Wellington Moreira de Oliveira

INTEGRATED ANALYSIS OF HETEROGENEOUS PROVENANCE GRAPHS

Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Doctor of Science. Area: SYSTEMS AND INFORMATION ENGINEERING.

Advisor: Prof. D.Sc. Vanessa Braganholo
Co-Advisor: Prof. D.Sc. Daniel de Oliveira

Niterói
2018

Ficha catalográfica automática - SDC/BEE

M835i Moreira de Oliveira, Wellington
Integrated Analysis of Heterogeneous Provenance Graphs / Wellington Moreira de Oliveira ; Vanessa Braganholo, orientadora ; Daniel De Oliveira, coorientadora. Niterói, 2018.
114 f. : il.

Tese (doutorado)-Universidade Federal Fluminense, Niterói, 2018.

1. Análise integrada de proveniência. 2. Interoperabilidade de dados de proveniência. 3. Workflow científico. 4. PROV e ProvONE. 5. Produção intelectual. I. Título II. Braganholo,Vanessa, orientadora. III. De Oliveira, Daniel, coorientadora. IV. Universidade Federal Fluminense. Escola de Engenharia.

CDD -

Bibliotecária responsável: Fabiana Menezes Santos da Silva - CRB7/5274

WELLINGTON MOREIRA DE OLIVEIRA

INTEGRATED ANALYSIS OF HETEROGENEOUS PROVENANCE GRAPHS

Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Doctor of Science. Area: SYSTEMS AND INFORMATION ENGINEERING.

Approved in April of 2018.

BANCA EXAMINADORA

Prof. D.Sc. Vanessa Braganholo – Advisor - UFF

Prof. D.Sc. Daniel de Oliveira – Co-Advisor - UFF

Prof. D.Sc. Alexandre Plastino de Carvalho - UFF

Prof. D.Sc. Luiz André Portes Paes Leme - UFF

Prof. D.Sc. Marta Mattoso - UFRJ

Prof. D.Sc. Maria Claudia Reis Cavalcanti - IME

Niterói

2018

ACKNOWLEDGMENTS

First of all, I would like to thank God for helping me in those moments I thought that it was impossible to complete this task. Thank Holy Spirit for clarifying and guiding my mind and my steps in this journey.

I would like to thank my professors and advisors Vanessa Braganholo and Daniel de Oliveira for all support and patience with me and my mistakes.

I would like to thank all the professors that I had contact during this period for all the contribution in my education. In particular, I would like to thank Prof. Paolo Missier for all the support during my period at the Newcastle University in UK. His contribution was very important for the progress of my work.

I would like to thank my classmates at UFF that helped me a lot during my PhD, especially when they made many jokes to forget the problems we were facing while working in the lab.

I would like to thank my wife Dalila Reis Albino, my daughter Sophia Albino de Oliveira, my parents Vicente Saulo de Oliveira and Maria Lúcia Moreira de Oliveira, my brother Washington Moreira de Oliveira, my sister Wanessa Moreira de Oliveira, and my friends that never stopped believing in me and gave me hope to achieve my goals.

Finally, I would like to thank UFF, CNPq and CAPES for the financial support of my studies in Brazil and UK.

RESUMO

A proveniência gerada por sistemas de *workflow* distintos é geralmente expressa usando diferentes formatos. Isto não é um problema quando cientistas analisam grafos de proveniência isolados, ou quando eles utilizam o mesmo sistema de *workflow*. Entretanto, quando há necessidade de analisar grafos de proveniência heterogêneos de múltiplos sistemas, as soluções existentes não fornecem o apoio necessário. Para resolver este problema, nós propomos uma arquitetura de integração de proveniência que adota o ProvONE como modelo de integração e mostramos como as bases de dados de proveniência distintas podem ser convertidas para um esquema global ProvONE. Desta forma, cientistas podem consultar esta base de dados integrada, explorando e interligando proveniência de vários sistemas e *workflows* diferentes que podem representar implementações distintas do mesmo experimento. Para ilustrar a viabilidade da nossa abordagem, nós desenvolvemos mapeamentos conceituais entre bases de dados de proveniência de quatro sistemas de *workflow* (e-Science Central, SciCumulus, Taverna e VisTrails). Nós fornecemos cartuchos que implementam tais mapeamentos e geramos uma base de dados integrada expressa em fatos Prolog. Nós também desenvolvemos regras Prolog que permitem que cientistas consultem a base de dados integrada. Resultados de uma avaliação experimental demonstram a efetividade e eficiência da nossa abordagem.

Palavras-chave: workflow científico, proveniência, proveniência prospectiva, proveniência retrospectiva, análise de proveniência, análise de proveniência integrada, interoperabilidade de dados de proveniência, PROV, ProvONE.

ABSTRACT

Provenance generated by different workflow systems is generally expressed using different formats. This is not an issue when scientists analyze provenance graphs in isolation, or when they use the same workflow system. However, when they need to analyze heterogeneous provenance graphs from multiple systems, the existing solutions do not provide the required support. To address this problem, we propose a provenance integration architecture that adopts ProvONE as an integration model and show how different provenance databases can be converted to a global ProvONE schema. Scientists can then query this integrated database, exploring and linking provenance across several different workflows that may represent different implementations of the same experiment. To illustrate the feasibility of our approach, we developed conceptual mappings between the provenance databases of four workflow systems (e-Science Central, SciCumulus, Taverna, and VisTrails). We provide cartridges that implement such mappings and generate an integrated provenance database expressed as Prolog facts. We have also developed Prolog rules that enable scientists to query the integrated database. Results of an experimental evaluation demonstrates the effectiveness and efficiency of our approach.

Keywords: scientific workflow, provenance, prospective provenance, retrospective provenance, provenance analysis, integrated provenance analysis, provenance data interoperability, PROV and ProvONE models.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API – Application Programming Interface

CM – Configuration Management

GUI – Graphical Use Interface

MFS – Multi-fasta file

MSA – Alignment file

OPM – Open Provenance Model

OS – Operating System

PGA – Provenance-gathering Activity

PH – Phylip file

WfMS – Workflow Management System

SUMMARY

Chapter 1 – **INTRODUCTION**

1.1 MOTIVATION

Over the last years, scientists have adopted Workflow Management Systems (WfMS) to execute their experiments based on computational simulations (COHEN-BOULAKIA *et al.*, 2017; CRUZ, SERGIO MANUEL SERRA DA *et al.*, 2009; CUI LIN *et al.*, 2009; TAYLOR *et al.*, 2014). Workflow Management Systems (WfMS) have been facilitating the design and implementation of data-driven computational science experiments, through a high-level programming model and a middleware-based runtime environment. Such experiments are usually very complex and require many executions with different algorithms and parameters so that several aspects of a hypothesis can be analyzed. Each of these executions is called a *trial* of the experiment. A trial of a scientific experiment is directly connected to a scientific workflow specification. There are several WfMS such as Kepler (ALTINTAS *et al.*, 2006), VisTrails (CALLAHAN *et al.*, 2006b), Taverna (HULL *et al.*, 2006), Swift (ZHAO, YONG *et al.*, 2007), Askalon (FAHRINGER *et al.*, 2005), Chiron (OGASAWARA *et al.*, 2013), Pegasus (DEELMAN *et al.*, 2005) and SciCumulus (DE OLIVEIRA *et al.*, 2010). Many of them are focused on high-performance computing (DE OLIVEIRA *et al.*, 2010; DEELMAN *et al.*, 2005; OGASAWARA *et al.*, 2013; ZHAO, YONG *et al.*, 2007), others on visualization (CALLAHAN *et al.*, 2006b; LIN *et al.*, 2008), while others focus on a specific domain (ABOUELHODA *et al.*, 2012; OINN *et al.*, 2004). However, all of them have a common characteristic: they offer mechanisms for capturing, storing and managing provenance data (FREIRE *et al.*, 2008).

Provenance is a fundamental part of the scientific experiment life cycle (Figure 1) that begins in the experiment specification (composition), proceeds to execution and then to the analysis phase (MATTOSO *et al.*, 2010). In this life cycle, the Analysis Phase holds the study of the results obtained in the execution phase through the collected provenance. Provenance can be defined as an audit trail of the experiment and captures information about the steps (*i.e.* activities) used to produce a data product (*e.g.* data files). Using provenance data, scientists are able to analyze the quality and authorship of data and reproduce the achieved results. It also helps scientists to discover new research opportunities, bringing to light new problems and challenges hidden in the traces of their experiments. It also aids scientists to detect and fix mistakes, acting similarly to a debug tool over the source code. In addition, provenance analytics is crucial for understanding a scientific experiment result, its dissemination,

reproduction, and evolution. As the life cycle suggests, provenance analytics can generate more data (graphs, visualizations, etc.), contributing to a data deluge and increasing the work of scientist over it (FREIRE; SILVA, 2008b). This way, data analysis and visualization become bottlenecks to scientific discovery (MATES *et al.*, 2011).

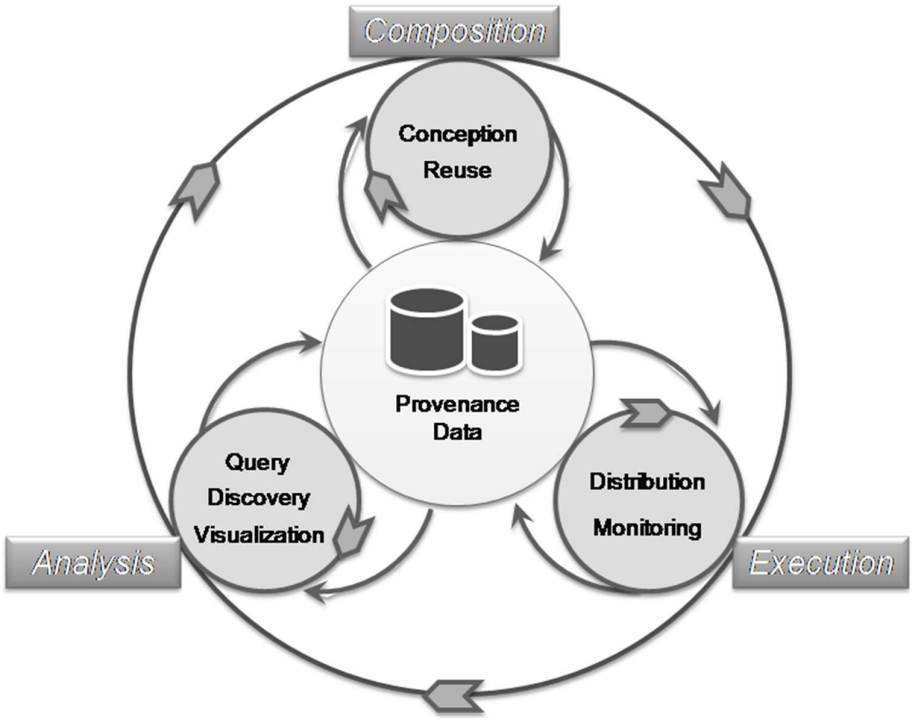


Figure 1. Scientific experiment life cycle as proposed by Mattoso *et al.* (2010)

Let us use, as example, a scenario where two or more collaborative research teams work independently on common (or similar) goals, adopting slightly different methods and procedures and thus producing workflows that differ in design, implementation, and execution middleware, but are otherwise similar in terms of intent, using comparable tools and algorithms. The two concrete examples we consistently use throughout this thesis is that of (i) four bioinformatics research groups, interested in generating phylogenetic trees and (ii) four clinician/geneticist research groups, interested in patient diagnosis. The first four groups independently designed and implemented SciPhy (OCAÑA, KARY *et al.*, 2011), SciEvol (OCAÑA, KARY A. C. S. *et al.*, 2012a), Phylo, and ML[1] workflows, using different WfMS, namely SciCumulus (DE OLIVEIRA *et al.*, 2010), VisTrails (CALLAHAN *et al.*, 2006b), Taverna (OINN *et al.*, 2004), and e-Science Central (WATSON *et al.*, 2010), respectively. On the other hand, the last four groups also designed and implemented four workflows named SVI, Pathogenesis, GeneClass, and PatientDiag. These groups also use distinct WfMS (e-Science

[1] http://eubrazilcloudconnect.eu/content/leishmaniasis-virtual-laboratory

Central, VisTrails, Taverna, and SciCumulus, respectively) to run the workflows. Each of these workflow systems has their specificities, but they are all capable of collecting retrospective provenance traces from their workflow runs, while a subset of them is capable of capturing both prospective (information about the workflow specification) and retrospective (information about the workflow execution) provenance. Since the phylogenetic analysis and patient diagnosis workflows use either the same or similar input data and produce similar outputs, it seems natural to try and use the provenance traces of their runs to compare and discuss produced results. However, each WfMS used to execute the workflows (VisTrails, Taverna, eScience Central and SciCumulus) has its own provenance schema and logical data model to represent prospective/retrospective provenance (relational, RDF, XML/relational, and graph-based, respectively) as well as to store it. Furthermore, the different nature of the WfMS leads to different levels of details in the provenance traces.

Thus, while in theory it should be possible for researchers to ask questions on any of these provenance graphs seamlessly and transparently, the heterogeneity in the design, implementation, and execution of their workflows translates into provenance traces that are themselves heterogeneous, making it difficult to analyze them jointly. Ultimately, this lessens the role of provenance in facilitating scientific discourse.

Aiming at verifying whether this kind of scenario could be real in practice, we have designed a survey[2] that was sent to scientists of different institutions around the world. This survey has the following question: "*Consider a collaborative science scenario where two teams execute variations of a given experiment and perform a joint analysis of these experiments, by comparing result data, methods, duration, and/or used parameters. In your experience, how likely is this scenario going to manifest itself in practice?*" We collected the responses and analyzed those from scientists which already have used a computational environment to design and execute an experiment (82 scientists fit this criteria). From those, just 6 % answered this kind of situation is not at all likely to happen.

Promoting provenance interoperability has been also the goal of several recent community efforts in provenance modeling, starting with the Open Provenance Model (OPM) (MOREAU; FREIRE; *et al.*, 2008) and culminating with PROV (MOREAU; MISSIER, 2013a), a W3C recommendation. Further, both ProvONE (MISSIER; DEY; *et al.*, 2013) and PROV-Wf (COSTA *et al.*, 2013a) independently extended PROV, adding explicit representation of *prospective* provenance (FREIRE *et al.*, 2008) to the model. In special,

[2] http://survey.npimentel.net/en/

ProvONE (which is used in this thesis) has been used by many research groups to represent provenance from workflow-based experiments. It is an abstract provenance model that can be implemented by many WfMS.

This thesis focuses in solving the following problem: How to enable integrated analysis on heterogeneous provenance databases generated by distinct WfMSs from similar (but not identical) experiments?

1.2 GOAL

In this thesis, we show how provenance interoperability that includes integration of the traces and their seamless querying, can be achieved in a practical setting where we assume a degree of similarity amongst the traces, as in the science and clinical scenario just outlined. Our main goal is to improve the efficiency and the effectiveness of provenance analysis by integrating heterogeneous provenance traces. Our approach aims at allowing scientists to go across multiple graphs and analyze provenance without worring about the format, structure, or language.

We argue that, to be useful, an integration model should include both retrospective and prospective provenance (which we henceforth concisely refer to as *r-prov* and *p-prov*). Hence, we develop a reference classification of provenance space that includes all possible matches between different provenance types and traces.

Specifically, we map provenance structures from different WfMSs to the ProvONE model and design an architecture that includes mechanisms to automatically translate particular provenance traces to ProvONE. We develop Prolog rules that enable scientists to query the integrated database, and new rules can be added as needed. We use Prolog as it provides great flexibility both in producing the integrated database, because provenance relationships translate to Prolog facts, and in formulating queries with inference capability, using Prolog rules. It is worth noticing that Prolog is considered a natural choice due to its syntactic similarity to PROV-N (MOREAU; MISSIER, 2013b). In addition, Prolog has been successfully used to query and analyze provenance data in approaches such as noWorkflow (MURTA *et al.*, 2014).

Some existing approaches (ELLQVIST *et al.*, 2009; SELTZER *et al.*, 2011; ZHAO, JING *et al.*, 2008) work on the integration and querying of provenance from different sources. Following work resulting from this thesis (OLIVEIRA *et al.*, 2016), Prabhune *et al.* (2018; 2017) also propose an integration approach to analyze heterogeneous provenance graphs by using RDF and SPARQL.

1.3 RESEARCH METHODOLOGY

Aiming to achieve our main goal, we define three stages in this work. The stages are: problem characterization, the design and implementation of our approach, and its evaluation. We present an overview of these stages as follows:

Problem characterization: in order to obtain the state of the art of provenance analysis, we conduct a literature review using the Snowballing technique (GOODMAN, 1961). In this technique, we first select a set of papers known to be relevant in the studied area. We call this set *S*. Then, we analyze all papers that are cited by the papers in *S*. Any paper that is considered relevant to the subject is included in *S*. We also analyze papers that cited the papers in *S* and include the relevant ones in *S* as well. Such procedure is repeated over all papers in *S* until no paper is added. Based on this set of papers, we analyze the several dimensions involved in provenance data analysis. From this, we create a taxonomy to classify and summarize the main existing approaches (OLIVEIRA *et al.*, 2018).

We also submitted a survey to scientists of different institutions around the world asking the follow question: "*Consider a collaborative science scenario where two teams execute variations of a given experiment and perform a joint analysis of these experiments, by comparing result data, methods, duration, and/or used parameters. In your experience, how likely is this scenario going to manifest itself in practice?*" We collect the responses and analyze those from scientists which already have used a computational environment to design and execute an experiment (82 scientists fit this criteria). From those, just 6 % answered this kind of situation is not at all likely to happen.

After performing the literature review and survey, we elaborate to the following research question: "*How to analyze heterogeneous provenance graphs, generated by similar but heterogeneous in silico experiments, taking into account the syntactic and semantic aspects of such graphs?*"

Design and implementation: based on the results obtained from the problem characterization stage, we design a reference provenance classification and an architecture that brings together heterogeneous provenance traces from different WfMS in a single knowledge base of Prolog facts.

The components of our architecture were developed and tested with different WfMSs. After running the implementation of our architecture, the provenance is consolidated in the knowledge base along with rules that can be used to query and analyze the workflow execution.

Evaluation: our approach was evaluated to verify its efficiency and effectiveness. Hence, we performed an experiment involving two groups (A and B) of volunteers (Computer Science students and professors) that have a basic knowledge level of Prolog. The experiment was conducted in a lab and the volunteers received instructions about the workflow and provenance terms and structures and about the knowledge base used. A digital form with 6 questions was delivered to them to be answered and they were instructed to register the time they spent to answer each of the questions. Group A had to answer the first 3 questions using the non-integrated approach and then answer 3 more questions using our integrated approach. Group B did the same in an inverse order. Besides that, we quantitatively evaluated our approach by comparing the storage space and time spent to translate provenance data.

1.4 CONTRIBUTIONS

In summary, this work has the following main contributions:

Problem characterization: We performed a survey asking scientists of different institutions around the world about how likely a scenario of integrated provenance analysis could manifest in practice. Just 6% of the respondents said this scenario is not at all likely to happen. This survey shows how important is the integrated provenance analysis in a real situation. We also performed a deep research about provenance analysis approaches and designed a new taxonomy that can guide current and future studies in this area.

Approach: We proposed an approach to analyze provenance from heterogeneous provenance graphs by using ProvONE as a canonical model. We also developed a reference classification of provenance space. Finally, the different provenance graphs were brought together in a knowledge base with Prolog rules that facilitate the query and analysis process.

Implementation: We developed cartridges to automatically translate provenance data from similar workflows executed in different and well-known WfMSs (Taverna, VisTrails, SciCumulus, and e-Science Central) to Prolog facts, following the ProvONE model.

Evaluation: The effectiveness and efficiency of our approach was evaluated by a statistical study over results obtained from an experiment with two groups (A and B) of volunteers (Computer Science students and professors). Each group received a list of questions about provenance generated from two different workflows and had to answer and register the time they spent to answer each of the questions. Group A performed the experiment with the non-integrated approach while group B performed the same experiment by using our integrated approach. We also quantitatively assessed our approach and performed comparative studies.

1.5 RESEARCH HISTORY

I started the PhD in the first semester of 2013 and finished the courses at the end of the same year. A summary research history timeline is presented in Figure 2. From 2013, we also started the literature review until 2018 when it culminated in a survey accepted at the ACM Computing Surveys Journal:

- *OLIVEIRA, Wellington; DE OLIVEIRA, Daniel; BRAGANHOLO, Vanessa. Provenance Analytics for Workflow-based Computational Experiments: a Survey. ACM Computing Surveys, p. 1–29, 2018 (to appear).*

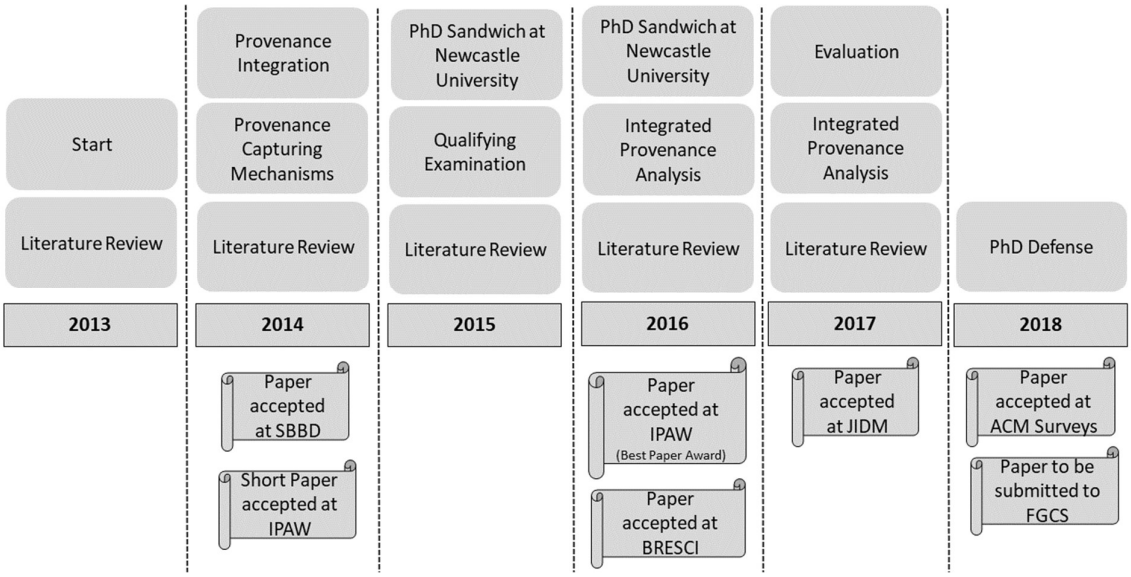


Figure 2. Research history timeline

In 2014, we started working with provenance integration and capture approaches. The results were published in two events:

- *OLIVEIRA, Wellington; NEVES, Victor C.; OCAÑA, Kary A. C. S.; MURTA, Leonardo; DE OLIVEIRA, Daniel; and BRAGANHOLO, Vanessa. Captura e Consulta a Dados de Proveniência Retrospectiva Implícita Intra-Atividade. In SBBD, 2014. p. 37-46.*
- *OLIVEIRA, Wellington; DE OLIVEIRA, Daniel; and BRAGANHOLO, Vanessa. Experiencing PROV-Wf for Provenance Interoperability in SWfMSs. In International Provenance and Annotation Workshop (IPAW), 2014. p. 294-296.*

In 2015, I prepared and presented my research proposal in the qualifying examination and started my PhD Sandwich Program at the Newcastle University in UK under the supervision of Prof. Paolo Missier.

During my PhD Sandwich, which finished in 2016, we developed an approach to analyze heterogeneous provenance graphs from different WfMSs. The result was written down in a paper that was accepted at IPAW (International Provenance and Annotation Workshop). This work was awarded the best paper of the workshop. We also performed a comparative study between ProvONE and PROV-Wf models. The results were published in the Brazilian e-Science Workshop in 2016.

- *OLIVEIRA, Wellington; MISSIER, Paolo; Ocaña, Kary A. C. S.; DE OLIVEIRA, Daniel; and BRAGANHOLO, Vanessa. Analyzing Provenance Across Heterogeneous Provenance Graphs. In International Provenance and Annotation Workshop (IPAW), 2016. p. 57-70.*
- *OLIVEIRA, Wellington; MISSIER, Paolo; DE OLIVEIRA, Daniel; and BRAGANHOLO, Vanessa. Comparing Provenance Data Models for Scientific Workflows: an Analysis of PROV-Wf and ProvOne. In Brazilian e-Science Workshop (BRESCI), 2016. p. 1-8.*

In 2017, we designed and implemented an architecture to import domain-specific data from external sources (not included in this thesis) and performed the evaluation of our approach:

- *OLIVEIRA, Wellington; OCAÑA, Kary A. C. S.; DE OLIVEIRA, Daniel; BRAGANHOLO, Vanessa. Querying Provenance along with External Domain Data Using Prolog. Journal of Information and Data Management (JIDM). v. 16, n. 1, p. 3–18, Apr. 2017.*

In 2018, I intend to submit a paper to the FGCS (Future Generation of Computer Science) journal:

OLIVEIRA, Wellington; MISSIER, Paolo; OCAÑA, Kary A. C. S.; DE OLIVEIRA, Daniel; and BRAGANHOLO, Vanessa. A Provenance Integration Architecture for Analyzing Heterogeneous Provenance Graphs. FGCS, to be submitted in 2018.

1.6 ORGANIZATION

The remainder of this thesis is organized as follows. Chapter 2 describes the provenance analysis approaches available in the literature. It also shows our provenance analysis taxonomy designed to guide new researches in this area. Chapter 3 presents our approach to analyze heterogeneous provenance graphs based on the ProvONE model. Chapter 4 describes the

experimental evaluation. First, we run an experiment with Computer Science students and professors to assess the effectiveness and efficient of our approach. Then, we performed a quantitative experiment and showed the comparative results. Finally, Chapter 5 concludes the thesis and points out future work.

Chapter 2 - PROVENANCE ANALYSIS FOR WORKFLOW-BASED COMPUTATIONAL EXPERIMENTS

2.1 CONTEXT

Although most authors agree that provenance data is useful for understanding the experiment and for reproducibility purposes, there is still a barrier to be overcome: how to analyze huge amounts of information generated by thousands (possibly millions) of large-scale workflow executions? In fact, the amount of published scientific papers on the subject evidences that provenance analytics has emerged as an important part of many research projects. Several technologies, platforms, applications, infrastructure, and standards are being proposed. However, to this date, the concepts involved with provenance analytics are not well organized, and the research results are incipient when compared to other aspects related to provenance such as capturing, modeling, and storing.

Considering the huge interest on this topic and the difficulty in finding organized definitions of its associated concepts, in this chapter, we propose a taxonomy for the provenance analytics field and use it to present a survey of the state of the art approaches. Our taxonomy provides an understanding of the domain and aims at helping the scientist to compare different approaches for provenance analytics. To achieve such purpose, we performed a bibliographical survey using a technique known as Snowballing (GOODMAN, 1961). In this technique, we first select a set of papers known to be relevant in the studied area. We call this set *S*. Then, we analyze all papers that are cited by the papers in *S*. Any paper that is considered relevant to the subject is included in *S*. We also analyze papers that cited the papers in *S* and include the relevant ones in *S* as well. Such procedure is repeated over all papers in *S* until no paper is added. Based on this set of papers, we analyzed the several dimensions involved in provenance data analysis. From this, we created a taxonomy to classify and summarize the main existing approaches (OLIVEIRA *et al.*, 2018).

The remaining of this chapter is organized as follows. Section 2.2 describes provenance data and its types. Section 2.3 introduces the taxonomy we propose to classify the main types of provenance analytics. Section 2.4 describes query languages and mechanisms to access provenance data. Section 2.5 brings different computing resources to explore provenance data such as data mining and collaboration. Finally, Section 2.6 summarizes the approaches and discusses open problems.

2.2 BACKGROUND ON PROVENANCE

Provenance describes the source and historical verification of the path traversed by a scientific workflow run to generate its result. It holds information about processes and data used to derive a result (DAVIDSON, SUSAN B.; FREIRE, 2008). Provenance gives credibility to the experiment, proves its results, makes its reproduction possible, and opens discovery opportunities in comparative studies. It improves the understanding and collaborates to a comprehension of the experiment as a whole. Furthermore, shared provenance repositories may guide the work of other scientists and speed up new scientific discoveries.

Bose and Frew (2005) survey a more general classification on provenance that can be applied in different areas such as e-Science, GIS (Geographic Information System), Databases, etc. The authors prefer to use the term lineage instead of provenance to describe the historical data derivation and transformation. Based on a literature review, they provide a metamodel for lineage retrieval with three main components: Workflow Model, Metadata, and Lineage Recovering Model. Similarly, Ragan *et al*. (2016) present an organizational framework of provenance types and purposes to different domains (scientific workflows, map creation, 3D modeling, financial analysis, etc.) focusing on visualization and data analysis. Considering provenance in e-Science, Simmhan *et al*. (2005) define provenance as a type of metadata that brings the data derivation history, starting from its original sources. Their work also exposes a taxonomy of provenance data features based on its use, granularity, representation, storage, and dissemination. Such classification aids the verification of issues related to provenance that were not resolved by existing work in the literature.

Especially for scientific workflows, provenance can be classified as prospective and retrospective. Prospective provenance represents the specification of computational tasks that will be executed. It corresponds to the steps to be followed to achieve a result. Retrospective provenance is given by executed activities and information about the environment used to produce a data product, consisting of a structured and detailed history of the execution of computational tasks (FREIRE *et al.*, 2008). It can also be specialized in explicit and implicit. Retrospective provenance is called explicit when it is previously declared in the workflow specification. On the other hand, retrospective implicit provenance (or just implicit provenance) represents hidden provenance data captured over accessed or changed objects inside implicit dataflows. It corresponds to the dataflows that were not explicitly declared in the workflow specification (MARINHO *et al.*, 2011; NEVES *et al.*, 2017). As an example, implicit

provenance allows us to debug an activity and to identify file changes that silently occurred due to the fact that the activity is being used as a black box.

Freire *et al.* (2008) classify the mechanisms for capturing provenance into three types: (i) workflow-based; (ii) activities-based; and (iii) operating system (OS)-based. Workflow-based approaches monitor the workflow as a whole. They are provided by the WfMS and thus are tightly coupled to them. Activities-based approaches are those in which the activities are responsible for collecting their own provenance. These mechanisms can be independent of the WfMS, but require adaptation/instrumentation of the workflow activities (MARINHO *et al.*, 2011; NEVES *et al.*, 2013). Finally, OS-based approaches can capture fine-grained retrospective provenance, but overall, they do not capture prospective provenance, or, when they do it, they are unable to relate prospective and retrospective provenance. Due to the huge amount of generated information, the capture and querying process can be complex (FREIRE *et al.*, 2008). In addition to these, we subdivide the activity-based approaches in (iv) inter and (v) intra-activities capture mechanisms. Inter-activity approaches capture information at the end of each activity. On the other hand, intra-activity approaches capture changes during the activity execution and can capture overlapping changes that occurred between the beginning and the end of the activity execution.

Cruz *et al.* (2009) use the capture mechanisms described by Freire et al. (2008) to classify provenance at capture levels in their taxonomy of provenance systems. Herschel and Hlawatsch (2016) give us a more general provenance classification. They define provenance in different types such as Data Provenance, Workflow Provenance, Information System Provenance, and Provenance (metadata). In their classification, Workflow Provenance can have Process Type, Computational Model, and Granularity that is divided in coarse-grained (control-flow-based) or fine-grained (individual data-based). Our taxonomy can be seen as an evolution of the taxonomies proposed by Bose and Frew (2005), Simmhan *et al*. (2005), and Cruz *et al.* (2009) and also the classification defined by Herschel and Hlawatsch (2016) and Herschel *et al.* (2017), focusing on provenance analytics. Section 2.3 describes the elements that compose our taxonomy for provenance analytics.

2.3 A TAXONOMY FOR PROVENANCE ANALYTICS

Provenance analytics can be performed over different provenance types and granularities. It can also use many access methods and heterogeneous formats for representing data. As mentioned in Section 2.2, both Bose and Frew (2005) and Simmhan et al. (2005) create metamodels or taxonomies to classify and describe types, granularities, and representations of

provenance. Besides them, Cruz et al. (2009) also define a general taxonomy to classify provenance characteristics in WfMS, and Herschel and Hlawatsch (2016) provide a general classification of provenance types and adapted visualizations. The taxonomy we propose in this chapter (Figure 3) differs from the previous ones in the sense that it focuses on the major features of provenance analytics. It provides the classification of the provenance analytics approaches into categories based on different aspects of this field and on the requirements of a scientific experiment.

This section describes six sub-taxonomies that compose the more general taxonomy we propose. For the sake of simplicity, our taxonomy, presented in Figure 3, classifies the characteristics of provenance analytics approaches regarding type, granularity, model, format, access methods, and computing resources. In the following, we describe each term associated with the provenance analytics taxonomy.

