Exploring the Business Process Agility Issue: An experience report

Abstract

Business process agility has drawn the attention of numerous researchers. Whilst this research activity constitutes a useful contribution towards the attainment of business process agility, most of them focus on agility during execution phase. Therefore, although business process design is an equally important phase of the business process lifecycle the exploration of agility from the designer's perspective has not been given the attention it deserves. In this paper, we discuss our point of view regarding business process agility, as it was shaped during a case study concerning medical processes. Through this study, we identified important requirements for the attainment of business process agility, which were subsequently combined into a holistic picture constituting a comprehensive suggestion for the practical realization of business process agility. The objective of this paper is to contribute to a deeper understanding of business process agility and ultimately to its practical realization through the proposed holistic solution.

.

Keywords: business management, healthcare, business process agility, business process modeling, agility requirements

INTRODUCTION

Business process agility (or flexibility) has been a matter of interest for numerous researchers (Milanovic et al., 2011; Van der Aalst et al., 2009; Snowdon et al., 2007; Pesic et al., 2007; Daoudi & Nurcan, 2007; Reijers, 2006; ShuiGuang et al., 2004; Rinderle et al. 2004; Mangan & Sadiq, 2002; Millie & Balasubramanian, 1997). Agility in the context of business processes can be defined as the ability of an organization to effect changes in the process components (activities, roles, resources, information etc.) in a timely manner, usually in response to changes in business environment and stakeholders' needs (Alexopoulou et al., 2008). The intense interest on business process agility stems from the fact that business process automation supported by the utilization of process-aware information systems (Dumas et al., 2005) has increased accuracy and efficiency in process execution on one hand, but it has also rendered business process modification a complex and time-consuming task. This is because well-structured business process models executed by Business Process Management Systems (BPMS) (Dumas et al., 2005) proved to be inflexible to change. Since modern enterprises operate in highly turbulent environments having to cope with a frenetic pace of change (Oosterhout et al., 2006) and continuously sense opportunities for competitive action in their product-market spaces, it is business process agility, which underlies enterprises' success in constantly enhancing and redefining their value creation in highly dynamic environments (Sambamurthy et al., 2003).

In an effort to make business process agility true, researchers propose various methods, techniques or approaches in general, focusing on business process automation. One paradigm for the creation of agile business processes is based on the definition of abstract or loose models (Lin & Orlowska, 2005; Herrmann 2000) as opposed to strict, well-structured ones. Van der Aalst (1999), for example, proposes a general model from which several variations may be produced during execution through inheritance, which constitute specific models. ShuiGuang et al. (2004) suggest a modeling method according to which a business process model includes, apart from concrete parts of activity flows, some unspecified parts whose structure is formed during execution time, based on current conditions and by picking the appropriate activity from a 'pool' of activities. The same idea lies behind 'flexibility pockets' (Shazia et al., 2001) that represent activities that are orchestrated into specific flows during execution time. Other researchers contribute in the solution of the business process agility issue through the formal definition of modification rules (Casati et al., 1996; Weske, 2001; Clarence et al., 1995). $\text{ADEPT}_{\text{flex}}$ (Dadam & Reichert, 2009) is an endeavour based on this logic. It represents a set of operations, which enable the modification of business processes under execution, ensuring that structural correctness and consistency of the running instances is retained. Other approaches adopt formalisms based on the specification of constraints (Sadiq et al., 2005). Such an approach, for example, is proposed by Dourish et al. (1996), according to which instead of following a predetermined order of activities for the execution of a business process, actors are free to choose the activity they want to execute, provided that they do not violate specific constraints.

Whilst this research activity constitutes a useful contribution towards the attainment of business process agility, most of the proposed approaches focus on agility during execution phase, which concerns dynamic modifications of active business process instances. Therefore, although business process design is an equally important phase of the business process lifecycle (Weske, 2007), the exploration of agility in the design phase, associated with the ability of the process designer to easily and effectively describe business process modifications, has not been given adequate attention.

In this paper, we discuss our experience pursuing business process agility stemming from the agile automation of medical business processes in a Greek hospital. Based on this case study, we identified important requirements for the attainment of business process agility, which are analytically presented in the paper and concern both business process design and execution. These requirements are combined into a holistic picture constituting a comprehensive suggestion for the practical realization of business process agility. Through these requirements and the overall conclusions drawn from the case study, both researchers and practitioners can be guided in developing or choosing the appropriate modelling approach that can satisfy their requirements for business process agility. Overall, as business process agility is a polymorphous concept, the objective of this paper is to present our point of view on agile business process models and contribute to its deeper understanding and ultimately to its practical realization through the proposed holistic solution.

The paper is organized as follows. Section 2 aims at familiarizing the reader with the basic concepts of business process modeling. Our experience report is analytically presented in section 3. Section 4 summarizes our point of view towards the attainment of business process agility. The last section of the paper gathers our concluding remarks and identifies issues for further research.

BASIC CONCEPTS OF BUSINESS PROCESS MODELING

A business process constitutes an organized representation of some specific functionality of the enterprise. Business processes are identified through the analysis of the organizational environment. Then they are illustrated through explicit models usually expressed in a graphical notation. These constitute the analysis and design phases of the business process lifecycle (Weske, 2007), as illustrated in Figure 1. The configuration phase concerns the implementation of the process. Business process implementation can be interpreted in two ways. If no BPMS is used, the implementation involves the specification of procedures that should be accomplished by the personnel, following specific rules and policies imposed by the company administration. On the other hand, if a BPMS is employed, execution of business processes will be automated, so the processes are configured according to the requirements imposed by the type of BPMS infrastructure. The automated execution of a business process means that the coordination of the involved employees and the distribution of work will be controlled by a software system. As explained in the introduction, business process agility has turned into a hot issue with the evolution of business process automation. Thus, this paper concerns the automated implementation of business processes using a BPMS. After the implementation, business processes are tested and validated and the configuration phase ends. Then, the actual business process instances are enacted. The enactment phase concerns the run time of business processes. The BPMS executes the business process instances according to the respective business process models. The BPMS needs to provide for a correct process orchestration, guaranteeing that the process activities are performed according to the execution constraints specified in the process model. Lastly, based on execution logs and using monitoring techniques, business processes can be evaluated. The evaluation of a business process may reveal the necessity to change the business process model or even lead to its total redesign.

