Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 82, Issue 14, March/April 2018, Pages 38–53
https://doi.org/10.7250/csimq.2018-14.03

Effort Estimation in BPMS Migration

Christopher Drews and Birger Lantow*

University of Rostock, Chair of Business Information Systems,
18051 Rostock, Germany

{win.office, birger.lantow}@uni-rostock.de

Abstract. Usually Business Process Management Systems (BPMS) are highly integrated in the IT of organizations and are at the core of their business. Thus, migrating from one BPMS solution to another is not a common task. However, there are forces that are pushing organizations to perform this step, e.g. maintenance costs of legacy BPMS or the need for additional functionality. Before the actual migration, the risk and the effort must be evaluated. This work provides a framework for effort estimation regarding the technical aspects of BPMS migration. The framework provides questions for BPMS comparison and an effort evaluation schema. The applicability of the framework is evaluated based on a simplified BPMS migration scenario.

Keywords: BPMS, Migration, Effort estimation, Workflow management, BPMS architecture.

1 Introduction

Workflow Management Systems (WfMS) and their successors – Business Process Management Systems (BPMS) – have their origins in the last decades of the 20th century. In contrast to traditional document or data centric ERP Systems, their main focus is the control of Business Processes. Thus, they are more flexible when enterprises develop new ways of doing their business. However, like ERP Systems, they are at the core of the enterprise and it is hard to replace them. Furthermore, they are deeply integrated with other systems of the enterprise [1]. Advances in Service Oriented Architectures, Cloud Computing, and Continuous Delivery have created a demand to replace legacy BPMS for reducing their maintenance efforts. As a consequence, a BPMS migration has to be considered.

The aim of this work is to examine ways to predict the effort of product migrations from one BPMS to another. The respective literature offers a wide range of information about Business Process Management, its value and critical success factors. Compared to this, publications towards the systems which implement the key element of BPM, the process automation, are rare.

Another important aspect is the availability of a methodology for effort estimation. A broad range of approaches regarding effort estimation for software engineering has been developed.

* Corresponding author

Reference: C. Drews and B.Lantow, “Effort Estimation in BPMS Migration,” Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 14, pp. 38–53, 2018. Available: https://doi.org/10.7250/csimq.2018-14.03

Additional information. Author’s ORCID iD: B. Lantow – orcid.org/0000-0003-0800-7939. PII S225599221800082X. Received: 12 February 2018. Accepted: 6 April 2018. Available online: 30 April 2018.

However, a BPMS migration is a different endeavour. Yet, general effort estimation approaches can be adapted to specific areas. This is shown by Simperl et al. in [2] by their approach to cost estimation for ontology engineering (ONTOCOM). They identified five general cost estimation approaches:

1. **Analogy Method.** This approach extrapolates available data from similar projects. It depends on empirical data from previous projects and the accurate determination of similarities and differences of the projects.
2. **Bottom-Up Method.** The effort of project components is estimated separately and summed up to a project effort. This approach depends on an existing component structure and is likely to produce more accurate results than the other approaches.
3. **Top-Down Method.** Here, general project parameters are used as a base for effort estimation. A coarse partition of the project is used to address, e.g. the characteristics of different project phases. Using global cost estimators, the estimation is less accurate.
4. **Expert Judgment/Delphi Method.** This approach uses a structured process to collect knowledge from a group of experts.
5. **Parametric/Algorithmic Method.** Mathematical equations based on research and previous project data are used for effort estimation. The method identifies and applies main cost drivers of specific projects.

Some of these approaches can be combined. For instance, for ONTOCOM, a top-down approach has been selected and parametric methods are used. Furthermore, an expert-based method was applied. For the effort estimation of BPMS migration, a top-down approach could be too coarse. In contrast to ontology engineering – where ontology size is a good estimator – BPMS migration involves a lot of heterogeneous artifacts such as different technical components and models (see Section 2). Thus, heterogeneous metrics have to be applied for cost estimation; and cost drivers, as well as their influence on the actual effort, vary. The idea is to provide means for a bottom-up effort estimation by identifying component types of a BPMS and the cost drivers relevant for them with respect to BPMS migration. This sets a base for a fine-grained use of parametric methods. Having information regarding the several BPMS components, as a result of this work, could also help experts to make better estimations.

A systematic literature review is used in Section 2 to identify the available approaches to analyse and compare BPM systems based on their components. The gathered approaches will be evaluated regarding their use in effort estimation of BPM system migration projects in Section 3. The aim of the article is to provide or framework that supports the decision process while planning a BPMS migration project and assessing BPMS alternatives.

The scope of the article is limited to the evaluation of the effort of migration projects. Thus, the process of finding a suitable BPMS for the specific needs of a company is not the subject of matter. The assumption in this work is that the examined BPMS suits the requirements of a respective run-time scenario. Section 4 demonstrates the use of the constructed framework in a particular BPMS migration scenario. Section 5 discusses the validation of the proposed approach and concludes the article. The article extends the authors work presented at the International Conference on Perspectives of Business Informatics Research (BIR 2017) [3].

2 Systematic Literature Analysis

Before [3], no work has been found that is directly dealing with the effort estimation regarding the migration of BPMS or WfMS. However, it is assumed that having a reference for BPMS components could help to identify necessary tasks for the migration and thus to be able to find appropriate work packages of a migration project. Each of these work packages can then be a subject to effort estimation. The complexity of estimation could be reduced by the divide-and-conquer principle. Since the technical aspects of migration are in focus, reference architectures for BPMS seem to be a good starting point.

2.1 Analysis Process

A literature analysis has been conducted in order to find approaches for BPMS comparison. The systematic Literature Analysis (SLA) approach by Kitchenham [4] provided guidelines for this research. A first step according to these guidelines is the selection of representative publication sources for the area of interest. The following sources were selected for the SLA:

- The International Conference on Business Process Management (BPM) that calls itself *"the premier conference for researchers and practitioners in the field of Business Process Management."* [5]
- The Business Process Management Journal (BPMJ) that is published by Emerald Group Publishing since 1995 and aims to *"[...] examine how a variety of business processes intrinsic to organizational efficiency and effectiveness are integrated and managed for competitive success."* [6]
- Lustratus Research that describes itself as *"[...] a leading infrastructure software market analyst and consultancy firm"* [7]. Lustratus Research provides different kinds of technical reports via their online store at www.lustratus.com.

While BPM Conference and BPM Journal are seen as sources for scientific research in the area, Lustratus Research is seen as a more industry oriented source, in contrast.

