Knowledge and Information Systems

A Systematic Review of Provenance Systems

Beatriz Pérez[1], Julio Rubio[1] and Carlos Sáenz-Adán[1]

[1]Department of Mathematics and Computer Science, University of La Rioja, La Rioja, Spain

Abstract. Provenance refers to the entire amount of information, comprising all the elements and their relationships, that contribute to the existence of a piece of data. The knowledge of provenance data allows a great number of benefits such as verifying a product, result reproductivity, sharing and reuse of knowledge, or assessing data quality and validity. With such tangible benefits, it is no wonder that in recent years, research on provenance has grown exponentially, and has been applied to a wide range of different scientific disciplines.

Some years ago, managing and recording provenance information were performed manually. Given the huge volume of information available nowadays, the manual performance of such tasks is no longer an option. The problem of systematically performing tasks such as the understanding, capture and management of provenance has gained significant attention by the research community and industry over the past decades. As a consequence, there has been a huge amount of contributions and proposed provenance systems as solutions for performing such kinds of tasks.

The overall objective of this paper is to plot the landscape of published systems in the field of provenance, with two main purposes. First, we seek to evaluate the desired characteristics that provenance systems are expected to have. Second, we aim at identifying a set of representative systems (both early and recent use) to be exhaustively analyzed according to such characteristics. In particular, we have performed a systematic literature review of studies, identifying a comprehensive set of 105 relevant resources in all.

The results show that there are common aspects or characteristics of provenance systems thoroughly renowned throughout the literature on the topic. Based on these results, we have defined a six-dimensional taxonomy of provenance characteristics attending to: *general aspects*, *data capture*, *data access*, *subject*, *storage*, and *non–functional aspects*. Additionally, the study has found that there are 25 most referenced provenance systems within the provenance context. This study exhaustively analyzes and compares such systems attending to our taxonomy, and pinpoints future directions.

Received Dec 17, 2016
Revised Dec 15, 2017
Accepted Dec 30, 2017

Keywords: provenance systems; provenance aspects; computer science; systematic review

1. Introduction

The increasing amount of accessible data, together with the constantly speeding up of information flow induced by the Internet or within organizations, is increasing the interest for detailed information about the origin of data, in order to assess its quality or validity. Assessing both the quality and validity of a piece of data typically involves examining the context in which the data appears, as well as determining its original sources or reviewing the process through which it was generated. However, this task could become rather cumbersome when dealing with digital data which might have been generated by complex successive transformations together with derivations from numerous sources, possible over long periods of time [1].

1.1. State of the Art in Provenance

The term *provenance* is commonly used synonymously with the word *lineage* to refer to the source or origins of an object or piece of data. More specifically, provenance refers to the entire amount of information comprising all the elements and their relationships that contribute to the existence of a piece of data. Thus, the knowledge of provenance not only includes aspects such as sources and processing steps, but also dependencies and contextual information [1]. In this context, two basic types of information are usually considered to form the provenance of a data item. First, the *source* provenance refers to the information about the data that was involved in the creation of a data item [2, 3], that is, this type of provenance is related to the origin or source of the data. Second, *transformation* provenance refers to the information about the transformations that were involved in the creation of a data item, that is, this type of provenance identifies how the data was created or derived [2, 3].

In its origins, the term provenance was mainly used to refer to the history of ownership of works of art, which helped to determine the value of the piece. At present, it is used in a wide variety of application contexts. More specifically, in recent years research on provenance has grown exponentially [4]. This subject area is vast and affects a range of different scientific disciplines, with a special focus on the field of computer science [4, 5]. In this field, provenance is usually applied to elements generated by scientific applications, considering from scientific results to data products in general. Provenance allows users to verify a data product, to determine its authorship, or to infer its quality. In the particular case of scientific results, provenance is used as the proof of correctness of the result, providing the means to interpret and understand it; to analyze the process of steps that lead to the result; to ensure that the experiment was performed through quality control procedures, and even reproduce it [3, 4, 6]. Considering these potential benefits, it is no wonder that provenance is consequently of paramount and increasing importance. In particular, the growing interest in this topic is evidenced by its use in a wide number of areas within the computer science field: e-science, data warehousing, semantic web, curated databases, Linked Data, or Big Data [4].

While some years ago managing and recording provenance information was performed manually, the large amount of provenance information complicates the manual performance of such tasks [6]. Scientists and engineers need to expend substantial effort both managing data and recording provenance information, to answer basic questions such as: who created a specific data product and when?, who modified it and when?, what process(es) created the data product?, which data products were derived from this data product? [6, 7]. Not only does that process become time-consuming and cumbersome, but it is also error-prone [7].

The systematic performance of tasks such as the understanding, capture and management of provenance information has received recently significant attention [6, 7] in a wide range of domains and applications (such as biology, business and financial sectors, or computer science). There have been a wide number of proposals published in the literature ranging from techniques developed with the aim of addressing different aspects of the emerging problems, to concrete provenance management systems, which adopt such techniques to varying extents. Different definitions of a provenance management system have been given depending on the perspective adopted by researchers. For example, generally speaking, a *data management system* must provide users with unifying mechanisms mainly for storing, retrieving, preserving, and manipulating data. Each type of data management system focuses on a different aspect, and provides specific mechanisms for data and metadata manipulation [8]. If we focus on a *provenance management system*, or simply, a *provenance system*, Glavic [9] defines it as a practical system that supports generating, querying and storing provenance information. Groth et al. [10] consider it as “a computer system that deals with all issues pertaining to the recording, maintaining, visualising, reasoning and analysis of the documentation of process that underpins the notion of provenance”.

1.2. Motivation for a Systematic Review in Provenance Domain

As discussed previously, in recent years there has been a rapid growth of the provenance field, in general, and of provenance solutions, in particular, which has derived into a large and heterogeneous research corpus of approaches to address a variety of provenance concerns. Even so, at the present time there appears to be no clear consensus or common ground on aspects such as what requirements a provenance system should support, or what technical details are involved in making these systems possible [1, 7]. Among such corpus of research, a non-negligible number of surveys have touched on the issue of provenance, distinguishing mainly among those which focus on aspects within scientific data processing and scientific workflow management systems (SWfMS) ([6, 7, 11, 12, 13]), and those within database management systems (DBMS) ([2, 3, 14, 15, 16]). Other works focus on neither SWfMS nor DBMS, providing aspects of provenance in general ([1, 4, 5, 17, 18, 19, 20, 21, 22]). Although all these surveys provide to some extent an overview of research issues in provenance, they tackle provenance issues from a different perspective, even using different terminology to refer to the same concept. Only a few of these works explicitly present some kind of categorization scheme or taxonomy for provenance characteristics ([2],

[12], [13], [18]), making it difficult to provide a comparison and to identify equivalent provenance aspects among these surveys. Although these works identify a significant number of dimensions of comparison, they miss tackling aspects such as security, reproducibility or provenance interoperability. Some of these surveys apply their classifications to specific provenance systems, being [20] the most complete proposal comparing just 13 systems among the wide number of existing solutions.

Given the size and heterogeneity of the literature, and taking into account that to the best of our knowledge existing surveys not only are restricted to specific provenance issues but they have also been conducted neither with an appropriate level of formality nor in a systematic way, we have seen the need to organize and synthesize the existent research corpus, by performing a systematic review [23, 24, 25]. A systematic review is a means of identifying, evaluating and interpreting all available research relevant to a particular phenomenon of interest in a thorough and unbiased manner [23]. In contrast to a usual survey, it follows a controlled, rigorous and auditable procedure for searching, collecting and selecting the acceptable quality published studies to be considered in the review. Another difference from surveys is that the process followed in systematic reviews is formally documented and hence repeatable.

The systematic review herein presented comes with the main aim of providing a definitely, broad overview of research issues in provenance systems, and identifying not only the different features that such systems are expected to have, but also the techniques or methods proposed to address such characteristics. We also aim at providing a broad overview of the most representative provenance systems within the computer science area, identifying the techniques they propose to tackle the features from those we identified. Thus, our main goals justify several of the motivations for performing systematic reviews such as describing and organising the state-of-the-art in an area [26], learning from studies and getting knowledge [26], or identifying the main research techniques used in a research topic or field [27].

1.3. Contribution

The contributions of this work are thus twofold. First, based on a comprehensive, systematic overview covering both early and recent work in the area, we have defined a unified taxonomy of provenance systems characteristics. This taxonomy encompasses the overall classifications considered by other works, and more fully captures the unique characteristics of provenance systems. Our approach, in contrast to other proposals, is limited neither to a specific area of data management nor to a domain of application. Additionally, given the wide variety of terms and concepts used in the literature for provenance related issues, not only do we provide the reader with a complete and consistent background of provenance concepts and techniques, but also we relate them to the terminology used by other researchers.

Second, we have identified existing provenance systems considering not only those which have been in wide use for a long time but also those which have emerged recently. Given the wide number of provenance systems published in the literature (we identify 251 in this review), we cannot provide a comprehensive coverage of all systems due to space reasons. We do review a representative set of systems which we have considered as the most representative or influential. More specifically, based on our taxonomy, we have performed an exhaustive analysis and comparison of 25 systems, which illustrates different solutions covering a broad spectrum of alternatives along the aspects identified in our taxonomy. As a result, not only do we review a more complete set of provenance solutions than the analysed in previous works but also, since our taxonomy covers a more complete spectrum of provenance aspects, we encompass a broader analysis and overview of such systems, highlighting future directions in the provenance area.

Our work is thus intended to provide several benefits. First, our taxonomy and review of related background aim at informing and enhancing the understanding of the field to potential researchers, software developers or provenance technology users. Second, this work can enable general users to distinguish between different perspectives of provenance and guide them in their decision towards the selection of the most suitable solution for their needs. Third, our target readership could also be those potential researchers and software developers who are interested in up-to-date approaches available, as well as open problems being seen in practice. Last but not least, the results provided by this review could be particularly relevant for potential researchers aimed at identifying provenance research issues that have been already tackled or directions for future research.

The paper is structured as follows: Section 2 briefly describes the methodology followed to perform this review. Results are presented in Section 3. Section 4 discusses the main findings of the review and identifies future directions in the provenance area. Section 5 concludes the paper.

2. Research Method

As explained previously, the study presented in this paper has been undertaken as a systematic review based on the original guidelines as described in [23, 24, 25]. More specifically, the process we have followed has consisted of three stages: *planning* the review, *conducting* the review, and *reporting* the results of the review. For the sake of brevity, next we provide some details regarding the *planning* and the *conducting* stages, but the interested reader is referred to [28] for a complete description of such stages. Later we devote Section 3 to describe the results of the *reporting* stage.

2.1. Planning

Firstly, the *planning* stage mainly deals with identifying the need for the review, and establishing a review protocol. Regarding the interest for the review, we firstly performed preliminary searches aimed not only at identifying existing systematic reviews or surveys, but also assessing the volume of potentially relevant studies. This initial informal search confirmed that there were a substantial number of papers on the topic (none of them performed as a systematic review), and that, as described previously, a systematic review would be appropriate. Concerning a review protocol, it states an accurate and reliable methodology to perform the review, specifying: research questions to be addressed, inclusion and exclusion criteria, search and selection strategy, quality assessment and data extraction, and the data synthesis.

Taking this into account, after performing the pilot search, we developed the following research questions:

RQ1 What are the different aspects to take into account in provenance systems?

RQ2 What are the proposed techniques to address such different aspects?

RQ3 Taking into account the most representative provenance systems within the provenance field, how these systems have addressed the identified aspects?

To make sure that the studies included in the review were clearly related to the research topic, we defined detailed general guidelines for inclusion and exclusion criteria. In particular, the scope of this review is limited to the literature that: (i) presents or describes solutions for research in provenance systems within the computer science context, and/or (ii) surveys or analyzes the previous type of studies. We did not impose any restrictions on a specific domain of application. We excluded pure discussion or opinion papers, tutorials and any study that tackles provenance in a context other than the computer science field. We also exclude any study reported in a language other than English.

2.2. Conducting

Secondly, after *planning* the review, we started the *conducting* stage by following our review protocol. This stage mainly consisted of: performing the search strategy, identifying the relevant studies taking into account the research questions, carrying out the quality assessment to analyze and assess such studies, accomplishing the data extraction and, performing the data synthesis by collating and summarizing the results of the included primary studies. The *conducting* stage, as well as the *reporting* stage, was carried out by performing two review processes. The first review process was conducted from September 2015 to June 2016, and it included research studies published up to and including December 2015. This review was renewed by performing a second review process which took place between July and August 2017, and which covered research studies published from January 2016 to July 2017. Thus, this systematic review covers studies published up to and including July 2017.

During the search process in both review processes, we paid special attention to published surveys and reviewing papers (from now on we refer to these types of studies as simply *surveys*) on the topic, since we consider them as potentially relevant for the study. Additionally, we looked for relevant conference/workshop proceedings and journals in which possible candidate papers were published. In particular, we identified four conference/workshop proceedings and journals: (1) the biennial International Provenance & Annotation Workshop (IPAW), (2) the annual workshop on Theory and Practice of Provenance (TaPP), (3) the Future Generation Computer Systems (FGCS) journal, and (4) several ACM SIGMOD resources (including the SIGMOD Record journal, the ACM SIGMOD International Conference on Management of Data (MOD), and the ACM SIGMOD Symposium on Principles of Database Systems (PODS)).

From the search and selection steps performed during both review processes, we identified a total of 251 provenance systems, being 25 by far the most referenced for their repercussion and consolidation within the provenance context. Thus, we decided to apply the selection filter of the papers previously identified, looking for those which (i) specially tackled the chosen systems, or (ii) perform any type of quality analysis of the systems (such as surveys). As a result, we selected a total of 105 papers classified into two groups attending to (1) those

Table 1. Selected studies regarding the concrete systems

Id	System name	Authors' Rep.Nam.	References
S1	VisTrails	J. Freire et al.	[29, 30, 31, 32, 33]
S2	myGrid/Taverna	K. Wolstencroft et al.	[34, 35, 36, 37]
S3	Kepler provenance	I. Altintas et al.	[38, 39]
S4	PASS	D. A. Holland et al.	[40, 41, 42]
S5	Trio	J. Widom et al.	[43, 44, 45, 46, 47, 48]
S6	Karma	L.Y. Simmhan et al.	[49, 50, 51, 52]
S7	Chimera	I. Foster et al.	[53, 54, 55]
S8	ZOOM	O. Biton et al.	[56, 57, 58, 59]
S9	Cui 2000	Y. Cui et al.	[60, 61, 62, 63]
S10	Swift	I. Foster et al.	[64, 65, 66, 67]
S11	PASOA/PreServ	P. Groth et al.	[68]
S12	Tioga	M. Stonebraker et al.	[69, 70, 71]
S13	Wings-Pegasus	E. Deelman et al.	[72, 73, 74, 75, 76, 77, 78]
S14	Buneman	P. Buneman et al.	[79]
S15	SPADE	A. Gehani et al.	[80, 81, 82, 83, 84]
S16	ORCHESTRA	Z.G. Ives et al.	[85, 86, 87, 88, 89]
S17	Perm-GProM	B. Glavic et al.	[9, 90, 91, 92, 93, 94, 95, 96, 97, 98]
S18	ES3	J. Frew et al.	[99, 100]
S19	COMAD	S. Bowers et al.	[101]
S20	DBNotes	L. Chiticariu et al.	[102, 103]
S21	LipStick	Y. Amsterdamer et al.	[104]
S22	REDUX	R. Barga et al.	[105]
S23	BURRITO	P.J. Guo et al.	[106] [107]
S24	PLUS	B. Blaustein et al.	[108, 109, 110, 111]
S25	RAMP	R. Ikeda	[112, 113]

which refer to surveys (19 papers), and (2) those that tackle any of the concrete provenance systems classified for our review (86 papers).