Typically, during the design phase, the Process Designer creates the business process models using a Business Process Modeling Tool. Models are usually created using graphical languages. The graphical model is subsequently transformed into an executable model expressed in an executable language, so that it can be executed by the engine of a BPMS (BPM Engine). The executable model includes the technical details required for the execution of a business process by a specific BPMS. Alternatively, if there is a graphical notation for the executable model, modeling during the design phase can be performed using this notation. Therefore, model transformation is not required in such a case. In general, the transformation is necessary when the graphical and the executable model are not based in the same metamodel. A metamodel is itself a model which defines the language used to create a model (OMG, 2001).

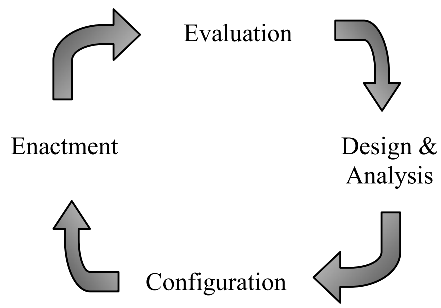


Figure 1. Business Process Lifecycle

A plethora of business process modelling approaches has been proposed in the literature (Scheer 1999; Muller et al., 2006; Balabko et al., 2004). Most business process modeling approaches are activity-driven (Scheer 1999) and concern well-structured business processes. The reason is that well-structured processes (e.g. manufacturing processes) were the first to be automated as they have well-defined steps. Basically, the objective of activity-driven business process modeling is to identify the activities of a specific functionality context and combine them appropriately in a process graph so that a goal is satisfied. In that sense, emphasis is laid on *how*. The activity-driven paradigm is eligible for cases where the actors should be enforced to follow a specific flow of steps. While activity-driven modeling is characterized by a complete and rigid process logic, in data-driven modeling, process logic is more loosely and partially defined. This is why data-driven modeling is more suitable for cases where the business process graph is extremely complicated and thus cannot be easily depicted. Such complexity may stem, for example, from multiple nested conditions or multiple reverses to the same prior actions causing a chaotic structure. Data-driven approaches (Muller et al., 2006) focus on identifying the data entities managed within a specific functionality context, i.e. emphasis is given on *what*, not on the specific process steps followed. When organizational functionality is modeled, there may be cases that an activity is not initiated due to data modifications or because an activity sequence has to be followed. Rather it is initiated because something happened that needs to be handled somehow. In such cases, the conditions under which an activity should be initiated can be expressed in a more abstract manner through events. Such conditions may arise from data modifications, human decisions, timing states or anything that could lead to a situation that should be handled, which can even be of an unknown source. Anything that happens signifies an *event*. An event denotes *when* (not necessarily in terms of time) an activity should be initiated. Events have been mainly used in ECA model (Dayal et al., 1990). The event-driven paradigm inherently supports the description of processes that are affected by unpredicted contingencies. Since contingencies are unpredicted events, following an event-driven approach would better facilitate the incorporation of the new events in the current model. While not common, there are also role-driven modeling approaches (Balabko et al., 2004). In role-driven approaches, modeling usually focuses on the identification of roles i.e. actor categories, involved in a specific functionality context. In that sense, emphasis is laid on *who*. Role-driven approaches focus on specifying interactions between roles. As such, they can be suitable for modeling communication-based processes, e.g. B2B process.

The modifications in business process models can be static or dynamic (Casati et al., 1996). Static changes concern model modifications during the design phase. If there are active instances of the modified model, these instances, depending on the policies and decisions of the company, can be aborted, flushed (i.e. completed following the previous version of the model) or adjusted so that their execution can be continued based on the modified model. The third case corresponds to dynamic modifications. Dynamic modifications include also the case of adjusting a specific instance due to special conditions without altering the respective model (ad hoc changes). Obviously, dynamic changes constitute a greater challenge than static. However, taking into account that organizational functionality is often described through large-scale business processes of high complexity, static changes are not a trivial task and as such they should be equally emphasized for a spherical approach to business process agility.

EXPERIENCE REPORT

Typical examples of organizations having strong agility requirements can be identified in the medical arena (Müller and Rahm, 1999; Mulyar et al., 2007). Medical processes cannot be easily modeled through the conventional, well-structured activity-driven models, as they are affected to a great extent by human decisions, which are often made up in an ad hoc manner based on frequently arising contingencies. While there are research endeavors (Lenz and Reichert, 2007; Boxwala et al., 2004) targeting the automation of medical processes, the BPMS technology has not been yet broadly established in hospitals.

Towards investigating agile business process automation of medical processes, we got engaged in a project in which we cooperated with a central Greek University hospital, where we interviewed both medical and administrative personnel in order to acquire information about their everyday tasks and understand their real needs in respect to business process automation. The specific hospital owns information systems for the provision of specific services such as patient administration and billing, laboratory examination management, etc., but workflow among the provided services is not supported. Our main focus during our study in this project was to identify a modeling approach that could ensure the agile execution of medical processes. Thus, based on the information gathered regarding the supported medical processes, we explored the modification of instances of medical process models and the models themselves. Some of the medical processes explored were the patient admission, the patient treatment and the procurement of consumables. In the following the discrete steps of our study are analytically discussed.

Identifying requirements towards business process agility

To ensure dynamic changes of business processes, we sought a way to describe them so that modifications during execution could be facilitated. Today, the typical way of executing a business process through a BPMS includes the transformation of the corresponding business process model as an integral unit (i.e. as a whole) into an executable format, followed by the execution of the process by the BPM engine using this format. This constitutes a stateful execution of the model as indicated in Figure 2. Stateful execution does not support the connection between the running instances and the corresponding model of a business process. However, agility is effectively facilitated if the modifications of a process model are reflected on its running instances at real time, in an effective and consistent manner. Therefore, we thought that what would be required is a BPMS created to support the parallel realization of design and execution phases, as opposed to current approaches, which impose a strict sequential fashion. Agility, in other words, would be promoted through the constant interaction between the design and execution phases and the corresponding tools/software environments, as illustrated in Figure 3. Such interaction presupposes a stateless execution of the BPM engine, which means that after the initiation of each activity, the BPM engine does not remain engaged in the business process instance in order to initiate the subsequent activities. In other words, stateless execution, where the BPM engine has no memory of previous and subsequent steps, may serve agility more effectively.