Overall the literature basis of this SLA consisted of around 1000 papers derived from these sources. The next steps according to Kitchenham are population, intervention, and union followed by the paper selection. The initial population was done by using the following search term:

> *"Business Process Management Systems" OR "BPMS" OR "BPM suites" OR "BPM solution" OR "BPM offering" OR "BPM product" OR "WfMS" OR "Workflow Management System"*

The search string was applied to the titles as well as to the abstracts and keywords of the papers. For the scientific sources, the Scopus search was used. The pool of documents at Lustratus was searched manually. 76 relevant papers have been identified. The intervention step is used in order to narrow down the selection of papers further on. Here, the focus on BPMS comparison was added in the search. The following search term was used:

> *"compare" OR "comparing" OR "comparison" OR "architecture"*

For the reasons given in the introduction of Section 2, "architecture" was chosen as a part of the search term. The union step formed an intersection of the sets resulting from population and intervention. After this step 14 papers remained (see Table 1).

Table 1. Numbers of selected papers in SLA process

Step	BPM Conf	BPMJ	Lustratus	Total
Population	53	9	5	**67**
Intervention+Union	5	5	4	**14**

Afterwards, the data extraction and a further selection of sources has been performed by reading the papers. The selection criterion was the description of general BPMS components or a general BPMS architecture. At the same time, the respective data has been extracted.

Out of the five sources selected from the BPM Conference, three were only of marginal relevance. Kannengießer et al. [8] discussed the construction of a special adapter connecting a BPMS to Open Platform Communications Unified Architecture (OPC UA) components. Reijers et al. [9] aimed at the specific problem of task assignment in WfMS. Rulle and Siegeries [10] provided an approach to Business Process Modeling.

Two other papers gave more information on BPMS components and architecture. Ferme and Pautasso presented in [11] a simple architectural model of a WfMS. It covered (1) a Workflow Engine that executes the processes, (2) Web Services to connect to domain specific functionality, and (3) a storage system. Al-Ghaiati provides a more comprehensive view on BPMS architecture in [12]. His more relevant BPMS components we represented as following: (1) Extended Directory Application (EDA) which implements authorisation, authentication, and an organizational model, (2) Enterprise Messaging Services (EMS) and Enterprise Messaging Exchange Engine (EMEEE) which provide external and internal integration functionalities, (3) Enterprise Knowledge Layer Assemblers (KLA) which abstract the storage system and the respective data model, (4) Workflow Engine that executes the processes, and (5) configuration and management tools. However, Al-Ghaiati did not explain all elements of the proposed architecture and a coherent level of abstraction was not used in the paper. Some architecture elements remained as general functionalities or repositories and others were specific BPMS component implementations. Thus, the paper provided a particular input for deriving an implementation independent view of BPMS architecture, but it did not let to see a unified representation of BPMS components.

Two of the five sources from BMJ were considered irrelevant. Cheung and Hidders [13] just discussed Business Process Model transformation to Business Process Analysis Models. Janiesch et al. [14] focused on message handling. Both papers treated BPMS systems as black boxes.

Ferriera and Ferreira [15] described the following components of a BPMS: (1) Workflow Engine that executes the processes, (2) Integration Infrastructure, (3) Modeling Tools, and (4) Administration and Monitoring Tools. A combination of an Agent Based System and a Workflow Management System was discussed by Verginades and Mentzas in [16]. Besides the components specific for agent based systems, the following BPMS components were identified: (1) Information Management which abstracts from storage systems and the respective data models, (2) Communication which implements the integration functionality, (3) System Management, (4) Workflow Modeling, (5) Workflow Engine that executes the processes, (6) User Interface, and (7) Access Control which implements authorization, authentication and the organizational model.

The remaining BPMJ paper by Shaw et al. [17] provided a pyramid architecture for BPMS. The purpose of this pyramid architecture was providing a frame for BPMS analysis. This was in alignment with our intention to find a general tool for BPMS comparison which is needed for migration effort estimation. Thus, the architecture by Shaw et al. is discussed in more detail in Section 2.2.

The four papers originating from Lustratus Research [18], [19], [20], [21] turned out to be applications and refinements of the ‘frame of reference’ for BPMS comparison by the author of these papers – Steve Craggs. This approach also fitted well to the intention of our research and is discussed in detail in Section 2.3.

2.2 BPMS Pyramid Architecture

The approach by Shaw et al. [17] is visualized as a pyramid of blocks with two legs. The blocks represent BPMS core technologies while the two legs emphasize the independence between formal model constructs and software applications. The arrangement of the blocks symbolizes relationships between the elements. A core technology on a certain level is a prerequisite for the technologies on the next higher level. Within the horizontal dimension there are no interdependencies between the blocks. On top of the pyramid, the *BPMS* itself is constructed on the base of all other blocks. One layer below lies the *Enactable Business Model* as the core of a BPMS. Five types of models can be distinguished here: static, dynamic, passive, active and enactable [22, pp. 38–44]. Shaw et al. define an enactable model as a “*[...] composition of model constructs that is derived from the properties of the physical, hardware or software modeling medium that together naturally display characteristics that exactly replicate those of the subject abstraction”* [17, p. 5]. Thus the model controls and reflects the current process states (the process is seen as a *subject of abstraction*).

Further on, the leg of model constructs contains the following blocks from top to bottom (see Figure 1): (1) *Formal Model Constructs*, (2) *Formal Modeling Notation* and *Ontological Modeling Grammar*, (3) *Model Abstraction*, and (4) *Subject Modeled* (Process).

The software application leg contains the blocks from top to bottom: (1) *Software Application*, (2) *Software Language*, (3) *Software Notation* and *Software Grammar*, (4) *Software Formalism*, and spanning several layers (5) *Technical Infrastructure*.

