Ontology-based change impact assessment in dynamic business processes

Mohammed Oussama Kherbouche, Adeel Ahmad, Mourad Bouneffa, Henri
Basson

Ontology-based change impact assessment in dynamic business processes

Oussama Mohammed Kherbouche, Adeel Ahmad, Mourad Bouneffa, Henri Basson
Université Lille Nord de France
Laboratoire d'Informatique, Signal et Image de la Côte d'Opale
BP-719 62228 CALAIS Cedex FRANCE
Email: {kherbouche, ahmad, bouneffa, basson}@lisic.univ-littoral.fr

Abstract—**The business process models are often subjected to change rapidly in order to cope with the market demands. It may be useful for companies to adapt a monitoring mechanism to achieve flexible business process models. It is also desirable to control the ripple effects of a change on whole or part of the business process and its running instances. It requires an exhaustive understanding of concerned changes and their application levels. In this paper, we propose a methodology based on dependency analysis for an *a priori* change impact analysis in the business process models. The approach is based on the ontology definitions to describe the dependency relationships. The major objective is to obtain a knowledge base to help the designers and business experts to estimate the associated risk of intended changes and the effort required for their implementation.**

Keywords—Business process model; change impact analysis; dependency relationships; Ontology

I. INTRODUCTION

The Business Process models (BPM) [1, 2] are often subjected to multiple changes throughout their life cycle. The reasons of these changes can be manifold such as the correction of errors, managing exceptions, business innovations, performance improvements (optimization), or to respect the new legal laws, etc. However, the absence of an appropriate mechanism to manage change consequences can create difficult situations like deadlocks, infinite executions, multiple terminations [3, 4], non-compliance with regulations [5], inconsistencies or semantic conflicts [6]. Furthermore, the ability to efficiently deal with a changing process has been identified as one of the critical success factors for any BPM model [7, 8].

Several approaches and paradigms [9-14] have been proposed to cope with the changing processes and their flexibility. In [9], the author suggests a flexible modeling and execution of workflow activities based on a meta-model of business. This approach supports dynamic changes such as adding or deleting activities, but it requires that the activity is not in the running state during the change incorporation. Another approach proposed by [10] is based on the use of Petri nets to calculate the minimal region affected by the changes. The authors, in [11], provide another method for the construction of Compliance Graph between the old and new workflow versions. Similarly, ADEPT-flex [12] is a graph based workflow model for the integration of dynamic changes even during the execution instances of the model without losing control and structural coherence. Similarly, the authors in [13]

propose the combination of a set of patterns of change and the seven characteristics of change management. YAWL [14] is an initiative based on formal foundations which provides significant support of a number of distinct flexibility approaches.

However, despite the great benefit of the different approaches, proposed in the literature, we observe that there is an increased deficiency about the change impact analysis in business processes to better identify the change side effects and control its ripple effects. In this paper, we propose an approach for an *a priori* analysis of change impact propagation in business processes through dependency relationships analysis between the changed part and other potentially affected parts. It emphasizes the use of ontology to achieve this goal. In this regard, the different dependencies relationships are captured by an instance of ontology derived from business process models.

The rest of the article is structured as follows. The section II briefly narrates the theoretical overview of semantic business process management. Section III discusses the change impact analysis in the life cycle of business process model. We describe the typology of the dependency relationships in the business process models, in the section IV. The section V, discusses the ontology definition for the change incorporation in process models. The ontology-based change management approach is presented in detailed in Section VI. Later in Section VII, we conclude our contribution.