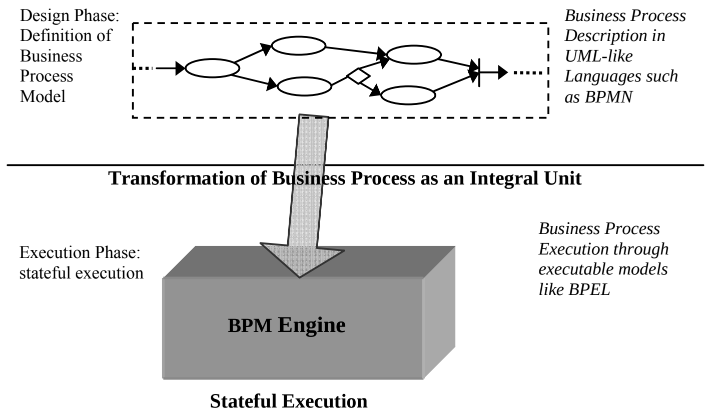


Figure 2. Typical approach in business process execution through a BPMS

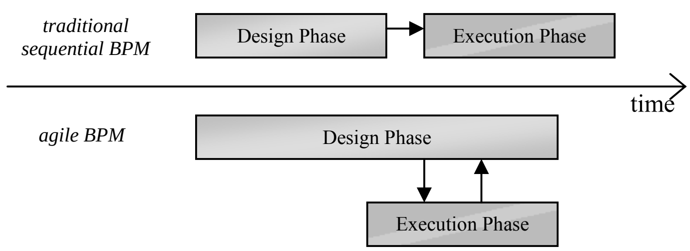


Figure 3: Agile vs. traditional BPM

Taking for granted the stateless execution of BPM engine, it is entailed that the generated *business process models should be modular* as shown in Figure 4. Modular models can facilitate interventions in business process steps during execution phase and thus enable the stateless execution of business processes. As such, alterations that concern reconfiguration and insertion/deletion of process steps at real time could be more easily implemented.

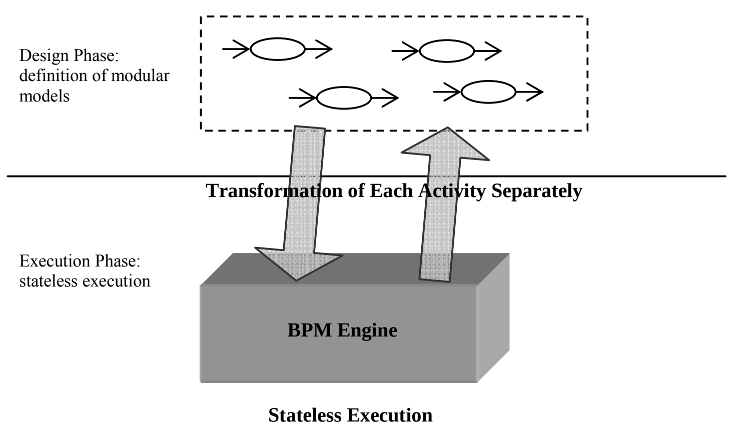


Figure 4: Interpreter-like execution of business processes through modular models

Coming up with a solution

Towards this direction we decided to design each of the activities involved in each process in an autonomous manner. This would produce modular models which would better serve dynamic modification.

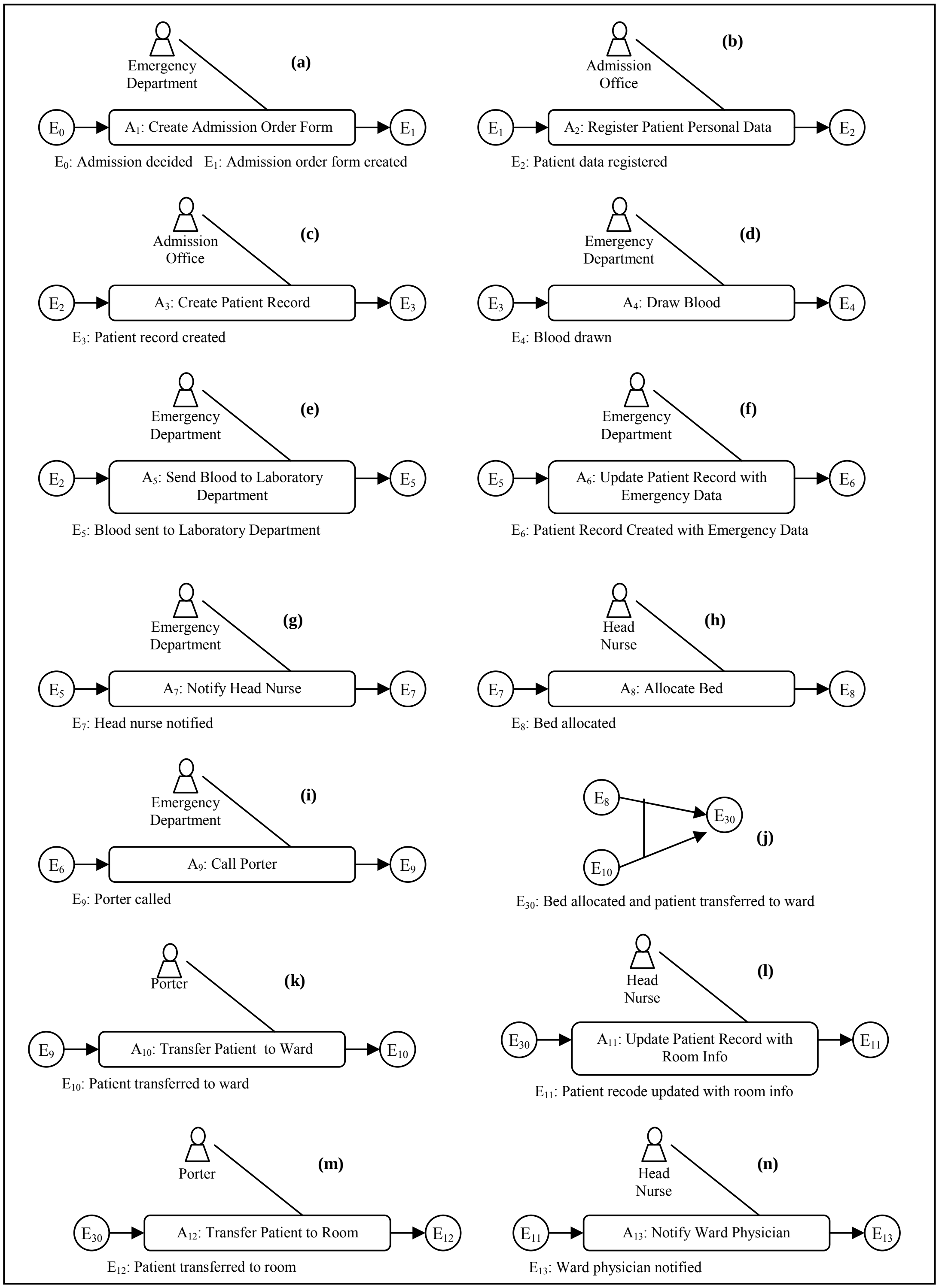


Figure 5: Modeling Patient Admission in a modular fashion

However, we had to find a way to depict in the modular models, either explicitly or implicitly, the information concerning the sequential relation between the activities, so that they could be executed in the right order at run time. To achieve this, the concept of events was adopted. More specifically, in order to achieve modularity, the sequence of activities was divided into autonomous parts consisting of autonomous activities accompanied by initiating and resulting events.