3. Results

This section describes the analysis of the data extracted from the selected papers. First, in Subsection 3.1 we present general aspects of our 105 primary studies. Second, in Subsection 3.2 we answer the research questions *RQ1* and *RQ2* by providing a taxonomy of provenance systems features, together with general background regarding the techniques tackled to address such features. Finally, in Subsection 3.3 we answer the research question *RQ3* by synthesizing the data from all the selected papers according to our taxonomy.

3.1. Description of Primary Studies

As explained previously, the selected papers can be classified into two groups. The first group of papers, which refer to surveys, are: Su1 [1], Su2 [4], Su3 [17], Su4 [14], Su5 [15], Su6 [16], Su7 [7], Su8 [11], Su9 [6], Su10 [2], Su11 [12], Su12 [18], Su13 [3], Su14 [5], Su15 [13], Su16 [19], Su17 [20], Su18 [22], and Su19 [21]. In particular, these studies were published in a range of years from 2001 to 2016. In relation to the publication type, 10 papers were published in journals, while 7 papers came from conference/workshop proceedings and other 2 were published as technical reports. As for the second group of papers, which tackle any of the concrete 25 provenance systems chosen in the selection process, they have

Table 2. Comparative of selected surveys

Comparison aspect / Survey	Area of study	Explicit taxonomy or categorization scheme	Analized systems	Dimensions and aspects included in the proposed taxonomy																							
				General Aspects					Data Capture				Data Access		Subject						Storage				Non-functional requirements		
				Data processing category	Application domain	Purpose	Extended/ Intended	Availability	Tracing	Levels	Mechanism	Technique	Accessing	Querying	Contents	Abstraction	Interoperability/ Exchange	Phase	Orientation	Granularity	Scalability	Coupling	Persistence	Archiving	Security/Privacy	Repeata./Repla. /Reproduci.	Verification
Su1	not spec.		8							X				X						X					X		
Su2	not spec.													X	X	X											
Su3	not spec.			X																							
Su4	DBMS														X												
Su5	DBMS											X			X					X							
Su6	DBMS								X						X												
Su7	SWfMS						X			X								X	X								
Su8	SWfMS												X														
Su9	SWfMS		10					X		X				X									X		X		
Su10	DBMS	X	6						X	X		X	X	X	X							X			X		
Su11	SWfMS	X	5	X	X	X				X		X	X	X	X				X	X	X		X				
Su12	not spec.	X	9	X	X	X			X	X		X	X	X	X				X	X	X		X				
Su13	DBMS											X			X					X							
Su14	not spec.																X			X			X		X		
Su15	SWfMS	X	11						X	X	X	X	X	X				X	X	X	X	X	X	X			
Su16	not spec.								X			X	X	X			X				X				X		
Su17	not spec.		13		X	X						X	X	X	X					X	X						
Su18	not spec.																								X		
Su19	not spec.		7							X			X		X	X										X	

been classified depending on the provenance system they tackle (see Table 1). In this table we have considered for each system: (i) a unique identifier used from now on to refer to each system (*Id*), (ii) the system name (*System name*), (iii) a representative authors' name of all papers published by the same authors describing similar research work on the same provenance system (*Authors' Rep. Nam.*), and (iv) the bibliographic references of the papers (*References*). As for the place of publication of this second group of papers, 24 were published in journals, 53 came from conference/workshop proceedings, 2 came from book chapters, 2 came from PhD Thesis, and other 5 were published as technical reports.

3.2. Taxonomy of Provenance Systems Features

Among the information extracted from the selected papers, we note that the vast majority was gathered from the identified surveys. In Table 2 we compare the different surveys along with a set of dimensions. As explained previously, the majority of work made on surveying provenance issues has been undertaken within SWfMS systems (5 papers), and DBMS systems (5 papers), being other 9 surveys which focus on neither SWfMS nor DBMS, providing aspects of provenance in general (see column *Area of study* in Table 2).

As stated in column *Explicit taxonomy or categorization scheme*, only a few of the selected surveys provide some kind of categorization scheme or taxonomy for provenance characteristics (in particular, [Su10], [Su12], [Su11], which is based on the taxonomy presented by the same authors in [Su12], and [Su15], based on [Su11]). These surveys apply their classifications to current research efforts on specific provenance systems. Other works do not present a taxonomy, but they compare different systems in base on a set of arbitrary provenance items (see column *Analyzed systems* in Table 2). While some of these works form a valuable, comprehensive overview of existing approaches, to the best of our knowledge, they fail in establishing those classifications based on an analysis of other works. Additionally, none of the surveyed works that compare provenance systems present the reasons why they have chosen such systems.

Finally, we have defined our taxonomy taking as a starting point the taxon-

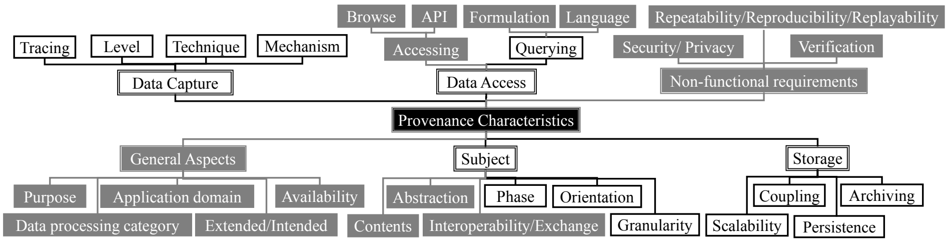


Fig. 1. Taxonomy of Provenance Systems Features

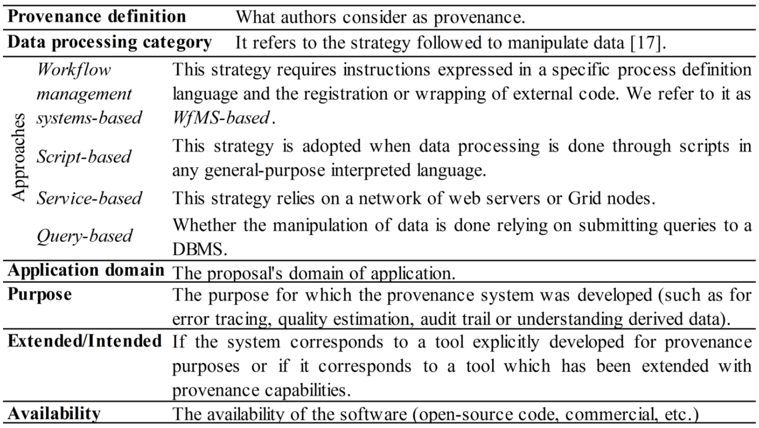
omy proposed in [Su15], since we considered it the most complete one. We have also analysed the remainder works, matching common aspects and incorporating additional ones not considered in [Su15]. In Figure 1 we present the resulted taxonomy in which we have depicted in white colour aspects common to the taxonomy presented in [Su15], reserving grey colour either to new aspects or to aspects which have been modified from others given by [Su15]. The taxonomy establishes six dimensions for comparing provenance systems, attending to:

- *general aspects*, which describes general background regarding provenance systems.
- *data capture*, which refers to the way in which provenance data can be captured on the existing provenance systems.
- *data access*, which refers to how users can access provenance data repositories.
- *subject*, which refers to the different subjects or levels of detail in which provenance data can be represented into, also considering interoperability aspects.
- *storage*, which describes the different approaches used by provenance systems to register provenance information.
- *non–functional requirements*, refers to non–functional requirements of provenance systems, such as security.

Additionally, in each dimension we have identified several categories of comparison which would correspond to concrete aspects to focus on when contrast provenance systems (for example, regarding *data capture* in Figure 1, we can distinguish among *tracing*, *level*, *technique*, and *mechanism*). Thus, these dimensions together with their categories or aspects would constitute a response to question *RQ1*. Aimed at comparing our taxonomy with the ones presented in the analyzed surveys, in Table 2 we also present the dimensions and aspects tackled by such surveys from those included in our taxonomy. We show that there is no survey which encompasses all the dimensions as a whole, being scarce the number of aspects tackled by the surveys. Thus, we provide a unified taxonomy which encompasses the overall classifications given by other proposals, and which more fully captures the unique characteristics of provenance systems.

Next, based on the information obtained from our data extraction process, we have identified the techniques adopted by each approach to tackle the aspects included in our taxonomy. Given the wide number of terms and fundamental techniques commonly used in such solutions, we have decided to provide a review of general background to give the reader with the necessary knowledge to follow the taxonomy. We have also unified the different notions distinct authors use to describe the same concepts or similar strategies. The identification of

Table 3. Data extracted regarding *General Aspects* dimension



Provenance definition		What authors consider as provenance.
Data processing category		It refers to the strategy followed to manipulate data [17].
Approaches	*Workflow management systems-based*	This strategy requires instructions expressed in a specific process definition language and the registration or wrapping of external code. We refer to it as *WfMS-based*.
	Script-based	This strategy is adopted when data processing is done through scripts in any general-purpose interpreted language.
	Service-based	This strategy relies on a network of web servers or Grid nodes.
	Query-based	Whether the manipulation of data is done relying on submitting queries to a DBMS.
Application domain		The proposal's domain of application.
Purpose		The purpose for which the provenance system was developed (such as for error tracing, quality estimation, audit trail or understanding derived data).
Extended/Intended		If the system corresponds to a tool explicitly developed for provenance purposes or if it corresponds to a tool which has been extended with provenance capabilities.
Availability		The availability of the software (open-source code, commercial, etc.)

these techniques constitutes a response to question *RQ2*. In particular, the different aspects included in our *six-dimensional* taxonomy are presented in Tables from 3 to 8. The techniques commonly used to tackle each aspect are presented under the "Approaches" section in these tables (only when the number of techniques was bounded or clearly identified). These tables are also supported by additional comments including, when applicable, advantages and disadvantages of the approaches.

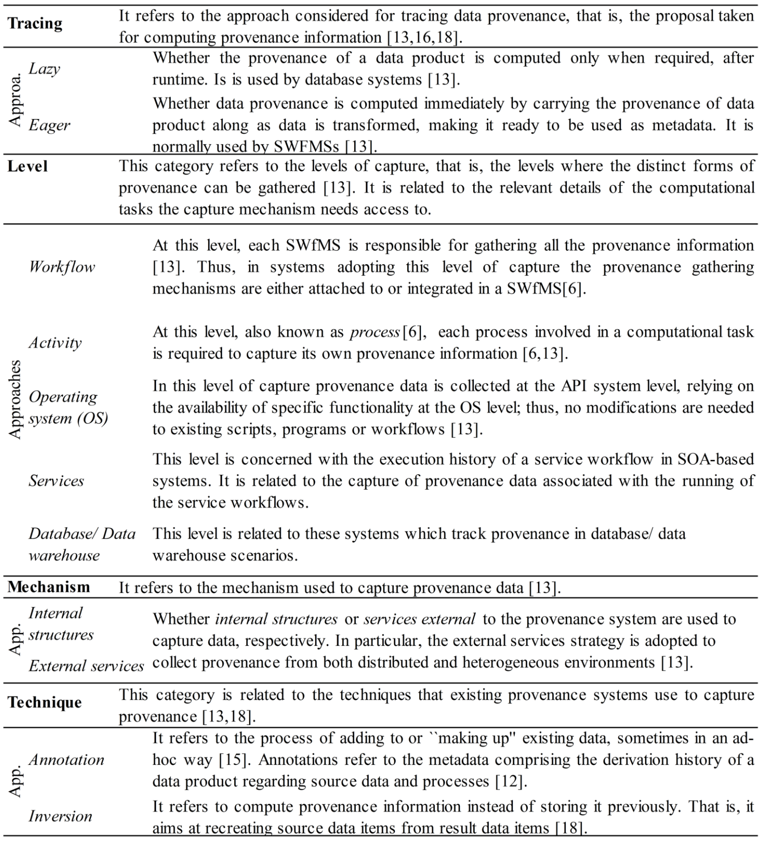
3.2.1. General Aspects *dimension*

As we show in Table 3, this dimension just identifies general characteristics of the provenance systems. In particular, the *provenance definition* aspect aims at identifying the definitions of *provenance* considered by the different authors in order to be analyzed by means of the technique called *References-enriched Concept Maps* (RCM) [114]. This technique, inspired by Concept Maps, can be used to compare definitions and therefore improve the understanding of terms, keeping track of the publications in which the different definitions were proposed. Later, the RCM can be automatically created using the tool Sáenz presented with García-Izquierdo et al. in [115]. As we will show in Section 3.3 "Analyzing the Reviewed Provenance Systems", we have used such a tool to create the RCM defined from the provenance definitions extracted from the primary studies. As for the *data processing* category, the key distinction between the different strategies is the focus of data processing control, or what drives data transformations [17]. *Script-based* strategy suggests the enactment of one or more single-user processing threads at the operating system or scripting environment level. *WFMS-based* processing is controlled by a workflow engine, while *Query-based* relies on the use of one or more DBMSs. Finally, *Service-based* depends on a network of web servers or Grid nodes [17].

3.2.2. Data Capture *dimension*

Table 4 shows the categories included in this dimension. Next, we give specific remarks of several of such categories.

