FLEXIBLE REALIZATION OF BUSINESS PROCESSES USING EXISTING SERVICES

Jelena Zdravkovic
Deptartment of Computer Sciences,University of Gävle and Royal Institute of Technology,

Martin Henkel
Department of Sydtem of Computer Sciences, Stockholm University and Royal Institute of Technology,

Keywords: Business processes, Executable processes, Service coordination, Flexibility.

Abstract: When realizing executable business process models, in most situations process specifications collide with specific properties of existing services. In this paper we propose an approach for relaxation of the business process specification to enable flexible integration between the process and existing services. The approach is based on the notion of visibility, which allows a categorized relaxation of the process specification by not requiring every process state to be distinguished after the process is realised with existing services. The categories of visibility presented in this paper are applied by indicating flexible elements in the process design phase. The presented approach stimulates the alignment between business processes and existing services, facilitating a larger scale of transparent process realisations.

1 INTRODUCTION

Cross-enterprise e-collaboration requires that both business activities and their supporting systems are coordinated. From the business perspective, business activities must be designed to cope with requirements from all involved actors; for example an e-business solution might need to deal with both companies and private persons. The business activities, the involved actors, and the information the business need to control, are commonly described by using the notion of a *business process*. From the system perspective, business activities must be supported by technology dependant system services. These services need to be coordinated in order to support the business process. The coordination of services can be achieved by using executable process description languages such as BPEL4WS (BEA, 2003). As well as handling business documents, the system services must cope with additional details, for example, with message format transformations and differences in communication protocols. Due to the added technical details we denote a process that coordinates system services as a *technical process*. A business process thus deals with pure business concepts, while a technical process additionally deals with the technical environment (protocols, software products, etc.) that are required to provide system support for the business.

It is paramount that a business process and its corresponding technical process are designed such that the technical process can represent the possible states in the business process. It means that the technical process must be aligned with the business process. An effect of this alignment is that changes made to the business process might affect the technical process and vice versa.

An increasingly important characteristic of executable processes that span across organizations is flexibility. Constructing flexible business and technical processes means that they can accommodate changing business requirements, without a major redesign. A problem with respect to flexibility is that large organizations rely on legacy systems for their core business; thus the technical foundation is fixed rather than flexible. This means that changes in the business requirements that affect the technical process can be difficult to implement. One solution to the problem is simply to avoid business process designs that break the alignment

Zdravkovic J. and Henkel M. (2006).
FLEXIBLE REALIZATION OF BUSINESS PROCESSES USING EXISTING SERVICES.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 165-172
DOI: 10.5220/0002444401650172

with current support systems. However, this is far from ideal since it would let the legacy systems govern the business development; also, it is infeasible for the business process designers to keep track of all system limitations. Another solution, which we propose in this paper, is to let the business designer govern the quality of alignment between the business and technical process.

Existing proposals on how to introduce flexibility in process specifications can be classified into two categories. Firstly, some authors suggest making the process specification more abstract. By raising the abstraction level the process loses details, but at the same time captures a wider range of behaviour. Examples of constructs that raise the abstraction level are the use of ad-hoc sub-processes (Heinl, 1999), activity inheritance (Aalst, 1999), (Ribó, 2001) and patterns of flexibility (Sadiq, 2001). The second approach to process flexibility is to introduce constructs that are tailored to handle complex behaviour. These constructs are commonly "cross-cutting", i.e. they control behaviour on a set of activities or an entire process. Examples of such constructs are Event-Condition-Action (ECA) rules (Joeris, 1999), process parameterization (Aalst, 1999) and the use of profiles to describe complex error handling (Chopra, 2004). The approach presented in this paper belongs to the first category - the notion of visibility levels is a construct that enables the business process designer to construct processes that are less specified, and thus more abstract. However, our approach differs in two ways. Firstly, we introduce the notion of process visibility. This enables us to augment an existing process specification without changing its activities or flow constructs. Secondly, we specifically target the case where a process needs to fit on top of existing technical foundations. Thus our approach is to keep the alignment of two processes, business and technical, flexible.

