Towards Business Process Execution Adequacy Criteria

Antonia Bertolino, Antonello Calabró, Francesca Lonetti and Eda Marchetti

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"
Consiglio Nazionale delle Ricerche (CNR)
via G. Moruzzi, 1 - 56124 Pisa, Italy
{firstname.lastname}@isti.cnr.it

Abstract. Monitoring of business process execution has been proposed for the evaluation of business process performance. An important aspect to assess the thoroughness of the business process execution is to monitor if some entities have not been observed for some time and timely check if something is going wrong. We propose in this paper business process execution adequacy criteria and provide a proof-of-concept monitoring framework for their assessment. Similar to testing adequacy, the purpose of our approach is to identify the main entities of the business process that are covered during its execution and raise a warning if some entities are not covered. We provide a first assessment of the proposed approach on a case study in the learning context.

1 Introduction

Nowadays, more and more industrial organizations are using Business Process Model and Notation (BPMN) for process modeling. The main benefits of BPMN commonly rely on the possibility of having a simple and standard notation for creating a description of processes (in terms of participants and activities) and develop executable frameworks for the overall management of the process itself. Monitoring the business process execution represents a key aspect both for business process management and business process validation. Existing works [1, 2, 3] focus on monitoring and analysis of the factors that influence the performance of business processes. Specific key performance indicators (KPIs), including time based and cost based parameters, are defined together with their target values based on business goals. In this paper, we focus on monitoring the adequacy of the business process execution by defining coverage based adequacy criteria and a proof-of concept framework able to assess the BPMN execution adequacy. The main idea is to assess if a business process execution is *adequate*, i.e., if all the main entities (activities, connection objects, swimlanes, etc.) of interest of the

Manuscript Version. Published version available at: `http://link.springer.com/chapter/10.1007%2F978-3-319-27033-3_3`

business process are covered during its execution or if the business process execution misses some of them with consequent unexpected behaviour or security flaw. Our proposal extends a more generic notion of adequacy criterion, presented in our previous work [4], by defining and implementing an instantiation of this adequacy criterion for the business process execution. However, the goal of this paper is not to assess the monitor adequacy as in [4], but the adequacy of the business process execution using monitoring facilities for measuring the proposed adequacy criteria. The main idea is to define what are the relevant entities that we would expect to observe during BPMN execution and hence to set adequacy criteria on such entities, using monitoring facilities for observing and reporting about the percentage of entities that have been covered during business process execution. In such way we can become aware that some expected entities (for instance activities or swimlanes) have not been covered for some time, and then timely check whether this happens because something is going wrong during business process execution or this is simply due to a temporary decrease of users interest for those entities. This idea takes inspiration from coverage-based test adequacy that have been extensively studied in software testing, e.g. referring to coverage of entities in the program control-flow or data-flow [5], and nowadays constitutes a fundamental instrument for test suites evaluation. Similarly to testing adequacy, we introduce here the notion of business process execution adequacy. A difference with the traditional notion of testing adequacy is the concept of observation window, namely the period along which the business process execution is assessed. It is out of scope in this paper to address the problem of how to set the length of the observation window. We refer to [4] for an overview of existing methodologies for properly setting the length of such observation window.

The contribution of this paper can be summarized into: i) the definition of business process execution adequacy criteria; ii)a proof-of-concept framework able to measure the BPMN execution adequacy; iii) a preliminary assessment of the proposed proof-of-concept framework on a case study developed in the learning context.

The remainder of this paper is structured as follows: Section 2 introduces business process modeling notation and the notion of testing adequacy; Section 3 illustrates the business process execution adequacy criteria whereas Section 4 describes the architecture of a proof-of-concept framework able to measure the BPMN execution adequacy. Section 5 provides a preliminary assessment of the proposed approach. Finally, Section 6 puts our work in context of related work whereas Section 7 concludes the paper also hinting at future work.

2 Background

This section introduces the background behind the proposed approach. Specifically, we first present some key concepts of the Business Process Modeling Notation and then we focus on test coverage as adequacy criterion in software testing.