Table 4. Data extracted regarding *Data Capture* dimension

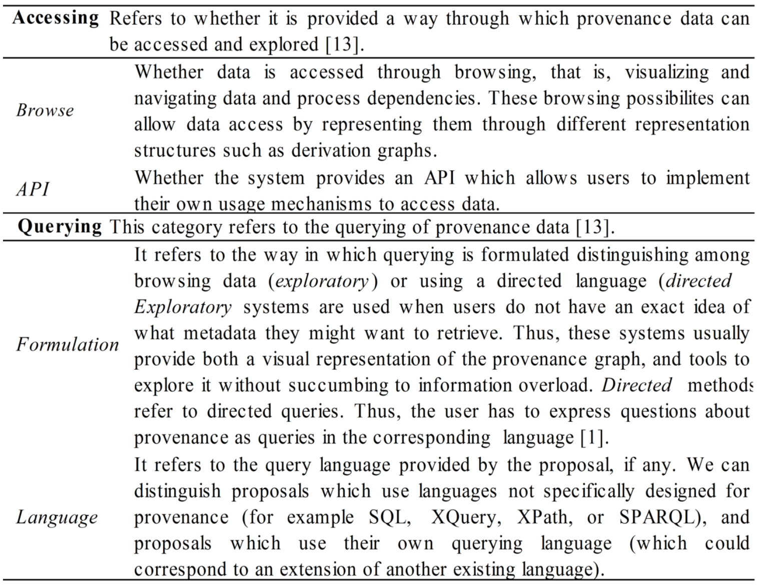


Tracing		It refers to the approach considered for tracing data provenance, that is, the proposal taken for computing provenance information [13,16,18].
Approa.	*Lazy*	Whether the provenance of a data product is computed only when required, after runtime. Is is used by database systems [13].
	Eager	Whether data provenance is computed immediately by carrying the provenance of data product along as data is transformed, making it ready to be used as metadata. It is normally used by SWFMSs [13].
Level		This category refers to the levels of capture, that is, the levels where the distinct forms of provenance can be gathered [13]. It is related to the relevant details of the computational tasks the capture mechanism needs access to.
Approaches	*Workflow*	At this level, each SWfMS is responsible for gathering all the provenance information [13]. Thus, in systems adopting this level of capture the provenance gathering mechanisms are either attached to or integrated in a SWfMS[6].
	Activity	At this level, also known as *process*[6], each process involved in a computational task is required to capture its own provenance information [6,13].
	Operating system (OS)	In this level of capture provenance data is collected at the API system level, relying on the availability of specific functionality at the OS level; thus, no modifications are needed to existing scripts, programs or workflows [13].
	Services	This level is concerned with the execution history of a service workflow in SOA-based systems. It is related to the capture of provenance data associated with the running of the service workflows.
	Database/ Data warehouse	This level is related to these systems which track provenance in database/ data warehouse scenarios.
Mechanism		It refers to the mechanism used to capture provenance data [13].
App.	*Internal structures* *External services*	Whether *internal structures* or *services external* to the provenance system are used to capture data, respectively. In particular, the external services strategy is adopted to collect provenance from both distributed and heterogeneous environments [13].
Technique		This category is related to the techniques that existing provenance systems use to capture provenance [13,18].
App.	*Annotation*	It refers to the process of adding to or ``making up" existing data, sometimes in an ad-hoc way [15]. Annotations refer to the metadata comprising the derivation history of a data product regarding source data and processes [12].
	Inversion	It refers to compute provenance information instead of storing it previously. That is, it aims at recreating source data items from result data items [18].

Regarding the *tracing* category, the approaches adopted for tracing data provenance are *eager* and *lazy.* The advantage of an *eager* method is that it has access to all run-time properties of a transformation (that is, a process that was involved in the creation of a data item), and thus, it can be applied to a wider range of transformation types. However, computing provenance eagerly incurs run-time overhead for the transformation to compute the output, and requires additional storage space to store the produced provenance information. In contrast, a *lazy* computation mostly does not result in additional storage space and run-time overhead, but it is not applicable for all types of transformations and it can slow down provenance retrieval [16, 13, 9]. As stated in [9], in many settings *lazy* and *eager* aspects can be combined to benefit from the advantages of both methods, by computing provenance on-demand by re-executing the transformation and recording provenance *eagerly.*

In the case of the *level* category, in Table 4 we show the most significant approaches. The most common proposal is the *workflow-level.* One of the main characteristics of this solution is that, being usually tightly coupled with SWfMS systems [6], it enables a straightforward capture of the process while the workflow is being enacted. For such a task it uses system's APIs [13]. In contrast, its dependency on the SWfMS makes it difficult to use the same mechanism

Table 5. Data extracted regarding *Data Access* dimension



Accessing	Refers to whether it is provided a way through which provenance data can be accessed and explored [13].
Browse	Whether data is accessed through browsing, that is, visualizing and navigating data and process dependencies. These browsing possibilites can allow data access by representing them through different representation structures such as derivation graphs.
API	Whether the system provides an API which allows users to implement their own usage mechanisms to access data.
Querying	This category refers to the querying of provenance data [13].
Formulation	It refers to the way in which querying is formulated distinguishing among browsing data (*exploratory*) or using a directed language (*directed* *Exploratory* systems are used when users do not have an exact idea of what metadata they might want to retrieve. Thus, these systems usually provide both a visual representation of the provenance graph, and tools to explore it without succumbing to information overload. *Directed* methods refer to directed queries. Thus, the user has to express questions about provenance as queries in the corresponding language [1].
Language	It refers to the query language provided by the proposal, if any. We can distinguish proposals which use languages not specifically designed for provenance (for example SQL, XQuery, XPath, or SPARQL), and proposals which use their own querying language (which could correspond to an extension of another existing language).

in other scenarios, being applicable only in non-heterogeneous scenarios where only one SWfMS is used. *Activity-level*, on the other hand, is independent from the SWfMS. Additionally, this level allows capturing information with a smaller granularity than the one captured by the workflow proposal, since the mechanism is attached individually at each concrete activity [13]. The *Operating System (OS)-level* provides the possibility of recording a low level of the metadata completeness, which would be difficult or impossible to achieve with *workflow-level* or even with *activity-level* solutions. It is also independent from the SWfMS. The *services-level* has been recently considered given the growing attention the research on data provenance has received in service oriented computing. In particular, in a Service Oriented Architecture (SOA), resources on the Web/Grid (including hardware, software code, application systems, and knowledge) are regarded as services, and such services are brought together to solve a given problem typically via a workflow that specifies their composition. For more pros and cons of these levels, the reader is referred to [13, 6].

Finally, another aspect that deserves special attention is the *technique* category and the two strategies used to capture provenance (*annotation* or *inversion*). Focusing on providing a comparison between both strategies, it is worth remarking that the *inversion* method provides a more compact representation of the provenance than the given by the *annotation* strategy [12, 18]. However, the information it provides is sparse and limited to the derivation history of the data. On the other hand, *annotations* give more flexibility in the richness of provenance metadata and the provenance need not be computed "just-in-time" like in the *inversion* method [18]. *Annotations* can be provided both manually by users, or automatically by applications. In order to avoid the tedious, time–consuming and error-prone task of a manual record, *annotations* are usually recorded automatically by SWfMS systems or databases systems [13]. We note that most of the *eager* approaches rely on *annotations*, while most of the *lazy* approaches are based on *inversion* [2].

3.2.3. Data Access *dimension*

Accessing provenance data repositories by scientists constitutes a key aspect in any provenance system. Sometimes the huge amount of data makes it so difficult to locate, retrieve or query the knowledge given by provenance records. In this dimension, we have identified two ways by means of which scientists can retrieve provenance data from repositories: *accessing* and *querying* (see Table 5). In the particular case of the *querying* category, we have distinguished among: *formulation* and *language*. As stated by [1], *formulation* is recognized as a challenging problem mainly due to two reasons: (1) users often want to query over a broad range of information, and (2) users ask questions that the designers of the provenance system did not anticipate. The result of the question depends on the granularity of the capture in such a way that the system either might have insufficient data for a meaningful answer, or it might produce so much data that it is difficult to explore and understand the answer. Two core querying paradigms have emerged: *exploratory* and *directed*, being a smaller number of systems which use a hybrid of both approaches. *Exploratory* proposals require a visual representation of the provenance graph, which constitutes a notably hard problem since even small provenance graphs can easily contain thousands of nodes. *Directed* methods are effective if the user knows precisely what information is required, but unlike *exploratory* methods, the directed query approach does not facilitate discovery of new insights about the provenance graph [1]. As for the *languages* used to query provenance, we refer to [116] for a detailed explanation.

3.2.4. Subject *dimension*

A summary of the different categories included in this dimension is presented in Table 6. The first category, which we refer to as the *contents* category, is related to the different formalisms for provenance considered in the database community. We note that, although these are the most common forms of provenance, in the literature there have been other deviations. An example is the *why not provenance* or *why not query* [4], or more generally, first-order queries with negation. Since *why*, *how*, and *where*–provenance are the most used proposals, we refer to [16] for a detailed review.

Regarding the *abstraction* category, we note that the use of *workflow-induced views* is the most widely extended approach, adopted by most of the SWfMSs. The remainder abstraction technologies are commonly adopted by systems of concrete nature. For example, *tracers* are usually adopted by systems which capture low-level provenance. The *layering* proposal is adopted, as expected, by systems which capture information from different levels. For example, there are systems that consider layers that capture separately data such as the specifications of individual workflows or the traces of the workflow execution [4]. Finally, given that *accounts* are technology independent, this approach is considered by both workflow and non-workflow based systems (a more complete explanation can be seen in [4]).

As for *interoperability/exchange*, we introduce the Provenance Challenges series [117], initiated at IPAW'06 during a discussion on the need for provenance standardization. These series aimed to understand aspects such as the differ-

Table 6. Data extracted regarding *Subject* dimension

Contents		The most common forms of *database provenance* describe relationships between data in the source and in the output regarding the effect of queries. In many cases, the result of executing a query consists mainly of copying a data element from some source to some target database [15]. At this respect, several are the notions proposed in the literature, which are the followings:
Approaches	*Lineage*	*Lineage* considers tuples to belong to the provenance if they "contribute" to an output tuple [61]. Introduced by Cui and Widom in the early 2000s [61], it was the earliest formalism. They proposed the notion of *lineage* for data warehouses as a way of relating the tuples in a query output to the tuples in the query input that ``contribute" to them.
	Where-provenance	*Where-provenance* refers to the identification of the source elements *where* the data in the target is copied from, that is, it refers to *where* output data came from in the input [14]. Together with the *why-provenance*, it was introduced by Buneman et al.[14] roughly concurrently to the *lineage*.
	Why-provenance	This form of provenance explains *why* a piece of data is being created. More specifically, it justifies the data appearing in the output, that is, represents the origins that were involved in calculating a single entry of a query result [14].
	How-provenance	This form of provenance explains *how* source data are involved in the creation of the result, that is how their involvement proves the result [4,129]. It was introduced later by Green et al.[129].
Abstraction		Refers to the use of approaches which help to structure the provenance information in order to deal with its complexity and size, easying the understanding of provenance and make sense of it.
Approaches	*Workflow-induced views*	Taking into account that the amount of provenance information may be overwhelming, the notion of workflow-induced views (or user views) refers to the possibility of using abstraction mechanisms which help users focus on information the users find relevant [58].
	Accounts	Accounts are a workflow-independent mechanism that allows to introduce abstraction and structure in a provenance trace. Accounts allow for multiple descriptions of a given execution to co-exist in a provenance trace [4].
	Layering	A proposal consisted in layering captures provenance information separately from different layers.
	Tracers	Tracers refer to unique tokens which are propagated by services at execution time through interactions. A tracer can be used to delimit a process (or subprocess) or activities with some properties, and bundle all provenance information regarding the process (or subprocess) or activities [4].
Interoperatibility/ Exchange		Refers to the interoperatibility and exchange of data among the different provenance systems. There have been several attempts to establish a model to particularly represent workflow executions as provenance records, and which allows importing and exporting provenance among existing systems.
App.	*OPM*	The Open Provenance Model (OPM) is considered the de facto standard for representing provenance [118].
	PROV	The W3C PROV standard is the successor of OPM, which mainly covers some gaps of OPM and adds new relationships. PROV constitutes a family of specifications. It is basically a set of recommendations on how to model and interchange provenance in the Web [119].
Phase		Refers to the different moments in time in which provenance is captured. We distinguish between execution (or *retrospective* provenance) and composition (or *prospective* provenance) [7].
Approaches	*Retrospective Provenance*	It refers to the recording of when and where each procedure ran, and how each invocation behaved. It captures the steps that were executed as well as the information about the execution environment.
	Prospective Provenance	It captures the specification of the workflow procedure calls and data dependencies. It corresponds to the steps that need to be followed to generate a data product or class of data products.
Orientation		Provenance data can be either available explicitly or deduced indirectly. Taking this into account, different provenance models can be considered distinguishing among *Data-oriented* models, or *explicit* models, and *Process-oriented* models, or *indirect* models [18]. Both types of models are strictly related to the two different types of information usually considered to form provenance data *(source* and *transformation* provenance, respectively), as presented in Section 1.
App.	*Data-oriented*	In a *data-oriented* model, provenance metadata is specifically gathered about the data product. It refers to a model which focuses on explicitly provenance data [18].
	Process-oriented	In a *process-oriented* model, the primary entities for which provenance is collected are the deriving processes, and the inputs and outputs of these processes are inspected to obtain the data provenance. It refers to a model which focuses on deriving processes [18].
Granularity		Refers to the level of detail of a specific data product. More specifically, it refers to the types of objects for which a system maintains provenance. The granularity gathered by a specific proposal constitutes an aspect specially remarkable in such a proposal since it establishes the usefulness of provenance.
App.	*Fine-grained*	This type of granularity, also known as *data provenance* [5], is concerned with the provenance of relatively small pieces of data [5].
	Coarse-grained.	This granularity, which is also called *workflow provenance* [5,15], records a complete history of the derivation of some data product generated through processing a workflow [13,15].

ent representations used for provenance, its common aspects, or the reasons for its differences. The community agreed that a "Provenance Challenge" should be set to compare and understand existing approaches. There have been four challenges [117]. In particular, in the Third Provenance Challenge there was an increasing effort in adapting existing provenance systems to support the Open Provenance Model (OPM) [118]. A current best practice for provenance representation is PROV, an OPM successor [119].

A remark must also be made regarding the *phase* category (see Table 6) and its relationship with the different levels of *capture* of provenance (see Subsection 3.2.2). Since *workflow-level* mechanisms have access to workflow definitions and control their execution, these mechanisms can capture both *prospective* and *retrospective* provenance. In contrast, *OS-level* and *Activity-level* mechanisms require reconstructing causal relationships through provenance queries, thus they give only support to *retrospective* provenance [6].

As for the *orientation* category (see Table 6), it is worth remarking that while the *process-oriented* models are useful for documenting the data generation process, or for reproducing or validating a given process, *data-oriented* models are useful to examine the data dependencies [6]. As stated in [18], either of these methods may be used depending on the application context in which provenance is captured.