The paper is structured as follows. Section 2 illustrates a business process and shows how its realization as a technical process is influenced by existing services. Section 3 introduces the notion of process visibility and categorizes it into three levels. In Section 4, using a structured process design framework, we define the rules for how to determine the levels of visibility to align a business process with its technical realization. Section 5 explains how to apply the proposed notion during design and implementation of business processes. Finally, Section 6 concludes the paper.

2 EXAMPLE CASE

An example of a business process and its realization as a technical process is shown in Figure 1. The model illustrates a process developed under the Serviam project (Serviam, 2005) to investigate capabilities of the SEB bank (a North European financial group) to integrate and coordinate its ERP systems in the form of Web services. The business process in Figure 1(a) depicts an excerpt of the process used to supply customers with various types of furniture, using Itea (Itea, 2005), a virtual sale portal. The model is expressed in the Business Process Modelling Notation (White, 2004). The BPMN is used to visually model a process management, which might further be converted to a process language (such as BPEL4WS).

Upon receiving an order request, Itea retrieves the customer contact and the customer's order history and then verifies the order amount and details. After the order is processed by a supplier, if the furniture is available, the customer's account is debited for the amount of the purchase. The order confirmation is then sent to the customer.

The corresponding technical process, Figure 1(b), is based on existing services, provided by Itea's internal Order system and by its partners (the bank - SEB and the furniture manufacturers – Mio or its partners). Compared to the business process, the technical process must adhere to capabilities of existing services:

- The customer contact and the order history are retrieved with a single activity, because the needed information is provided by a single service.
- The execution order of verification of the order details and amount is not visible, because these activities correspond to a single service.
- Processing of an order and debiting of customer account are managed as an atomic (two-phase commit) transaction (AT), because the corresponding services do not support compensations needed for a long-running transaction (LRT).
- Order confirmation is sent by a single activity, because it is implemented as a single service, which encloses use of e-mail and fax protocols.

The example shows the impact that existing services might have on the realization of a business process. It is obvious that due to particular properties of the services (granularity, task ordering, transactional properties, etc.), the technical process cannot capture all states the business process passes through.

In the next section we define conditions under which a technical process might be considered as aligned with a business process, and how those conditions may be relaxed to increase abilities for the alignment.

3 LEVELS OF VISIBILITY FOR BUSINESS PROCESSES

A technical process is a realization of a business process by the use of existing services. When designing processes, different aspects are to be considered. In our previous work (Henkel, 2004), we have, based on the workflow-modelling studies (Jablonski, 1998) and (Rausch-Scott, 1997), defined a framework containing five main aspects that constitute process design: *functional*, *behavioural*, *organizational*, *informational* and *transactional*. When designing business and technical processes, each of the five aspects must be regarded. In (Zdravkovic, 2005), we argued that the basic criterion for a technical process to *realize* a business process is that *a technical process must be designed to trace all states of a business process, where the content of a single process state comprises the statuses of all five design aspects.*

In reality, specific properties of existing services often collide with detailed specifications of business processes. It is therefore difficult to obtain a technical process that strictly realizes a business process specification. In the example case in Figure 1, the business process retrieves the customer information with two activities "Get customer contact" and "Get order history", while in the technical process the same information is obtained from a single activity "Get customer information"; this means that the technical process can not trace the state between the two activities in the business process. As another example, it might be that a set of business process activities is governed by a long-running transaction, while the corresponding services might be managed only as an atomic transaction (i.e. as a "black-box"). In both discussed examples the business process cannot be realized with the existing services because the required business states are not captured in the technical process.

