A Software Framework for Data Provenance

Tassio Sirqueira[1], Marx Viana[1], Nathalia Nascimento[1] and Carlos Lucena[1]
[1]Pontifical Catholic University of Rio de Janeiro
Software Engineering Laboratory (LES)
Rio de Janeiro, RJ – Brazil
{tmartins, mleles, nnascimento, lucena}@inf.puc-rio.br

Abstract— **Data provenance refers to the historical record of the derivation of the data, allowing the reproduction of experiments, interpretation of results and identification of problems through the analysis of the processes that originated the data. Data provenance contributes to the evaluation of experiments. This paper presents a framework for data provenance using the W3C provenance data model, called PROV-DM. Such framework aims at contributing to, and facilitating, the collection, storage and retrieval of provenance data through a modeling and storage layer based on PROV-DM, yet is compatible with other representations of PROV such as PROV-O. To demonstrate the utilization of the framework, it was used in an IoT application that performs the gas classification to identify diseases.**

Keywords- Data Provenance; PROV Framework; PROV W3C.

I. INTRODUCTION

The term "provenance" refers to the origin or provenance of data, that is, it is a record of the data derivation history, which enables reproducibility of experiments, interpretation of results and diagnosis of problems [1]. The provenance is the complementary documentation of a data, containing information of "how," "when," "where" and "why" the data was obtained, "who" got it and "how much" cost this in time or effort.

The provenance provides a look beyond the specifications of domains and suggests the adoption of disciplined models, where data provenance information can be used to learn or understand design methods and rules. It can further assist users in similar investigations in order to understand data correlations and to improve future investigations. Currently, the provenance of data is successfully applied in different areas, mainly e-science [2].

One of the problems of provenance refers to the lack of agreement as to the comprehensiveness of the data to be captured, in addition to the absence of a clear definition of how this procedure should be carried out [7]. Other issues raised with respect to the provenance of data are reliability of data, integrity, confidentiality about its use, availability to other people, beyond efficacy in relation to what is being captured and the efficiency with which this is done, ensuring that all relevant information is captured.

The goal of this section is to discuss the benefits that the data provenance can bring, considering the captured data, how they were modeled and stored, and the type of information to be obtained from them. In this context, we present a framework for data provenance (FProvW3C).

DOI reference number: 10.18293/SEKE2017-085

To illustrate the use of FProvW3C, we will present an example from a system based on the Internet of Things (IoT) [13]: a system that collects data from gas sensors and assists in the classification of gases emitted by humans. We have chosen this example because IoT is an exciting and emerging approach that has gained both academic and industrial attention.

The remainder of this paper is organized as follows. Section II presents the W3C PROV [3] model. Section III discusses related work. Section IV details the FProvW3C and Section V presents the framework instance. Finally, Section VI presents our conclusions and future work.

II. PROV W3C

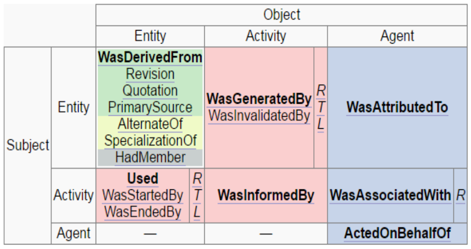
According to [4], the data provenance can be divided into three types: i) Prospective: it is the sequence of processes used in data generation; ii) Retrospective: this is the information obtained during the execution of data and environment generation processes, and iii) User data: any information that the user deems necessary for future analysis. In addition, the provenance can be obtained in two ways according to [8], which are: i) Lazy: the provenance is obtained from the moment that its capture is requested, and ii) Anxious: provenance is obtained at all times and is readily available. The best way to collect it will depend on the application to be used.

Currently, there are two main patterns for data capture from provenance: i) the OPM model [5], with three vertices, five causal relationships, and ii) the PROV model [3], with three main vertices and seven basic relations, plus complementary ones. In this work, the provenance model used was the PROV [3] by its amplitude and greater number of causal relations for knowledge representation.

The PROV [3] model consists of 12 documents that define their specification. Among the main documents are the PROV-DM, which specifies the data capture model; the PROV-CONSTRAINTS, which is the set of constraints applicable to the data model (PROV-DM), and the PROV-O, an ontology for mapping the data model.