Regarding the *granularity* category, the exact meaning of granularity depends on the underlying data model of the application. More specifically, database provenance systems could store provenance metadata for an entire table, a row within the table, or for each cell. Thus, provenance captured at the table level would be considered as *coarse–grained* granularity. Additionally, *coarse* granularity would determine the relationships between tables. Of course, multiple granularities can be considered at the same time. The amount and cost of provenance information can be inversely proportional to the granularity [13]. More specifically, the provenance information can grow to be larger than the data it describes if the data is *fine-grained* and the provenance information rich [13, 18]. Thus, the manner in which the provenance metadata is captured is important to its scalability. We refer to [116] for a more detailed analysis of both types of granularities.

3.2.5. Storage *dimension*

A summary of the different categories included in this dimension is shown in Table 7. Next, we present specific remarks regarding several of such categories.

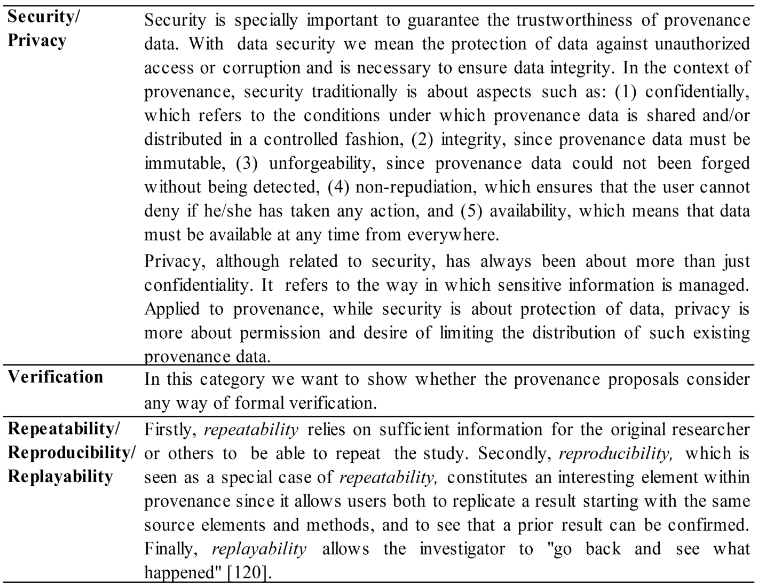
As for the *storage scalability* dimension, both alternatives have advantages and disadvantages depending on the application they will be deployed on [20]. In particular, in a *distributed* system maintenance is easy but it is hard to search and query provenance information since there is not a systematic design behind. In *centralized* systems, although there is a connection between the data and metadata, they are stored in different systems with different representations. Maintenance is more difficult in a *centralized* approach, but it is easier to query and search provenance since the mechanism is designed keeping this requirement in mind [20].

Regarding the *persistence* dimension, we just want to say that the *relational model* constitutes one of the most commonly used storage techniques. On the contrary, as stated in [116], many of the systems (mainly those with a *service-based* architecture) which use the annotation approach, adopt XML as primary

Table 7. Data extracted regarding *Storage* dimension

Storage Scalability		With storage scalability of the provenance data we refer to whether the data is stored in a centralized or distributed way [20].
Approa.	*Centralized*	We consider a centralized scalability whether the provenance system stores the data in one single repository at one location.
	Distributed	In a distributed proposal, multiple logically interrelated repositories are considered, which are distributed over a computer network. Each of these repositories is managed by a storage system (such as a RDBMS or filesystem) that can run autonomously. As in [13], we distinguish between *homogeneous,* if every site runs the same type of storage system, and *heterogeneous,* if different sites run different types of storage systems.
Coupling		This category describes the relationship between the provenance data and the data which is the target of provenance recording [2].
Approaches	*High-coupling*	This recording strategy, also known as *tight-coupling*, stores provenance directly associated with the data for which provenance is recorded [2].
	Loose-coupling	This alternative uses a mixed storage scheme where provenance and data are stored in one storage system but logically separated [2].
	No-coupling	This strategy stores provenance information in one or many provenance repositories. These repositories store only provenance data [2], generally for several data repositories in distributed systems [13].
Persistence		Refers to the expressiveness of the conceptual models used by a provenance management system to persist provenance. A provenance conceptual model defines what information is supported in a provenance system. A wide variety of storage systems have been used in storing provenance data, however, most existing systems use relational databases, XML or RDF.
Approaches	*Relational*	The relational model is an approach to managing data using a structure and language consistent with first-order predicate logic.
	XML	Extensible Markup Language or XML is a markup language, defined by the World Wide Web Consortium (W3C), that establishes a set of rules for encoding documents in a format both human and machine-readable.
	RDF	Resource Description Framework or RDF constitutes a part of the W3C's Semantic Web Activity and is a framework for describing resources on the web. RDF, together with SPARQL and OWL, are the three foundational Semantic Web technologies.
Archiving		This category is related to the different approaches taken to archive provenance: *sequence of delta* and *time-stamping* [13].
App.	*Sequence of delta*	It stores a reference version of data item or a sequence of forward or backward delta between successive versions of a workflow.
	Time-stamping	It stores all versions, and timestamps are used to mark the existence of data products at various times.

Table 8. Data extracted regarding *non-functional requirements* dimension



Security/ Privacy	Security is specially important to guarantee the trustworthiness of provenance data. With data security we mean the protection of data against unauthorized access or corruption and is necessary to ensure data integrity. In the context of provenance, security traditionally is about aspects such as: (1) confidentially, which refers to the conditions under which provenance data is shared and/or distributed in a controlled fashion, (2) integrity, since provenance data must be immutable, (3) unforgeability, since provenance data could not been forged without being detected, (4) non-repudiation, which ensures that the user cannot deny if he/she has taken any action, and (5) availability, which means that data must be available at any time from everywhere. Privacy, although related to security, has always been about more than just confidentiality. It refers to the way in which sensitive information is managed. Applied to provenance, while security is about protection of data, privacy is more about permission and desire of limiting the distribution of such existing provenance data.
Verification	In this category we want to show whether the provenance proposals consider any way of formal verification.
Repeatability/ Reproducibility/ Replayability	Firstly, *repeatability* relies on sufficient information for the original researcher or others to be able to repeat the study. Secondly, *reproducibility,* which is seen as a special case of *repeatability,* constitutes an interesting element within provenance since it allows users both to replicate a result starting with the same source elements and methods, and to see that a prior result can be confirmed. Finally, *replayability* allows the investigator to "go back and see what happened" [120].

format for message exchange. It is worth saying that some of the systems which use *annotations* also capture semantic information within provenance using domain ontologies in languages such as RDF and OWL [18]. Due to space reasons, for an explanation of advantages and drawbacks of these storage techniques in the provenance field, the reader is referred to [116].

Finally, as far as *archiving* is concerned, we want to note that the *sequence of delta* and *time-stamping* approaches differ only in their change representation. While the former technique describes the difference between two versions around time (through a delta), the latter technique describes the difference around data (through time-stamps) [13].

3.2.6. Non–Functional Requirements *dimension*

An explanation of the different categories included in this dimension is presented in Table 8. In particular, as for the *security/privacy* category, although these aspects are different, we have decided to join them into one category since, as we will see later in Section 3.3 "Analyzing the Reviewed Provenance Systems", most of the works published on provenance systems hardly make any appointment on security or privacy and, if made, they mix both aspects. A *verification* category has been also included since we consider of importance for a provenance system to adopt a verification approach for aspects such as the verification of provenance data, or the verification and trustiness of the business processes through which such data is produced. Regarding the last category, we have adopted the definitions from [120] for being the most representative and clear ones.

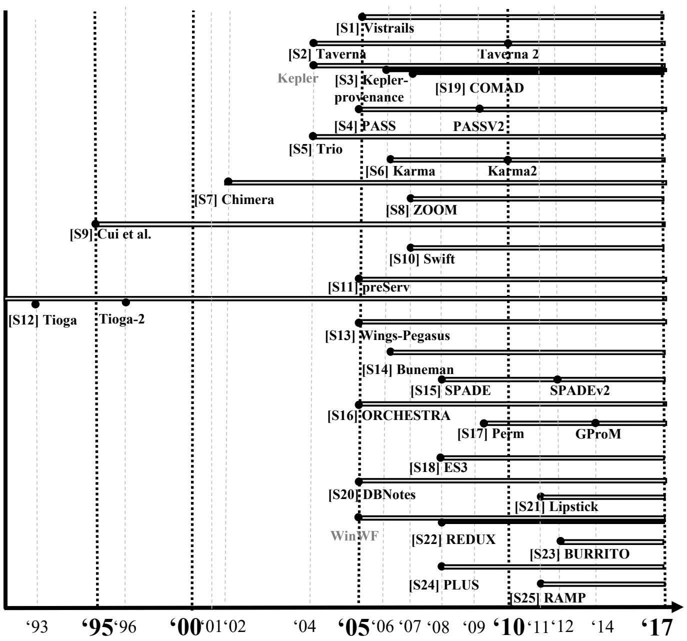


Fig. 2. The timeline of surveyed systems.

3.3. Analyzing the Reviewed Provenance Systems

To contribute to the understandability of the analysis, in Figure 2 we present a general timeline of the surveyed systems, considering their evolution along time, their extension relationships, and the versions of these systems explicitly identified in the literature (which particularly provide additional innovative improvements compared to the previous versions). In particular, [S17] GProM is an *evolution* of the Perm system. The [S19] COMAD framework and the [S3] Kepler provenance module (*Kepler provenance* for short) are *extensions* of the Kepler system (see black filled lines in Figure 2). Similarly, [S22] REDUX is an extension of the WinWF framework [105]. Among the different *versions* we note [S2] myGrid/Taverna-myGrid/Taverna2, [S4] PASS-PASSV2, [S6] Karma-Karma2, [S12] Tioga-Tioga2 and [S15] SPADE-SPADEv2 (we have considered the properties of the most recent system version). Finally, [S13] Wings-Pegasus corresponds to two systems which have been *integrated* to achieve specific provenance characteristics.

Based on the first years of release of the systems, they cover the timespan 1993–2014, noticing an increasing interest from 2004 onwards. The interested reader is also referred to [28] for a brief description of the surveyed systems and extended information with regard to how the provenance systems have been evolving across the years.

Additionally, we have conducted a review of cross-citations between the papers analyzed. For each primary study tackling a reviewed system, we have looked for the remainder 24 reviewed systems cited by it. As a result, we have created a graph of citations (see Figure 3) where each node represents a reviewed system, and each directed edge between nodes *(Sx,Sy)* shows that there is at least

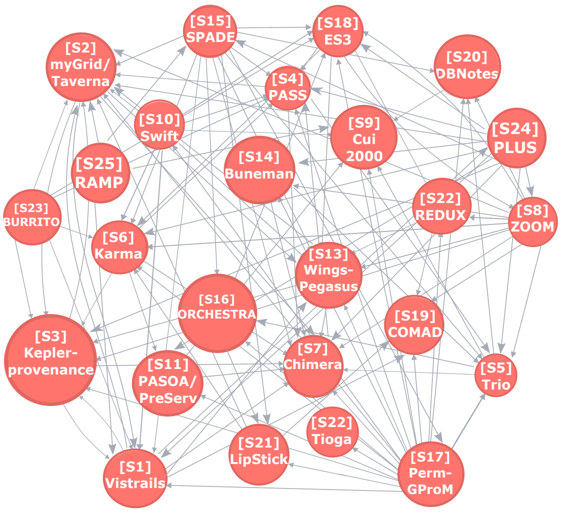


Fig. 3. Overview of connections among the systems

a paper tackling the system *Sx*, which cites another paper tackling the system *Sy*. The large number of connections among the systems particularly shows that along time, provenance systems' developers have been aware of other solutions. More information about this graph is available at [28].

Next, we present the different techniques used by the selected 25 provenance systems to address the different aspects included in our taxonomy. The analysis we have performed is presented by means of a set of tables and figures in which we have used the following general notation. In the case of tables: (i) an empty cell represents that the authors do not mention anything about either the aspect in question or the technique used to tackle such an aspect, (ii) the symbol "N/A" indicates that the aspect is not applicable to the specific system for some reason, and (iii) "No" means that the work explicitly claims that the system does not support the aspect in question. In other cases we have included in the corresponding cell the specific technique. In the case of figures: (i) a coloured circle is used to enclose all the systems which use a specific technique for the aspect in question, (ii) a non-coloured circle is used to enclose all the systems which do not mention anything about the aspect or technique, (iii) "Not applicable" shows that the aspect is not applicable to the systems enclosed, and (iv) "Not supported" shows that such systems do not support the concrete aspect. Tables and figures are also accompanied by explanatory texts when we have considered necessary to give readers a better understanding of the analysis. At this point, we note the difficulty of extracting some data items from certain papers because of the way they were reported.

3.3.1. Results regarding General aspects *dimension*

Within this dimension (see Table 9) a remarkable category is *data processing*, which is closely related to the *purpose* category. The most notable strategies followed to manage data among the surveyed systems are *WfMS* and *query-based*. It is worth remarking [S21] LipStick, whose overall contribution is a framework

Table 9. *General Aspects* dimension

Proposal	Data processing	Application domain	Purpose	Extended or Intended	Availability
[S1] Vistrails	WfMS-based	Ecology, Meterology		Extended	Open source
[S2] myGrid/Taverna	WfMS-based	Generic (Bioinformatics, music, meteorology, medicine, Linked Data)		Extended	Open source
[S3] Kepler provenance	WfMS-based	Biology, Ecology, Geology, Linked Data, Big Data		Extended	Open source
[S4] PASS	API-based	Generic (Big Data, Sensor Network)	Audit trail	Intended	Open source
[S5] Trio	Query-based	Generic (Big Data, Sensor Network)	Data quality	Intended	Open source (BSD licence)
[S6] Karma	Service-based	Biology		Intended	Open source
[S7] Chimera	Service-based	Physics, Astronomy, Grid Computing	Understanding derived data	Intended	
[S8] ZOOM	WfMS-based	Bioinformatics, Big Data		Intended	Open source
[S9] Cui 2000	Query-based	Generic	Understanding derived data	Extended	
[S10] Swift	Script-based	Biology	Audit trail	Intended	Open source
[S11] PASOA/PreServ	Service-based	Biology	Audit trail	Intended	Open source
[S12] Tioga	Query-based	Atmospheric Science	Understanding derived data	Extended	Open source
[S13] Wings-Pegasus	WfMS-based	Grid Computing		Intended	Open source
[S14] Buneman	Query-based	Generic	Understanding derived data	Intended	
[S15] SPADE	Graph-based	Physics, Engineering, Sensor Network	Understanding derived data	Intended	Open source
[S16] ORCHESTRA	Query-based	Bioinformatics	Understanding derived data	Intended	Open source
[S17] Perm-GProM	Query-based	Generic (Linked Data)	Understanding derived data	Intended	Open source
[S18] ES3	Script-based	Earth Sciences, Linked Data	Audit trail	Intended	Proprietary
[S19] COMAD	WfMS-based	Generic		Extended	Open source
[S20] DBNotes	Query-based		Understanding derived data	Intended	
[S21] LipStick	WfMS-based and Query-based	Data-intensive application domains and Big Data		Intended	Open source
[S22] REDUX	WfMS-based	Bioinformatic, Neuroscience, and image processing		Extended	Proprietary
[S23] BURRITO	User Interface	Computational research	Audit trail	Intended	Copyright©
[S24] PLUS	WfMS-based	Generic (Linked Data)	Data quality, audit trail	Intended	Apache License
[S25] RAMP	WfMS-based	Generic (Big Data)	Audit trail	Extended	Open source

that marries database-style and workflow provenance models by means of *WfMS* and *query-based* approaches, respectively. In addition, there are strategies which manipulate data using *services* and allow users to develop scripts. Nevertheless, systems such as [S4] PASS, [S15] SPADE and [S23] BURRITO consider their particular data processing categories. [S4] PASS sets up a *Disclosed Provenance API (DAPI)* which allows transferring of provenance among the components of the system. [S15] SPADE allows users to manipulate provenance data by means of *graph-based* actions according to constraints over the internal data model (OPM). The [S23] BURRITO platform is made up of a set of applications with a graphical *user interface* which allows the user to manage data provenance.

