Comparison of Business Process Modelling Approaches

Šarūnė Sielskaitė[1]

[1] *Vilnius Gediminas Technical University,*
Saulėtekio al. 11, Vilnius, Lithuania

Abstract

The software development process (SDP) is complex, dynamic, and knowledge-intensive by its nature. Therefore, we need a suitable approach to model it. Some authors have proposed different business process (BP) modelling approaches, like variant-based, rule-based, declarative, workflow-based, and case handling for various BP, such as dynamic, knowledge-intensive, goal-oriented, context-sensitive BP. This paper aims to investigate which BP modelling approach is most suitable for SDP modelling. Consequently, a set of necessary criteria is defined, and several BP modelling approaches are assessed. The obtained results show that the most suitable BP modelling approach for SDP is case handling, as it focuses on what can be done to achieve a predefined process goal rather than using a predefined workflow of the process.

Keywords

Software Development Process, Case Handling, Business Process Modelling.

1. Introduction

The software development process (SDP) is complex, dynamic, and knowledge-intensive, multiform, and divergent by its nature [12]. Consequently, we need a suitable approach to model it. However, because of mentioned complexities of SDP, there is a knowledge gap on how better it is to model SDP.

Business process (BP) modelling is used to cope with its complexity and management [3]. BP models are intended for business analysis and improvement [8], BP automation [8], etc. Nowadays, there are few quite well know BP modelling approaches, such as Business Process Model and Notation (BPMN), Unified Modelling Language (UML), Integrated Definition (IDEF) and others [7]. Nevertheless, there is a knowledge gap in how to model SDP applying existing BP modelling approaches.

Given the growing need to model SDP to capture software development, and the wide range of methods available for BP modelling, we want to know and learn more about the suitability of different BP modelling approaches for SDP.

Consequently, the research questions are defined as follows: 1) What are the existing BP types and their modelling approaches? 2) What are the requirements for modelling SDP? 3) Which existing BP modelling approaches are the most suitable for SDP?

This study aims to investigate which BP modelling approach is most suitable for modelling SDP. Therefore, this paper compares different BP modelling approaches such as context-sensitive, variant-based, rule-based, declarative, goal-oriented, knowledge intensive, workflow-based and case handling.

Baltic DB&IS 2022 Doctoral Consortium and Forum, July 03–06, 2022, Riga, LATVIA
EMAIL: sarune.sielskaite@vilniustech.lt (A. 1)
ORCID: 0000-0002-4842-0202 (A. 1)

This study contributes to SDP research by investigating the BP modelling approaches suitable for SDP modelling. By choosing the most suitable BP modelling approach, we could get the more precise SDP model.

Note that in the practice there is plenty of software management approaches, such as Agile, Waterfall, Scrum, Kanban, and all others [23]. However, in this paper we concentrate on SDP modelling but not on management activities.

The rest of this paper is structured as follows. Section 2 introduces related works on the comparison of BP modelling approaches. Section 3 presents the BP types and modelling approaches. Section 4 introduces SDP and main requirements for its modelling Section 5 shows the comparison of various BP modelling approaches. Finally, Section 6 concludes the paper.

2. Related works

Various BP modelling approaches have been offered in the literature, based on the user's needs and a BP type. In [11] authors analyze the declarative approach for BP modelling, which focuses on explicitly stating the business issues which control BP.

The authors in [18] compare two BP models that are based on two approaches: graphs and rules. The strengths and weaknesses of two approaches are examined in terms of: 1) expressiveness [18] (i.e., the ability of a process modelling language to communicate precise process requirements that represent the aim of process model-ling and execution determines its expressive power); 2) flexibility (i.e., the ability of a BP to function on a loosely or partially described model [18], with the full specification being made at runtime); 3) adaptability [18] (i.e., the ability to respond to unusual circumstances); 4) dynamism [18] (i.e., the workflow process' flexibility to adjust as the BP evolves); and 5) complexity [18] (i.e., the difficulties of modelling, analyzing, and deploying a process model, as well as support for dynamic and evolving BP).

