Evaluation of Paradigms Enabling Flexibility
BPMSs Comparative Study

Asma Mejri[1] and Sonia Ayachi Ghanouchi[2]
[1]*Laboratory RIADI-GDL, ENSI, Mannouba 2010, Tunisia*
[2]*High Institute on Mangament of Sousse, Sousse, Tunisia*

Keywords: Comparative Study, Constraint, Rule, Case Handling, Adaptive Process Support, Taxonomy, Flexibility, BPMS.

Abstract: In this paper, we make a comparative study between several paradigms that provide flexibility: constraint based, rule based, case handling and adaptive process support paradigms. We evaluate existing Business Process Management Systems (BPMSs) using the taxonomy of Regev et al. in order to assign a flexibility score to each of the corresponding paradigms.

1 INTRODUCTION

In the past decades, flexibility has gained a strong foothold in various fields, notably in the business management and information systems disciplines. Flexibility is important, because continuously changing conditions force organizations to rapidly and flexibly adapt their processes. Flexibility is defined as a key consideration of effective processes. It is their ability to deal with both foreseen and unforeseen changes in the context or environment in which they operate (Schonenberg et al., 2008).

Thus the real challenge for business process models consists in providing information systems with adequate information to deal with the often conflicting requirements of flexibility.

To overcome the limitations caused by traditional business process management paradigms, several paradigms have emerged. The most popular ones are: rule-based, constraint based, case handling and adaptive process support paradigms.

In this paper we deal with these four paradigms. We aim at comparing them using the taxonomy of flexibility in business process, which was suggested by Regev et al. in their paper (Regev et al., 2006). Regev et al. define business process flexibility as the capability of implementing changes in business process type and instances by changing only those parts that need to be changed and keeping other parts stable. Regev et al. (Regev et al., 2006) consider in his taxonomy.

The aim of this paper is to answer the following questions:

- How to compare the flexibility of existing BPMS?
- How to precise the weight of the different criteria?
- How to measure the ability of existing paradigms to deal with process change?

The remainder of this chapter is structured as follows: A review of BPMS is first presented in section 2 to provide insights into current BPMS. In section 3, we present the taxonomy of Regev et al.. Based on this taxonomy, we performed a comparison between BPMS, under several criteria. Then we compare the different paradigms. We conclude with a summary and outlook on future work.

2 A REVIEW OF EXISTING BPMS

In this section we present different BPMS that have a great impact in the business process management field. We provide a brief description of each of these BPMS. These BPMSs were selected because they are some of the most famous ones in the studied field of flexibility enabling BPMSs. Moreover, experts involved in the development/use of these BPMSs filled the proposed questionnaire which will be presented in section 4. Our study is in fact based

Mejri A. and Ayachi Ghanouchi S..
Evaluation of Paradigms Enabling Flexibility - BPMSs Comparative Study.
DOI: 10.5220/0005379102910298
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (ICEIS-2015), pages 291-298
ISBN: 978-989-758-098-7

on the analysis of the answers given by these experts to this questionnaire.

2.1 Declare

DECLARE system provides a broad range of functionalities ranging from design, enactment and dynamic change to verification, discovery and recommendation (Pesic et al., 2009).

DECLARE illustrates how declarative approaches can indeed be used to realize more flexible BPM solutions, while providing various types of support (Pesic et al., 2009).

The main functionalities of DECLARE system are creating the constraint templates, creating, executing and verifying constraint models and dynamically changing instances of constraint models.

Declare system is based on a declarative language "declare" that combines a formal semantic grounded in Linear Temporal Logic (LTL) on finite traces, with a graphical representation (Maggi et al., 2013).

2.2 ESProNA

ESProNa (Engine for Semantic Process Navigation) is a declarative business process modelling system.

ESProNa supports the definition of functional, behavioral, organizational, data and operational process perspectives, resulting in an expressive and flexible modeling language. It uses constraints for representing inter-process dependencies and constraint propagation for finding which processes are executable in user selected scenarios or given ones (Igler et al., 2010).