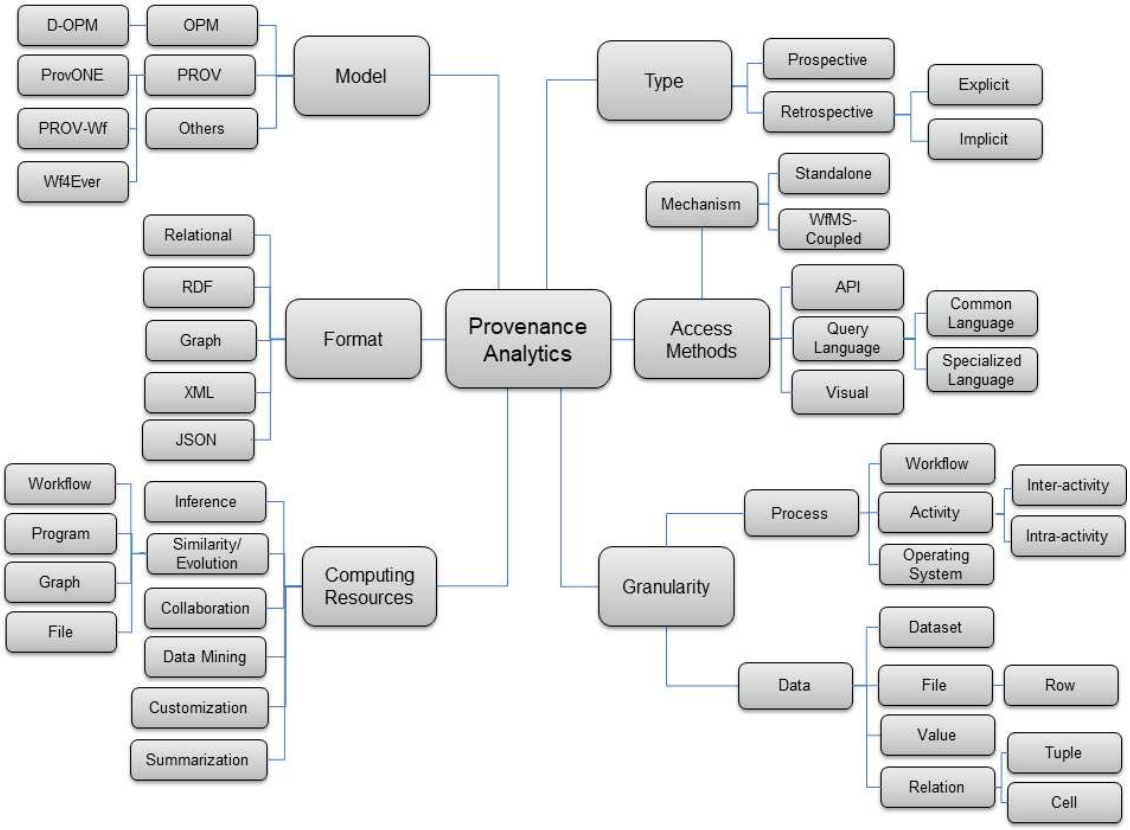


Figure 3: Provenance analytics taxonomy

1. **Type**. As discussed in Section 2.2, provenance can be classified as prospective and retrospective. Usually, the majority of the analytic tools just consider these types of provenance. However, to analyze the complete provenance trail, scientists need to connect

all types of generated provenance data (prospective, explicit retrospective, and implicit retrospective provenance).

2. **Granularity**. There are many ways to capture and analyze provenance data. We follow, in part, the capture mechanisms presented by Freire et al. (2008) (workflow-based, activities-based, and OS-based) and described in Section 2.2 to classify the granularities of provenance analytics. Furthermore, we distinguish two types of granularities: process granularity and data granularity. Process granularity corresponds to the kind of process the provenance is linked to, for instance, workflow, activity (inter and intra-activity) or OS. It is the owner of the provenance. On the other hand, data granularity distinguishes the different levels of details (fine or coarse grain). It has four levels, as described below.
    2.1. **Dataset**. Provenance data can be organized in a set of similar or related data. In this classification, datasets are coarse-grained data used or generated by processes.
    2.2. **File**. At this granularity level, provenance is captured and analyzed from information about accesses and changes occurred over physical raw files generated or consumed by workflow activities (i.e. filename, directory, type of modification, modification time, etc.).
        2.2.1. **Row**. Analysis performed over provenance data at row level considers data about accesses and changes occurred over each file content line generated or consumed by workflow activities. It is the finest grain that can be captured by a provenance system over this type of artifact.
    2.3. **Value**. This provenance granularity level concerns input and output data exchanged by activities or programs/processes. Normally, it corresponds to return of function calls, function or procedure parameters, and message exchange among services.
    2.4. **Relation**. A relation (*e.g.* a relational table) is a structured type that consists of one or more columns. Instances of a relation have a set of tuples. Each tuple has a single value for each column. A relation may be associated with other relations.
3. **Model**. The facilities provided by the use of WfMS make the development and management of in silico experiments much easier. However, there are issues related to the heterogeneity of the provenance data they generate. Each workflow system uses a different syntactic structure (model) to represent and store provenance, which creates a barrier to provenance integration. This way, several recent community efforts have culminated with the development of generic models to represent provenance and to promote provenance interoperability. We include the most expressive of these models in our taxonomy.

3.1. **OPM**. The Open Provenance Model (OPM) was the first model designed to represent retrospective provenance data. It enables exchanging provenance information and the development and share of tools that operate on it (MOREAU *et al.*, 2011). OPM has three types of elements (Artifact, Process, and Agent) that can be related through predetermined relationships (used, wasGeneratedBy, wasControlledBy, wasTriggeredBy, and wasDerivedFrom).

3.1.1. **D-OPM**. DataONE-OPM (D-OPM) is a provenance model that extends OPM. It was designed in the context of the DataONE Project, and it can represent the workflow structure, traces from workflow executions, data structure, and workflow evolution (CUEVAS-VICENTTIN *et al.*, 2012).

3.2. **PROV**. PROV provides a generic data model (*i.e.*, PROV-DM) to outline provenance. It is a W3C recommendation and was designed to be an agnostic model to represent provenance from different areas. Its data model is capable of representing data transformations, ownership, etc. PROV may also be extended to fulfill requirements of particular domains. The elements Entity, Activity, and Agent along with the relationships Used, WasGeneratedBy, WasInformedBy, WasAssociatedWith, WasDerivedFrom, WasAttributedTo, and ActedOnBehalfOf form its core.

3.2.1. **ProvONE**. ProvONE extends the PROV model with an explicit representation of prospective provenance, thus capturing the most relevant information on scientific workflow processes. It is designed to accommodate extensions for specific scientific workflow systems (MISSIER; DEY; *et al.*, 2013). It includes both prospective and retrospective provenance and allows for easy integration of terms from external vocabularies, including Dublin Core or WfMS.

3.2.2. **PROV-Wf**. PROV-Wf is a conceptual model for the representation of prospective and retrospective provenance collected from the execution of scientific workflows. It is also a specialization of the PROV model and provides specific elements to the scientific experiments context. According to COSTA *et al.* (2013b), their elements can be classified into three main types: (i) Structure of the Experiment; (ii) Execution of the Experiment; and (iii) Environment Configuration.

3.2.3. **Wf4Ever**. The Wf4Ever team also extends PROV and uses the term wfdesc and wfprov to describe prospective and retrospective provenance respectively in the context of research objects (CORCHO *et al.*, 2012). Their model can represent both executable and abstract workflows.

4. **Format**. Usually, provenance captured by one WfMS is stored using a specific format that can vary according to the domain and user needs (FREIRE *et al.*, 2008). Even when the abstraction model differs from the storage model, they share an essential information type: processes and data dependency. Normally, provenance data is represented as a directed acyclic graph (DAG). On the other hand, many provenance systems have used different formats to represent and store provenance data such as RDF triples, relational tables, graphs, XML, and JSON.
5. **Access Methods**. Due to the large volume of provenance data an experiment can generate, it is necessary to use appropriate and friendly ways to access, filter, and group it. Moreover, access methods are very dependent on the models and formats in which provenance is represented and stored. We classify the mechanisms or tools to access provenance data as WfMS-coupled (the mechanism works inside the system) and standalone (the mechanism is decoupled from the WfMS that generates the provenance data). Furthermore, each mechanism may use different ways to access provenance data, as described next.
    5.1. **Query Language**. Data analyses approaches can use common languages such as SQL (ELMASRI; NAVATHE, 2010), XQuery (BOAG *et al.*, 2010), XPath (BERGLUND *et al.*, 2010), and SPARQL (PRUD'HOMMEAUX; SEABORNE, 2008), depending on the data storage model (Relational, XML, RDF) or specialized languages such as VDL (Virtual Data Language) (FOSTER *et al.*, 2002), OPQL (OPM Query Language) (LIM *et al.*, 2011), QLP (Query Language for Provenance) (ANAND, M.K. *et al.*, 2010), among others, which were specially developed to query provenance data.
    5.2. **Visual**. Some approaches use visualization tools to present provenance data through images (hierarchical trees, graphs, timelines, etc.). These tools ease the construction of human inferences and the identification of patterns in the analyzed data. Visualization can also be used to add meaning and summarize
    5.3. **API**. APIs provide specialized algorithms, search mechanisms, and interfaces to query provenance data on various data sources.
6. **Computing Resources**. Scientists may explore and extract useful information over provenance data through different computing resources. Each one is built upon a specific provenance model. Our taxonomy describes some of the main available resources as follows.
    6.1. **Inference**. In some cases, provenance data is better understood when we have information on both its structure (syntactic) and semantics. Inference in this context

plays an important role because it can derive new relations on both the syntactic and semantics aspects.

6.2. **Similarity/Evolution**. Each piece of data can be versioned so that its evolution can be tracked. In the provenance analytics context, this becomes interesting when applied to Workflows, Programs, Graphs, and Files. It can help us to follow their changing history, reuse it, or compare versions to find out their differences and similarities.

6.3. **Collaboration**. Crowdsourcing (HOWE, 2006) has helped many initiatives in the computing area. There are already some Web systems that aid users to share workflows, provenance data, their understandings, and questions about scientific experiments. These systems inherit features from blog systems and social networks such as Facebook, MySpace, Twitter, and Flickr. Usually, these environments offer wide space to discuss and disseminate ideas.

6.4. **Data Mining**. This resource allows the discovery of new patterns that could not be observed in a direct way in the experiment. These patterns may help scientists to extract hidden information over huge amounts of data generated by numerous workflow executions. Applying data mining over provenance repositories can generate data clusters, creating compositions that aggregate data and facilitate provenance analytics.

6.5. **Customization**. Analyzing provenance is far from trivial due to the large amount of data produced by the execution of various experiment trials on WfMS. In this sense, customization of provenance graphs and views can be a very useful mechanism to filter information and show just pieces of information that make sense for a given scientist perspective.

6.6. **Summarization**. Summarization plays an important role on provenance analytics by shrinking data, spotting the most useful information, and throwing distractions out.

The next sections discuss the state of the art approaches for provenance analytics available in the literature using our proposed taxonomy as a guide. We focus on Access Methods (Section 2.4) and Computing Resources (Section 2.5), since the other aspects of our taxonomy are not directly related to analysis. Instead, they classify provenance types, models, formats, and granularity, which are crucial for the analysis tools that run over provenance data.

2.4 ACCESS METHODS FOR PROVENANCE ANALYTICS

Approaches that provide access methods for provenance analytics use query languages, APIs, and visual mechanisms to facilitate the scientists' analysis. Sometimes is not easy to identify the best analytical method. However, people tend use queries and/or API when they

know what they are looking for (HERSCHEL; HLAWATSCH, 2016), and use visualizations otherwise.

Figure 4 shows a timeline of the provenance data access approaches based on our provenance analytics taxonomy. The year 2008 concentrates more papers about provenance querying. Visual approaches appear consistently almost on the entire timeline, while specialized languages are more present among 2008 and 2012 (just one language appears in 2002), and few API are provided.

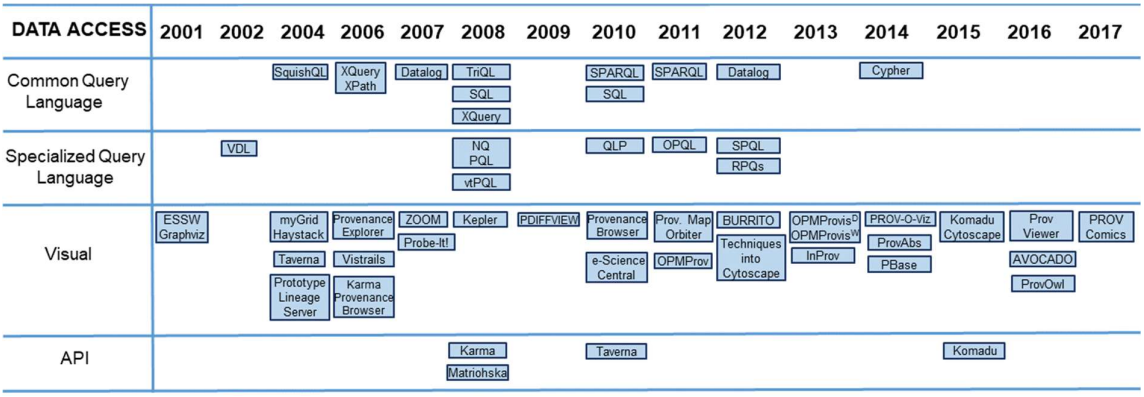


Figure 4: Provenance data access approaches timeline

This section is organized as follows. Section 2.4.1 describes initiatives that propose or use query languages to query provenance. Section 2.4.2 lists some approaches to visualize provenance data. Finally, Section 2.4.3 presents API to query provenance data. Approaches are discussed in chronological order as much as possible and grouped by similar features.

2.4.1 QUERY LANGUAGES

Several systems in the literature use query languages to provide scientists with analysis capabilities. Foster *et al.* (2002) propose a specialized language named VDL (Virtual Data Language). VDL is simpler than other languages such as SQL, SPARQL or Datalog (CERI *et al.*, 1989). It constitutes a bridge to provenance query on different relational data sources. It is implemented in SQL and supports data definition and query statements. It works as a lingua franca to the Chimera virtual data grid (FOSTER *et al.*, 2002). VDL is independent of the catalog schema and its query results are tasks represented as DAG, which abstracts the data storage system and is a well-suited format for analysis. VDL is also able to recursively query the provenance graph. Furthermore, Foster *et al.* (2002) define an interpreter to convert VDL to SQL. In a later effort, Zhao *et al.* (2006) propose some extensions to the Virtual Data Model

aiming at providing an integration between prospective, retrospective provenance, and semantic annotations.

The Prototype Lineage Server (BOSE, R.; FREW, 2004) allows for browsing lineage metadata in a workflow invocation through HTML links that shows provenance information in XML. It also has an RDF vocabulary that enables to create properties, relationships, and query RDF/XML provenance using SquishQL[3].

Chebotko *et al.* (2010) present a system named RDFPROV. It stores RDF provenance triples (ontology-based) in a relational database. Queries are written in SPARQL and converted to SQL through a mapping layer. RDFPROV also translates queries regardless of the schema. RDFPROV is used with the VIEW system (LIM *et al.*, 2011, 2013) and shown to be better for providing provenance metadata than Sesame (BROEKSTRA *et al.*, 2002) and Jena (CARROLL *et al.*, 2004). RDFPROV comes with a provenance ontology that bears many similarities to OPM (MOREAU; FREIRE; *et al.*, 2008), although they were developed in parallel and independently. In the same way, Gaspar *et al.* (2011) propose an architecture called SciProv that represents provenance according to the OPM model. It uses Web semantic tools (RDF, ontology, and OWL) that allow the lineage inference. The architecture of its provenance system enables users to perform queries on SPARQL. Data collected from workflows are obtained by means of instrumentation (using Web services) and stored in a relational database.

Developed to capture fine-grained provenance, PASS (Provenance-aware storage systems) (HOLLAND, DAVID A. *et al.*, 2008) collects implicit retrospective provenance at the OS level. Holland *et al*. (2008) developed a SQL-like provenance query language called nq (new query) to PASS. This language represents each provenance entity as a row and each attribute as a column. In another work, Holland *et al*. (2008) define a new semi-structured language to PASS called PQL (Provenance Query Language). PQL extends the semi-structured language Lorel (ABITEBOUL *et al.*, 1997). The difference between Lorel and PQL is that in the later the edges between nodes were extended to be bidirectional (HOLLAND, D. *et al.*, 2008). This enables ascendant and descendant navigation between nodes. PQL keeps what is fundamental to provenance and other traditional languages ignore: the idea of path (HOLLAND, D. *et al.*, 2008). Similarly, ES3 (FREW, JAMES *et al.*, 2008) has an OS-level capture approach that captures implicit retrospective provenance from passive trace monitoring over Linux operating system processes. It represents and stores all generated provenance in an XML database. To analyze the collected provenance, it uses XML provenance requests with

[3] http://ilrt.org/discovery/2001/02/squish/

XQuery constraint expressions (FREW, JAMES *et al.*, 2008). Like PASS, ES3 does not consider workflow representations and both approaches retrieve large amounts of information that makes their analysis more difficult. Furthermore, they depend on a specific operating system to be executed and require a post-processing phase to interrelate the implicit provenance with the prospective provenance (inferred by them).

The QLP (Query Language for Provenance) is a query language based on graphs. Its syntax is similar to other languages that work with XML and XPath, as well as to languages that use generalized path expressions such as Lorel (ANAND, M.K. *et al.*, 2010). QLP allows for querying lineage relations between nodes and invocations, in-out edges, and other input and output invocation structures. They also have structural relations between nodes. QLP language brings a series of desirable features to a provenance query language such as physical data independence, workflow system independence, preservation of provenance relationships, incremental query and transparent optimization (ANAND, MANISH KUMAR *et al.*, 2009; ANAND, M.K. *et al.*, 2010). It returns a set of lineage edges instead of a set of nodes that would require additional steps to reconstruct their relationships. The results are shown in a graph form by a system called Provenance Browser that also internally converts QLP to SQL (ANAND, M.K. *et al.*, 2010). Provenance Browser runs standalone or integrated into the Kepler system.

Lim *et al*. (2011, 2013) describe a query language based on OPM called OPQL (OPM Query Language). OPQL generates a new provenance graph as the output of a query over a given provenance graph. The language is weakly coupled to the storage schema and was implemented in the OPMProv system. OPMProv uses a relational database and is capable of storing, processing and querying provenance data according to the OPM model (LIM *et al.*, 2011). The OPM model works as a conceptual model to OPM-Prov that maps OPQL language to SQL. Provenance data representations in XML (that are structured according to an XML Schema, following the PROV model (MOREAU; MISSIER, 2013b)) can be inserted in OPMProv using a mapping procedure. Such procedure fragments the XML document on tuples and stores them into a relational database. Lim *et al.* (2013) extend their work and describe a Web service to OPM-Prov, allowing users to perform queries using OPQL in a provenance browser called OPMProvisD (desktop version) and OPMProvisW (Web version). Both browsers enable users to visualize and navigate on provenance graphs of the “original” provenance data and provenance data generated after the execution of queries on OPQL. They also allow for Zoom-in and Zoom-out functions as well as grouping or ungrouping displayed objects. OPMProv can also work as a provenance manager to the View system (LIM *et al.*,

2013). Provenance queries can also be performed directly on the View system using SPARQL and SQL (LIN *et al.*, 2008).

Similarly, Gadelha *et al.* (2012) developed a structured language called SPQL (Structured Provenance Query Language). SPQL's syntax is similar to SQL. It uses embedded functions, stored procedures, and materialized views. It also allows recursive queries based on the SQL:1999 standard. Queries on SPQL are internally translated to SQL by the MTCProv system. This system is integrated to the parallel script system Swift. MTCProv is a provenance query framework for Swift and a successor to the VDS (Virtual Data System) (FOSTER *et al.*, 2002).

VisTrails offers a specialized query language called vtPQL (SCHEIDEGGER *et al.*, 2008). vtPQL uses tags to search provenance that is stored at different layers: vistrail level (vt), workflow level (wf) and execution level (log). vtPQL is similar to SQL with attributes, predicates, and additional functions such as *upstream(x)* that returns all activities that preceded *x*, and *executed(x)* that returns true in case activity *x* has been executed (SCHEIDEGGER *et al.*, 2008). The vtPQL language is WfMS dependent (VisTrails) and strongly coupled to the storage schema (relational).

Extending OPM to represent workflow structure and evaluation, Cuevas-Vincenttin *et al.* (2012) present a new provenance model called D-OPM. They provide a reference implementation aiming at interoperation with multiple systems and a query mechanism based on Regular Path Queries (RPQs). RPQs return pairs of nodes in a graph according to a regular expression. The query mechanism was developed using relational database facilities to achieve interoperability and extensibility. In a later work, Dey *et al.* (2013) demonstrate the implementation of RPQ variants by using Datalog and RDBMS. Dey *et al.* (2012) also present a provenance model based on graphs that extends the OPM model. Unlike the OPM model, instead of considering only retrospective provenance, it defines a unified provenance model that also considers prospective provenance and temporal dimension of entities relationships. Provenance queries over this model are performed through Datalog (CERI *et al.*, 1989).

Both Dey *et al.* (2012) and Cohen *et al*. (2006) use Datalog to define rules, including temporal dimension and prospective provenance data to the OPM graphs (DEY *et al.*, 2012) or creating user views (COHEN *et al.*, 2006) over provenance data. The authors use clear semantics and the power of the Datalog inference to elaborate recursive and non-recursive queries on causal dependency graphs. Despite adopting a more direct and compact form of rules definitions and queries, Datalog imposes a certain level of difficulty to non-experienced users.

Built on the Neo4j graph database, PBase (CUEVAS-VICENTTÍN *et al.*, 2014) provides a unified provenance repository that follows the ProvONE model. PBase also offers an interface to pose provenance queries on a NoSQL database (Neo4J) by using the Cypher language. For now, PBase supports uploads of the VisTrails XML format. In another work, Vicenttin *et al*. (2014) change the PBase format to RDF (stored in the TDB, the RDF triple store of the Jena Framework) and adopt SPARQL as PBASE's query language.

2.4.2 VISUAL

The visual data analysis can enable for detection, comparison, and validation of expected results, hiding details, and showing the semantic necessary to its understanding. It also improves the data interpretation, facilitates decision making, and leads scientists to unexpected science discovery (HANSEN *et al.*, 2011). As opposed to data mining that automatically discovers patterns (see Section 2.5.2), visualization requires that users infer "the pattern". However, the major barrier to the effective use of visualization is the lack of appropriate data management techniques that are needed to make data exploration scalable (CALLAHAN *et al.*, 2006a). Thus, visualization should be considered as part of the data exploration process and not only as visualization of results (SILVA; FREIRE, 2008). In fact, as in the Big Data analytics scenario, visualization can represent an ample range of information that could be very hard to understand by using just numbers or text. In this context, the old maximum "a picture is worth a thousand words" expresses how meaningful a visual representation can be. In this section, we describe several approaches that apply tools and techniques that aim to aid scientists in performing visual provenance analysis. Many approaches we describe here work on visualization of provenance data represented as a graph with nodes that represent data or activities invocation and edges that represent relationships between these nodes. Some of them allow for graph summarization, grouping, ungrouping, and workflow versioning to verify more interesting provenance features and improve the knowledge about the workflow.

We start by discussing tools designed to create visualization (VIEGAS *et al.*, 2007). Then, we discuss approaches that provide provenance views based on Semantic Web concepts (CHEUNG; HUNTER, 2006; DEL RIO; DA SILVA, 2007; HOEKSTRA; GROTH, 2014; ZHAO, JUN *et al.*, 2004). After that, we present tools that allow users for customizing and/or personalizing provenance views (ANAND, MANISH KUMAR *et al.*, 2009, 2012; ANAND, M.K. *et al.*, 2010; BITON, O. *et al.*, 2008; BITON, OLIVIER *et al.*, 2007; CHEN *et al.*, 2012; COHEN-BOULAKIA *et al.*, 2008; KARSAI *et al.*, 2016; KOHWALTER *et al.*, 2016; MISSIER; WOODMAN; *et al.*, 2013; SELTZER; MACKO, 2011; STITZ *et al.*, 2016).

Following, we list visualization tools that allow for provenance querying by using previous workflows and controlling the workflow evolution (HLAWATSCH *et al.*, 2015; SILVA *et al.*, 2007). Then, we present approaches that work with filesystem provenance visualization and diff (BORKIN *et al.*, 2013; GUO; SELTZER, 2012). Finally, we discuss approaches that take into account metadata integration (FREW, J.; BOSE, 2001; SIMMHAN, Y.L. *et al.*, 2006).

While running part of a workflow can generate a new dataset in minutes, systems such as SciRun spend hours or even days to generate the visualization of these data (FREIRE *et al.*, 2006). Similar to SciRun, the systems Paraview and MayaVi allow for creating and manipulating complex visualizations of dataflows. However, they lack support to large scale data exploration (CALLAHAN *et al.*, 2006a; SILVA *et al.*, 2007). On the other hand, initiatives such as Many Eyes (VIEGAS *et al.*, 2007) have encouraged users with little ability on the creation and manipulation of visualization tools. Many Eyes offers an open website that enables users to load data and use different components to create visualizations. Similarly, various proposals try to provide a user-friendly tool aiming at extracting important information over provenance data.

Working on the myGrid project, Zhao *et al*. (2004) use semantic web technologies such as RDF, ontology and semantic inferencing mechanisms (Algernon) for representing and adding semantics to heterogeneous provenance data. They demonstrate how provenance graphs in RDF can be visualized on the Haystack, a Semantic Web browser (QUAN; KARGER, 2004). Based on the ABC ontology model, Cheung and Hunter (2006) propose a standalone system named Provenance Explorer that takes provenance in the RDF format and yields customized visualizations of provenance graphs by using Jena (CARROLL *et al.*, 2004). The provenance graphs are personalized according to the user's requirements or access permissions. Its GUI shows provenance in a coarse-grained view and allows users for expanding it in a fine-grained view. Provenance Explorer also has a platform for publishing scientific results.

Probe-It! (DEL RIO; DA SILVA, 2007) is another visualization tool for rendering provenance information from inference engines and workflows. Provenance is encoded as Proof Markup Language (PML) documents, that references sources stored in an IW-Base repository (MCGUINNESS *et al.*, 2004). It has multiple viewers, each suited to different provenance elements. Provenance information in Probe-It! is composed of results (intermediate and final), justifications (a graph of the workflow trace), and provenance (information about the services and sources). Del Rio *et al*. (2010) improved the usability and performance of Probe-It! by adding support to a Google Earth-like navigation, zoom in/out, information abstraction and a preprocessing system that caches visualization.

Working on PROV-O RDF serialization of PROV, Hoekstra and Groth (2014) propose a Web-based visualization tool named PROV-O-Viz to analyze provenance from different sources. They use Sankey Diagrams to determine important activities and understand how data flows through and between activities. It highlights entities and activities that have more dataflows by changing their size in the diagram. It is also able to infer missing information.

In the same direction, Anand *et al*. (ANAND, MANISH KUMAR *et al.*, 2009) define a navigation model with three granularity levels for visualizing dataflow graphs: (i) actor level, (ii) invocation level, and (iii) data dependency level. They also define navigation model operators: expand, collapse, group, ungroup, filter, navigate, standard views and flow-graph views. Furthermore, they present an architecture that computes the difference between the current view (created by the user) and the original provenance stored in the database (ANAND, MANISH KUMAR *et al.*, 2009). The implementation of this approach is described by Anand *et al*. (2012) by using a relational model.

Provenance Browser is a system that can run either integrated to the Kepler system or standalone (ANAND, M.K. *et al.*, 2010). It offers an interface to visualize, navigate, and query provenance. Provenance Browser's architecture allows data to be introduced directly in the browser or through a relational provenance database. It also allows for navigating on different views: dependency history, collection history (composed nodes), and invocation dependency (ANAND, M.K. *et al.*, 2010). In a view, the user can go forward or backward in the execution history. Provenance Browser has a generic provenance model that has a straightforward conversion to OPM. A part of its exhibition window also shows the collection structure of XML nodes together with details of activity invocations (also called actor). The functions available to query (by using QLP), group, ungroup, filter, and navigate help to improve the user understanding over the displayed provenance graph.

Davidson *et al*. (2007) propose an approach to omit provenance data that is not of interest for a particular user. In this approach, the union of various relevant modules (or activities) to a particular user module composes views. The user indicates which workflow modules are relevant to her and then, from an abstraction mechanism, provenance information is shown according to the view defined by each user. In this way, the user can only visualize data passed between modules of his user view (workflow modules partition) and cannot visualize data that are internal to the composed modules. Biton *et al*. (2007) implement this user view idea in the ZOOM*UserView system. This system aims to construct user views and provides an interface to query provenance data. In the ZOOM system, the causal graph representation is constructed from input (read) and output (write) events of the provenance base

tables and of their relationships with other tables in the schema (COHEN-BOULAKIA *et al.*, 2008). The user interface graphically shows the workflow with its atomic modules, allowing users to select modules that are relevant to them. From this selection, the system creates composed modules. The interface also presents the created composed modules (on a graph) to allow users to perform queries. Thus, by clicking on an edge, the user can see the data set that passed between two activities (also called steps or modules) (BITON, O. *et al.*, 2008). To create a user view model, Cohen *et al.* (2008) use Datalog. Datalog is also used to execute queries over views created on provenance data. Despite the benefits reached with the creation of the user views by ZOOM (BITON, O. *et al.*, 2008; BITON, OLIVIER *et al.*, 2007), its implementation (aggregation of activities in compositions) does not allow the whole original dependency path of provenance (before the aggregation) to be discovered. Davidson *et al*. (2007, 2009) expose these issues, calling them unsound views. These views received this name because they do not preserve information about the original dependency flow of provenance data. Sun *et al.* (2009A) and Hu *et al*. (2012) also work on this problem.

Exploring the provenance graph in an interactive manner without requiring users to specify rules in advance is a desirable feature in provenance management systems. Provenance Map Orbiter is a system that works this way (SELTZER; MACKO, 2011). It automatically performs the data summarization, without the necessity of user intervention. Provenance Map Orbiter uses graph summarization (RDF or OPM) to present to the user a high-level view of the provenance graph and semantic zoom, showing only relevant nodes. In the first step of the algorithm, all activities are considered primary nodes and classified as summary nodes (the user can also designate nodes that are more relevant as primary). Next, nodes that represent the generated steps by one summary node are grouped into this generator node. Finally, all immediate neighbor nodes (descendants and ascendants) of all nodes belonging to the summary node are inserted in it. Visualization is given through a timeline that shows the workflow activities tree. Provenance Map Orbiter also allows for using filters that are graphically exhibited. Borkin *et al*. (2013) compare Provenance Map Orbiter to their new filesystem provenance visualization tool named InProv. InProv uses a different way to represent provenance views. Filesystem provenance is shown with an interactive radial-based tree layout rather than node-link diagram. It summarizes activities and shows the inter-relationships within the data. They also developed a time-based hierarchical node grouping that can be used to match the user's mental model. Results from a qualitative evaluation indicate that the time-based node grouping improves the performance and usability of both InProv and Provenance Map Orbiter systems. For now, InProv does not have a similarity functionality to compare files. Similarly,

BURRITO (GUO; SELTZER, 2012) captures and analyzes provenance on OS-level and has a Computational Context Viewer that offers an HTML page to graphically display the diff between different input files versions and command-line parameters and relates them to the effects over output file versions.

Also working on graph summarization, Stitz *et al*. (2016) present the AVOCADO (Adaptive Visualization of Comprehensive Analytical Data Origins) approach aiming for reducing the complexity of the graph using hierarchical and motif-based aggregation. AVOCADO also allows users to expand regions in the provenance graph that are interesting to the user based on a degree-of-interest (DoI) function.

Based on graph abstraction, Missier *et al*. (2014) define a Provenance Abstraction Model (PAM) and a simple policy model implemented into the ProvAbs tool. Their tool loads provenance in PROV-N format and stores provenance in a Neo4J database. From this database, queries can be posed by using Neo4J Traverse API and Cypher language. In this work, they only include the generation and usage relations on Activity and Entity nodes.

Chen *et al.* (2012) present a set of techniques for exploring and explaining provenance such as a layout algorithm, visual style, graph abstraction techniques, and graph matching algorithm. They implement these mechanisms into Cytoscape (SHANNON *et al.*, 2003). Cytoscape reads XML files yielded by the Karma provenance server (SIMMHAN, Y.L. *et al.*, 2006) and generates provenance graphs. Using Cytoscape, Karsai *et al.* (2016) design a prototype called ProvOwl to simplify provenance graphs visualization. They developed this prototype based on a clustering approach from a previous work (ProvAbs) (MISSIER *et al.*, 2014). The provenance graph uses the PROV model representation. ProvOwl allows users to combine several nodes, zoom-in/out, filter, or rearrange nodes. Similarly, Kohwalter *et al.* (2016) present a PROV-N compliant tool named Prov Viewer. It shows provenance as a graph and integrates many features and mechanisms such as filtering, collapsing, zooming, coloring, graph merging, and domain configuration that add semantics. It allows users to analyze provenance in different granularities.

VisTrails uses the VTK (Visualization Toolkit) library to create visualizations of provenance data (CALLAHAN *et al.*, 2006b). It offers scalable parameter exploration and makes use of 3D visualization techniques, volume renderization, and isosurfacing (SILVA *et al.*, 2007). It also allows visual queries through QBE (query-by-example) that enable querying provenance using previous workflows. A QBE query is constructed in the same interface used to design the workflow. Conditions and parameter values can also be set up to filter the query result. The tool also provides the version history of each generated workflow. It is linked to a

particular workflow configuration (pipeline), and to the executions (log) with their respective provenance data that can be queried in a specific field. Hlawatsch *et al.* (2015) add new features to VisTrails to visualize the evolution of workflow modules. They include visualizations of module lifetimes and events with grouping and filtering mechanisms, a visual representation of version branches, and it is able to combine multiple visualizations.