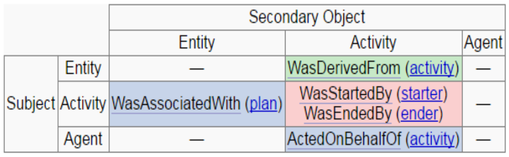
The PROV-DM creates a separation of types and causal relations, the types being: i) Entity: it is either a physical, digital or conceptual type, or something with fixed aspects, in that entities can be real or imaginary. ii) Activity: it is something that occurs over a period time and acts on entities. iii) Agent: it is something that has some kind of responsibility for the activity and the existence of an entity, or for the activity of another agent. Agent, in the PROV model, can be classified as an organization, a person, or a software agent.

In relation to causal relations they are divided into two subsets: primary and secondary (optional) relations. Fig. 1 shows the primary relations in bold and Fig. 2 presents the (optional) secondary relations.



		Object		
		Entity	Activity	Agent
Subject	Entity	**WasDerivedFrom** Revision Quotation PrimarySource AlternateOf SpecializationOf HadMember	**WasGeneratedBy** WasInvalidatedBy R T L	**WasAttributedTo**
	Activity	**Used** WasStartedBy WasEndedBy R T L	**WasInformedBy**	**WasAssociatedWith** R
	Agent	—	—	**ActedOnBehalfOf**

Figure 1. Primary relations of the PROV[1].



		Secondary Object		
		Entity	Activity	Agent
Subject	Entity	—	WasDerivedFrom (activity)	—
	Activity	WasAssociatedWith (plan)	WasStartedBy (starter) WasEndedBy (ender)	—
	Agent	—	ActedOnBehalfOf (activity)	—

Figure 2. Secondary relations of the PROV[2].

One of the advantages of using the PROV and the PRV-O ontology is that PROV-DM can be represented by using OWL2 (Web Ontology Language) [6]. They can also be used to represent and exchange information of provenance generated in different systems and in different contexts.

In addition, another advantage of using the PROV model is with respect to the storage model, where different sources of information are converted into a model standardized by the W3C. This in turn facilitates the understanding, traceability and reproducibility of a data, due to the process that originated it. Furthermore, it allows semantic annotation using the PROV-O.

The provenance may be an important quality metric in the experiment, since the data derivation process has implications for both data quality and the errors introduced by faulty data as they propagate in other derivations [9]. Provenance in the experimentation process can help increase the validity of the experiments, since the reliability of the data will be monitored.

The main objective of the FProvW3C framework is to simplify the capture of provenance data and to facilitate the use of the PROV model, thus allowing for more reliable experiments.

III. Related Work

Provenance can be used as a quality metric for data evaluation because it has significant implications on data quality and errors introduced by faulty data, which increase as derivations are propagated [9]. To support the research developed in this article, a structured search was carried out. Among the results were selected studies that are applied explicitly to the data provenance based on the PROV model.

Starting with ProvToolbox[3] is a Java library to create PROV-DM representations and convert them between RDF, PROV-XML, PROV-N, and PROV-JSON. It is not geared towards capturing and storing data in a DBMS (Database Management System), moreover, is a set of independent tools for each form of representation of the data.

The prov-api[4] is a Java API to create and manipulate provenance graphs. Currently, API only implements PROV's essential terms. The focus of the prov-api is the inference and query in the PROV-O ontology and not in the data storage using the PROV-DM, as well as its use in data capture from provenance.

The PROV Python library[5] is a library that provides an implementation of PROV-DM in Python. Although it is a library close to the concerns of the framework of this article, being in python makes it difficult to capture data coming from multi-agent systems, since most agent platforms are in Java.

The E-SECO ProVersion presented by [2], is a management platform for scientific workflows. Although the application works with provenance data in the PROV-DM model, it uses a lazy approach to data capture. In addition, the PROV model is integrated to the platform code, making it difficult to reuse. The ProvManager, presented by [7], is a data storage and analysis tool, uses prolog for queries and, like E-SECO, does not have mechanisms for integration with other systems, depending on a particular form of data entry.

For his part [10], presents what he later called Prov-Process, a platform for collection, storage and analysis of provenance data. However, a standard model must be used for data entry in the ".csv" format and does not allow integration with other systems.

Although there are several applications aimed at the provenance of data, they do not have features that assist users in data capture and storage. In the next section the details of the framework will be presented.

IV. FProvW3C – A Framework For Data Provenance

As addressed by [2], the data provenance is something constant, and it should follow all the steps performed to obtain concise results. One way to observe this is to consider the life cycle of a data, where not only the data is important but also the process that originated it.