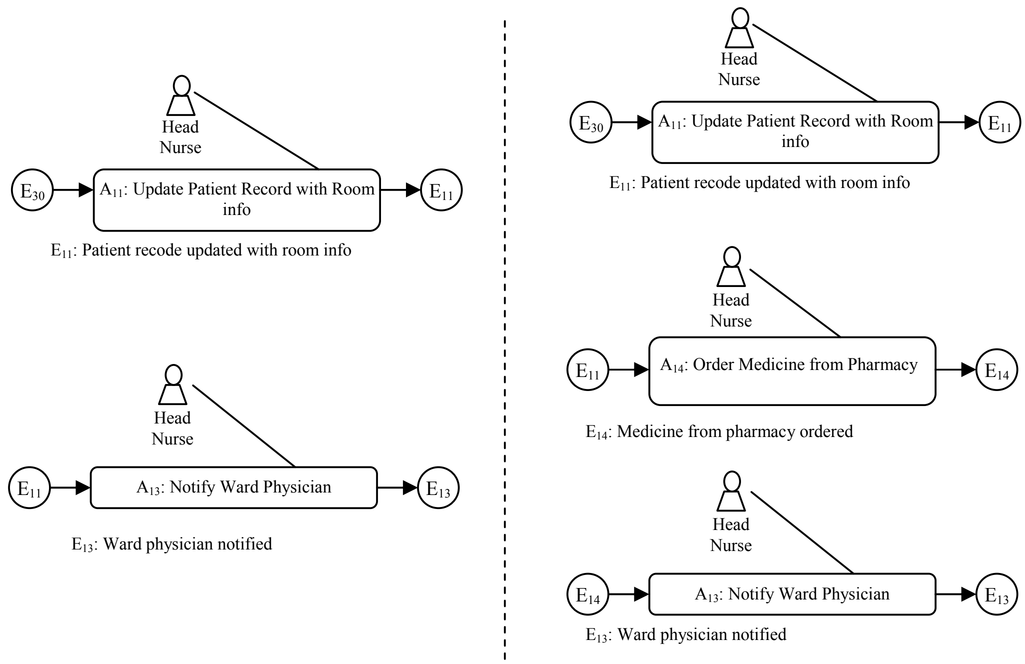
Figure 5 illustrates a modular process model for Patient Admission (Alexopoulou, et al., 2010). Patient admission is initiated whenever it is decided at the Emergency Department that a patient who has arrived at the hospital has a serious health problem and thus needs to be hospitalized. The first activity that takes place is the creation of an admission order form filled by the Emergency Department. This order form is transferred to the Admission Office where the patient personal data are registered. Based on these data, a patient record with a specific ID is created. Subsequently, blood is drawn from the patient, which is sent to the Laboratory Department. Additional examinations required as well as emergency history and any medication provided to the patient at the Emergency Department are registered in the patient record. Afterwards, a porter is notified so that the patient is transferred to the ward. In parallel, the head nurse of the ward is notified that the specific patient is going to be transferred there. The head nurse allocates a bed to the patient and the porter transfers the patient to the bed allocated to him/her. Also, the head nurse updates the patient record with the corresponding room and bed number and notifies the ward physician that a new patient has been admitted to the ward.

The activities depicted in Figure 5 are regarded autonomous, i.e. they do not belong to a business process sequence. Each activity is initiated by a specific event and also leads to the generation of new events upon its completion. Such events as well as event combinations conceal the information regarding the sequential relation between the activities. Let us discuss the case of activity "A_{11}: Update Patient Record with Room Information" (Figure 5 (l)). This activity is initiated by the event E_{30}. E_{30} is a complex event (Luckham, 2002) which is produced by the conjunction of events E_8 and E_{10} (Figure 5 (j)). Essentially, this means that activity A_{11} is initiated when a bed has been allocated to the patient (event "E_8: Bed allocated") and the patient has been transferred to the ward (event "E_{10}: Patient transferred to ward"). Activity A_{11} generates the event "E_{11}: Patient record updated with room information" upon its completion. This event is the one initiating activity "A_{13}: Notify Ward Physician" (Figure 5 (n)), which, in practice, is the subsequent activity of A_{11} as indicated in Figure 5.

An appropriate event-driven process engine such as that proposed in (Geppert & Tombros, 1996) may ensure the execution of autonomously designed activities in the proper order, which is implicitly specified through the events and event combinations defined in the model. Essentially, a business process sequence will be generated on-the-fly during execution. As the engine will not be aware of a business process sequence, it will function in a stateless manner and hence modifications during run time will be facilitated. Therefore, the requirements for modular and stateless execution were satisfied.

Evaluating business process agility

Based on the developed approach, we evaluated the agility of the process admission model. We considered the case the patient admission process needed to be modified by adding another step right after activity "A_{11}: Update Patient Record with Room Info" that would concern ordering the required medicine from the hospital pharmacy department, named "A_{14}: Order Medicine from Pharmacy". This addition required appropriate event modifications. In specific, event "E_{11}: Patient record updated with room info" had to initiate the new activity A_{14} and the new event "E_{14}: Medicine ordered" had to replace event E_{11} in the definition of activity "A_{13}: Notify Ward Physician", as indicated in Figure 6.



