service from a service provider.

Service integration follows either the development step, or the configuration step. It supports the integration strategy. Integration is twofold. Both, developed services, and services from business partners need to be integrated before they can be used for process configurations.

The development of an application function is called *Service development*. This is wrapped as a service and fulfills the requirement of an atomic business task. The objective is to develop a service which is not bound to a specific business process. The service development results in a self contained service which is easy to reuse. Since business analysts break business processes down into loosely coupled business tasks and decompose business tasks (treated as business processes) recursively into atomic business tasks, the whole complexity and semantic is broken down into seizable and easy to communicate requirements, which are understood by software developers. Software developers grasp the requirements and build an application function using programming languages such as Java, or .NET and wrap this function as a reusable service.

Process deployment refers to the final transformation of the process configuration onto a platform. It supports the transformation strategy. However, the deployment should not have an influence on other running processes on that platform. The executable process configuration needs to be transfered to the execution engine. The configuration of other middleware technologies, which are involved in the process, as well as security policies, is optional.

Process execution alludes to the state where configured business processes are executed to support business processes. The procedure to execute a process differs in terms of how processes are triggered and whether the service is known in the first hand. First, process consumers need to know about the service. Second, processes need to be triggered. Process might be triggered by an event, other services or by human interaction through a process portal. Third, running processes are referred to as process instances. Process instances save the state of a process, as well as message and variable values.

The tracking of the process performance and to display it in a human understandable format is referred to as *Process monitoring.* The formal goals are to highlight bottlenecks and offer suggestions for action in order to improve process performances. Often, a dashboard is used as a paradigm. Business owners should be in the position to view the performance of process instances on a consolidated level. It should be allowed to break the level down to a single instance which may be a source of failure or a bottleneck. Business owners are then allowed to take action, e.g. improve the process.

Process improvement points to the adaption of processes due to a changing situation or environment. Objectives are to improve execution time, execution cost (total cost of ownership), and process efficiency. Business analysts and business owners gather new information and feedback from process monitoring, process users and business owners. They change the process diagram according to the new requirements. The new process diagram needs to be validated. Moreover, depending processes need to be checked whether they are affected by the current change of the original process. Adapted processes are configured and deployed for execution.

4 Evaluation Criteria

The criteria used to evaluate BPMS tools take a holistic view on the entire process. The 23 criteria are clustered into three layers which are introduced in [2]: questions 1–9 cover the business layer, questions 10–19 address the integration layer, and questions 20–23 address the execution layer. The questions represent real-world requirements originating from an industry project. These criteria are used to evaluate the different tools when realizing the example process that is introduced next.

4.1 Business Level

1. *What kind of people are involved during design and improvement in the BPM life cycle.* These steps need to be business driven, and flexible, thus, people who manage business processes, need to be in the position to express their understanding of business, without technically founded limitations.

2. *Standard or proprietary design notation* points out if the process design notation in question was standardized by a group such as OMG or OASIS, or if it is a vendor specific format. Moreover, does the standard cover the graphical elements and the persistence of the notation? By using a standard notation, it is easy to switch process design tools or exchange process diagrams between different process design tools.

3. *Industry acceptance* shows if a process design notation is widely used in industry. Established notations are more likely to provide supporting technologies and middle-ware. In addition, if a design notation is widespread, it might undergo further and constant improvements.

4. *Completeness* of process design notations denotes the expressive power of a notation (cf [12]). Business analysts need elements to express business tasks, business objects, and business partners. Missing elements result in complex process diagrams emulating missing constructs, which are difficult to maintain.

5. *Data management.* indicates the possibility to design business objects with the process design tool. Business objects make the process diagram semantically richer and better to understand for process stake-holders.

6. *Is a methodology behind process design notation.* A methodology covers the semantics of the notation and reduces the complexity of business process design via guidelines how to use and how to combine the elements of the notation.

7. *Does the design tool support the full design notation in its recent version.* The more a tool supports a design notation standard, the greater the ability to exchange process diagrams.

8. *Diagram repository* states if the process design tool accesses diagrams from a shared repository or from a local machine. A process repository has the advantage that more people are allowed to access processes, thus, processes are viewed and re-viewed by more people, which, to an extend, improve process diagrams.

9. *Process version control* shows if a process design tool contains or has access to version control. Next to a shared repository, this is a very useful tool for maintaining process diagrams. Business analysts are able to roll back to a prior version of the process, if necessary, or browse the evolution of a process for a better understanding of the meaning behind the current version.

4.2 Integration Level

10. *What kind of people are involved in the configuration and the integration step of the life cycle.* Process diagrams should not be altered much to be executed. No change in business logic should be needed, but a technical mapping is required. People on this level must not be faced with the complexity of business logic.

11. *Compatibility of design notation and execution language* refers to what extent the design notation is transformable into the execution language. There are two main reasons for incompatible languages: (i) languages are either block-oriented or graph oriented or, (ii) languages may support different concepts and use richer semantics.
12. *Standard or proprietary execution language* points out if the execution language in question was standardized, or if it is vendor specific. This covers the language and the persistence of the language. Using a standard language eases switching execution engines or exchanging process configurations between different engines.
13. *Industry acceptance* shows if an execution language is widely used in industry. Besides the importance to use standards, it is necessary to find supporting technologies and middleware to support execution languages.
14. *Message type management.* Is it possible to design or even import message types with the configuration tool? Next to configure the flow of business tasks between applications, departments and companies, it is necessary to define message types. These types may be imported from service definitions, database table definitions or class definitions from a programming language. Otherwise, they might be defined with the configuration tool.
15. *Configuration complexity* measures how many tools are needed for a successful process configuration. Besides an integrated configuration tool, it may necessary to apply configuration to other middle-ware before deployment is possible. The more tools and middleware need to be configured, the higher the complexity.
16. *Is process configuration part of a shared repository.* This criteria points out if the configuration tool accesses process configurations from a shared repository or from a local machine. The former has the advantage that more people access the configuration, thus process configuration might be adapted by many people.
17. *Is the process configuration attached to the process diagram.* If there is a well-defined link between a process diagram and the process configuration, changing the diagram as well as the configuration consistently becomes much easier.
18. *Is process configuration bound to one execution platform* refers to the *vendor lock* issue. This is the case, if process configurations are only be executed on the platform which the process diagram was configured with. This may happen if execution engines do not support standards or industry accepted execution languages. A vendor lock makes it difficult to switch between different execution engines.
19. *Legacy applications integration* explains what kind of applications and their services may be integrated. However, middle-ware technology makes it possible to integrate those application as services.