Business Process Modeling Notation. Business Process Model and Notation (BPMN) is a standard notation by the Object Management Group (OMG) [6] for specifying business process. BPMN provides a graphical notation for supporting business process management that allows to fill the gap between technical users and business users by providing a notation that is intuitive to business users and able to represent complex process semantics. There are three basic types of sub-models within an end-to-end BPMN model: Processes (Orchestration), Choreographies, and Collaborations. The five core notation elements of BPMN are: i) flow objects that allow to model event, activity, and gateway; ii) data items that model data within the process flow and are represented by four elements: data objects, data inputs, data outputs, and data stores; iii) connection objects that connect the flow objects to each other and are: sequence flow, message flow, and association; iv) swimlanes to model process participants; v) artifacts (group and text annotation) are used to provide additional information about the process.

Testing Adequacy. In software testing, coverage of entities of program control-flow or data-flow is a test adequacy criterion that has been proposed as an indicator of testing effectiveness for selection and evaluation of different test cases. Code coverage e.g. is measured as the fraction of program code that is executed at least once during the test execution. Various code coverage criteria have been suggested [7], including statement coverage, decision coverage, path coverage, C-use coverage, P-use coverage, etc. whereas different coverage metrics have been proposed for different languages and application domains. The correlation between code coverage and fault detection capability has been extensively studied but it remains nowadays a controversial issue. Some previous studies [8, 7] show that high code coverage implies high software reliability and low fault rate. Experimental studies [9] focus on coverage testing and mutation testing in order to investigate the relationship between code coverage and fault detection capability of a test suite. Others studies [10] show that the relationship between code coverage and fault detection varies under different testing profiles and it is affected by the different code coverage metrics. In this paper we propose to measure the adequacy of the business process execution by identifying what are the relevant entities to be covered and by assessing if all of them, or otherwise what percentage, have been covered. It is out of scope of this paper to investigate the relationship between business process coverage and fault detection.

3 Defining Business Process Execution Adequacy

In this section we introduce the generic concept of business process execution adequacy, without considering a specific coverage measure or application domain.

In test coverage criteria, a set of requirements that a test suite must fulfill is established and it is mapped onto a set of entities that must be covered when the test cases are executed, as for instance all statements or all branches of a program control-flow. The coverage criterion is satisfied if all the entities

are covered; otherwise, the percentage of covered entities represents a quality measure of the test suite.

The intuitive motivation behind measuring test coverage is that if some entity has never been tested, it might contain undetected faults. Obviously, the converse reasoning does not apply: if we had covered all entities and detected no failure, this does not necessarily imply that the program is correct. In a similar way, we propose here to assess the adequacy of the business process execution by identifying what are the relevant entities to be covered and by assessing if all of them, or otherwise what percentage, have been executed. To do this, we propose a proof-of-concept monitoring framework to observe and collect business process execution traces and measure the coverage of the entities belonging to these traces. As for test adequacy, the motivation behind assessing business process execution adequacy is that if some entities are not covered, we cannot exclude that these might hide some problem or security flaw. Similarly to the notion of testing session, namely the period along which the test adequacy is measured, we define the observation window as the length of a considered observation period associated to the business process execution coverage measure. Intuitively, a sliding observation window over a time measurement unit can be established, which could be either continuous (e.g. the execution traces collected in the last 120 sec) or discrete (e.g., the most recent 15 traces). The proposed business process execution adequacy criteria extend a general monitoring adequacy criterion presented in [4], by defining and implementing an instantiation of this adequacy criterion for the business process execution.

In the rest of this section the generic concept of business process (BP) execution adequacy criterion, without binding its definition to a specific coverage measure, is introduced. In fact, the notion of business process execution adequacy is neutral with respect to both the entities to be covered (i.e. activity, tasks, paths and so on) and the application domain. In the next section the generic concept of business process execution adequacy criterion is instantiated considering three different entities of the business process (Activity Entity, Sequence Flow Entity, Path Entity) and a set of coverage adequacy criteria is presented.