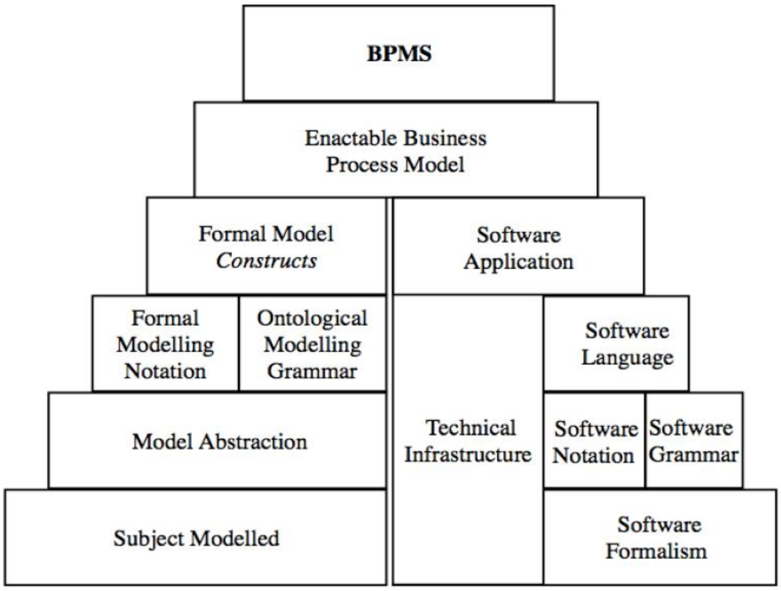
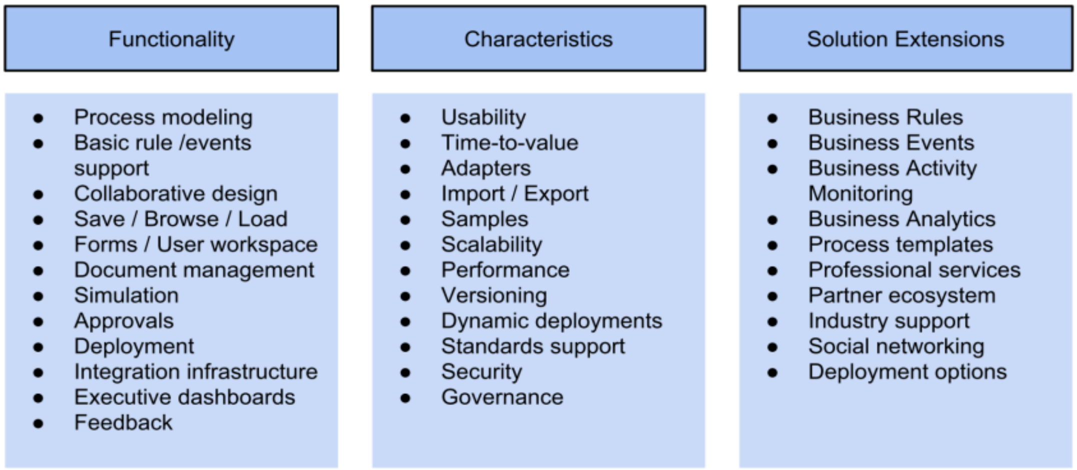


Figure 1. BPMS Pyramid Architecture [17]

While in the application leg of the pyramid architecture some of the blocks seem to be appropriate for BPMS comparison regarding BPMS migration, the architecture as whole is still too coarse grained for this purpose.

2.3 Frame of Reference by Lustratus

As mentioned in Section 2.1, this frame of reference is applied by Craggs in all four reports ([18], [19], [20], [21]) originating from Lustratus Research between 2009 and 2012. Craggs divides the frame of reference in three main areas (see Figure 2): (1) Functionality, (2) Characteristics, and (3) Solution extensions.



Functionality

- Process modeling
- Basic rule /events support
- Collaborative design
- Save / Browse / Load
- Forms / User workspace
- Document management
- Simulation
- Approvals
- Deployment
- Integration infrastructure
- Executive dashboards
- Feedback

Characteristics

- Usability
- Time-to-value
- Adapters
- Import / Export
- Samples
- Scalability
- Performance
- Versioning
- Dynamic deployments
- Standards support
- Security
- Governance

Solution Extensions

- Business Rules
- Business Events
- Business Activity Monitoring
- Business Analytics
- Process templates
- Professional services
- Partner ecosystem
- Industry support
- Social networking
- Deployment options

Figure 2. Frame of reference by Lustratus [21]

Each area summarizes a set of key categories. The *Functionality* area, contains the BPMS functionalities that are considered to be “standard”. Thus, these describe the basic BPMS functionalities. It includes the core functionalities of Process Modeling, Execution, and Monitoring. The *Characteristics* area aims at traditional software quality measures, representing non-functional requirements. However, there are also functionalities included. For instance, Import/Export, Versioning, and Governance provide means for a higher flexibility and efficiency in basic BPMS functionalities. An indirect influence on software quality is assumed here. *Solution Extensions* provide additional functionalities that are not considered as basic functionalities of a BPMS.

This framework of reference is more detailed than the pyramid architecture by Shaw et al. [17], but some of the items, especially in the characteristics area, do not fit to the purpose of migration effort estimation. They are system inherent characteristics that should be evaluated during system selection before planning the technical implementation of a BPMS migration. Additionally, the mapping of the items to the areas changes over time as well as some of the items themselves.

3 Framework Construction

The discussion of the found literature sources and, especially, the pyramid architecture [17] and the frame of reference by Lustratus [21] provide general ideas on how to compare BPMS. The ideas from Shaw et al. [17] and Craggs [21], taken together, subsume the architecture components of the other approaches presented in Section 2. An important aspect that can be derived for BPMS migration is that not only technical components that provide functionality have to be considered, but also existing models that describe the processes and their context should be taken into account.

However, the presented approaches do not directly fit to the task of migration effort estimation. For instance, the runtime environment of the BPMS components is not considered. More important, cost drivers in terms of Boehms work on CoCoMo [23] need to be derived in order to assess the characteristics/quality of source and target BPMS for effort estimation. Craggs [21] covers some of the relevant quality aspects, but his work has a different focus and does not contain migration specific quality attributes. Therefore, this section introduces a framework for BPMS comparison that is based on the results of the analysis presented in the previous sections. The framework is shown in Figure 3.

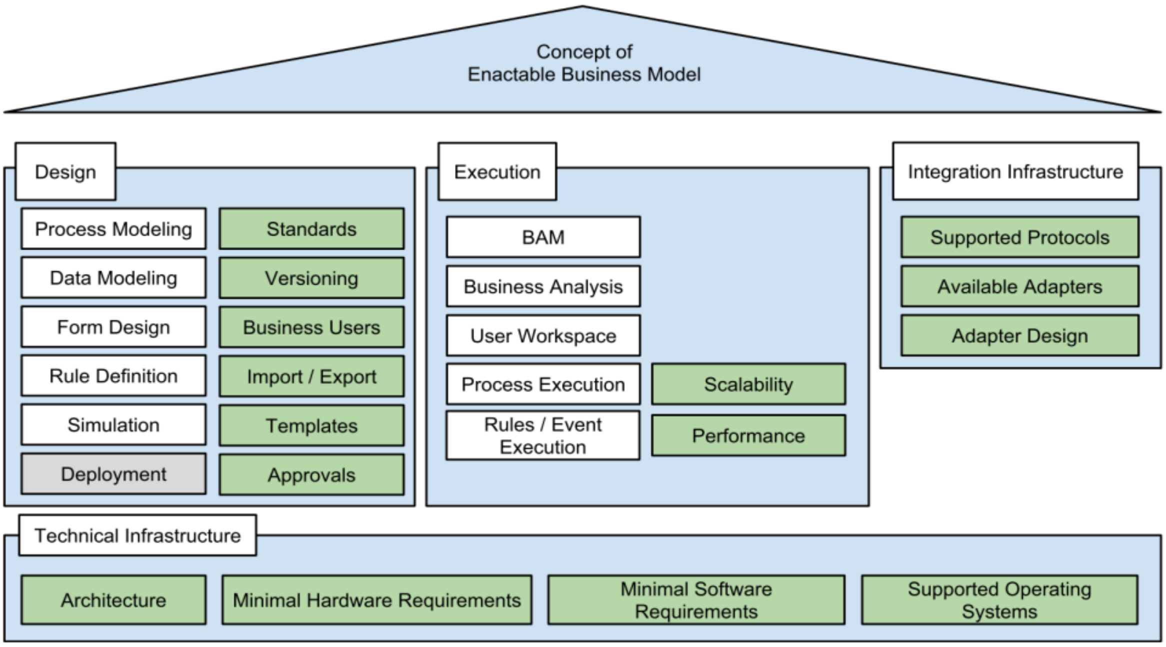


Figure 3. Proposed framework for BPMS comparison (available also in [3])