II. SEMANTIC BUSINESS PROCESS MANAGEMENT

Business processes are generally expressed using the Business Process Model Notations[1] (BPMN). The primary goal of BPMN is to provide graphical notations which are readily understandable by all business users. It creates a standardized bridge for the gap between the business process design and process implementation. Yet, there exists a lack of semantics in BPMN [4, 16]. In this regard, the Semantic Business Process Management (SBPM) is recent approach to complement this requirement. It aims to increase the level of automation in the translation between the business requirements view and the IT systems along with the resource levels. It involves the use of ontology languages and Semantic Web Services framework [15]. In fact, Francescomarino *et al.* [16] propose to formalize, verify and integrate the domain knowledge in BPMN 1.1 diagrams by applying semantic web techniques. The Semantics Utilized for Process Management or (SUPER EU project)

---

1 http://www.omg.org/spec/BPMN/2.0/PDF

within and between Enterprises creates a framework combining BPM enriched with machine readable semantics by employing Semantic Web technology[2]. An approach for representing a BPMN diagram by using ontology based formalism is also proposed in [17]. Among others, the authors in [18], present BPMN 2.0 ontology which can be used as a knowledge base for learning BPMN, as a syntax checker to validate separate BPMN 2.0 models.

III. THE CHANGE IMPACT ANALYSIS

We can define the change impact analysis in business process model as a process which allows determining the elements of the model and its instances which can be susceptible to be directly or indirectly affected after each change. The changes at process instance level (also known as instance-specific changes) are often applied in an ad-hoc manner to deal with the exceptions (unanticipated situations) which can occur during the execution of an instance [19, 20]. In many cases, changing the state of a running instance is not sufficient for resolving problems, but the process structure itself has to be adapted as well [21]. For this reason, the process type level (schema evolution) is necessary to deal with the evolving nature of process roles (e.g., to adapt them to new legal requirement or new policies). Such a schema evolution often necessitates the propagation of respective changes on whole or part of the business processes and to ongoing process instances (global impact) [22], particularly if these instances have a longer runtime (e.g., medical or handling of leasing contracts, etc.) [23]. As a consequence of the changes there can result a difference (denoted as Δ) between the initial process schema S_0 and the obtained process schema S_1 which can be expressed as follows:

$$S_1 = S_0 + \Delta$$
$$\Delta = |S_1 - S_0|$$

The variant (Δ) can generate post-change impacts (in structural, functional, behavioral, logical, and qualitative aspects) on whole or part of the process and its instances. Therefore, an *a priori* analysis of this variant (also called impact analysis) is important to ensure the correctness and consistency of the conduction of changes. Besides traceability analysis [24, 25], the dependency relationship is one of the most commonly identified common thread for the change impact analysis process.

The change impact analysis in BPM model can be explained with the help of a Meta model as shown in Fig. 1. It provides a useful overview of different concepts of change and kinds of impact to support the change impact analysis in business processes. A change can be considered as an entity applied at process type level or process instance level. Any change in a business process can propagate its impact in multiple aspects i.e. structural, functional, behavioral, logical, and qualitative. Therefore, it necessitates a comprehensive analysis as required by its definitions.

IV. DEPENDENCY RELATIONSHIP ANALYSIS

Like any other information system, a business process is composed of different kinds of components or entities and
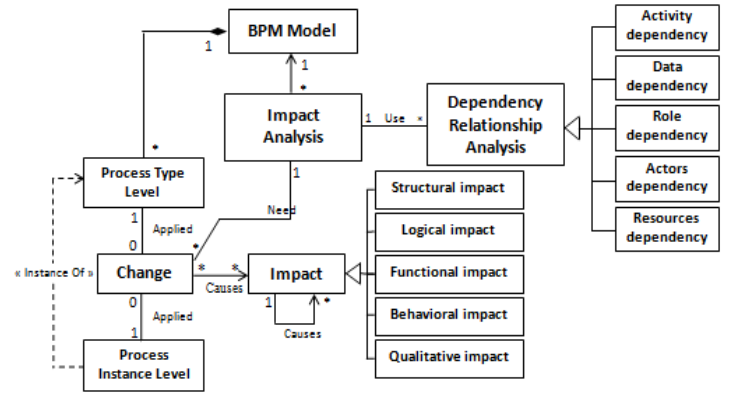
Fig. 1. A meta-model of change impact analysis

there always exist either direct or indirect relationships among them. The dependency analysis is a process of identifying the entities that are dependent on a given entity in a business process model.

