Trusted Provenance of Collaborative, Adaptive, Process-Based Data Processing Pipelines

Ludwig Stage(✉)

Information Systems Group, University of Groningen, Groningen, The Netherlands
l.stage@rug.nl

Abstract. The abundance of data nowadays provides a lot of opportunities to gain insights in many domains. *Data processing pipelines* are one of the ways used to automate different data processing approaches and are widely used by both industry and academia. In many cases data and processing are available in distributed environments and the workflow technology is a suitable one to deal with the automation of data processing pipelines and support at the same time collaborative, trial-and-error experimentation in term of pipeline architecture for different application and scientific domains. In addition to the need for flexibility during the execution of the pipelines, there is a lack of trust in such collaborative settings where interactions cross organisational boundaries. Capturing provenance information related to the pipeline execution and the processed data is common and certainly a first step towards enabling trusted collaborations. However, current solutions do not capture change of any aspect of the processing pipelines themselves or changes in the data used, and thus do not allow for provenance of change. Therefore, the objective of this work is to investigate how provenance of workflow or data change during execution can be enabled. As a first step we have developed a preliminary architecture of a service – the Provenance Holder – which enables *provenance of collaborative, adaptive data processing pipelines in a trusted manner*. In our future work, we will focus on the concepts necessary to enable trusted provenance of change, as well as on the detailed service design, realization and evaluation.

Keywords: Provenance of Change · Reproducibility · Trust · Collaborative Processes · Data Processing Pipelines · Workflow evolution provenance · Provenance of ad-hoc workflow change

1 Introduction and Motivation

Data-driven research and development in and for enterprises is currently one of the most investigated topics with specific focus on data analysis, simulations, machine learning algorithms and AI. In the scope of such initiatives, both, academic and industrial research and development in different domains show great

T. P. Sales et al. (Eds.): EDOC 2023 Workshops, LNBIP 498, pp. 363–370, 2024.
https://doi.org/10.1007/978-3-031-54712-6_25

effort in automation and deployment of data processing in enterprise computing environments in order to leverage operational improvement opportunities and to profit from the available data.

Automation of computations and data processing is done by using data processing pipelines. One major challenge of this automation is the identification of the best approach towards the actual automation of such pipelines since they can be implemented using different methodologies and technologies. Furthermore, integration of computational resources, the ability to use data from different sources in different formats and varying quality properties, the flexibility of data pipelines, the modularity and reusability of individual steps, the ability to enable collaborative modelling and execution of data processing pipelines, as well as their provenance and reproducibility are hard requirements. Consequently, there are a lot of research results in literature on the application of different technologies and concepts in different domains such as eScience, scientific computing and workflows, data science, intelligent systems, business processes, etc.

The topic of *provenance*[1] has been researched predominantly in the field of scientific experiments and scientific workflows, which led to the definition of the characteristics of Findable Accessible Interoperable Reusable (FAIR) results [1,2] and Robust Accountable Reproducible Explained (RARE) experiments [1]. In this field, scientific experiments are considered to be of good provenance if they are reproducible. Enabling reproducibility of experiment results, typically by means of tracking the data through all processing, analysis and interpretation steps of the experiment, has been one of the main objectives of scientific workflow systems, in addition to the actual automation of scientific experiments. The importance of provenance in in-silico experiments has been identified and discussed and approaches have been partly implemented more recently in e.g. [3–6] and are relevant to enabling the provenance of data processing pipelines. Furthermore, there are initiatives towards standardization of representing provenance information for the purposes of both modeling provenance information and establishing an interchangeable format for such information, e.g. PROV-DM[2].

To the best of our knowledge, the ability to reproduce the changes on either workflow or choreography models or instances made by collaborating organisations in the course of running their data processing pipelines in a trusted manner, has not been the subject of other works. Towards closing this gap in research, we propose a solution [7], called *Provenance Holder service*, that has to track and record all changes made on choreography and/or workflow models or instances to support their provenance in a trusted manner and allow collaborating organisations to retrace and reproduce their data processing pipelines exactly the same way as they have been carried out, including all changes made on both data and software used during the execution.

The contributions we intend with this work are: (i) A workflow provenance taxonomy to account for adaptation based on existing taxonomies from literature, (ii) Identification of provenance requirements for the Provenance Holder

[1] "The provenance of digital objects represents their origins."[2].