Definition 1 *Denote $r_i \in R$ the i-th entity to be covered, and by $\delta_i \in \Delta$ the length of its associated observation window. The business process execution adequacy criterion C dynamically measures the coverage on R for a given entity i at each time unit t as follows:*

$$C[R, \Delta](t) = \frac{\sum_{i=1}^{|R|} \lambda_i(t)}{|R|}$$

where for $r_i \in R$ and $\delta_i \in \Delta$

$$\lambda_i(t) = \begin{cases} 1 & \text{if } r_i \text{ is covered at least once in } [t - \delta_i, t] \\ 0 & \text{otherwise} \end{cases}.$$

According to this definition the length of δ_i could be different for each r_i, or could be the same for all entities. In summary the definition of business process execution adequacy introduces the following concepts:

- an "adequate business process execution" is a business process execution on which a set of entities r_i to be covered in a window δ_i is defined (this is similar to the instrumentation phase of coverage testing);
- a monitoring tool that, at every instant t, can provide a coverage measure as in Def 1 and, if this is less than 1, can provide a list of those entities that have not been covered;
- an entity that is not covered is an entity of the business process that has not been executed for some time. In such a case a warning message could be raised by the monitoring tool.

3.1 Entity Definition

Inside a business process execution the definition of what is an entity to be covered can be provided at different levels and with different targets. We consider the following definitions:

Definition 2 (Activity Entity) *Given a BP, an activity entity is one of the activities specified in the BP that can be executed at least once.*

Definition 3 (Activity Coverage Domain) *Considering a BP, the activity coverage domain is the set of all the activity entities of the BP.*

Definition 4 (Percentage of Activity Coverage) *With reference to Def 1, the percentage of activity coverage at time t is given by 100*C, where R is the activity coverage domain.*

Consequently, at a given instant a business process execution is adequate with respect to the activity coverage criterion if the percentage of activity covered is 100% (or greater than an established threshold level).

Definition 5 (Sequence Flow Entity) *Given a BP, a sequence flow entity is one of the sequence flow* [1] *specified in the BP that can be executed at least once.*

Definition 6 (Sequence Flow Domain) *Considering a BP, the sequence flow coverage domain is the set of all the sequence flows entities of the BP.*

Definition 7 (Sequence Flow Coverage) *With reference to Def 1, the percentage of sequence flow coverage at time t is given by 100*C, where R is the sequence flow coverage domain.*

[1] Sequence Flow is used to show the order in which activities of a process will be performed. A Sequence Flow connection is represented with a solid line and a solid arrowhead in a Business Process Model.

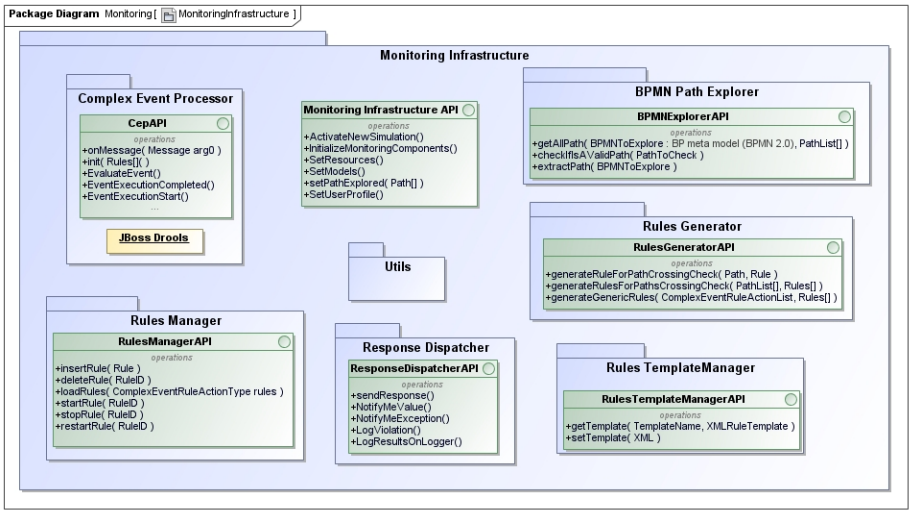

Fig. 1. Monitoring Package Diagram.

Consequently, at a given instant a business process execution is adequate with respect to the sequence flow coverage criterion if the percentage of sequence flow entities covered is 100% (or greater than an established threshold level).

Definition 8 (Path Entity) *Given a BP, a path entity is one of the paths specified in the BP that can be executed at least once.*

Definition 9 (Path Domain) *Considering a BP, the path coverage domain is the set of all the path entities of the BP.*

Definition 10 (Path Coverage) *With reference to Def 1, the percentage of path coverage at time t is given by 100*C, where R is the path coverage domain*

Consequently, at a given instant a simulation is adequate with respect to the path coverage criterion if the percentage of path entities covered is 100% (or greater than an established threshold level).