The dependency relationships are defined considering the specifications of business process models. We propose a multi-dimensional business process dependency model that includes the following two major dependency relationships in business processes:

A. Activity dependency (routing)

The activity dependency (also called routing dependency) describes the execution order of activities within a business process through the control-flows (i.e. "sequence flow" and "message flow"). These dependencies define not only the execution order but also the semantics associated with this order. For example, an *AND-Join* routing of the three activities *A, B* and *C* such that *A* and *B* must execute before *C*. We can distinguish two kinds of activity dependency relationships [26].

1) *Intra-dependency:* It exists within a process. The intra-dependency refers to routing relationships between neighboring activities within the same process, as shown in Fig. 2 (within the Retailer process).

2) *Inter-dependency:* The inter-dependency relationship is concretely represented by the routing relationship between activities in the different processes (messages exchange). As shown in the Fig. 2 (between the *Retailer* and *Financial institution* processes).

The activity dependency can be formally defined as: $d_a = (\mathcal{D}_p, \Omega)$ over a set of activities $A = \{a_1, ..., a_n\}$ and a set of control-flows $T = \{t_1, ..., t_n\}$., where:

$$\mathcal{D}_p = \mathcal{D}_{pi}(a) \cup \mathcal{D}_{po}(a) \quad whereas \ a \in A.$$

$\mathcal{D}_{po}(a)$ is a set of all succeeding activities $a_i \in A$ (denoted as: $a_i \rightarrow a$) where the executions are dependent on activity *a*. The relationship can be one-to-many i.e., multiple activities depend on one activity. In the same way, $\mathcal{D}_{pi}(a)$ is a set of all preceding activities $a_i \in A$ (denoted as: $a \rightarrow a_i$) on which the execution of activity *a* is dependent. The relationship may be a many-to-one, i.e., one activity depends on multiple activities.
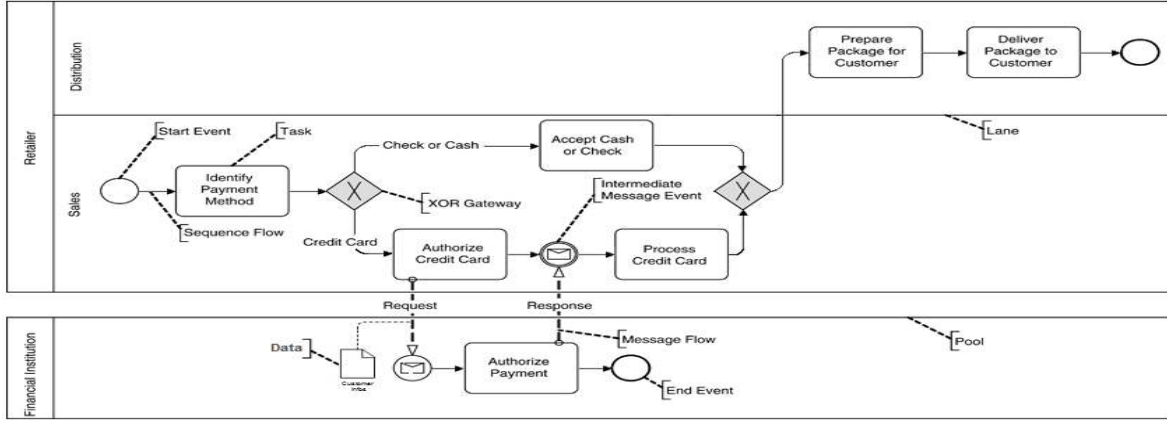
Fig. 2. The payment process

The set of control flows can be formaly shown as:

$$\Omega = \Omega i \ \cup \ \Omega o$$

The Ωi is a set of control-flows, $ti \in T$, connecting each activity $a_i \in \mathcal{D}i(a)$ to a, i.e. all incoming arcs $(\mathcal{D}i(a), a)$ of a. While the Ωo is a set of control-flows, $ti \in T$, connecting a to each activity $a_i \in \mathcal{D}o(a)$, i.e. all outgoing arcs $(\mathcal{D}o(a), a)$ of a.