4.3 Execution Level

20. *What kind of people are involved in the deployment step.* System analysts should be qualified to accomplish this task. If other than the system analyst needs to be involved, process deployment is a too complex step.
21. *Deployment tool integration* tells whether a deployment tool is integrated into an IDE or not. Users do not need different tools, the acceptance of the user is higher and users already know how the tools behave.

22. *Deployment complexity* measures how many tools are needed for a successful process deployment. Next to an integrated deployment tool, it may be necessary to deploy to more than one execution engine. The more deployment steps are required, the higher is the complexity for process deployment.
23. *Process version control.* This refers to what will happen if instances of a process are running and a new version of that process will be deployed. There are four possibilities. Firstly, all instances are stopped and deleted. The new process will be deployed. Secondly, a deployment of a new version is refused, when instances of that process are still running. Thirdly, the tool tries to merge running instances with the new process definition. If a merge is possible, the new version will be deployed, otherwise the deployment will be refused. Running instances may run until they terminate. New instances are based on the new version of the process. The old version of that process will be archived when every instance has been terminated.

5 Case Study

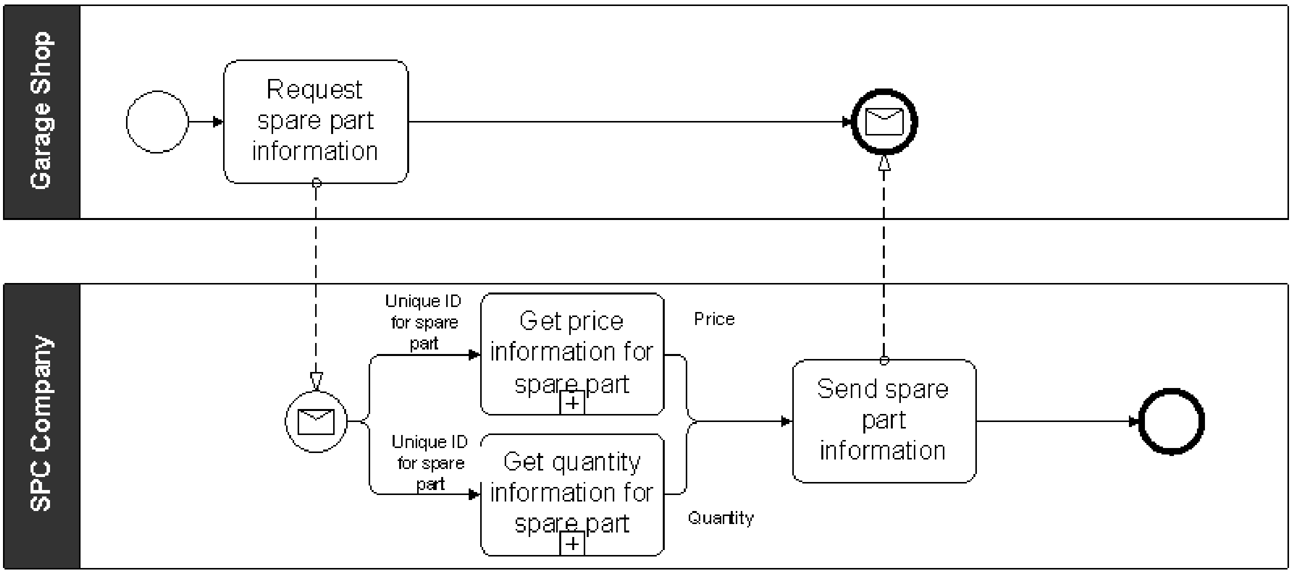
Garage Shop
Request spare part information
SPC Company
Unique ID for spare part
Get price information for spare part
Price
Unique ID for spare part
Get quantity information for spare part
Quantity
Send spare part information

Figure 2: Case Study Process.

Two companies are involved in the case study: *Shade Tree Garage* (STG), a garage shop in New Jersey, repairs cars for nearly all makes of cars whereas the *SPC* company manufactures car spare parts and distributes them to garage shops. Prices for spare parts are not fixed and change on a daily basis. Shade Tree Garage wants to minimize its stocking costs and to maximize planning reliability. SPC identifies this demand as a selling proposition, and intends to offer a *Garage Shop Information System* (GSIS) to garage shops.

The business process, which is shown in figure 2, offers price information and quantity information for spare parts to garage shops. On the business level, the following business tasks are identified: (1) *Request spare part information* on the garage shop side, (2) *Receive spare part information request*, (3) *Get price information for spare part*, (4) *Get quantity information for spare part*, and (5) *Send spare part information* on the SPC side. The business objects include (1) *unique ID for spare parts*, (2) *Price*, and (3) *Quantity.* On the service level, two services are

	Design	Configuration	Integration	Deployment	Execution
IDS Scheer (ARIS)					
SOA Architect		X	X		
Business Architect	X				
Oracle					
Process Manager					X
BPEL Designer		X	X	X	
Intalio					
Process Server					X
Process Designer	X	X	X	X	

Table 1: BPM tools in the process life cycle

needed: (1) *PriceService*, and (2) *QuantityService*. Both services are available as web services and provide a WDSL file. The appropriate message exchange pattern between SPC and garage shops is a Request-Response pattern. To access the GSIS, the *GSISRequestMessage* is used which contains a placeholder for a spare part ID. Spare part information is received by the *GSISResponseMessage* which contains a placeholder for price and quantity information. The case study comprises an end-to-end business process that contains reasonable business logic and has relevance in today's business. Moreover, it spans more than one company's department and more than a single application. Hence, it is suitable to check technical capabilities and business to business integration issues.