Also working on provenance visualization, Cuevas-Vincenttin *et al*. (2014) present PBase, a visualization tool to analyze provenance. It has a Web-based graphical user interface that can simultaneously represent prospective and retrospective provenance as a graph with identified nodes.

Frew and Bose (2001) present the Earth System Science Workbench (ESSW) that manages a data infrastructure and logs experiments through an API. The processes and their relationships are captured as XML and stored in the MySQL database. ESSW uses Graphviz for generating provenance graphs visualizations from CGI scripts. It also allows accessing metadata information by clicking on the depicted graph nodes. Each experiment (process) in ESSW is related to a model (workflow template). Karma also has its own tool to visualize provenance graphs called Karma Provenance Browser. It integrates provenance, metadata, services, workflows and data products through the API service (SIMMHAN, Y.L. *et al.*, 2006).

Finally, Schreiber and Struminski (2017) introduce a visualization technique for provenance using comics strips aiming for self-explaining and easy-to-understand visualization of data provenance called PROV Comics. The comics are generated automatically from provenance graphs compliant with PROV model. They create a visual mapping between each PROV element (Agent, Activity, and Entity) and visual/textual elements such as shapes, colors, icons, letters, and labels.

2.4.3 API

Due to the transparency provided by different APIs, the use of complex algorithms can be facilitated. Some APIs also allow one to search various datasets with different structures and formats in a convenient way. In the paragraphs that follow, we present some provenance analysis approaches that rely on these features by using Web Services (DA CRUZ, S.M.S. *et al.*, 2008; SIMMHAN, Y.L. *et al.*, 2006), Java APIs (WOODMAN *et al.*, 2011), among other APIs to provide independent provenance query mechanisms.

The service-oriented Karma provenance framework provides a query mechanism through a Web service interface (SIMMHAN, Y.L. *et al.*, 2006). For querying workflow traces and provenance from data and processes, its Web service interface requires information such

as Workflow ID, Service ID and Data Product ID. Other queries can be performed over provenance data in the XML format and their results are stored in a relational database. An evaluation carried out by Simmhan *et al.* (2008) showed that the API and the Karma service client get to answer just four out of nine queries proposed by the First Provenance Challenge (MOREAU; LUDÄSCHER; *et al.*, 2008). A reimplementation of Karma, called Komadu, is presented by Suriarachchi *et al.* (2015). Komadu is a standalone and open source tool that aims at capturing and visualizing provenance from scientific tools, infrastructures, and repositories. While Karma uses OPM to represent provenance, Komadu uses the PROV model and provides an ingest API to fed the system with provenance notifications and another API to query provenance data. To allow provenance graph visualization, it uses the Cytoscape system. Differently from Karma, Komadu allows for tracking provenance starting from some data product or agent.

Similarly, Matriohska (DA CRUZ, S.M.S. *et al.*, 2008) offers a service architecture that works in cluster, grid and cloud environments. It traces and stores the history of the distributed execution process over distributed, autonomous, replicated, and heterogeneous resources. Matriohska has a provenance query API that allows scientists to query multiple data sources.

Woodman *et al.* (2011) propose operations to query provenance traces generated by workflows enacted in the e-Science Central platform (WATSON *et al.*, 2010). This platform stores provenance traces in their natural format (DAG) on Neo4j (a non-relational graph database) (MILLER, 2013). Queries can be posed using the Java library available to the Neo4j framework and leverage the facilities provided by its storage format. Neo4j provides a traversal operation that allows users to define a starting node and traversal rules (WOODMAN *et al.*, 2011). It also has a visualization tool that allows scientists to navigate backward and forward through the provenance graph.

2.5 COMPUTING RESOURCES FOR PROVENANCE ANALYTICS

Provenance data access can be performed through query languages and visualizations, but scientists also need tools that allow them to manipulate and extract useful information from provenance data. There are some computing resources available to aid them in this task as we present in this section. Figure 5 shows a timeline of such approaches.

This section is organized as follows. Section 2.5.1 presents systems that use similarity mechanisms to compare provenance data. Section 2.5.2 describes initiatives that apply data mining and algorithms to ease provenance analytics. Finally, Section 2.5.3 lists some approaches that use collaboration to analyze provenance data.

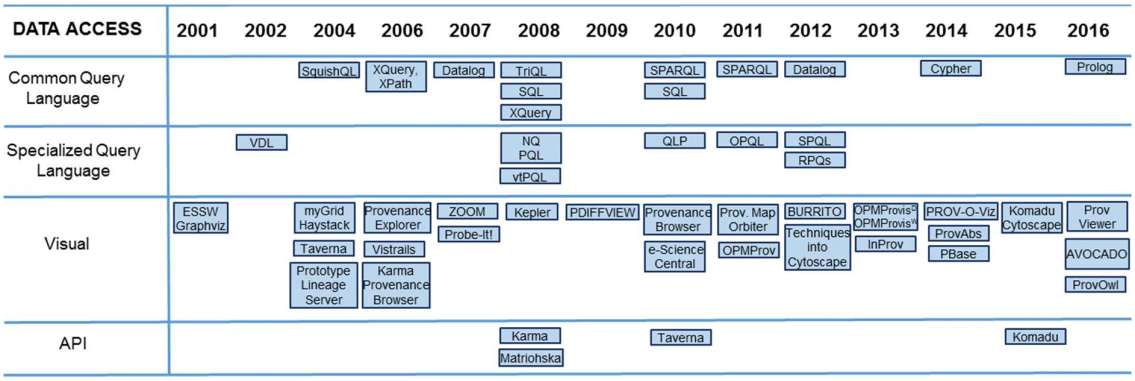


Figure 5: Timeline of Computing Resources for provenance analytics

2.5.1 SIMILARITY

Comparison of different workflow executions can help to verify mistakes and improve workflow quality. In this sense, Bao *et al.* (2009) present a prototype called PDIFFView (Provenance Difference Viewer). It works over series graphs (child nodes ordered by loop) and parallel graphs (child nodes disordered by fork) forming an SPFL (series-parallel forking and looping). It defines the difference between two executions as the minimum sequence cost of path edition operations. Transformations from one execution to another can contain the following path edition operations: insertion, deletion, expansion, and contraction. Composite modules that hold some internal difference are marked in different colors. When the user clicks on a node or edge, she can see a note about the used parameters and data. It is also possible to expand the composite module to see other details. PDIFFView allows scientists to load and save workflow specifications and executions from/to local library or import and export from/to XML files (BAO; COHEN-BOULAKIA; DAVIDSON; GIRARD, 2009).

Missier *et al.* (2013) develop an algorithm named PDIFF to analyze and compare provenance graphs. Their algorithm queries provenance traces generated by workflow executions in the e-Science Central platform (WATSON *et al.*, 2010). PDIFF assists reproducibility analysis by identifying possible causes of divergent results such as data and workflow evolution, service version upgrades (problems related to the workflow decay (ROURE *et al.*, 2011)), and also non-deterministic behavior in some of the services. It is also able to compare three different types of files: text or CSV, XML, and mathematical models. PDIFF was added to the e-Science Central platform to help scientists to detect diverging outputs when trying to reproduce workflows.

VisTrails can compute the difference between workflows. It saves the workflow evolution history and offers visual tools to compare execution results of different versions of

workflows. This mechanism computes the difference between two nodes in the history tree. VisTrails has a semi-automatic analogy mechanism. It enables scientists to apply the difference between two workflow versions over a third workflow. Such mechanism allows scientists to save time and effort to construct new workflows (FREIRE; SILVA, 2008b). Despite the fact that VisTrails provides a visual tool to compare results of different executions, it does not offer a mechanism to compare the path traversed (dependency graph) to achieve that result.

Similar to PASS and ES3, the provenance system BURRITO (GUO; SELTZER, 2012) captures implicit retrospective provenance in the OS level. BURRITO was developed to be an electronic lab notebook to researchers and can cover a wider number of domains such as bioinformatics, CS, and finance. It comprises a core (GUI window interactions, OS-level provenance, and NILFS versioning file system) and a set of plugins (audio recordings, digital sketches, sticky notes, web browsing history, text editor interactions, command invocation, and clipboard events). BURRITO stores user's activities in the MongoDB database (ABRAMOVA; BERNARDINO, 2013). To analyze the collected provenance, it has a Computational Context Viewer that offers an HTML page to graphically display the diff between different input files versions and command-line parameters and relates them to the effects over output file versions. BURRITO also has an Activity Context Viewer that is similar to the Computational Context Viewer and displays information in four fields about one version of the chosen source file: diffs, resources read, resources written, and annotations. With the more general propose, BURRITO does not relate retrospective provenance with the prospective provenance generated by one WfMS.

2.5.2 DATA MINING TECHNIQUES TO PROVENANCE ANALYTICS

According to Davidson and Freire (2008), provenance data mining can lead to the discovery of new patterns that may simplify and refine the workflow. Some of these patterns are imperceptible to human eyes and depend on automated application of data mining techniques over the dataset. Data mining generates patterns that describe and distinguish provenance dataset properties, detects lack of provenance data, and finds out more descriptive knowledge of provenance groupings (CHEN *et al.*, 2012, 2014). In the same direction, Big Data analytics try to extract useful information from data sets. However, Big Data analytics works on voluminous data sets and takes into account scalability issues. The approaches presented by Chen *et al*. (2012, 2014) and described in this section, also considers these aspects. Moreover, most of the approaches presented in this section address different types of pattern discovery problems of provenance analysis by using techniques such as clustering and association rules.

We open this section by presenting a tool that mines semantically annotated provenance (ZHAO, JUN *et al.*, 2008). Then we describe an approach to simplify large provenance datasets to be mined (CHEN *et al.*, 2012, 2014). We finish by discussing an approach to aggregate provenance graphs by types (MOREAU, 2015).

Ouzo is a system that combines different types of provenance supported by Taverna through semantic annotations. It overlaps secondary provenance on logs and primary lineage data represented by an ontology defined in RDF (ZHAO, JUN *et al.*, 2008). Using Ouzo, Zhao *et al.* (2008) present a component called Provenance Query and Answer (ProQA). ProQA mines the Ouzo database to capture provenance data, provenance abstraction, and semantic logic. It allows for abstraction over primary provenance through a set of typed views or user specifications. ProQA supports an abstraction interpretation over user tags and queries of internal and external provenance data. It offers a "provenance workflow" to analyze provenance. Its queries can be performed on nested workflows and are capable of returning provenance information generated from one or various executions. ProQA can also perform ontological inference and inference based on an informal taxonomy for provenance queries.

Chen *et al.* (2012) describe a provenance representation based on logical time aiming at reducing the feature space (number of characteristics) of large provenance datasets. They present an algorithm called Logical Clock-P that divides an OPM provenance graph in a sorted partition. After that, it organizes the representations of each subset in a sequence. They create representations for time and frequency to support clustering, classification, and association rule mining. Logical Clock-P algorithm reduces the feature space, keeping only the most important to be mined. In a more recent work (CHEN *et al.*, 2014), they evaluate the potential of data mining using their temporal representation. According to their findings, k-means is the best algorithm for clustering workflows based on such representation.

Moreau (2015) presents an approach to automatically combine provenance graphs by using an Aggregation of Provenance Types (APT). It converts provenance paths up to some length *k* into attributes (provenance types). Also, it groups nodes that have the same type. Both of these steps are performed using SPARQL queries. Moreau's approach also includes numeric values to represent the frequency of nodes and edges to enable outliers' detection. A conformance check is done by converting an APT summary into an OWL2 ontology without the frequency information.

2.5.3 COLLABORATION

Websites of social networks and blogs allow that millions of people share information about their personal life, disseminate their work, entrepreneurship, discuss ideas and clarify doubts. Such information dissemination, sharing, and discussion initiatives have motivated the use of this mass of engaged users to constitute one collective intelligence. In this sense, Viegas *et al.* (2007) describe a website named Many Eyes. This website offers an analysis environment of social data over visualizations constituted from data loaded by various users around the world. It allows users to create visualizations combining their own data with visualization components available on the website. Furthermore, it offers a blog-like discussion mechanism. Instead of scaling just to data size, Many Eyes scales the audience (discussions about created visualizations).

Based on this same approach, Freire and Silva (2008a) begin a discussion about the use of social data analysis on the comprehension, refinement, and reuse of shared provenance repositories. By querying and analyzing information in shared repositories, scientists could use the crowd wisdom to take lessons, build new experiments, and reduce the time to new insights. From this point, Mates *et al.* (2011) present a system called CrowdLabs that adopts a model used in social websites. CrowdLabs enables users to share data, workflow versions, libraries, packages, datasets, and results. It allows sharing and visualization of provenance data. CrowdLabs has a Client API that allows client applications (VisTrails, Wiki, CMS and other clients) to connect to the Web server and to publish workflows and provenance data. Moreover, VisTrails servers are interconnected to the same provenance database.

myExperiment is also a "scientific website." It provides a virtual environment to collaboration and sharing of workflows, experiments, files, research objects, groups, among other digital objects (DE ROURE *et al.*, 2009). myExperiment allows scientists to find out new workflows, download, execute and edit them in Taverna (HULL *et al.*, 2006) and then load the updated version back to myExperiment. The myExperiment website also allows workflow execution (DE ROURE *et al.*, 2009). However, myExperiment is focused on workflows that integrate Web services related to the bioinformatics domain (MATES *et al.*, 2011).

The scientific social websites myExperiment (DE ROURE *et al.*, 2009) and CrowdLabs (MATES *et al.*, 2011) offer support for searching, sharing, visualizing, and discussing provenance data. However, their infrastructure still does not allow more advanced, data mining, and 3D visualization. Furthermore, there are still scalability limitations on the size of the provenance dataset and the security model (MATES *et al.*, 2011). According to Altintas *et al.*

(2011), the “collaborative provenance” requires its own data model that extends provenance models of common workflows by the introduction of attributes that characterize the nature of user collaborations as well as their strength/weight.

2.6 DISCUSSION AND OPEN PROBLEMS

Table 1 and Table 2 summarize our survey of the main languages, computing resources, mechanisms and tools proposed to the provenance analytics and list some of their most relevant features according to our taxonomy. While Table 1 focuses on WfMS built-in approaches, Table 2 focuses on standalone approaches for provenance analytics. Most of the approaches analyze provenance in the workflow or activity granularity. We can also notice that most of the visual approaches use specialized provenance query languages, except for ZOOM, which uses Datalog, and others that do not have a query language (i.e., Provenance Map Orbiter and PDIFFView).

A large part of the provenance systems uses RDBMS, RDF, or XML to store provenance data. Some provenance systems provide common query languages directly related to these storage schemas (FOSTER *et al.*, 2002), such as SQL, SPARQL, Datalog, XQuery or XPath. For example, SciProv (GASPAR *et al.*, 2011) uses the SPARQL, Taverna (ZHAO, JUN *et al.*, 2008) uses TriQL, Karma and SciCumulus use SQL, PreServ (SIMMHAN, YOGESH L. *et al.*, 2006) uses XQuery and XPath, and Dey *et al*. (2012) use Datalog to query a data model that extends OPM. Despite the fact that there are trends in working with queries applied directly on a graph structure, aggregation operations (i.e., count, max, min, etc.) in this type of model are costly. In contrast, in the relational model, they are straightforward. On the other hand, common languages are considered high-level languages for computing science professionals, but they may not be suitable for scientists. Scientists need more user-friendly languages that allow them, for example, to elaborate recursive queries and group data in a transparent and easier manner.

All provenance analytics approaches work on explicit retrospective provenance types, except PASS, ES3, Matriohska, and BURRITO, that work just on implicit retrospective provenance. Most of them also work on prospective provenance. Provenance types are intrinsically connected to the provenance granularity. Hence, prospective and explicit retrospective provenance may be analyzed from approaches that consider workflow and activity granularity. On the other hand, approaches that take into account file, value, and/or tuple granularities may analyze implicit retrospective provenance.

Table 1: WfMs built-in approaches classified by the provenance analytics taxonomy

Approach	Access Methods	Query Language/Tool	Format/Model	Type	Granularity	Computing Resources
Taverna	Visual, Common Query Language	Process Tree, TriQL, ProQA, and Querying Algorithm by Missier *et al*. (2010)	XML, Graph, RDF, and Relational	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	Data Mining, Inference
e-Science Central	Visual, API	PDIFF, Neo4J Java API	OPM, Graph	Prospective and Explicit Retrospective	Workflow, Activity, File, Value	Collaboration, Similarity
SciProv	Common Query Language	SPARQL	RDF, OPM, Relational	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	Inference
Karma	Visual, Common Query Language, API	Karma Provenance Browser, SQL, Web Service	OPM, XML, Relational	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	-
PreServ	Common Query Language	XQuery, XPath	XML	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	-
Chimera	Specialized Query Language	Virtual Data Language (VDL)	XML and Relational Database	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	-
View	Common and Specialized Query Language, and Visual	SPARQL, SQL, OPQL, RDFPROV, OPMProvis (also standalone)	RDF, OPM, Relational	Prospective and Explicit Retrospective	Workflow, Activity, File, Value	Summarization
Swift	Specialized Query Language	SPQL, MTCProv	Relational	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	-
Kepler	Specialized Query Language, Visual	QLP, Provenance Browser (also standalone)	Relational, XML	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	Summarization
VisTrails	Specialized Query Language, Visual	vtPQL, QBE	Relational, XML	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	Similarity, Workflow Evolution
SciCumulus	Common Query Language	SQL, (GONÇALVES *et al.*, 2012)	Relational	Prospective, Explicit Retrospective, Implicit Retrospective	Workflow, Activity, File, Value, Tuple	-
Chiron	Common Query Language	SQL	Relational	Prospective, Explicit Retrospective	Workflow, Activity, File, Value, Tuple	-

Table 2: Standalone approaches classified by the provenance analytics taxonomy

Approach	Access Methods	Query Language/Tool	Format/Model	Type	Granularity	Computing Resources
Dey *et al*. (2012)	Common Query Language	Datalog	OPM, Graph, File System	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	Inference
ES3	Common Query Language	XML requests with XQuery	XML	Implicit Retrospective	OS, File, Value	-
PASS	Specialized Query Language	New Query (nq), PQL	Berkeley DB, Graph/ Semi structured Data, File System	Implicit Retrospective	OS, File, Value, Tuple	-
Provenance Map Orbiter	Visual	Graphic Filters and Timeline	OPM, RDF/N3	Explicit Retrospective	Activity and OS, File, Value	Summarization
ZOOM	Common Query Language and Visual	Datalog and Graphic filters	Relational, XML	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	Customization, Summarization, Inference
PDIFFView	Visual	PDIFFView tool	Series-parallel	Prospective, Explicit Retrospective	Workflow, Activity,	Similarity
Logical Clock-P	-	Logical Clock-P	OPM, Graph	Explicit Retrospective	Activity, File, Value	Data Mining
CrowdLabs	Visual, API	Social website	Relational, XML	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	Collaboration
myExperiment	Visual, API	Social website	XML, RDF	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	Collaboration
Matriohska	API	-	Document	Implicit Retrospective	OS, File, Value	-
Prov Viewer	Visual	-	PROV, graph	Explicit Retrospective	Activity, File, Value	Summarization
Prov-O-Viz	Visual, API	-	RDF	Implicit, Explicit Retrospective	Activity, File, Value	Inference
Provenance Explorer	Visual	-	RDF	Explicit Retrospective	Activity, File, Value	Summarization, Customization, Inference
Probe-It!	Visual	-	RDF	Explicit Retrospective	Activity, File, Value	Customization, Summarization

Table 2: Standalone approaches classified by the provenance analytics taxonomy (cont.)

Approach	Access Methods	Query Language/Tool	Format/Model	Type	Granularity	Computing Resources
Visualization Techniques into Cytoscape	Visual	-	XML	Explicit Retrospective	Activity, File, Value	Summarization, Similarity
InProv	Visual	Timeline	Files	Implicit, Explicit Retrospective	Activity, OS, File, Value	Summarization
ProvOwl	Visual	Cytoscape	PROV, XML, Graph	Explicit Retrospective	Activity, File, Value	Summarization
ProvAbs	Visual, Common Query Language	Neo4J Traverse API, Cypher	PROV, Graph	Explicit Retrospective	Activity, File, Value	Summarization
AVOCADO	Visual	-	JSon	Explicit Retrospective	Activity, File, Value	Summarization, Customization
PBase	Visual, Common Query Language	Cypher, SPARQL	ProvONE, RDF	Prospective, Explicit Retrospective	Workflow, Activity, File, Value	-
ESSW, Prototype Lineage Server	Visual Common Query Language, Visual	Graphviz SquishQL	XML, Relational RDF/XML	Explicit Retrospective Explicit Retrospective	Activity, File, Value Activity, File, Value	-
Komadu	Visual, API	Cytoscape	PROV, XML, CSV, Relational	Explicit Retrospective	Activity, File, Value	-
BURRITO	Visual	-	File, noSQL	Implicit Retrospective	Activity, OS, File, Value	Similarity
D-OPM	Specialized Query Language	RPQs variants	D-OPM, Graph, Relational, Datalog	Prospective, Retrospective	Workflow, Activity, File, Value	Inference
APT	Common Query Language	SPARQL, OWL2	RDF	Explicit Retrospective	Activity, File, Value	Summarization
myGrid	Visual	Haystack	RDF	Explicit Retrospective	Activity, File, Value	-

The study of methods and techniques to analyze provenance is still a relatively unexplored field when compared to the existing research about capture, storage, and provenance management. Some information summarization approaches (BITON, OLIVIER *et al.*, 2007; COHEN-BOULAKIA *et al.*, 2008; LIM *et al.*, 2011, 2013) try to hide details so that the scientist can focus on information that is more important to the experiment understanding, allowing them to save time. The definition of views over provenance based on profiles also benefits the analysis work (BITON, OLIVIER *et al.*, 2007), since each scientist has a different capability of perception and identification of patterns. This allows for richer inferences.

New specialized provenance query languages are being driven by provenance models such as OPM and PROV. This type of initiative can help the community to define, in the future, a standardized common provenance query language that is storage, schema, and WfMS independent. Despite being a powerful tool, the analysis of a huge provenance dataset through query languages can be arduous, both regarding query construction and performance. Furthermore, when using languages such as SPARQL to query relational databases, scientists have to face the learning challenge of a new language. They will also face limitations about nested queries and aggregations (HOLLAND, D. *et al.*, 2008).

Collaborative work has emerged as a good ally of provenance analytics since it allows using a collective intelligence. This collective intelligence can open scientific discussions to users spread across the world. However, much still needs to be done for these approaches to meet the performance and security needs of the current scientific community.

We believe the refinement of all these analysis methods and tools and their intensive use by scientists will allow a huge acceleration of science, on an even larger scale than that occurred with the popularization of the WfMS. Until then, there is a huge path ahead for computer scientists. In this thesis, we propose an integrated provenance analysis approach that allows scientists to traverse heterogeneous provenance graphs. Our goal is to reduce the gap between the different provenance formats and models and solve the problems faced by the scientists in the analysis of heterogeneous traces.

Chapter 3 - QUERYING PROVENANCE ACROSS HETEROGENEOUS PROVENANCE GRAPHS

3.1 INTRODUCTION

Provenance generated by different workflow systems is generally expressed using different formats. This is not an issue when scientists analyze provenance graphs in isolation, or when they use the same workflow system. However, analyzing heterogeneous provenance graphs from multiple systems poses a challenge. The provenance graphs generated by these workflow systems can have different formats (*i.e.,* RDF, XML, relational tables, etc.) and a proprietary structure. Hence, analyzing provenance across these heterogeneous graphs and exploring the possible intersections between them become a hard and error prone task for scientists.

To address the aforementioned problem, we propose a reference classification and a provenance integration architecture that adopts ProvONE as an integration model and show how different provenance databases can be converted to a global ProvONE schema. Scientists can then query this integrated database, exploring and linking provenance across several different workflows that may represent different implementations of the same experiment. To illustrate the feasibility of our approach, we developed conceptual mappings between the provenance databases of four workflow systems (e-Science Central, SciCumulus, Taverna, and VisTrails). We provide *cartridges* that implement these mappings and generate an integrated provenance database expressed as Prolog facts. To demonstrate its usage, we have developed a set of Prolog rules that enable scientists to query the integrated database.

The choice of Prolog comes naturally in this scenario, since, besides being able to represent a wide variety of data (MURTA *et al.*, 2014; OLIVEIRA *et al.*, 2016), it also allows for more expressive power than SQL since it adds inference capabilities. Datalog would also provide these same advantages. In fact, syntactically, Datalog is a subset of Prolog and its clauses can be parsed and executed by a Prolog interpreter (CERI *et al.*, 1989). Although Datalog is more efficient on relational database queries (it processes the whole relation by using a set-oriented approach rather than the one-tuple-at-time approach of Prolog), Datalog limits the way rules can be written. As an example, it does not allow a rule to use a variable that does not appear in its body (CHONG, 2016). Thus, examples such as the one on Figure 6 would not be allowed in Datalog.

```
foo(X, Y) :- bar(X).
foo(X, Y) :- bar(Y).
foo(X, Y) :- foo2(X), foo2(Y).
```

Figure 6. Example of rule that cannot be expressed in Datalog

The remainder of this chapter is organized as follows. Section 3.2 describes the workflows used as running examples and the semantic mapping between them. A reference classification of the provenance space as well as the mapping between the different WfMs are presented in Section 3.3. Section 3.4 describes the integration architecture. Section 3.5 presents Prolog rules and queries. Finally, Section 3.6 concludes this chapter.

3.2 RUNNING EXAMPLES

This Section aims at presenting two types of workflows: one that aims at performing phylogenetic analysis, and another one that focuses on diagnosing patients. We use them as running examples. Each of these workflows is implemented in four different WfMS (Taverna, SciCumulus, VisTrails, and e-Science Central). We have executed all of them aiming at collecting provenance. The semantic mapping between the implementations of each of the workflows is made by linking similar activities.

3.2.1 PHYLOGENETIC ANALYSIS WORKFLOW

Our first example is a phylogenetic analysis experiment designed by four research groups and executed in four different WfMS. This analysis aims at generating phylogenetic trees from DNA, RNA and amino acid sequences, along with other statistics, which can then be used to infer the evolutionary relationship of a set of genes, species, or other taxa (a group of one or more populations of an organism or organisms used to form a biological unit). This experiment is modeled by four workflows named SciPhy, ML, SciEvol, and Phylo.

As illustrated in Figure 7, the SciPhy workflow consists of five activities: (i) DataSelection; (ii) Mafft; (iii) ReadSeq; (iv) ModelGenerator; and (v) RAxML. The ML workflow is composed of six activities: (i) ImportFile; (ii) FilterDuplicates; (iii) ClustalW; (iv) MEGA-Maximum Likelihood; (v) CSVExport; and (vi) ExportFiles. The SciEvol workflow has four activities: (i) Mafft; (ii) ReadSeq; (iii) RAxML, and (iv) Codeml. Finally, the Phylo workflow has 8 activities: (i) FindDir; (ii) Clear; (iii) Alignment; (iv) Convertion; (v) Evolutionary Model; (vi) GenerateTree1; (vii) GenerateTree2; and (viii) GenerateTree3. All workflows were set up with similar input data and parameters. Although the number of activities differs among them, two key activities appear in all workflows, namely *sequence alignment* and *tree generation*. Their mappings (Mafft ≡ ClustalW and RAxML ≡ MEGA ≡

RAxML/Codeml ≡ GenerateTree1/ GenerateTree2/ GenerateTree3) help us compare the critical elements of the workflows. The remaining activities are responsible for format conversions and some optional optimizations in the process.

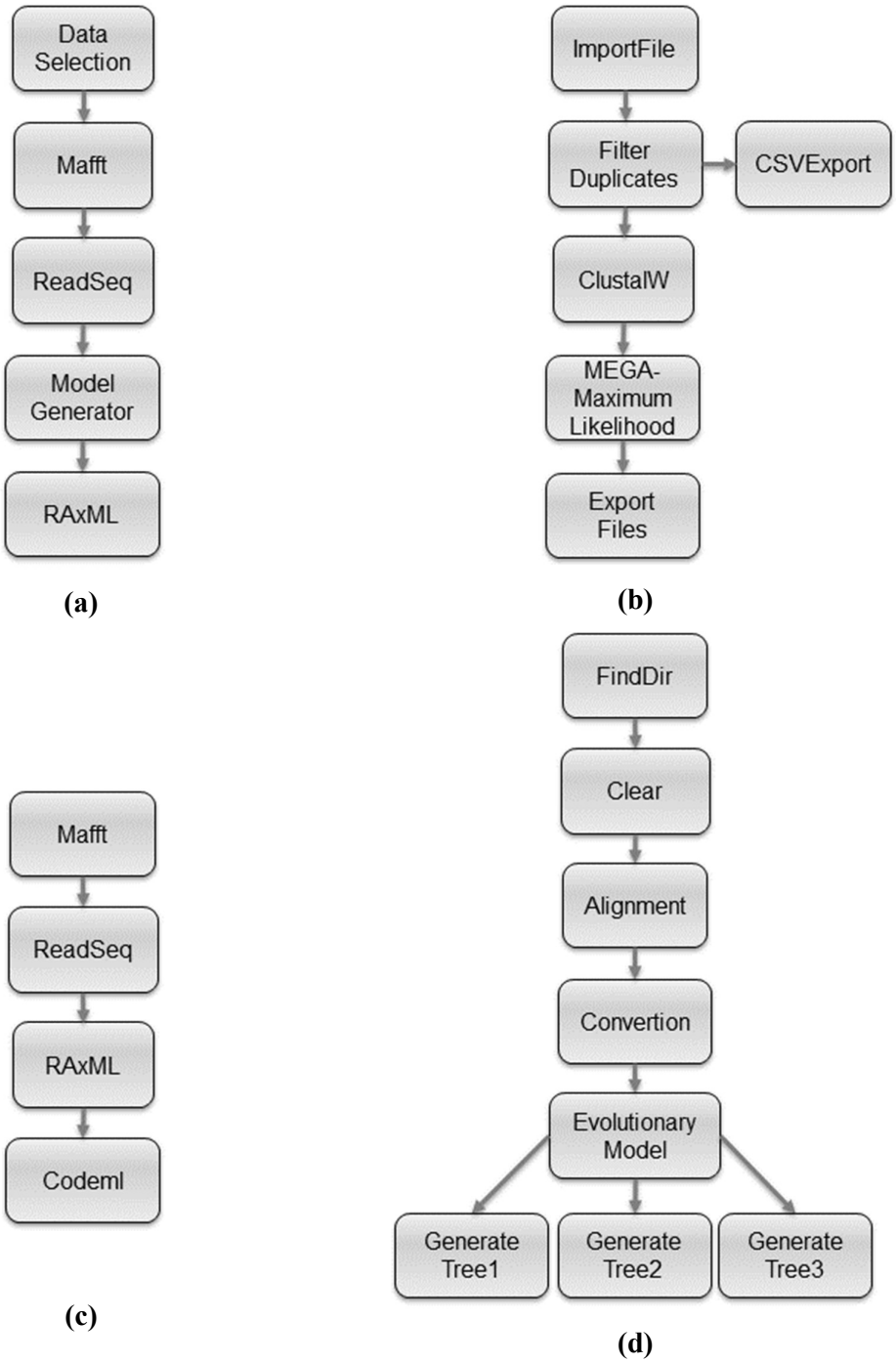


Figure 7. Four Phylogenetic Analysis Workflow implementations: (a) SciPhy, (b) ML, (c) SciEvol, and (d) Phylo.

These workflow abstractions presented in Figure 7 could be set up by a collaborative group of scientists that work independently on similar goals (*i.e.,* generated phylogenetic trees

for evolutionary analysis). Despite the fact that they adopt slightly different methods and procedures and thus producing workflows that differ in design, implementation, and execution middleware, the workflows are similar in terms of intent. Additionally, they use comparable tools and algorithms. Since the phylogenetic analysis workflow implementations use either the same or similar input data and produce similar outputs, it seems natural to try and use the provenance traces of their executions to compare and discuss produced results. However, the heterogeneity in the design, implementation, and execution of these workflows translates into provenance traces that are themselves heterogeneous, making it difficult to analyze them jointly.

3.2.2 DIAGNOSIS ANALYSIS WORKFLOWS

The Diagnosis Analysis workflow automate the clinical pathogenic diagnosis for patient by using the gene variants of a person (obtained from the genome processing) along with external data sources. Each implementation of this workflow receives as input variant records (single-nucleotide mutations or indels) of a patient and a set of phenotypes and it is able to classify genes as pathogenic, benign, or unknown (MISSIER *et al.*, 2015). The classification is performed by using information from external knowledge sources such as OMIM[4] and ClinVar[5], which holds the disease-gene and disease-variant mappings, respectively. There are four workflows that implement Diagnosis Analysis named SVI, Pathogenesis, GeneClass, and PatientDiag. Those workflows were also designed and executed by four different WfMS.

As depicted in Figure 8, the SVI (Single-nucleotide Variant Integration) workflow is composed of seven activities: (i) Patient_Filter; (ii) Gene_in_scope; (iii) Patient_Gene_Join; (iv) Clinvar_Right_Join; (v) Filter_Pathogenic; (vi) Filter_Benign; and (vii) Filter_Amber. The Pathogenesis workflow consists of five activities: (i) FilterVariants; (ii) FilterGene; (iii) JoinVariantGene; (iv) JoinClinvar; and (v) Classify. The GeneClass workflow also holds five activities: (i) FilterVariants; (ii) FilterGene; (iii) JoinVariantGene; (iv) JoinClinvar; and (v) Classify. In the same way, PatientDiag includes five activities: (i) filter_variant; (ii) filter_gene; (iii) join_var_gene; (iv) join_clinvar; and (v) classify_clinvar.