B. Data dependency

The data dependencies emerge from common resources or data related to multiple activities. There exist three major types of data dependencies [27]:

1) Flow dependencies: It emerges whenever one activity produces a resource or data that is used by another activity.

2) Sharing dependencies: It occurs whenever multiple activities use the same resource or data.

3) Fit dependencies: It arises when multiple activities collectively produce a single resource or data.

We can formalize a set of all data transferred between activities [12] as:

$$D= \{d_1, d_2 \ ... \ d_n\}$$

Every activity $a \in A$ has input and output parameters, denoted as *InPARs(a)* and *OutPARs(a)* parameters, respectively. The symbol *dc* represents a data connection as:

$$dc = \{d, a, par, mode\}$$

Where $d \in D$, $a \in A$, $par \in InPARS(a) \cup OutPARs(a)$, and $mode \in \{read, write\}$. The set of all data connections can be represented as:

$$DC = \{dc_1, dc_2, \ ... \ , dc_n\}$$

An activity a_i depends on another activity a_j (denoted as: $a_i \xrightarrow{(D)} a_j$) iff: $\exists \, dcx, dcy \in DC$, Such that:

$$dcx = (d, a_j , pars, write)$$
$$dcy = (d, a_i , part, read)$$

Where $d \in D$, $part \in InPARs(ai)$, $pars \in OutPARs(aj)$ and a_j precedes a_i in process schema.

V. ONTOLOGIES DEFINITION FOR BPMN EVOLUTION

In this section, we discuss our approach, as illustrated in Fig. 3, which aims to analyze an *a priori* change impact propagation in business processes through dependency relationships analysis between the changed part and other potentially affected part. We capture the different dependency relationships by an instance of ontology derived from business process models. There are various languages for the explicit and formal representation of ontology. OWL is one of these languages and is defined by the W3C. It is widely used, especially in the web semantic area and is accepted as a standard.

To deal semantically with the BPM change impact analysis, we extend the BPMN ontology defined in [28] through the definition of different dependency relationships. We add new concepts defined by classes and properties focusing on BPM change impact and the relationships conducting such an impact. The proposed framework establishes the links between business process model elements and ontology concepts based on BPMN specifications[3]. It expresses each BPMN model element as a class in the ontology and its corresponding attributes as attributes of the class.

The ontology currently consists of 95 classes, 108 object properties, 70 data properties and 439 class axioms. The elements are divided into two categories representing *Supporting Elements* and *Graphical Elements*, the latter category is further refined in Flow Object, Connecting Object, Swimlane, and Artifact. Every BPMN element is represented as a *class* [28].

Subsequently, the BPMN model elements can be represented through the instantiation of these classes and properties in the ontology. The linkage of the ontology and the BPMN model element instances is accomplished by the help of properties. These properties specify the semantics of a BPMN model element with the help of a relationship of an ontology instance with formal semantics defined by the ontology.

---

3 www.omg.org/spec/BPMN/1.1/PDF.

We propose then to add a new class called *Impact Analysis* added at the same level as *Supporting Elements* and *Graphical Elements* that are of type *BPMN_ELEMENT*. This class is further refined in the various dependency relationships. We use the Description Logic (DL) notation[4] for OWL to exemplify the axioms in our extended BPMN ontology in their logical form. The redefinition of a *BPMN_ELEMENT* class in DL formulas is shown in List 1. Similarly, The *Impact Analysis* and *Dependency relationship* classes are defined in List 2 and List 3, respectively. In the same way, the List 4 and the List 5 define the *Activity Dependency* class and the *Data Dependency* class, respectively.