4 Framework

With reference to Figure 1, we present in this section the components of a proof-of-concept framework able to measure the business process execution adequacy, we have implemented:

– BPMN Path Explorer. This component is in charge to explore and save all the possible entities (Activity Entity, Sequence Flow Entity, Path Entity) reachable on a BPMN. The paths extraction is realized by an optimized unfolding algorithm that exploits advantages provided by the use of BPMN 2.0. The

goal is to derive an acyclic graph, defining a partial order on its nodes. In particular, the exploration reduces the required space and time thanks to a more efficient management of the interleaving among different activities, taking into account the characteristics of a BPMN 2.0 model and of pools, parallel and exclusive gateways, and tasks sending and receiving messages within the model. More details about the BPMN exploration approach are in [11]. Once extracted, the paths will be provided to the Rules Manager that through the Rules Generator will create, using the templates of rules stored into the Template Manager, a set of rules that aims to check the coverage of all the feasible paths of the business process.

- Complex Event Processor (CEP). It is the rule engine which analyzes the events, generated by the business process execution. Several rule engines can be used for this task like Drools Fusion, VisiRule, RuleML. Our instance is realized using Drools Fusion [12], that is able to detect patterns and monitor the business process execution adequacy.
- Rules Generator. The Rules Generator is the component in charge to generate the rules needed for the monitoring of the business process execution. It uses the templates stored into the Rules Template Manager. These rules are generated according to the specific adequacy criterion to be assessed and the entities to be covered. For each entity, the rule generator generates one corresponding rule for the CEP. A generic rule consists of two main parts: in the first part the events to be matched (the entities to be covered) are specified; the second part includes the events/actions to be notified after the rules evaluation (the *covered* attribute is set to true if the entity is covered).
- Rules Template Manager. This component is an archive of predetermined rules templates that will be instantiated by the Rules Generator. A rule template is a rule skeleton, the specification of which has to be completed by instantiating a set of template-dependent placeholders. The instantiation will refer to appropriate values inferred from the specific adequacy criterion to be assessed. Once the synthesis of the new set of rules is completed, the new rules are loaded by the Rule Generator into the Rules Template Manager.
- Rules Manager. The complex event detection process depends directly from the operation done by the Rules Manager component which is in charge to load and unload set of rules into the complex event processor and fire it when needed.
- Response Dispatcher. The Response Dispatcher is a registry that records the business process execution adequacy monitoring requests. Once it receives the advice of a rule firing or pattern completion from the CEP, it stores coverage information. It elaborates statistics about the overall percentage of the covered entities and raises warning messages for the entities that are not covered to the consumer/requester of the business process adequacy evaluation.

5 Preliminary Assessment

In this session we present a preliminary assessment of the proof-of-concept framework presented in Section 4 on a case study in the learning context, developed inside the Learn PAd European project [13]. The main goal of the Learn PAd project is to foster an innovative learning platform for Public Administrations, based on enriched business process models, where the steps performed by the learner during a learning session are associated to the execution of the entities of a business process.

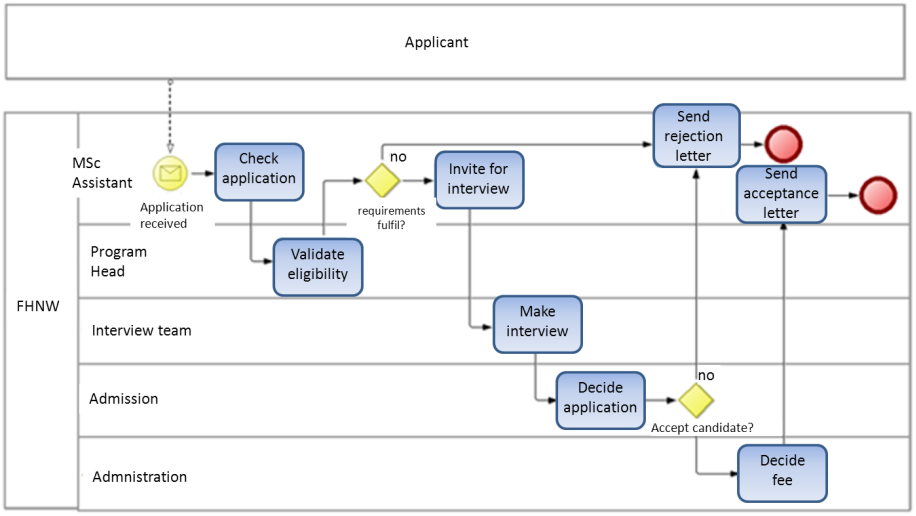

Fig. 2. Overview of the student admission process (from [14]).