[2] https://www.w3.org/TR/prov-primer/.

service, (iii) Detailed definition of the properties of the Provenance Holder service that will guarantee trusted provenance of collaborative, adaptive data processing pipelines, (iv) functional architecture, which is generic in nature and applicable in any application domain and is easy to integrate with other flexible Management System (WfMS) systems, (v) concepts and data structures necessary for capturing the adaptations, and (vi) an implementation as a proof of concept and its evaluation. We also intend to explicitly identify (vii) the *prerequisites* for employing the Provenance Holder with other WfMS environments, namely the ability to support the trial-and-error manner of experimenting (as in e.g. Model-as-you-go-approach [8] or ability to change and propagate change in choreographies [9]) and the ability to provide workflow monitoring data that allows for data and workflow provenance in a trusted manner [7].

2 Scope and Research Questions

In the scope of our work are automated data processing pipelines, which use only software implementations of computational and data transformation tasks and excludes data processing pipelines in which participation of physical devices (e.g. microscopes, wet labs, sensors and actuators) is directly visible in the pipeline. We aim at enabling the *provenance of flexible, a.k.a. adaptive, data processing pipelines that are carried out in collaboration* among identifiable organisational entities. The matter of *trust among the collaborating parties* is of utmost importance in the context of our work, in particular because of the need to capture the origins of change that can be carried out by any of the participating parties at any point in the execution of the pipelines.

Our technology of choice for modelling and running collaborative data processing pipelines is *service-based, adaptable processes, both workflows and choreographies*, that are well known from the field of Business Process Management (BPM) [10] and conventional Workflow Management Technology [11] and for their beneficial properties such as modularity, reusability, interpretability, transactional support, scalability and reliability.

We have identified four requirements on the Provenance Holder [7,12] in order to be enable reproducible, trusted and adaptive collaborations (cf. Table 1).

Table 1. Provenance Holder Requirements adopted from [7,12]

Requirement	Description
R1	**Adaptability** to adhere to the adaptability of experiments
R2	**Provenance** to enable FAIR results [1]
R3	**Reproducibility** for RARE experiments [1]
R4	**Trust** among collaborating parties to also enable accountability

To the best of our knowledge, the ability to reproduce the changes on either workflow or choreography models or instances made by collaborating organisations in the course of running their data processing pipelines in a trusted manner, has not been the subject of other works. We call this type of provenance "*trusted provenance of change*". Based on the existing taxonomies for provenance as summarized by [4], to accommodate the provenance of adaptation, we identified which new types of provenance need to be considered (cf. Fig. 1).

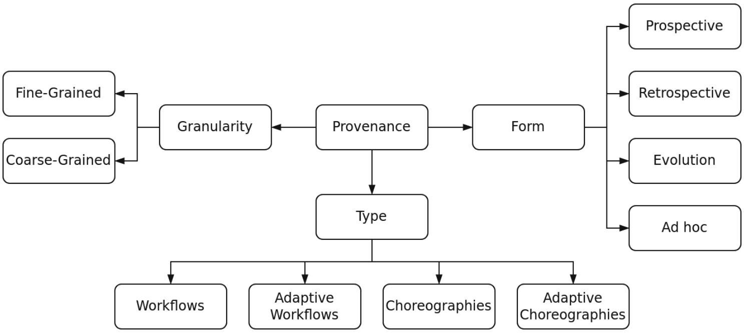

Fig. 1. Workflow Provenance types taxonomy adopted from [4] and [13]

With our work we aim to answer four research questions (cf. Table 2).

Table 2. Research questions

RQ1	How can we bring traceability, reproducibility, accountability and trust to Automated, Collaborative and Adaptive, Process-based Data Processing Pipelines?
RQ2	What does a system look like that provides traceability, reproducibility, accountability and trust for Automated, Collaborative and Adaptive, Process-based Data Processing Pipelines?
RQ3	What are the requirements on such a system?
RQ4	What are the prerequisites for the environment such a system is to be integrated?

3 Provenance Holder Properties and Architecture

The Provenance Holder is a service responsible for collecting all information necessary to ensure provenance and reproducibility of and trust in the collaborative adaptations and enabling the four properties (cf. Table 3). We aim at providing a generic, reusable and non-intrusive solution across different scenarios and separation of concerns [14]. We realize P1 via electronic signature, P2 with (trusted) timestamping, we will investigate how P3 can be enabled using non-interactive zero knowledge proofs ([15], as it presents a systematic overview over the greater topic of verifiable privacy-preserving computations), and P4 by linking provenance information objects.