List 1. BPMN element class

```
Class: BPMN_ELEMENT
1. Label: BPMN element
2. Description: Base element
3. BPMN_ELEMENT ≡ GRAPHICAL_ELEMENT
   ⊔ SUPPORTING_ELEMENT ⊔ IMPACT_ANALYSIS
4. GRAPHICAL_ELEMENT⊑¬SUPPORTING_ELEMENT
5. GRAPHICAL_ELEMENT ⊑ ¬IMPACT_ANALYSIS
6. SUPPORTING_ELEMENT⊑ ¬IMPACT_ANALYSIS
```

List 2. Impact analysis class

```
Class: IMPACT_ANALYSIS
1. Label: Impact analysis
2. Description: Impact analysis
3. IMPACT_ANALYSIS ≡
   DEPENDENCY_RELATIONSHIPS
```

List 3. Dependency relationship class

```
Class: DEPENDENCY_RELATIONSHIPS
1. Label: Dependency relationship
2. Description: Dependency relationship define
   the existing dependencies concerning activities
   and data.
3. DEPENDENCY_RELATIONSHIPS ≡
   ACTIVITY_DEPENDENCY ⊔ DATA_DEPENDENCY
4. ACTIVITY_DEPENDENCY ⊑ ¬DATA_DEPENDENCY
```

List 4. Activity dependency class

```
Class: ACTIVITY_DEPENDENCY
1. Label: Activity Dependency
2. Description: Activity Dependency
3. ACTIVITY_DEPENDENCY ⊑ CONNECTING_OBJECT
4. ACTIVITY_DEPENDENCY ⊑ (≥1)
            has_connecting_activity_name
5. Property: has_connecting_activity_name
6. Label: Name
7. Description: Name is an attribute that is text
   description of the concerned acivity.
8. has_connecting_activity_name has domain
   ACTIVITY
9. has_connecting_activity_name has range
   xsd:string
10.ACTIVITY_DEPENDENCY ⊑ (≥1)
   has_connecting_activity_source_ref.ACTIVITY ⊔
 (≥1) has_connecting_activity_target_ref.ACTIVITY
11. Property: has_connecting_activity_source_ref
12. Label: aSourceRef
13. Description: aSourceRef is an attribute that
    identifies which Activity the Connecting
```

4 http://www.cs.ox.ac.uk/ian.horrocks/Seminars/download/Horrocks_Ian_pt1.pdf

```
   Object is connected from.
14. has_connecting_activity_source_ref has domain
    CONNECTING_OBJECT
15. has_connecting_activity_source_ref has range
    ACTIVITY
16. Property: has_connecting_activity_target_ref
17. Label: aTargetRef
18. Description: aTargetRef is an attribute that
    identifies which Activity the Connecting
    Object is connected to.
19. has_connecting_activity_target_ref has domain
    CONNECTING_OBJECT
20. has_connecting_activity_target_ref has range
    ACTIVITY
```

List 5.Data dependency class

```
Class: DATA_DEPENDENCY
1. Label: Data Dependency
2. Description: Data Dependency
3. DATA_DEPENDENCY ⊑ CONNECTING_OBJECT
4. DATA_DEPENDENCY ⊑ (≥1)
            has_activity_input_sets.ACTIVITY
   ⊔ (≥1)has_activity_output_sets.ACTIVITY
 has_activity_input_sets.ACTIVITY ≡ (DATA_OBJECT)
```