Authors in [16] also compare three different modelling approaches: BPMN2.0, IDEF0 and IDEF3. The main goal of this research is to understand main differences and weakness of each modelling approach based on its usability, representation, communication and alignment, readability, and creativity. It is based on survey.

Other authors compare BP in a particular context. Authors in [5] compare BP modelling approaches in the context of process-level audit risk assessment according to main modelling constructs and requirements for process-level audit risk assessment.

As can be summarized, in the literature we can find different comparisons of BP modelling approaches. However, they are limited according to some features, like declarative BP [11], graphs or rules [18], or a particular context [5]. Contrary, we need a process modelling approach, which is mostly suitable for SDP modelling.

3. Business process types and modelling approaches

Because of the changing environment, BP becomes dynamic. Therefore, a uniform BP model cannot be defined for a dynamic business process (DBP) in all cases [14], i.e., it could adapt to a continuously moving and changing environment. Dynamic and flexible systems offer advantages for businesses in addressing dynamic uncertain factors and implementing DBP [14]. DBP modelling and simulations allows the process operator to see the impact of possible solutions or changes on the process end, identify problems and preventive measures to these problems at a process in-stance run-time.

Processes, where knowledge is a key characteristic, are called knowledge intensive processes (KIP) [6]. From the traditional, activity-centered point of view, KIP are challenging to automate, control, and test for a compliance, since KIP emphasizes the importance of the experience or tacit knowledge of process participants called knowledge workers [1, 33]. The role of knowledge workers is to accomplish specific activities that are characterized by complexity or require creativity [4]. In [13], authors also recognize flexibility in KIP as a crucial factor for their efficiency.

Goal-oriented process modelling is driven by the need to ensure congruence of BP and decisions with the values and vision of the business while meeting continuous demands for increased business

productivity [2]. It aims to extend traditional BP modelling that address the "how" of BP concerned with the efficient execution of BP to also include the "why" to ensure effectiveness of BP [2].

Context-sensitive processes use context to provide more relevant information to support users while performing their tasks. A context is any information that can be used to characterize the situation in which something exists or occurs [27]. Context-sensitive processes demand that designers consider new aspects and challenges in comparison to traditional approaches [26]. A context-sensitive process can adapt its instances at run-time to the changing context, defined by variables containing all relevant information about the design and execution of BP [32].

According to [33], authors model BP using families of BP variants, where a family of BP variants is presented via a single model, from which each variant can be de-rived through certain transformations (i.e., adding or deleting fragments) of the mod-el.

A BP model is called rule-based if the logic of its control flow, data flow and re-source allocation is declaratively expressed by means of business rules (BR). A rule-based process model represents BP as a description of its state space and a set of BR that constrain the valid movements in that state space. Rule-based process modelling allows to include a lot of useful information that otherwise would remain implicit in procedural process models. The advantages of rule-based process modelling manifest themselves during both the design and enactment phase of the BP management (BPM) lifecycle.

According to some [9] declarative models specify what should be done without specifying how it should be done. These models specify a set of constraints, BR, event conditions, or other expressions that define the properties of, and dependencies between, activities in BP [11]. So, BP instances are constructed according to predefined BR and do not violate them. These BR restrict the final execution path without defining the process model. In [9], the author states that declarative artifact-centric process models, which present knowledge-intensive processes, use BR that define how knowledge experts can make progress in a process. However, in many business situations knowledge experts must deal with uncertainty, which is modelled using fuzzy logic, and extends declarative artefact-centric processes models with fuzzy logic [9].

In a workflow-based BP, the business activity consists of atomic tasks, data types, and human actors [31]. A workflow-based modelling approach is focused on a static sequence of those tasks and their workflow scenarios.

Case management is a model for enabling flexible, knowledge-intensive BP. It is heavily reliant on data. Case handling, unlike workflow management, focuses on what can be done to achieve a business goal rather than using established process control structures describing what should be done during a workflow process. When it comes to case handling, the knowledge worker in charge of a certain case actively decides how that case's goal will be met [1]. In the case handling allocation system, at any time, workers aware of the whole case, which can be advantageously viewed as a single BP instance. This decreases processing time and eliminates errors because the knowledge worker already knows the situation and can address the difficulties and handle it quicker than a colleague who is unfamiliar with it [33].