(a) before the addition of activity A_{14} **(b)** after the addition of activity A_{14}

Figure 6: Adding activity A_{14}

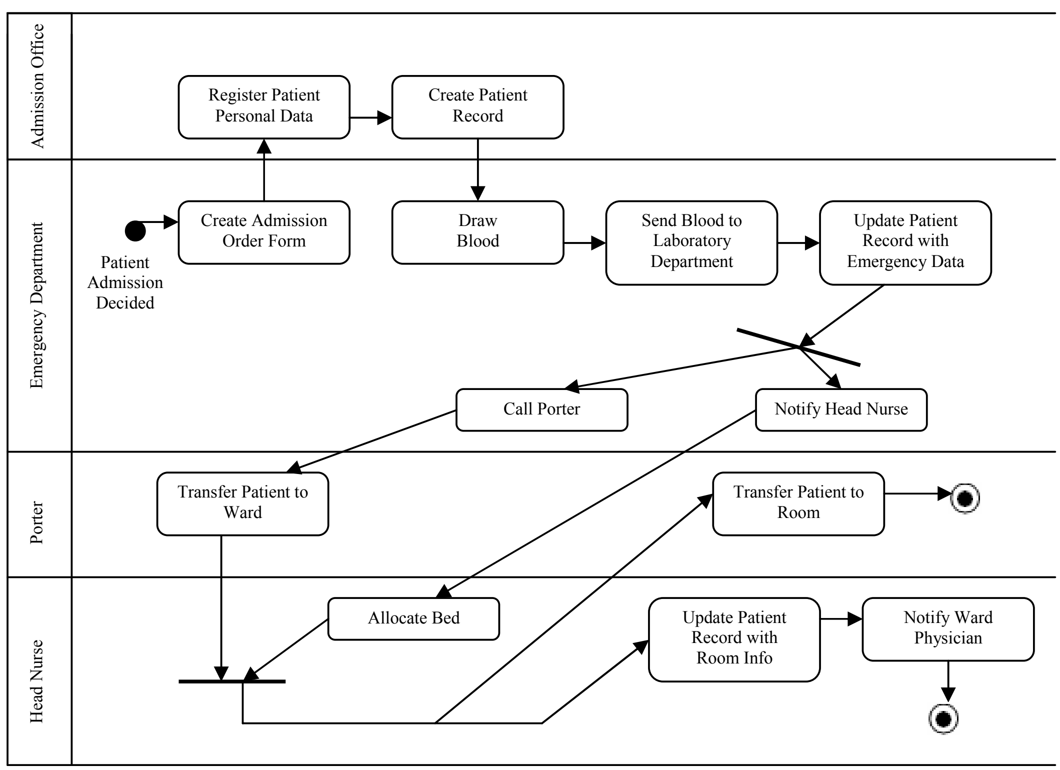


Figure 7: The Patient admission process designed using a typical activity graph

These changes can take place for the active instances of the patient admission process. If for example the engine is executing at a specific moment the event "E_9: Porter called" (initiating activity "A_{10}: Transfer Patient to Ward") for a particular instance, it will continue later with the execution of the new event E_{14} without being aware that this is a newly added event which did not exist in the model when the execution of the specific instance started. However, while dynamic modifications are indeed facilitated, changing the model as described above with the addition of activity "A_{14}: Order Medicine from Pharmacy" does not seem to be straightforward for the business process designer. The designer, when trying to update the events and perhaps the event combinations, has in mind the implied sequence that needs to be ensured and since he/she is not able to view a visual representation of this sequence, the model modification turns into an arduous procedure, especially if there are complicated event combinations that need to be updated. It follows that a typical activity-driven diagram, such as that presented in Figure 7, can better serve the business process designer not only in describing the business process but also in performing modifications (i.e. adding or deleting activities), since the sequence is there and does not need to be deduced. As such, the modifications of this patient admission model can be performed by the designer with greater ease and rapidity. Consequently, the model of Figure 7, although not modular, seems to be more agile from the designer's perspective than the model of Figure 5, so what went wrong?

Lesson learned

The basic mistake was that we did not adequately emphasize the designer's view. Rather, we merely focused on agile business process execution. However, if the business process modeling approach adopted does not ensure an efficient and easy way for the designer to modify the business process model, the overall business process agility is hindered. Thus, we realized that business process agility should be analyzed from two separate views: both execution and design. This means that business process agility, as a concept, comprises two equally important constituents: design agility and execution agility. While one may intuitively feel familiar with this conclusion, it is evident, however, in the literature that design agility has not been given the attention it deserves, as also happened in our case. This stems from the fact that researchers often propose a modelling approach having in mind the execution phase, i.e. they focus on the achievement of flexible process instance adaptations rather than the effort required for the modification of a model at design time (ShuiGuang et al., 2004; Dadam & Reichert, 2009). But how can design agility be interpreted?

Understanding the implications of business design process agility

In the project we were working for, we had actually not taken into consideration the nature of the patient admission process. As a result, although execution agility was facilitated, the agility of the business process model itself from the designer's perspective was considerably reduced. Apparently, patient admission is a well-structured process with a clear sequence. Why then use another type of model that is not based on a typical activity-driven diagram, since a typical activity-like diagram (using for example BPMN (OMG, 2011)) would be most efficient for the illustration of such a process? Besides, although the alternative model presented in Figure 5 seems to be based on the event-driven paradigm, during its creation at the design phase we were actually thinking in an activity-driven fashion, since our thought was not driven by events but by the implied order of activities. As such, it is the predetermined implied sequence of activities that indicates the way the events should be updated, so, in practice, the modeling approach is not driven by events but rather by activities. This proves that the nature of a business process cannot be violated. As such, for the attainment of business process agility, *the modeling approach should be in harmonization with the nature of the actual business process to be modeled.* This requirement implies that a business process model can be characterized agile only in relation to a specific business process or a process type. Thus, when a modeling approach is proposed, the type of business processes it is appropriate for should also be specified. However, the intimate relation between design business process agility and business process nature has not been clearly stated in the literature. As such, there are modelling approaches such that proposed in (Saidani & Nurcan, 2006) aiming at business process agility for which it is not clearly stated what kind of business processes they are suitable for.

Apart from the patient admission process, we also examined patient treatment. From the lesson learned through patient admission, the first step towards specifying an agile modelling approach for this process, was to closely examine its nature. Patient treatment, as opposed to patient admission, is not well-structured. While there may be cases of activities carried out in a regular fashion, most often, they are performed whenever required, following human decision. Clinical examinations, for example, are performed every morning as well as whenever an unexpected symptom occurs. Thus, a typical activity-driven model for the patient treatment process would be characterised by multiple reverses to prior activities and ambiguities regarding the flow. It follows that creating a clear and comprehensive activity diagram for patient treatment is practically unfeasible. As opposed to the activity-driven model, the event-driven approach that we developed (Alexopoulou et al., 2009) aligned with the nature of the patient treatment process. In the event-driven model, no process flow is defined. Instead, for each event identified, the corresponding invoked action is specified, which in case of patient treatment is usually data registration or participant notification. Also unpredicted conditions that often arise during a patient's treatment (e.g. a intense chest pain, sudden temperature increase, etc.) were inherently modeled through events. Therefore, patient treatment was effectively designed using the event-driven modeling paradigm.