The proposed business process execution adequacy criterion can be applied for providing some business process coverage measurements useful for assessing the adequacy of a business process based learning session. The intuitive motivation of using the proposed business process execution adequacy criterion for evaluating a learning session is that if some part of the business process has never been executed, the learner might have not exercised important steps of the business process and therefore his/her acquired knowledge could not be completed. To assess the adequacy of a learning session it is important to identifying what are the relevant entities to be covered and monitoring if all of them, or otherwise what percentage, have been observed. Through the analysis of data monitored over the learning session, the learner from one side can become conscious of the level of knowledge he/she acquired and can receive an evaluation/score of the progress done. Form the other side he/she can have a clearer picture of his/her exploration over a learning session, i.e to know exactly the entities (either the events, or message interactions, or business patterns) so far not executed, so

either to timely decide how to continue the learning activity or get his/her evaluation score.

The business process considered in this preliminary assessment, presented in Figure 2, is the mock-up of the real process "Student Admission" described in [14], that refers to the process for regulating the admission of applicants to the study program MSc in Business Information Systems (BIS) within the school of business at FHNW.

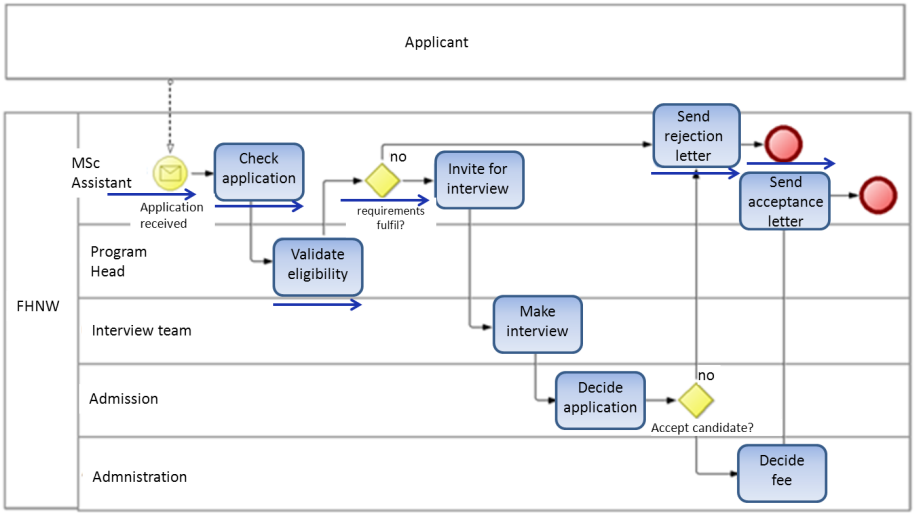

Fig. 3. Activity entities monitored over a learning session.

In this section we applied the proposed business process execution adequacy criteria and the implemented proof-of-concept framework described in Section 4, to the business process presented in Figure 2. The observation window for the assessment of the business process execution adequacy criteria has been fixed to the duration of a learning session.

In the performed learning session we varied the following independent variables:

- Coverage Criterion (CC): this parameter indicates the entities to be covered, i.e., CC $\in$ activity, path[2]. The activities for the BP of Figure 2 are for instance: *Invite for interview*, *Make Interview*, *Send acceptance letter* and so on. The cardinality of activity coverage domain for the BP of Figure 2 is 13, whereas that of path coverage domain is 3;
- Coverage Threshold (CT): this parameter indicates the coverage threshold according to which the business process execution is considered adequate.

[2] Note that for this specific business process path entity coincides with sequence flow entity.