It has been developed using declarative programming, namely Prolog and Logtalk, to implement the different functionalities (Igler et al., 2010).

EsProNa was implemented using a Log talk application running on SWI-Prolog extended with the CLP (FD) constraint library and the N3 parser Henry (Igler et al., 2010).

2.3 JRules (IBM WebSphere ILOG JRules)

Jrules/JSolver is a business rule management system (BRMS). JRules offers an important set of components and capabilities to enable business users and developers to manage business rules directly with various levels of implication, from limited review to complete control over the specification, creation, testing, and deployment of business rules (Boyer and Mili, 2011).

JRules was initially developed by ILOG Corp, a software vendor founded in 1987, and acquired by IBM in 2009 (Boyer and Mili, 2011).

The JRules BRMS platform is a collection of modules that operate in different environments while working together to provide a comprehensive Business Rule Management System (BRMS). A BRMS helps to manage business rule independently of the business application. A BRMS enables business and IT to collaborate, author, manage, and execute business rules (Boyer and Mili, 2011).

JRules enables us to create different types of rule artifacts, depending on the complexity of the business logic, on the regularity of its structure, and on its specific use (Boyer and Mili, 2011).

2.4 Adept2

Based on a conceptual framework for dynamic process changes, on novel process support functions, and on advanced implementation concepts, the developed system enables the realization of adaptive process-aware information systems (Reichert et al., 2005)

The ADEPT2 system enables support for a broad spectrum of processes, ranging from simple document-centered workflows to complex production workflows, which integrate heterogeneous, distributed application components (Dadam et al., 2007). Thus it can be used in a variety of application domains.

The ADEPT2 technology has been transferred into an industrial-strength product and forms the technological base of the AristaFlow BPM Suite.

The most important goals of the ADEPT2 system were to provide the full spectrum of change operation for updating a process model, and to be able to migrate process instances (including those that were individually modified) to a new model version (Martinho, 2010).

2.5 PHILharmonicFlows

The PHILharmonicFlows framework (Process, Humans and Information Linkage for harmonic Business Flows) is a framework targeting on comprehensive support of object-aware processes.

It comprises both modeling and runtime environment enabling full lifecycle support for object-aware processes (Chiao et al., 2013). In fact, the framework comprises modules for the modeling, execution and monitoring of object-aware processes.

In this framework, object behavior is captured through micro processes. In turn, object interactions are captured by a macro process.

In PHILharmonicFlows framework, data is modeled separately for micro and macro processes (Chiao et al., 2014). A micro process captures the behavior of an object, while the macro process realizes the objects interactions (Chiao et al., 2013).

2.6 ProdProc

Product and Production Process Modeling and Configuration (ProdProc) is a developed system in a research project in (Campagna, 2012). It's a declarative constraint-based framework for defining models of both configurable products and their production processes. It allows a user to easily create product and process structures.

ProdProc allows the user to model a configurable product and takes into consideration models aspects of the production process for a product that may affect product configuration.

Also it allows the user to couple a product and a process description, in order to avoid or reduce planning impossibilities due to product configuration, and configuration impossibilities due to production planning.

Constraint Programming techniques were exploited in the development of ProdProc in order to guide the configuration of a product and its production process given the respective ProdProc model (Campagna, 2012). A ProdProc model consists of a description of a product, a description of a process, and a set of constraints coupling the two (Campagna, 2012).

3 THE TAXONOMY OF REGEV ET AL.

In this paper, we adopt the taxonomy of business process flexibility proposed by Regev et al. (Regev et al., 2006). In fact this taxonomy provides a means for classifying flexibility with respect to the types of changes it enables.

We find the taxonomy generic enough to allow choosing the flexibility criteria of our approach. These flexibility criteria will be specified in the next section.

Regev et al. presented in (Regev et al., 2006) a taxonomy of flexibility in business process. The taxonomy includes three orthogonal dimensions:

- the abstraction level of change
- the subject of change
- the properties of change

The abstraction level of change concerns the changes in the business process type or in business process instances.