The four workflows shown in Figure 8 are set up with similar activities, input data, and parameters. As aforementioned in the phylogenetic workflow analysis scenario, other collaborative groups of scientists, which work on diagnosis analysis, could also have interest in discussions about the results produced by the different workflow implementations. After

[4] https://omim.org/downloads/
[5] ftp://ftp.ncbi.nlm.nih.gov/pub/clinvar/

running their workflows, they can compare and analyze provenance traces from distinct diagnosis workflows, improve the workflow settings and experiment results. Although the number of the workflow activities differs, all of them are related and have similar goals.

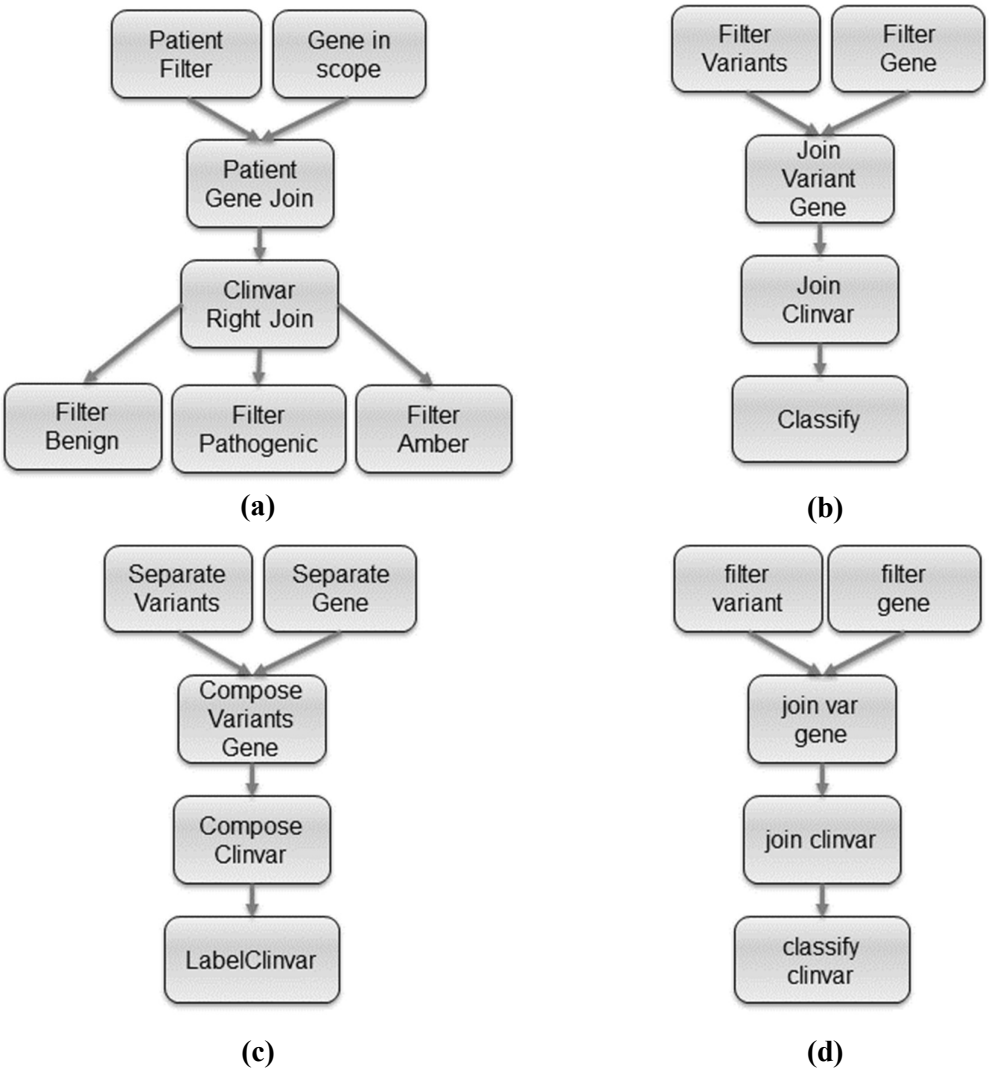


Figure 8. Four Diagnosis Analysis Workflow implementations: (a) SVI, (b) Pathogenesis, (c) GeneClass, and (d) PatientDiag.

3.2.3 SEMANTIC MAPPING

Each type of workflow (Phylogenetic and Diagnosis) was implemented using different design and specification (*e.g.* number and name of activities), but each group of four workflows has similar goals, which make it useful to compare the achieved results. To clarify the use of specific parameter values in both the Phylogenetic and Diagnosis workflows, domain experts from each group defined semantic mappings between pairs of workflow activities in all of the

workflow implementations, as presented in Table 3 and Table 4. We use this mapping to compare the provenance of similar or equal data and activities from distinct and heterogeneous provenance graphs, and later to drive the design of cross-traces queries. The related set of data and activities are placed inside of each bracket after the *label*. In this way, a single query can go across two or more provenance graphs and bring together information related to equal or similar data and activities.

Table 3. Semantic relationships among activities of four implementations of the Phylogenetic analysis workflow

SciPhy	ML	SciEvol	Phylo	Description
DataSelection	ImportFile and FilterDuplicates	-	FindDir, Clear	Importing, filtering, and selection of data.
Mafft	ClustalW	Mafft	Alignment	Sequence alignment.
ReadSeq	-	ReadSeq	Convertion	Conversion of alignment format.
ModelGenerator	-	-	Evolutionary Model	Choice of the evolutionary model.
RAxML	MEGA-Maximum Likelihood and ExportFiles	RAxML, Codeml	GenerateTree1 GenerateTree2 GenerateTree3	Generation of the phylogenetic tree.
-	CSVExport	-	-	Exporting filtered sequences on CSV format.

Table 4. Semantic relationships among activities of the four implementations of the Diagnosis Analysis workflow

SVI	Pathogenesis	GeneClass	PatientDiag	Description
Patient_Filter	FilterVariants	SeparateVariants	filter_variant	Filtering of patient's variants
Gene_in_scope	FilterGene	SeparateGene	filter_gene	Filtering of gene map
Patient_Gene_Join	JoinVariantGene	ComposeVariantsGene	join_var_gene	Joining of filtered patient's variants and gene map
Clinvar_Right_Join	JoinClinvar	ComposeClinvar	join_clinvar	Joining the previous result with clinvar data
Filter_Pathogenic, Filter_Benign, and Filter_Amber	Classify	LabelClinvar	classify_clinvar	Classification of the gene as pathogenic, benign or amber

3.3 PROVENANCE ANALYSIS ACROSS HETEROGENEOUS PROVENANCE GRAPHS

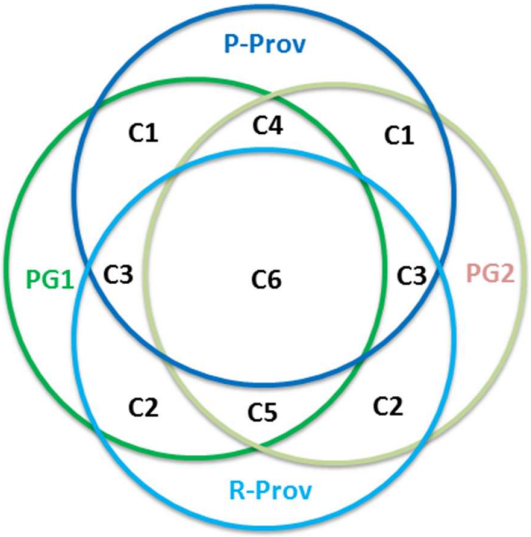
This section describes a reference classification that illustrates the different kinds of intersections between *p-prov*, *r-prov*, and single or multiple provenance graphs. Then, it introduces a mapping among entities and relationships of ProvONE and four proprietary models. Finally, it shows ProvONE assertions as Prolog facts.

3.3.1 A REFERENCE CLASSIFICATION OF THE PROVENANCE SPACE AND OF ITS QUERIES

We argue that, in the collaborative scenario outlined in the introduction of this thesis, scientists can benefit from provenance graphs that (a) include both *p-prov* and *r-prov*, and (b) include traces from similar workflows. The combination of *p-prov* and *r-prov* has been proposed before in many papers (BELHAJJAME *et al.*, 2015; COSTA *et al.*, 2013a; MISSIER; SAHOO; *et al.*, 2010), and *p-prov* enables new types of queries to be made on *r-prov*, such as *find all data produced by any activity that occurs downstream from block X in the workflow*. Other interesting queries that span *r-prov* and *p-prov* are presented later in this section. The case for point (b) is that the ability to perform analysis on combined provenance graphs will help collaborative teams to obtain deeper understanding from related workflows with different levels of details. As we have seen in the running examples of this chapter, this is possible because these workflows typically share similarities on their activities, data flows, or input parameters. When detailed provenance graphs from similar workflows are available, scientists can use those sources to clarify their understanding and get more insights about the experiment.

Given two provenance traces *PG1* and *PG2*, each from a different workflow run (from the same or different workflow implementations), and each providing both *r-prov* and *p-prov*, we can categorize the set of all possible provenance queries as illustrated in Figure 9. In this Venn diagram, queries are classified according to the provenance data needed to answer them. For instance, queries in class *C1* operate on *p-prov* only and on one graph at a time, while *C3* queries require both *p-prov* and *r-prov*, on one graph. Class *C6* is perhaps the most challenging, as it operates simultaneously on *p-prov* and *r-prov*, and on both graphs. Note that our classification is conceptual, and the actual fragment returned by a query is sensitive to the values of query parameters.

Example queries for each of the classes are listed in Table 5. Note that queries from classes C1, C2 and C3 are easily answered using provenance captured by most WfMS. However, queries of classes C4, C5 and C6 require additional mapping information that is not automatically provided by those systems. This mapping encompasses two aspects: (a) a syntactic mapping between heterogeneous schemas of provenance data and (b) a semantic mapping that informs the similarity or equivalence between *p-prov* elements. The syntactic mapping of local and global provenance schemas using ProvONE is described in the next section, while a sample of a semantic mapping of four workflows specifications for the phylogenetic analysis experiment appears in Table 3.



Class	P-Prov	R-Prov	Single Graph	Both Graphs
C1	✓	✗	✓	✗
C2	✗	✓	✓	✗
C3	✓	✓	✓	✗
C4	✓	✗	✗	✓
C5	✗	✓	✗	✓
C6	✓	✓	✗	✓

Figure 9. Classification of provenance fragments and corresponding queries

Table 5. Provenance queries on intersection classes

#	Queries	Class
Q1	Retrieve all programs with their input and output ports for the workflow *w'* and provenance graph *g'*.	C1
Q2	Retrieve all activity executions with their generated data for the workflow execution *w'* and provenance graph *g*'.	C2
Q3	Retrieve the time consumed by each activity execution for the workflow execution *w'* and provenance graph *g*'.	C2
Q4	Retrieve the complete activity execution trace that influenced the generation of the data *d'*.	C2
Q5	Retrieve the complete dataflow trace of the output data *d'* for the workflow execution *w'* and provenance graph *g*'.	C2
Q6	Retrieve all programs (plans) of each execution and their input parameters for the workflow execution *w'* and provenance graph *g*'.	C3
Q7	Retrieve the workflow version, and the time consumed by each workflow execution for the workflow *wf'* and provenance graph *g*'.	C3
Q8	Retrieve all programs with their input and output ports for each workflow specification.	C4
Q9	Retrieve all activity executions with their generated data for each workflow execution.	C5
Q10	Retrieve the time consumed by each activity execution for each workflow execution.	C5
Q11	Retrieve the ports, workflow executions, provenance graphs, and the complete activity execution trace that influenced the generation of all data.	C6
Q12	Retrieve the complete dataflow trace and workflow for each workflow execution.	C6
Q13	Retrieve the time consumed by each workflow execution for each workflow and provenance graph.	C6
Q14	Retrieve all programs (plans) of each activity execution and their input parameters for each workflow *wf*'.	C6

Note that the semantic mapping is informed by researchers/domain experts. It is used as support information to compare data used or generated by similar activities. Later, we will come

back to the queries and classes presented in this section and we will demonstrate how an integrating architecture enables their implementation.

3.3.2 MAPPING PROVENANCE MODELS TO PROVONE

Executing queries in each of the classes presented in Figure 9 requires converting *PG1, PG2, … PGN* to a common provenance model. We now illustrate the integration process using four different WfMSs, SciCumulus, Taverna, VisTrails, and e-Science Central. As mentioned before, SciPhy, SciEvol, Phylo, and ML, which run on each of these WfMS respectively, share the common goal of generating phylogenetic trees while SVI, Pathogenesis, GeneClass, and Patient_Diag also run on those WfMS and share the common goal of generating patient's diagnosis. All WfMS collect provenance data at different levels of detail, and heterogeneity is present both in format as well as in content.

SciCumulus captures *p-prov* and *r-prov* and stores them in a relational database (tables) in a PostgreSQL database system. VisTrails also captures *p-prov* and *r-prov* and records them as XML files or in a relational database (MySQL). Taverna stores *p-prov* and *r-prov* as RDF files described by a compact W3C Recommendation Language named Turtle[6] and following the PROV and Wf4Ever (wfdesc, wfprov)[7] provenance models. Finally, e-Science Central stores just *r-prov* as a graph in a Neo4J database. However, it maintains information about the workflow structure in a relational database (PostgreSQL) enriched with several additional data related to the workflow viewing (*i.e.*, coordinates of each graph object) and exports it to JSON files.

We use ProvONE (Figure 10) as a global schema for integrating provenance traces produced by the four systems. ProvONE extends the PROV model with an explicit representation of *p-prov*, thus capturing the most relevant information on scientific workflow processes, and is designed to accommodate extensions for specific scientific workflow systems (MISSIER; DEY; *et al.*, 2013).

Table 6 and Table 7 describes the logical mapping between elements of the four source provenance traces, and the corresponding ProvONE elements. The structure of each relational table from SciCumulus and VisTrails, the RDF file from Taverna, and JSON file from e-Science Central, which hold *p-prov*, were mapped to the corresponding ProvONE entities and relationships (light blue rectangles in Figure 10). Furthermore, the nodes and edges of e-Science Central database (Neo4J), relational tables of SciCumulus and VisTrails, and RDF triples of

[6] https://www.w3.org/TR/turtle/
[7] http://wf4ever.github.io/ro/

Taverna that hold *r-prov* were mapped to ProvONE entities and relationships (dark yellow rectangles in Figure 10). The gaps in the SciCumulus, VisTrails, Taverna, and e-Science Central column indicate missing information.

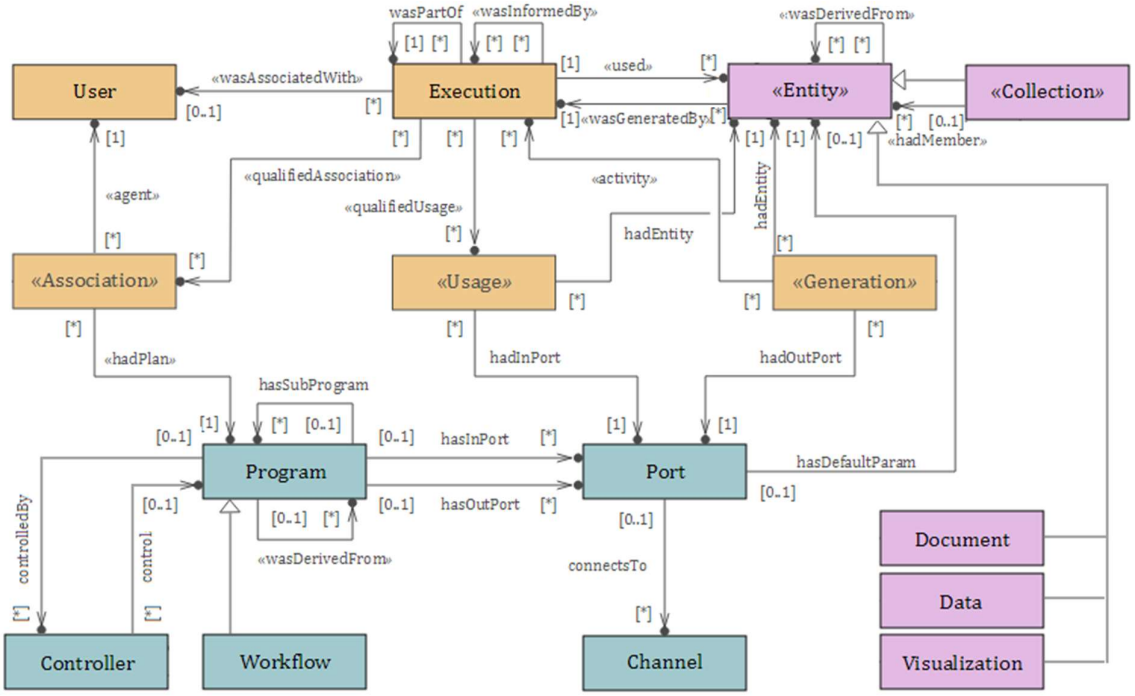


Figure 10. ProvONE conceptual model, from the DataONE documentation[8]

As there is no previous relation between *p-prov* and *r-prov* in the e-Science Central database and the exported JSON files, we use some information such as invocations (activities call) and blocks (or activities) identifiers to unify them. The relation between *p-prov* and *r-prov* is straightforward in SciCumulus since it stores *p-prov* and *r-prov* in the same database during the workflow execution (*i.e.* at runtime). VisTrails and Taverna also generate *p-prov* that can be exported as relational tables and Turtle files respectively.

3.3.3 PROVONE ASSERTIONS AS PROLOG FACTS

As aforementioned, we use ProvONE as our canonical model. By using ProvONE as a bridge to link entities and relationships from heterogeneous provenance structures, we bring all data together facilitating the query and analysis process. To leverage these benefits, we have to stablish a mapping between the source (from different WfMSs) and target (ProvONE) structures.

[8] http://jenkins-1.dataone.org/jenkins/view/Documentation%20Projects/job/ProvONE-Documentation-trunk/ws/provenance/ProvONE/v1/provone.html

In this section, we show the mapping and examples of how provenance traces from specific workflow executions can be represented as Prolog facts. We have chosen Prolog as it allows great flexibility both in producing the integrated database (provenance relationships are translated to facts) and in formulating powerful queries with inference capability (rules).

Four fragments of provenance graphs for e-Science Central, VisTrails, Taverna, and SciCumulus, respectively, are depicted in Figure 11 and Figure 12, Figure 13, and Figure 14 after mapping to ProvONE. Gray boxes represent *p-prov*, orange boxes correspond to *r-prov*, and light blue boxes are entities (*p-prov* and *r-prov*). Since all four provenance graphs are represented using the same model, queries can easily traverse all of these provenance graphs.

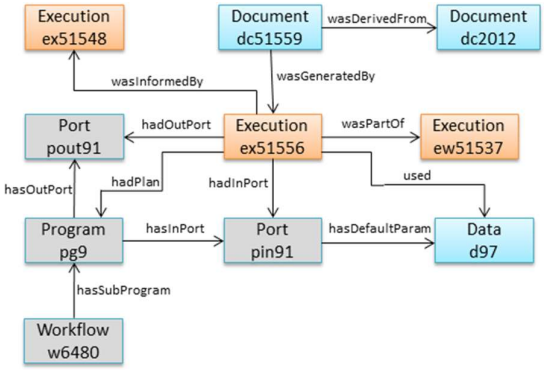


Figure 11. Part of e-Science Central provenance for a phylogenetic workflow

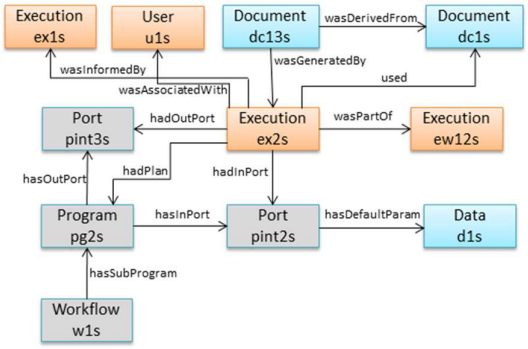


Figure 12. Part of SciCumulus provenance for a phylogenetic workflow

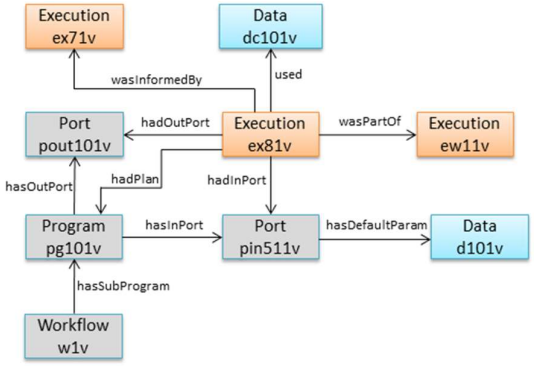


Figure 13. Part of VisTrails provenance for a diagnosis workflow

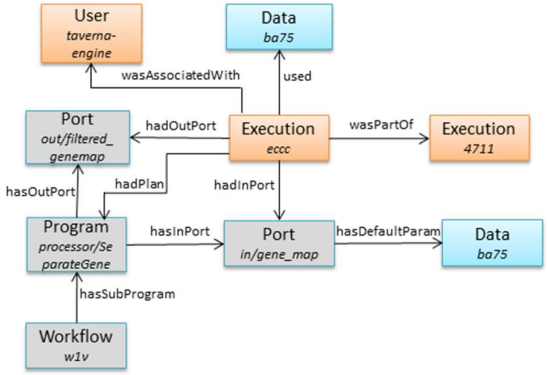


Figure 14. Part of Taverna provenance for a diagnosis workflow

Table 6 and Table 7 present the mapping between SciCumulus, e-Science Central, VisTrails, and Taverna schemas and ProvONE. Table 8, Table 9, Table 10, and Table 11 present examples of Prolog facts for the aforementioned workflow fragments (the complete set of facts and rules is available at GitHub at https://github.com/dew-uff/integrated-provenance-analysis). As the syntax of Prolog facts is similar to the PROV-N notation, each entity and activity was named and labeled in a similar style, using an identifier followed by a set of properties

Table 6. Mapping between ProvOne, SciCumulus, e-Science Central, VisTrails, and Taverna provenance models

#	ProvONE	SciCumulus	e-Science Central	VisTrails	Taverna
1	provone:workflow	cworkflow	invocation	workflow	wfdesc:Workflow
2	provone:program	cactivity	blocks	module, annotation	wfdesc:Process
3	provone:port	crelation	connections	port, function	prov:Role, wfdesc:Input, wfdesc:Output
4	provone:execution	eworkflow, eactivity, eactivation	Service Run, Workfow Run	module_exec, annotation	wfprov:ProcessRun
5	provone:execution (Workflow Execution)	eworkflow, eactivity, eactivation	Service Run, Workfow Run	workflow_exec	wfprov:WorkflowRun
6	provone:user	emachine	-	machine	prov:agent, prov:Association
7	provone:document	efile	DataVersion	parameter, function	wfprov:Artifact
8	provone:data	idataselection, odataselection, omafft, oreadseq, omodelgenerator, oraxml	properties	parameter, function	wfprov:Artifact
9	provone:visualization	-	-	thumbnail	wfprov:Artifact
10	provone:hadPlan	eactivation, eactivity, cactivity, eworkflow, cworkflow	Service Run, blocks	workflow_exec, workflow, module, module_exec	wfprov:ProcessRun, wfprov:describedByProcess
11	prov:wasDerivedFrom (Data)	efile, cmapping	Used, DataVersion	-	wfprov:Artifact, prov:used, prov:wasGeneratedBy
12	prov:wasDerivedFrom (Program)	-	Run_Of, Instance_Of, Service Run, Service Version, Workflow Version	-	-
13	prov:used	efile, cmapping	Used, DataVersion, Service Run	module_exec, parameter, function, workflow_exec	wfprov:ProcessRun, prov:used

Table 7. Mapping between ProvOne, SciCumulus, e-Science Central, VisTrails, and Taverna provenance models (cont.)

#	ProvONE	SciCumulus	e-Science Central	VisTrails	Taverna
14	prov:wasGeneratedBy	eafile	Was_Generated_By, DataVersion, Service Run	-	wfprov:Artifact, prov:wasGeneratedBy
15	prov:wasAssociatedWith	eactivation, emachine	-	workflow_exec, machine	wfprov:ProcessRun, prov:wasAssociatedWith
16	prov:wasInformedBy	cmapping	Used, Was_Generated_By, Service Run	module_exec, port	wfprov:ProcessRun, prov:wasInformedBy
17	provone:hasInPort	crelation, cmapping, cactivity	blocks, connections	module, port	wfdesc:hasInput, wfdesc:Process
18	provone:hasOutPort	crelation, cmapping, cactivity	blocks, connections	module, port	wfdesc:hasOutput, wfdesc:Process
19	provone:hasSubProgram	cworkflow, cactivity	invocation, blocks	workflow, module	wfdesc:Workflow, wfdesc:hasSubProcess
20	provone:hasDefaultParam	cfield	connections, properties	parameter, function	wfprov:describedByParameter
21	provone:wasPartOf	eworkflow, eactivity, eactivation	Contained, Service Run	workflow_exec, module_exec	wfprov:wasPartOfWorkflowRun, wfprov:ProcessRun
22	provone:hadInPort	crelation, cmapping, cactivity, eactivity, eactivation	Service Run, connections	module, module_exec, port	prov:Role, wfdesc:Input
23	provone:hadOutPort	crelation, cmapping, cactivity, eactivity, eactivation	Service Run, connections	module, module_exec, port	prov:Role, wfdesc:Output

Table 8. Prolog instances for each SciCumulus, e-Science Central, VisTrails, and Taverna ProvOne construct of a phylogenetic workflow

#	Prolog Instances for SciCumulus	Prolog Instances for e-Science Central	Prolog Instances for VisTrails	Prolog Instances for Taverna
1	entity(w1s,[prop(prov:type,['prov:plan', 'provone:workflow']),prop(prov:label,'sciphy')]).	entity(w6480,[prop(prov:type,['prov:plan', 'provone:workflow']),prop(prov:label,'ML Pipeline')]).	entity(w1v,[prop(prov:type,['prov:plan','provone:workflow']),prop(prov:label,'SciEvol')]).	entity('http://ns.taverna.org.uk/2010/workflowBundle/bf1675f4-adc3-41dd-829c-7cfd1888e02b/workflow/Workflow1/',[prop(prov:type,['prov:plan','provone:workflow']),prop(prov:label,'Workflow1')]).
2	entity(pg2s,[prop(prov:type,['prov:plan', 'provone:program']),prop(prov:label,'mafft')]).	entity(pg9,[prop(prov:type,['prov:plan','provone: program']),prop(prov:label,'CSVExport')]).	entity(pg391v,[prop(prov:type,['prov:plan','provone:program']),prop(prov:label,'RAxML 7.2.8')]).	entity('http://ns.taverna.org.uk/2010/workflowBundle/bf1675f4-adc3-41dd-829c-7cfd1888e02b/workflow/Workflow1/processor/Convertion/',[prop(prov:type,['prov:plan','provone:program']),prop(prov:label,'Convertion')]).
3	agent(u1s,[prop(prov:type,['provone:user']),prop(prov:label,'wellington-VirtualBox')]).	-	agent(u11v,[prop(prov:type,['provone:user']),prop(prov:label,'wellmor')]).	agent('taverna-engine',[prop(prov:type,['provone:user']),prop(prov:label,'TavernaEngine')]).
4	entity(dc13s,[prop(prov:type,['provone: document']),prop(prov:label,'FILE13'), prop(prov:value,'ORTHOMCL256.mafft')]).	entity(dc51559,[prop(prov:type,['provone:document']),prop(prov:label,'sequence-map.csv'), prop(prov:type,'null'),prop(prov:value,'null')]).	entity(d1641v,[prop(prov:type,['provone:data', 'String']),prop(prov:label,'MafftDir'),prop(prov:value,'C:/bda/mafft-win')]).	entity('http://ns.taverna.org.uk/2011/data/a7433bae-822a-43fa-896c-7073b13da84b/error/a313534c-6c4e-40d1-94d4-0cad73bb3e0e/0',[prop(prov:type,['provone:data']),prop(prov:label,'model'),prop(prov:value,'a313534c-6c4e-40d1-94d4-0cad73bb3e0e.err')]).
5	hadPlan(ex2s,pg2s).	hadPlan(ex51556,pg9).	hadPlan(ex31v,pg391v).	hadPlan('http://ns.taverna.org.uk/2011/run/a7433bae-822a-43fa-896c-7073b13da84b/process/77bedf56-baa0-4c6a-ad24-bd28f3838f21/','http://ns.taverna.org.uk/2010/workflowBundle/bf1675f4-adc3-41dd-829c-7cfd1888e02b/workflow/Workflow1/processor/Convertion/').

Table 9. Prolog instances for each SciCumulus, e-Science Central, VisTrails, and Taverna ProvOne construct of a phylogenetic workflow (cont.)

#	Prolog Instances for SciCumulus	Prolog Instances for e-Science Central	Prolog Instances for VisTrails	Prolog Instances for Taverna
6	wasDerivedFrom(dc13s,dc1s).	wasDerivedFrom(dc51559,dc2012).	-	wasDerivedFrom('http://ns.taverna.org.uk/2011/data/a7433bae-822a-43fa-896c-7073b13da84b/error/00aa1dfd-48a9-4fb4-878e-8ae728e6bf9d/0','http://ns.taverna.org.uk/2011/data/a7433bae-822a-43fa-896c-7073b13da84b/error/a313534c-6c4e-40d1-94d4-0cad73bb3e0e/0').
7	-	wasDerivedFrom(pg9, pgV50025).	-	-
8	used(ex2s,dc1s).	used(ex51556,d97).	used(ex51556,d97).	used('http://ns.taverna.org.uk/2011/run/a7433bae-822a-43fa-896c-7073b13da84b/process/76ee6ee2-cbcc-46ec-a29f-b696f9908ba4/','http://ns.taverna.org.uk/2011/data/a7433bae-822a-43fa-896c-7073b13da84b/error/a313534c-6c4e-40d1-94d4-0cad73bb3e0e/0').
9	wasGeneratedBy(dc13s,ex2s).	wasGeneratedBy(dc51559,ex51556).	-	wasGeneratedBy('http://ns.taverna.org.uk/2011/data/a7433bae-822a-43fa-896c-7073b13da84b/error/00aa1dfd-48a9-4fb4-878e-8ae728e6bf9d/0','http://ns.taverna.org.uk/2011/run/a7433bae-822a-43fa-896c-7073b13da84b/process/419b56fd-d66c-4750-ab15-fb3083f3ffac/').
10	wasAssociatedWith(ex2s,u1s).	-	wasAssociatedWith(ew11v,u11v).	wasAssociatedWith('http://ns.taverna.org.uk/2011/run/a7433bae-822a-43fa-896c-7073b13da84b/process/76ee6ee2-cbcc-46ec-a29f-b696f9908ba4/','taverna-engine').

Table 10. Prolog instances for each SciCumulus, e-Science Central, VisTrails, and Taverna ProvOne construct of a diagnosis analysis workflow

#	Prolog Instances for SciCumulus	Prolog Instances for e-Science Central	Prolog Instances for VisTrails	Prolog Instances for Taverna
1	entity(w1s,[prop(prov:type,['prov:plan','provone:workflow']),prop(prov:label,'patientdiag')]).	entity('esc:svi-esc/workflow/694/30049',[prop(prov:type,['prov:plan','provone:workflow']),prop(prov:label,'SVI')]).	entity(w1v,[prop(prov:type,['prov:plan','provone:workflow']),prop(prov:label,'Pathogenesis')]).	entity('http://ns.taverna.org.uk/2010/workflowBundle/ea0b115a-fbda-4caa-9f9b-e015fa884ed5/workflow/Workflow8/',[prop(prov:type,['prov:plan','provone:workflow']),prop(prov:label,'Workflow8')]).
2	entity(pg1s,[prop(prov:type,['prov:plan','provone:program']),prop(prov:label,'filter_variant')]).	entity('esc:svi-esc/block/blocks-core-manipulation-3rowjoin/797',[prop(prov:type,['prov:plan','provone:program']),prop(prov:label,'3wayRowJoin')]).	entity(pg141v,[prop(prov:type,['prov:plan','provone:program']),prop(prov:label,'GeneMap')]).	entity('http://ns.taverna.org.uk/2010/workflowBundle/ea0b115a-fbda-4caa-9f9b-e015fa884ed5/workflow/Workflow8/processor/ComposeVariantsGene/',[prop(prov:type,['prov:plan','provone:program']),prop(prov:label,'ComposeVariantsGene')]).
3	agent(u1s,[prop(prov:type,['provone:user']),prop(prov:label,'wellmor')]).	agent('esc:svi-esc/engine/IP:10.2.0.5',{'esc:Architecture':"x86_64",'esc:CPUModel':"Xeon",'esc:CPUVendor':"Intel",'esc:OS':"Linux"}).	agent(u11v,[prop(prov:type,['provone:user']),prop(prov:label,'wellmor')]).	agent('taverna-engine',[prop(prov:type,['provone:user']),prop(prov:label,'TavernaEngine')]).
4	entity(d28s,[prop(prov:type,['provone:data']),prop(prov:label,'varpath'),prop(prov:value,'svi-classification-MUN0785-CV1602GM160308.csv')]).	entity('esc:svi-esc/document/2175/2176',[prop(prov:type,['provone:document']),prop(prov:label,'genemap2-161031-esc'),prop(prov:value,'genemap2-161031-esc.txt')]).	entity(d51v,[prop(prov:type,['provone:data']),prop(prov:label,'value'),prop(prov:value,'variant_summary.csv')]).	entity('http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/fbca0988-9237-4e3a-abf3-d78013741817',[prop(prov:type,['provone:data']),prop(prov:label,'join_variants_gene'),prop(prov:value,'join_variants_gene')]).
5	hadPlan(ex1s,pg1s).	hadPlan('esc:svi-esc/invocation/30115/block/58644E52-D670-0900-6105-0CD62F6E70C7','esc:svi-esc/block/blocks-core-manipulation-3rowjoin/797').	hadPlan(ew11v,w1v).	hadPlan('http://ns.taverna.org.uk/2011/run/76676230-4711-4a10-b7bc-9a0d093dc526/process/cff62007-0684-410f-a9a3-7279c67c4fa1/','http://ns.taverna.org.uk/2010/workflowBundle/ea0b115a-fbda-4caa-9f9b-e015fa884ed5/workflow/Workflow8/processor/SeparateVariants/').