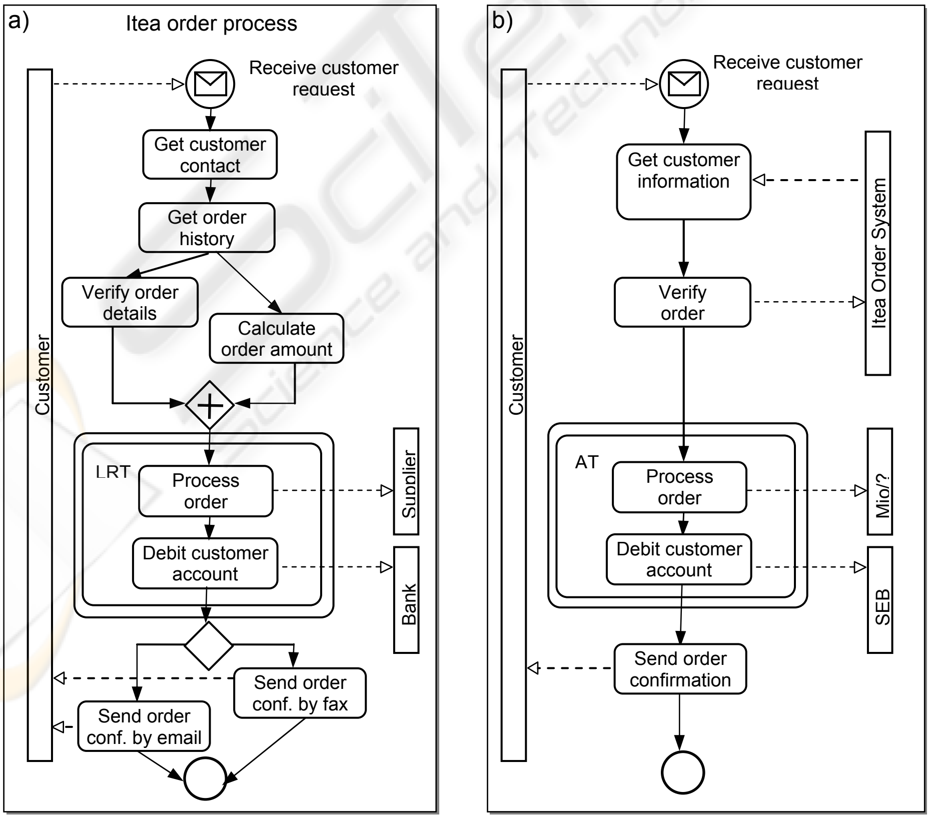

Figure 1: Business process (a) and technical process (b), presented in the BPMN form (White, 2004).

To enlarge abilities for realizations of business processes, it is thus important to have a mean to relax the realization requirements. A way to achieve this is to distinguish the states in the business process that must be *visible* in the technical process, from those states that might be hidden in services (Figure 2). The distinction is determined according to:
A business process state must be visible in the technical process if its content is used by the business process environment, i.e. by internal and/or external actors that interact with the process; otherwise the state needs not to be visible.

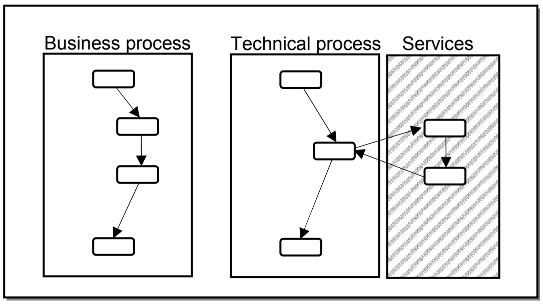

Figure 2: States hidden in a service implementation.

For example, it might be important that a set of business activities are executed in a predefined order, but the business might not require visible tracking of the run-time execution order. Not requiring visible tracking allows the use of existing services that hide the execution order of activities. In this way a business process that was unrealizable, or "incompatible" with existing services, becomes realizable in a limited way. It might be also required to support visibility of the process states for a group of instances, while not for the others. Following this, flexibility in design of a business process may be discerned in three *levels of visibility*:

- *Loss-full visibility:* the flexibility is chosen when a set of states of the business process need not to be captured, because the contents of those states are not used by the process environment for any of the process instances.
- *Constrained visibility:* the flexibility is chosen when a set of states of the business process need not to be captured for particular process instances, while they must be captured for the other instances.
- *Lossless visibility:* the flexibility is chosen when a set of states of the business process must be captured, because their contents are used by the process environment for all process instances.