3.1 View on BPMS

The framework is mainly inspired by Lustratus' frame of reference [21], but it is based on a new structure with a different view on BPMS. The general idea is to divide all functionalities of BPMS in four logical components. In this context logical means that the individual characteristics are considered independently from the architecture of the tools that implement them. This allows analyzing BPMS in more detail without increasing the complexity of the result by including irrelevant aspects of its structure. Figure 3 illustrates the four components, namely, *Design, Execution, Integration Infrastructure,* and *Technical Infrastructure*. These components divide BPMS into three main functional areas that have been derived from the investigation of existing systems like TIBCO[†] Active matrix. *Design* tools and activities can be examined mostly independently from *Execution* regarding the methodology and architecture. *Integration Infrastructure* has been seen as an important separate component. Since processes run across functional areas, integration of specialized function oriented information systems is the main task of the BPMS. The *Technical Infrastructure* represents the common system requirements for running a certain BPMS.

In the proposed framework all components are seen under the umbrella of the *Concept of Enactable Business Model*. Here, an idea of the pyramid architecture has been adopted. The general question – What concepts are included in the Business Model and what are not? – influences the complete BPMS functionality. For instance, TIBCO ActiveMatrix, in comparison to TIBCO iProcess, adds organizational and integration related concepts to the model. However, not all available concepts need always to be used in the source model.

The white boxes within the components of Figure 3 represent key functionalities, while relevant characteristics are displayed using the green boxes. The idea behind this is to derive activities for migration from the functionalities and to derive cost drivers in terms of Boehms work on CoCoMo [23] from the characteristics. The Technical Infrastructure is seen as a special case of a cost driver. The provision of the necessary infrastructure for a BPMS covers activities that are independent from the special tasks of BPMS implementation and migration. Furthermore, a majority of the costs stems from software licenses, hardware components, etc.

Applying these general ideas, for instance, to the *Design* component of the framework, there are five functionalities considered relevant for migration effort: (1) Process Modeling, (2) Data Modeling, (3) Form Design, (4) Rule Definition, (5) Simulation, which all result in the creation of the respective design artifacts. The characteristics in green boxes can be applied to all these functionalities. Deployment is not considered because most of the characteristics are not applicable. The characteristics are directly inspired by Cragg's frame of reference [21]. Redundant aspects, such as collaborative design and time-to-value, are removed. Collaborative design is by definition based on versioning and the support of business users. Business users (e.g. business analysts) are called non-technical users in here. Time-to-value is indirectly influenced by all characteristics; and all green boxes in the Design component affect time-to-value.

Since this BPMS comparison framework is based on the evaluation of only two approaches, a new version may follow. The general structure presented in this section allows flexibility regarding future additions and changes in functionalities and characteristics.

3.2 General Questions of BPMS Migration Effort Estimation

Based on the given framework for BPMS comparison, specific questions have been formulated in order to provide a guide for migration planning and effort estimation (see Table 2). A label is assigned to each question for further identification purposes. In order to avoid complex answers, the expected answer types for all questions are given in the third column of the table.

[†] https://www.tibco.com/

All questions having *QD* in their labels, deal with the *Design* component. While $QD9_{\text{Form}}$ deals with the forms functionality only, the other eight questions are applicable to all the relevant design time functionalities depicted in Figure 3. The first four questions refer to the technical support of design artifact transfer between start and target systems. As a result of answering these questions, one of the following three action scenarios (AS) can be identified: artifact transfer using export and import functionalities (*AS*1, *QD*1 *true*); artifact transfer possible using an adapter technology (*AS*2, *QD*1 *false and QD*2–4 *true*), artifact transfer requires manual recreation of the artifacts in the target system (*AS3, QD1false and one of QD*2–4 *false*). The migration effort *f*(*x*) differs between the scenarios: $f(AS1_i) < f(AS2_i) < f(AS3_i)$; $i \in \{Process, Data, Form, Rule, Simulation\}$.

Table 2. Guiding questions

Label	Question	Answer Type
*QD*1	Is the transfer of design artifacts between the start and target systems possible using available export and import functionalities?	boolean
QD2	Does the start system provide an export functionality for the design artifact type?	boolean
QD3	Does the target system provide an import functionality for the design artifact type?	boolean
QD4	Is an adapter technology available?	boolean
QD5	Are the same notations used to represent the design artifacts in the start and target systems?	boolean
QD6	Which notation is used to represent the design artifact in the target system?	'name of the standard' / 'custom'
QD7	Does the target system offer interfaces for non-technical users?	boolean
*QD*8	Does the target systems provide templates for the design artifacts modeled in the start system?	percentage
$QD9_{\text{Form}}$	Does the target system provide automatic form generation?	boolean
QI	How much of the integration adapters required by the project are available by the vendor of the target system?	percentage
QT1	Do the start and the target system share a design tool?	boolean
*QT*2	Do the start and the target systems share an integration tool?	boolean
*QT*3	Do start and target system share the execution environment?	boolean
QC	Are the concepts of the enactable business model from the start and the target systems compatible?	boolean

The remaining questions related to the design component (*QD*5 to $QD9_{\text{Form}}$) in Table 2 aim at estimating the effort in the *AS*3 situations, meaning a transfer of a design object by recreating it within the target system, e.g. by remodeling a process model. Therefore, *QD*5 and *QD*6 refer to the notations of respective models. If the start and the target system use different notations for the same artifact the type the migration requires the acquisition of know-how related to the notation of the target system. Commonly, this is possible either by hiring experts or by training the available staff. Furthermore, it can be assumed that knowledge about a standard notation can be acquired easier compared to custom notations. This assumption is based on the fact that a larger number of experts and training offerings are available on the market for standard notations, since standards are not limited to a single vendor. In consequence, different Knowledge Demands (*KD*) generating different effort degrees are identified: The same notation in source and target system (*KD*1, *QD*5 *true*); Different notations and standard notation in target system (*KD*2, *QD*5 *false and QD*6 = *'standard'*); and Different notations and custom notation in target system (*KD*3, *QD*5 *false and QD*6 = *'custom'*). General assumptions regarding the effort are: f($KD1_i$) < f($KD2_i$) < f($KD3_i$)*; i* ∈{*Process, Data, Form, Rule, Simulation*}.