6 Case Study Implementation

This section implements the *Garage Shop Information System*. The BPMS tools used in the life cycle are depicted in table 1. The table shows that Intalio's BPMS covers all steps in the life cycle, whereas the tools from IDS Scheer and Oracle only in combination cover the whole life cycle.

6.1 Mulit-Vendor BPMS

Process Design: The business analysts from SPC starts to design the overall business process with the *ARIS Business Architect*. The ARIS Business Architect allows to design business processes with the EPC notation. It is possible to express business tasks and business objects, and to refine these elements. The business analyst designs the business tasks (1) *Get price information for spare part*, and (2) *Get quantity information for spare part* as functions in the EPC diagram. Following this, the business analyst designs the business objects (1) *unique ID for spare parts*, (2) *Price*, and (3) *Quantity* as information objects in the EPC diagram. It would be possible to further refine those elements. However, that is not required at the moment.

Service Integration: With the *ARIS SOA Architect* the system analyst integrates WSDL files as services and XSD files as message type definitions into the ARIS repository. The *PriceService* and the *QuantityService* is integrated along with the message type definitions used as inputs and outputs, which are not further explained. Services and message type definitions are then ready to be mapped to business tasks and business objects.

Process configuration: The system analyst uses the *ARIS SOA Architect* to access the business diagram from the ARIS process repository. He studies the business task, and the business objects. For the business tasks suitable services need to be discovered. The system analyst queries the ARIS database with keywords and natural language. The *PriceService* is selected and mapped to the business task *Get price information for spare part.* The *QuantityService* is selected and mapped to the business task *Get quantity information for spare part.* The business object *unique ID for spare parts* is mapped to the message type definition *GSISRequestMessage*, which contains a placeholder for a spare parts ID. The business objects *Price*, and *Quantity* are mapped to the message type definition *GSISResponseMessage*, which contains a placeholder for the price and quantity of a spare part. Services and message type definitions are now part of the business process diagram. The system analysts now transforms the enriched EPC diagram into a BPEL skeleton and a corresponding WSDL file. Both files are imported into Oracle's BPEL Designer for further configuration, such as assigning service output's messages to service input's messages (message mediation).

Process deployment and process execution: After fully process configuration, the system analyst uses the BPEL Designer to deploy the process to Oracle's Process Manager. The process is now accessible at the process portal which is part of the Process Manager. Shade Tree Garage accesses the process portal, select the GSIS process, provide a spare parts ID, and receive the current price and quantity for the requested spare part.

6.2 Integrated BPMS

Process Design: Intalio's Process Designer embodies all functionality from process design to process deployment. The tool allows business analysts to design business process diagrams using BPMN. SPC's business analyst designs business tasks for both, the GSIS side and the garage shop side. Figure 2 shows the resulting diagram. Business objects are annotated in terms of text next to business tasks.

Service Integration: The system analyst integrates WSDL files as services and XSD files as message type definitions into the project work space within the file system. The *PriceService* and the *QuantityService* is integrated along with the message type definitions used as inputs and outputs, which are not further explained. Services and message type definitions are then ready to be mapped to business tasks and business objects.

Process configuration: Referring to the pools shown in figure 2 the pool *SPC company* is set to executable, and the pool *Garage Shop* is set to not executable, respectively. This indicates that the tasks on the executable pool are intended to be supported by an automated service. The non-executable pool represents the human interaction with the executable pool. Following that, SPC's system analyst searches for an appropriate service in the project space. Only the information within the WSDL files are shown in the project space. On this basis the system analyst needs to discover the best matching service. Selected services are then mapped to business tasks by drag and drop. The system analyst maps the *PriceService* to the business task *Get price information for spare part*, and the *QuantityService* to the business task *Get quantity information for spare part.* The *GSISRequestMessage* which contains the spare part's ID, is mapped to the first message connection between both swim lanes. The *GSISResponseMessage*, which contains the spare part's price and quantity, is mapped to the second message connection. Following that, the system analyst takes care of message mediation. The unique ID for spare parts from the *GSISRequestMessage* is mapped to the *PriceService*'s and *QuantityService*'s input parameter. The *PriceService*'s and *QuantityService* output parameter is mapped to the *GSISResponseMessage.*

Process deployment and process execution: The system analyst directly deploys the process from the Process Designer to Intalio's Process Server with a single click. Shade Tree Garage accesses the process portal, select the GSIS process, provide a spare parts ID, and receive the current price and quantity for the requested spare part.

7 Results

This section applies the BPMS evaluation framework, which was introduced in section 4, to the BPMS tools, which were introduced in section 6, on the basis of the case study in section 5.

7.1 Business Level

IDS Scheer's ARIS Business Architect provides adequate support for business analysts, process owners, and process participants. Business analysts are in the position to design and improve business processes by using the tool without any technical knowledge to decompose business tasks and business objects. The ARIS filter concept supports the separation between business elements and technical elements from the EPC notation. This makes it possible to keep process documentation and process configuration in only one diagram. Intalio's BPMN Designer offers a tool to design business processes using BPMN. Business analysts are able to design business process without technical hassle. However, once the process is configured, process diagrams contain technical elements which are not hidden from business analysts. In consequence, business analysts are not as flexible anymore to improve processes, and a technical barrier is created which might hinder business analysts to even use the process diagram any further. This fact blurs the business level and the integration level.

ARIS products from IDS Scheer support the EPC notation, which is well-known in the industry. Slowly, BPMN becomes more and more accepted in the industry and by business analysts, since more tools become available and the swim-lane concept is supported, which is missing in the EPC notation. However, unless OMG will not come up with an exchange format, it will be difficult to convince decision makers to introduce BPMN as a design notation. The lack of an exchange format hinders attempts to standardize transformations between BPMN and process execution languages.