The FProvW3C Framework works with an anxious approach [8], so that the data is collected at all times and can be consulted next. Currently, in its architecture the FProvW3C framework presents all the specification of the PROV-DM with the annotations in Java Persistence API (JPA), according to Fig. 3.

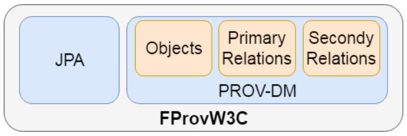


Figure 4. Framework architecture.

[1, 3] PROV-DM: http://www.w3.org/TR/prov-dm/

[3] http://lucmoreau.github.io/ProvToolbox/

[4] https://github.com/dcorsar/prov-api/

[5] http://pypi.python.org/pypi/prov

Besides modeling, the FProvW3C framework brings all persistence annotations. This reduces the work of users and avoids mapping errors by those who do not have extensive knowledge of the PROV. Being a framework makes it possible to integrate it into different systems. The framework frozen-spots are the main vertices and the hot-spots are the causal relations, both defined by the PROV model, and they adapt to different application contexts. The framework was developed in Java and the annotations are based on Java Persistence API (JPA), which makes the framework independent of the DBMS that will be applied to store the collected data that leaves this choice up to the user.

The FProvW3C is relational object mapping framework in charge of creating the database, the tables and their respective attributes in the DBMS. The PROV model determines the classes and how they relate in addition to the basic attributes. In this way, the framework provides the classes with the basic attributes (frozen-spots) and, moreover, allows the creation of hot-spots by extending both the attributes of each class and the relationships. These attributes are intended to represent the characteristics of the system in which FProvW3C will be applied. As the framework performs data mapping according to the PROV, its use is made easier for the user, where are able to apply the data provenance in their applications.

V. Usage Scenario: Gas Analysis

As described by [11], a person emits various gases from different parts of the body (e.g. flatulence, eructation, exhalation) and these gases could be useful for the diagnosis of a set of intestinal and stomach diseases.

Nevertheless, there are very few technology approaches to facilitate the analysis of flatulence and other gases daily emitted by humans. In addition, there is a need for high data reliability in these approaches, since it uses the identification of gas-based diseases, i.e., all information must be reliable and agents cannot fail to capture or classify data.

The FProvW3C framework was used in the "Gases Device" application [11]. This application is based on the Internet of Things (IoT), which uses sensors to measure gases in the environment and uses software agents to provide data classification. Fig. 7 shows the Gases Device application architecture extending FProvW3C classes.

A. Overview

FProvW3C creates an intermediary layer between the application and the database. This layer makes it possible to treat and map the data to be stored by linking the information source to them. In addition, FProvW3C creates a knowledge base based on the use of the application. Therefore, all data entered by users or sensors into the application, as well as data manipulated by the application are registered in this base.

Fig. 6 shows that every time a gas sensor captures the variation of a gas, the persistence of the data in the database is invoked via the FProvW3C framework, registering data provenance. In this way, all the capture and manipulation of data are recorded, such as a filming, forming the historical basis of the application.

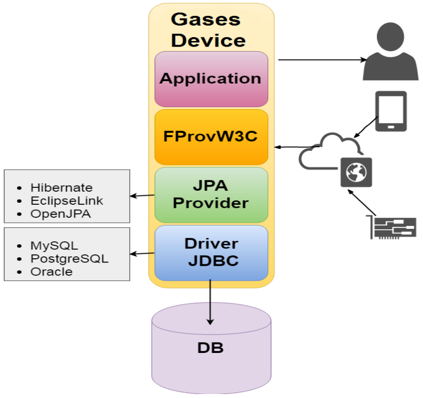
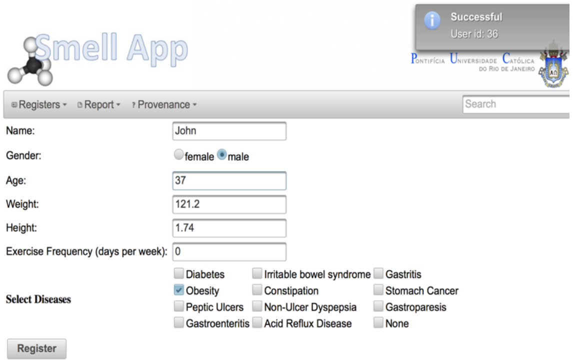


Figure 6. Gases Device architecture with FProvW3C.