Regev et al. suggest that the subject of change in business processes can be traced to five perspectives: the functional, the organizational, the behavioural, the informational and the operational perspective (Regev et al., 2006).

The functional perspective describes the process itself. In particular, it describes the number of times that a process must or can be executed. Complex processes can be decomposed into a set of steps or sub-processes. The behavioural perspective describes the order in which the process steps have to be executed. The organizational perspective defines persons that are responsible for the execution of a given step. The data perspective describes the data consumed and produced by a process step. Taken together, the data input and output of each process define the data flow of a process model. The operational perspective defines tools and systems that support the execution of a process step respectively generating data (Igler et al., 2010).

Regev et al. consider four properties of change: the extent, the duration, the swiftness and the anticipation of change.

The extent of a change can be incremental or revolutionary. Incremental changes start from an existing process type and only introduce changes to the already existing process type. Revolutionary changes abolish the existing process type and create a completely new one.

The duration of change can be either temporary or permanent.

4 EVALUATION OF BPMSs

We proposed the use of the taxonomy of Regev et al. as the starting point for evaluating BPMSs. In this section we evaluate the different BPMS using a questionnaire that was sent to the most senior personnel responsible for the development of the different BPMSs.

4.1 The Flexibility Criteria

In this section we identify a set of criteria that flexible BPMSs should be evaluated against. Using the taxonomy of Regev et al., we have specified eleven Flexibility Criteria (FC).

The different flexibility criteria concern the following questions:

- FC1: Does the BPMS support changes to process models which will affect all new process instances?
- FC2: Does the BPMS support changes at the instance level, and that will only affect certain selected instances, in order to accommodate exceptional situations?
- FC3: To which extent does the BPMS modellers have to describe the process control flow?
- FC4: To which extent does the BPMS support descriptive modelling and execution of process activities?
- FC5: To which extent does the BPMS support descriptive modelling and execution of the preconditions of the activities?
- FC6: To which extent does the BPMS support descriptive modelling and execution of data/information exchanged between process activities?
- FC7: To which extent does the BPMS support descriptive modeling and execution of roles associated to process activities?
- FC8: Can the BPMS support incremental change and/or revolutionary change?
- FC9: How would the duration of change that the BPMS support be characterized: temporary and/or permanent?
- FC10: Is the BPms able to deal with immediate and/or deferred change?
- FC11: Can the BPMS support planned / ad-hoc changes?

FC1 refers to the criteria of change in the business process type. FC2 refers to the criteria of change in the business process instance.

FC3, FC4, FC5, FC6 and FC7 concern respectively functional, organizational, behavioural, informational and operational perspective.

FC8 refers to the extent of change. FC9 refers to the duration of change. The swiftness of change is presented in FC10. Finally FC11 is the criterion of the anticipation of change.

4.2 Determination of the Flexibility Scores

We have developed a questionnaire using the taxonomy of Regev et al. The questionnaire was specifically designed to seek responses from the most senior personnel responsible for the development of the different BPMSs. Using the responses to this questionnaire, we calculated the derived flexibility score (FS) for each BPMS.

It's important to mention that the FCs have the same weight. We have specified for the FCs a scale in order to get consistent results. The table 1 presents the FCs' scales.

Table 1: The FC scales.

FC	SCALE
FC1, FC2	yes : 1 no : 0
FC3, FC4, FC5, FC6,FC7	0 (not descriptive) to 5 (very descriptive)
FC8, FC9, FC10, FC11	1: only one of the values, 2 : both, 0: none

Using theses scales, we have calculated a Flexibility Score (FS) for each of the BPMSs.

The ProdProc system supports only changes caused by modification of the process definition.