*QD*7 evaluates if there is a support for non-technical users. Craggs [18] argues that non-technical users such as business analysts provide knowledge of the business case that should be automated by the BPMS. In *AS*3 scenarios this knowledge has to be remodeled. Non-technical user interfaces, such as drag-and-drop capabilities, allow storing business design information into a BPMS without expert knowledge about the used notation. This decreases the migration effort for two reasons: the number of required technical users can be minimized by including business analysts in the migration process and less know-how about the notation of the target is needed. This decreases the knowledge acquisition costs.

*QD*8 aims to include the influence of industry knowledge provided by the BPMS vendor on the migration effort. As mentioned in the application of the frame of reference by Lustratus [19], some vendors of BPMS offer industry knowledge in the form of templates, e.g. business process model templates for common business cases. The answer of *QD*8 should state how many design artifacts do not need to be recreated manually in the target system because of templates that cover these artifacts. Since this statement has to be relative to the overall size of the artifact of a certain migration project, it is given in percentages.

$QD9_{\text{Form}}$ assesses the existence of generators that can create forms automatically based on data objects. These forms still require customization to be used in production environments, but it can be assumed that form generators decrease the migration effort related to forms. The impact of form generators depends on their quality, thus, some metrics regarding the generator quality have to be taken into account in effort estimation.

QI concerns the *Integration Infrastructure*. *QI* refers to the available adapter components for the integration of existing information systems. The estimation is based on the ratio of existing adapters to the total number of needed adapters. While there is a base effort for adapter configuration, the effort increases if adapters have to be designed and implemented. BPMSs support for this task may differ. The supported protocols of the Integration Infrastructure are not further considered since these are reflected in the available adapters.

Architecture is a relevant characteristic of the *Technical Infrastructure* component. The example of TIBCO iProcess and TIBCO ActiveMatrix BPM shows that two different BPMSs can share design and integration tools. Thus, these BPMS parts do not need to be migrated. A considerable amount of migration effort can be avoided. *QT*1 to *QT*3 assess the possibility to share BPMS components. Other characteristics of the *Technical Infrastructure* such as hardware and software requirements are not further considered regarding the migration effort because this is not a special issue of BPMS but rather a standard problem of IT management.

However, even if core components are shared between BPMS, a migration remains necessary when the start and the target systems do not share the same concept of an enactable business model. This essential aspect is covered in question *QC*. With respect to a BPMS migration, different concepts of enactable business models mean different demands of information for the start and the target BPMS (cf. Section 2.2). Therefore, an additional effort has to be considered that covers the acquisition and implementation of respective design artifacts.

The overall effort of a migration project is based on two main aspects: the effort for transferring the all design artifacts from the start system to the target system and the effort of integrating all the required business software into the target system. Moreover, overall effort of a migration project has to include the installation of the design tools, the integration infrastructure and the tools related to the Execution component such as the process engine, analysis tools and, additionally, the business rules or event engines. Another aspect that has not been discussed in the context of this framework is the training effort for the different user groups. Only the costs of knowledge acquisition effort regarding different notations are covered. However, the technical users, the non-technical users and the end user have to become familiar with the respective user interfaces of the target system. If certain components are shared, the training effort for these components would be reduced.

3.3 Operationalization of the Cost Drivers

After discussing the general influences on BPMS migration effort based on the proposed framework, cost drivers have to be operationalized in order to provide metrics for a calculation of the estimated effort. Not all items of the provided calculation schema are discussed in detail here. However, the complete sets of functions and drivers used in the schema are depicted in Figures 4 and 5.

$$\begin{aligned} f_{total}\,(X, bv, A, u) = {} & \neg g(QT1, bv) * (\sum_{\mathrm{i}} f_{design}\,(X_i, bv) + \varsigma_{design} \\ & + \neg g(QT2, bv) * (f_{adapters}\,(A, bv) + \varsigma_{integration}\,) + \varsigma_{execution} \\ & + f_{user}\,(u, bv) + \neg g(QC, bv) * (\theta_{bv} + f_{create}\,(X_{unknown}\,, bv) \end{aligned}$$

$$\begin{aligned} f_{design}\,(X_i, bv) = {} & g(QD1_i, bv) * f_{import}\,(X_i) \\ & + g(QD2_i, bv) * g(QD3_i, bv) * g(QD4_i, bv) * f_{indirect}\,(X_i) \\ & + \neg(g(QD2_i, bv) * g(QD3_i, bv) * g(QD4_i, bv)) * f_{recreate}\,(X_i, bv) \end{aligned}$$

$$f_{import}\,(X_i) = |X_i| * (\iota_{ex} + \iota_{cp} + \iota_{im})$$

$$f_{indirect}\,(X_i) = |X_i| * (\kappa_{ex} + \kappa_{cp} + \kappa_{im})$$

$$f_{recreate}(X_i, bv) = f_{transfer}\,(X_i) + f_{knowledge}\,(X_i, bv)$$

$$\begin{aligned} f_{create}\,(X_{Unknown}\,, bv) = {} & f_{transfer}\,(X_{Unknown}\,) + f_{Unknown}\,(X_i, bv) \\ & + g_{collect}\,(X_{Unknown}\,) \end{aligned}$$

$$f_{user}\,(u, bv) = u_t * u_{non-t} * \eta_{non-t,bv} + u_{end} * \eta_{end\,,bv}$$

$$f_{adapters}\,(A, bv) = (\frac{g(QI, bv)}{100} * |A| * \epsilon + (1 - \frac{g(QI, bv)}{100}) * f_{develop}\,(A, bv_{target}\,))$$

$$f_{develop}\,(A, bv_{target}\,) = \sum_{j=1}^{|A|} g_{adapter}\,(a_j) * \varepsilon_{target}$$

$$f_{transfer}\,(X_i, bv) = (1 - \frac{g(QD8_i, bv)}{100}) * f_{manual}\,(X_i) + \frac{g(QD8_i, bv)}{100} * \gamma_i * |X_i|$$

$$f_{manual}\,(X_i) = \sum_{j=1}^{|X_i|} g_{complex}\,(x_j) * \beta_i$$

$$f_{generate}\,(X_{Form}\,, bv) = g(QD9_{Form}\,, bv) * \delta_{target} * f_{manual}\,(X_{Form}\,)$$