B. Results and Discussion

The recorded data of this application includes information from system users, sensors, monitored gases and software agents [11]. The recorded data were linked to the actions of users and software agents, helping in the traceability of each executed action. For example, Fig. 7 illustrates the registration moment of a user with obesity in the Smell App Website. To elaborate an initial database to improve the system's classification, first users were asked to inform personal health information before using the Gases Device. As shown in Fig. 8, the system attributed the ID 36 to this registered user. Then, by using the FProvW3C's structure, it was possible to track all data generated during this registration operation. We can verify in Fig. 8 that the person agent with User ID 36 was successfully created, and the entity "Obesity" was attributed to this agent.



Successful
User id: 36

Name: John

Gender: ○ female ● male

Age: 37

Weight: 121.2

Height: 1.74

Exercise Frequency (days per week): 0

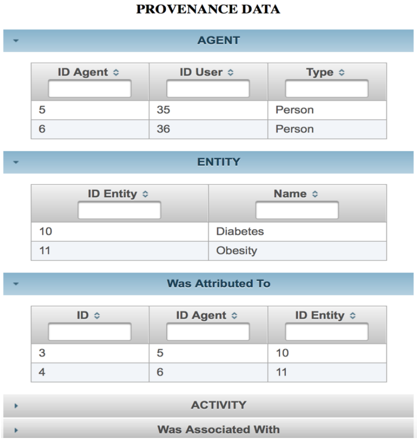
Select Diseases

□ Diabetes	□ Irritable bowel syndrome	□ Gastritis
☑ Obesity	□ Constipation	□ Stomach Cancer
□ Peptic Ulcers	□ Non-Ulcer Dyspepsia	□ Gastroparesis
□ Gastroenteritis	□ Acid Reflux Disease	□ None

Figure 7. Smell App Screenshot.

After the user had registered, he/she connected a Gases Device to the system and started an exhalation report. This action initiated three software agents: i) Gases Device Agent, which collected data from Arduino; ii) Analyzer Agent, which preprocessed and saved data on the database, and iii) Alert Agent, which evaluated all exhalation reports based on the diseases described in [11] and generated alerts.

To evaluate the operation of a multi-agent system is not a trivial task [12]. As data come from several sources, it is difficult to identify the origin of a problem. However, as shown in Fig. 9, the provenance facilitated the evaluation of this multi-agent system by allowing us to track all activities that were performed during its execution.



PROVENANCE DATA

AGENT

ID Agent	ID User	Type
5	35	Person
6	36	Person

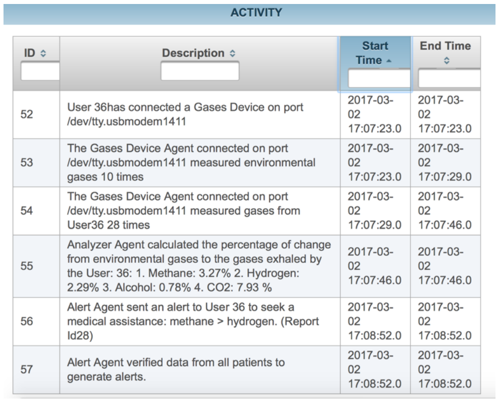
ENTITY

ID Entity	Name
10	Diabetes
11	Obesity

Was Attributed To

ID	ID Agent	ID Entity
3	5	10
4	6	11

Figure 8. Provenance of Smell App Data.



ACTIVITY

ID	Description	Start Time	End Time
52	User 36has connected a Gases Device on port /dev/tty.usbmodem1411	2017-03-02 17:07:23.0	2017-03-02 17:07:23.0
53	The Gases Device Agent connected on port /dev/tty.usbmodem1411 measured environmental gases 10 times	2017-03-02 17:07:23.0	2017-03-02 17:07:29.0
54	The Gases Device Agent connected on port /dev/tty.usbmodem1411 measured gases from User36 28 times	2017-03-02 17:07:29.0	2017-03-02 17:07:46.0
55	Analyzer Agent calculated the percentage of change from environmental gases to the gases exhaled by the User: 36: 1. Methane: 3.27% 2. Hydrogen: 2.29% 3. Alcohol: 0.78% 4. CO2: 7.93 %	2017-03-02 17:07:46.0	2017-03-02 17:07:46.0
56	Alert Agent sent an alert to User 36 to seek a medical assistance: methane > hydrogen. (Report Id28)	2017-03-02 17:08:52.0	2017-03-02 17:08:52.0
57	Alert Agent verified data from all patients to generate alerts.	2017-03-02 17:08:52.0	2017-03-02 17:08:52.0

Figure 9. Provenance of Activities of Software Agents and Users.