The selected systems have a wide variety of *application domains* ranging from ecology to music going through medicine (see Table 9). While some of the systems are explicitly defined as "generic" systems not tied to a concrete application domain (see Table 9), others have been applied to a concrete domain of application (we also show between parentheses known application domains of the surveyed systems). In the particular case of [S20] DBNotes, we did not find any application domain within the surveyed papers. Within the computer science field, Linked Data and Big Data application domains specially stand out as the most trending fields. Firstly, provenance for Linked Data [121, 122] is a means of adding information with the aim of allowing a future reasoning about quality, reliability or trustworthiness. As we will see in Section 3.3.4, systems which provide interoperability through PROV such as [S2] myGrid/Taverna, [S3] Kepler provenance, [S17] Perm-GProM, [S18] ES3, and [S24] PLUS (see Figure 5), could be used for Linked Data since PROV standard provides an OWL2 ontology allowing the mapping to RDF [119]. Secondly, provenance for Big Data (also called Big Provenance [123]), has been considered as critical in applications with Big Data characteristics to tackle aspects related to the four V's: *volume*, *velocity*, *variety* and *veracity* of data [123, 124]. In the context of Big Data, MapReduce applications such as Hadoop have been adopted. More specifically, Hadoop as a MapReduce agent has become synonymous with Big Data processing and analysis [125]. Among the surveyed systems, [S21] LipStick can be used in Hadoop since it has been developed on top of Pig Latin, and [S25] RAMP captures provenance using wrappers in Hadoop. Finally, we note that no provenance solution declares to fit best to a concrete application scenario.

There exist inherent *purposes* when a system acquire provenance capabilities. Among these purposes, we have considered *audit trail*, *data quality*, and *understanding derived data*. Although all the systems undertake these purposes to a greater or lesser extent, there are systems which highlight a particular one. The main purpose generally remarked by the proposals is *understanding derived data*. In particular, it is one of the most important purposes when provenance capabilities are implemented in database systems ([S9] Cui 2000, [S12] Tioga, [S14] *Buneman*, [S16] ORCHESTRA, [S17] Perm-GProM and [S20] DBNotes). Other systems also share this purpose ([S7] Chimera and [S15] SPADE). Another widely used purpose is *audit trail*. More specifically, systems which address this purpose capture provenance information from processes invisibly, sometimes at operating system level ([S4] PASS, [S18] ES3 and [S23] BURRITO) or other levels of capture ([S10] Swift, [S11] PASOA/PreServ, [S24] PLUS and [S25] RAMP). We note that one of the major purposes within the [S5] Trio system is to generate a confidence value associated with a query result. This value is used to assess the *data quality* of such a result. Since all the systems address the purposes identi-

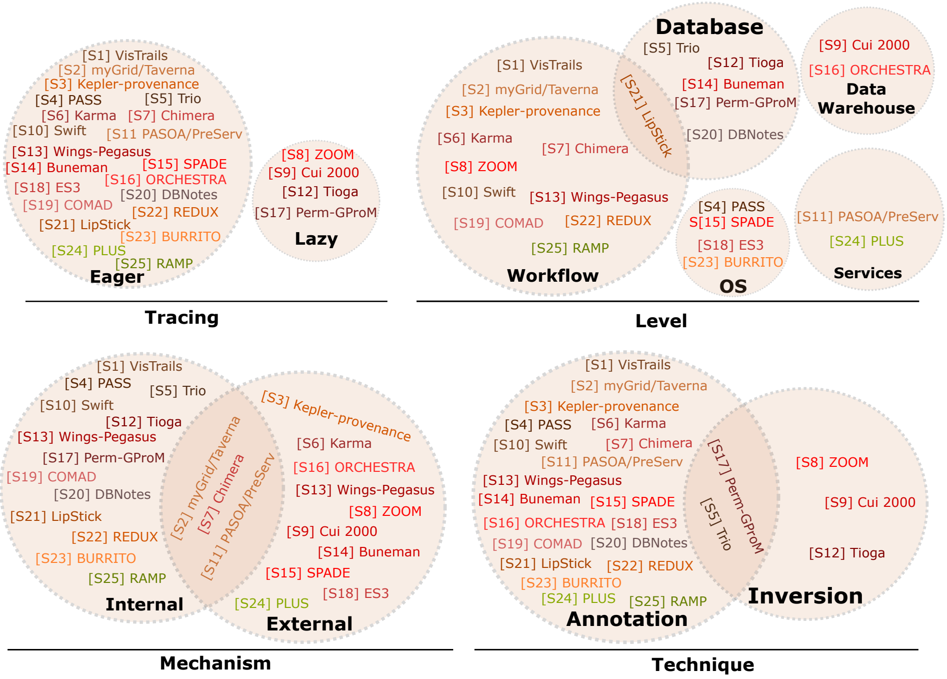


Fig. 4. *Data Capture* dimension

fied previously, in Table 9 we only depict the most significative purposes of each system.

As for tools that *extend* Hadoop by wrapping it (e.g. [S25] RAMP), it is claimed [126, 124, 125] that they have a limited use in production due to their runtime overhead. To mitigate this issue, in [126] authors consider provenance as an intrinsic feature of Hadoop, by updating its source code.

Regarding *availability*, open source solutions are the most widely chosen options (see Table 9). There are also systems which are based on proprietary approaches, such as [S22] REDUX, which has been developed using the privative Microsoft technology *Windows Workflow Foundation (WF)*.

Finally, *provenance definitions* extracted from surveys are not represented in Table 9 because of space reasons. Instead of that, in [28] we present a document with the extracted definitions, and an RCM depicting the most significative terms and particular nuances within the definitions.

3.3.2. Results regarding Capture *dimension*

As for the strategy to *trace* provenance data (i.e., *when* computing provenance), most of the systems compute provenance *eager*ly (see Figure 4). Only [S8] ZOOM, [S9] Cui 2000, [S12] Tioga, and [S17] Perm-GProM choose a *lazy* proposal.

Regarding the *level* where the distinct forms of provenance can be gathered, most of the surveyed systems capture data at the *workflow level* (see Figure 4). There are approaches which manage to capture provenance at several levels, such as [S21] LipStick and [S23] BURRITO. [S21] LipStick captures provenance at the *workflow* and *database* levels. More specifically, this system presents a novel provenance framework that joins "database-style and workflow-style provenance, throughout Pig Latin to expose the functionality of workflow modules" [104]. Keeping this overall goal in mind, it captures two levels of provenance, the internal state and fine-grained dependencies in workflow provenance. [S23] BURRITO uses the *SystemTap Tool* to collect *OS-level* provenance [106]. This tool allows capturing the timestamps and parameters values of system calls (into a log file), as well as the *user's interface* interactions by means of an *Assistive Technology Service Provider Interface* (AT-SPI). In addition, it gathers provenance information in *filesystem* level using a log-structured filesystem called NILFS.

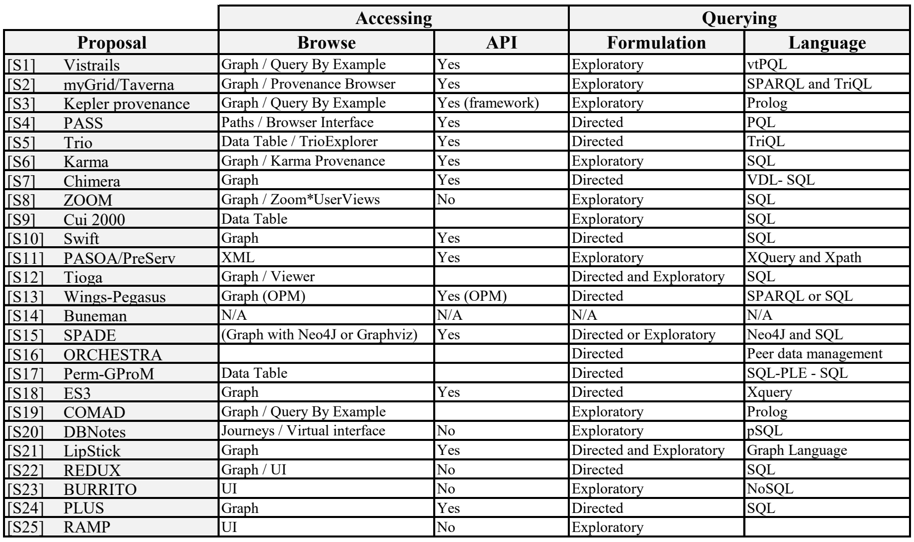
Regarding the *mechanisms* used to capture provenance data, by way of example, we remark [S4] PASS (*internal* approach), and [S7] Chimera and [S11] PASOA/PreServ (both *internal* and *external* approach). [S4] PASS is a Linux framework which implements an *internal* mechanism to capture provenance invisibility through a *collector*. This component intercepts and translates system calls (low-level events) into sequences of meaningful provenance entries. [S7] Chimera allows combining information from both [S7] Chimera (*internal*) and other Data Grid components (*external*). [S11] PASOA/PreServ is a Web Service implementation of the p-assertion Recording Protocol. This system is not only composed of *internal* components which manage captured data, but also of several plugins which implement the same interface in Java to capture provenance *externally*.

As for the *technique* adopted to capture provenance, in the particular case of *annotations*, one of the most commonly adopted proposals is the use of semiring models. As a way of example, Green et al. in [127] propose a model in which every tuple in a relation is annotated with elements from a commutative semiring. Karvounarakis and Green [128], based on [127], present a formal framework for answering queries on data with annotations for commutative semirings. In particular, systems such as [S16] ORCHESTRA and [S17] Perm-GProM use the semiring model presented in [127] to annotate data, and [S21] LipStick uses the semiring model presented in [128]. Regarding systems that adopt an *inversion* technique, [S12] Tioga was one of the earliest systems to represent provenance using inverse functions registered for user defined functions. There are also systems which use both *annotations* and *inversions*. First, [S5] Trio provides an extended model of SQL called ULDB [45] which uses *annotations* to extend SQL functionality. In ULDB, there are annotations for representing aspects such as uncertainty about the presence of a tuple, or numerical confidence values attached to tuples. The aggregation of annotations associated with an output tuple is used to determine the degree of uncertainty of that output tuple. As for the *inversion* model in [S5] Trio, it automatically identifies the source data for tuples created by view queries, by translating such queries into inverse queries. Second, [S17] Perm-GProM uses *annotation* propagation and query rewrite (*inversion*) techniques for computing, querying, storing, and translating the provenance of SQL queries, updates, transactions, and *across* transactions [94]. Finally, we note that the *annotation* approach stands out as the most favorable in the context of computing Linked Data provenance, since it provides richer information of the data and the dataset [129]. Systems such as [S3] Kepler provenance and [S18] ES3 use this technique.

3.3.3. Results regarding Access *dimension*

Regarding the *browse* option (see Table 10), we have used the symbol "/" to distinguish among the representation structures the proposals use to depict the

Table 10. *Data Access* dimension



	Accessing		Querying	
Proposal	Browse	API	Formulation	Language
[S1] Vistrails	Graph / Query By Example	Yes	Exploratory	vtPQL
[S2] myGrid/Taverna	Graph / Provenance Browser	Yes	Exploratory	SPARQL and TriQL
[S3] Kepler provenance	Graph / Query By Example	Yes (framework)	Exploratory	Prolog
[S4] PASS	Paths / Browser Interface	Yes	Directed	PQL
[S5] Trio	Data Table / TrioExplorer	Yes	Directed	TriQL
[S6] Karma	Graph / Karma Provenance	Yes	Exploratory	SQL
[S7] Chimera	Graph	Yes	Directed	VDL- SQL
[S8] ZOOM	Graph / Zoom*UserViews	No	Exploratory	SQL
[S9] Cui 2000	Data Table		Exploratory	SQL
[S10] Swift	Graph	Yes	Directed	SQL
[S11] PASOA/PreServ	XML	Yes	Exploratory	XQuery and Xpath
[S12] Tioga	Graph / Viewer		Directed and Exploratory	SQL
[S13] Wings-Pegasus	Graph (OPM)	Yes (OPM)	Directed	SPARQL or SQL
[S14] Buneman	N/A	N/A	N/A	N/A
[S15] SPADE	(Graph with Neo4J or Graphviz)	Yes	Directed or Exploratory	Neo4J and SQL
[S16] ORCHESTRA			Directed	Peer data management
[S17] Perm-GProM	Data Table		Directed	SQL-PLE - SQL
[S18] ES3	Graph	Yes	Directed	Xquery
[S19] COMAD	Graph / Query By Example		Exploratory	Prolog
[S20] DBNotes	Journeys / Virtual interface	No	Exploratory	pSQL
[S21] LipStick	Graph	Yes	Directed and Exploratory	Graph Language
[S22] REDUX	Graph / UI	No	Directed	SQL
[S23] BURRITO	UI	No	Exploratory	NoSQL
[S24] PLUS	Graph	Yes	Directed	SQL
[S25] RAMP	UI	No	Exploratory	

accessed data (located on the left), and the name of the visual tool used to browse the data, if any (located on the right). In particular, the most commonly way to depict data relies on *derivation graphs*, graphs composed of one branch called *paths* ([S4] PASS), or *journeys* of piece of data ([S20] DBNotes). One proposal widely extended to compose graphs is GraphViz [130], which is a package for drawing graphs specified in DOT language scripts. GraphViz is particularly used by [S1] VisTrails, [S15] SPADE and [S18] ES3.