The EPC notation is complete. Business tasks, business objects, business partners are covered. Business tasks are expressed with *functions*, business objects with *information objects*, and business partners with *organizational units*. Relationships between the terms are expressed with the *control flow*, the *data flow*, and *logical operators*. Both, the ARIS Business Architect and the ARIS SOA Architect support these requirements. BPMN is also complete. Business tasks are expresses with *activities*, business objects with *data objects*, and business partners with different *pools*. Relationships between these terms are expressed with the *sequence flow*, the *message flow*, and plain *associations*.

Besides business processes, business analyst must be capable to design business objects to document companies' information systems. As mentioned earlier, EPC supports information objects to define information. Furthermore, ARIS tools allow to decompose business objects by linking a new diagram to a business object. Intalio's BPMN Designer does not support data objects. This is an issue, which limits the flexibility of business analysts and limits the decomposition strategy between the business level and the integration level.

A process design methodology supports business analysts during the design step. The EPC notation is part of the ARIS HOBE[1] developed by Scheer and Nuettgens [4]. EPC diagrams are embedded in the methodology as the controlling view, next to other views on information systems. BPMN is not in possess of a methodology. However, the BPMN specification from White [5] covers the semantic of each BPMN element and how to combine different elements.

Due to the later transformation of process diagrams into execution languages, tools limit elements to those, which are transferable. The tools from IDS Scheer support the full EPC notation in general. However, EPC diagrams which are intended to be transformed into BPEL must not use the logical `OR`. Only the `AND`, and `XOR` are allowed. Intalio also limits BPMN to elements which are transferable into BPEL.

A process repository supports business analysts, process owners, and business owners to exchange and review business processes. IDS Scheer's ARIS tools come along with a global database. The database stores any diagram and keeps them consistent. Moreover, the database supports user rights management. Thus, stakeholders are qualified to access process diagrams regardless the used ARIS tool. Intalio's BPMN Designer does not support a process repository itself. Since the designer is based on the Eclipse's rich Client Platform, it is possible to integrate other tools to support process diagram exchange. That might be Concurrent Versions System (CVS). Neither IDS Scheer nor Intalio offers process version control.

Table 2 summarizes the findings.

7.2 Integration Level

The configuration step, the integration step, and the deployment step, should be operated by system analysts as mediators between business logic and application support. The ARIS SOA

[1] House of Business Engineering

	ARIS Business Architect	Intalio BPMN Designer
1.	Process owner, business analyst, process participant	Business analyst
2.	EPC; not a standard, was published in 1992, XML based, thus open.	BPMN is a standard from OMG, may replace UML activity diagram, only standard for graphical elements, not easy exchanged.
3.	EPC is accepted in the industry for documenting and communicating processes within a company. Recently, SAP and Oracle use ARIS tools to enable business analysts to interact with middleware.	BPMN is seen as a workflow definition language, since it is very rich in graphical elements. Business people hesitate to use it since it is so rich at technical elements. Tool support is available.
4.	Complete	BPMN in general.
5.	Possible	Possible with limitations
6.	Architecture Integrated Information Systems (ARIS)	Specification with instructions how to use the notation
7.	Full support	Limited support
8.	Global or local diagram repository	No repository
9.	No version control	No version control

Table 2: BPMS evaluation – Criteria 1-9

Architect and the BPEL Designer allow system analysts to configure, to map and to transform business processes, and to integrate services. System analysts will consult business analyst for support during the mapping process, and software developers to communicate requirements for services and message types. However, due to appearances of errors during process deployment, software developers need to be consulted for interpretation. In addition, due to the tool switch, during the configuration step, system analysts might need support for export and import of BPEL code. System analysts who use Intalio's BPMN Designer, are able to map atomic business tasks to services, to integrate services onto the integration level, and to deploy the process configuration to the execution platform. However, in case, process configurations do not validate, system analysts are faced with rather technical messages which need to be interpreted by software developers who are more familiar with this kind of messages. System analysts using Intalio's BPMN Designer, however, need more support from business analysts, since system analysts are not well supported during the search for appropriate services.

To transform process configuration into an executable process without troubles, process design notations and execution languages need to be compatible. Since the EPC notation is not as rich in elements as BPEL version 1.1, the transformation is easy, though it is not possible to express as much as needed to use only the EPC notation for process configuration. This is the major reason why a BPEL designer is necessary in the first place. The EPC to BPEL transformation's result is a service orchestration which may be used as a skeleton for further technical configuration. Partner links, business tasks, and message types are part of this BPEL skeleton. Some adoptions are not part of the global database, such as message mediations, and in consequence, this configuration will be lost, in case of process improvement. Furthermore, in case a business analyst uses the logical operator *OR*, system analysts must alter the business logic to avoid the operator to be able to transform the EPC diagram. This fact limits the business-driven goal of BPM. Intalio's BPMN Designer only supports a limited BPMN specification to allow a transformation from business process diagrams into BPEL. The BPEL version 2.0 was published in January 2007. Intalio supports this version of BPEL since it is a richer language than the former version. According to Ouyang et al. [6], it is possible to translate a subset of BPMN into BPEL version 1.1. This stresses the point that the combination of BPMN and BPEL was not intended in the first place. Both approaches face limitations regarding transformation, due to the fact that a graphic-oriented notation needs to be transformed into a block-oriented language.

Oracle's BPEL Designer is able to transform BPEL version 1.1 code into Java Byte Code It is possible to use the BPEL code in other BPEL processing tools. Further, BPEL version 1.1 is widely accepted in the industry. Many tools are able to process BPEL code. That makes it possible to use process configuration on other platforms. Intalio, however, make use of BPEL version 2.0. As mentioned earlier, it is a very young specification, and in consequence not well supported by configuration tools, execution engines, and vendors. In consequence, it is not easy to switch to another execution engine. However, since process configuration is conducted using BPMN, it is possible to reuse process configuration in other platforms, once a persistence standard for BPMN is established. Both execution languages, BPEL version 1.1 and version 2.0 are OASIS standards. Thus, it is possible using tools to extract information from the execution language and to reuse it in some way.