4. Software development process

In this paper, we are interested in SDP modelling. Therefore, in this section, we define main requirements for SDP modelling as follows:

1. Dynamics: the developing SDP model should be dynamic, i.e., its model and in-stances should support structural changes (i.e., there is no predefined sequence of activities) at SDP instance run-time according to its context and rules, and that can be implemented with minimal delay (adopted from [14]).
2. Visible context: in the SDP model should be possible to define and observe the context or a business environment of the SDP instance [19].
3. Knowledge-intensive: in the SDP model, the knowledge worker should be responsible for the sequence of activities and how the goal of SDP is reached [30].
4. A single process instance: the focus should be placed on a single process instance rather than on a workflow [17].
5. Parallel assignment or skip role [20]:

a. The SDP model should support parallel modelling of activities in a SDP.

b. The SDP model should support parallel assignment of activities to one employee/resource in the same and/or in the separate process cases/instances.

6. Risk management: In the SDP model, it should be the possibility to assign risks for activities or an SDP instance/case [29].

Those requirements become the main comparison attributes for modelling approaches.

5. Comparison of various process modelling approaches

Table 1 presents the comparison of the five process modelling approaches (in columns) based on the predefined requirements (see Section 4) (in rows). The table con-sists of the following rows: 1-6) requirements from Section 5, 7) Primary driver – the main driver that is running BP; 8) and 9) are existing modelling language and its implementation in a tool. Those comparison criteria were developed based on requirements that were raised for SDP.

Table 1
Comparison of various process modelling approaches (NA – not found)

BP modelling approaches/ Attributes for comparison	Workflow-based [31]	Case-handling [1]	Rule-based [10]	Declarative-based [21]	Variant-based [25]
Dynamics	Static order of activities	Yes	Yes	Yes	Yes
Visible context	No	Yes	NA	NA	Yes [25]
Knowledge-intensive	No	Yes [1]	Yes	Yes	Yes
A single-process instance	No (Sequence of activities)	Yes (whole case data)	No (Rules for data transfer)	No (tasks and their relationships)	No (variability)
Parallelism	Yes	Yes	NA	NA	Yes
Risk management	Partially	Yes	Yes	NA	Partially
Primary driver	Sequence of activities	A whole case data	Rules for data transfer	A number of tasks possible for execution, relations between tasks**	A family of process variants
Existing modelling notation/language	BPMN, UML activity diagram	CMMN	DMN, SBVR, BPMN, Decision tables/matrixes *	ConDec	BPMN extensions (S-BPM [15]), Petri net extensions [25]
Implementation in a tool	MagicDraw, Bizagi Modeler	Flowable, Camunda, Trisotech	Trisotech, Camunda	Declare [28], [21]	Some plugins [25], [24]

* A rule-based approach for process modelling has no separate notation. Existing notations, like DMN, Semantics of Business Vocabulary and Business Rules (SBVR), BPMN, can be used to model rules in processes.

** The relationships between tasks in ConDec are understood as constraints, which represent policies (or business rules) [24].

From this table, the most suitable modelling approach is case handling as it meets all the comparison attributes. It could be used for DBP (Requirement 1). In this approach, the context can be visible (Requirement 2). There is nothing defined on how to specify a context in a declarative-based and a variant-based approaches. Since the primary purpose of those approaches is not related to the context,

it is therefore not described in the analyzed papers. A knowledge-intensive feature is occurring in all analyzed approaches, except a workflow-based approach (Requirement 3). A case handling approach is also suitable because it focuses on a whole case (Requirement 4) and allows us to model activities and task in parallel (Requirement 5). The same functionality could be seen in workflow-based and variant-based approaches as well. What is more, risk management could be adaptable to the case handling approach (Requirement 6), where a case can depict the risk management. All modelling approaches have notations and/or modelling languages (Table 1, rows 8-9). Some notations and tools could be applicable for few modelling approaches as they have similarities in characteristics.

6. Discussion and conclusions

At the beginning of this paper, we have defined three research questions to answer. Here we present the answers. The analysis of the BP modelling approaches shows that BP is a collection of main business activities and tasks that should be arranged in a specific sequence. It is significant to understand how BP are working in a particular domain area. For this reason, BP modelling has become crucial, and various BP types have their modelling approaches.