There are also a lot of systems which implement a provenance graph browser (see Table 10). These browsers usually provide features whereby users can "fly over" viewing areas of interest. Other proposals which use a graph have no visual tool to examine the information ([S7] Chimera, [S10] Swift, [S21] LipStick and [S24] PLUS), hence, the user must make a high accuracy directed query in order to get the expected results. Another alternative to browse data apart from graphs is to use *data tables* ([S5] Trio, [S9] Cui 2000, [S15] SPADE and [S17] Perm-GProM). We particularly highlight [S5] Trio that implements the *TrioExplorer* which provides numerous features for browsing and exploring provenance. Finally, [S23] BURRITO and [S25] RAMP provide a user interface to show provenance information. In particular, while [S25] RAMP's interface allows the user to browse input and output elements, as well as backward-trace output elements, [S23] BURRITO provides different utilities (i.e. Activity Feed, Computational/Activity Context Viewer, Lab notebook Generator) to show provenance from different sources and in several ways.

Following on from the foregoing, some systems provide an *API* to expand the basic access functionality. This goal has been addressed in different manners. [S1] VisTrails has both a visualization API and a script API in order to interact with the system. [S2] myGrid/Taverna is implemented by means of a plugin architecture which enables easier code contributions and extensions (in order to personalize the core functionality). [S7] Chimera has a Graph Transversal API which enhances the power and reduces the complexity of the Virtual

Data System provenance queries. [S10] Swift has an interface to support access to various repositories (string mapping, file-system mapping and CSV mapping are implemented). [S15] SPADE provides an abstract storage interface whereby users can implement their own storage. [S18] ES3 has a web service middleware layer which allows the implementation of different monitoring nodes (called plugins). Finally, [S24] PLUS supplies an API for logging, managing and retrieving provenance data. In this category, cells with a "No" text mean that authors have confirmed us (by email) that their system does not provide such an *API*.

Regarding the *querying* option, and in particular the *formulation* category, we first focus on the *exploratory* paradigm (see Table 10). Among the systems that have implemented this paradigm, we distinguish those which offer a graphical exploration, from those which provide specific functionality in order to extract information. Systems which offer a graphical exploration through a graph are usually related to WfMSs (i.e. [S1] VisTrails, [S2] myGrid/Taverna, [S3] Kepler provenance, [S8] ZOOM, [S19] COMAD and [S21] LipStick), but there are also database oriented systems following this paradigm (i.e. [S9] Cui 2000 and [S20] DBNotes). This kind of systems usually depicts the entire workflow and then, to ease the workflow understandability, allows users to explore it, without succumbing to information overload. In contrast, there are systems which explore the workflow through specific predefined mechanisms (i.e. in [S6] Karma the user can retrieve and visualize a provenance graph for a given workflow ID, and [S11] PASOA/PreServ uses a mechanism for navigating through a Provenance Store's p-structure).

The *directed* paradigm has been implemented by systems which use a *directed querying* language. For example [S4] PASS, [S5] Trio, and [S7] Chimera use PQL, TriQL and VDL, respectively (later explained). It is worth remarking [S15] SPADE and [S21] LipStick, which use a hybrid of both approaches (*directed* and *exploratory*). [S15] SPADE uses a *directed* approach using Neo4J or the SQL database specific language, and *exploratory* approach when the Graphviz's storage proposal is used. [S21] LipStick captures both *coarse* and *fine-grained* provenance; while *coarse-grained* provenance is queried using an *exploratory* paradigm, *fine-grained* provenance is queried by means of a *directed* approach [104].

There are systems which use an existing querying *language* according to their storage system. For instance SQL ([S6] Karma, [S8] ZOOM, [S9] Cui 2000, [12] Tioga, [15] SPADE, [S22] REDUX and [S24] PLUS), XML–based query languages ([S11] PASOA/PreServ and [18] ES3), graph languages ([S21] LipStick), peer data management techniques (such as the ones used by [S16] ORCHESTRA), noSQL ([S23] BURRITO) and RDF–based query languages ([S2] myGrid/Taverna and [S13] Wings-Pegasus). It is worth noting that the last approach provides a great advantage in the context of Linked Data, since the open data published is in RDF language [129]. There are also systems which implement their own language which corresponds to an extension of another existing one. For example, [S1] VisTrails implements *visTrails Provenance Query Language* (vtPQL). This language consists of simple SQL-like expressions with additional functions, predicates and attributes. The Trio Query Language (TriQL) adapts SQL to allow querying lineage, uncertainty or both together. [S4] PASS develops *Path Query Language* (PQL) based on *Lorel query language* [41]. Finally, there are systems which develop their own query language from scratch, such as [S7] Chimera [53], which uses Virtual Data Language (VDL) to support both data definition statements and query statements. We remark that this dimension is not applicable ("N/A") to [S14] *Buneman* because there is no information about

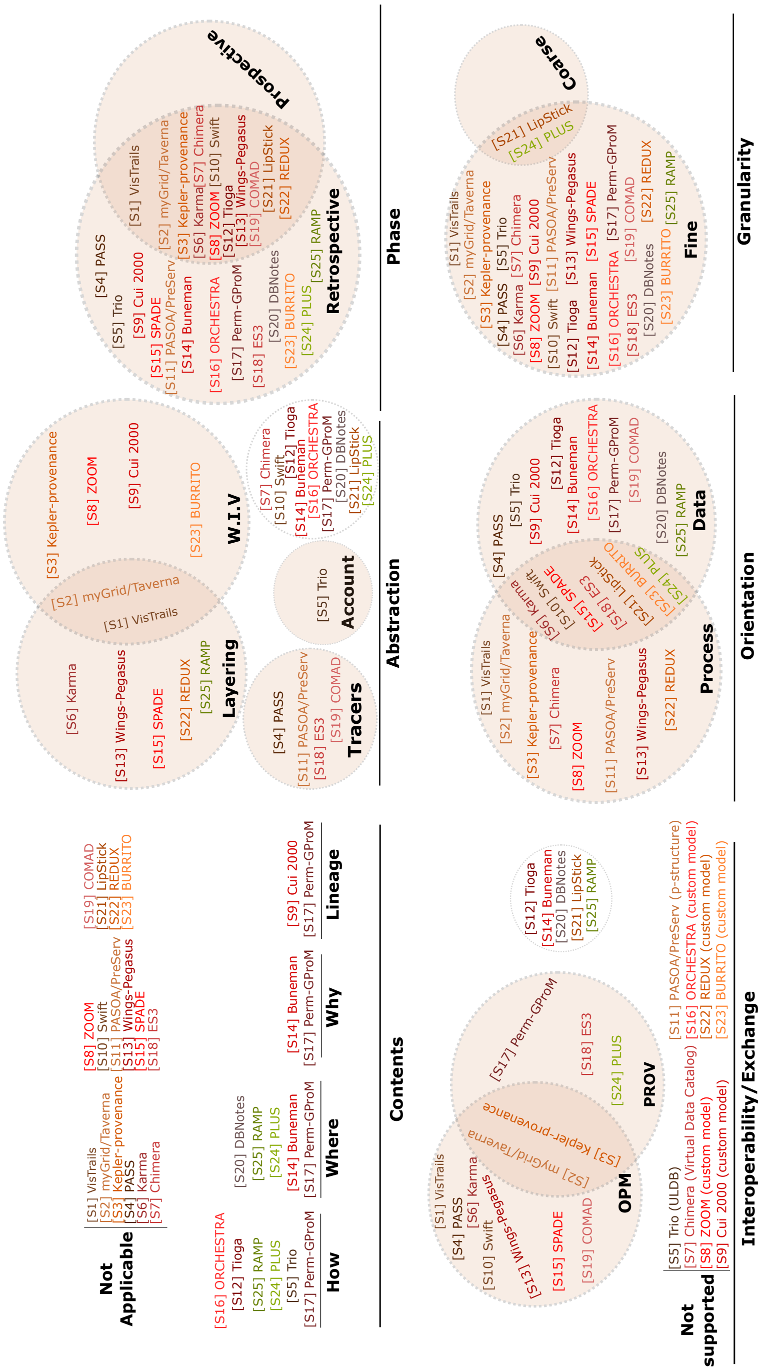


Fig. 5. *Subject* dimension

a particular implementation within its literature. The empty cell in [S25] RAMP represents that such a system does not provide a query language.

3.3.4. Results regarding Subject *dimension*

Regarding the *contents* category (see Figure 5), only [S17] Perm-GProM follows all the approaches. It allows users to know what input of a transformation belongs to the provenance of a given part of the output. This feature has been implemented following the *why* and *where* provenance described in [14], *how* provenance described in [127], and *lineage* proposed by [S9] Cui 2000. Other systems such as [S14] *Buneman*, [S24] PLUS and [S25] RAMP follow two approaches. While [S14] *Buneman* considers *why* and *where* provenance, [S24] PLUS and [S25] RAMP follow *how* and *where* provenance. More precisely, [S24] PLUS and [S25] RAMP use the term backward/forward lineage and backward/forward tracing, respectively. Such a terminology bears a strong relationship with the notion of *where* provenance and *how* provenance. Backward lineage/tracing refers to the capability to trace provenance information that were used to give a result (*where* provenance), whereas forward lineage/tracing refers to how such a result was subsequently used (*how* provenance). [S16] ORCHESTRA uses *how*-provenance to describe how data tuples are evolving among a lot of sources. It satisfies two goals [85]: "(i) reconciliation can choose between transactions based on user preferences, and (ii) efficient incremental recomputation of the target data instance and provenance is possible." The *contents* aspect is not applicable to the remainder systems.

Regarding the *abstraction* category, and based on the most commonly used strategies, we have identified: *Workflow-induced views* ("W.I.V." in Figure 5), *accounts*, *layering*, and *tracers*. Among the surveyed systems most of them consider a *layering* proposal. In [S1] VisTrails, provenance information is organized into three layers: *workflow evolution* layer, which mainly captures the evolution relationship between workflow specifications; the *workflow* layer, which consists of individual workflow specifications; and the *execution* layer, which stores runtime information of workflow execution. Within [S2] myGrid/Taverna, provenance information is recorded by a framework into a log. In particular, it records information (1) related to the process context (input/output, benchmarks...), or (2) inferred from processes which describe the derivation path of a product. It also stores information that links provenance from different layers, or general provenance metadata for experiments. [S6] Karma keeps information from both execution level and higher level process details (registry level). [S13] Wings-Pegasus considers three layers: a *workflow template* layer, a *workflow* instance layer, and an *executable workflow* layer. In particular, Wings implements the first two layers, while Pegasus supports the third one [73]. [S15] SPADE could also operate at different levels of abstraction and uses a crosslayer composition filter to combine provenance data from the different sources. [S22] REDUX follows a four layered model [105]. More specifically, the first layer considers an *abstract* description of the experiment. The second layer represents an instance of the abstract model. The third layer considers the execution of the workflow instance specified in the second layer. The final layer represents runtime-specific information. Finally, [S25] RAMP captures provenance from the five components which conform a MapReduce job: Record-reader, Mapper, Combiner, Reducer and RecordWriter [112]. Again, we remember that the no-coloured circle represents the fact that the authors do no mention anything about the aspect or technique.

Another alternative is allowing users to focus on a higher level of abstraction by defining user *workflow induced views* (W.I.V. in Figure 5). For example, user views in [S8] ZOOM use composite modules, i.e. modules which may themselves contain subworkflows, and hide portions of a workflow run (thus simplifying both the workflow specification and provenance information). More specifically, [S8] ZOOM dynamically constructs user views using a *bottom-up* approach which takes as input a workflow specification and a set of relevant modules, and produces a user view as output.

The systems [S4] PASS, [S11] PASOA/PreServ, [S18] ES3 and [S19] COMAD follow a *tracers-based* approach. Finally, the *account* proposal is mainly presented by the notion of *account* in OPM [118]. In particular, the [S5] Trio system implements the uncertainly-lineage-database (ULDB) [45] data model which considers the notion of *alternative*, feature strongly related to the notion of *account* in OPM.

Regarding the *phase* or moment in time in which provenance is captured, the analysed systems can capture provenance at execution time (*retrospective*) or at composition time (*prospective*) (see Figure 5). The latter approach, which is adopted by systems which store the workflow specification before the execution (abstract workflow descriptions), is not considered alone but together with the *retrospective* approach. The *retrospective* approach is the most widely used, and it is adopted by systems which capture information while the workflow is executing (workflow execution logs). As a result, twelve systems capture both types of provenance, while thirteen only capture *retrospective* provenance.

As discussed previously, since the Third Provenance Challenge (3PC), there has been an increased effort to give support to *interoperability* using OPM. In particular, some systems such as [S13] Wings-Pegasus and [S15] SPADE have been developed using the OPM recommendations since the release of OPM. The most widely adopted approach acquired is adding import/export capability to OPM. However, OPM-compliant frameworks ([S6] Karma and [S15] SPADE) have also been developed to manage provenance data. In particular, [S6] Karma takes both *wasExecutedOn* and *wasConnectedTo* relationships from OPM.

Regarding the PROV standard, we note that it is considered as a Linked Data provenance representation language [129] which allows users to publish provenance data as Linked Data. In order to add PROV capabilities, some systems have implemented a plugin (i.e. [S2] myGrid/Taverna or [S18] ES3 [131]), whereas other systems have developed export/import provenance functionalities. For example, the implementation in GProM presented in [95] allows the generation of PROV documents, [S3] Kepler allows the serialization of provenance as PROV-JSON by means of a plugin, and finally, [S24] PLUS API service supports the representation of provenance using PROV. Nevertheless, there are systems that, although they implement their own internal model, they do not add interoperability capabilities.

As for *orientation*, there are proposals which explicitly consider both models: *process* and *data* (see Figure 5). Among these proposals, we note the [S6] Karma's provenance model which encompasses both *process* provenance, which is focused on metadata describing the workflow's execution together with associated service invocations, and *data* provenance, which is related to data usage and generation.