For message type management, IDS Scheer's ARIS SOA Architect, Oracle's BPEL Designer, and Intalio's BPMN Designer rely on XML Schema Definition (XSD) files. It is possible to integrate these files into the configuration project, by means of importing files from the file system. It is not possible to alter XSD files with the SOA Architect. Since the BPEL Designer is based on Oracle's JDeveloper, and the BPMN Designer on Eclipse, it is possible to use build-in editors to modify XSD files, and thus, alter the data. It is not desired to modify message types frequently. A better idea is to extract message type definitions from existing data bases.

The complexity of process configuration is an important point, since this step needs to be repeated for every change in the business logic. As for the IDS Scheer-Oracle approach, system analysts use the SOA Architect for mapping, integration, and transformation. The tool switch during configuration increases the process configuration complexity. The BPEL Designer support system analysts during the further transformation and deployment. For the system analysts using Intalio's BPMS, only the BPMN Designer is used during the configuration step.

A global service repository allows collaboration between different stakeholders. The SOA Architect has access to the ARIS database. Each diagram is accessible for any ARIS product. However, once system analysts switch to the BPEL Designer, further process configuration is not part of the repository. Thus, every configuration done in the BPEL Designer is overwritten every time business logic changes. Intalio's BPMS does not contain a process configuration repository.

Linkage of process design diagrams and process configuration reduces the overall complexity of BPMS. Since some parts of process configuration are done in the EPC diagram using the SOA Architect, the process diagram and this process configuration is linked to some extent.

	ARIS SOA Architect, Oracle BPEL Designer	**Intalio BPMN Designer**
10.	System analyst, software developers	Business analyst, system analyst, software developer
11.	EPC → BPEL 1.1 Poorer to richer semantic translation	BPMN → BPEL 2.0 Different semantics
12.	BPEL 1.1, Specification from IBM & Microsoft	BPEL 2.0, Specification from OASIS
13.	Accepted language, great tool support, missing concepts, such as human people integration	BPEL 2.0 Specification not yet adopted, no other tool support
14.	Complete	Complete
15.	Configuration with two major tools necessary	Configuration with one tool
16.	Global or local service repository	Local service repository
17.	Process diagram and process configuration are linked, though only the configuration done in the SOA Architect. It is possible to synchronize changes.	Process diagram and process configuration are attached. Changes to either process diagram or process configuration affects the other.
18.	No vendor lock	Since BPEL 2.0 is not widespread, process configuration is limited to Intalio's BPMS
19.	Integration through web services	Integration through web services (Connectors for SAP)

Table 3: BPMS evaluation – Criteria 10-19

Once system analysts transformed the configured EPC diagram into a BPEL diagram, a link is established between these two diagrams. Thus, system analysts might navigate to and fro the two diagram types. However, system analysts need to keep track of process configuration inside the BPEL Designer. In theory, it is possible to merge changes done in both diagram types. In case, system analysts altered the BPEL process, it is possible to merge this alteration into the EPC diagram, and vice versa. For Intalio, both, process design and process configuration is done in one diagram. Changes in process design affects process configuration, and vice versa. Since no version control is available in these two BPMS, it is not desired that system analysts overwrite process design diagrams.

Since both BPMS use standards for process configuration, it is possible to use the process configuration on other execution platforms. Though, Oracle's BPEL Designer transforms BPEL version 1.1 into Java byte code for deploying the process, still, the BPEL code is available and is exported to other systems which are capable to process BPEL version 1.1.

Both BPMS integrate applications using web services. Intalio offers for SAP customers an *SAP connector* which wraps SAP applications, and make them accessible by web services, which then are able to be used in business processes.

Table 3 summarizes the findings.

	Oracle Process Manager	Intalio Process Server
20.	System Analyst Software Developer	System Analyst Software Developer
21.	Integrated into the BPEL Designer	Integrated into the BPMN Designer
22.	BPEL Designer and Process Manager	BPMN Designer and Process Server
23.	Parallel	Newer versions overwrite older versions

Table 4: BPMS evaluation – Criteria 20-23

7.3 Execution Level

Both BPMS allow system analysts to deploy configured processes to execution engines. However, software developers knowledge is needed in case process configuration validation fails. Deployment functionality is embedded in the process configuration tools Oracle BPEL Designer and Intalio's BPMN Designer. System analysts are in the position to deploy processes using just a single button. This supports the complexity reduction during the process deployment step.

In case of a process redeployment, the BPEL Designer in connection with Oracle's execution engine, allow version control. Thus, system analysts neither overwrite existing processes, nor do they delete running process instances. Rather a new version of a process will be deployed. It is possible to test and review the changes. The BPMN Designer in connection with Intalio's execution engine, overwrite former versions of the process. Running process instances are stopped by the engine.

Table 4 summarizes the findings.

8 Conclusion

The main advantage of a multi-vendor BPMS is that the best product for a specific step in the life cycle can be chosen. For example, the ARIS Business Architect is a convenient and mature tool to design business processes, which is Intalio's BPMN Designer not. The drawback for multi-vendor BPMS is that different tools must be integrated which are not intended to be used together. This leads to media-breaks which need to be dealed with. A methodology which addresses how to use all these different tools together needs to be developed. A multi-vendor BPMS should be considered in case some tools are already used within a company. For example if the ARIS toolset is already used to document business processes, then it makes no sense to switch to an integrated BPMS with a different process design tool.

The advantages of an integrated BPMS is that they are fast and easy to use. The integrated solution already contains a methodology how to use it. The main drawback for an integrated BPMS is that there might be limited functionality for certain steps in the life cycle compared to dedicated tools, such as Oracle's BPEL Designer. An integrated BPMS should be considered to be used in case there is no existing tool established within a company. Also, single projects within a company might only require a solution which fits to an integrated BPMS, without the hassle of multi-tool configuration.

Future work has to include better integration of different tools into a BPMS. Further adoption and improvement of standards, such as BPEL, and WSDL might tackle this issue. Tool providers must enhance tool functionality and better separate the roles in the life cycle. Moreover, process design notations must advance in the direction of BPMS, that is, that business processes are intended to be supported by webservices. Lastly, as business processes become executable and traceable by means of process portals, BPMS should permit the monitoring of important data and processing of this data. Business analysts might even model Key Performance Indicators (KPI) with a process design notation and get processed results for process instances on those indicators. This allows to identify bottlenecks and shortages in business processes.