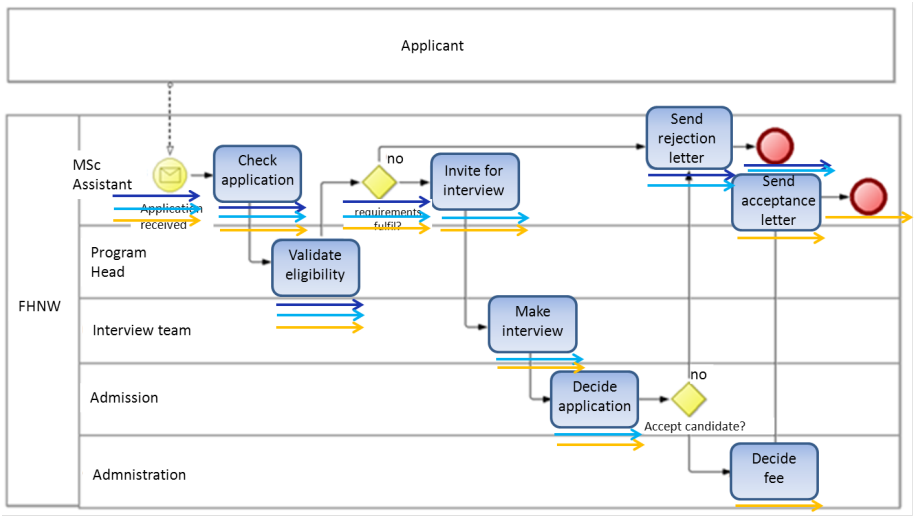

Fig. 4. Path entities monitored over a learning session.

In the fist step of our experiment the entity has been defined at the granularity of activity. The percentage of activity coverage within the established observation window was 6/13 *100 = 46,15 where: 6 was the number of activity entities monitored in the observation window (the activity entities marked with a (blue) arrow in Figure 3); 13 is the cardinality of the activity coverage domain. Since the learning adequacy level established were 100% of the activity entities, the performed learning session were not adequate with respect to the activity coverage criterion.

In a second step of the experiment, the entity has been defined at the granularity of path and the set of monitored (executed) path entities that have been observed in the observation window was represented just by the one marked with blue arrows in Figure 4. In this case the percentage of path coverage within the established observation windows is 1/3 *100 = 33 where: 1 is the number of path entities monitored in the observation window, 3 is the cardinality of path coverage domain. Since the learning adequacy level established in the observation window were 33% of the path entities, the considered learning session is adequate with respect to the path coverage criterion.

6 Related Work

The adoption of business process modeling promotes and makes easier the use of model-based approaches for verifying the dynamic distributed systems. Monitoring is assuming a key role for tracking the states of a business process and for evaluating its execution performance [1, 15]. Particular interest has been dedicated to “smart” monitoring approaches, i.e., monitors enhanced beyond the

passive observation of system executions, with the aim of preventing or anticipating potential risks. As a trend, several researchers start to consider monitoring a useful instrument to observe the behavior of business processes not only to report about problems that have already occurred but also to predict likely problems in the near future.

However, most of the predictive approaches remain limited to the elaboration of the passively captured executions. A first attempt to provide a monitoring approach also able to raise a warning of not having observed for some time interesting behaviours or situations is presented in [4] where the general concept of monitor adequacy and two adequacy criteria for service compositions are provided. This approach takes inspiration from the passive testing approaches, which refer to the observation of the input/output behavior of a system during normal operation for the purpose of detecting faults [16]. Our proposal extends the general adequacy criterion presented in [4] by providing the definition of a business process execution adequacy criterion and a proof-of-concept monitoring framework able to assess the business process execution adequacy. Differently from existing approaches on monitoring of business process that focus on QoS metrics and continuous evaluation of key performance indicators (KPI), our goal is to measure the adequacy of business process execution by identifying what are the relevant entities to be covered and by assessing if all of them, or otherwise what percentage, have been covered. This idea is similar to testing adequacy where coverage of entities of program control-flow or data-flow is a test adequacy criterion for assessing the test effectiveness. We refer to [17] for an overview on coverage measurements and coverage-based testing tools.

In the context of learning, that is the application domain of the proposed case study, contemporary Learning Content Management Systems (LCMSs) provide rather basic feedbacks about the learning process, such as simple statistics on technology usage or low-level data on students activities (e.g., page view). Some tools [18] have been developed for providing feedbacks on the learning activities by the analysis of the user tracking data in order to propose customized learning paths that learners can follow according to their knowledge and learning requirements. Our proposal applied to the model based learning allows to assess the adequacy of a learning session by providing feedback on the executed learning activities and identifying the learning paths that are not covered.