The analysis of BP types shows that the main of them are as follows: dynamic, knowledge-intensive, goal-oriented, context-sensitive. The found BP modelling approaches are the following: variant-based, rule-based, declarative, workflow-based, and case handling. Those found BP types and their modelling approaches are focused on a particular entity in a process, like activity, task, context, etc. Moreover, those BP modelling approaches view BP from a particular perspective, such as variants, rules, contexts, cases, workflows, etc.

The analysis of the related works allows us to formulate and specify six requirements of the SDP. Those SDP requirements are based on the main SDP characteristics. Based on the proposed SDP requirements, a comparison of the found BP model-ling approaches is presented.

The obtained comparison results show and teach us that the most suitable BP modelling approach for SDP is case handling, as it meets all defined requirements for SDP. Moreover, it focuses on what can be done to achieve a predefined process goal rather than using a predefined workflow.

In future works, we plan to use the case-handling approach to model and simulate SDP. Moreover, since SDP is knowledge-intensive and knowledge workers should deal with uncertainty, we will apply fuzzy set theory for knowledge in SDP modelling.

7. References

[1] Van Der Aalst WMP, Weske M, Grünbauer D (2005) Case handling: a new paradigm for business process support. Data Knowl Eng 53:129–162. doi: 10.1016/J.DATAK.2004.07.003

[2] Andersson B, Bider I, Johannesson P, Perjons E (2005) Towards a formal definition of goal-oriented business process patterns. Bus Process Manag J 11:650–662. doi: 10.1108/14637150510630846/FULL/PDF

[3] Becker J, Rosemann M, Von Uthmann C (2000) Guidelines of Business Process Modeling. LNCS 1806:30–49

[4] Boissier F, Rychkova I, Le Grand B (2019) Challenges in knowledge intensive process management. Proc - IEEE Int Enterp Distrib Object Comput Work EDOCW 2019-October:65–74. doi: 10.1109/EDOCW.2019.00023

[5] Carnaghan C (2006) Business process modeling approaches in the context of process level audit risk assessment: An analysis and comparison. Int J Account Inf Syst 7:170–204. doi: 10.1016/J.ACCINF.2005.10.005

[6] Di Ciccio C, Marrella A, Russo A (2015) Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches. J Data Semant 4:29–57. doi: 10.1007/S13740-014-0038-4/TABLES/2

[7] Cremonini Entringer T, Cristina De Oliveira Nascimento D, Da A, Ferreira S, Maurício P, Siqueira T, De Souza Boechat A, Cerchiaro IB, Bichara Mendonça S, Ramos RR (2019) Comparative Analysis main Methods Business Process Modeling: Literature Review, Applications and Examples. Int J Adv Eng Res Sci 6:2456–1908. doi: 10.22161/ijaers.6.5.15