$$\begin{aligned} f_{transfer}\,(X_{Form}\,, bv) = {} & (1 - \frac{g(QD8_i, bv)}{100}) * f_{generate}\,(X_{Form\,,bv}) \\ & + \frac{g(QD8_i, bv)}{100} * \gamma_{Form} * |X_{Form}\,| \end{aligned}$$

$$\begin{aligned} f_{knowledge}\,(X_i, bv, u) = {} & g(QD7_i, bv) * g_{tech}(X_i)^{\frac{u_{non-t}}{u_t}} * f_{acq}\,(bv, u, i)) \\ & + \neg g(QD7_i, bv) * f_{acq}\,(bv, u, i)) \end{aligned}$$

$$f_{acq}\,(bv, u, i) = \neg g(QD5_i, bv) * u_t * \alpha_{notation}$$

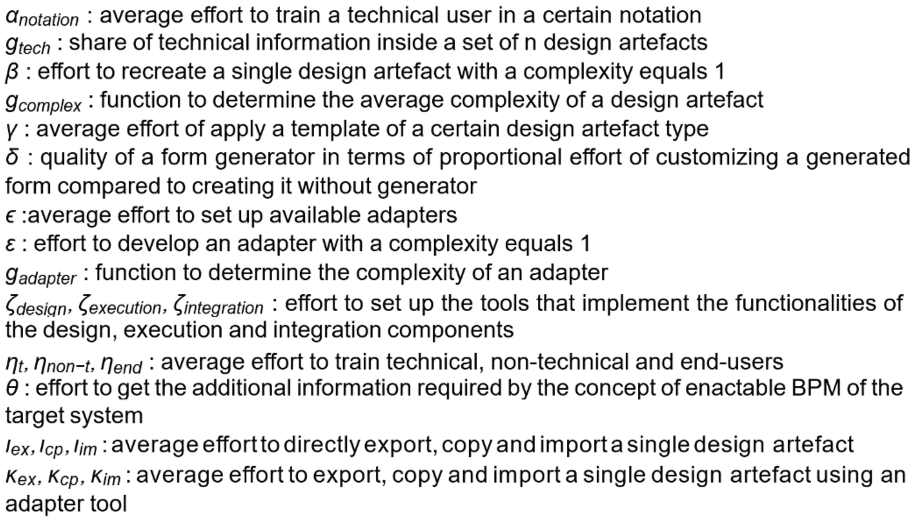
$$notation = g(QD6_i, bv)$$

Figure 4. Estimated effort calculation

The total effort f_{total} is a function of the design artifact sets X_{i} (Data, Process, …) that have to be migrated between the BPMS, the vector of involved BPMS *bv*, the set of available adapters for artifact transfer A, and the number of users u (u_t = technical users and $u_{non\text{-}t}$ = non-technical users like business analysts) involved.

f_{design} calculates the effort of transferring all artifacts of a certain type, which basically means that one of three sub-efforts needs to be determined. f_{import} represents the effort of ($AS1_i$), $f_{indirect}$ is related to the effort of ($AS2_i$) and $f_{recreate}$ calculates the effort of ($AS3_i$). For the $AS1_i$ situations (which mean that matching import and export functionalities are provided by the start and the target systems) the effort can be identified by multiplying the size of a set of design artifacts with the sum of the average costs of exporting a single element of this set (ι_{ex}), copying it to the target system (ι_{cp}) and importing it (ι_{im}). In case the respective tools of both BPMS allow importing all elements of a certain artifact type at the same time, f_{import} can be described as the sum effort of these single export(κ_{ex}), copy(κ_{cp}) and import(κ_{im}) processes.

$f_{adapters}$ calculates the effort for integration. This includes the effort for integration adapter development $f_{develop}$. $f_{transfer}$ calculates the effort for the manual transfer of design artifacts from the source to target BPMS. This includes effort savings depending on the existence of templates. Additionally, in the case of forms artifacts, the influence of form generators has to be considered. $f_{knowledge}$ addresses the need of knowledge acquisition by the users involved in BPMS migration. As stated in the previous section, this effort depends on the existence of interfaces for non-technical users and, of course, on the number of users. The calculations are based on several values that need either expert estimation from past experiences with BPMS maintenance, implementation and use or empirical data on these efforts. These values are treated as constants in the calculation model (see Figure 5). A special function *g(Q*,bv)* operationalizes the answers to the questions. A special design artifact type *unknown* has been added for the case when the concepts of the enactable business models differ.



$\alpha_{notation}$: average effort to train a technical user in a certain notation
g_{tech} : share of technical information inside a set of n design artefacts
β : effort to recreate a single design artefact with a complexity equals 1
$g_{complex}$: function to determine the average complexity of a design artefact
γ : average effort of apply a template of a certain design artefact type
δ : quality of a form generator in terms of proportional effort of customizing a generated form compared to creating it without generator
ϵ :average effort to set up available adapters
ε : effort to develop an adapter with a complexity equals 1
$g_{adapter}$: function to determine the complexity of an adapter
$\zeta_{design}, \zeta_{execution}, \zeta_{integration}$: effort to set up the tools that implement the functionalities of the design, execution and integration components
$\eta_t, \eta_{non-t}, \eta_{end}$: average effort to train technical, non-technical and end-users
θ : effort to get the additional information required by the concept of enactable BPM of the target system
$\iota_{ex}, \iota_{cp}, \iota_{im}$: average effort to directly export, copy and import a single design artefact
$\kappa_{ex}, \kappa_{cp}, \kappa_{im}$: average effort to export, copy and import a single design artefact using an adapter tool

Figure 5. Cost drivers for effort estimation

4 Application Demonstration

In order to demonstrate the application of the framework, effort estimation on an exemplary migration from TIBCO iProcess to jBPM from JBoss is shown in the following section: $bv = (iProcess, jBPM)$. First, the migration scenario is introduced and in the next step the guiding questions and the formulas for effort estimation are applied.

4.1 Migration Scenario

For the reason of brevity, a BPMS migration of a single process only is considered in a scenario with limited complexity. A single business process for the approval of customer orders is being migrated. In this process, an order is created by a sales representative. Such an order might be subject to an approval process which includes, for instance, a team leader or a branch manager. If the order is approved, it will be available for further processing. If the approval is denied, the sales representative is requested to change the order in negotiation with the customer. If this negotiation fails or a threshold number of order approval attempts is reached, the order will be cancelled. The applied business rule is shown in Table 3.