As it may be seen from the above categorization, the loss-full visibility gives maximum flexibility for realizing a business process, while the lossless visibility gives minimum. The constrained visibility is used when neither lossless nor loss-full visibility might be applied – in that situation, the business process states are set to be visible on the "case" basis. Having in this way categorized concepts of flexibility, allows the business process designer to relax requirements for alignment of a business process with its technical realization, by selecting flexible process elements with an adequate level of visibility.

4 REQUIREMENTS FOR VISIBILITY IN PROCESS DESIGN

Using the five-aspect design framework that we have introduced in the previous section, in the following, we define criteria for discerning the levels of visibility for each of the design aspects.

Functional Aspect. The functional aspect considers the activities that are to be executed in a process. For each activity the functionality is determined by three elements: the activity name which describes the *result* to be achieved, *exchanged messages*, and input and output constraints that form *pre-conditions* and *post-conditions*. In a business process, decomposition of activities is done according to recognized business tasks. For instance, the distinguished business concepts "verify order amount" and "debit customer account" will be administered by two distinct process activities.

Existing services might be designed to support functionality required by a business process, but without "notifying" the process about the fulfillment. This is the case when the granularity of a single service is designed such that the service encompasses functionality of more than one business activity. Thereby, the technical process is not able to capture the states to distinguish the exchanged messages or results or pre- and post-conditions of each of the business activities.

For instance, in Figure 1, we may see that in the business process the customer information is retrieved with two activities - "Get customer contact" and "Get order history", because from the business perspective those information concepts are handled by distinct business tasks. Since the two messages defined in the business process are exchanged internally, following the rule defined in the previous section, the minimal level of visibility is determined by the requirements of the internal actors governing the customer information. If none of them need to distinguish the customer contact from the order history, the visibility of the two activities might be set to the loss-full. Thereby, the technical process, exchanging the customer information with a single service, will become aligned with the business process. If, however, the messages has to be separately available for all customers (lossless visibility), or at least for some instances (constrained visibility), alignment between the two processes is not reachable without redesigning existing services.

In general, the requirement for visibility of the functional aspect of a business process can be determined by the business process designer by applying the following guiding question:

	Visibility of the functional aspect
Q	Is it mandatory to capture the messages, the result, or the pre- and post-conditions of a business process activity, in the technical process?
A	- No, because the business process environment does not use (i.e. does not need to distinguish) any of the three functional elements of the activity (loss-full visibility, LFV) - Yes, because the business process environment uses at least one of the functional elements of the activity (lossless visibility, LLV). - Yes, for some process instances (constrained visibility, CRV).

Behavioural Aspect. The behavioural aspect depicts process control flow, i.e. when an activity is to be executed in relation to others. For specification of dependencies and coordination rules among activities, process specifications rely on a set of basic control flow constructs: ordering (sequence, parallel execution), and conditional branching (OR/XOR). In a business process the use of the control flow constructs is determined by identifying flow dependencies among business activities.

When realizing a business process, existing services are used to implement the required flow constructs. The granularity of these services might be such that they encompass some execution order as well as branching conditions, i.e. existing services might govern it internally. For instance, examining visibility of branching conditions, it may be seen that in the business process in Figure 1(a) the order confirmation is sent using one of the two protocols: e-mail or fax. The selection of the protocols is chosen based on the customer profile data. However, as Itea does not oblige to inform customers on the protocol used for sending the order confirmation, it means that the visibility of the selected condition is not required in the business process (i.e. might be set to loss-full). As the implementation is provided by a single service that includes the selection of a protocol, the corresponding technical process, not supporting the condition selection, becomes aligned with the original process.

	Visibility of the behavioural aspect
Q	Is it mandatory to capture the ordering of a set of activities in a business process? Is it mandatory to capture the conditions of a branch in a business process?
A	No - LFV; Yes - LLV; for some instances - CRV.

Informational Aspect The informational aspect of a process concerns the concepts needed for representing process internal data and the data that the process exchanges with the environment in the form of messages (and documents). In a business process the documents are modelled to capture relevant information on business concepts such as customers, orders, products, etc.