7 Conclusion and Future Work

In this paper we introduced the notion of business process execution adequacy and provided a proof-of-concept monitoring framework able to measure the proposed adequacy criteria. We presented a first assessment of the proposed approach on a case study developed inside the Learn PAd European project [13]. Even if preliminary, the experimentation evidenced the effectiveness of the proposed approach in providing some measures about the coverage of a business process, useful for evaluating the adequacy of the associated learning session.

In future work we intend to perform a more extensive assessment of the approach to evaluate its costs and benefits. We plan to refine and enhance the proposed business process execution adequacy notion in order to consider the adequacy relative to a specific role and/or level of the business process and provide the associated relative coverage measures. A further research direction deals with investigating on the length of the observation window, namely the period along which the business process execution is assessed.

Acknowledgments

This work has been partially funded by the Model-Based Social Learning for Public Administrations project (EU FP7-ICT-2013-11/619583).

References

1. Wetzstein, B., Leitner, P., Rosenberg, F., Brandic, I., Dustdar, S., Leymann, F.: Monitoring and analyzing influential factors of business process performance. In: Enterprise Distributed Object Computing Conference. (Sept 2009) 141–150
2. Bertoli, P., Dragoni, M., Ghidini, C., Martufi, E., Nori, M., Pistore, M., Di Francescomarino, C.: Modeling and monitoring business process execution. In: Service-Oriented Computing. Springer (2013) 683–687
3. Calabró, A., Lonetti, F., Marchetti, E.: Monitoring of business process execution based on performance indicators. In: The Euromicro Conference series on Software Engineering and Advanced Applications (SEAA). (2015)
4. Bertolino, A., Marchetti, E., Morichetta, A.: Adequate monitoring of service compositions. In: Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE'13. (2013) 59–69
5. Rapps, S., Weyuker, E.: Selecting software test data using data flow information. IEEE Transactions on Software Engineering **SE-11**(4) (April 1985) 367–375
6. OMG: Business Process Model and Notation (BPMN). (2011) 20th ed.: Object Management Group.
7. Horgan, J.R., London, S., Lyu, M.R.: Achieving software quality with testing coverage measures. Computer **27**(9) (1994) 60–69
8. Weyuker, E.: The cost of data flow testing: an empirical study. IEEE Transactions on Software Engineering **16**(2) (Feb 1990) 121–128
9. Lyu, M., Huang, Z., Sze, S., Cai, X.: An empirical study on testing and fault tolerance for software reliability engineering. In: 14th International Symposium on Software Reliability Engineering. (Nov 2003) 119–130
10. Cai, X., Lyu, M.R.: The effect of code coverage on fault detection under different testing profiles. SIGSOFT Softw. Eng. Notes **30**(4) (May 2005) 1–7
11. Falcioni, D., Polini, A., Polzonetti, A., Re, B.: Direct Verification of BPMN Processes through an Optimized Unfolding Technique. In: 12th International Conference on Quality Software (QSIC). (Aug 2012) 179–188
12. Drools, J.: Drools Fusion: Complex Event Processor `http://www.jboss.org/drools/drools-fusion.html`.

13. Learn PAd Project: Model-Based Social Learning for Public Administrations Project. http://www.learnpad.eu/
14. Thönssen, B., Hinkelmann, K., Witschel, F.: Models for setting the wiki. In Thönssen, B., Zhang, C., eds.: Del. D5.1 (The Learn PAd Consortium). (2015)
15. Koetter, F., Kochanowski, M.: A model-driven approach for event-based business process monitoring. In: Business Process Management Workshops, Springer (2013) 378–389
16. Lee, D., Netravali, A., Sabnani, K., Sugla, B., John, A.: Passive testing and applications to network management. In: Proceedings of International Conference on Network Protocols. (Oct 1997) 113–122
17. Yang, Q., Li, J.J., Weiss, D.M.: A survey of coverage-based testing tools. The Computer Journal **52**(5) (2009) 589–597
18. Ali, L., Hatala, M., Gašević, D., Jovanović, J.: A qualitative evaluation of evolution of a learning analytics tool. Computers & Education **58**(1) (2012) 470–489