Table 3. Approval level selection rule

Order Type	Contact Type	Approval Level
Reimbursement up to 50 Euro	Any	No approval required
Reimbursement from 50 to 5000 Euro	By telephone	Require authorizing authority
Reimbursement from 50 to 5000 Euro	Personal	Require team leader
Reimbursement from 50 to 5000 Euro	Internal	Require authorizing authority
Reimbursement of 5000 Euro and higher	By telephone	Require branch manager
Reimbursement of 5000 Euro and higher	Personal	Require branch manager
Reimbursement of 5000 Euro and higher	Internal	Require branch manager

To keep the example case small, only design artifacts and workflow execution components are considered for migration. The source BPMS for the migration is a TIBCO iProcess installation that uses the iProcess Modeler as a tool to create the necessary design artifacts. There is an alternative Solution – the TIBCO Business Studio – which supports BMN 2.0 notation and XPDL export. However, TIBCO iProcess was initially delivered with the iProcess Modeler. It has not been replaced in this scenario. Consequently, all design artifacts and the Execution component use proprietary TIBCO data formats and interfaces. jBPM, the target BPMS, provides design tools that support the involvement of non-technical users in the design of processes, data, and forms. Process models in BPMN 2.0 are used. Import and export are possible using the XPDL format. Drools is used for rule execution. Thus rules can be defined using the Drools specific notation. Apart from these points, jBPM is also using proprietary data formats and interfaces for design and process execution.

In this small scenario one technical user is performing the migration task while 30 end-users have to be trained on the new BPMS: $u = (1,0,30)$.

Based on the migrated process, the following design artifacts have to be migrated: $X_{process}$ = {*Order Approval Process*}, X_{data} = {*Order Data*}, X_{form} = {*Create Order Form, Change Order Form, Approve Order Form, Cancel Order Form*}, X_{rule} = {*Approval Level Selection Rule*}, $X_{simulation}$={}. Table 4 shows a specification of the data and form artifacts.

Table 4. Data object and form specification

Attribute	Type	Create	Change	Approve	Cancel
Order id	String	-	Show	Show	Show
Creator Id	String	-	Show	Show	Show
Evaluator Id	String	-	Show	Show	Show
Last Editor Id	String	-	Show	Show	Show
Order Type	String	Edit	Edit	Show	Show
Value	Float	Edit	Edit	Show	Show
Contact Type	String	Edit	Edit	Show	Show
Contact Id	String	Edit	Edit	Show	Show

4.2 Effort Estimation

The first step in applying the framework for effort estimation is answering the guiding questions. Thus, the appropriate steps for the migration of design artifacts and BPMS components can be determined. Based on this, cost functions for the effort estimation are selected. Tables 5 and 6 show the resulting answers.

Table 5. Guiding questions and answers (1)

Question	Answer
QI	-
*QT*1	False
*QT*2	-
*QT*3	False
QC	True

Table 6. Guiding questions and answers (2)

	Process	Data	Forms	Rules
*QD*1	False	False	False	False
QD2	False	False	False	False
QD3	True	True	True	True
QD4	False	False	False	False
QD5	False	False	False	False
QD6	BPMN	Custom	HTML + CSS	Drools
QD7	True	True	True	False
QD8	0 %	0 %	0 %	0 %
QD9_{Form}	-	-	True	-

Based on the given answers, the resulting migration process can be described as follows. Although the target jBPM provides import functionalities for design artifacts (*QD*3), no appropriate export functionality is available in TIBCO iProcess (*QD*1, *QD*4). The only option to migrate the design artifacts is a recreation in the target system (jBPM). This is described as an *AS*3 scenario in Section 3. There are no design templates available that would lower the effort for recreation (*QD*8). However, a form generator that automatically creates forms based on the data model is available (*QD*9). The modeling tool of jBPM supports the involvement of non-technical users (*QD*7), this could reduce the modeling effort. In the example case, this is not possible because a technical user only is planned for the migration. *QD*6 allows an assessment of needed skills for BPMS migration by collecting used notations. The total effort is determined by the following formula:

$$f_{total}(X, bv, A, u) = \sum_i f_{design}(X_i, bv, u) + \zeta_{design} + \zeta_{execution} + f_{user}(u, bv) \qquad (1)$$

Thus, the total effort consists of (1) the effort for design artifact migration, (2) the installation and configuration effort for the modeling and execution environment, and (3) the effort for end user training. Based on the formulas in Figure 4 and the answers to the guiding questions, the following formula applies for the effort of design artifact migration (derived from Figure 4):

$$f_{design}(X_i, bv, u) = f_{recreate}(X_i, bv, u) = f_{manual}(X_i, bv) + f_{knowledge}(X_i, bv, u) \qquad (2)$$

The effort of design artifact migration is the sum of the effort for manual recreation of design artifacts and the effort for knowledge acquisition regarding notations and interfaces of the target system. An exception is the creation of forms. Here, the automated form generation lowers the effort. Thus, a special term $f_{generate}(X_{Form}, bv)$ applies. As a result, the total effort function can be broken down to cost drivers and effort estimation for single tasks in the migration process (derived from Figure 4):

$$
\begin{aligned}
f_{total}\,(X, bv, A, u) = & \sum_{i \neq Form} \Big(\sum_{j=1}^{|X_i|} g_{complex}\,(x_j) \times \beta_i\Big) \\
& + \delta_{target} \times \sum_{j=1}^{|X_{Form}|} g_{complex}\,(x_j) \times \beta_{Form} \\
& + \sum_{i} \alpha_g(QD6_i, bv) \\
& + \zeta_{design} + \zeta_{execution} \\
& + u_t \times \eta_{t,bv} + u_{non-t} \times \eta_{non-t,bv} + u_{end} \times \eta_{end,bv}
\end{aligned} \tag{3}
$$

Now, the values of the cost drivers and normalized effort for single tasks have to be determined. This can be done either based on the analysis of historical empirical data or by experts. For the presented case, an expert estimation has been performed. The results can be seen in Tables 7 and 8. While there is only one process, one data object and one rule, there are 4 forms to be considered.

Table 7. Cost drivers and normalized efforts (design artifacts)

	Process	Data	Forms	Rules
Complexity $g_{complex}(x)$	43.1	8.2	4; 8.4; 8.8; 8.8	9
Effort for artifact creation $\beta_i$in h	0.125	0.075	0.01	0.1
Effort for knowledge acq. $\alpha_i$in h	5	10	5	15

Table 8. Cost drivers and normalized efforts

Correction factor for form generation δ_{target}	0.2
Installation and configuration design environment ζ_{design}	0.25 h
Installation and configuration execution components $\zeta_{execution}$	0.25 h
Training effort per technical user $\eta_{t,bv}$	1.5 h
Training effort per end user $\eta_{end,bv}$	3.0 h

Since no distributed environment had been set up for the example case, the installation and configuration effort for the design environment and the execution components was estimated rather small. Additionally, jBPM was available in a complete installation package that included both, all required design tools and the execution components. Overall, a total effort of $f_{total}(X, bv, A, u) = 134$ h has been estimated. Omitting the effort for end-user training and knowledge acquisition, this comes down to a migration effort of 9 h for an experienced IT professional. This is roughly the time needed for an experimental jBPM setup of the case which took 9.9 h.