Table 11. Prolog instances for each SciCumulus, e-Science Central, VisTrails, and Taverna ProvOne construct of a diagnosis analysis workflow (cont.)

#	Prolog Instances for SciCumulus	Prolog Instances for e-Science Central	Prolog Instances for VisTrails	Prolog Instances for Taverna
6	-	wasGeneratedBy('tr-30115-36ED8706-6AE1-83E0-97D8-EB10CD8080FC-imported-data','esc:svi-esc/invocation/30115/block/36ED8706-6AE1-83E0-97D8-EB10CD8080FC').	-	wasDerivedFrom('http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/d61a950d-edab-4b8e-b0fa-b3a72191bd58','http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/fbca0988-9237-4e3a-abf3-d78013741817').
7	-		-	-
8	used(ex1s,d28s).	used('esc:svi-esc/invocation/30115/block/502FD710-7C1A-05CD-2BFC-89CD28547DAB','tr-30115-4CE28CBA-9234-3AA8-1564-36A10B955350-filtered-data').	used(ew11v,d81v).	used('http://ns.taverna.org.uk/2011/run/76676230-4711-4a10-b7bc-9a0d093dc526/process/cff62007-0684-410f-a9a3-7279c67c4fa1/','http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/dea13e4c-b8ea-4d5e-8450-311b90842ffe').
9	wasGeneratedBy(d38s,ex1s).	wasGeneratedBy('tr-30115-36ED8706-6AE1-83E0-97D8-EB10CD8080FC-imported-data','esc:svi-esc/invocation/30115/block/36ED8706-6AE1-83E0-97D8-EB10CD8080FC').	-	wasGeneratedBy('http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/12c1dee8-292f-45a9-b6e6-e4743ff9b783','http://ns.taverna.org.uk/2011/run/76676230-4711-4a10-b7bc-9a0d093dc526/process/7b48a5f5-5555-44e3-b246-6ad005f949ca/').
10	wasAssociatedWith(ew8s,u1s).	wasAssociatedWith('esc:svi-esc/invocation/30115','esc:svi-esc/engine/IP:10.2.0.5').	wasAssociatedWith(ew11v,u11v).	wasAssociatedWith('http://ns.taverna.org.uk/2011/run/76676230-4711-4a10-b7bc-9a0d093dc526/process/2b2e2efb-d872-4d53-ba8d-603a0f651543/','taverna-engine').

delimitated by brackets. The general form is:

prov_element(element_id, [prop(), prop()…])

As an example, the fact *entity(w1s,[prop(prov:type,['prov:plan', 'provone:workflow']), prop(prov:label,'sciphy')])* represents an entity with identifier *w1s* and two properties: *prov:type* and *prov:label*. This entity represents a workflow named *sciphy*. All entities and activities are linked to the PROV and ProvONE models by using the prefix "*prov:*" and "*provone:*", respectively. The type of entities and the attributes described in the PROV and ProvONE models are placed after the entities prefix, as can be seen in the last example. Furthermore, entity identifiers were modified to make them unique in the global schema and facts were created to identify the provenance graphs and relate them to their workflows. The provenance graph facts follow the pattern:

dataSet(provenance_graph_id, provenance_graph_name)

Relationships use the identifiers of each ProvONE element. The relationships written in Prolog use the same structure of the PROV and ProvONE models. First, they have the relationship name and then the identifiers of each involved entity, activity, or agent. Those are placed between parenthesis and separated by comma. The general form for this element is:

relationship_name(element_id1, element_id2)

As an example, the fact *wasGeneratedBy(d38s, ex1s)* represents a relationship between the data *d38s* and the execution *ex1s*. Hence, the fact *wasGeneratedBy(d38s, ex1s)* can be read in this way: "the data *d38s* was generated by the execution *ex1s*".

The semantic mapping described in Section 3.2.3 is also defined in Prolog by adding facts that follow the form:

sameAs(label,[],[]) and *equivalentTo(label, [],[])*

These facts represent the semantic relationships between data and activities of heterogeneous provenance graphs. The element *label* of *sameAs* and *equivalentTo* facts

corresponds to the action or abstraction of interrelated data or activities inserted between brackets. As an example, we have the fact *sameAs("input_sequences", [d28s], [d51v])*, where we assume the data *d28s* is equal to the data *d51v*. Another example is *equivalentTo ("alignment", [ex2s], [ex81v])*, where *ex2s* and *ex81v* are considered equivalent executions.

Regarding completeness, note that the e-Science Central provenance graph (rows 6 of Table 6 and 15 of Table 7) does not hold information about the agent, while the SciCumulus, Taverna, and VisTrails provenance graph do not store the program versions (row 12 of Table 6). Besides that, VisTrails does not capture information about the relationships *wasDerivedFrom* and *wasGeneratedBy*.

3.4 PROVENANCE INTEGRATION ARCHITECTURE

This section presents the proposed Provenance Integration Architecture**.** Converting from SciCumulus, VisTrails, Taverna, and e-Science Central proprietary provenance to ProvONE requires the implementation of specialized adapters, or *cartridges* in the proposed architecture, one for each system. Provenance obtained from these cartridges is stored in a unified knowledge base as Prolog facts, as previously discussed. The cartridges may be implemented in any language, but in the version presented in this thesis they are implemented in Java using the mapping of ProvONE, SciCumulus, VisTrails, Taverna, and e-Science Central provenance models presented in Table 6. The implementation for each of the cartridges is available at GitHub (https://github.com/dew-uff/integrated-provenance-analysis).

Using the knowledge base, various teams may access provenance and work collaboratively on the provenance analysis task. They can use pre-defined logical rules to query the provenance database, and thus get more information about similar experiments. Figure 15 gives an overview of the provenance gathering, conversion, integration and query processes.

The SciCumulus cartridge gets *p-prov* and *r-prov* from the relational database and converts them to Prolog by using the mappings presented in Table 6. The e-Science Central cartridge fetches *r-prov* from the graph database and extracts *p-prov* from JSON files. The cartridge developed for VisTrails reads its provenance repository in MySQL, stored as relational tables, and translate them to facts. Finally, Taverna's cartridge reads provenance data that is represented as RDF files (textually represented as Turtle documents) and structured as research objects, and converts them to Prolog.

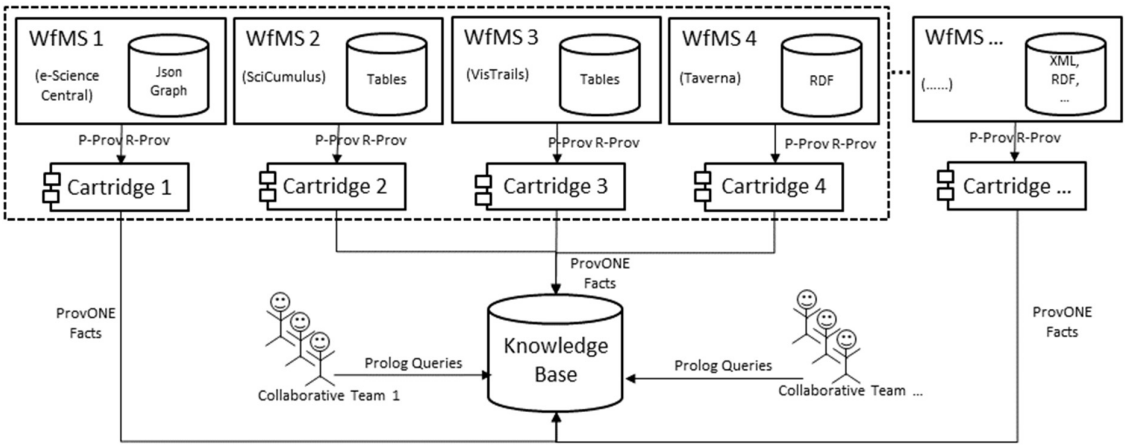


Figure 15. Provenance integration architecture

Clearly, extending the approach to integrating other provenance sources requires new cartridges to be developed. This effort is similar to database integration efforts that are well known in the literature (BATINI *et al.*, 1986). In our approach, implementing new cartridges does not require the development of ontologies, as proposed by other approaches (FILETO *et al.*, 2003; SAHOO *et al.*, 2009). Usually, the definition of entities and relationships of an ontology is not trivial and require a consensus among experts of a given domain.

3.5 QUERYING THE INTEGRATED TRACES

Using the proposed architecture, we are now able to express queries that span different types of provenance and different types of graphs. Queries performed on the integrated schema are expressed in Prolog as rules. To illustrate, we have implemented the queries listed in Table 5, which exemplify the intersection classes of Figure 9. Specifically, the *dataTrace* and *dataFlow* rules implement queries *Q5* and *Q12*. Query *Q5* covers class *C2* and retrieves *r-prov* from either provenance graph *PG1* or provenance graph *PG2*, while *Q12* covers class C6 and retrieves *p-prov* and *r-prov* from both PG1 and PG2. Although these queries are quite similar, *Q12* retrieves the trace of data for all executions, while *Q5* considers only one of the workflow systems. The rules were designed for retrieving all data that influenced the generation of a particular data product. The query result is a historical data trace that shows which input files influenced the generation of a given output.

Table 12 and Table 13 show the query calls (and their associated results) with the parameters used to query the data trace for a specific result generated by SciCumulus, e-Science Central, VisTrails, and Taverna running the phylogenetic analysis workflows. On the other

hand, Table 14, and Table 15, show the queries and results generated by the same WfMSs running the diagnosis analysis workflows.

```
dataTrace(PGName, WkfName, WExName, OutputId, InputId) :-
      distinct(trace(PGName, WkfName, WExName, OutputId, InputId)).

trace(PGName, WkfName, WExName, OutputId, InputId) :-
      dataSet(PGId, PGName),
      hasDataSet(WkfId, PGId),
      activity(WExId,[prop(prov:type,['provone:execution']),
               prop(prov:label, WExName),_,_,_]),
      entity(WkfId,[prop(prov:type,ETypes),prop(prov:label,WkfName)]),
      member('provone:workflow', ETypes),
      hadPlan(WExId,WkfId),
      wasPartOf(ExId, WExId),
      wasGeneratedBy(OutputId, ExId),
      dataFlow(OutputId, InputId).

dataFlow(Output, Input) :-
      wasDerivedFrom(Output, Input).

dataFlow(Output, Input) :-
      wasDerivedFrom(Output, X),
      dataFlow(X, Input).
```

Query Q5 (Table 12 and Table 14) retrieves the input files that influenced the generation of a specific output file that belongs to a particular workflow execution, workflow specification (phylogenetic and diagnosis analysis workflows), and WfMS informed in the query statement as input parameters. To perform comparative analysis between the different provenance graphs, a simple query (*i.e., sameAs(Label,[X],[Y]) and equivalentTo(Label,[X],[Y])*) could also be posed to obtain the same or equivalent output files identifiers. As VisTrails provenance graph does not hold information about the dataflow (*wasDerivedFrom* relationship), there is no answer for the data trace queries.

Query Q12 (Table 13 and Table 15) returns the input files that influenced the generation of all output files that belongs to a particular workflow execution, workflow specification, and WfMS. In this case, the query has no values for all parameters. Hence, the query returns the dataflow for all workflow executions, specifications, and provenance systems stored in the knowledge base.

Table 12. Prolog queries (Q5) and results for the phylogenetic workflows

Workflow	WfMS	Prolog Query and Results

SciPhy	**SciCumulus**	`dataTrace('SciCumulus', 'sciphy', 'sciphy-1', dc19s, InputId).`
		
		`InputId = dc6s; InputId = dc12s; InputId = dc13s; InputId = dc14s; InputId = dc1s;`

ML	**e-Science Central**	`dataTrace('e-Science Central', 'ML Pipeline', 'Testing ML Pipeline', dc51559, InputId).` `InputId = dc2012.`
SciEvol	**VisTrails**	`dataTrace('VisTrails', 'SciEvol', 'SciEvol*', Data, InputId).` -
Phylo	**Taverna**	`dataTrace('Taverna', 'Workflow1', 'Workflow run of Workflow1@en', 'http://ns.taverna.org.uk/2011/data/a7433bae-822a-43fa-896c-7073b13da84b/error/00aa1dfd-48a9-4fb4-878e-8ae728e6bf9d/0', InputId).` `InputId = 'http://ns.taverna.org.uk/2011/data/a7433bae-822a-43fa-896c-7073b13da84b/error/a313534c-6c4e-40d1-94d4-0cad73bb3e0e/0'.`

Table 13. Prolog queries (Q12) and results for the phylogenetic workflows

Workflow	WfMS	Prolog Query and Results
All	All	`dataTrace(Dataset, Workflow, WorkflowExecution, OuputId, InputId),nl.`

		`Dataset = 'e-Science Central',`
		`Workflow = 'ML Pipeline',`
		`WorkflowExecution = 'Testing ML Pipeline',`
		`OuputId = dc51559,`
		`InputId = dc2012;`
		
		`Dataset = 'Taverna',`
		`Workflow = 'Workflow1',`
		`WorkflowExecution = 'Workflow run of Workflow1@en',`
		`OuputId = 'http://ns.taverna.org.uk/2011/data/…/error/00aa1dfd-48a9-4fb4-878e-8ae728e6bf9d/0',`
		`InputId = 'http://ns.taverna.org.uk/2011/data/a7433bae-822a-43fa-896c-7073b13da84b/error/a313534c-6c4e-40d1-94d4-0cad73bb3e0e/0';`
		
		`Dataset = 'SciCumulus',`
		`Workflow = sciphy,`
		`WorkflowExecution = 'sciphy-1',`
		`OuputId = dc6s,`
		`InputId = dc1s;`
		
		`Dataset = 'SciCumulus',`
		`Workflow = sciphy,`
		`WorkflowExecution = 'sciphy-1',`
		`OuputId = dc12s,`
		`InputId = dc1s;`

Table 14. Prolog queries (Q5) and results for the diagnosis workflows on SciCumulus, e-Science Central, VisTrails, and Taverna provenance graphs

Workflow	WfMS	Prolog Query and Results

PatientDiag	**SciCumulus**	`dataTrace('SciCumulus', patientdiag, 'patient diag-exec', d128s, InputId).`
		`InputId = d108s; InputId = d118s; InputId = d78s;`
		`InputId = d88s; InputId = d98s; InputId = d38s;`
		`InputId = d48s; InputId = d58s; InputId = d68s;`
		`InputId = d18s; InputId = d28s;`

SVI	**e-Science Central**	`dataTrace('eSC', 'SVI', 'SVI_Exec', 'tr-30115-36ED8706-6AE1-83E0-97D8-EB10CD8080FC-imported-data', InputId).`
		`InputId = 'esc:svi-esc/document/2175/2176';`
		`InputId = 'esc:svi-esc/invocation/30115/block/36ED8706-6AE1-83E0-97D8-EB10CD8080FC/properties';`

Pathogenesis	**VisTrails**	`dataTrace('VisTrails', 'Pathogenesis', 'Pathogenesis*', d51v, InputId).` -
GeneClass	**Taverna**	`dataTrace('Taverna', 'Workflow8', WExName, 'http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/9d1a8823-daff-41d0-ac0f-c6da25bbeec4', InputId).` `InputId = 'http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/12c1dee8-292f-45a9-b6e6-e4743ff9b783';`

These query instances hide the complexity of the Prolog rules and become suitable for non-experts in the Prolog language. Note that the user may bind none, one, or multiple parameter values. For example, if one specifies no parameter values (*i.e.,* Q12), the query will return the graph name, workflow name, execution name, along with the input and output data for all datasets. This makes Prolog queries a flexible resource to retrieve provenance according to specific requirements.

3.6 CONCLUDING REMARKS

The integration of heterogeneous provenance graphs can be a powerful tool for provenance analytics. In particular, it can provide considerable advantages for research teams that work collaboratively on similar experiments. In this chapter, we have presented an approach that enables integrating and querying provenance data from similar workflows designed and implemented in different systems with different specifications.

We propose a Provenance Integration Architecture that uses an integration model (ProvONE) that includes both *p-prov* and *r-prov* and create cartridges that convert different

Table 15. Prolog queries (Q12) and results for the diagnosis workflows on SciCumulus, e-Science Central, VisTrails, and Taverna provenance graphs

Workflow	WfMS	Prolog Query and Results
All	All	dataTrace(Dataset, Workflow, WorkflowExecution, OuputId, InputId),nl. Dataset = 'SciCumulus', Workflow = patientdiag, WorkflowExecution = 'patientdiag-exec', OuputId = d148s, InputId = d18s; Dataset = 'SciCumulus', Workflow = patientdiag, WorkflowExecution = 'patientdiag-exec', OuputId = d148s, InputId = d28s; Dataset = eSC, Workflow = 'SVI', WorkflowExecution = 'SVI_Exec', OuputId = 'tr-30115-36ED8706-6AE1-83E0-97D8-EB10CD8080FC-imported-data', InputId = 'esc:svi-esc/document/2175/2176'; Dataset = eSC, Workflow = 'SVI', WorkflowExecution = 'SVI_Exec', OuputId = 'tr-30115-36ED8706-6AE1-83E0-97D8-EB10CD8080FC-imported-data', InputId = 'esc:svi-esc/invocation/30115/block/36ED8706-6AE1-83E0-97D8-EB10CD8080FC/properties'; Dataset = 'Taverna', Workflow = 'Workflow8', WorkflowExecution = 'Workflow run of Workflow8@en', OuputId = 'http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/d61a950d-edab-4b8e-b0fa-b3a72191bd58', InputId = 'http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/fbca0988-9237-4e3a-abf3-d78013741817'; Dataset = 'Taverna', Workflow = 'Workflow8', WorkflowExecution = 'Workflow run of Workflow8@en', OuputId = 'http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/d61a950d-edab-4b8e-b0fa-b3a72191bd58', InputId = 'http://ns.taverna.org.uk/2011/data/76676230-4711-4a10-b7bc-9a0d093dc526/ref/9d1a8823-daff-41d0-ac0f-c6da25bbeec4';

provenance databases to a global ProvONE schema of Prolog facts. Our approach introduces classes that explore intersection between *p-prov*, *r-prov*, and heterogeneous provenance graphs and presents related queries that run across both provenance graphs and retrieve information

with different contents and levels of detail. Prolog rules were developed for each pre-defined query, taking advantage of inference and unification facilities provided by Prolog.

In the next chapter, we evaluate the effectiveness and efficiency of our approach and the overhead it imposes.

Chapter 4 – EXPERIMENTAL EVALUATION

4.1 INTRODUCTION

As discussed in the Introduction, we performed a survey about provenance integrated analysis, which demonstrates that this is a realistic scenario. However, in such scenario, is the proposed integrated analysis more efficient and effective than analyzing the non-integrated provenance graphs individually? How much overhead the translation to Prolog poses on the repository? To answer these questions, we designed two experiments to assess whether scientists can benefit from the proposed approach to correctly query heterogeneous provenance graphs in shorter time.

4.2 EFFICIENCY AND EFFECTIVENESS EVALUATION

As presented in the previous chapters, our approach uses Prolog to represent and query provenance that is stored in a single knowledge base of facts and rules. Since not everyone has a previous experience with Prolog, we invited computer science students and professors with various levels of knowledge in the Prolog language. Then, the volunteers filled a personal profile form (Appendix A) with their personal information, their Prolog knowledge level and programing experience. Based on this information, we divided the 22 volunteers into 2 groups (named *Group 1* and *Group 2*), aiming at balancing the knowledge level of Prolog across the groups. The vast majority of the volunteers had basic knowledge in Prolog (*i.e.,* they are aware on how to elaborate rules and queries by using unification, conjunction, and disjunction operations).

As most students and professors are not scientific workflows experts, we created 2 simple workflows (named *Curriculum* and *Skills*) that aim at evaluating people's resumes. The workflows are composed by programs that evaluate common resume sections such as education, professional experience, publication, among others. The *Curriculum* workflow was designed and executed in VisTrails while the *Skills* workflow was designed and executed in SciCumulus. The *p-prov* and *r-prov* generated from both workflow systems were converted to Prolog facts using the cartridges we developed for these systems (see Chapter 3). Figure 16 shows the graphical representation of *Curriculum* (a) and *Skills* (b) workflows. In those representations, ellipse elements represent the input and output data and rectangles represent activities. The arrows show the dataflow between activities through their input and output ports (little black squares).

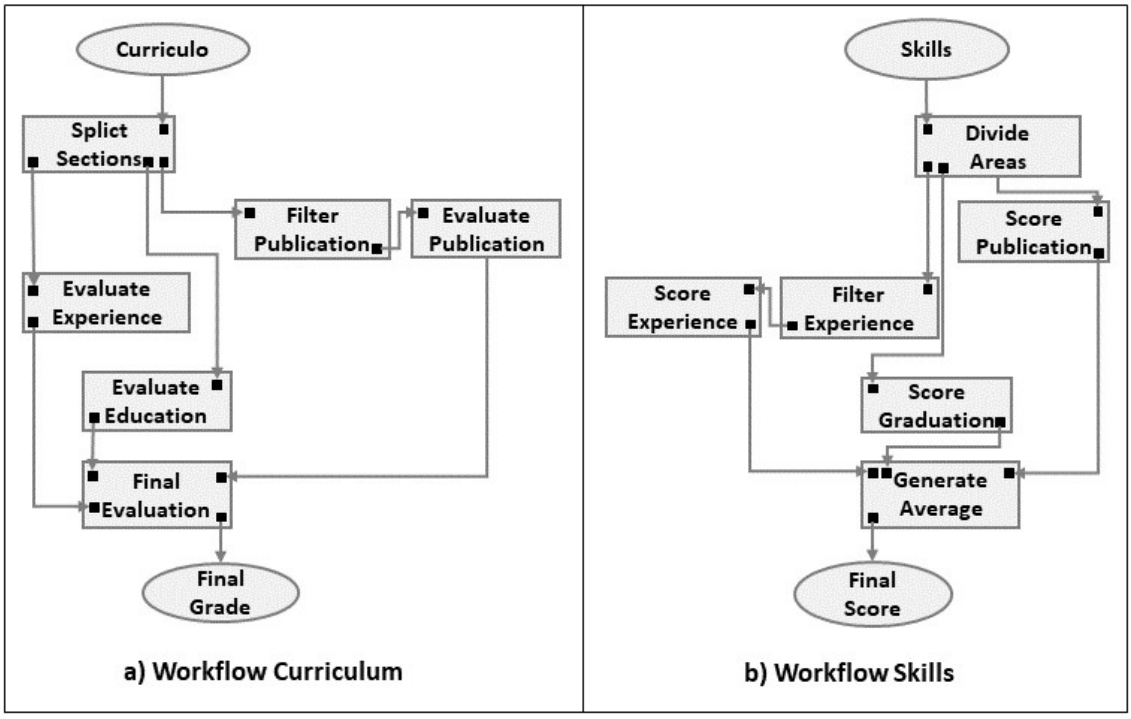


Figure 16. Graphical representation of Curriculum and Skills workflows

Since we wanted to evaluate which method would be better to analyze heterogeneous provenance graphs, we then designed 6 questions that cover all classes that include 2 provenance graphs (C4, C5, and C6), described in our reference classification in Chapter 3. Table 16 lists those questions, their related class, and the complexity score for each question. The complexity score is based on the kind of class. Furthermore, examples of possible Prolog queries for each question and approach are shown in Table 17 and Table 18.

Table 16. Questions and their related classes

#	Question	Class	Complexity Score
1	Which of the workflows executed in less time?	C5	1
2	List the programs of both workflows with their respective executions.	C6	2
3	What was the input data used in the execution of programs "*Final_Evaluation*" and "*Score_Graduation*"?	C6	2
4	Analyze the workflows and list equivalent or identical data and programs.	C4	1
5	What are the input parameters of all programs in both workflows?	C4	1
6	Are the names of users/agents the same in both workflow executions?	C5	1

Table 17. Example of Prolog queries using the *Standalone Approach*

#	Prolog Query
1	entity(WkfId,[prop(prov:type,['prov:plan','provone:workflow']),_]), activity(ExWId,[_,prop(_,ExName),prop(_,StartTime),prop(_, EndTime), _]), hadPlan(ExWId,WkfId), nl.
2	entity(WkfId,[prop(prov:type,['prov:plan','provone:workflow']), prop(prov:label,WkfName)]), entity(PgId,[prop(prov:type,['prov:plan','provone:program']),prop(pro v:label,PgName)]), activity(ExId, [_, prop(_, ExName),_, _, _]), hasSubProgram(WkfId, PgId), hadPlan(ExId,PgId), nl.
3	entity(PgId,[_,prop(_,PgName)]), activity(ExId, [_, prop(_, ExName),_, _, _]), used(ExId, Data), hadPlan(ExId,PgId), entity(PgId,[_,prop(_,'Final_Evaluation ')]), nl. entity(PgId,[_,prop(_,PgName)]), activity(ExId, [_, prop(_, ExName),_, _, _]),used(ExId, Data), hadPlan(ExId,PgId), entity(PgId,[_,prop(_,'Score_Graduation')])), nl.
4	entity(PgId,[prop(prov:type,['prov:plan','provone:program']),prop(pro v:label,PgName)]),nl; entity(DtId,[prop(prov:type,['provone:data',_]),prop(prov:label,DtNam e),prop(prov:value,DtValue)]),nl.
5	entity(WkfId,[prop(prov:type,['prov:plan','provone:workflow']), prop(prov:label,WkfName)]), hasSubProgram(WkfId, PgId), entity(PgId,[prop(prov:type,['prov:plan','provone:program']),prop(pro v:label,PgName)]), entity(Param,[prop(prov:type,['provone:data',_]),prop(prov:label,Para mName),prop(prov:value,ParamValue)]), hasDefaultParam(Port,Param), hasInPort(PgId, Port), nl.
6	agent(UserId, [_, prop(_,User)]),wasAssociatedWith(WkfEx,UserId), nl.

We include new facts *sameAs(label,[],[])* and *equivalentTo(label, [],[])* to describe the semantic relationships between data and activities of *Curriculum* and *Skills* workflows. The element *label* of *sameAs* and *equivalentTo* facts corresponds to the action or abstraction of interrelated data or activities. The related set of data and activities are placed inside of each bracket after the *label*. In this way, a single query can go across two or more provenance graphs to bring together information related to equal or similar data and activities. Table 19 lists data and activities of *Curriculum* and *Skills* workflow that have compatible content or behavior.

Table 18. Example of Prolog queries questions using *Integrated Approach*

#	Prolog Query
1	activity(ExWId, [_, prop(_, ExName),prop(_, StartTime), prop(_, EndTime), _]), hadPlan(ExWId, _), hadDataSet(ExWId,_),nl.
2	entity(WkfId,[prop(prov:type,['prov:plan','provone:workflow']), prop(prov:label,WkfName)]), entity(PgId,[prop(prov:type,['prov:plan','provone:program']),prop(prov:label,PgName)]), activity(ExId, [_, prop(_, ExName),_, _, _]), hasSubProgram(WkfId, PgId),hadPlan(ExId,PgId), nl.
3	entity(PgId,[_,prop(_,PgName)]), activity(ExId, [_, prop(_, ExName),_, _, _]), used(ExId, Data), hadPlan(ExId,PgId), (entity(PgId,[_,prop(_,'Avaliacao_Final')]); entity(PgId,[_,prop(_,'pontuar_titulacao')])), nl.
4	sameAs(Tag,X) ; equivalentTo(Tag,X), nl.
5	entity(WkfId,[prop(prov:type,['prov:plan','provone:workflow']), prop(prov:label,WkfName)]), hasSubProgram(WkfId, PgId), entity(PgId,[prop(prov:type,['prov:plan','provone:program']),prop(prov:label,PgName)]), entity(Param,[prop(prov:type,['provone:data',_]),prop(prov:label,ParamName),prop(prov:value,ParamValue)]), hasDefaultParam(Port,Param), hasInPort(PgId, Port), nl.
6	agent(UserId, [_, prop(_,User)]), wasAssociatedWith(WkfEx,UserId), nl.

Table 19. Semantic relationships between data and activities of two resume workflows

Curriculum	Skills	Description
Curriculo	Skills	Input file containing the Resume
Splict_Sections	Divide_Areas	Fragments of the resume into sections
Filter_Publication, Evaluate_Publication	Score_Publication	Evaluates the academic publications, providing a score as output
Evaluate_Experience	Filter_Experience, Score_Experience	Evaluates the professional experience, producing a score as output
Evaluate_Education	Score_Graduation	Evaluates academic degrees (education), producing a score as output
Final_Evaluation	Generate_Average	Averages all previous scores
Final_Grade	Final_Score	Generates a file containing the final score achieved

The experiment took place in computing science labs of three different institutions (Universidade Federal de Juiz de Fora - UFJF[9], Universidade Federal Fluminense - UFF[10], and Instituto Federal do Sudeste de Minas Gerais - IFSEMG[11]). Before the volunteers started to answer the proposed questions, they received an explanation about both workflows (*Curriculum* and *Skills*), provenance terms, and the structure of the knowledge base (provenance datasets in Prolog). After that, they were advised to follow this step-by-step: (i) fill their names in a form; (ii) read one of the questions; (iii) take note of the start time; (iv) verify the datasets and workflow mappings; (v) write and execute the Prolog queries; (vi) copy and paste the query to the form; (vii) copy and paste the query result to the form; (viii) answer the proposed question, and (ix) finally, take note of the end time. After that, go back to step (ii) until there are no more questions to be answered.

The experiment questions were divided in two stages (*Stage 1* and *Stage 2*) and answered by using two different approaches (*Approach A* and *Approach B*). *Approach A* represents the original non-integrated approach. By using *Approach A,* the volunteers have to query two different provenance databases (SciCumulus and VisTrails) stored in two different knowledge bases (Prolog files). On the other hand, *Approach B* represents the integrated approach we propose in this thesis. By using the *Approach B* volunteers are able to query both provenance databases (SciCumulus and VisTrails), which are stored in a single knowledge base enriched with semantic information. From now on, we will refer to them as *Standalone Approach* and *Integrated Approach*, respectively.

Following the Latin square technique (COCHRAN; COX, 1950) to ensure the equality of the experiment, each group of volunteers (*Group 1* and *Group 2*) answered 3 questions in each stage by using *Standalone Approach* and *Integrated Approach* in an inverse order. First, in the *Stage 1*, *Group 1* answered the first three out of six proposed questions using *Standalone Approach*. Then, in *Stage* 2, they answered the last three questions using *Integrated Approach*. *Group 2* did the opposite. They answered the same three first questions using *Integrated Approach* and the last three questions using *Standalone Approach*.

4.2.1 RESULTS AND EVALUATION PER QUESTION

The experiment was designed to evaluate the efficiency and effectiveness of the proposed integration architecture (*Integrated Approach*) when compared to the original stand-alone

[9] www.ufjf.br
[10] www.ic.uff.br
[11] www.riopomba.ifsudestemg.edu.br

approaches (*Standalone Approach*). First of all, we grade the volunteer`s answers. Each correct answer received the score 1, wrong answers received 0, and whenever the answer was somehow correct it received 0.5. Then, we compute the time spent to answer each question. Hence, in our evaluation, the efficiency is defined as the ratio of the question's score by the time consumed to answer it, while effectiveness is simply the achieved score. It is important to highlight that it is possible that one can be efficient by giving half-correct answers very fast. However, it is worth noting that one does not get any efficiency gains by answering wrong (score = 0), so efficiency is a measure of actual effort.

