

Case Study: Applying Business Process Management Systems

Gregor Scheithauer* and Guido Wirtz

Distributed and Mobile Systems Group, University of Bamberg

Feldkirchenstraße 21, 96052 Bamberg, Germany

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

Abstract

Business Process Management Systems aim to support the Business Process Management paradigm and to ease legacy application integration. However, do such systems really meet real-world requirements? This paper introduces and discusses a set of criteria which are important for business process management systems and applies these criteria in comparing tools from three important vendors, namely IDS Scheer, Oracle and Intalio based on a real-world case study.

Keywords: BPMS, usability criteria, Webservices

1. Introduction

Business Process Management Systems (BPMS) [7] are sets of tools to support the Business Process Management (BPM) life-cycle [5] that are either offered by one vendor, or multiple vendors offer parts of a BPMS. Smith [6] sees a list of key advantages in using a modern BPMS: it bridges heterogenous 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 [7] 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, automatization, and collaboration.

Table 1 depicts which BPMS tools support what step in the BPM life cycle. Business process design includes process documentation with a process notation, such as Event-driven Process Chain (EPC) [3] notation and Business Process Modeling Notation (BPMN) [9]. Configuration includes the transformation [2] from process models into formal languages such as the Business Process Execution Language (BPEL) [1]. 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. Based on a detailed list of evaluation criteria covering all steps relevant for BPMS (section 2), a real-life scenario is used to evaluate 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* (section 3). The results are summarized in table 2. A more detailed description of the scenario and the case study implementation as well as a more in-depth discussion of the results is documented in [4].

2. 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 [5]: 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.

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 [8]). Business

*The first author is funded by means of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ07012.

Table 1. BPM tools used in the process life cycle

		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	

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.
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 middle-ware 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 middle-ware 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.
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.

These criteria are used to evaluate the different tools when realizing the example process that is introduced next.

3. Case Study and Results

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 1, 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 needed: (1) *PriceService*, and (2) *QuantityService*. Both services are available as web services and provide a WSDL 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. The results of applying our criteria when implementing GSIS using two different BPMS are summarized in table 2; for a detailed discussion refer to [4].

4. Outlook

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 web services. 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] A. Alves, A. Arkin, S. Askary, C. Barreto, B. Bloch, F. Curbera, M. Ford, Y. Goland, A. Guzar, N. Kartha, and C. K. Liu. Specification: Business Process Execution Language for Web Services version 2.0. Tech. rep. 2.0, OASIS, January 2007.
- [2] C. Ouyang, van der Aalst, M. Dumas, ter Hofstede, and A. H. M. From business process models to process-oriented software systems: The BPMN to BPEL way, October 2006.
- [3] A.-W. Scheer and M. Nuettgens. Architecture and Reference Models for Business Process Management. *Lecture Notes in Computer Science*, 1806 / 2000:376–389, 2000.
- [4] G. Scheithauer and G. Wirtz. Applying Business Process Management Systems – a Case Study. *Bamberger Beiträge zur Wirtschaftsinformatik 76*, Bamberg University, May 2008. ISSN 0937-3349.
- [5] G. Scheithauer, G. Wirtz, and C. 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.
- [6] H. Smith. The Emergence of Business Process Management. *Information & Software Technology*, 45(15):1065–1069, December 2003.
- [7] H. Smith and P. Fingar. *Business Process Management, the third wave*. Meghan-Kiffer Press, first edition edition, January 2003.
- [8] R. Weber. Ontological foundations of information systems. *Coopers & Lybrand: Accounting Association of Australia and New Zealand, Melbourne*, 1997.
- [9] S. A. White. Specification: Business Process Modeling Notation Specification, February 2006.

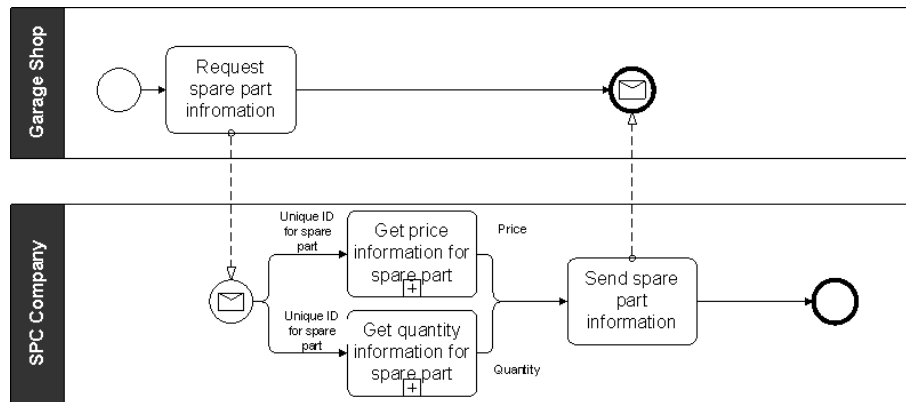


Figure 1. Case Study Process

Table 2. Evaluation: 1-9 – Business Level / 10-19 – Integration Level / 20-23 – Execution Level

ARIS Business Architect

1. Process owner, business analyst, process participant
2. EPC; not a standard, was published in 1992, XML based, thus open.
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.
4. Complete
5. Possible
6. Architecture Integrated Information Systems (ARIS)
7. Full support
8. Global or local diagram repository
9. No version control

ARIS SOA Architect, Oracle BPEL Designer

10. System analyst, software developers
11. EPC → BPEL 1.1 Poorer to richer semantic translation
12. BPEL 1.1, Specification from IBM & Microsoft
13. Accepted language, great tool support, missing concepts, such as human people integration
14. Complete
15. Configuration with two major tools necessary
16. Global or 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.
18. No vendor lock
19. Integration through web services

Oracle Process Manager

20. System Analyst Software Developer
21. Integrated into the BPEL Designer
22. BPEL Designer and Process Manager
23. Parallel

Intalio BPMN Designer

- Business analyst
 BPMN is an OMG standard, may replace UML activity diagram, only standard for graphical elements, not easy exchanged.
 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.
 BPMN in general.
 Possible with limitations
 Specification with instructions how to use the notation
 Limited support
 No repository
 No version control

Intalio BPMN Designer

- Business analyst, system analyst, software developer
 BPMN → BPEL 2.0 Different semantics
 BPEL 2.0, Specification from OASIS
 BPEL 2.0 Specification not yet adopted, no other tool support
 Complete
 Configuration with one tool
 Local service repository
 Process diagram and process configuration are attached. Changes to either process diagram or process configuration affects the other.
 Since BPEL 2.0 is not widespread, process configuration is limited to Intalio's BPMS
 Integration through web services (Connectors for SAP)

Intalio Process Server

- System Analyst Software Developer
 Integrated into the BPMN Designer
 BPMN Designer and Process Server
 Newer versions overwrite older versions