Finally, regarding *granularity*, most of the systems capture *fine-grained* data. Examples of database systems that consider *fine* granularity are [S5] Trio, [S9] Cui

Table 11. *Storage* dimension

Proposal	Scalability	Coupling	Persistence	Archiving
[S1] Vistrails	Centralized	High-coupling	XML files or relational DB	Time-Stamp
[S2] myGrid/Taverna	Centralized	High-coupling	RDF triples in large DBs	Time-Stamp
[S3] Kepler provenance	Centralized	High-coupling	Modeling Markup Language (MoML) files (XML-based)	Time-Stamp
[S4] PASS	Centralized	High-Coupling	Low-level data representation	Time-Stamp
[S5] Trio	Centralized	Loose-coupling	Relational DB	Time-Stamp
[S6] Karma	Distributed / Heterogeneous	Loose-coupling	XML and Relational DB	Time-Stamp
[S7] Chimera	Centralized or Distributed / Homogeneous	No-coupling	Relational DB	Time-Stamp
[S8] ZOOM	Centralized	Loose-coupling	Relational DB	Time-Stamp
[S9] Cui 2000	Distributed / Heterogeneous	Loose-coupling	Relational DB	Time-Stamp
[S10] Swift	Centralized	High-coupling	Relational DB or XML Database	Time-Stamp
[S11] PASOA/PreServ	Centralized	High-coupling	p-assertions	Time-Stamp
[S12] Tioga	Centralized	No-coupling	Relational DB	Time-Stamp
[S13] Wings-Pegasus	Centralized	Loose-coupling	OWL and Relational DB	Time-Stamp
[S14] Buneman	Distributed / Heterogeneous	No-coupling	Relational DB, XML DB, File systems or Web sites	Time-Stamp
[S15] SPADE	Centralized	No-coupling	Relational DB or NoSQL	Time-Stamp
[S16] ORCHESTRA	Distributed / Heterogeneous	Loose-coupling	Database	Time-Stamp
[S17] Perm-GProM	Centralized	Loose-coupling	Relational DB	Time-Stamp
[S18] ES3	Centralized	Loose-coupling	XML Database	Time-Stamp
[S19] COMAD	Centralized	High-coupling	XML files	Time-Stamp
[S20] DBNotes	Centralized	High-coupling	Relational DB	N/A
[S21] LipStick	Distributed / Homogeneous	High-coupling	Database	Time-Stamp
[S22] REDUX	Centralized	High-coupling	Relational DB	Time-Stamp
[S23] BURRITO	Centralized	High-coupling	NoSQL	Time-Stamp and Seq. of Delta
[S24] PLUS	Distributed / Heterogeneous	No-coupling	Relational DB	N/A
[S25] RAMP	Distributed / Homogeneous	High-coupling	File systems	N/A

2000, and [S15] ORCHESTRA, which store provenance metadata at DB tuple-level, and [S12] Tioga and [S20] DBNotes, which capture provenance at DB attribute-level. Examples of OS-based systems with *fine-granularity* are [S18] ES3 and [S4] PASS, which track provenance of system calls. As described previously, several granularities can be considered at the same time ([S21] LipStick and [S24] PLUS). [S21] Lipstick proposes a kind of hybrid Big Data provenance framework that combines database-style (*fine-grained* dependencies) and workflow-style provenance (*coarse-grained* dependencies) on top of Pig Latin. As advanced previously, Lipstick can be used in Hadoop, making it useful on a larger scale. [S24] PLUS manages different levels of granularity such as tuples, tables, or whole DBs, and arbitrary size XML sub-trees. In the realm of quality of Linked Data, it is clear that higher granularity will ease the quality assessment of the published Linked Data, giving more refined results [129].

3.3.5. Results regarding Storage *dimension*

Scalability constitutes a crucial factor in provenance systems given the huge volume of data provenance which is managed. Most of the surveyed systems follow a *centralized* provenance storage (see Table 11). Only eight of them store provenance data *distributively*, such as [S16] ORCHESTRA, which ties together pre-existing, heterogeneous data sources in a *distributed* context. As an example of a *centralized* approach, the [S16] ZOOM system stores the provenance information in a single Oracle warehouse, including workflow specifications, user view definitions and log information obtained during a workflow execution [56]. In contrast, [S7] Chimera considers the possibility of adopting a *centralized* or *distributed* approach. More specifically, it offers a Virtual Data Catalog (VDC) for provenance information which registers transformations, data objects and derivations (an execution of a transformation). Although provenance data can be stored in a single VDC, VDC contents are typically distributed over multiple information resources [53].

As for the *coupling* aspect, broadly speaking, most approaches based on *annotations* use (i) a *high-coupling* strategy, by attaching provenance annotations to data items, or (ii) a *loose-coupling* strategy, by storing annotations in the same data repository, but logically separated from the data items. [S14] Buneman is an example of a *no-coupling* approach since it encompasses two databases: the *local* database with the raw data, and the *auxiliary* provenance database which stores provenance information. As for *high-coupling*, [S20] DBNotes can be considered representative since it allows associating every attribute value of every tuple with zero or more annotations. Finally, among the systems following a *loose-coupling* approach, we note [S8] ZOOM, which uses Oracle warehouse as persistence system. Oracle warehouse stores the provenance information together with the data, but logically separated in different tables [58].

As described previously, a wide variety of *persistence* approaches have been used for storing provenance ranging from specialized Semantic Web languages (e.g. RDF and OWL) and *XML* dialects, to tuples in a *relational DB* (see Table 11). Of course, the querying techniques are closely tied to the data and storage models these systems adopt (e.g. Relational DBs use SQL-based languages, XML-based approaches use XQuery and XPath, RDF-based proposals use SPARQL). A huge amount of the surveyed systems use a relational model as persistence system, either alone or combined with other storage techniques (see Table 11). Following on the database approach, [S15] SPADE and

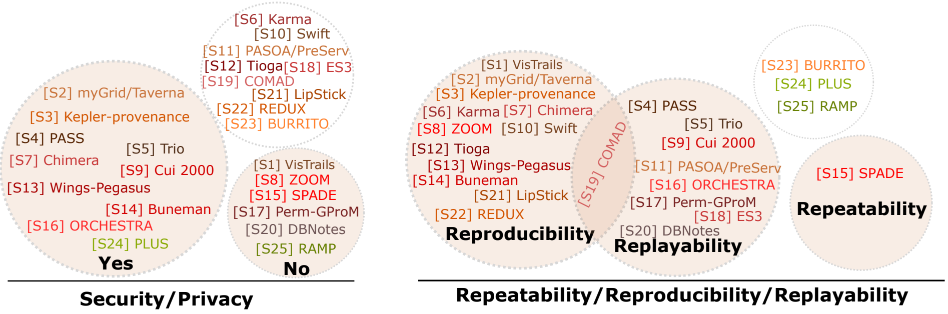


Fig. 6. *Non-functional requirements* dimension

[S23] BURRITO are the only systems which use a *noSQL-based* DB system (Neo4J and MongoDB, respectively), however, [S15] SPADE also allows Relational DB. [S16] ORCHESTRA and [S21] LipStick use DBs but their type is not specified in the literature. Other systems use a XML-based format, such as [S3] Kepler provenance (*Modeling Markup Language* files) and [S19] COMAD.

Among the systems that implement semantic technologies, we note [S2] myGrid/Taverna, which stores provenance data in RDF triples stored in large databases. Additionally, [S13] Wings-Pegasus uses the Web Ontology Language (OWL) for representing specific provenance information. Other systems combine or provide the option to use alternative techniques, such as [S1] VisTrails (which allows using XML files or relational DB), [S10] Swift (which allows both relational and XML DB) and [S6] Karma (where provenance in XML format is stored in a relational DB).

There are other systems which provide an internal private format mappable to other existing storage structures. This is the case of [S4] PASS, and [S11] PASOA/PreServ. [S4] PASS uses a simple, low-level data representation that is easily mapped to XML, a relational schema, or any other data format used by an existing provenance solution [40]. In particular, in [40] authors of [S4] PASS provide a prototype which uses a Berkeley DB-based implementation. Additionally, [S11] PASOA/PreServ architecture supports multiple backend storage systems to store its *p-assertions* (in memory, file system, and database backends). [S14] *Buneman* does not require to use a specific persistence storage but allows the source and target databases to be relational or XML DBMSs, or consist of files stored in file systems or Web sites.

Finally, we note [S25] RAMP, in the context of MapReduce. Since it has been built as an extension of Hadoop, it uses *Hadoop Distributed File System* (HDFS) which is a Java-based filesystem for storing data across multiple machines.

As for the *archiving* category, just to note that the vast majority of the selected systems adopt a *time-stamp* proposal, being only [S23] BURRITO which also considers the *sequence of delta* proposal (*Seq. of Delta* in Table 11).

3.3.6. Results regarding Non–functional requirements *dimension*

Regarding *security* and *privacy*, a huge number of surveyed systems assume that privacy and security rely on the host system or the file-system model. Since providing *security* and *privacy* for provenance records is not a popular topic, both aspects within papers are still scarce. Thus, extracting *security* and *privacy* information from primary studies has not been a straightforward task. Systems classified as "No" in Figure 6 mean that authors confirmed us (by email) that their system does not consider security/privacy aspects.

Regarding the set of systems analyzed in this review, we got *security/privacy* information about a limited number of systems. First, [S2] myGrid/Taverna allows users to access secure sources such as databases, R server, or data by means of different authorization mechanisms (digest, basic, credentials...). However, data provenance and previous runs security rely on the database security model. [S3] Kepler provenance relies its security features on the system where it is executing. [S4] PASS security depends on UNIX file-system security model. [S5] Trio computes confidence on query results through accuracy of data. [S9] Cui 2000 allows the user to select the desired level of consistency and system support algorithms to achieve the different levels [63]. [S14] *Buneman* considers records integrity as a motivation for working on provenance in curated DBs. [S16] ORCHESTRA's provenance model throughout its query language supports security access levels. Finally, [S24] PLUS provides two capabilities to assist security issues. First, it allows the user to assign privilege predicates to the nodes and the ability to restrict access to edges. Second, it is able to provide alternative information to the users not authorized to see the underlying provenance data using *surrogate nodes*. It is worth noting that *Grid Secure Infrastructure* (also called *Globus Security Infrastructure* or GSI) based authentication and authorization is supported by systems such as [S7] Chimera, [S3] Kepler provenance, [S13] Wings-Pegasus.

Regarding the formal *verification*, none of the surveyed systems uses formal verification methods neither to verify provenance data nor to check business processes. In fact, there is limited literature within the provenance context addressing formal verification. Although not included in our surveyed systems, it is worth noting the following works. In [132] it is presented the first systematic view on the logical foundations of formal verification of data provenance records by means of provenance specification logic (PSL^M). More recently, the validation of PROV documents against a set of PROV-constraints is presented in [133]. Proposals such as [S7] Chimera and [S16] ORCHESTRA use a trusted participating checking as a verification method, but we do not consider this verification method in our analysis since this manual verification uses neither a formal specification nor formal methods. Due to the scarce literature, we have not included a "verification" set in Figure 6.

As for *reproducibility*, it is one of the features that appears in almost all surveyed provenance systems. However, it is not considered a cause but rather a consequence of applying provenance capabilities within a provenance system. In this study we found that provenance systems related to *WfMS-based* systems ([S1] VisTrais, [S2] myGrid/Taverna, [S3] Kepler provenance, [S8] ZOOM, [S13] Wings-Pegasus, [S19] COMAD, [S21] LipStick and [S22] REDUX) allow users to rerun experiments by means of complete information gathered from previous executions. This reproducibility feature bears relation with *prospective* provenance (see the *Phase* category in Figure 5) since workflow specification (sometimes joined with *retrospective* provenance) allows the user to share and rerun executions in different environments. On the other hand, systems related to other *data processing* category also allow reproducibility ([S6] Karma, [S7] Chimera, [S10] Swift, [S12] Tioga and [S14] *Buneman*). [S6] Karma and [S7] Chimera offer reproducibility through the combination of *prospective* provenance and *retrospective* provenance. [S10] Swift has reproducibility capabilities by means of scripts in SwiftScript language which contain *prospective* information. [S12] Tioga allows reproducibility through *add program* and *load program* functionality within the user interface. [S14] *Buneman* provides predefined questions among which *src question* (*What transaction first created the data at a location?*) offers reproducibility capabilities.

Regarding *repeatability*, we have only identified one system, [S15] SPADE. In [134] it is claimed that the system tracks data objects and processes creating a provenance graph and capturing assurance attributes that facilitate such a functionality. As for *replayability*, a huge amount of systems add this capability (see Figure 6). To illustrate this functionality, we will explain two noteworthy approaches: [S19] COMAD and [S23] BURRITO. On one hand, [S19] COMAD is a Kepler extension which adds provenance annotations capabilities, thereby users could "go back and see what happened". On the other hand, [S23] BURRITO's main goal is to provide experimental documentation, by means of which users could search information about the experiment.

4. Discussion

This section has been structured in two main blocks encompassing (1) a critical analysis of findings, and (2) future directions in provenance, as identified in this study.

4.1. Analysis of findings

The work presented in this review shows that research on provenance has grown very quickly over the past decades, existing a huge corpus of contributions on the provenance issue. Among the authors of the revised papers, renowned names such as Luc Moreau, Peter Buneman, Wang C. Tan, Susan B. Davidson, Boris Glavic or Juliana Freire stand out.

The group of surveys and reviewing papers identified in this review have provided us with a high-level background on the provenance issue in general. In particular, these surveys give insight into concrete views of provenance by diverse research communities (scientific data processing, scientific workflow systems, and database management systems). This fact provides readers with a wide range of characteristics, but makes it difficult to identify cross-cutting concerns tackled by such different perspectives and their respective assumptions. The taxonomy presented in this paper aims at encompassing the overall classifications considered by other proposals, more fully capturing the unique characteristics of provenance systems. In this way, aspects considered by existing works for having hight interest on the provenance field, constitute an integral part of our taxonomy. As a result, our taxonomy focuses on the different directions in the provenance area including *general aspects*, *data capture*, *data access*, *subject*, *storage* and *non-functional requirements* of provenance systems.

Our taxonomy has also considered the impact of provenance along a wide number of different types of application domains, where provenance research is enabling new applications. These applications can range from concrete ones to more general uses. In the former case, examples such as in science collaboratories, where researchers are provided with new techniques and tools to share provenance information at large scale, or in the scientific exploration context, where "wisdom of the crowds" can avoid duplication, and encourage continuous, documented, and reproducible scientific progress [6]. In the latter case, it is worth remarking new trending application domains where provenance is being applied, such as Sensor Networks [20, 22], Linked Data [121, 122], or specially, Big Data [19, 123, 135]. In particular, a recent community white paper on the challenges and opportunities of Big Data [135] identifies provenance as one of the most important requirements for Big Data applications in the future. This white paper states that provenance should be included to facilitate the management of data in Big Data contexts, which may be noisy together with heterogeneous and unalike data. In Wireless Sensor Networks (WSN), provenance becomes crucial [20, 22] for issues such as assessing the trustworthiness of the received data, performance optimization, diagnosing network failures or detecting early signs attacks [20, 22].

Focusing on the insights provided by the overall review, we consider that the taxonomy together with the analysis of the different provenance systems give a valuable insight into different views of provenance. Additionally, they could be very helpful in identifying similarities and differences between provenance support and provenance systems.