VI. ONTOLOGY BASED CHANGE IMPACT ANALYSIS

The Fig. 3 shows our approach, that deals with the BPM change impact analysis by the use of ontologies. The processes formalized by means of BPMN are stored in a BPM repository that is a relational database. The objects of such a database are semantically annotated by the BPMN ontology extended by features concerning the BPM change and the BPM change impact analysis like dependency relationships, etc.
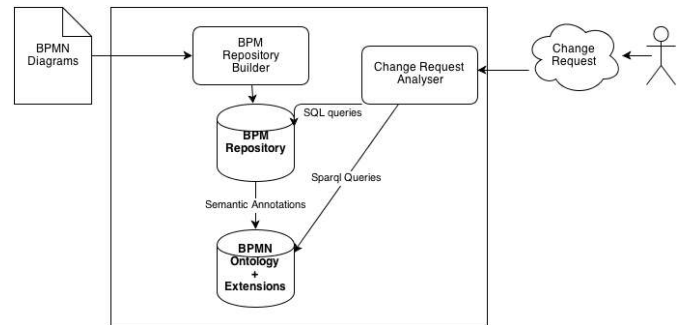
Fig. 3. Overview of our approach

The user, that is generally a BPM modeler, expresses BPM changes as change requests. The change request engine uses the rules to compute dependencies. These rules use the BPMN ontology and its extension. To do this the change request engine uses the ontology query language (SPARQL). The user may express a change concerning the deletion of a task. The change request engine fires all the rules computing the dependencies of the concerned task. This is done by means of SPARQL queries. The dependencies are then used as a mean to track the impact of the task deletion.

For the validation of this work, we have used one of the most popular tools called Protégé-OWL which supports building ontology based on the Web Ontology Language
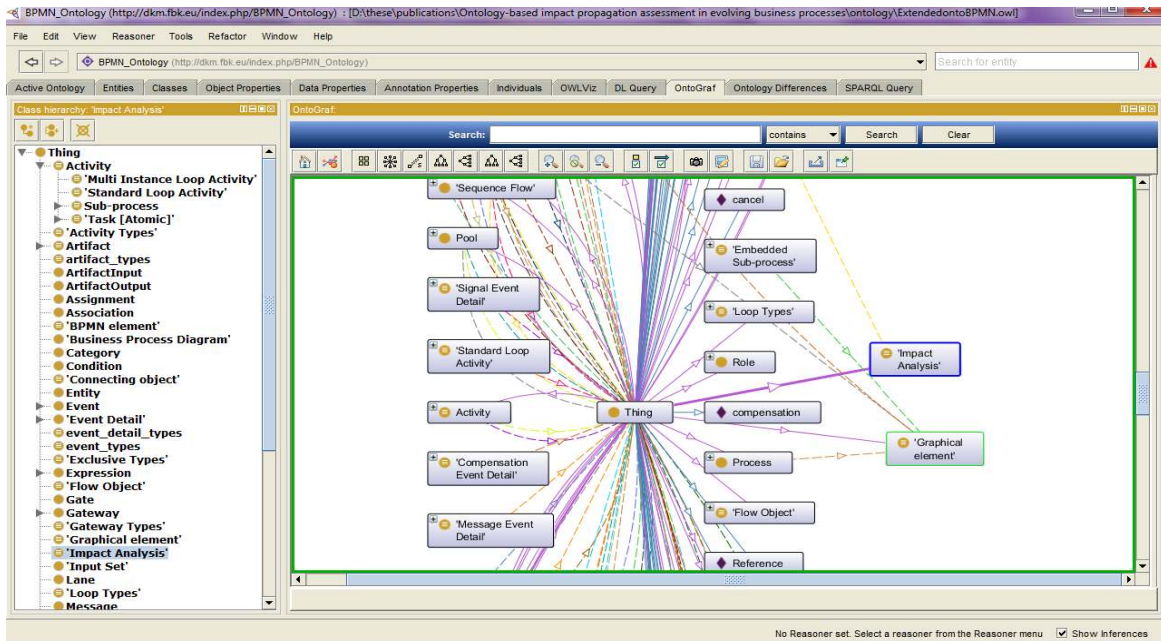
Fig. 4. Class Hierarchy of extended BPMN Ontology framework

Fig. 5. The extended BPMN ontology (represented in OWL)

(OWL) and it contains also knowledge-base framework editor. Fig. 4, shows the class hierarchy (as it appears in Protégé tool[5]) for the extended BPMN ontology that has been explained in this paper, and Fig. 5 shows the OWL file (ExtendedOntoBPMN.owl) which comprise the extended BPMN ontology.