Only after lots of ambitious efforts and their successful completion over the next years, BPMS will become easier to use during the entire life cycle of business processes.

References

[1] Howard Smith and Peter Fingar. *Business Process Management, the third wave.* Meghan-Kiffer Press, first edition edition, January 2003.

[2] Gregor Scheithauer, Guido Wirtz, and Candemir Toklu. Bridging the semantic gap between process documentation and process execution. In *The 2008 International Conference on Software Engineering and Knowledge Engineering (SEKE'08)*, 2008.

[3] Howard Smith. The Emergence of Business Process Management. *Information & Software Technology*, 45(15):1065–1069, December 2003.

[4] August-Wilhelm Scheer and Markus Nuettgens. Architecture and Reference Models for Business Process Management. *Lecture Notes in Computer Science*, 1806 / 2000:376–389, 2000.

[5] Stephen A. White. Specification: Business Process Modeling Notation Specification, February 2006.

[6] Chun Ouyang, van der Aalst, Marlon Dumas, ter Hofstede, and Arthur H. M. From business process models to process-oriented software systems: The BPMN to BPEL way, October 2006.

[7] Alexandre Alves, Assaf Arkin, Sid Askary, Charlton Barreto, Ben Bloch, Francisco Curbera, Mark Ford, Yaron Goland, Alejandro Guízar, Neelakantan Kartha, and Canyang Kevin Liu. Specification: Business Process Execution Language for Web Services version 2.0. Tech. rep. 2.0, OASIS, January 2007.

[8] Laury Verner. BPM the Promise and the Challenge. *ACM Queue: Tomorrow's Computing Today*, 2(1):82–91, March 2004.

[9] Jaelson Castro and Manuel Kolp. Towards Requirements-Driven Information Systems Engineering: The Tropos Project, January 2002.

[10] Arthur Ryman Sanjiva Weerawarana Roberto Chinnici, Jean-Jacques Moreau. Web services description language (wsdl) version 2.0 part 1: Core language. W3C Recommendation, 6 2007.

[11] Alexander Osterwalder, Christine Parent, and Yves Pigneur, editors. *Setting up an Ontology of Business Models.* Faculty of Computer Science and Information Technology, Riga Technical University, Riga, Latvia, 2004.

[12] R. Weber. Ontological foundations of information systems. *Coopers & Lybrand: Accounting Association of Australia and New Zealand, Melbourne*, 1997.

A List of previous University of Bamberg reports

Bamberger Beiträge zur Wirtschaftsinformatik

Stand April 28, 2008

Nr. 1 (1989) Augsburger W., Bartmann D., Sinz E.J.: Das Bamberger Modell: Der Diplom-Studiengang Wirtschaftsinformatik an der Universität Bamberg (Nachdruck Dez. 1990)

Nr. 2 (1990) Esswein W.: Definition, Implementierung und Einsatz einer kompatiblen Datenbankschnittstelle für PROLOG

Nr. 3 (1990) Augsburger W., Rieder H., Schwab J.: Endbenutzerorientierte Informationsgewinnung aus numerischen Daten am Beispiel von Unternehmenskennzahlen

Nr. 4 (1990) Ferstl O.K., Sinz E.J.: Objektmodellierung betrieblicher Informationsmodelle im Semantischen Objektmodell (SOM) (Nachdruck Nov. 1990)

Nr. 5 (1990) Ferstl O.K., Sinz E.J.: Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM)

Nr. 6 (1991) Augsburger W., Rieder H., Schwab J.: Systemtheoretische Repräsentation von Strukturen und Bewertungsfunktionen über zeitabhängigen betrieblichen numerischen Daten

Nr. 7 (1991) Augsburger W., Rieder H., Schwab J.: Wissensbasiertes, inhaltsorientiertes Retrieval statistischer Daten mit EISREVU / Ein Verarbeitungsmodell für eine modulare Bewertung von Kennzahlenwerten für den Endanwender

Nr. 8 (1991) Schwab J.: Ein computergestütztes Modellierungssystem zur Kennzahlenbewertung

Nr. 9 (1992) Gross H.-P.: Eine semantiktreue Transformation vom Entity-Relationship-Modell in das Strukturierte Entity-Relationship-Modell

Nr. 10 (1992) Sinz E.J.: Datenmodellierung im Strukturierten Entity-Relationship-Modell (SERM)

Nr. 11 (1992) Ferstl O.K., Sinz E. J.: Glossar zum Begriffsystem des Semantischen Objektmodells

Nr. 12 (1992) Sinz E. J., Popp K.M.: Zur Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der betrieblichen Diskurswelt

Nr. 13 (1992) Esswein W., Locarek H.: Objektorientierte Programmierung mit dem Objekt-Rollenmodell

Nr. 14 (1992) Esswein W.: Das Rollenmodell der Organsiation: Die Berücksichtigung aufbauorganisatorische Regelungen in Unternehmensmodellen

Nr. 15 (1992) Schwab H. J.: EISREVU-Modellierungssystem. Benutzerhandbuch

Nr. 16 (1992) Schwab K.: Die Implementierung eines relationalen DBMS nach dem Client/Server-Prinzip

Nr. 17 (1993) Schwab K.: Konzeption, Entwicklung und Implementierung eines computergestützten Bürovorgangssystems zur Modellierung von Vorgangsklassen und Abwicklung und Überwachung von Vorgängen. Dissertation

Nr. 18 (1993) Ferstl O.K., Sinz E.J.: Der Modellierungsansatz des Semantischen Objektmodells

Nr. 19 (1994) Ferstl O.K., Sinz E.J., Amberg M., Hagemann U., Malischewski C.: Tool-Based Business Process Modeling Using the SOM Approach

Nr. 20 (1994) Ferstl O.K., Sinz E.J.: From Business Process Modeling to the Specification of Distributed Business Application Systems - An Object-Oriented Approach -. 1st edition, June 1994