Even thought existing systems are designed to support the required business information, the information structures of services (i.e. input and output documents) might not provide the contents required by a business process.

Visibility of the information concepts of the business process in Figure 1 cannot be determined directly from the given BPMN model, as this requires comparison of the message documents in both processes. By doing that, as an example, we find that the business process requires for international customers the address structure in the “Get customer contact” message, to contain two addresses – one that is the customer’s registered address (in the bank), and the other used for the product delivery; the latter address is necessary to have to calculate the overall order amount. However, the delivery address is not needed for domestic customers, because for those customers delivery expenses are fixed (same). This means that the visibility of the address might be set to constrained. The requirement for visibility of the informational aspect of a business process is generally determined by using the following rule:

	Visibility of the informational aspect
Q	Is it mandatory to capture the content of an information concept defined in a business process?
A	No - LFV; Yes - LLV; for some instances - CRV.

Transactional Aspect The transactional aspect governs consistent execution of a set of activities (implemented by services). Process transactions comply with two different models. The atomic transaction (AT) model (Bernstein, 1987) is used to control a set of shorter services such that the outcome is visible only when all services within a transaction finish successfully. The long-running transaction (LRT) model (Garcia-Molina, 1991) rules more durable services, where each service enforces a globally visible outcome independently of the other services. This means that the models differ in the exposure of the intermediate transactional states. Thus, when designing transactions in a business process, the selection of the model is determined upon necessity on visibility of internal transactional states.

Concerning the example from Figure 1(a), business process activities "Process order" and "Debit customer account" are designed as long-running, in order to capture the supplier information on availability of the furniture. However, as Itea still does not offer the ability for a partial delivery (might be used when the supplier informs on partial order availability), the current business requirements would be satisfied even without ability for using the internal transactional results. This means that the visibility of the business process transaction might be set to loss-full. The technical process, supporting the atomic model, would then be aligned with the originally designed process.

The requirement for visibility of the transactional aspect of a business process is generally determined by using the following rule:

	Visibility of the transactional aspect
Q	Is it mandatory to capture the internal states of a transaction in a business process?
A	No - LFV; Yes - LLV; for some instances - CRV.

Organizational Aspect. The organizational aspect concerns the distribution and control of responsibility for executing activities. When designing a business process the responsibilities are allocated to business roles, such as "Bank", "Supplier", etc.

When the business process is realized with existing services, the responsibilities are transferred to the parties that host services. Those parties may perform the services themselves or they may forward them to third parties. This transformation may prevent the business process to "see" what parties actually executed these services.

Examining the example in Figure 1, it may be seen that the business process defines the Supplier business role as being responsible for the activity "Process order". Itea has a long-term contract with Mio, for the main supplier. From the contract perspective, Itea does not mind if the furniture is actually supplied by a third-party. This means that the visibility of the organizational aspect for the "Process order" activity might be set to loss-full. Knowing that for some styles of furniture, the Mio service forwards requests to its partner-suppliers, (not visible to the technical process), the loss-full visibility of the Supplier role enables the alignment between the two processes.

Following the outlined, the requirement for visibility of the organizational aspect of a business process can be determined by using the following rule:

	Visibility of the organizational aspect
Q	Is it mandatory to capture the information on what business party executed a business process activity?
A	No - LFV; Yes - LLV; for some instances - CRV.

5 APPLYING VISIBILITY LEVELS

The use of the visibility levels affects both the design and implementation of executable processes. During design, the business designer applies the levels by labelling business process elements (activities, control flow, transactions, information concepts, roles, etc.) with a desired level of visibility. The technical process designer later on uses these labels in order to implement the process on top of existing services.

In this section we outline how the business- and the technical- process designer use visibility levels. Our intention is to provide an overview of how the visibility levels affect the design process rather than providing a complete method description.

The business process designer starts with creating an "ordinary" business process model. This is done by analyzing the business requirements; neither the forthcoming visibility labelling, nor technical concerns affect this model.