The main objective has been to build a knowledge base from the results of instantiating ontologies. A business process definition with its control-flow can be visualized with the help of a knowledge base. It can help to assess the risks associated with a change. It can also help to determine

the impacted zone of intended changes by using combine concepts from *SPARQL* query language and rule-based system.

VII. CONCLUSION

It is inevitable for the business processes to incorporate changes during their life cycle. The changes can occur either at process instance level or process type level. It is necessary to analyze change and the generated ripple effects in the whole business process model. The change impact can propagate through traceability and dependency relationships. In the current article, we focus on the dependency relationships.

The business processes are generally implemented with the help of BPMN. We attempt to analyze two kinds of dependency relationships in BPMN models. The activity dependency or routing describes the execution order of activities in a process model through control-flows, while the data dependency describes the common usage of data among different elements of a process model.

We use the ontology to complement the semantic information in BPMN models. We extend the BPMN ontology framework with the addition of concepts (defined by classes and properties) to describe the activity and data dependencies. It can help to build the exhaustive model for an *a priori* analysis of the change impact in evolving business processes and thus establish a scalable base for progressively taking into account the other dependency dimensions such as roles, resources, actors, etc. The aim is to identify the potentially affected entities by a change in the process analysis domain.

REFERENCES

[1] R.Lu "Constraint-Based Flexible Business Process Management," in School of Information Technology and Electrical Engineering, University of Queensland, 2008.

[2] W. van der Aalst, and al, "Business Process Management: A Survey," in Proceedings of Conference on (BPM 2003), Eindhoven, Netherlands, pp 1-12, 2003.

[3] O.M Kherbouche, A.Ahmad, H. Basson, "Detecting structural errors in BPMN process models," in Proceedings of the 15th International Multitopic Conference (INMIC), Islamabad, Punjab, Pakistan, pp. 425- 431, IEEE Computer Society, 2012.

[4] O.M Kherbouche, A.Ahmad, H. Basson, " Using model checking to control the structural errors in BPMN models," in Proceedings of the 7th International Conference on Research Challenges in Information Science (RCIS), Paris, France, IEEE Computer Society, 2013.

[5] A. Awad, G. Decker, M. Weske, "Efficient Compliance Checking Using BPMN-Q and Temporal Logic," in Proceedings of the 6th International Conference on Business Process Management, pp. 326- 341, 2008.

[6] S. Rinderle, M. Reichert, and P. Dadam, "On dealing with semantically conflicting business process changes," Technical Report UIB-2003-04, University of Ulm, Computer Science Faculty, June 2003.

[7] P. Heinl, S. Horn, S. Jablonski, J. Neeb, K. Stein, and M. Teschke, "A Comprehensive Approach to Flexibility in Workflow Management Systems," In WACC '99: Proceedings of the international joint conference on Work activities coordination and collaboration, pp. 79- 88, New York, NY, USA, 1999. ACM

[8] H.A. Reijers, "Workflow Flexibility: The Forlorn Promise," in Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006), 26-28 June 2006, Manchester, United Kingdom, pp. 271-272. IEEE Computer Society, 2006.

[9] M. Weske, "Flexible modeling and execution of workflow activities," in Proceedings of the Thirty-First Hawaii International Conference, vol. 7, Jan. 6-9, pp. 713-722, 1998.

[10] P. Sun and C. Jiang, "Analysis of workflow dynamic changes based on Petri net," in Information and Software Technology, Volume 51, Issue 2, pp. 284-292 February 2008.