Ferstl O.K., Sinz E.J. : Multi-Layered Development of Business Process Models and Distributed Business Application Systems - An Object-Oriented Approach -. 2nd edition, November 1994

Nr. 21 (1994) Ferstl O.K., Sinz E.J.: Der Ansatz des Semantischen Objektmodells zur Modellierung von Geschäftsprozessen

Nr. 22 (1994) Augsburger W., Schwab K.: Using Formalism and Semi-Formal Constructs for Modeling Information Systems

Nr. 23 (1994) Ferstl O.K., Hagemann U.: Simulation hierarischer objekt- und transaktionsorientierter Modelle

Nr. 24 (1994) Sinz E.J.: Das Informationssystem der Universität als Instrument zur zielgerichteten Lenkung von Universitätsprozessen

Nr. 25 (1994) Wittke M., Mekinic, G.: Kooperierende Informationsräume. Ein Ansatz für verteilte Führungsinformationssysteme

Nr. 26 (1995) Ferstl O.K., Sinz E.J.: Re-Engineering von Geschäftsprozessen auf der Grundlage des SOM-Ansatzes

Nr. 27 (1995) Ferstl, O.K., Mannmeusel, Th.: Dezentrale Produktionslenkung. Erscheint in CIM-Management 3/1995

Nr. 28 (1995) Ludwig, H., Schwab, K.: Integrating cooperation systems: an event-based approach

Nr. 30 (1995) Augsburger W., Ludwig H., Schwab K.: Koordinationsmethoden und -werkzeuge bei der computergestützten kooperativen Arbeit

Nr. 31 (1995) Ferstl O.K., Mannmeusel T.: Gestaltung industrieller Geschäftsprozesse

Nr. 32 (1995) Gunzenhäuser R., Duske A., Ferstl O.K., Ludwig H., Mekinic G., Rieder H., Schwab H.-J., Schwab K., Sinz E.J., Wittke M: Festschrift zum 60. Geburtstag von Walter Augsburger

Nr. 33 (1995) Sinz, E.J.: Kann das Geschäftsprozeßmodell der Unternehmung das unternehmensweite Datenschema ablösen?

Nr. 34 (1995) Sinz E.J.: Ansätze zur fachlichen Modellierung betrieblicher Informationssysteme - Entwicklung, aktueller Stand und Trends -

Nr. 35 (1995) Sinz E.J.: Serviceorientierung der Hochschulverwaltung und ihre Unterstützung durch workflow-orientierte Anwendungssysteme

Nr. 36 (1996) Ferstl O.K., Sinz, E.J., Amberg M.: Stichwörter zum Fachgebiet Wirtschaftsinformatik. Erscheint in: Broy M., Spaniol O. (Hrsg.): Lexikon Informatik und Kommunikationstechnik, 2. Auflage, VDI-Verlag, Düsseldorf 1996

Nr. 37 (1996) Ferstl O.K., Sinz E.J.: Flexible Organizations Through Object-oriented and Transaction-oriented Information Systems, July 1996

Nr. 38 (1996) Ferstl O.K., Schäfer R.: Eine Lernumgebung für die betriebliche Aus- und Weiterbildung on demand, Juli 1996

Nr. 39 (1996) Hazebrouck J.-P.: Einsatzpotentiale von Fuzzy-Logic im Strategischen Management dargestellt an Fuzzy-System-Konzepten für Portfolio-Ansätze

Nr. 40 (1997) Sinz E.J.: Architektur betrieblicher Informationssysteme. In: Rechenberg P., Pomberger G. (Hrsg.): Handbuch der Informatik, Hanser-Verlag, München 1997

Nr. 41 (1997) Sinz E.J.: Analyse und Gestaltung universitärer Geschäftsprozesse und Anwendungssysteme. Angenommen für: Informatik '97. Informatik als Innovationsmotor. 27. Jahrestagung der Gesellschaft für Informatik, Aachen 24.-26.9.1997

Nr. 42 (1997) Ferstl O.K., Sinz E.J., Hammel C., Schlitt M., Wolf S.: Application Objects – fachliche Bausteine für die Entwicklung komponentenbasierter Anwendungssysteme. Angenommen für: HMD – Theorie und Praxis der Wirtschaftsinformatik. Schwerpunkheft ComponentWare, 1997

Nr. 43 (1997): Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using the Semantic Object Model (SOM) – A Methodological Framework - . Accepted for: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1997

Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using (SOM), 2nd Edition. Appears in: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1998

Nr. 44 (1997) Ferstl O.K., Schmitz K.: Zur Nutzung von Hypertextkonzepten in Lernumgebungen. In: Conradi H., Kreutz R., Spitzer K. (Hrsg.): CBT in der Medizin – Methoden, Techniken, Anwendungen -. Proceedings zum Workshop in Aachen 6. – 7. Juni 1997. 1. Auflage Aachen: Verlag der Augustinus Buchhandlung

Nr. 45 (1998) Ferstl O.K.: Datenkommunikation. In. Schulte Ch. (Hrsg.): Lexikon der Logistik, Oldenbourg-Verlag, München 1998

Nr. 46 (1998) Sinz E.J.: Prozeßgestaltung und Prozeßunterstützung im Prüfungswesen. Erschienen in: Proceedings Workshop „Informationssysteme für das Hochschulmanagement“. Aachen, September 1997

Nr. 47 (1998) Sinz, E.J.:, Wismans B.: Das „Elektronische Prüfungsamt“. Erscheint in: Wirtschaftswissenschaftliches Studium WiSt, 1998

Nr. 48 (1998) Haase, O., Henrich, A.: A Hybrid Respresentation of Vague Collections for Distributed Object Management Systems. Erscheint in: IEEE Transactions on Knowledge and Data Engineering

Nr. 49 (1998) Henrich, A.: Applying Document Retrieval Techniques in Software Engineering Environments. In: Proc. International Conference on Database and Expert Systems

Applications. (DEXA 98), Vienna, Austria, Aug. 98, pp. 240-249, Springer, Lecture Notes in Computer Sciences, No. 1460