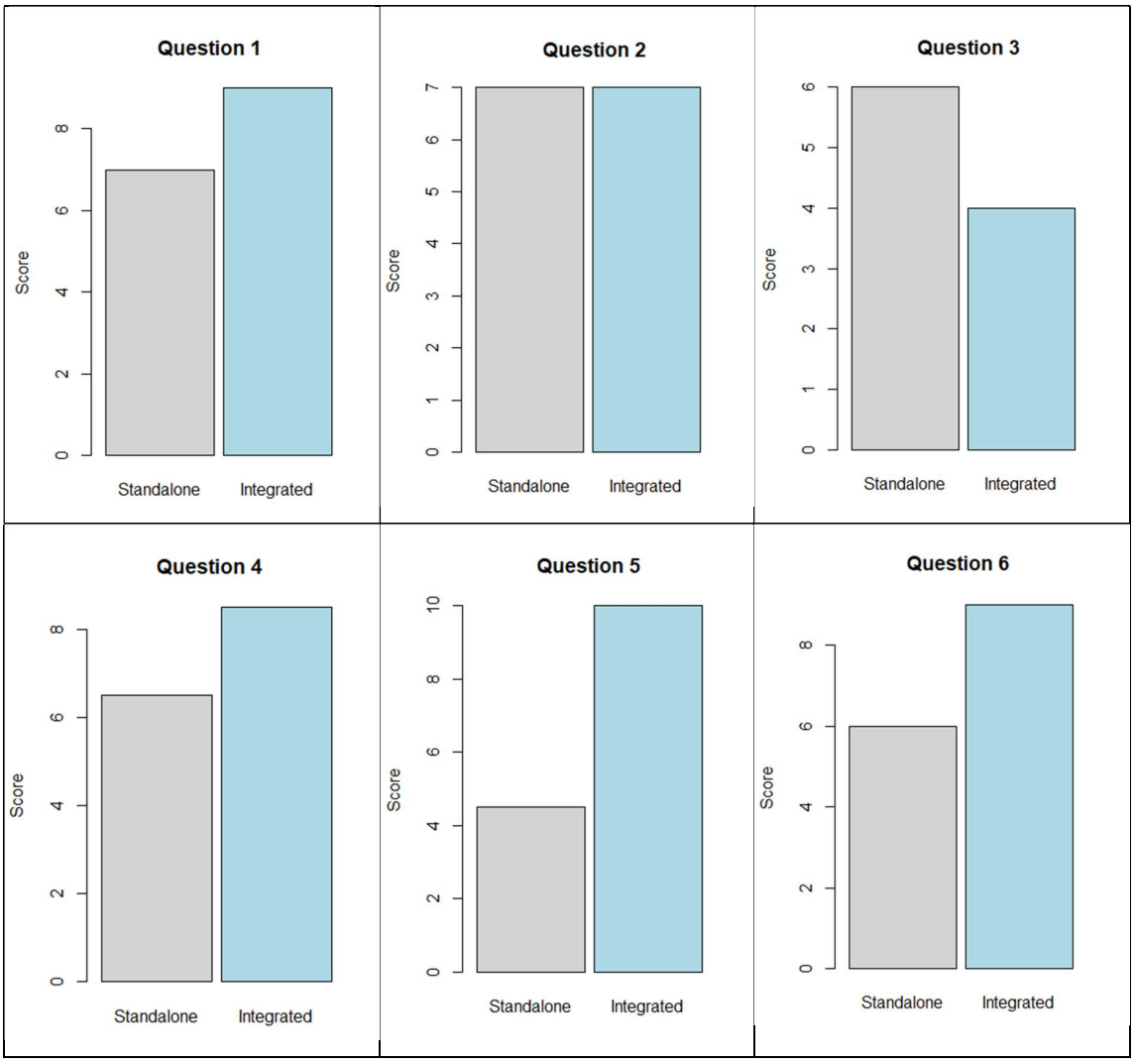


Figure 17. Results for the effectiveness variable of questions/answers 1 to 6

We analyze each question/answer separately and compare their results for the *Standalone Approach* and *Integrated Approach*. The results for the effectiveness, time spent and efficiency variables for each question/answer can be seen in Figure 17, Figure 18, and Figure 19, respectively.

Considering the effectiveness variable (Figure 17), the *Standalone Approach* and the *Integrated Approach* presented the same result for question 2. On the other hand, the *Integrated Approach* presented more correct answers in questions 1, 4, 5, and 6 while the *Standalone Approach* presented more correct answers in question 3. On the other hand, the time spent (Figure 18) was smaller by using the *Standalone Approach* in questions 2, 5, and 6 while questions 1, 3, and 4 were answered faster by using the *Integrated Approach*. Finally, the *Standalone Approach* was more efficient (Figure 19) in question 3 than the *Integrated Approach*. However, the *Integrated Approach* presented better efficiency in questions 1, 2, 4, 5, and 6.

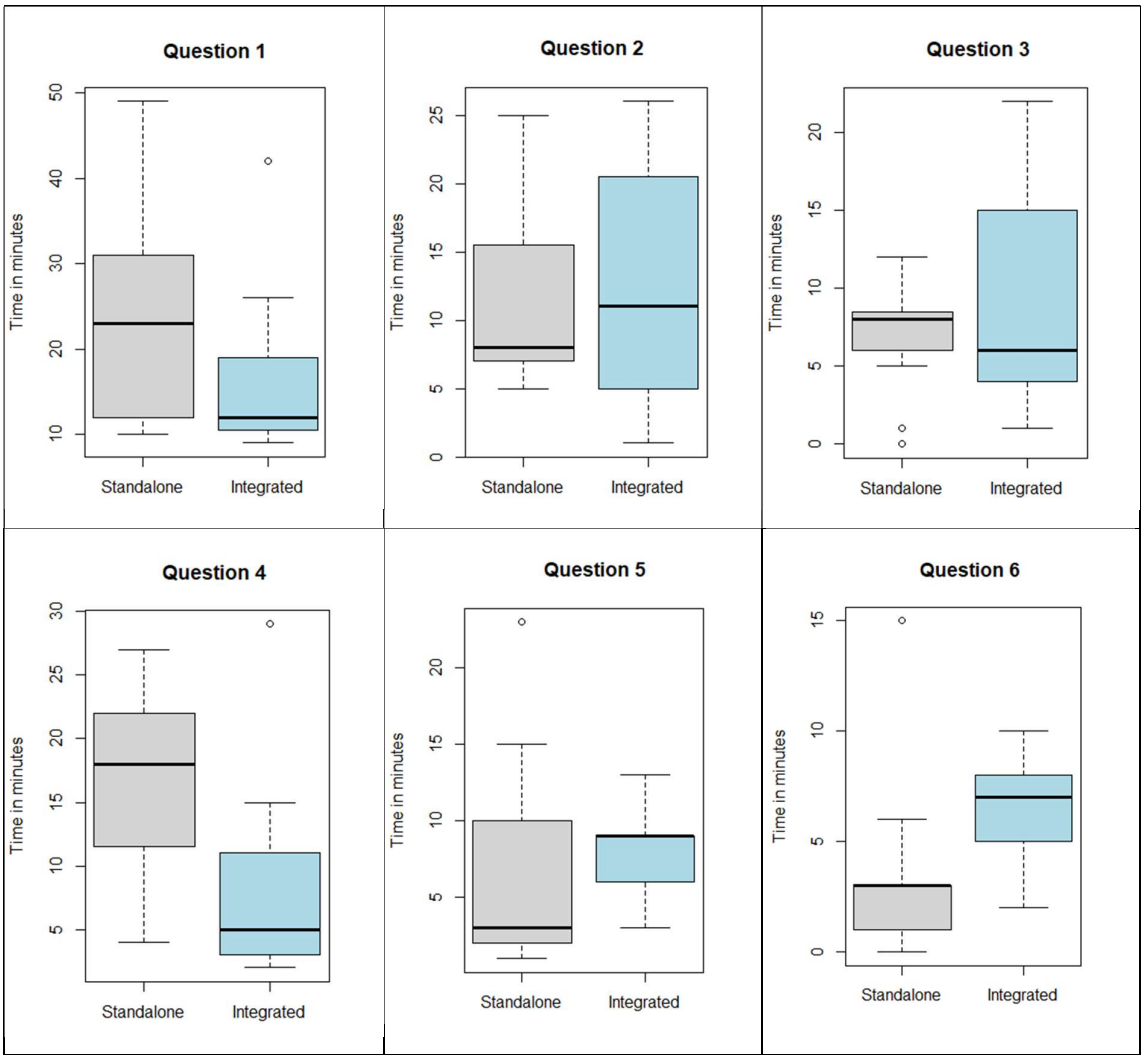


Figure 18. Results for the time spent in questions/answers 1 to 6

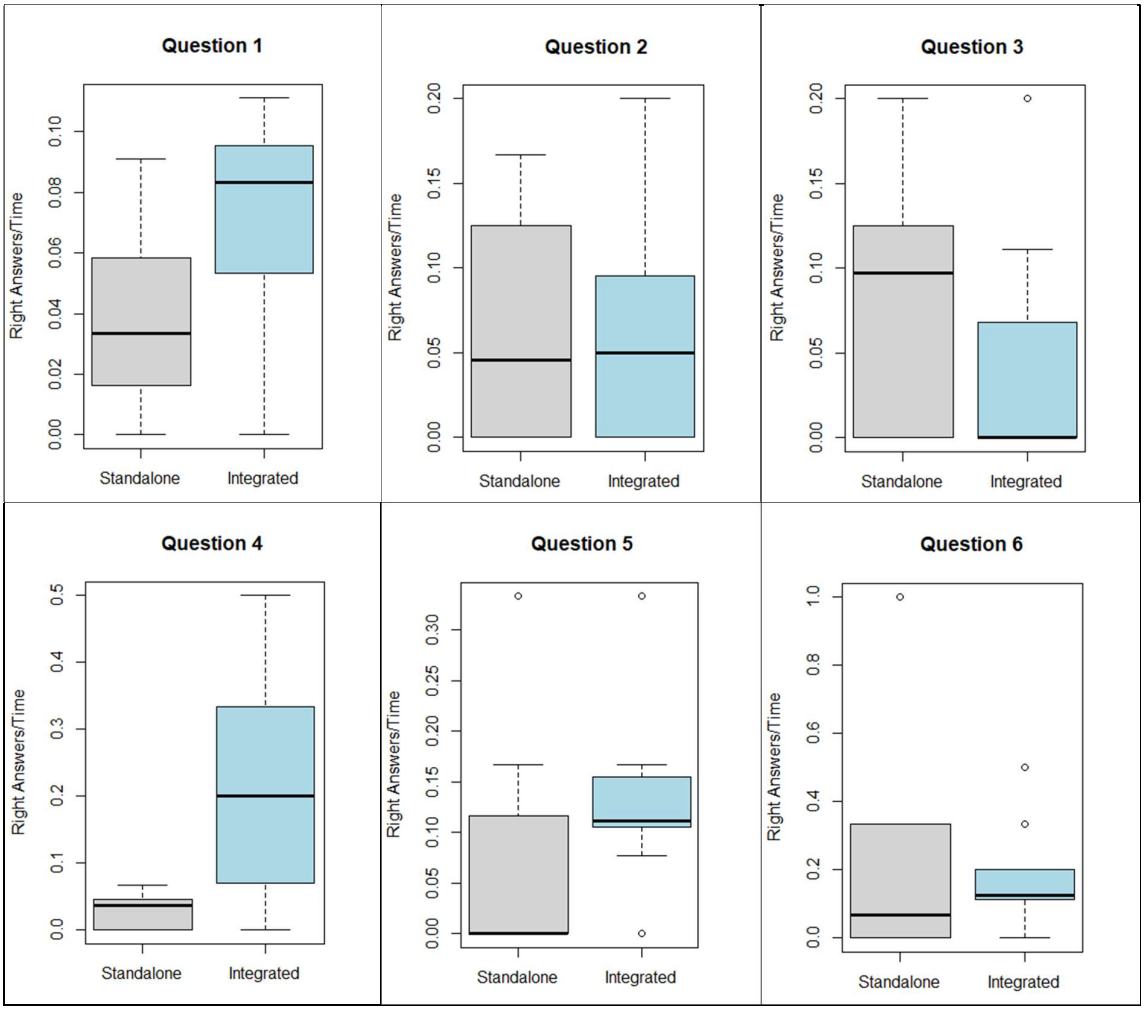


Figure 19. Results for the efficiency variable of questions/answers 1 to 6

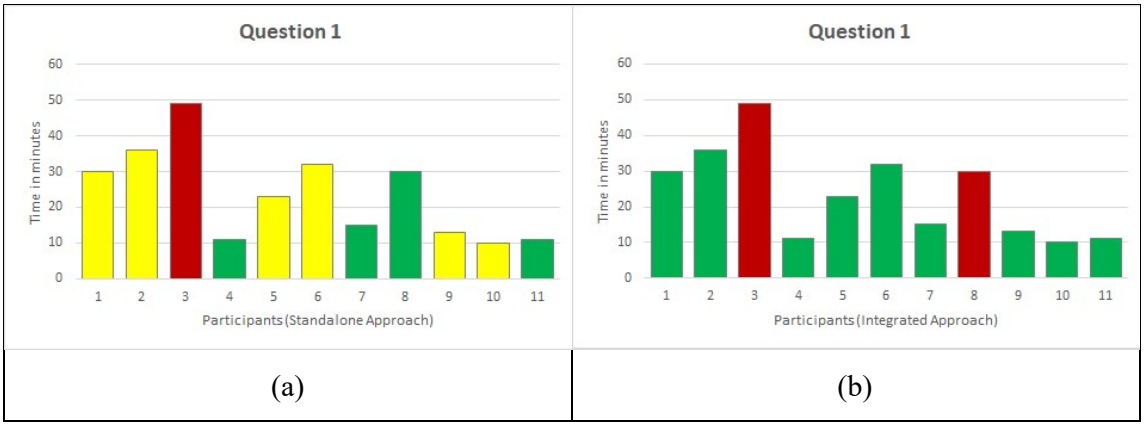
The Shapiro test showed a non-normal distribution for data *per* question. This way, we applied the Wilcoxon test considering the p-value < 0.5 (95% of confidence) to check the statistical significance of such results. As the number of answers per question is quite small (11 for each approach), the Wilcoxon test results could be compromised. This way, we also use the *Cliff Delta* (CLIFF, 1996) to evaluate the effect size of the results (the strength of their difference). *Cliff Delta* is a non-parametric test aiming to quantify the difference between two groups. Table 20 shows the *p-value* and *Cliff Delta* results for the effectiveness, time spent, and efficiency variables of each question/answer.

According to the results, only the effectiveness of question 5, the time spent of questions 4 and 6, and the efficiency of question 4 has statistical significance (*p-value* < 0.05). On the other hand, the *Cliff Delta* test reveals a medium to a large effect size for the effectiveness in

question 1, 3, 4, 5, and 6. The same occurred for the time spent in questions 1, 4, 5, and 6, and for the efficiency in questions 1, 3, 4, and 5.

Note that questions 2 and 3 are harder (they touch both types of provenance and are of class C6). We have listed the complexity level of each query previously in Table 16. For example, questions in the C6 class, which involves *p-prov* and *r-prov*, are classified as hard questions (score 2). On the other hand, questions in class C4, which involves just *p-prov*, and questions in class C5, which involves only *r-prov*, are classified as normal (score 1). Questions 2 and 3 are exactly the questions where the *Standalone Approach* was slightly better than *Integrated Approach* according to Figure 17, Figure 18, and Figure 19. However, the difference in this case is not statistically significant.

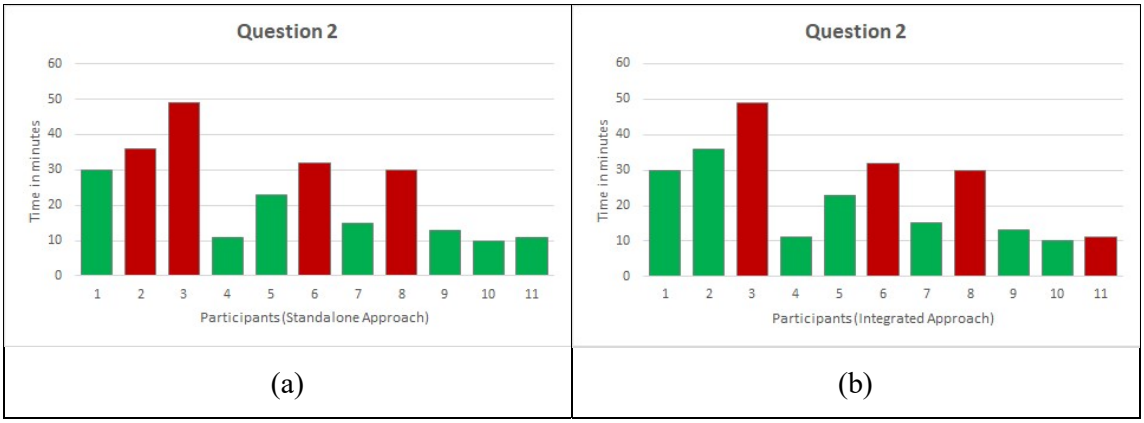
Although Questions 5 and 6 are not classified as hard, the total time of the *Integrated Approach* for these questions was higher than that of the *Standalone Approach*, and the difference in this case is statistically significant. To try to understand what happened in questions 5 and 6, and in the remaining questions more precisely, Figure 20, Figure 21, Figure 22, Figure 23, Figure 24, and Figure 25 present the histogram of the time spent by each participant to answer those queries. Each histogram shows the time consumed by each participant to answer the proposed questions. To clarify the understanding about the results, each bar on the histogram was filled with different colors. In this way, for correct answers the bar was filled with green, for wrong answers it was used red, and for half-correct answers it was painted with yellow.



(a) (b)

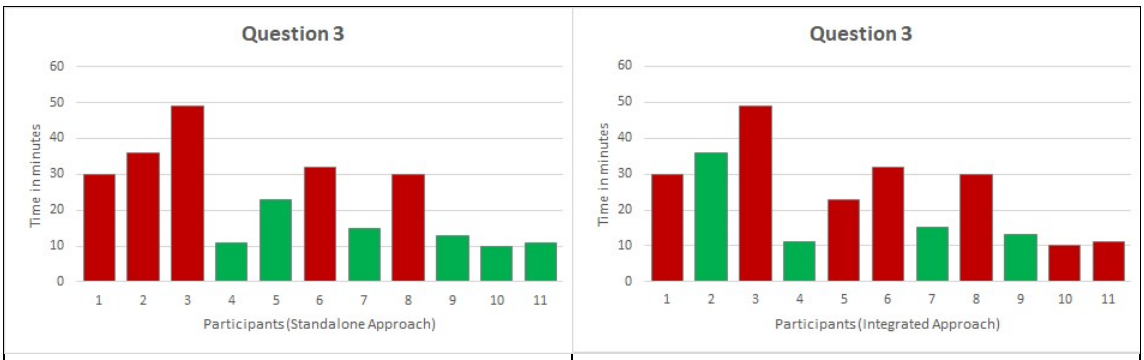
Figure 20. Histograms for the time spent by all participants in question 1 by using the Standalone (a) and Integrated (b) approaches

As can be seen in Figure 20 (a), even the most participants that used Standalone Approach getting half-correct answers, the total time consumed was greater than the Integrated Approach (as shown in Figure 18). On the other hand, most participants got more correct answers by using the Integrated Approach in less time.



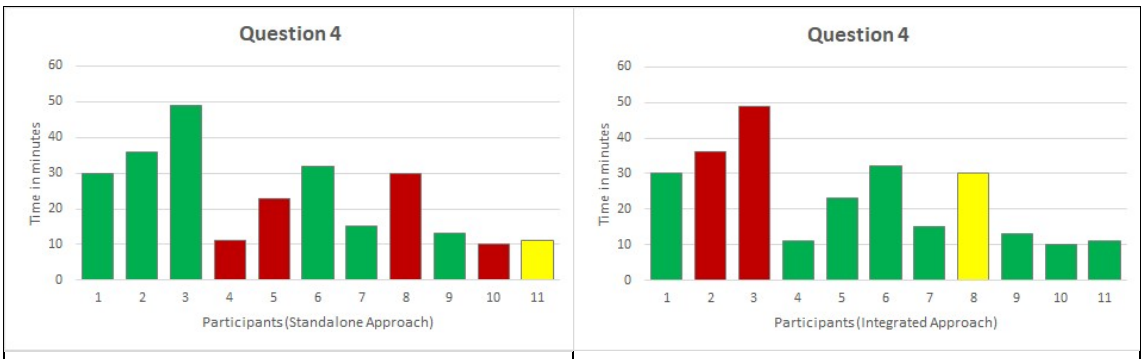
(a) (b)

Figure 21. Histograms for the time spent by all participants in questions 2 by using the Standalone (a) and Integrated (b) approaches



(a) (b)

Figure 22. Histograms for the time spent by all participants in questions 3 by using the Standalone (a) and Integrated (b) approaches



(a) (b)

Figure 23. Histograms for the time spent by all participants in questions 4 by using the Standalone (a) and Integrated approaches

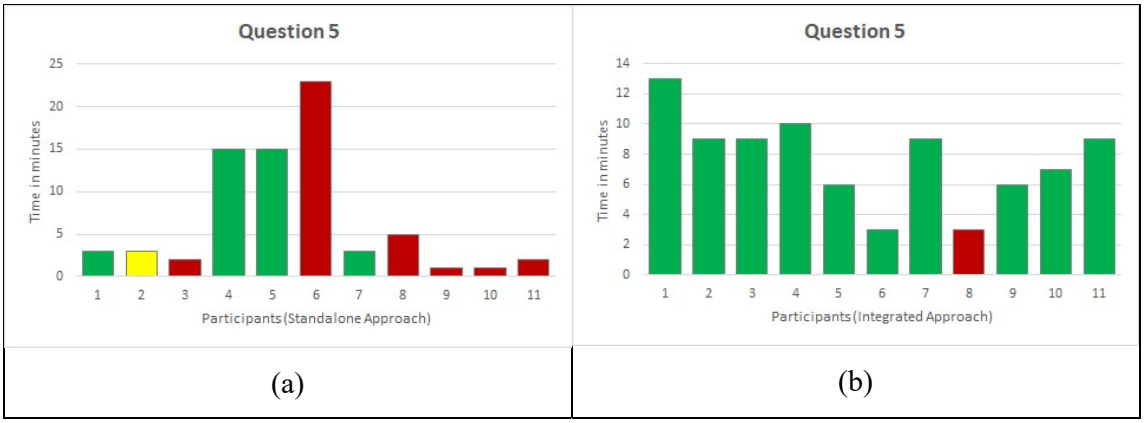


Figure 24. Histograms for the time spent by all participants in questions 5 by using the Standalone (a) and Integrated approaches

As previously described, the Standalone Approach got a better result in the total time spent in questions 5 and 6. However, as can be seen in Figure 24 and Figure 25, most questions answered by the participants that used the Standalone Approach were wrong while most questions answered by the participants that used the Integrated Approach were correct.

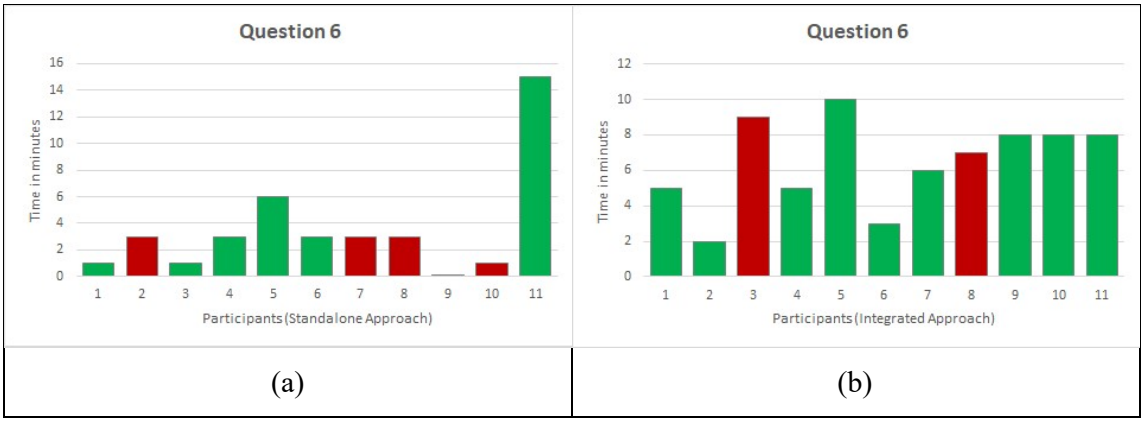


Figure 25. Histograms for the time spent by all participants in questions 6 by using the Standalone (a) and Integrated approaches

From the statistical point of view, a very consistent breakout result was not achieved. Part of it is due the multiple variables and different levels of complexity for each question. This can be seen in the results for *time spent* in question 2 where the volunteers took longer to answer. The same occurred with *efficiency* in question 3 that was classified as a hard question and most of the volunteers could not answer it correctly. In the next section, we show the general results that involve all questions and perform an overall evaluation considering the effectiveness, time spent, and efficiency of the *Standalone* and *Integrated* approaches.

Table 20. Statistical significance measured by *p-value* and *Cliff Delta*

Question	Effectiveness		time spent		Efficiency	
	p-value	*Cliff Delta*	*p-value*	*Cliff Delta*	*p-value*	*Cliff Delta*
1	0.1167	**-0.3553719 (medium)**	0.1471	**0.3719008 (medium)**	0.09942	**-0.4214876 (medium)**
2	1	1.011265e-17 (negligible)	0.9211	-0.03305785 (negligible)	1	-0.008264463 (negligible)
3	0.4245	0.1818182 (small)	0.8175	-0.0661157 (negligible)	0.209	0.3090909 (small)
4	0.3744	-0.1983471 (small)	**0.02121**	**0.5867769 (large)**	**0.01396**	**-0.6198347 (large)**
5	**0.01273**	**-0.5371901 (large)**	0.1446	**-0.3719008 (medium)**	0.09344	**-0.4214876 (medium)**
6	0.1933	-0.2727273 (small)	**0.01385**	**-0.6198347 (large)**	0.7127	-0.09917355 (negligible)

4.2.2 EXPERIMENT RESULTS AND OVERALL EVALUATION

In this section, we describe the big picture of the experimental results. We analyze all questions/answers and compare their results for the *Standalone Approach* and the *Integrated Approach*. Figure 26 shows the general results (all questions/answers) for the effectiveness, time spent, and efficiency, respectively. As presented in Figure 26, the *Integrated Approach* got a higher percentage of right answers (71.97%) when compared to the *Standalone Approach* (56.06%). The total time consumed by the *Integrated Approach* was slightly smaller than that of the *Standalone Approach (676 and 759 minutes, respectively)*. Considering efficiency, the *Integrated Approach* presented better results with a higher median than the *Standalone Approach*.

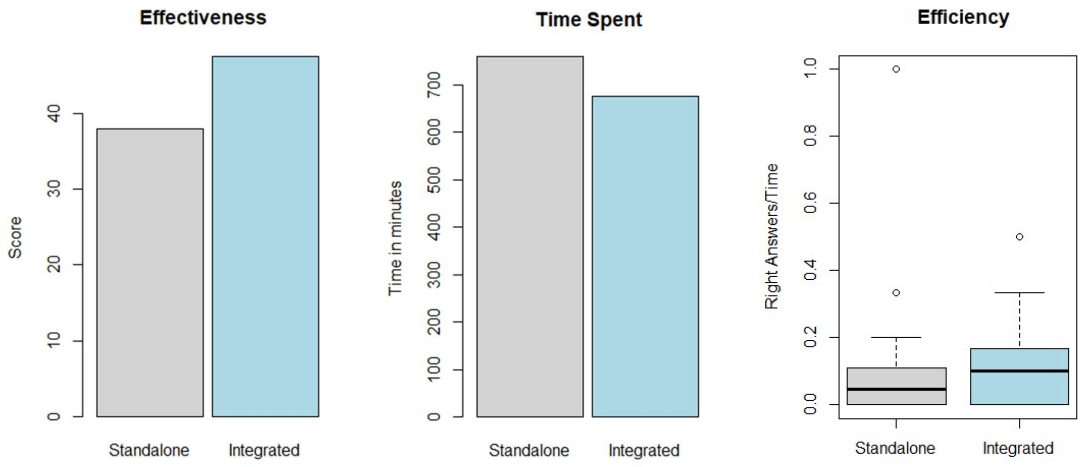


Figure 26. Effectiveness, Time Spent and Efficiency variables

We used the Shapiro Wilk test (SHAPIRO; WILK, 1965) to verify that none of the distributions satisfy the normality assumption. Thus, we used a Wilcoxon test (WILCOXON, 1945)) to confirm if the difference among the results were statistically significant. The results indicate that the results are significantly different for effectiveness (*p-value* = 0.0437) and efficiency (*p-value* = 0.03048) but not for time spent (*p-value* = 0.8948).

4.3 OVERHEAD EVALUATION

This section aims at evaluating possible storage and processing overheads related to the provenance translation to Prolog. As described in Chapter 3 and Chapter 1, our approach allows for analyzing provenance across heterogeneous graphs.

The evaluation was executed in a notebook with 2.0 GHz, 2 cores and 4 logical processors, 8 GB of RAM, 500 GB (SSD) of Hard Disk, and Microsoft Windows 10 operating system.

4.3.1 WORKFLOWS

For this experiment, we have chosen the SciEvol (OCAÑA, KARY A. C. S. *et al.*, 2012b) and Phylo workflows. SciEvol and Phylo are similar scientific workflows implementations that aim at generating phylogenetic trees from DNA, RNA, or amino acid sequences. Phylogenetic trees determine the inferred evolutionary relationships among various biological species. A graphical representation of SciEvol and Phylo is presented in Figure 27 and Figure 28 respectively. SciEvol is composed of four activities. They are responsible for executing the phylogenetic analysis (or gene phylogeny) and are named as: (a) Mafft, (b) ReadSeq, (c) RAxML, and (d) Codeml. The Phylo workflow comprises eight activities named as: (a) FindDir, (b) Clear, (c) Alignment, (d) Convertion, (e) Evolutionary Model, (f) Generate Tree1, (g) Generate Tree2, and (h) GenerateTree3. Each activity is associated with a specific program. *Mafft* and *Alignment* activities may be implemented by MAFFT (KATOH; TOH, 2010), Kalign (LASSMANN *et al.*, 2009), ClustalW (LARKIN *et al.*, 2007), or ProbCons (DO *et al.*, 2005). Each alignment program receives a multi-FASTA file as input, then produces a MSA as output. Multi-FASTA is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes. Besides producing the MSA file, some programs such as ClustalW produce auxiliary files such as a probabilistic matrix (PM) that can be used for reconstructing the sequences. PM is produced by ClustalW, but it is not informed by the program, *i.e.* scientists only discover that this auxiliary file exists if they search on the file directory (implicit provenance). Following, the *ReadSeq* and

Conversion are implemented by ReadSeq (GILBERT, 2002). ReadSeq receives the MSA in different formats (one for each MSA program) and then converts it to the PHYLIP format (FELSENSTEIN, 1989). The *Model* activity is implemented by ModelGenerator (KEANE *et al.*, 2006). There are several different evolutionary models available, and ModelGenerator chooses the best one) to use in the *CodeML, RAxML, GenerateTree1*, *GenerateTree2*, and *GenerateTree3* by RAxML (STAMATAKIS, 2006). In SciEvol and Phylo, we consider pre-chosen evolutionary models (BLOSUM62, CPREV, JTT, WAG, or RtREV) and we obtain several trees for each one of the MSA programs used. The activities *FindDir* and *Clear* use a simple algorithm that searches the workflow's directory and deletes all files generated in a previous workflow execution.

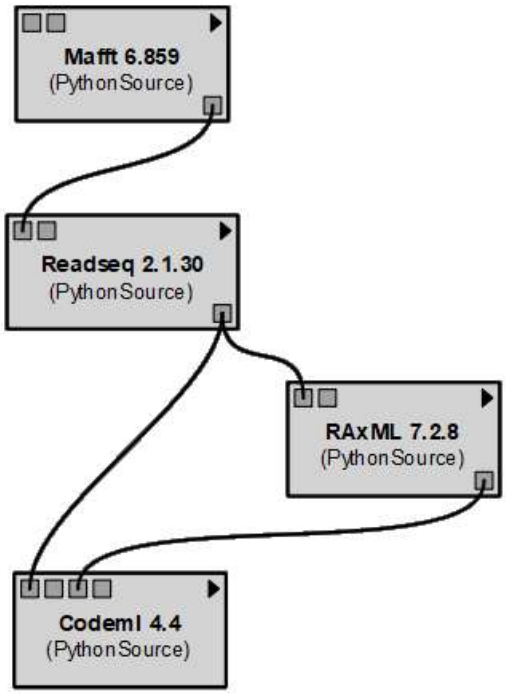


Figure 27. Graphical representation of the SciEvol workflow designed in VisTrails

We used VisTrails (CALLAHAN *et al.*, 2006b) to run SciEvol and Taverna (HULL *et al.*, 2006) to run Phylo in our experiment. VisTrails stores the provenance data in a MySQL relational database while Taverna stores provenance in Turtle files.

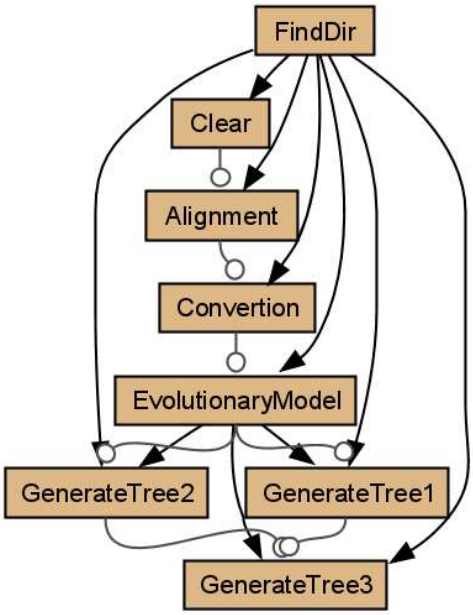


Figure 28. Graphical representation of the Phylo workflow designed in Taverna

After running the workflows, the scientist starts the Cartridges to translate provenance data into ProvONE facts and store it in the Knowledge Base. The translations are needed once each WfMS uses a specific format and structure to represent provenance (*i.e.,* relational, RDF, XML, etc). Then, the scientist informs the semantic mapping between the activities and data of the different workflow implementations. Once all data is stored into the provenance database as Prolog facts, scientists can start submitting Prolog queries that combine provenance from distinct workflow executions.