Table 3. Provenance Holder Properties and their mapping to statements made by choreography participants. In the statement column the pronoun **It** is information about either of the following: result, origin/predecessor or change. The text in bold highlights where the focus of each property lies. Adopted from [13].

Property	Statement by participant	Description
P1	"**I** know it"	A result/change/predecessor can be attributed to a certain identifiable entity, i.e. choreography participant
P2	"I knew it **before**"	A result/change/predecessor has been available/known or has happened at or before a certain point in time
P3	"I **actually** know it"	Prove that participants know of a result/change/predecessor (without information disclosure)
P4	"I know **where it came from**"	Participants have knowledge of the predecessor of a result/change/ predecessor

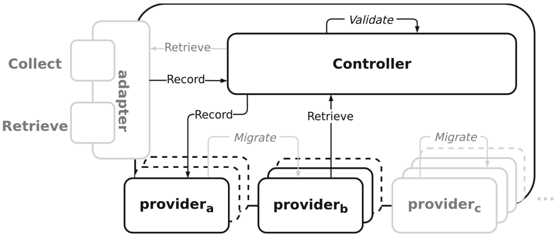

Fig. 2. Provenance Holder Architecture: components, external operations and internal methods, implemented ones are solid black (adopted from [13])

The Provenance Holder service provides *two main operations* as part of its interface (cf. Fig. 2): 1) ***Collect*** provenance data and 2) ***Retrieve*** provenance information; we call these operations also external operations. The controller, the adapter and one or more provenance providers are the *components of the Provenance Holder* and they carry out four *interaction scenarios* in order to realize the two externally provided operations of the Provenance Holder service. The interaction scenarios are always combinations of several of the internal methods[3]; the (internal) methods are: *Record*, *Retrieve*, *Validate* and *Migrate*.

[3] We use the term *method* for disambiguation purposes only.

The *adapter* is the component ensuring the *integration* of the Provenance Holder with other systems and provides the two external operations: *Collect* and *Retrieve*. Its actual design and implementation are specific for the system with which it has to work to enable the integration and correct communication.

Providers or *provenance providers* have to implement three methods, *record, retrieve and migrate*, certain requirements to fulfil, and ultimately store the provenance information. The implementation characteristics and complexity strongly depend on the employed (storage) technology and the needs of different workflow types also come into play when deciding which technology to use.

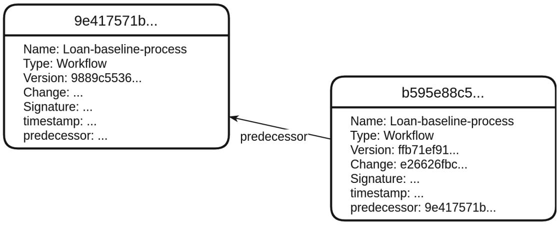

Fig. 3. Provenance Information objects representing a change [13]

The *controller* is in charge of the interaction between the adapter and the provenance providers so that the Provenance Holder can provide the provenance service operations to each workflow and choreography. The controller combines the four methods: record, validate, retrieve and migrate into the realization of the two operations provided by the Provenance Holder: Collect and Retrieve. For the *collect provenance data operation* the controller receives, validates and relays the provenance information to the providers. For the *operation retrieve provenance information*, it combines the methods retrieve and validate. Data structures to capture and store provenance information, e.g. of change, had to be defined and supported by all components (cf. Fig. 3). During the execution and adaptation of workflows and choreographies the Provenance Holder constantly collects provenance data on a very detailed level, including on per-workflow-activity level. The *Record* method selects appropriate provider components for a certain workflow type out of the available providers and uses them to store the provenance information. Data is validated (with the validation method) before it is actually handed over to a provider for storage. The *Retrieve* method is used to fetch the desired provenance information from the provider components via their interfaces. The actual data retrieval is done by each provider itself and returned to the *retrieve* method. After retrieval, the information is validated before it is handed over to the adapter component, i.e. the Provenance Holder's interface implementation. The *validation* method is called during *Recording* to verify the signature and identify the signee the data is "recorded". If the signature verification fails due to an invalid signature or an unknown signee, the information will not be "recorded". When calling the *Retrieve* method, the provenance information is fetched from the provenance provider and then validated. The *Migrate* method is only used if stored information has to be transferred to a new provider type or instance, in case such an addition or change is desired/needed and provides the ability to retrieve all stored provenance information from a provider at once. Migrations can be triggered both automatically or manually by an administrator; the actual procedure for migration is out of scope of our work as related work like [16] is available.