[8] Decker G, Dijkman R, Dumas M, García-Bañuelos L (2010) The Business Process Modeling Notation. Mod Bus Process Autom YAWL its Support Environ 347–368. doi: 10.1007/978-3-642-03121-2_13
[9] Eshuis R (2018) Feature-Oriented Composition of Declarative Artifact-Centric Process Models. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 11080 LNCS:66–82. doi: 10.1007/978-3-319-98648-7_5
[10] Goedertier S, Haesen R, Vanthienen J (2008) Rule-based business process modelling and enactment. Int J Bus Process Integr Manag 3:194–207. doi: 10.1504/IJBPIM.2008.023219
[11] Goedertier S, Vanthienen J, Caron F (2014) Declarative business process modelling: principles and modelling languages. http://dx.doi.org/101080/175175752013830340 9:161–185. doi: 10.1080/17517575.2013.830340
[12] Grambow G, Oberhauser R, Reichert M Knowledge Provisioning A Context-sensitive Process-oriented Approach Applied to Software Engineering Environments. doi: 10.5220/0004083005060515
[13] Herrmann C, Kurz M (2011) Adaptive Case Management: Supporting Knowledge Intensive Processes with IT Systems. Commun Comput Inf Sci 213 CCIS:80–97. doi: 10.1007/978-3-642-23471-2_6
[14] Kalibatiene D, Vasilecas O (2021) A survey on modeling dynamic business processes. PeerJ Comput Sci 7:1–52. doi: 10.7717/PEERJ-CS.609/SUPP-1
[15] Kang G, Yang L, Zhang L, Liu J, Wen Y (2021) Organization-driven business process configurable modeling for spatial variability. China Commun 18:229–242. doi: 10.23919/JCC.2021.11.016
[16] Kelani KT (2018) MODELLING TECHNIQUES: comparison between BPMN2. 0 and IDEF (0&3) (Doctoral dissertation)
[17] Lemos AM, Sabino CC, Lima RMF, Oliveira CAL (2011) Using process mining in software development process management: A case study. Conf Proc - IEEE Int Conf Syst Man Cybern 1181–1186. doi: 10.1109/ICSMC.2011.6083858
[18] Lu R, Sadiq S (2007) A Survey of Comparative Business Process Modeling Approaches. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 4439 LNCS:82–94. doi: 10.1007/978-3-540-72035-5_7
[19] MacCormack A, Verganti R (2003) Managing the Sources of Uncertainty: Matching Process and Context in Software Development. J Prod Innov Manag 20:217–232. doi: 10.1111/1540-5885.2003004
[20] Martin RH, Raffo D (2000) SOFTWARE PROCESS IMPROVEMENT AND PRACTICE A Model of the Software Development Process Using Both Continuous and Discrete Models Research Section. Softw Process Improv Pr 5:147–157. doi: 10.1002/1099-1670
[21] Pesic M, Schonenberg H, van der Aalst WMP (2008) DECLARE: Full Support for Loosely-Structured Processes. 287–287. doi: 10.1109/EDOC.2007.14
[22] La Rosa M, Van Der Aalst WMP, Dumas M, Milani FP (2017) Business Process Variability Modeling. ACM Comput Surv 50. doi: 10.1145/3041957
[23] Stuart CE (2020) Cynefin-Framework as a Guide to Agile Leadership Which Project Management Method for Which Type of Project? - Waterfall, Scrum, Kanban?
[24] Tiam RT, Seriai A-D, Michel R An Operational Model of Variable Business Process. doi: 10.5220/0005372301620172
[25] Tiam RT, Seriai AD, Michel R (2015) VASCO: Variability Specification in Business Process Models. Lect Notes Bus Inf Process 241:612–632. doi: 10.1007/978-3-319-29133-8_30
[26] Vieira V, Tedesco P, Salgado AC (2009) A process for the design of context-sensitive systems. Proc 2009 13th Int Conf Comput Support Coop Work Des CSCWD 2009 143–148. doi: 10.1109/CSCWD.2009.4968049
[27] Vieira V, Tedesco P, Salgado AC (2011) Designing context-sensitive systems: An integrated approach. Expert Syst Appl 38:1119–1138. doi: 10.1016/J.ESWA.2010.05.006
[28] Westergaard M, Maggi FM Declare: A Tool Suite for Declarative Workflow Modeling and Enactment
[29] Ye T (2008) A study of software development project risk management. Proc - 2008 Int Semin Futur Inf Technol Manag Eng FITME 2008 309–312. doi: 10.1109/FITME.2008.125

[30] Ye Y (2006) Supporting Software Development as Knowledge-Intensive and Collaborative Activity. Proc 2006 Int Work Work Interdiscip Softw Eng Res - WISER '06 15–22. doi: 10.1145/1137661
[31] Zhang Y, Perry DE (2014) A data-centric approach to optimize time in workflow-based business process. Proc - 2014 IEEE Int Conf Serv Comput SCC 2014 709–716. doi: 10.1109/SCC.2014.129
[32] Zhou J, Gilman E, Palola J, Riekki J, Ylianttila M, Sun J (2011) Context-aware pervasive service composition and its implementation. Pers Ubiquitous Comput 15:291–303. doi: 10.1007/S00779-010-0333-5/FIGURES/8
[33] (2007) Business Process Management Architectures. Bus Process Manag 305–343. doi: 10.1007/978-3-540-73522-9_7