4.3.2 PROCESSING TIME

In the processing time evaluation, we considered the time spent by the developed mechanisms to translate provenance in Prolog facts. We use the provenance graphs generated by Taverna and VisTrails when they executed the phylogenetic workflows Phylo and SciEvol, respectively. Table 21 shows the results of the processing time evaluation. We use an input file with the size of 3KB in the execution of Phylo workflow and 4KB in the execution of SciEvol workflow.

The *Translation Time* corresponds to the time spent in the translation process, where each provenance repository is translated to Prolog facts and stored in the knowledge base. The *Workflow Execution Time* shows the time consumed by each WfMS to execute the workflows. All measurements are given in seconds. To facilitate the comparison, each dataset was

populated with one workflow specification and one execution. As can be seen in Table 21, the *Translation Time* is very small and the overhead in the translation process is negligible.

Table 21. Time spent in the translation and workflow execution processes

Workflow	Provenance Dataset	Translation Time	Workflow Execution Time
Phylo	Taverna	1.17 sec	179.00 sec
SciEvol	VisTrails	0.34 sec	13.00 sec

Comparing the workflow execution time of Phylo workflow in Taverna, the time consumed in the provenance translation process was much smaller (179 sec and 1.06 sec, respectively). Additionally, the time spent in the translation of the provenance generated by VisTrails was 35.13 times smaller than the execution time of the SciEvol workflow.

4.3.3 STORAGE OVERHEAD

We also measure the size of each provenance dataset before and after the translation. Aiming at collecting the dataset growth for different numbers of workflow executions, we have analyzed 1, 10, and 100 executions of Phylo and SciEvol workflows. The results of such measurement are shown in Table 22, Table 23, and Table 24. In those tables, the *Original Dataset* shows the size of the dataset generated by a WfMS in a specific format and proprietary model. In this case, Taverna uses RDF (Turtle files) to store provenance while VisTrails uses relational tables (MySQL). On the other hand, the *Translated Dataset* represents the size of the knowledge base composed by Prolog facts.

Table 22. Size of the datasets before and after the translation process for 1 execution

Workflows	Provenance Dataset	Original Dataset	Translated Dataset
Phylo	Taverna	76.9 KB	25 KB
SciEvol	VisTrails	736 KB	13 KB

Table 23. Size of datasets before and after the translation process for 10 executions

Workflows	Provenance Dataset	Original Dataset	Translated Dataset
Phylo	Taverna	776 KB	250 KB
SciEvol	VisTrails	736 KB	33 KB

Table 24. Size of datasets before and after the translation process for 100 executions

Workflows	Provenance Dataset	Original Dataset	Translated Dataset
Phylo	Taverna	7,884 KB	2,500 KB
SciEvol	VisTrails	800 KB	233 KB

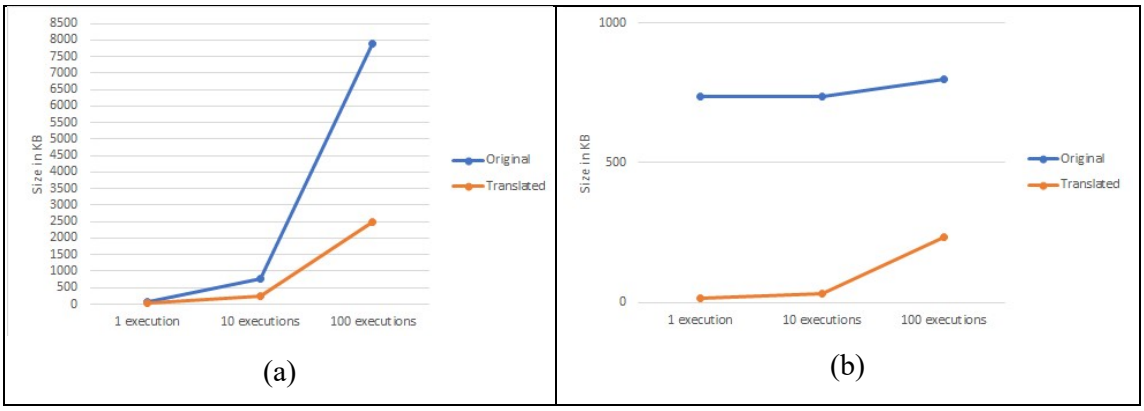


Figure 29. Sizes of the datasets for 1 execution of Phylo (a) and SciEvol (b) workflows

As presented in

Table 22, the *Translated Dataset* size of Phylo workflow is more than 3 times smaller than the size of the original dataset in Taverna. On the other hand, the *Translated Dataset* size of SciEvol is 56.6 times smaller than the original dataset in VisTrails for 1 workflow execution.

The original dataset generated after 10 executions of Phylo and SciEvol workflows is of size 776 KB and 736 KB, respectively. The size of the *Original* and *Translated* datasets for 10 executions of Phylo and SciEvol workflows are shown in Table 23. The *Translated Dataset* is more than 3 times smaller than the *Original Dataset* of Phylo workflow while the *Translated Dataset* of SciEvol workflow is 22.3 times smaller than the *Original Dataset*.

After 100 executions of Phylo and SciEvol workflows, the *Original Dataset* of each one got 7,884 KB and 800 KB, respectively. The size of each dataset (*Original and Translated*) is listed in Table 24 and graphically exposed in the charts of Figure 29. The *Original Datasets* of Phylo and SciEvol workflows are more than 3 times greater than the *Translated Datasets* in both cases.

The *Translated Dataset* produced by 1, 10, and 100 executions of Phylo and SciEvol workflows got smaller size than the *Original Dataset* in all cases, as shown in the charts of Figure 29. This is due the fact that just the provenance elements that have their correspondent elements in ProvONE were extracted and translated from the heterogeneous provenance datasets to the global knowledge base of Prolog facts. Moreover, the knowledge base structure

is composed of simple elements representing facts and rules that have no big impact over the provenance dataset.

4.4 CONCLUDING REMARKS

The experiments presented in this chapter evaluated the answers given by volunteers to questions based on classes 4, 5, and 6 (described in Chapter 3) that consider distinct provenance graphs. We compare the effectiveness, time spent and efficiency variables based on the answers given by two different group of volunteers using the *Standalone* and *Integrated* approach. The results show the *Integrated* approach is more effective in most questions/answers (1, 4, 5, and 6). The result for the efficiency variable was even better for the *Integrated* approach where it got higher values for questions/answers 1, 2, 4, 5 and 6. The time spent was small in the half of the questions/answers (2, 5, and 6) by using the *Standalone* approach. This is due to the fact that most answers given by the volunteers (questions 5 and 6) that used the *Standalone* approach were not correct. On the other hand, the general results showed that the *Integrated* approach was better than the *Standalone* approach for all variables (effectiveness, time spent, and efficiency). Besides that, we got statistical significance ($p\text{-}value < 0.05$) for the effectiveness and efficiency variables in the general results for all questions.

We also performed an overhead evaluation to assess the impact of the translation and process. The time spent in this process was negligible if compared with the workflow execution time. The *Translated* datasets size was smaller than the *Original* datasets for all numbers of executions (1, 10, and 100).

Chapter 5 – **CONCLUSION**

5.1 FINAL REMARKS

Analyzing provenance data is still an open, yet fundamental problem that deserves special attention from the scientific community. In this sense, the integration of heterogeneous provenance data sources can be a powerful tool for provenance analytics. In particular, it can provide considerable advantages for research teams that work collaboratively on similar experiments. In this thesis, we have presented an approach that enables integrating and querying provenance data from similar workflows designed and implemented in different systems with different specifications. We also developed mechanisms that allow for scientists to analyze different types of provenance data.

To achieve the provenance heterogeneous analysis, we propose a Provenance Integration Architecture that uses an integration model (ProvONE) that includes both *p-prov* and *r-prov* and create cartridges that convert different provenance databases to a global ProvONE schema of Prolog facts. Our approach introduces classes that explore intersection between *p-prov*, *r-prov*, and heterogeneous provenance graphs and presents related queries that run across both provenance graphs and retrieve information with different contents and levels of detail. Prolog rules were developed for each pre-defined query, taking advantage of inference and unification facilities provided by Prolog.

In our case studies, Prolog queries were executed and they could retrieve the data traces from both provenance graphs. New Prolog rules can easily be designed to accommodate new requirements, and new cartridges can be developed for other workflow systems using the proposed architecture. The development of cartridges for the provenance translation of four well-known workflow systems shows its feasibility and encourages the development of new cartridges.

We conduct an experimental evaluation with volunteers where the *Standalone* provenance analysis approach was compared to the *Integrated* approach. The experiment evaluated the answers given by volunteers to questions that consider distinct provenance graphs. We compare the effectiveness, time spent and efficiency variables based on the answers given by the volunteers using the *Standalone* and *Integrated* approach. The results show the *Integrated* approach is more effective in most questions/answers. The result for the efficiency variable was even better for the *Integrated* approach where it got higher values for questions/answers. The time spent was smaller in half of the questions/answers by using the

Standalone approach. This is due to the fact that most answers given by the volunteers that used the *Standalone* approach were not correct. On the other hand, the general results showed that the *Integrated* approach was better than the *Standalone* approach for all variables. Besides that, we got statistical significance (*p-value* < 0.05) for the effectiveness and efficiency variables in the general results for all questions.

There are some limitations, however. In our evaluation, volunteers that have different knowledge levels of Prolog were invited to participate of the experiment. We tried to make a balance in the distribution of volunteers between the *Standalone* and *Integrated* approaches based on their expertise informed in the personal profile form (Appendix A). However, there is no way to confirm that 100% of those levels were correctly informed. Some of them, for example, have studied or used Prolog some years ago. Hence, these volunteers had some basic difficulties during the experiment that could somehow have influenced the results. During the experiment, we also detected that some volunteers had some difficulties with the SWI Prolog program used in Linux OS. This could have made them spend more time to run the queries and answer the questions.

Before the beginning of the experiment, we presented terms, concepts, and technologies related to workflows and provenance. Therefore, some volunteers that have some previous contact with this subject could have better results than the volunteers that have no idea about what workflow or provenance are.

We also performed an overhead evaluation to verify possible processing and storage problems. The time spent in those processes was negligible if compared with the workflow execution time. The *Translated* datasets size was smaller than the *Original* datasets for all numbers of executions (1, 10, and 100).

The overhead evaluation included different number of executions to see how the datasets scale in different scenarios (workflows execution and provenance translation). However, the impact of huge datasets with 1,000 executions or more, for example, was not measured and evaluated.

5.2 CONTRIBUTIONS

A list of the contributions of this thesis is as follows:

- We conducted a survey with 82 scientists from different countries that confirm the scenario of integrated provenance analysis can manifest in practice;
- We published a survey about provenance analysis approaches;
- We designed a new taxonomy that can guide current and future studies in this area;

- We defined a reference classification of the provenance space;
- We proposed an approach to analyze provenance from heterogeneous provenance graphs by using ProvONE as a canonical model;
- We developed cartridges to automatically translate provenance data from similar workflows executed in different and well-known WfMSs (Taverna, VisTrails, SciCumulus, and e-Science Central) to Prolog facts, following the ProvONE model;
- We developed mechanisms to infer some relationships that are not present in the original provenance datasets;
- We designed and implemented a new form of provenance representation in Prolog that includes PROV and ProvONE prefix and attributes;
- We constituted a global knowledge base of Prolog facts and rules structured following the ProvONE model;
- We implemented a set of Prolog rules that facilitate the query and analysis process of heterogeneous provenance;
- We conducted an experimental evaluation including the effectiveness, time spent, and efficiency variables;
- We conducted an overhead evaluation of the translation process (execution and storage).

All of those contributions were compiled in the form of papers/articles that were submitted to different Computer Science events and journals. A list of the papers written during the PhD period is presented as follows:

- *OLIVEIRA, Wellington; DE OLIVEIRA, Daniel; BRAGANHOLO, Vanessa. Provenance Analytics for Workflow-based Computational Experiments: a Survey. ACM Computing Surveys, p. 1–29, 2018 (to appear).*
- *OLIVEIRA, Wellington; NEVES, Victor C.; OCAÑA, Kary A. C. S.; MURTA, Leonardo; DE OLIVEIRA, Daniel; and BRAGANHOLO, Vanessa. Captura e Consulta a Dados de Proveniência Retrospectiva Implícita Intra-Atividade. In SBBD, 2014. p. 37-46.*
- *OLIVEIRA, Wellington; DE OLIVEIRA, Daniel; and BRAGANHOLO, Vanessa. Experiencing PROV-Wf for Provenance Interoperability in SWfMSs. In International Provenance and Annotation Workshop (IPAW), 2014. p. 294-296.*
- *OLIVEIRA, Wellington; MISSIER, Paolo; Ocaña, Kary A. C. S.; DE OLIVEIRA, Daniel; and BRAGANHOLO, Vanessa. Analyzing Provenance Across Heterogeneous Provenance Graphs. In International Provenance and Annotation Workshop (IPAW), 2016. p. 57-70.*

- *OLIVEIRA, Wellington; MISSIER, Paolo; DE OLIVEIRA, Daniel; and BRAGANHOLO, Vanessa. Comparing Provenance Data Models for Scientific Workflows: an Analysis of PROV-Wf and ProvOne. In Brazilian e-Science Workshop (BRESCI), 2016. p. 1-8.*
- *OLIVEIRA, Wellington; OCAÑA, Kary A. C. S.; DE OLIVEIRA, Daniel; BRAGANHOLO, Vanessa. Querying Provenance along with External Domain Data Using Prolog. Journal of Information and Data Management (JIDM). v. 16, n. 1, p. 3–18, Apr. 2017.*

OLIVEIRA, Wellington; MISSIER, Paolo; OCAÑA, Kary A. C. S.; DE OLIVEIRA, Daniel; and BRAGANHOLO, Vanessa. A Provenance Integration Architecture for Analyzing Heterogeneous Provenance Graphs. FGCS, to be submitted in 2018.

5.3 FUTURE WORK

As future work, we plan to develop a benchmark of completeness to evaluate provenance from different WfMSs. We intend to investigate how to cover gaps in similar provenance graphs by using our intersection classes. Furthermore, a semi-automatic mechanism to suggest semantic links between data and processes is in progress. In this sense, we intend to investigate other solutions such as the one proposed by Leme *et al.* (2009) that developed similarity functions to schema matching for an OWL dialect.

We also plan to make querying easier by providing an interface where the user would be able to select attributes, and the system would automatically generate the corresponding Prolog query. This would eliminate the need of a Computer Scientist to write the rules in our current approach. An initial idea is discussed by Martins (2013).

To facilitate the comparison between the activities and data from heterogeneous provenance graphs, we plan to extract metadata from the file system about the size, modified date, extension, and other properties of data and programs. From these information, we could provide suggestions about similar data and activities. Once the scientist confirms the equality or similarity between the them, such mechanism could add this information automatically in the knowledge base.

The cartridges developed in this work cover well-known WfMS that are used for many scientists around the world. However, new cartridges can be implemented to translate provenance from other WfMS to a global knowledge base of facts that follows ProvONE model.

We intend to design a Web portal and implement services to provide the translation, sharing, integration, and querying of provenance datasets for all scientists spread out across the

globe. Such tool will help scientists that want to share their results and improve their experiments analysis by working collaboratively.

Finally, we plan to evaluate the efficiency and effectiveness of our approach in real teams of scientists that work collaboratively.

REFERENCES

ABITEBOUL, Serge; QUASS, Dallan; MCHUGH, Jason; WIDOM, Jennifer; WIENER, Janet. The Lorel Query Language for Semistructured Data. *International Journal on Digital Libraries*, v. 1, n. 1, p. 68–88, 1997.

ABOUELHODA, Mohamed; ISSA, Shadi; GHANEM, Moustafa. Tavaxy: Integrating Taverna and Galaxy workflows with cloud computing support. *BMC Bioinformatics*, v. 13, p. 77, 2012.

ABRAMOVA, Veronika; BERNARDINO, Jorge. NoSQL Databases: MongoDB vs Cassandra. C3S2E '13, 2013, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2013. p. 14–22. Available: <http://doi.acm.org/10.1145/2494444.2494447>. Accessed: 23 aug. 2014.

ALTINTAS, Ilkay; ANAND, Manish K.; VUONG, Trung N.; BOWERS, Shawn; LUDÄSCHER, Bertram; SLOOT, Peter M. A. A Data Model for Analyzing User Collaborations in Workflow-Driven eScience. *International Journal of Computers and Their Applications*, v. 18, p. 160–179, 2011.

ALTINTAS, Ilkay; BARNEY, Oscar; JAEGER-FRANK, Efrat. Provenance Collection Support in the Kepler Scientific Workflow System. *Provenance and Annotation of Data*. LNCS. [S.l.]: Springer Berlin, 2006. v. 4145. p. 118–132. Available: <http://dx.doi.org/10.1007/11890850_14>. Accessed: 9 jan. 2009.

ANAND, Manish Kumar; BOWERS, Shawn; LUDÄSCHER, Bertram. A Navigation Model for Exploring Scientific Workflow Provenance Graphs. WORKS '09, 2009, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2009. p. 1–10.

ANAND, Manish Kumar; BOWERS, Shawn; LUDÄSCHER, Bertram. Database Support for Exploring Scientific Workflow Provenance Graphs. In: AILAMAKI, ANASTASIA; BOWERS, SHAWN (Org.). . *Scientific and Statistical Database Management*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2012. p. 343–360.

ANAND, M.K.; BOWERS, S.; LUDASCHER, B. Provenance browser: Displaying and querying scientific workflow provenance graphs. In: 2010 IEEE 26TH INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE), 2010, [S.l: s.n.], 2010. p. 1201–1204.

BAO, Zhuowei; COHEN-BOULAKIA, S.; DAVIDSON, S.B.; EYAL, A.; KHANNA, S. Differencing Provenance in Scientific Workflows. In: IEEE INTERNATIONAL CONFERENCE ON DATA ENGINEERING, 2009, [S.l: s.n.], 2009. p. 808–819.

BAO, Zhuowei; COHEN-BOULAKIA, Sarah; DAVIDSON, Susan B.; GIRARD, Pierrick. PDiffView: Viewing the Difference in Provenance of Workflow Results. *VLDB Endowment*, v. 2, n. 2, p. 1638–1641, 2009.

BATINI, C.; LENZERINI, M.; NAVATHE, S. B. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Comput. Surv.*, v. 18, n. 4, p. 323–364, dec. 1986.

BELHAJJAME, Khalid; ZHAO, Jun; GARIJO, Daniel; GAMBLE, Matthew; HETTNE, Kristina; PALMA, Raul; MINA, Eleni; CORCHO, Oscar; GÓMEZ-PÉREZ, José Manuel; BECHHOFER, Sean; KLYNE, Graham; GOBLE, Carole. Using a suite of ontologies for preserving workflow-centric research objects. *Web Semantics: Science, Services and Agents on the World Wide Web*, v. 32, p. 16–42, may 2015.

BERGLUND, Anders; BOAG, Scott; CHAMBERLIN, Don; FERNÁNDEZ, Mary F.; KAY, Michael; ROBIE, Jonathan; SIMÉON, Jérôme. *XML Path Language (XPath) 2.0 (Second Edition)*. Available: <http://www.w3.org/TR/xpath20/>. Accessed: 21 jan. 2014.

BITON, O.; COHEN-BOULAKIA, S.; DAVIDSON, S.B.; HARA, C.S. Querying and Managing Provenance through User Views in Scientific Workflows. In: IEEE 24TH INTERNATIONAL CONFERENCE ON DATA ENGINEERING, 2008. ICDE 2008, 2008, [S.l: s.n.], 2008. p. 1072–1081.

BITON, Olivier; COHEN-BOULAKIA, Sarah; DAVIDSON, Susan B. Zoom*UserViews: Querying Relevant Provenance in Workflow Systems. 2007, Vienna, Austria. *Anais*... Vienna, Austria: VLDB Endowment, 2007. p. 1366–1369.

BOAG, Scott; CHAMBERLIN, Don; FERNANDEZ, Mary F.; FLORESCU, Daniela; ROBIE, Jonathan; SIMÉON, Jérôme. *XQuery 1.0: An XML Query Language*. Available: <www.w3.org/TR/xquery>. Accessed: 9 jun. 2011.

BORKIN, M. A.; YEH, C. S.; BOYD, M.; MACKO, P.; GAJOS, K. Z.; SELTZER, M.; PFISTER, H. Evaluation of Filesystem Provenance Visualization Tools. *IEEE Transactions on Visualization and Computer Graphics*, v. 19, n. 12, p. 2476–2485, dec. 2013.

BOSE, R.; FREW, J. Composing lineage metadata with XML for custom satellite-derived data products. In: 16TH INTERNATIONAL CONFERENCE ON SCIENTIFIC AND STATISTICAL DATABASE MANAGEMENT, 2004. PROCEEDINGS, jun. 2004, [S.l: s.n.], jun. 2004. p. 275–284.

BOSE, Rajendra; FREW, James. Lineage Retrieval for Scientific Data Processing: A Survey. *ACM Computing Surveys*, v. 37, n. 1, p. 1–28, 2005.

BROEKSTRA, Jeen; KAMPMAN, Arjohn; HARMELEN, Frank Van. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In: HORROCKS, IAN; HENDLER, JAMES (Org.). . *The Semantic Web — ISWC 2002*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2002. p. 54–68. Available: <http://link.springer.com/chapter/10.1007/3-540-48005-6_7>. Accessed: 24 jul. 2015.

CALLAHAN, Steven P.; FREIRE, Juliana; SANTOS, Emanuele; SCHEIDEGGER, Carlos E.; SILVA, Cláudio T.; VO, Huy T. *Using provenance to streamline data exploration through visualization*. , UUSCI.Technical Report, nº 2006-016. Utah - USA: SCI Institute–Univ. of Utah, 2006a.

CALLAHAN, Steven P.; FREIRE, Juliana; SANTOS, Emanuele; SCHEIDEGGER, Carlos E.; SILVA, Cláudio T.; VO, Huy T. VisTrails: Visualization Meets Data Management. 2006b, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2006. p. 745–747. Available: <http://doi.acm.org/10.1145/1142473.1142574>. Accessed: 16 mar. 2014.

CARROLL, Jeremy J.; DICKINSON, Ian; DOLLIN, Chris; REYNOLDS, Dave; SEABORNE, Andy; WILKINSON, Kevin. Jena: Implementing the Semantic Web Recommendations. WWW Alt. '04, 2004, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2004. p. 74–83. Available: <http://doi.acm.org/10.1145/1013367.1013381>. Accessed: 24 jul. 2015.

CERI, S.; GOTTLOB, G.; TANCA, L. What you always wanted to know about Datalog (and never dared to ask). *EEE Transactions on Knowledge and Data Engineering*, v. 1, n. 1, p. 146–166, 1989.

CHEBOTKO, Artem; LU, Shiyong; FEI, Xubo; FOTOUHI, Farshad. RDFProv: A relational RDF store for querying and managing scientific workflow provenance. *Data & Knowledge Engineering*, v. 69, n. 8, p. 836–865, 2010.

CHEN, Peng; PLALE, Beth; AKTAS, Mehmet. *Temporal Data Mining of Scientific Data Provenance*. , TR701.Technical Report. Bloomington - USA: Indiana University Computer Science, 2012.

CHEN, Peng; PLALE, Beth; AKTAS, Mehmet S. Temporal representation for mining scientific data provenance. *Future Generation Computer Systems*, Special Section: Intelligent Big Data ProcessingSpecial Section: Behavior Data Security Issues in Network Information PropagationSpecial Section: Energy-efficiency in Large Distributed Computing ArchitecturesSpecial Section: eScience Infrastructure and Applications. v. 36, p. 363–378, jul. 2014.

CHEUNG, Kwok; HUNTER, Jane. Provenance Explorer – Customized Provenance Views Using Semantic Inferencing. Lecture Notes in Computer Science, 5 nov. 2006, [S.l.]: Springer Berlin Heidelberg, 5 nov. 2006. p. 215–227. Available: <http://link.springer.com/chapter/10.1007/11926078_16>. Accessed: 6 aug. 2016.

CHONG, S. *Logic Programming*. Computing Science Technical Report, nº lec19. Cambridge, MA: Harvard School of Engineering and Applied Sciences, 2016.

CLIFF, Norman. *Ordinal Methods for Behavioral Data Analysis*. [S.l.]: Psychology Press, 1996.

COCHRAN, William G.; COX, Gertrude M. Experimental Designs. *Soil Science*, v. 70, n. 2, p. 164, aug. 1950.

COHEN, Shirley; COHEN-BOULAKIA, Sarah; DAVIDSON, Susan. Towards a Model of Provenance and User Views in Scientific Workflows. In: LESER, ULF; NAUMANN, FELIX; ECKMAN, BARBARA (Org.). . *Data Integration in the Life Sciences*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2006. p. 264–279.

COHEN-BOULAKIA, Sarah; BELHAJJAME, Khalid; COLLIN, Olivier; CHOPARD, Jérôme; FROIDEVAUX, Christine; GAIGNARD, Alban; HINSEN, Konrad; LARMANDE, Pierre; BRAS, Yvan Le; LEMOINE, Frédéric; MAREUIL, Fabien; MÉNAGER, Hervé; PRADAL, Christophe; BLANCHET, Christophe. Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. *Future Generation Computer Systems*, v. 75, p. 284–298, 1 oct. 2017.

COHEN-BOULAKIA, Sarah; BITON, Olivier; COHEN, Shirley; DAVIDSON, Susan. Addressing the provenance challenge using ZOOM. *Concurrency and Computation: Practice and Experience*, v. 20, n. 5, p. 497–506, 2008.

CORCHO, Oscar; GARIJO VERDEJO, Daniel; BELHAJJAME, K.; ZHAO, Jun; MISSIER, P.; NEWMAN, David; PALMA, R.; BECHHOFER, S.; GARCÍA CUESTA, Esteban; GÓMEZ-PÉREZ, José Manuel; KLYNE, Graham; ROOS, Marco; RUIZ, José Enrique;

SOILAND-REYES, Stian; VERDES-MONTENEGRO, Lourdes; DE ROURE, D.; GOBLE, C. Workflow-centric research objects: First class citizens in scholarly discourse. In: 9 TH EXTENDED SEMANTIC WEB CONFERENCE HERSONISSOS, 2012, Hersonissos, Creta (Grecia). *Anais*... Hersonissos, Creta (Grecia): Facultad de Informática (UPM), 2012. p. 1–12. Available: <http://sepublica.mywikipaper.org/sepublica2012.pdf>. Accessed: 4 sep. 2016.

COSTA, Flavio; SILVA, Vítor; DE OLIVEIRA, Daniel; OCAÑA, Kary; OGASAWARA, Eduardo; DIAS, Jonas; MATTOSO, Marta. Capturing and querying workflow runtime provenance with PROV: a practical approach. 2013a, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2013. p. 282–289.

COSTA, Flavio; SILVA, Vítor; DE OLIVEIRA, Daniel; OCAÑA, Kary; OGASAWARA, Eduardo; DIAS, Jonas; MATTOSO, Marta. Capturing and Querying Workflow Runtime Provenance with PROV: A Practical Approach. 2013b, New York, NY, USA. *Anais*... New York, NY, USA: ACM Press, 2013. p. 282–289.

CRUZ, Sergio Manuel Serra Da; CAMPOS, M.; MATTOSO, M. Towards a Taxonomy of Provenance in Scientific Workflow Management Systems. 2009, Los Angeles, California, United States. *Anais*... Los Angeles, California, United States: [s.n.], 2009.

CUEVAS-VICENTTIN, V.; DEY, S.; WANG, M.L.Y.; SONG, Tianhong; LUDASCHER, B. Modeling and Querying Scientific Workflow Provenance in the D-OPM. In: HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS (SCC), 2012 SC COMPANION:, nov. 2012, [S.l: s.n.], nov. 2012. p. 119–128.

CUEVAS-VICENTTÍN, Víctor; KIANMAJD, Parisa; LUDÄSCHER, Bertram; MISSIER, Paolo; CHIRIGATI, Fernando; WEI, Yaxing; KOOP, David; DEY, Saumen. The PBase Scientific Workflow Provenance Repository. *International Journal of Digital Curation*, v. 9, n. 2, p. 28–38, 23 oct. 2014.

CUI LIN; SHIYONG LU; XUBO FEI; CHEBOTKO, A.; DARSHAN PAI; ZHAOQIANG LAI; FOTOUHI, F.; JING HUA. A Reference Architecture for Scientific Workflow Management Systems and the VIEW SOA Solution. *IEEE Transactions on Services Computing*, v. 2, n. 1, p. 79–92, mar. 2009.

DA CRUZ, S.M.S.; BARROS, P.M.; BISCH, P.M.; CAMPOS, M.L.M.; MATTOSO, M. Provenance Services for Distributed Workflows. In: INTERNATIONAL SYMPOSIUM ON CLUSTER COMPUTING AND THE GRID, 2008, [S.l: s.n.], 2008. p. 526–533.

DAVIDSON, Susan B.; FREIRE, Juliana. Provenance and Scientific Workflows: Challenges and Opportunities. 2008, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2008. p. 1345–1350.

DAVIDSON, Susan; CHEN, Yi; SUN, Peng; COHEN-BOULAKIA, Sarah. On User Views in Scientific Workflow Systems. In: SWPM, 2009, Washington, USA. *Anais*... Washington, USA: [s.n.], 2009.

DAVIDSON, Susan; COHEN-BOULAKIA, Sarah; EYAL, Anat; LUDASCHER, Bertram; MCPHILLIPS, Timothy; BOWERS, Shawn; ANAND, Manish Kumar; FREIRE, Juliana. Provenance in Scientific Workflow Systems. *Bulletin of the IEEE Computer Society Technical Commit on Data Engineering*, v. 30, n. 4, p. 44–50, 2007.

DE OLIVEIRA, Daniel; OGASAWARA, Eduardo; BAIÃO, Fernanda; MATTOSO, Marta. SciCumulus: A Lightweight Cloud Middleware to Explore Many Task Computing Paradigm in Scientific Workflows. 2010, Washington, DC, USA. *Anais*... Washington, DC, USA: [s.n.], 2010. p. 378–385. Available: <http://dx.doi.org/10.1109/CLOUD.2010.64>. Accessed: 2 nov. 2011.

DE ROURE, David; GOBLE, Carole; STEVENS, Robert. The design and Realisation of the Virtual Research Environment for Social Sharing of Workflows. *Future Generation Computer Systems*, v. 25, n. 5, p. 561–567, 2009.

DEELMAN, Ewa; SINGH, Gurmeet; SU, Mei-Hui; BLYTHE, James; GIL, Yolanda; KESSELMAN, Carl; MEHTA, Gaurang; VAHI, Karan; BERRIMAN, G. Bruce; GOOD, John; LAITY, Anastasia; JACOB, Joseph C.; KATZ, Daniel S. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, v. 13, n. 3, p. 219–237, 1 jan. 2005.

DEL RIO, Nicholas; DA SILVA, Paulo. Probe-It! Visualization Support for Provenance. *Advances in Visual Computing*. [S.l: s.n.], 2007. p. 732–741. Available: <http://dx.doi.org/10.1007/978-3-540-76856-2_72>. Accessed: 1 may 2009.

DEY, Saumen; CUEVAS-VICENTTÍN, Víctor; KÖHLER, Sven; GRIBKOFF, Eric; WANG, Michael; LUDÄSCHER, Bertram. On Implementing Provenance-aware Regular Path Queries with Relational Query Engines. EDBT '13, 2013, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2013. p. 214–223. Available: <http://doi.acm.org/10.1145/2457317.2457353>. Accessed: 30 aug. 2016.

DEY, Saumen; KÖHLER, Sven; BOWERS, Shawn; LUDÄSCHER, Bertram. Datalog as a Lingua Franca for Provenance Querying and Reasoning. 2012, [S.l: s.n.], 2012.

DO, Chuong B.; MAHABHASHYAM, Mahathi S. P.; BRUDNO, Michael; BATZOGLOU, Serafim. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Research*, v. 15, n. 2, p. 330–340, 1 feb. 2005.

ELLQVIST, Tommy; KOOP, David; FREIRE, Juliana; SILVA, Cláudio; STRÖMBÄCK, Lena. Using Mediation to Achieve Provenance Interoperability. In: 2009 WORLD CONFERENCE ON SERVICES - I, jul. 2009, [S.l.]: IEEE, jul. 2009. p. 291–298. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5190671>. Accessed: 23 mar. 2014.

ELMASRI, Ramez; NAVATHE, Shamkant. *Fundamentals of Database Systems*. 6. ed. [S.l.]: Addison-Wesley, 2010.

FAHRINGER, T.; PRODAN, R.; RUBING DUAN; NERIERI, F.; PODLIPNIG, S.; JUN QIN; SIDDIQUI, M.; HONG-LINH TRUONG; VILLAZON, A.; WIECZOREK, M. ASKALON: a Grid application development and computing environment. 13 nov. 2005, Seattle, Washington, USA. *Anais*... Seattle, Washington, USA: IEEE, 13 nov. 2005. p. 122–131.

FELSENSTEIN, J. PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics*, v. 5, p. 164–166, 1989.

FILETO, Renato; MEDEIROS, Claudia Bauzer; LIU, Ling; PU, Calton; ASSAD, Eduardo Delgado. Using Domain Ontologies to Help Track Data Provenance. In: BRAZILIAN SYMPOSIUM ON DATABASES, 2003, [S.l: s.n.], 2003. p. 84–98.