ON DEVELOPING A HOLISTIC APPROACH FOR THE ATTAINMENT OF BUSINESS PROCESS AGILITY

As revealed by our experience, for agility to be attained both design and execution phases should be equally emphasized. As opposed to the majority of the current research approaches, which restrict agility exploration to a technological framework, business process agility should also be examined beyond the boundaries of IT infrastructure. The technology-neutral dimension in the exploration of business process agility is necessary to unearth requirements that have to do with the process itself, independently of the underlying technology. Therefore, we suggest analyzing design business process agility separately from execution agility.

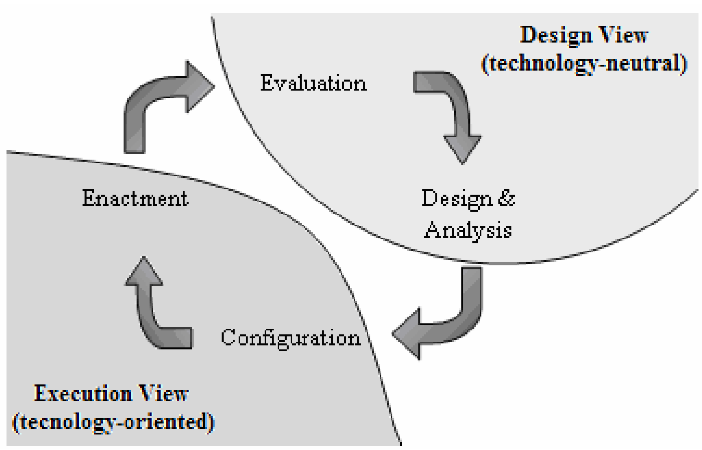


Figure 8. Mapping design and execution views to business process lifecycle phases

As indicated in Figure 8, enactment and configuration correspond to the execution view, since automated execution of business processes entails that these phases are directly related to the IT infrastructure. On the other hand, design and evaluation concern the business process itself and even if they are supported by a software tool, they do not consider the means that will be used for business process execution. As such, they are associated with the design view, which is regarded technology-neutral.

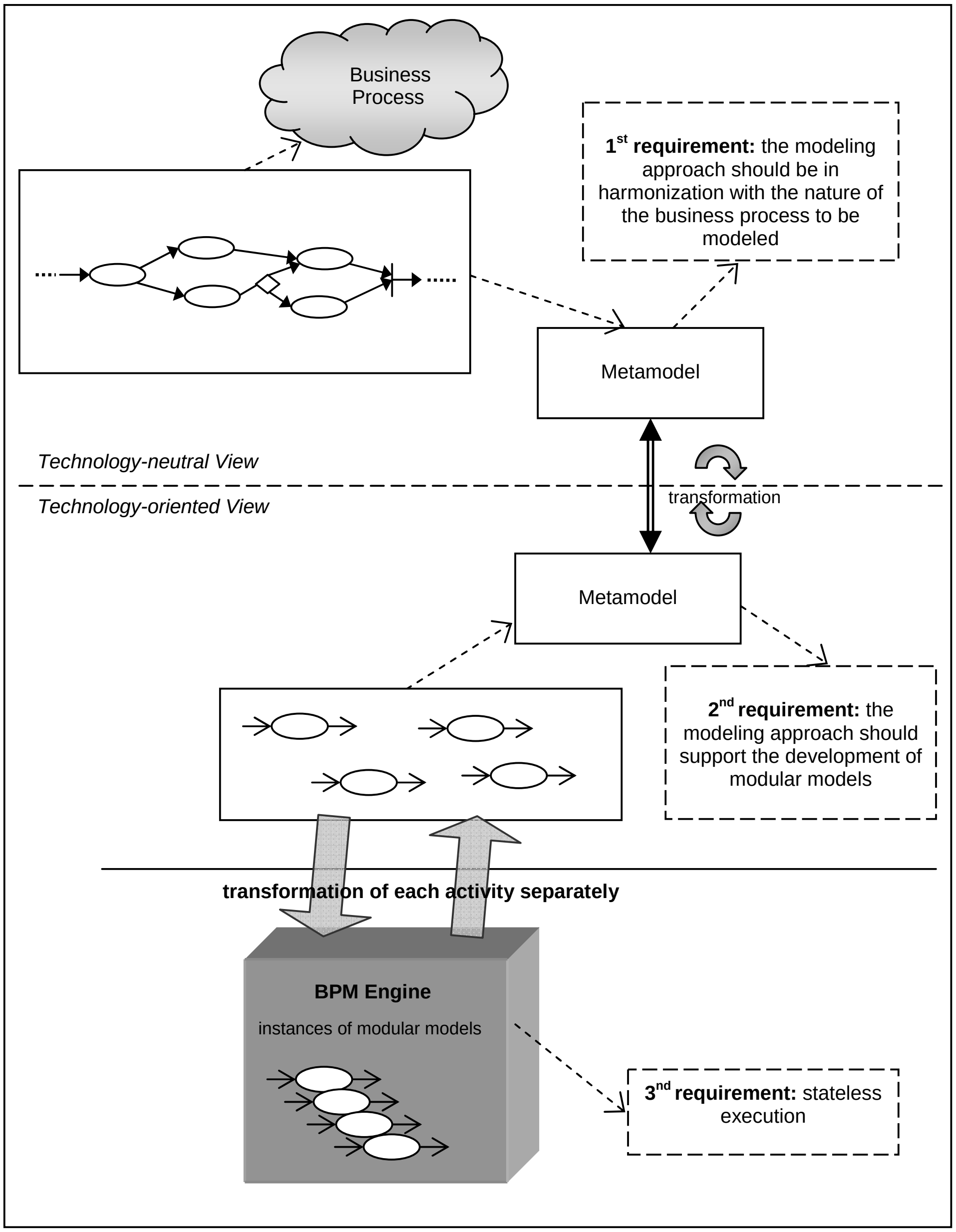


Figure 9: The integrated picture regarding the attainment of business process agility

The requirements for achieving business process agility that were confirmed through our study conducted under the project aiming at the automation of agile medical processes are as follows:

- *Design view*
  - the modeling approach should be in harmonization with the nature of the actual business process to be modelled

- *Execution view*
  - the business process models executed by the BPM engine should be modular
  - the BPM engine should function in a stateless manner

How are these requirements interpreted in practice? The first one indicates, as already stated, that when a specific business process is modeled, its nature should be carefully examined so that a suitable modeling paradigm is selected. The selection of a suitable modeling paradigm means that an appropriate metamodel is adopted (upper part of Figure 9). Taking into account this requirement combined with the fact that enterprises own usually multiple business processes, which are of diverse nature, it is entailed that an enterprise should adopt multiple business process metamodels in order to achieve business process agility. The other two requirements indicate how the business process should be executed so that agility is ensured. The main point of these two requirements is that the models executed by the BPM engine should be generated based on a metamodel that supports modularity (lower part of Figure 9).