4 Conclusions and Future Work

The goal of this work is to support trusted provenance in collaborative, adaptive, process-based data processing pipelines. We currently provide the concepts of capturing provenance of change in such pipelines as well as the architecture of the corresponding system, including the detailed design of the controller and provider components of the Provenance Holder. The prototypical implementation of these components and properties P1, P2 and P4 is available at https://github.com/ProvenanceHolder/ProvenanceHolder.

Refining the concepts, identification of the adapter requirements, its detailed architecture and implementation are the future steps in our research. This also includes the identification and differentiation of change, its capturing and visualisation, how changes are communicated and all supporting data structures. Subsequently we will work towards the evaluation and extension of both, the approach and the proof-of-concept implementation.

We recognise that the approach and its realization is not only applicable to Scientific Workflows, but also to Workflows and to process-based data processing pipelines in general. Furthermore, we do not rule out the possibility that the approach may also go beyond this. These are two of the reasons why we will be following a generic research path and at the same time we have a specific use case.

References

1. Mesirov, J.P.: Accessible reproducible research. Science **27**, 415–416 (2010)
2. Wilkinson, M., et al.: The fair guiding principles for scientific data management and stewardship. Sci. Data **3**, 1–9 (2016)
3. Atkinson, M., et al.: Scientific workflows: past, present and future. Future Gener. Comput. Syst. **75**, 216–227 (2017)
4. Herschel, M., et al.: A survey on provenance - what for? what form? what from? Int. J. Very Large Data Bases (VLDB J.) **26**, 881–906 (2017)
5. Alper, P., et al.: Enhancing and abstracting scientific workflow provenance for data publishing. In: Proceedings of the Joint EDBT/ICDT Workshops (2013)
6. Freire, J., Chirigati, F.S.: Provenance and the different flavors of reproducibility. IEEE Data Eng. Bull. **41**, 15 (2018)
7. Stage, L., Karastoyanova, D.: Provenance holder: bringing provenance, reproducibility and trust to flexible scientific workflows and choreographies. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) BPM2019. LNBIP, vol. 362, pp. 664–675. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37453-2_53

8. Sonntag, M., Karastoyanova, D.: Model-as-you-go: an approach for an advanced infrastructure for scientific workflows. J. Grid Comput. **11**, 553–583 (2013)
9. Fdhila, W., et al.: Dealing with change in process choreographies: design and implementation of propagation algorithms. Inf. Syst. **49**, 1–24 (2015)
10. Weske, M.: Business Process Management - Concepts, Languages, Architectures, 3rd edn. Springer, Heidelberg (2019)
11. Leymann, F., Roller, D.: Production Workflow: Concepts and Techniques. Prentice Hall PTR, Hoboken (2000)
12. Karastoyanova, D., Stage, L.: Towards collaborative and reproducible scientific experiments on blockchain. In: Matulevičius, R., Dijkman, R. (eds.) CAiSE 2018. LNBIP, vol. 316, pp. 144–149. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92898-2_12
13. Stage, L., Karastoyanova, D.: Trusted provenance of automated, collaborative and adaptive data processing pipelines (2023). https://doi.org/10.48550/arXiv.2310.11442. Accessed 26 Nov 2023
14. Dijkstra, E.W.: On the role of scientific thought. In: Selected Writings on Computing: A Personal Perspective. Texts and Monographs in Computer Science. Springer, New York (1982). https://doi.org/10.1007/978-1-4612-5695-3_12
15. Bontekoe, T., Karastoyanova, D., Turkmen, F.: Verifiable privacy-preserving computing (2023). https://doi.org/10.48550/arXiv.2309.08248. Accessed 13 Oct 2023
16. Strauch, S., et al.: Migrating enterprise applications to the cloud: methodology and evaluation. Int. J. Big Data Intell. **5**, 127–140 (2014)