While ESProNa and DECLARE support changes at the instance level. In ADEPT2, PHIharminicFlows and JRules BPMS, process changes can take place at the type as well as the instance level. The ADEPT2 interviewee said "Our system is very flexible in this respect. It allows for changes of single instances (e.g. to deal with exceptions) as well as changes of a process model at the type level and the propagation of these changes to all or selected process instances of this type". According to the PHILharmonicFlows interviewee, "Changes in the PHILharmonicFlows are less required compared to ADEPT2; instead PHILharmonicFlows inherently allows for more execution paths that may be flexibly chosen by users".

The different perspectives, defined in the Regev et al. taxonomy, were considered in the different BPMSs, with verifying extents.

By combining the different perspectives with a medium extent, a comprehensive description of a process can be accomplished in the ESProNa system. The ESPRoNa interviewee said when explaining the behavioural perspective : "In ESPRoNa only the default-path through a process model is modelled and the preconditions are modelled inside the process-perspectives, the user sometimes cannot see this when only looking/concentrating to the flow as it is implicit in that situation".

ADEPT2 offers powerful concepts for supporting the five perspectives which allow a comprehensive description of the business processes. It’s the most successful BPMS supporting all the subjects of change defined in the taxonomy of Regev et al..

In PHILharmonicFlows BPMS, the behavioral perspective (FC5) of the software system is represented in two different levels of granularity: micro and macro process types. A micro process type defines the behaviour of a particular object type. A macro process type, in turn, defines how object instances interact with other objects instances. Also, by modeling the object types and their relations, fundamental insights into information perspective (FC6) can be obtained using PHILharmonicFlows BPMS (Chiao et al., 2013b). Additionally, the organizational perspective (FC7) of the software system is represented using user roles and types.

The functional (FC3) and operational (FC4) perspectives are well defined in JRules, unlike the informational (FC6) and organizational perspectives (FC7).

For Declare, the major efforts have been put in the development and improvement of the description of the activities executed during the process (FC4).

ProdProc provides a basic support to the definition of the information which shall be exchanged between activities and to describe the different roles (FC6 and FC7), while the functional perspective is limited (FC3). In addition, A MART model in ProdProc does not simply represent a single production process. Instead, it represents a configurable production process, whose configuration can lead to the definition of different executable processes (FC4) (Campagna, 2012).

In Jrules, the policy manager makes incremental change. The revolutionary change is identified by DECLARE, ESProNa and PHILharmonicFlows BPMS. Both the incremental and revolutionary changes are only provided in ADEPT2. In ProdProc, no specific support is provided for explicitly representing the extent of change (FC8).

Concerning the duration of change (FC9), the permanent change is supported by all evaluated BPMS. However, ADEPT2 allows also for the temporary change.

In the context of the swiftness of change (FC10), the deferred change is supported by all evaluated BPMS. However, ADEPT2 allow also for the immediate change. In fact, ADEPT2 applies immediately to all family-related process models and instances, even the running ones (includes runtime migration strategies).

For the anticipation of the change (FC11), in Declare, Jrules and PHILharminicFlows BPMS, explicit support for planned changes is provided. Nevertheless, to deal with exceptions BPMSs must support unplanned changes. For instance, the ADEPT2 and ESProNa BPMS provide also some form of ad-hoc changes. The ad-hoc changes which are based on adaptation patterns are possible in ADEPT2. The anticipation of the change, either planned or ad-hoc, are not considered in ProdProc.

The score for the output analysis of flexibility criteria for each of the tools is presented in table 2.

Table 2: Calculated flexibility scores for each BPMS.

	DECLARE	ESProNa	JRules/ JSolver	ADEPT 2 / AristaFlow BPM Suite	PHIharmonic Flows	PordProc
FC1	0	0	1	1	1	1
FC2	1	1	1	1	1	0
FC3	2	3	4	4	2	2
FC4	5	3	4	5	3	4
FC5	2	4	3	3	5	4
FC6	2	3	1	5	5	3
FC7	2	3	2	5	4	3
FC8	1	1	1	2	1	0
FC9	1	1	1	2	1	1
FC10	1	1	1	2	1	1
FC11	1	2	1	2	1	0

We calculated then the FS for the different BPMSs (table 3). To do that, we sum the different FC taking into account the scales. The calculated FS is calculated as follows:

$$\text{Calculated FS} = FC1 + FC2 + FC3/5 + FC4/5 + FC5/5 + FC6/5 + FC7/2 + FC8/2 + FC9/2 + FC10/2 \quad (1)$$

Table 3: Calculated FS for the different BPMS.