This section shows that data provenance can be used for a wide range of purposes in computational applications. Furthermore, when we tracked the system execution in order to understand its operation, we also used data provenance to verify errors in data classified by agents and in data collected from sensors.

VI. Conclusion and Future Work

Registering the data provenance is a necessity in several scenarios, especially those that have complex execution. It is necessary to have a history of each of the steps. The PROV model aims at storing data provenance in a detailed manner, focusing on the responsibilities of agents in each item of provenance.

This paper proposes the FProvW3C framework, designed to capture and store data provenance using the PROV model of W3C. The data provenance helps to trace the origin of the data and the derivation processes that occurred between the origin of the data and the state in which the data is currently found. Considering that the provenance model contributes to evaluate the quality of the data and consequently the process that generated it, this helps increase the validity of the experiments since the data is monitored. The data provenance could be used in the construction of knowledge bases that help in the: i) traceability of actions; ii) identification of errors; iii) follow-up of the steps of a study, and iv) viability of verify results.

For future work, we hope to expand the FProvW3C framework so that it can convert the captured data to an ontology following the PROV-O model. For example, it is possible to extend FProvW3C to support: i) data semantics and syntax, and ii) the ontology of PROV. In addition, we aim at exploring the data provenance in multi-agent systems, recording each piece of information about the agent, its relations with other agents and with the external environment. As such, we could capture information from the agent's decision-making process, expanding the work of [11] and we could use the information obtained to help track errors and answer questions about the agent's behavior, since it is an autonomous entity.

Acknowledgment

The authors thank CAPES, Faperj, CNPq, PUC-Rio and LES for the support, financial resources and encouragement to the research.

References

[1] Lim, C., Lu, S., Chebotko, A., & Fotouhi, F. (2010). Prospective and retrospective provenance collection in scientific workflow environments. In 2010 IEEE International Conference on Services Computing. p. 449-456. IEEE.

[2] Sirqueira, T. F., Dalpra, H. L., Braga, R., Araújo, M. A. P., David, J. M. N., & Campos, F. (2016). E-SECO ProVersion: An Approach for Scientific Workflows Maintenance and Evolution. Procedia Computer Science, 100, 547-556.

[3] Missier, P., Belhajjame, K., & Cheney, J. (2013). The W3C PROV family of specifications for modelling provenance metadata. In Proceedings of the 16th International Conference on Extending Database Technology. p. 773-776. ACM.

[4] Davidson, S. B., & Freire, J. (2008). Provenance and scientific workflows: challenges and opportunities. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data. p. 1345-1350. ACM.

[5] Moreau, L., Freire, J., Futrelle, J., McGrath, R. E., Myers, J., & Paulson, P. (2008). The open provenance model: An overview. In International Provenance and Annotation Workshop. p. 323-326. Springer Berlin Heidelberg.

[6] Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., & Zhao, J. (2013). Prov-o: The prov ontology. W3C Recommendation, 30.

[7] Marinho, A., Werner, C. M. L., & Murta, L. G. P. (2009). ProvManager: uma abordagem para gerenciamento de proveniência de workflows científicos. In Workshop de Teses e Dissertações em Engenharia de Software, XXIII SBES (Vol. 14). (in portuguese)

[8] Tan, W. C. (2004). Research Problems in Data Provenance. IEEE Data Eng. Bull., v. 27(4), p. 45-52.

[9] Veregin, H., & Lanter, D. P. (1995). Data-quality enhancement techniques in layer-based geographic information systems. Computers, Environment and Urban Systems, v. 19(1), p. 23-36.

[10] Dalpra, H. L., Costa, G. C., Sirqueira, T. F., Braga, R., Werner, C. M., Campos, F., & David, J. M. N. (2015). Using Ontology and Data Provenance to Improve Software Processes. In Proceedings of the Brazilian Seminar on Ontologies (pp. 10-21).

[11] Nascimento, N. M., Viana, M. L., Lucena, C. J. P. (2016) An IoT-based Tool for Human Gas Monitoring, XV Congresso Brasileiro de Informática em Saúde, p. 96-98.

[12] Coelho, R., Cirilo, E., Kulesza, U., von Staa, A., Rashid, A., & Lucena, C. (2007). Jat: A test automation framework for multi-agent systems. In IEEE International Conference on Software Maintenance. p. 425-434.

[13] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, v.29(7), p. 1645-1660.

[14] Doukas, C. (2012). Building Internet of Things with the ARDUINO. CreateSpace Independent Publishing Platform.