Overall, the identified requirements indicate that in order to attain business process agility, an enterprise should adopt multiple modeling paradigms to facilitate the description of all supported business process types and transform these models into modular executable ones in order to achieve agile execution of business processes, as illustrated in Figure 10. Referring again to the patient admission process, the activity graph of Figure 7 may be transformed into a modular model, such as the one of Figure 5 during execution. Alternatively, other techniques can be used such as that proposed in (Zakir & Sanjay, 2006), where ECA (Event-Condition-Action) rules (Dayal et al., 1990) are used for the dynamic generation of BPEL – a well-established standard for the automated execution of business processes (OASIS, 2007) - code at run time.

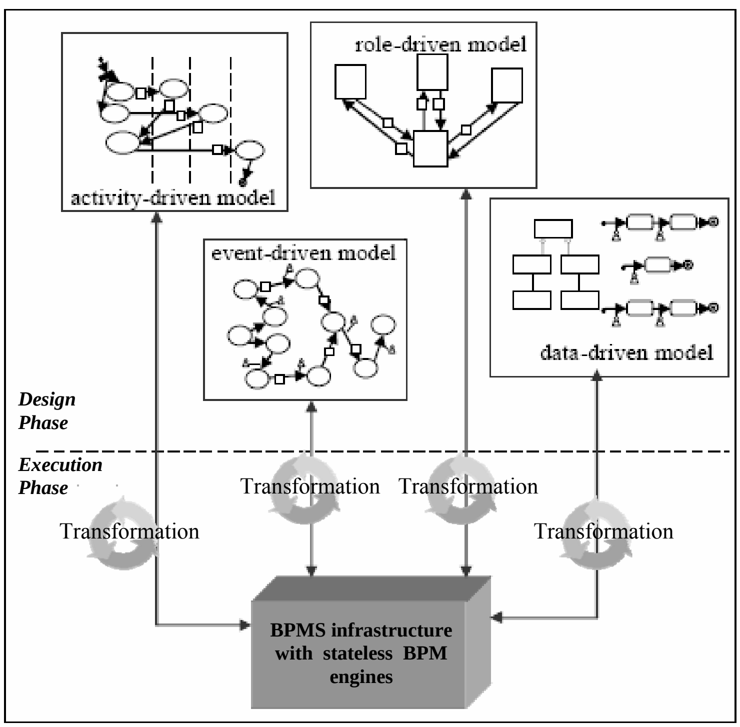


Figure 10: A BPMS executing multiple business process models

It should be stressed that implementing a BPMS infrastructure that supports multiple business process models is not a trivial task. The issues associated with the implementation of such a BPMS need to be deeply explored so that the best alternative can be unearthed. A basic dilemma, for example, is whether it is more efficient to implement a BPMS that comprises multiple interacting BPM engines, as many as the discrete business process models, or it is better to transform the discrete models into a common one at enactment level, which will then be executed by a single BPM engine.

Currently, enterprises usually describe their processes using the modeling language offered by the BPMS tool they possess, which usually follows the BPEL standard. Hence, if a company uses, for example, IBM Websphere (IBM, 2006), they describe their processes according to the modeling approach supported by the specific tool, which constitutes a proprietary representation of BPEL. As a result, alternative business process models are not supported and thus business process agility is not promoted. BPEL in its current form does not suffice for the attainment of business process agility. Besides, it has been recognized that BPEL is not flexible (Marlon et al., 2005). For agility to be ensured, BPEL should support modularity. Also, its semantics should be extended to support alternative business process modeling paradigms. Otherwise, more standards should be developed to cover the need for automating business processes of diverse modeling paradigms.

CONCLUSIONS

In this paper, we delineated how business process agility was explored through a research project we were engaged in, which aimed at the agile automation of medical business processes. This exploration led us to a number of important conclusions, which are listed in the following:

- agility implications are different for business process design and execution phases. Design agility should be equally emphasized with execution agility for the attainment of business process agility.
- for design agility to be achieved, it should be ensured that the modeling approach is harmonized with the nature of the actual business process to be modeled.
- for execution agility to be achieved the business process models executed by the BPM engine should be modular and the BPM engine should function in a stateless manner.
- in practice, design agility is a prerequisite of execution agility.
- business process agility requires design and execution phases to be more “interwoven” with each other.
- a model can be characterized agile only in relation to a specific business process. As such, when a modeling approach is proposed, the type of business processes it is appropriate for should also be specified.
- an agile enterprise should support multiple business process models ensuring, though, their integration and seamless intercommunication.
- the semantics of existing standards such as BPEL and BPMN should be extended to support alternative business process modeling paradigms. Otherwise, more standards should be developed to cover the need for automating business processes of diverse modeling paradigms.

Regarding our future work, we intend to further explore the existing modeling methods so as to identify additional modeling paradigms and the business process categories they are suitable for. Our ultimate target is to develop a model integration framework, adopting for this purpose the concepts of MDA (Model Driven Architecture) (OMG, 2003), to enable enterprises to efficiently support multiple business processes of discrete modeling paradigms.

References

Alexopoulou Nancy, Kanellis Panagiotis, Nikolaidou Mara, Martakos Drakoulis, (2008). “A Holistic Approach for Enterprise Agility”. Handbook of Research On Enterprise Systems, Idea Group Inc

Alexopoulou Nancy, Nikolaidou Mara, Anagnostopoulos Dimosthenis, Martakos Drakoulis. “An Event-Driven Modeling Approach for Dynamic Human-Intensive Business Processes”, edBPM Workshop 2009.

Alexopoulou Nancy, Nikolaidou Mara, Kanellis Panagiotis, Mantzana Vasiliki, Anagnostopoulos Dimosthenes, Martakos Drakoulis. “Infusing agility in Business Processes through an Event-Centric Approach”. International Journal for Business Information Systems, Special Issue on “Simulation and Business Modeling”, Vol 6., Issue 1, 2010.

Balabko Pavel, Wegmann Alain, Ruppen Alain, Clément Nicolas: The Value of Roles in Modeling Business Processes. BPMDS (2004).

Boxwala, A., M.Peleg, Tu, S., O.Oqunyemi, Zeng, Q., Wang, D., et al.: GLIF3: a representation format for sharable computer-interpretable clinical proactive guidelines. Biomedical Informatics 37(3) (2004) 147.161

Casati F., Ceri S., Pernici B., and Pozzi G.. Workflow Evolution. In *Proceedings of ER ’96*, pages 438–455, Cottubus, Germany, Oct 1996.

