VALIDITY CHECKING OF PROVENANCE DATA FROM SOFTWARE DEVELOPMENT PROCESSES

Marcela Gomes Pinheiro and Gabriella Castro Barbosa Costa

Federal Center for Technological Education of Minas Gerais (CEFET-MG) - 36700-001 - Leopoldina - MG - Brazil

ABSTRACT

Provenance refers to the origin of data, in other words, a historical record of data. After capturing provenance, it is possible to perform interpretation and analysis of what occurred during the process of data creation and transformation and a diagnosis of the problems that may have occurred during these processes is provided. Among the problems related to the use of data provenance in software development processes there is a definition of how comprehensive the data should be captured and the procedure that should be used for this capture, including checking if the captured provenance data is valid. Considering the definition of the data that should be captured, some provenance models are proposed in literature. As examples, we can cite the PROV model, which can be used to enable the inter-operable interchange of provenance information in heterogeneous environments such as the Web, and PROV-SwProcess model, developed as an extension of the PROV model to capture provenance data from software development processes. This paper brieﬂy details a work in progress about the procedure for capturing provenance data from the software development process, mostly regarding the validity checking of the captured data, using a tool called ProvValidator. In the end, an example composed of a set of data from a software development process is checked through the validation proposal presented in this work.

KEYWORDS

Provenance Data, Software Development Process, Provenance Data Validity

1. INTRODUCTION

In the development of high-quality software, it is necessary to combine two fundamental pillars, which are the good practices of software engineering with an efﬁcient development process. The software development process (SDP) is aimed at creating safe and quality software, through methodologies and actions that will ensure that the basic criteria for creation are met, demonstrating successful software engineering (Pressman, 2019). During the SDP, some data can be captured to be analyzed in the evaluation phase of the process. The analysis of these data can be of two different types: deductive analysis and retrospective analysis (Wolf and Rosenblum, 1993). The last one has as goal by discovering patterns of anomalous behavior that can be eliminated in future enactments of the process. Besides that, and regarding the provenance data to be captured during the SDP, data provenance models can be used.

Provenance refers to the origin or provenance of data, that is, a record of their history, which enables the interpretation and analysis of what occurred during the processes of creation and transformation of data and even the diagnosis of problems that occurred during these processes (Lim et al., 2010). Buneman et al. (2000) deﬁnes data provenance as a complementary documentation that contains the information of how, when, where and why certain data were obtained and who obtained them. Among the problems related to the use of data provenance in SDP there is the deﬁnition of how comprehensive the data should be captured, as well as the procedure to be adopted for this capture, including checking if the provenance data is valid. Related to the deﬁnition of how comprehensive data provenance should be captured, the use of provenance models is an alternative. PROV model (Moreau and Groth, 2013) allows the interoperable exchange of provenance information in heterogeneous environments, such as the web. In turn, the PROV-SwProcess model (Costa et al., 2018), was developed as an extension of the PROV model speciﬁcally to capture provenance data from SDP. This model provides a framework of a more in-depth analysis which actually occurred during software process development (Costa et al., 2018). After capturing the provenance data of SDP, validating whether they are in accordance with the adopted provenance model becomes crucial, in order
to allow all the analysis and possible improvements to be made in the process in question. In this sense, one of the existing tools is ProvValidator (Moreau et al., 2014), a tool that performs model validations that extend from the PROV model.

The main goal of this work is to detail the methodology for checking the validity of the captured data, which consists of the last step of the procedure for capturing data from SDP. Besides that, an example composed of a set of data from a SDP is checked through the validation proposal that will be presented. The remaining of this paper is organized as follows: Section 2 details our procedure for capturing provenance data from SDP; Section 3 presents the proposed methodology for checking the validity of provenance data from SDP and, finally, in Section 4 there is the conclusions followed by acknowledgments and references.

2. CAPTURING PROVENANCE DATA FROM SDP

The systematics presented in Figure 1 was defined considering the need of capturing data from SDP and/or adapting them to a provenance model that should be used. It was an improvement of the systematics proposed on the iSPuP approach (Costa et al., 2018). Considering the activity Define the provenance data to be captured, a set of SDP execution can be stored for each SDP, according to PROV-SwProcess: (i) Executed processes with their name and responsible; (ii) Performed activities of each process, with their name, start, and end time; (iii) Stakeholders associated with the performed activity (mandatory) and their specific role (optional); (iv) Artifacts changed, used, or generated by the performed activity; (v) Procedures adopted for the execution of the performed activity (optional); (vi) Hardware and / or Software resources used by the performed activity (optional); (vii) Responsibility among stakeholders (optional); (viii) Process standard model and process intended model definition (optional). Activities 2, 3, and 4 are automated tasks with additional toll support to the capture, transformation, and storage. Finally, activity 5 Check the validity of provenance data is the focus of this work and is presented in the following section.

![Figure 1. Systematics for SDP provenance data capture and storage](image)

3. METHODOLOGY

In order to Check the validity of provenance data from some SDP (captured or transformed according to PROV-SwProcess) ProvValidator tool was used, according to the following step-by-step and the Figure 2:

1. Go to the site: https://openprovenance.org/services/view/validator
2. To perform validation using a file, simply save it in one of the acceptable formats (xml, ttl,rdf, provn, trig, json). Attach them to the indication [2] and right after clicking the "Validate" button indicated by [1].
3. If the validation to be performed is through URL, simply add in the indication [5] and click the "Validate" button indicated by [1].

4. There is also the option to perform validation through code fragments, just select the desired language in the indication [3], attach the code fragment to [4] and click the "Validate" button indicated by [1].

5. After proper validations, the system will display a screen indicating possible errors or showing the validation of the proposed entry.

![ProvValidator Interface](image)

**Figure 2. ProvValidator Interface**

To illustrate the use of the ProvValidator tool the following example was used. It shows a Software Process identified as New_Resource_Development. This Software Process was composed by the activities New_Resource_Specification, Codification, Test_Cases_Definition, Test, and Deploy (line 10 of Figure 3), and was assigned to the Stakeholder Anna, using the relation wasAttributedTo (line 11 of Figure 3). In lines 13 and 14 of Figure 3, the prospective provenance of this Software Process was established. It is composed by the same five activities listed on its retrospective provenance (New_Resource_Specification, Codification, Test_Cases_Definition, Test, and Deploy) and Anna stakeholder is the process responsible. In this example, there were no differences between what was planned and what was actually executed. Finally, Figure 4 shows the validation result performed on the tool.

![Example](image)

**Figure 3. Example**

A similar work to the one proposed in this article is Blinker (Bose et al., 2019), a framework that allows a hierarchical visualization of provenance graphs and query results. One difference between the model proposed in this article and Blinker is the number of tools and processes that would be necessary to adequately capture data provenance. Another related work that can be mentioned is DQProv Explorer (Bors et al., 2019), a system that captures provenance through conflicting data operations and allows them to be visualized in graph form. Both use extensions of the PROV model and assume forms of model validation test of different models than the proposal in this paper, where a validator from PROV itself is used.
4. CONCLUSION

This paper details a systematics for capturing provenance data from the SDP, specifically regarding the validity checking of the captured data. It uses a tool called ProvValidator and shows how to check the validity of an example composed of a set of data from a real SDP. As a work in progress, our next steps mainly focus on improving the process to check the validity of provenance data with more automation of this process, obtaining the ontology with the data and performing its verification through a ProvValidator API, without the need to perform this task manually, accessing the tool via a web application.

ACKNOWLEDGEMENT

We want to thank CEFET-MG and CNPq for financial support.

REFERENCES