As the next step, the **business process designer** must change mindset in order to apply the visibility levels. The visibility levels should reflect the need of the environment to monitor the process execution at runtime. As stated in Section 3, the process environment consists of external and internal actors that interact with the process. External actors are commonly depicted in the business process model, for example "Customer", "Supplier" and "Bank" in Figure 1(a). Internal actors are those within the organization that are interested in monitoring the process states. In order to view the process from the internal viewpoint it might be helpful for the designer to take the viewpoint of a process supervisor. A process supervisor is responsible for the execution of the process, that is the completion of the process cases (Aalst, 2002). The reason to take a process environmental view is to be able to pinpoint the process elements that do not need to be visible at runtime even thought these elements are being executed. Guided by the questions from Section 4, the designer applies the loss-full visibility (LFV), or constrained visibility (CRV) labels to the elements that do not need full visibility during execution. Elements from all five process aspects are labelled as following:

- Functional – Activities and sub-processes
- Behavioural – Branching and ordering constructs
- Informational – Information concepts
- Transactional – Transactions boundaries
- Organizational – Roles/organizations that are participating in the process

Figure 3(a) depicts a labelled excerpt from the beginning of the example case in Section 2. In this case, the designer decides to label the first activity as LFV, because the start and completion of the activity "Get customer contact" does not have to be monitored at runtime. The "Get order history" activity is labelled as CRV because there is a desire to get a notification whenever a "gold" customer with an annual order history exceeding 5000€ places an order. In this case, a note is placed beside the CRV label to indicate for which instances lossless visibility is needed.

At the end, the business process designer gets a business process with elements labelled with the desired visibility, from a process supervisor's point of view.

When the business process is labelled, it is up to the **technical process designer** to construct an executable process that adheres to the business process design and that utilizes existing services. Ideally, the business process can be implemented as-is, with no changes applied. However, existing services might not allow the implementation of the lossless visibility for all constructs in the process. For example, certain information concepts might be hidden inside old legacy systems, and therefore be unavailable to the technical process. Since the business designer has labelled the elements with their desired run-time visibility, the technical process designer has obtained flexibility for designing the technical process. Unmarked (loss-less visibility) elements must still be implemented as-is, but the elements labelled with LFV or CRV can be implemented by applying "black-boxing" and selective black-boxing:

- *Black-boxing* can be applied when an element is labeled with the loss-full visibility (LFV). For example, information concepts and behavioral branching can be hidden inside legacy services.
- *Selective black-boxing* can be applied where lossless visibility is needed for some instances; this is applicable for elements marked as CRV (constrained visible).

Figure 3(b) depicts how black-boxing is applied to two activities - they are simply implemented as one in the technical process. Note that applying

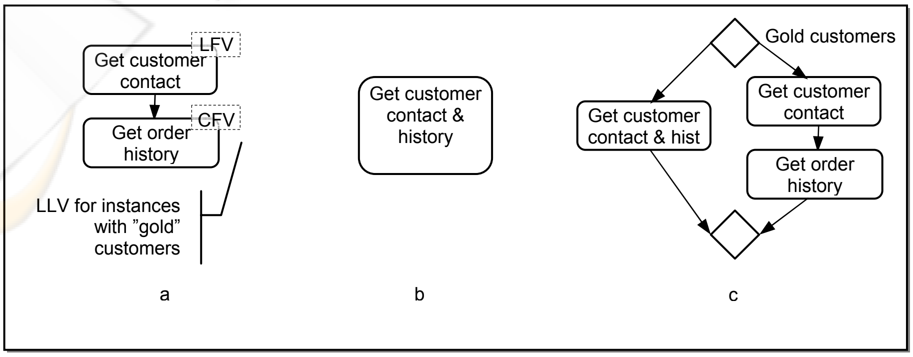

Figure 3: Labelling business process activities with levels of visibility (a); implementing the business process activities in the technical process according to existing services (b and c).

black-boxing in this way violates the business process design, since some instances (those with gold customers) need full visibility. However, if both the activities in figure 3(a) would be labelled as LFV, this would be a valid construct.

Figure 3(c) depicts how selective black-boxing is applied. In this case two branches are introduced, one that handles "gold" customers and one that handles the other customers. The branches are implemented by using different services that represent two existing solutions.