BPMS	Calculated FS
DECLARE	5,6
ESProNa	6,7
JRules/JSolver	6,8
ADEPT 2 / AristaFlow BPM Suite	9,9
PHILharminicFlows	7,8
ProdProc	5

The different BPMSs are evaluated according to the taxonomy of Regev et al. defined in the previous section. We conclude that the different BPMSs provide flexibility with different degrees.

5 COMPARISON OF FLEXIBILITY DEGREE FOR DIFFERENT PARADIGMS

5.1 Existing Paradigms

Literature provides various process modelling paradigms that we classify into: constraint-based, rule-based, case handling and adaptive process support paradigms. Each category has its underlying approach that may be examined in terms of its appropriateness to flexible process modelling.

Constraint-based paradigm focuses on constraints as rules that have to be followed during the process execution. Possible executions of constraint-based models are specified implicitly as all executions that satisfy the model constraints, which make it not necessary to explicitly predict all possible executions in advance.

The central concept, for the case handling paradigm, is the case and not the activities or the routing. The case is the ‘‘product’’ which is manufactured, and at any time workers should be aware of this context. Case Handling is a paradigm for supporting flexible and knowledge-intensive processes by strongly integrating them with data (Chiao et al., 2013a). Case Handling follows a revolutionary approach, departing from traditional workflow processes and their strict separation of data and control flow.

A paradigm is called rule-based if the logic of its control flow, data flow and resource allocation is declaratively expressed by means of business rules. Business rules are recognized as powerful representation forms that can potentially define the semantics of business process models and business vocabulary.

The adaptive Process Management can be seen as an evolutionary technique, solidly based on traditional workflow, while extending it with features to dynamically and safely adapt the process definition at any point in time (Martinho, 2010).

5.2 Classification of the BPMSs

According to the questionnaire’s results, the table 4 resume the answer to the following question:

“Which BPMS underlies modelling or execution paradigm?”

DECLARE is based on rule-paradigms and constraint-based paradigms. In fact, DECLARE is a constraint-based system that is focused on modelling constraints between processes. DECLARE uses the ConDec modelling language. Modelled constraints in ConDec are translated to a Linear Temporal Logic (LTL) formula. An automaton is generated for every specific constraint in order to verify it. Furthermore, a second automaton is generated over all constraints. Also, DECLARE allows for customized specification of relation types which are constraint templates. The DECLARE interviewee said “DECLARE is rule based because it is driven by Declare rules that are, at the end, LTL rules”.

JRules, which is a business rule management system, is based on a rule-based approach. A rule-driven BPMS is a superset of rule management systems and BPMS, which provides rich development options for many aspects of BPMS development, of both procedural (graph) and declarative rule approach (Lu and Sadiq, 2007). In addition, Jrules uses another tool, which is JSolver, which is a constraint solver.

The Case handling paradigm meets the needs and requirements of object aware processes (Chiao, Künzle and Reichert,2013a). Thus, PHILharmonicFlows is based on case handling.

The ProdProc was implemented using constraint logic programming of such product/process configuration system. A ProdProc model consists of a description of a product, a description of a process, and a set of constraints coupling the two. A product is modeled as a multi-graph, called product model graph, and a set of constraints (Campagna, 2012).

The basic design rationale of ESProNa is the separation of process model, process state and reasoner. The process model, which represents the constraints on all processes, is loaded into EsProNa.

The table 4 presents a classification of the considered BPMSs depending on the paradigms.

Table 4: BPMS classification.