Nr. 50 (1999) Henrich, A., Jamin, S.: On the Optimization of Queries containing Regular Path Expressions. Erscheint in: Proceedings of the Fourth Workshop on Next Generation Information Technologies and Systems (NGITS'99), Zikhron-Yaakov, Israel, July, 1999 (Springer, Lecture Notes)

Nr. 51 (1999) Haase O., Henrich, A.: A Closed Approach to Vague Collections in Partly Inaccessible Distributed Databases. Erscheint in: Proceedings of the Third East-European Conference on Advances in Databases and Information Systems – ADBIS'99, Maribor, Slovenia, September 1999 (Springer, Lecture Notes in Computer Science)

Nr. 52 (1999) Sinz E.J., Böhnlein M., Ulbrich-vom Ende A.: Konzeption eines Data Warehouse-Systems für Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule“ im Rahmen der 29. Jahrestagung der Gesellschaft für Informatik, Paderborn, 6. Oktober 1999

Nr. 53 (1999) Sinz E.J.: Konstruktion von Informationssystemen. Der Beitrag wurde in geringfügig modifizierter Fassung angenommen für: Rechenberg P., Pomberger G. (Hrsg.): Informatik-Handbuch. 2., aktualisierte und erweiterte Auflage, Hanser, München 1999

Nr. 54 (1999) Herda N., Janson A., Reif M., Schindler T., Augsburger W.: Entwicklung des Intranets SPICE: Erfahrungsbericht einer Praxiskooperation.

Nr. 55 (2000) Böhnlein M., Ulbrich-vom Ende A.: Grundlagen des Data Warehousing. Modellierung und Architektur

Nr. 56 (2000) Freitag B, Sinz E.J., Wismans B.: Die informationstechnische Infrastruktur der Virtuellen Hochschule Bayern (vhb). Angenommen für Workshop "Unternehmen Hochschule 2000" im Rahmen der Jahrestagung der Gesellschaft f. Informatik, Berlin 19. - 22. September 2000

Nr. 57 (2000) Böhnlein M., Ulbrich-vom Ende A.: Developing Data Warehouse Structures from Business Process Models.

Nr. 58 (2000) Knobloch B.: Der Data-Mining-Ansatz zur Analyse betriebswirtschaftlicher Daten.

Nr. 59 (2001) Sinz E.J., Böhnlein M., Plaha M., Ulbrich-vom Ende A.: Architekturkonzept eines verteilten Data-Warehouse-Systems für das Hochschulwesen. Angenommen für: WI-IF 2001, Augsburg, 19.-21. September 2001

Nr. 60 (2001) Sinz E.J., Wismans B.: Anforderungen an die IV-Infrastruktur von Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule 2001“ im Rahmen der Jahrestagung der Gesellschaft für Informatik, Wien 25. – 28. September 2001

Änderung des Titels der Schriftenreihe *Bamberger Beiträge zur Wirtschaftsinformatik* in *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik* ab Nr. 61

Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik

Nr. 61 (2002) Goré R., Mendler M., de Paiva V. (Hrsg.): Proceedings of the International Workshop on Intuitionistic Modal Logic and Applications (IMLA 2002), Copenhagen, July 2002.

Nr. 62 (2002) Sinz E.J., Plaha M., Ulbrich-vom Ende A.: Datenschutz und Datensicherheit in einem landesweiten Data-Warehouse-System für das Hochschulwesen. Erscheint in: Beiträge zur Hochschulforschung, Heft 4-2002, Bayerisches Staatsinstitut für Hochschulforschung und Hochschulplanung, München 2002

Nr. 63 (2005) Aguado, J., Mendler, M.: Constructive Semantics for Instantaneous Reactions

Nr. 64 (2005) Ferstl, O.K.: Lebenslanges Lernen und virtuelle Lehre: globale und lokale Verbesserungspotenziale. Erschienen in: Kerres, Michael; Keil-Slawik, Reinhard (Hrsg.); Hochschulen im digitalen Zeitalter: Innovationspotenziale und Strukturwandel, S. 247 – 263; Reihe education quality forum, herausgegeben durch das Centrum für eCompetence in Hochschulen NRW, Band 2, Münster/New York/München/Berlin: Waxmann 2005

Nr. 65 (2006) Schönberger, Andreas: Modelling and Validating Business Collaborations: A Case Study on RosettaNet

Nr. 66 (2006) Markus Dorsch, Martin Grote, Knut Hildebrandt, Maximilian Röglinger, Matthias Sehr, Christian Wilms, Karsten Loesing, and Guido Wirtz: Concealing Presence Information in Instant Messaging Systems, April 2006

Nr. 67 (2006) Marco Fischer, Andreas Grünert, Sebastian Hudert, Stefan König, Kira Lenskaya, Gregor Scheithauer, Sven Kaffille, and Guido Wirtz: Decentralized Reputation Management for Cooperating Software Agents in Open Multi-Agent Systems, April 2006

Nr. 68 (2006) Michael Mendler, Thomas R. Shiple, Gérard Berry: Constructive Circuits and the Exactness of Ternary Simulation

Nr. 69 (2007) Sebastian Hudert: A Proposal for a Web Services Agreement Negotiation Protocol Framework to be announced

Nr. 70 (2007) Thomas Meins: Integration eines allgemeinen Service-Centers für PC-und Medientechnik an der Universität Bamberg – Analyse und Realisierungs-Szenarien to be announced

Nr. 71 (2007) Andreas Grünert: Life-cycle assistance capabilities of cooperating Software Agents for Virtual Enterprises to be announced

Nr. 72 (2007) Michael Mendler, Gerald Lüttgen: Is Observational Congruence on µ-Expressions Axiomatisable in Equational Horn Logic?

Nr. 73 (2007) Martin Schissler: to be announced

Nr. 74 (2007) Sven Kaffille, Karsten Loesing: Open chord version 1.0.4 User's Manual

Nr. 75 (2008) Karsten Loesing (Hrsg.): Extended Abstracts of the Second Privacy Enhancing Technologies Convention (PET-CON 2008.1)