The above basic steps outline how the business- and the technical- process designer apply the visibility levels to achieve alignment between business and technical processes. It must be stated that the goal of the technical designer is to keep maximum visibility (LLV); the lower levels of visibility are considered when it is of great cost to change existing services. The benefit of striving towards high visibility in the technical realization is to keep important flow logic inside the technical process, rather than scattering it across services.

6 CONCLUSION

In this paper, we have proposed an approach for flexible alignment between business processes and their technical realizations in the environment of existing services. The approach is based on the notion of visibility. The use of the notion of visibility enables a process designer to distinguish states in the business process that must be captured (i.e. visible) in the final technical process. By studying the notion, we have defined three levels of visibility, where each determines a degree of process flexibility: loss-full, constrained and lossless. Based on a process description framework grounded on five main design aspects, we have then defined a set of rules for discerning minimal level of visibility that might be set when designing business processes.

Our concept of flexibility enables a relaxation of requirements for alignment of a business process with its technical process, by selecting flexible process elements with an adequate level of visibility. In this way defined, the concept of visibility facilitates a process realization where existing services might implement a process without enabling the business to monitor every single process state. From the evolution perspective, the notion of visibility gives ability to the business process designer to assess the design of a process to abstract (i.e. loose) the parts that need not to be captured in the final technical process; for the technical process designer, the notion of visibility guides needed refinements of existing services.

REFERENCES

Aalst W.M.P., Hee, K., 2002. *Workflow Management Models, Methods and Systems*. The MIT Press.

Aalst W.M.P., 1999. Flexible Workflow Management Systems: An Approach Based on Generic Process Models. In *DEXA'99*. LNCS 1677, pp. 186-195.

BEA, IBM, Microsoft, SAP and Siebel, 2003. Business Process Execution Language for Web Services. In http:*//www.ibm.com/developerworks/library/ws-bpel*

Bernstein, P., Hadzilacos, V., Goodman, N., 1987. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley.

Chopra A., Singh M., 2004. Commitments for Flexible Business Processes, In *AAMAS'04*. IEEE Computer Society 2004.

Garcia-Molina, H., 1991. *Modelling Long-Running Activities as Sagas*. IEEE Data Engineering Bulletin, Vol. 14/1, 1991, 14–18.

Heinl P., Horn S., Jablonski S., Neeb J., Stein K., Teschke M., 1999. A Comprehensive Approach to Flexibility in Workflow Management Systems. In *WACC '99.* ACM WACC proceedings, pp. 79-88.

Henkel, M., Zdravkovic, J., Johannesson, P., 2004. Service-Based Processes – Design for Business and Technology. In *ICSOC'04*. ACM Press, 2004.

Itea portal, 2005. In *http://itea.dsv.su.se*

Jablonski, S., 1998. A Software Architecture for Workflow Management Systems. In *DEXA'98*. IEEE Computer Society, 1998, 739-744.

Joeris, G., and Herzog O., 1999. Towards Flexible and High-Level Modeling and Enacting of Processes, In *CAiSE'99*. LNCS 1626, pp. 88-102.

Rausch-Scott, S., 1997. TriGSflow – Workflow Management Based on Object-Oriented Database Systems and Extended Transaction Mechanisms. In *PhD Thesis, University at Linz*.

Ribó J.M., Franch X., 2001. Building Expressive and Flex. Process Models Using UML-Based Approach. In *EWSPT'01*. LNCS 2077, pp. 152–172.

Sadiq, Sh., Sadiq, W., Orlowska, M., 2001. Pockets of Flexibility in Workflow Specifications. In *ER2001*. LNCS 2224, pp. 513-526.

SERVIAM Project , 2005. In *http://www.serviam.se*

White, S., 2004. Business Process Modeling Notation 1.0, (BPMN). In *http://www.bpmi.org*. Business Management Initiative.

Zdravkovic J., Henkel M., Johannesson P., 2005. *Moving from Business to Technology with Service-Based Processes*. IEEE Magazine of Internet Computing, Vol. 9/3, May/June 2005.