Approach / BPMS	Constraint based	Rule based	Case handling	Adaptive Process Support
DECLARE	✔	✔		
ESProNa	✔			
JRules/ JSolver	✔	✔		
ADEPT 2 / AristaFlow BPM Suite				✔
PHILharminic-Flows			✔	
ProdProc	✔			

5.3 Paradigms’ Flexibility Measurement Score

In this paper, we aim at comparing the flexibility of the paradigms using the taxonomy of Regev et al. We have the calculated FS of the different BPMSs and we have classified the BPMSs depending on the paradigms: constraint-based, rule-based, case handling and adaptive process support. Then we calculated the average of these FS for each paradigm. The table 5 presents the PFMS (paradigms' flexibility measurement score). This score is calculated using this formula:

PFMS of (paradigm)= SUM (Calculated FS of each BMPS that support the paradigm) / (sum of the BPMS that support the paradigm) (2)

For instance, to calculate the rule based paradigm's flexibility measurement score, we calculate: PFMS of (rule-based) = SUM (Calculated FS of (Declare,JRules/JSolver) / 2 = (5,6 + 6,8)/2 = 6,2

Table 5: Paradigms' flexibility measurement score.

Paradigm	PFMS
Constraint-based	6,03
Rule-based	6,2
Case handling	7,8
Adaptive Process Support	9,9

The results of the evaluation (table 5) underline that the paradigms, that we have analysed, are able to provide a support to the requirements of the Regev et al. taxonomy.

On the other hand, the Adaptive Process Support paradigm provides a more complete support for process flexibility. It provides evolution in case of unanticipated exceptions, both at process schema and instance level. The strategy used for devising a recovery procedure is manual, though, and requires the human intervention at run-time.

The three remaining paradigms, constraint-based, rule-based and case handling, all three qualify for business process management. These paradigms have different principles that determine the flexibility of the process for different dimensions.

6 RELATED WORKS

Evaluation of flexibility in the business process domain has a rich research background. Günther et al. focus on the two paradigms: adaptive process management and case handling. The authors compare both approaches with respect to their strong and weak points (Martinho, 2010).

In (Vullers and Netjes, 2006) authors considered a number of software tools and examined their suitability for BPS. The tools have been evaluated on their modelling capabilities, simulation capabilities and possibilities for output analysis.

In (Vanderfeesten, Reijers and Aalst, 2007), the research work investigates to which degree current case handling systems (FLOWer and Activity Manager) are able to support Product Based Workflow Design.

In (Di Ciccio, Marrella and Russo, 2014), they present a critical analysis of a number of existing process-oriented approaches by discussing their efficiency against the knowledge-intensive processes requirements.

In (Weber, Rinderle-Ma and Reichert, 2007), they evaluated selected approaches and systems regarding their ability to deal with process change.

7 CONCLUSIONS

This paper has presented the result of a critical and comprehensive analysis of the four prominent paradigms, with the focus on flexibility. The presented survey gives an overview of BPMSs and an analysis of the BPMSs' strengths and weaknesses in terms of flexibility considering the taxonomy of Regev et al. The BPMSs were selected because of their frequent usage in business process management field.

In future work, using the comparison between the considered BPMSs, we will adopt a user guidance generic framework which will provide a methodological guidance for users to choose the most appropriate paradigm in order to model their processes.

ACKNOWLEDGEMENTS

We are very grateful to Pr Ricardo Martinho for his help, time and consideration.

REFERENCES

Boyer, Jerome and Hafedh Mili. 2011. Agile Business Rule Development - Process, Architecture, and JRules Examples. Springer.

Campagna, Dario. 2012. Product and Production Process Modeling and Configuration PhD thesis Università Degli Studi Di Perugia.