[11] S. Sadiq., M. Orlowska, "Architectural considerations in systems supporting dynamic workflow modification," in Proceedings of the workshop of Software Architectures for Business Process Management at the CaiSE'99 Heidelberg, Germany, pp. 14-18,1999.

[12] M. Reichert and P. Dadam, "ADEPTflex - Supporting Dynamic Changes of Workflows without Losing Control," Journal of Intelligent Information Systems, Special Issue on Workflow Management, Vol. 10, pp. 93-129, 1998.

[13] B.Weber, M. Reichert, S. Rinderle, "Change patterns and change support features – Enhancing flexibility in process-aware information systems," in Proceedings of the 19th international conference on Advanced information systems engineering, pp. 574-588, 2008.

[14] W.M.P. van der Aalst and A.H.M. ter Hofstede, "YAWL: Yet Another Workflow Language, "Information Systems, 30(4) pp.245- 275, 2005.

[15] J. Cardoso, M. Hepp, M. D. Lytras,"The Semantic Web: Real-World Applications from Industry", 2007.

[16] C. Di Francescomarino, P. Tonella, "Supporting Ontology-Based Semantic Annotation of Business Processes with Automated Suggestions," pp. 211–223, 2009.

[17] A. De Nicola, T. Di Mascio, M. Lezoche, F. Taglino, and I. Iasi, "Semantic Lifting of Business Process Models," in Proceedings of the 12th Enterprise Distributed Object Computing Conference Workshops, 2008.

[18] Ch. Natschläger, "Towards a BPMN 2.0 Ontology," Lecture Notes in Business Information Processing, vol. 95, no. Business Process Model and Notation, pp. 1-15, 2011.

[19] M. Minor, D. Schmalen, A. Koldehoff, R. Bergmann, "Structural adaptation of workflows supported by a suspension mechanism and by case-based reasoning", in: Proc. WETICE'07, 2007, pp. 370–375.

[20] M. Weske, "Workflow Management Systems: Formal Foundation, Conceptual Design, Implementation Aspects," University of Münster, Habil Thesis, 2000.

[21] M. Reichert, P. Dadam, T. Bauer, "Dealing with forward and backward jumps in workflow management systems," Software and System Modeling 1 (2), pp. 37-58, 2003.

[22] S. Nurcan, "A Survey on the Flexibility Requirements Related to Business Processes and Modeling Artifacts," in Proceedings of the 41st Hawaii International Conference on System Sciences HICSS'2008, pp. 378-378, 2008.

[23] S. Rinderle, M. Reichert, P. Dadam, "Flexible support of team processes by adaptive workflow systems," Distributed and Parallel Databases 16 (1), pp. 91–116, 2004.

[24] A. Marcus, J. I. Maletic, "Recovering Documentation to Source Code Traceability Links using Latent Semantic Indexing," in Proceedings of the 25th International Conference on Software Engineering, pp. 125- 135, 2003.

[25] A. V. Knethen, "A Trace Model for System Requirements Changes on Embedded Systems," in Proceedings of the 4th International Workshop on Principles of Software Evolution, pp. 17-26, 2001.

[26] G. Grossmann, G. Schrefl, M. Stumptner, "Modelling inter-process dependencies with high-level business process modelling languages," in Proceeding of the Asia-Pacific Conference on Conceptual Modelling (APCCM). Series of Conferences in Research and Practice in Information Technology (CRPIT) pp. 89–102, 2008.

[27] T. W.Malone, "Tools for inventing organizations: Toward a handbook of organizational processes", Published in Management Science 45(3) pp. 425-443, 1999.

[28] C. Ghidini, M. Rospocher, L. Serafini, "A formalisation of BPMN in Description Logic," Technical report, FBK, 2008.DOI = https://dkm.fbk.eu/index.php/BPMN_Ontology