Although no empirical data was available to tune the influence factors of the cost drivers, this demonstration showed a good estimation accuracy.

5 Conclusion

The framework for effort estimation has been demonstrated as being applicable for its purpose based on a small case study evaluating a BPMS migration scenario. The framework is open for additional design artifact sets. One set that could be added is the organizational model, looking at the case study using TIBCO iProcess and Active Matrix BPM. The comparison of BPMS based on the framework is not bound to the purpose of migration; it could also be applied for BPMS introduction. However, its application should be embedded in a BPMS selection process. The estimated effort is strictly related to the technical effort of migration. Maintenance is not considered. The training costs are only calculated for the staff that is involved in migration. Since the BPMS in the case study did not share execution components (see *QT3* in Section 3.2 and Table 5 in Section 4), this aspect has not been investigated in the calculations. However, in this regard it has to be taken into account that, based on existing executable process model standards (e.g. BPEL or XPDL), there may be cases of a shared execution engine of different BPMS.

The framework provides guidelines and ideas for the estimation of BPS migration effort. Further steps of its refinement and validation are necessary

Regarding validity of the presented results, there is a limitation due to the small number of publications found. Furthermore, more of known BPMS architectures should be compared to the presented framework and a larger case study considered.

References

[1] D.Karagiannis, “BPMS: business process management systems,” *ACM SIGOIS Bulletin*, vol. 16(1), pp. 10–13, 1995. Available: https://doi.org/10.1145/209891.209894

[2] E. Paslaru Bontas Simperl, C. Tempich, and Y. Sure, “Ontocom: A cost estimation model for ontology engineering,” *in The Semantic Web - ISWC 2006. Lecture Notes in Computer Science*, vol. 4273. Springer, pp. 625–639, 2006. Available: https://doi.org/10.1007/11926078_45

[3] C. Drews and B. Lantow, “A Framework for Effort Estimation in BPMS Migration,” *Joint Proceedings of the BIR 2017 pre-BIR Forum, Workshops and Doctoral Consortium co-located with 16th International Conference on Perspectives in Business Informatics Research (BIR 2017)*, *CEUR Workshop Proceedings*, vol. 1898, 2017.

[4] B. Kitchenham. *Procedures for Undertaking Systematic Reviews*. Computer Science Department, Keele University and National ICT Australia Ltd, 2004.

[5] Conferences on Business Process Management. Available: https://bpm-conference.org/BpmConference/ (cited 6.2.2018)

[6] Business Process Management Journal [Online]. Emerald Publishing. ISSN: 1463-7154 Available: http:\\www.emeraldgrouppublishing.com\products\journals\journals.htm?id=BPMJ (cited 7.7.2017)

[7] Lustratus research limited [Online]. Available: http://www.lustratus.com (cited 7.7.2017)

[8] U. Kannengiesser, M. Neubauer, and R. Heininger, “OPC UA interface for a BPM suite to enable seamless process management,” *BPM Demo Session 2015, BPMD 2015 – co-located with the 13th International Conference on Business Process Management, BPM 2015, Innsbruck, Austria, CEUR Workshop Proceedings*, vol. 1418, pp. 95–99, 2015.

[9] H. A. Reijers, M. H. Jansen-Vullers, M. Z. Muehlen, and W. Appl, “Workflow management systems + swarm intelligence = Dynamic task assignment for emergency management applications,” *5th International Conference on Business Process Management, BPM 2007, Brisbane, Australia, Lecture Notes in Computer Science*, vol. 4714, Springer, pp. 125–140, 2007. Available: https://doi.org/10.1007/978-3-540-75183-0_10

[10] A. Rulle and J. Siegeris, “From a family of state-centric PAIS to a configurable and parameterized business process architecture,” *in Business Process Management. BPM 2014. Lecture Notes in Computer Science*, vol. 8659, Springer, pp. 333–348, 2014. Available: https://doi.org/10.1007/978-3-319-10172-9_21

[11] V. Ferme, A. Ivanchikj, and C. Pautasso, “A framework for benchmarking BPMN 2.0 workflow management systems,” *in Business Process Management. BPM 2015. Lecture Notes in Computer Science*, vol. 9253, Springer, pp. 251–259, 2015. Available: https://doi.org/10.1007/978-3-319-23063-4_18

[12] F. M. Al-Ghaiati, “A proposal for an open solution business process integration and management implementation framework,” *in Business Process Management. BPM 2006. Lecture Notes in Computer Science*, vol. 4102, Springer, pp. 338–347, 2006. Available: https://doi.org/10.1007/11841760_23

[13] M. Cheung and J. Hidders, “Round-trip iterative business process modelling between BPA and BPMS tools,” *Business Process Management Journal*, vol. 17(3), pp. 461–494, 2011. Available: https://doi.org/10.1108/14637151111136379

[14] C. Janiesch, M. Matzner, and O. Müller, “Beyond process monitoring: A proof-of-concept of event-driven business activity management*,” Business Process Management Journal*, vol. 18(4), pp. 625–643, 2012. Available: https://doi.org/10.1108/14637151211253765

[15] D. M. R. Ferreira and J. J. P. Ferreira, “Towards a workflow-based integration architecture for business networking,” *Business Process Management Journal*, vol. 11(5), pp. 517–531, 2005. Available: https://doi.org/10.1108/14637150510619867

[16] Y. Verginadis and G. Mentzas, “Agents and workflow engines for inter-organizational workflows in e-government cases,” *Business Process Management Journal*, vol. 14(2), pp.188–203, 2008. Available: https://doi.org/10.1108/14637150810864925

[17] D. R. Shaw, C. P. Holland, P. Kawalek, B. Snowdon, and B. Warboys, “Elements of a business process management system,” *Business Process Management Journal*, vol. 13(1), pp. 91–107, 2007. Available: https://doi.org/10.1108/14637150710721140

[18] S. Craggs. *Comparing bpm from appian, oracle and ibm*. Lustratus Research, 2012.

[19] S. Craggs. *Comparing bpm from pegasystems, ibm and tibco*. Lustratus Research, 2011.

[20] S. Craggs. *Comparing bpm from ibm, oracle and sap.* Lustratus Research, 2009.

[21] S. Craggs. *Comparing bpm from ibm, software ag and pegasystems.* Lustratus Research, 2010.

[22] B. Warboys, P. Kawalek, I. Robertson, and M. Greenwood. *Business Information Systems: A Process Approach*, McGraw-Hill, New York, 1999.

[23] B. W. Boehm, R. Madachy, and B. Steece, *Software cost estimation with Cocomo II with Cdrom*, Prentice Hall PTR Upper Saddle River, NJ, USA, 2000.