FOSTER, I.; VÖCKLER, J.; WILDE, M.; ZHAO, Yong. Chimera: a virtual data system for representing, querying, and automating data derivation. In: 14TH INTERNATIONAL CONFERENCE ON SCIENTIFIC AND STATISTICAL DATABASE MANAGEMENT, 2002. PROCEEDINGS, 2002, [S.l: s.n.], 2002. p. 37–46.

FREIRE, Juliana; KOOP, David; SANTOS, Emanuele; SILVA, Cláudio T. Provenance for Computational Tasks: A Survey. *Computing in Science and Engineering*, v. 10, n. 3, p. 11–21, 2008.

FREIRE, Juliana; SILVA, Cláudio. Towards Enabling Social Analysis of Scientific Data. In: CHI SOCIAL DATA ANALYSIS WORKSHOP, 2008a, Florence, Italy. *Anais*... Florence, Italy: ACM, 2008. p. 3977–3980.

FREIRE, Juliana; SILVA, Cláudio T. Simplifying the Design of Workflows for Large-Scale Data Exploration and Visualization. In: IN MICROSOFT ESCIENCE WORKSHOP, 2008b, North Carolina - USA. *Anais*... North Carolina - USA: [s.n.], 2008. p. 1–3.

FREIRE, Juliana; SILVA, Cláudio T.; CALLAHAN, Steven P.; SANTOS, Emanuele; SCHEIDEGGER, Carlos E.; VO, Huy T. Managing Rapidly-Evolving Scientific Workflows. In: MOREAU, LUC; FOSTER, IAN (Org.). . *Provenance and Annotation of Data*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2006. p. 10–18.

FREW, J.; BOSE, R. Earth System Science Workbench: a data management infrastructure for earth science products. In: THIRTEENTH INTERNATIONAL CONFERENCE ON SCIENTIFIC AND STATISTICAL DATABASE MANAGEMENT, 2001. SSDBM 2001. PROCEEDINGS, 2001, [S.l: s.n.], 2001. p. 180–189.

FREW, James; METZGER, Dominic; SLAUGHTER, Peter. Automatic Capture and Reconstruction of Computational Provenance. *Concurrency and Computation: Practice and Experience*, v. 20, n. 5, p. 485–496, 2008.

GADELHA, Luiz M. R.; WILDE, Michael; MATTOSO, Marta; FOSTER, Ian. MTCProv: a practical provenance query framework for many-task scientific computing. *Distributed and Parallel Databases*, v. 30, n. 5–6, p. 351–370, 2012.

GASPAR, Wander; BRAGA, Regina; CAMPOS, Fernanda. SciProv: An Architecture for Semantic Query in Provenance Metadata on e-Science Context. In: BÖHM, CHRISTIAN; KHURI, SAMI; LHOTSKÁ, LENKA; PISANTI, NADIA (Org.). . *Information Technology in Bio- and Medical Informatics*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2011. p. 68–81.

GILBERT, Don. Sequence File Format Conversion with Command-Line Readseq. *Current Protocols in Bioinformatics*. [S.l.]: John Wiley & Sons, Inc., 2002. . Available: <http://onlinelibrary.wiley.com/doi/10.1002/0471250953.bia01es00/abstract>. Accessed: 3 nov. 2016.

GONÇALVES, João Carlos de A. R.; OLIVEIRA, Daniel De; OCAÑA, Kary A. C. S.; OGASAWARA, Eduardo; MATTOSO, Marta. Using Domain-Specific Data to Enhance

Scientific Workflow Steering Queries. In: INTERNATIONAL PROVENANCE AND ANNOTATION WORKSHOP (IPAW), LNCS, 2012, Santa Barbara, CA. *Anais*... Santa Barbara, CA: [s.n.], 2012. p. 152–167.

GOODMAN, Leo A. Snowball Sampling. *The Annals of Mathematical Statistics*, Zbl: 0099.14203, v. 32, n. 1, p. 148–170, mar. 1961.

GUO, Philip J.; SELTZER, Margo. BURRITO: Wrapping Your Lab Notebook in Computational Infrastructure. 2012, Berkeley, CA, USA. *Anais*... Berkeley, CA, USA: USENIX Association, 2012. p. 1–4. Available: <http://dl.acm.org/citation.cfm?id=2342875.2342882>. Accessed: 24 mar. 2015.

HANSEN, CHARLES; JOHNSON, CHRIS R; SILVA, ClAUDIo T. Visualization for Data-Intensive Science. [S.l: s.n.], 2011. p. 151–161.

HERSCHEL, Melanie; DIESTELKÄMPER, Ralf; LAHMAR, Houssem Ben. A survey on provenance: What for? What form? What from? *The VLDB Journal*, v. 26, n. 6, p. 881–906, 1 dec. 2017.

HERSCHEL, Melanie; HLAWATSCH, Marcel. Provenance: On and Behind the Screens. SIGMOD '16, 2016, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2016. p. 2213–2217. Available: <http://doi.acm.org/10.1145/2882903.2912568>. Accessed: 13 aug. 2016.

HLAWATSCH, M.; BURCH, M.; BECK, F.; FREIRE, J.; SILVA, C.; WEISKOPF, D. Visualizing the Evolution of Module Workflows. In: 2015 19TH INTERNATIONAL CONFERENCE ON INFORMATION VISUALISATION, jul. 2015, [S.l: s.n.], jul. 2015. p. 40–49.

HOEKSTRA, Rinke; GROTH, Paul. PROV-O-Viz - Understanding the Role of Activities in Provenance. In: INTERNATIONAL PROVENANCE AND ANNOTATION WORKSHOP (IPAW), 9 jun. 2014, Cologne, Germany. *Anais*... Cologne, Germany: Springer International Publishing, 9 jun. 2014. p. 215–220. Available: <http://link.springer.com/chapter/10.1007/978-3-319-16462-5_18>. Accessed: 10 aug. 2016.

HOLLAND, D.; BRAUN, U.; MACLEAN, D.; MUNISWAMY-REDDY, K.; SELTZER, M. Choosing a Data Model and Query Language for Provenance. 2008, Salt Lake City, UT, USA. *Anais*... Salt Lake City, UT, USA: [s.n.], 2008. p. 1–8.

HOLLAND, David A.; SELTZER, Margo I.; BRAUN, Uri; MUNISWAMY-REDDY, Kiran-Kumar. PASSing the provenance challenge. *Concurrency and Computation: Practice and Experience*, v. 20, n. 5, p. 531–540, 2008.

HOWE, Jeff. The Rise of Crowdsourcing. *Wired*, v. 14, n. 6, p. 1–4, 2006.

HU, Hua; LIU, Zhanchen; HU, Haiyang. Reconstructing Unsound Data Provenance View in Scientific Workflow. In: WANG, HUA; ZOU, LEI; HUANG, GUANGYAN; HE, JING; PANG, CHAOYI; ZHANG, HAO LAN; ZHAO, DONGYAN; YI, ZHUANG (Org.). . *Web Technologies and Applications*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2012. p. 212–220. Available: <http://link.springer.com/chapter/10.1007/978-3-642-29426-6_25>. Accessed: 2 dec. 2013.

HULL, Duncan; WOLSTENCROFT, Katy; STEVENS, Robert; GOBLE, Carole; POCOCK, Mathew R; LI, Peter; OINN, Tom. Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, v. 34, n. 2, p. 729–732, 2006.

KARSAI, Linus; FEKETE, Alan; KAY, Judy; MISSIER, Paolo. Clustering Provenance Facilitating Provenance Exploration Through Data Abstraction. HILDA '16, 2016, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2016. p. 6:1–6:5. Available: <http://doi.acm.org/10.1145/2939502.2939508>. Accessed: 13 aug. 2016.

KATOH, Kazutaka; TOH, Hiroyuki. Parallelization of the MAFFT multiple sequence alignment program. *Bioinformatics (Oxford, England)*, v. 26, n. 15, p. 1899–1900, 1 aug. 2010.

KEANE, Thomas M.; CREEVEY, Christopher J.; PENTONY, Melissa M.; NAUGHTON, Thomas J.; MCLNERNEY, James O. Assessment of methods for amino acid matrix selection and their use on empirical data shows that ad hoc assumptions for choice of matrix are not justified. *BMC Evolutionary Biology*, v. 6, p. 29, 2006.

KOHWALTER, Troy; OLIVEIRA, Thiago; FREIRE, Juliana; CLUA, Esteban; MURTA, Leonardo. Prov Viewer: A Graph-Based Visualization Tool for Interactive Exploration of Provenance Data. In: MATTOSO, MARTA; GLAVIC, BORIS (Org.). . *Provenance and Annotation of Data and Processes*. Lecture Notes in Computer Science. [S.l.]: Springer International Publishing, 2016. p. 71–82. Available: <http://link.springer.com/chapter/10.1007/978-3-319-40593-3_6>. Accessed: 4 aug. 2016.

LARKIN, M A; BLACKSHIELDS, G; BROWN, N P; CHENNA, R; MCGETTIGAN, P A; MCWILLIAM, H; VALENTIN, F; WALLACE, I M; WILM, A; LOPEZ, R; THOMPSON, J D; GIBSON, T J; HIGGINS, D G. Clustal W and Clustal X version 2.0. *Bioinformatics (Oxford, England)*, v. 23, n. 21, p. 2947–2948, 1 nov. 2007.

LASSMANN, Timo; FRINGS, Oliver; SONNHAMMER, Erik L. L. Kalign2: high-performance multiple alignment of protein and nucleotide sequences allowing external features. *Nucleic Acids Research*, v. 37, n. 3, p. 858–865, 1 feb. 2009.

LEME, Luiz André P. Paes; CASANOVA, Marco A.; BREITMAN, Karin K.; FURTADO, Antonio L. Instance-Based OWL Schema Matching. In: INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, Lecture Notes in Business Information Processing, 6 may 2009, [S.l.]: Springer, Berlin, Heidelberg, 6 may 2009. p. 14–26. Available: <https://link.springer.com/chapter/10.1007/978-3-642-01347-8_2>. Accessed: 4 mar. 2018.

LIM, Chunhyeok; LU, Shiyong; CHEBOTKO, A.; FOTOUHI, F. OPQL: A First OPM-Level Query Language for Scientific Workflow Provenance. In: 2011 IEEE INTERNATIONAL CONFERENCE ON SERVICES COMPUTING (SCC), 2011, [S.l: s.n.], 2011. p. 136–143.

LIM, Chunhyeok; LU, Shiyong; CHEBOTKO, Artem; FOTOUHI, Farshad; KASHLEV, Andrey. OPQL: Querying scientific workflow provenance at the graph level. *Data & Knowledge Engineering*, v. 88, p. 37–59, 2013.

LIN, Cui; LU, Shiyong; LAI, Zhaoqiang; CHEBOTKO, Artem; FEI, Xubo; HUA, Jing; FOTOUHI, Farshad. Service-Oriented Architecture for VIEW: A Visual Scientific Workflow Management System. 2008, [S.l.]: IEEE Computer Society, 2008. p. 335–342. Available: <http://portal.acm.org/citation.cfm?id=1447882>. Accessed: 5 mar. 2010.

MARINHO, Anderson; MATTOSO, Marta; WERNER, Claudia; BRAGANHOLO, Vanessa; MURTA, Leonardo. Challenges in managing implicit and abstract provenance data: experiences with ProvManager. In: 3RD USENIX WORKSHOP ON THE THEORY AND PRACTICE OF PROVENANCE, 2011, Crete, Greece. *Anais*... Crete, Greece: USENIX Association, 2011. p. 1–6.

MARTINS, G.; LARCHER, J.; OLIVEIRA, Alessandreia; MURTA, Leonardo; BRAGANHOLO, Vanessa. XChange: Compreensão de Mudanças em Documentos XML. In: SESSÃO DE DEMOS DO SIMPÓSIO BRASILEIRO DE BANCO DE DADOS (SBBD), 2013, Recife, PE. *Anais*... Recife, PE: SBC, 2013. p. 31–36.

MATES, Phillip; SANTOS, Emanuele; FREIRE, Juliana; SILVA, Cláudio T. CrowdLabs: Social Analysis and Visualization for the Sciences. *Scientific and Statistical Database Management*. Lecture Notes in Computer Science. [S.l.]: Springer, 2011. p. 555–564.

MATTOSO, Marta; WERNER, Claudia; TRAVASSOS, Guilherme Horta; BRAGANHOLO, Vanessa; OGASAWARA, Eduardo; OLIVEIRA, Daniel; CRUZ, Sergio; MARTINHO, Wallace; MURTA, Leonardo. Towards supporting the life cycle of large scale scientific experiments. *International Journal of Business Process Integration and Management*, v. 5, n. 1, p. 79–92, 2010.

MCGUINNESS, Deborah L.; SILVA, Paulo Pinheiro; CHANG, Cynthia. *IW-Base: Provenance Metadata Infrastructure for Explaining and Trusting Answers from the Web*. . [S.l.]: in Proc. Conf. ALGORITMY ’97, Zuberec, West Tatra Mountains, September 1-5, 2004.

MILLER, Justin. Graph Database Applications and Concepts with Neo4j. *SAIS 2013Proceedings*, 18 may 2013. Available: <http://aisel.aisnet.org/sais2013/24>.

MISSIER, Paolo; BRYANS, Jeremy; GAMBLE, Carl; CURCIN, Vasa; DANGER, Roxana. ProvAbs: Model, Policy, and Tooling for Abstracting PROV Graphs. In: INTERNATIONAL PROVENANCE AND ANNOTATION WORKSHOP (IPAW), Lecture Notes in Computer Science, 9 jun. 2014, [S.l.]: Springer International Publishing, 9 jun. 2014. p. 3–15. Available: <http://link.springer.com/chapter/10.1007/978-3-319-16462-5_1>. Accessed: 10 aug. 2016.

MISSIER, Paolo; DEY, Saumen; BELHAJJAME, Khalid; CUEVAS-VICENTTÍN, Víctor; LUDÄSCHER, Bertram. D-PROV: Extending the PROV Provenance Model with Workflow Structure. 2013, [S.l: s.n.], 2013. Available: <http://dl.acm.org/citation.cfm?id=2482949.2482961>. Accessed: 30 apr. 2014.

MISSIER, Paolo; PATON, Norman W.; BELHAJJAME, Khalid. Fine-grained and Efficient Lineage Querying of Collection-based Workflow Provenance. In: 13TH INTERNATIONAL CONFERENCE ON EXTENDING DATABASE TECHNOLOGY, EDBT ’10, 2010, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2010. p. 299–310.

MISSIER, Paolo; SAHOO, Satya S.; ZHAO, Jun; GOBLE, Carole; SHETH, Amit. Janus: From Workflows to Semantic Provenance and Linked Open Data. Lecture Notes in Computer Science, 15 jun. 2010, [S.l.]: Springer, 15 jun. 2010. p. 129–141. Available: <http://link.springer.com/chapter/10.1007/978-3-642-17819-1_16>. Accessed: 12 feb. 2016.

MISSIER, Paolo; WIJAYA, Eldarina; KIRBY, Ryan; KEOGH, Michael. SVI: A Simple Single-Nucleotide Human Variant Interpretation Tool for Clinical Use. In: INTERNATIONAL CONFERENCE ON DATA INTEGRATION IN THE LIFE SCIENCES, Lecture Notes in

Computer Science, 9 jul. 2015, [S.l.]: Springer, Cham, 9 jul. 2015. p. 180–194. Available: <https://link.springer.com/chapter/10.1007/978-3-319-21843-4_14>. Accessed: 10 oct. 2017.

MISSIER, Paolo; WOODMAN, Simon; HIDEN, Hugo; WATSON, Paul. Provenance and data differencing for workflow reproducibility analysis. *Concurrency and Computation: Practice and Experience*, p. 1–21, 1 apr. 2013.

MOREAU, Luc. Aggregation by Provenance Types: A Technique for Summarising Provenance Graphs. In: GRAPHS AS MODELS, 2015, University of Twente, Netherlands. *Anais*... University of Twente, Netherlands: [s.n.], 2015. p. 129–144. Available: <http://arxiv.org/abs/1504.02616>. Accessed: 10 aug. 2016.

MOREAU, Luc; LUDÄSCHER, Bertram; *et al.* Special Issue: The First Provenance Challenge. *Concurrency and Computation: Practice and Experience*, v. 20, n. 5, p. 409–418, 2008.

MOREAU, Luc; CLIFFORD, Ben; FREIRE, Juliana; FUTRELLE, Joe; GIL, Yolanda; GROTH, Paul; KWASNIKOWSKA, Natalia; MILES, Simon; MISSIER, Paolo; MYERS, Jim; PLALE, Beth; SIMMHAN, Yogesh; STEPHAN, Eric; DEN BUSSCHE, Jan Van. The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems*, v. 27, n. 6, p. 743–756, 2011.

MOREAU, Luc; FREIRE, Juliana; FUTRELLE, Joe; MCGRATH, Robert E.; MYERS, Jim; PAULSON, Patrick. The Open Provenance Model: An Overview. Lecture Notes in Computer Science, 2008, [S.l.]: Springer, 2008. p. 323–326. Available: <http://link.springer.com/chapter/10.1007/978-3-540-89965-5_31>.

MOREAU, Luc; MISSIER, Paolo. *PROV-DM: The PROV Data Model*. Monograph. Available: <http://www.w3.org/TR/2013/REC-prov-dm-20130430/>. Accessed: 17 feb. 2014a.

MOREAU, Luc; MISSIER, Paolo. *PROV-N: The Provenance Notation*. Monograph. Available: <http://eprints.soton.ac.uk/356852/>. Accessed: 8 feb. 2016b.

MURTA, Leonardo; BRAGANHOLO, Vanessa; CHIRIGATI, Fernando; KOOP, David; FREIRE, Juliana. noWorkflow: Capturing and Analyzing Provenance of Scripts. In: IPAW, 2014, [S.l: s.n.], 2014. p. 1–12.

NEVES, Vitor C.; BRAGANHOLO, Vanessa; MURTA, Leonardo. Implicit Provenance Gathering through Configuration Management. In: 5TH INTERNATIONAL WORKSHOP ON SOFTWARE ENGINEERING FOR COMPUTATIONAL SCIENCE AND ENGINEERING (SE-CSE), 2013, San Francisco, CA, USA. *Anais*... San Francisco, CA, USA: IEEE, 2013. p. 92–95.

NEVES, Vitor C.; DE OLIVEIRA, Daniel; OCAÑA, Kary; BRAGANHOLO, Vanessa; MURTA, Leonardo. Managing Provenance of Implicit Data Flows in Scientific Experiments. *ACM Transactions on Internet Technology*, v. to appear, 2017.

OCAÑA, Kary A. C. S.; OLIVEIRA, Daniel De; HORTA, Felipe; DIAS, Jonas; OGASAWARA, Eduardo; MATTOSO, Marta. Exploring Molecular Evolution Reconstruction Using a Parallel Cloud Based Scientific Workflow. In: SOUTO, MARCILIO C. DE; KANN, MARICEL G. (Org.). . *Advances in Bioinformatics and Computational Biology*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2012a. p. 179–191. Available: <http://link.springer.com/chapter/10.1007/978-3-642-31927-3_16>. Accessed: 18 feb. 2014.

OCAÑA, Kary A. C. S.; OLIVEIRA, Daniel De; HORTA, Felipe; DIAS, Jonas; OGASAWARA, Eduardo; MATTOSO, Marta. Exploring Molecular Evolution Reconstruction Using a Parallel Cloud-based Scientific Workflow. In: BSB, LNCS, 2012b, Berlin, Heidelberg. *Anais*... Berlin, Heidelberg: Springer, 2012. p. 179–191. Available: <http://www.springer.com/computer/ai/book/978-3-642-31926-6>.

OCAÑA, Kary; OLIVEIRA, Daniel De; OGASAWARA, Eduardo; DÁVILA, A.; LIMA, A.; MATTOSO, Marta. SciPhy: A Cloud-Based Workflow for Phylogenetic Analysis of Drug Targets in Protozoan Genomes. Lecture Notes in Computer Science, 2011, Angra dos Reis, Brazil. *Anais*... Angra dos Reis, Brazil: Springer, 2011. p. 66–70. Available: <http://link.springer.com/chapter/10.1007/978-3-642-22825-4_9>. Accessed: 14 mar. 2014.

OGASAWARA, Eduardo; DIAS, Jonas; SILVA, Vítor; CHIRIGATI, Fernando; OLIVEIRA, Daniel; PORTO, Fabio; VALDURIEZ, Patrick; MATTOSO, Marta. Chiron: A Parallel Engine for Algebraic Scientific Workflows. *CCPE*, 00002, v. 25, n. 16, p. 2327–2341, 2013.

OINN, Tom; ADDIS, Matthew; FERRIS, Justin; MARVIN, Darren; SENGER, Martin; GREENWOOD, Mark; CARVER, Tim; GLOVER, Kevin; POCOCK, Matthew R.; WIPAT, Anil; LI, Peter. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, v. 20, n. 17, p. 3045–3054, 2004.

OLIVEIRA, Wellington; DE OLIVEIRA, Daniel; BRAGANHOLO, Vanessa. Provenance Analytics for Workflow-based Computational Experiments: a Survey. *ACM Computing Surveys*, n. 1, p. 1–29, 2018.

OLIVEIRA, Wellington; MISSIER, Paolo; OCAÑA, Kary; OLIVEIRA, Daniel De; BRAGANHOLO, Vanessa. Analyzing Provenance Across Heterogeneous Provenance Graphs. In: MATTOSO, MARTA; GLAVIC, BORIS (Org.). . *Provenance and Annotation of Data and Processes*. Lecture Notes in Computer Science. [S.l.]: Springer International Publishing, 2016. p. 57–70. Available: <http://link.springer.com/chapter/10.1007/978-3-319-40593-3_5>. Accessed: 1 aug. 2016.

PRABHUNE, Ajinkya. *Generic and Adaptive Metadata Management Framework for Scientific Data Repositories*. 2018. 177 f. Doctoral dissertation – Heidelberg University, Germany, 2018.

PRABHUNE, Ajinkya; ZWEIG, Aaron; STOTZKA, Rainer; HESSER, Jürgen; GERTZ, Michael. P-PIF: a ProvONE provenance interoperability framework for analyzing heterogeneous workflow specifications and provenance traces. *Distributed and Parallel Databases*, p. 1–46, 11 dec. 2017.

PRUD'HOMMEAUX, Eric; SEABORNE, Andy. *SPARQL Query Language for RDF*. Available: <http://www.w3.org/TR/rdf-sparql-query/>. Accessed: 21 jan. 2014.

QUAN, D. A.; KARGER, R. How to Make a Semantic Web Browser. WWW '04, 2004, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2004. p. 255–265. Available: <http://doi.acm.org/10.1145/988672.988707>. Accessed: 19 aug. 2016.

RAGAN, E. D.; ENDERT, A.; SANYAL, J.; CHEN, J. Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE Transactions on Visualization and Computer Graphics*, v. 22, n. 1, p. 31–40, jan. 2016.

RIO, Nicholas Del; SILVA, Paulo Pinheiro Da; PORRAS, Hugo. Browsing Proof Markup Language Provenance: Enhancing the Experience. In: MCGUINNESS, DEBORAH L.; MICHAELIS, JAMES R.; MOREAU, LUC (Org.). . *Provenance and Annotation of Data and Processes*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2010. p. 274–276. Available: <http://link.springer.com/chapter/10.1007/978-3-642-17819-1_31>. Accessed: 11 aug. 2016.

ROURE, David De; BELHAJJAME, Khalid; MISSIER, Paolo; MANUEL, José; PALMA, Raul; RUIZ, José Enrique; HETTNE, Kristina; KLYNE, Graham; ROOS, Marco; GOBLE, Carole. Towards the preservation of scientific workflows. In: INTERNATIONAL CONFERENCE ON PRESERVATION OF DIGITAL OBJECTS, nov. 2011, [S.l.]: ACM, nov. 2011. p. 1–4.

SAHOO, Satya S.; WEATHERLY, D. Brent; MUTHARAJU, Raghava; ANANTHARAM, Pramod; SHETH, Amit; TARLETON, Rick L. Ontology-Driven Provenance Management in eScience: An Application in Parasite Research. In: OTM CONFEDERATED INTERNATIONAL CONFERENCES "ON THE MOVE TO MEANINGFUL INTERNET SYSTEMS", Lecture Notes in Computer Science, 1 nov. 2009, [S.l.]: Springer, Berlin, Heidelberg, 1 nov. 2009. p. 992–1009. Available: <https://link.springer.com/chapter/10.1007/978-3-642-05151-7_18>. Accessed: 21 nov. 2017.

SCHEIDEGGER, Carlos; KOOP, David; SANTOS, Emanuele; VO, Huy; CALLAHAN, Steven; FREIRE, Juliana; SILVA, Cláudio. Tackling the Provenance Challenge one layer at a time. *Concurrency and Computation: Practice and Experience*, v. 20, n. 5, p. 473–483, 2008.

SCHREIBER, Andreas; STRUMINSKI, Regina. Visualizing Provenance using Comics. In: TAPP 2017, 2017, Seattle, WA. *Anais*... Seattle, WA: USENIX Association, 2017.

SELTZER, Margo I.; ANGELINO, Elaine Lee; BRAUN, Uri Jacob; HOLLAND, David A.; MACKO, Peter; MARGO, Daniel Wyatt. Provenance Integration Requires Reconciliation. In: TAPP, 2011, Heraklion, Crete, Greece. *Anais*... Heraklion, Crete, Greece: [s.n.], 2011. Available: <http://dash.harvard.edu/handle/1/5168853>. Accessed: 22 mar. 2015.

SELTZER, Margo I.; MACKO, Peter. Provenance Map Orbiter: Interactive Exploration of Large Provenance Graphs. Engineering and Applied Sciences, 2011. Accessed: 1 dec. 2013.

SHANNON, Paul; MARKIEL, Andrew; OZIER, Owen; BALIGA, Nitin S.; WANG, Jonathan T.; RAMAGE, Daniel; AMIN, Nada; SCHWIKOWSKI, Benno; IDEKER, Trey. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, v. 13, n. 11, p. 2498–2504, nov. 2003.

SHAPIRO, S. S.; WILK, M. B. An Analysis of Variance Test for Normality (Complete Samples). *Biometrika*, v. 52, n. 3/4, p. 591–611, 1965.

SILVA, C. T.; FREIRE, J. Software Infrastructure for exploratory visualization and data analysis: past, present, and future. *Journal of Physics: Conference Series*, v. 125, n. 1, p. 012100, 2008.

SILVA, C.T.; FREIRE, J.; CALLAHAN, S.P. Provenance for Visualizations: Reproducibility and Beyond. *Computing in Science Engineering*, v. 9, n. 5, p. 82–89, 2007.

SIMMHAN, Y.L.; PLALE, B.; GANNON, D. A Framework for Collecting Provenance in Data-Centric Scientific Workflows. 2006, [S.l: s.n.], 2006. p. 427–436. Available: <10.1109/ICWS.2006.5>. Accessed: 19 jul. 2010.

SIMMHAN, Yogesh L.; PLALE, Beth; GANNON, Dennis. A Survey of Data Provenance in e-Science. *ACM SIGMOD Record*, v. 34, n. 3, p. 31–36, 2005.

SIMMHAN, Yogesh L.; PLALE, Beth; GANNON, Dennis. Query capabilities of the Karma provenance framework. *Concurrency and Computation: Practice and Experience*, v. 20, n. 5, p. 441–451, 2008.

SIMMHAN, Yogesh L.; PLALE, Beth; GANNON, Dennis; MARRU, Suresh. Performance Evaluation of the Karma Provenance Framework for Scientific Workflows. In: MOREAU, LUC; FOSTER, IAN (Org.). . *Provenance and Annotation of Data*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2006. p. 222–236.

STAMATAKIS, A. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, v. 22, n. 21, p. 2688–2690, 1 nov. 2006.

STITZ, Holger; LUGER, Stefan; STREIT, Marc; GEHLENBORG, Nils. AVOCADO: Visualization of Workflow-Derived Data Provenance for Reproducible Biomedical Research. *bioRxiv*, p. 044164, 18 mar. 2016.

SUN, Peng; LIU, Ziyang; DAVIDSON, Susan B.; CHEN, Yi. Detecting and Resolving Unsound Workflow Views for Correct Provenance Analysis. SIGMOD '09, 2009A, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2009A. p. 549–562.

SURIARACHCHI, Isuru; ZHOU, Quan; PLALE, Beth. Komadu: a capture and visualization system for scientific data provenance. *Journal of Open Research Software*, v. 3, n. 1, 2015. Available: <http://openresearchsoftware.metajnl.com/article/10.5334/jors.bq/>. Accessed: 26 aug. 2016.

TAYLOR, Ian J.; DEELMAN, Ewa; GANNON, Dennis B.; SHIELDS, Matthew. *Workflows for e-Science: Scientific Workflows for Grids*. 1. ed. [S.l.]: Springer, 2014.

VIEGAS, F.B.; WATTENBERG, M.; VAN HAM, F.; KRISS, J.; MCKEON, M. ManyEyes: a Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics*, v. 13, n. 6, p. 1121–1128, 2007.

WATSON, Paul; HIDEN, Hugo; WOODMAN, Simon. e-Science Central for CARMEN: Science As a Service. *Concurr. Comput. : Pract. Exper.*, v. 22, n. 17, p. 2369–2380, dec. 2010.

WILCOXON, Frank. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, v. 1, n. 6, p. 80–83, 1945.

WOODMAN, Simon; HIDEN, Hugo; WATSON, Paul; MISSIER, Paolo. Achieving Reproducibility by Combining Provenance with Service and Workflow Versioning. WORKS '11, 2011, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2011. p. 127–136. Available: <http://doi.acm.org/10.1145/2110497.2110512>. Accessed: 9 mar. 2014.

ZHAO, Jing; SUN, Fan; TORNIAI, Carlo; BAKSHI, Amol; PRASANNA, Viktor. A provenance-integration framework for distributed workflows in Grid environments. In:

WORKSHOP ON GRID AND UTILITY COMPUTING, 2008, Cochin, India. *Anais*... Cochin, India: [s.n.], 2008. p. 17–20.

ZHAO, Jun; GOBLE, Carole; STEVENS, Robert; TURI, Daniele. Mining Taverna's semantic web of provenance. *Concurrency and Computation: Practice and Experience*, v. 20, n. 5, p. 463–472, 2008.

ZHAO, Jun; WROE, Chris; GOBLE, Carole; STEVENS, Robert; QUAN, Dennis; GREENWOOD, Mark. Using Semantic Web Technologies for Representing E-science Provenance. In: MCILRAITH, SHEILA A.; PLEXOUSAKIS, DIMITRIS; HARMELEN, FRANK VAN (Org.). . *The Semantic Web – ISWC 2004*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2004. p. 92–106. Available: <http://link.springer.com/chapter/10.1007/978-3-540-30475-3_8>. Accessed: 18 aug. 2016.

ZHAO, Yong; HATEGAN, M; CLIFFORD, B; FOSTER, I; VON LASZEWSKI, G; NEFEDOVA, V; RAICU, I; STEF-PRAUN, T; WILDE, M. Swift: Fast, Reliable, Loosely Coupled Parallel Computation. 2007, Salt Lake City, USA. *Anais*... Salt Lake City, USA: [s.n.], 2007. p. 206, 199. Available: <http://dx.doi.org/10.1109/SERVICES.2007.63>. Accessed: 18 feb. 2010.

ZHAO, Yong; WILDE, Michael; FOSTER, Ian. Applying the Virtual Data Provenance Model. In: MOREAU, LUC; FOSTER, IAN (Org.). . *Provenance and Annotation of Data*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2006. p. 148–161. Available: <http://link.springer.com/chapter/10.1007/11890850_16>. Accessed: 30 aug. 2016.

APENDIX A – PERSONAL PROFILE FORM

1. Personal Data

   Name _____

   E-mail_____

2. Academic Degree

   ( ) PhD

   ( ) PhD student

   ( ) Master

   ( ) Master student

   ( ) Undergraduate

   ( ) Undergraduate student

   Date of course start __/__/____

   Date of course completion (or expectation for the course completion) __/__/____

3. Experience

   3.1.How many years of experience in development for each type of software project bellow do you have?

	None	1 - 2 years	3 - 4 years	5 - 6 years	More than 6 years
Personal					
Academic					
Open Source					
Company					

   3.2.How many years of experience in Prolog development do you have?

   ( ) None

   ( ) 6 months

   ( ) 1-2 years

   ( ) 3-4 years

   ( ) More than 4 years

   3.3. Please, check your experience level in Prolog language study and/or usage?

a) None
b) I have studied and practiced at school
c) I have studied and practiced in short courses
d) I have studied and practiced by myself searching in books and in the internet
e) I have used in company projects
f) I have used in personal projects

4. Experiment

4.1. Whether you profile fill the desired requirements for the experiment, would you like to participate in it?

The experiment will take about 1 hour and 30 minutes and will take place at UFFJ/UFF/IFSEMG Computer Science lab in the first week of June of 2017.

( ) Yes
( ) No

4.2. Whether your answered was “yes” to the previous question, please, check your all available schedules to participate in the experiment.

( ) Monday morning
( ) Monday afternoon
( ) Monday night
( ) Tuesday morning
( ) Tuesday afternoon
( ) Tuesday night
( ) Wednesday morning
( ) Wednesday afternoon
( ) Wednesday night
( ) Thursday morning
( ) Thursday afternoon
( ) Thursday night
( ) Friday morning
( ) Friday afternoon
( ) Friday night