Clarence A. Ellis, Karim Keddara, Grzegorz Rozenberg: Dynamic change within workflow systems. COOCS 1995: 10-21.

Dadam P. and Reichert M. 2009. The ADEPT project: A decade of research and development for robust and flexible process support. Computer Science—Research and Development, 23(2):81–97.

Daoudi F. and Nurcan S. 2007. A benchmarking framework for methods to design flexible business processes. Software Process Improvement and Practice, 12:51–63.

Dayal U., Hsu M., and Ladin R.: Organizing Long-Running Activities with Triggers and Transactions. Proceedings of ACM International Conference on Management of Data, pp.204-214 (1990).

Dourish Paul, Holmes Jim, MacLean Allan, Marqvardsen Pernille and Zbyslaw Alex, 1996. “Freeflow: Mediating Between Representation and Action in Workflow Systems”. *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, CSCW

Dumas M., Aalst W., Hofstede A., Process-Aware Information Systems (John Wiley & Sons INC, 2005).

Geppert A. and Tombros D., Event-based Distributed Workflow Execution with EVE, Technical Report, 1996, Department of Computer Science, University of Zurich.

IBM. “Patterns: SOA Foundation Service Creation Scenario”. IBM. 2006, ibm.com/redbooks.

Lenz Richard and Reichert Manfred, IT support for healthcare processes - premises, challenges, perspectives in *Data and Knowledge Engineering*, 2007, 61 pp. 39-58.

Lin J.Y.C. and Orlowska M.E., (2005). Partial completion of activity in business process specification. Proceedings of IRMA 2005, San Diego, CA, USA, pp.186-189.

Luckham David: The Power of Events. Addison-Wesley (2002).

Mangan P. and Sadiq S.: On Building Workflow Models for Flexible Processes. 13th Australasian Database Conference (ADC2002), Melbourne, Australia. Vol. 5 (2002).

Marlon Dumas, Tore Fjellheim, Stephen Milliner and Julien Vayssiere. Event-based coordination of process oriented composite applications, BPM, 2005.

Milanovic, M.; Gasevic, D.; Rocha, L., 2011. Modeling Flexible Business Processes with Business Rule Patterns. Distributed Object Computing Conference (EDOC) 15th IEEE International, Belgrade, Serbia, Aug. 29 -Sept. 2, pp. 65 – 74.

Millie M. K. and Balasubramanian P. R.: Dynamic Workflow Management: A Framework for Modeling Workflows. Proceedings of 13th HICSS, IEEE (1997).

Muller Dominic, Reichert Manfred and Herbst Joachim: Flexibility of Data-Driven Process Structures**.** BPM Workshops, LNCS 4103, pp.181-192 (2006).

Müller R., Rahm E. (1999). Rule-Based Dynamic Modification of Workflows in a Medical Domain Proceedings of BTW99 Freiburg im Breisgau, 1.-3. March, Spinger, Berlin 1999: 429-448.

Mulyar Nataliya, Pesic Maja, Wil M. P. van der Aalst, Peleg Mor: Declarative and Procedural Approaches for Modelling Clinical Guidelines: Addressing Flexibility Issues. Business Process Management Workshops 2007: 335-346

OASIS, 2007. Web Services Business Process Execution Language Version 2.0

OMG. Meta Object Facility (MOF) Core Specification, OMG Available Specification, Version 2.0 , 06-01-2001.

OMG. MDA Guide Version 1.0.1. 12th June 2003

OMG, Business Process Management Notation. Version 2.0, (OMG, 2011).

Oosterhout van Marcel, Waarts Eric and Hillegersberg van Jos (2006). Change factors requiring agility and implications for IT. European Journal of Information Systems, 15, pp.132-145.

Pesic M. Schonenberg M. H., Sidorova N., and W.M.P. van der Aalst, 2007. Constraint-Based Workflow Models: Change Made Easy. In F. Curbera, F. Leymann, and M.Weske, editors, Proceedings of the OTM Conference on Cooperative information Systems (CoopIS 2007), volume 4803 of Lecture Notes in Computer Science, pages 77–94. Springer-Verlag.

Reijers H.A., 2006. Workflow flexibility: The forlorn promise. In 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006), 26-28 June, Manchester, United Kingdom, pages 271–272. IEEE Computer Society.

Rinderle S., Reichert M., and Dadam P.: On Dealing with Structural Conflicts between Process Type and Instance Changes. BPM, pp. 274–289 (2004).

Sadiq Shazia, Orlowska Maria, Sadiq Wasim: Specification and validation of process constraints for flexible workflows. Inf. Syst. 30(5): 349-378 (2005).

Saidani Oumaima and Nurcan Selmin: A Role-based Approach for Modeling Flexible Business Processes. BPMDS (2006).

Sambamurthy, V., Bharadwaj, A., Grover, V. (2003). Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms. MIS Quarterly, Vol. 27, No. 2, pp. 237-263.

Scheer A.W. : ARIS-Business Process Modeling'. 2nd ed. Berlin: Springer (1999).

Shazia W. Sadiq, Wasim Sadiq, Maria E. Orlowska: Pockets of Flexibility in Workflow Specification. ER 2001: 513-526

ShuiGuang D., Zhen Y., ZhaoHui W., LiCan H.: Enhancement of Workflow Flexibility by Composing Activities at Run-time. Proceedings of the ACM Symposium on Applied Computing, pp. 667-673 (2004).

Snowdon R.A., Warboys B.C., Greenwood R.M., Holland C.P., Kawalek P.J, and Shaw D.R. 2007. On the architecture and form of flexible process support. Software Process Improvement and Practice, 12:21–34

Van der Aalst. How to handle dynamic change and capture management information? An approach based on generic workflow models CoopIS 1999: 115-126

Van der Aalst WMP, Pesic M, Schonenberg H, 2009. Declarative workflows: Balancing between flexibility and support. Computer Science—Research and Development, 23(2):99–113.

Weske, M. (2001): Formal Foundation and Conceptual Design of Dynamic Adaptations in a Workflow Management System. Accepted for: Minitrack Internet and Workflow Automation: Technical and Managerial Issues. 34th Hawaii International Conference on System Sciences (HICSS-34)

Weske Mathias, 2007. "Business Process Management: Concepts, Languages, Architectures". Springer-Verlag.

Zakir Laliwala and Sanjay Chaudhary, 2006. Event-Driven Dynamic Web Services Composition: From Modeling to Implementation. Innovations in Information Technology Conference, IEEE. Dubai 2006, pp.1-5.