Chiao, Carolina Ming, Vera Künzle and Manfred Reichert. 2013a. Enhancing the Case Handling Paradigm to Support Object-aware Processes. in Rafael Accorsi;

Paolo Ceravolo & Philippe Cudré-Mauroux, ed., 'SIMPDA' , CEUR-WS.org, , pp. 89-103.
Chiao, Carolina Ming, Vera Künzle and Manfred Reichert. 2013b. Integrated Modeling of Process- and Data-Centric Software Systems with PHILharmonicFlows. In 1st IEEE Int'l Workshop on Communicating Business Process and Software Models, Workshop held in conjunction with the 29th Int'l Conf on Software Maintenance. IEEE Computer Societey Press pp. 1-10.
Chiao, Carolina Ming, Vera Künzle and Manfred Reichert. 2014. Towards Schema Evolution in Object-aware Process Management Systems. In International Workshop on the Evolution of Information Systems and their Design Methods (EMISA 2014). Lecture Notes in Informatics (LNI) Koellen-Verlag.
Dadam, Peter, Manfred Reichert, Stefanie Rinderle-Ma, Hilmar Acker, Martin Jurisch, Ulrich Kreher, Kevin Göser and Markus Lauer. 2007. ADEPT2 - Next Generation Process Management Technology. In Proceedings Fourth Heidelberg Innovation Forum (Invited Paper).
Di Ciccio, Claudio, Andrea Marrella and Alessandro Russo. 2014. Knowledge Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches. Journal on Data Semantics pp. 1-29.
Igler, Michael, Paulo Moura, Michael Zeising and Stefan Jablonski. 2010. ESProNa: Constraint-Based Declarative Business Process Modeling. In EDOCW. IEEE Computer Society pp. 91-98.
Lu, Ruopeng and Shazia Sadiq. 2007. A Survey of Comparative Business Process Modeling Approaches. In In Proceedings 10th International Conference on Business Information Systems (BIS), number 4439 in LNCS. Springer pp. 82-94.
Maggi, Fabrizio Maria, Marlon Dumas, Luciano García-Bañuelos and Marco Montali. 2013. Discovering Data-Aware Declarative Process Models from Event Logs. In Business Process Management - 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings. pp. 81-96.
Martinho, R. (2010). ConFlex4SP - A Framework for Controlling Flexibility in Software Processes. PhD thesis, University of Trás-os-Montes e Alto Douro.
Pesic, M., H.M. Schonenberg and W.M.P. van der Aalst. 2009. DECLARE Demo: A Constraint-based Workflow Management System. In Business Process Management Demonstration Track (BPMDemos 2009), ed. A.K.A. de Medeiros and B. Weber. Vol. 489 of CEUR Workshop Proceedings CEURWS.org pp. 1-4.
Regev, Gil, Pnina Soffer and Rainer Schmidt. 2006. Taxonomy of Flexibility in Business Processes. In Proceedings of the CAISE*06 Workshop on Business Process Modelling, Development, and Support BPMDS '06, Luxemburg, June 5-9, 2006.
Reichert, Manfred, Stefanie Rinderle, Ulrich Kreher and Peter Dadam. 2005. Adaptive Process Management with ADEPT2. In ICDE. IEEE Computer Society pp. 1113-1114.
Schonenberg, Helen, Ronny Mans, Nick Russell, Nataliya Mulyar and Wil M. P. van der Aalst. 2008. Towards a Taxonomy of Process Flexibility. In CAiSE Forum, ed. Zohra Bellahsène, Carson Woo, Ela Hunt, Xavier Franch and Remi Coletta. Vol. 344 of CEUR Workshop Proceedings pp. 81-84.
Vanderfeesten, I., H.A. Reijers and W.M.P. van der Aalst. 2007. An Evaluation of Case Handling Systems for Product Based Workflow Design. In Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS 2007), Volume on Information Systems Analysis and Specication, ed. J. Cardoso, J. Cordeiro and J. Filipe. Institute for Systems and Technologies of Information, Control and Communication, INSTICC, Medeira, Portugal pp. 39-46.
Vullers, M. H. Jansen and M. Netjes. 2006. Business process simulation a tool survey. In In Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN.
Weber, Barbara, Stefanie Rinderle-Ma and Manfred Reichert. 2007. Change Support in Process-Aware Information Systems - A Pattern-Based Analysis. Technical Report TR-CTIT-07-76 University of Twente Enschede, The Netherland.