Aimed at helping the interested reader guide on their decision about a satisfactory provenance system, we consider that the benefits provided by the use of a particular provenance solution depend on the real interests, needs and expectations of potential users which, at the same time, ultimately rely on their domain of application. Similarly, all the work presented in this paper (the taxonomy together with the background, advantages and disadvantages of the techniques tackled to address the provenance characteristics, and the analysis of 25 systems based on the taxonomy) could be used to support a final decision. The following discussion aims at shedding more light on the issue, by going through the different dimensions of our taxonomy.

General Aspects. Although freedom programming is one of the hallmarks of computer science, the idea of encapsulating the computational methodology followed to conduct an experiment by a system is particularly remarkable (specially when users are not familiar with programming languages). This fact has led to the development of several systems based on the *WfMS-based* approach. Examples include [S1] Vistrails or [S2] myGrid/Taverna.

Data capture. Section 3.3.2 reflects differences in the approaches followed by the systems to data capture. Special attention should be given to the *tracing*, *level*, and *technique* categories. In the realm of *tracing*, the *eager* solution is suitable when re-engineering could be done, and the *lazy* one when re-engineering could not be done but techniques for reasoning about the results could be applied. It means that some systems following a *lazy* approach, such as [S8] ZOOM and [S9] Cui 2000, implement complex techniques for reasoning about the results. In the setting of *level*, the *OS* approach (e.g. [S4] PASS and [18] ES3) collects provenance information for all executions, not just those performed in the context of the *workflow*. In fact, such information may provide much irrelevant information and no particular nuances about an application. However, a great advantage is that *OS* approach does not require applications to be modified. As for the *technique*, both *annotation* and *inversion* approaches (see Figure 4) have some advantages and disadvantages. Whilst using the *annotation* approach is useful when the source data is unavailable after transformation, it takes more time and space for executing and storing the information than inversion approach. In fact, *inversion* approach does not have any performance or storage overhead during data transformation. However, it cannot compute provenance when source is unavailable after its transformation.

Data access. Systems which allow users to access to provenance data provide them with the possibility to extract information from such data. However, we can distinguish between two main stakeholders: users which lack query languages skills (*usual* users), and users familiar with specific query languages (*developers*). In case a user considers *querying formulation* for provenance retrieval as crucial, and such a user does not have an exact idea of what information wants to query, then choosing an *exploratory* approach would be the most suitable option. Additionally, the user does not require to be familiar with a specific query/programming language, so it is suitable for the first type of stakeholders. On the contrary, if the user knows precisely what information might want to query, a *directed* approach is more effective. However, it should be taken into account that it is mandatory to be knowledgeable about a specific query language, thus it is more recommendable for the second type of stakeholders. This fact makes systems such as [S1] Vistrails and [S8] ZOOM suitable for usual users, since they provide an *exploratory* approach for querying; and systems such as [S5] Trio and [S24] PLUS for developers, thus they require knowledge about the TriQL and SQL languages, respectively.

Subject. Within this dimension, it is worth noting the enormous effort performed by the provenance community to develop a standard model for provenance data exchange. Existing attempts, such as the Provenance Challenges series, shown the importance of providing such an exchange. Actually, the interoperability of a system by means of PROV could be seen as a mandatory capability in Linked Data context. As we will see in the next section, adding PROV capabilities is considered an open problem in provenance. In fact, examples include systems such as [S3] Kepler provenance, [S6] Karma or [S24] PLUS. We also consider remarkable the importance of *granularity* when the cost of extracting and storing provenance is crucial since, as stated previously, the cost can be inversely proportional to the granularity.

Storage. Traditionally, provenance systems were the focus of research in the database field (e.g. [S7] Chimera, [S9] Cui 2000, [S12] Tioga among others) which relied on relational databases. These systems usually follow two approaches: (1) they use a database just only for provenance, and another database for primary data (e.g. [S7] Chimera and [S12] Tioga); or (2) they use the same database for data and provenance, but logically separated (e.g. [S9] Cui 2000). In other words, they usually follow a *no-coupling* or *loose-coupling* approach.

Over the years, persistence systems have evolved, emerging new approaches which rely their data storage on a wide variety of persistence systems based on, for example, XML or noSQL. This is the case of [S23] BURRITO which relies on MongoDB (noSQL), based on two main reasons: (1) it is easy to serialyze Python objects into the JSON format that MongoDB requires and, (2) MongoDB's schema-free nature makes it easy to add new fields.

Non-functional. We note the importance of *security and privacy* in Big Data and Linked Data contexts. In terms of Big Data, in [125] it is claimed that we do not only should be focused on the way of capturing provenance, but the provenance records themselves should be securely stored and immutable to change. In fact, the provenance data is pointless for audit evidence if security issues have not been taken into account. In the Linked Data context, a large amount of data is published on the Web through large interlinked datasets.

Thus, applications that use this data should address the evaluation of qualities (e.g. reliability and trustworthiness) of the data retrieved from the Web. Since one of the main factors that influence the trust of users in Web content is provenance, a useful approach for data quality assessment is the analysis of provenance data [136].

Focusing on *reproducibility* (see Figure 6), it is remarkable that users should pay attention when a system makes use of third party web services in the composition of their workflow, since some workflows could rely on the availability of such services. Thus, an unavailable service means that the workflow cannot run. This issue has been identified in [S2] myGrid/Taverna by [137], but it can be extended to other systems which use external services such as [S7] Chimera and [S11] PreServ.

Finally, although the identified provenance systems show to be under active research, some of them do not support several aspects considered in our taxonomy or do not mention anything about them. Among such aspects, it is worth noting that *interoperability/exchange* (see in Figure 5) and *security/privacy* (see in Figure 6) stand out for being the least supported aspects. The *purpose* (see in Table 9), the *abstraction* (see in Figure 5) and finally, again the *security/privacy* (see in Figure 6), spotlight for being the scarcest mentioned aspects. This lack of support shows that, despite the high interest on provenance, it still constitutes an open field due to its complexity, and that there exist several challenges and future directions, which we describe next.

4.2. Future Directions

Next, we introduce some noticeable open problems:

Security. Dealing with provenance data has particular implications on data security but, as we show in the review, this issue still needs to be deeply investigated [1, 5, 3]. As stated in [1], systems with provenance capabilities fundamentally require mechanisms to manage and retrieve provenance under different access policies. However, as mentioned before, most implementations assume that security (as well as privacy) relies on the host system or file-system model, and they do not consider untrusted environments or user mistakes. This problem has been tackled for example in [138], which is based on a reference monitor so the security of the collected provenance does not rely on the system. Although this solution is feasible [1], there is no practical implementation.

On the other hand, there are specific contexts such as Big Data [19, 125] and Linked Data [77] where provenance specially constitutes, not only a security breach but also a privacy separation, for data target sources. In particular, in these contexts, providing secure provenance data techniques is still an open problem. A recent theoretical conceptual framework towards the secure storage of provenance records of the data and the audit workflows themselves in Hadoop/MapReduce applications, is presented in [125].

Confidentiality is also a current exploratory field, especially in contexts such as Big Data [19, 125], and Linked Data [77]. While in some situations the entire set of information has to be withheld from unauthorized users due to access control or lack of authorization, in provenance is different [21]. Since provenance information is crucial in deciding whether information can be trusted, omitting certain parts of the information gives users a wrong impression of the provenance information provided. However, there is still a need for new techniques and mechanisms that uphold confidentiality, and still provide users an estimated view of the provenance [21]. In the particular context of Linked Data, recent work is being developed to provide a conditional access Linked Data framework that allows users to access on the data based on their roles and privileges [77].

Privacy–preserving provenance. Respecting privacy while revealing provenance also constitutes an issue that requires further research [5, 1]. In the particular case of Big Data [19, 125] and Linked Data [77], smart control on provenance management has a direct privacy implication, that still constitutes a critical issue for both Big Data and Linked Data provenance techniques.

In the particular context of Big Data, as stated in [19], an alternative to achieve both security and privacy–related challenges could be to accept a compromise among privacy and security of data sources from a side, and provenance of data sources from the other side.

Reliability. Using provenance information as decision support, largely depends upon the trustworthiness of provenance [18]. Provenance data reliability could be ensured by providing assertions such as that the information has not been tampered with, that digital signatures have been used to sign provenance or that the multiple versions of truth involved in the data derivation have been mediated. Providing with suitable ways to assert provenance information's truthfulness needs study for it to be usable for decision making [12, 18, 22]. It is worth noting that although it would enhance its value, it in turn will promote a wider collection of provenance, requiring extra attention [18].

Provenance retrieval. The way by which the information is provided to the final user bears a strong relation with his/her knowledge about provenance and the degree of specialization required in the response. In spite of the research carried out so far towards querying and visualizing provenance, these are still challenging problems [1, 5]. More specifically, the heterogeneous techniques used by the surveyed systems towards querying and visualizing provenance denote that provenance retrieval is still an open problem that has not reached a general solution. It remains to be seen how existing solutions could be generalized, or whether totally different representations are required [1].

On the other hand, although provenance is regularly used to detect the lineage and the derivation of data and data objects, alternative uses could produce additional benefits. More specifically, next-generation cybersecurity systems could track records generated by a particular person in a specific research lab or detect the confidentiality of tracked records, relying on flexible provenance query tools [19].

Finally, dealing with Big Data deserves extra attention. Applying the classical data provenance techniques and methods for provenance retrieval introduces severe drawbacks when these techniques run over Big Data because of the prominent enormous-in-size data set which characterizes the Big Data field. More specifically, one of the major challenges when dealing with Big Data is scalability, which appears not only when analyzing Big Data using classical provenance techniques, but also when dealing with provenance of Big Data [19]. Additionally, Big Data provenance requires not only interactive intelligent tools for visualizing actual results and supporting next-step decisions, but also accessing and querying Big Data, even in an interactive manner. Both aspects, provenance visualization and query optimization tools, is a crucial open problem at now [19].

Knowledge extraction. Mining [139, 140] and extracting knowledge from provenance constitutes a tough problem that has been largely unexplored [7, 123].

Mining the data resulted from analyzing and creating insightful visualizations of provenance data may help discover workflows patterns. These patterns could potentially simplify the tedious, complicated and time-consuming process of designing and refining scientific workflows [7]. In the particular case of Linked Data, several proposals for mining data are summarized in [129] in which the user needs to acquire specific domain knowledge about schema names by browsing the datasets individually. To solve this issue, authors in [129] suggest to use a *schema level mapping provenance*, that is, the mappings between concepts of different datasets are done between schemas instead of between data sets. Using this kind of *schema level mapping* would also contribute to benefit Linked Data integration, quality and trustworthiness [129].

Standardisation. Most of the surveyed systems have their own model for managing provenance (see *interoperability/exchange* category in Figure 5) and the absence of standards for collecting, representing, storing and querying provenance is a hindrance to promoting interoperability [18]. Aimed at solving this problem, the provenance community has made a huge effort to set up a cross domain standard for provenance. It started with the 3PC which sought to the interoperability among systems by means of OPM and nowadays, using the PROV standard, as a way for interchange provenance information. It is clear that both performing further research work tackling the evolution of standards, and adding PROV capabilities within existing provenance systems will contribute to promoting the sharing and use of provenance information [18, 7].

Computational overhead. Another remarkable aspect is related to computational overhead. Classical provenance data techniques and methods are usually data-intensive, resource and time-consuming [19, 1]. It forces the need for providing new techniques that introduce a minimum computational overhead [19]. As advanced previously, aspects such as *granularity* may incur temporal and spatial overhead [1], being granularity inversely proportional to the amount and cost of provenance information [13]. Another aspect is *querying* [1, 22]; when some systems delay provenance construction in order to minimize capture overhead (such as [S4] PASSv2 or [S15] SPADEv2), others delay provenance collection to query time to avoid wasting resources computing provenance that will never be accessed (e.g. [S21] LipStick performs provenance construction only when a query is made, and [S8] ZOOM computes some of the provenance at query time, based on the current user view). As suggested by [1], to choose the most suitable system, it may be useful to predetermine what provenance information will be required to answer queries and at what granularity this information will be sufficient, mapping it to the appropriate system. Other trends aim to use techniques for data deduplication and compaction to avoid storage overhead, while adding time penalties [20, 5]. It is still a challenge to find a solution which finds a balance between both storage and computational problems [20]. Examples of current application domains where this issue is specially considered are extra large-scale WSN [22, 20], or Big Data provenance [19]. In WSN provenance research is being performed to answer domain specific complex queries [20]. Given the distributed nature of Big Data Workflows, minimizing the collection overhead during workflow execution gets specially hard, which is a cause of current work [124].

Managing provenance. The growing amount of raw data and provenance information motivates the need for providing efficient and effective techniques to manage these data [7, 123]. Provenance systems, as information management systems, are especially hard to use. Thus, addressing the usability of these systems is of paramount importance [7] given their growing acceptance and need in a wide range of applications [4, 5, 7]. There are specific contexts, such as WSN, where handling the provenance data constitutes a relevant challenge to be considered [20, 22]. Given the limited network bandwidth, an open problem in this context is the need for data to be small in size. Additionally, the provenance data operations must be performed simultaneously with the data flow which, together with the huge data flow, makes of provenance creation an issue [20].

Integration. Integration is another problem to face with when dealing with provenance. Not only does data integration a challenge, but also the multi-system nature of provenance systems and the need to integrate provenance with existing domain-specific systems. More specifically, provenance data could be captured from various sources which provide distinct provenance data [20, 2]. Domain objects can be distributed over multiple systems, layers or distributed repositories. Thus, the same item can be represented in two systems by two different objects, but the provenance system must be able to integrate both identities in order to get a unified view on the data [20, 2]. Even worse, some objects could have a definition in one system or layer, and no representation at all in another system or layer, which makes confusing to which objects these unmatching objects must be connected [20]. Designing an interface facing all these challenges would be very feasible since most of the provenance systems work in one layer and do not handle cross-layer applications [20]. In this context, recognizing duplication of data items (or even versions) is also an issue that has been tackled through the use of *semantic identity* [2]. For example, there have been attempts to extend provenance to existing systems, such as networked systems or wireless sensor networks, but problems related to heterogeneity (not all nodes being provenance-aware), among other issues, remain a hard task [1, 20]. Additionally, tracking the provenance of objects when they are out of the boundary of a system has been already tackled but still with various flaws which need to be addressed [21].

Other open research questions require further consideration. For example, *phantom lineage*, which refers to store provenance about data even after it was deleted or to trace the reason why a specific data was deleted (as opposed to how it was created) [12, 18]. Using provenance to efficiently trace databases versions whose schemas evolve over time, constitutes another research direction [3, 15].

5. Conclusions

In this paper we provide a systematic literature review of peer-reviewed published studies that focus on research issues in provenance systems. We highlight the three main contributions provided by this review as: (1) establishing a general taxonomy scheme of provenance characteristics that enables to distinguish between different kinds of provenance and that could lead to a better understanding of provenance in general (*RQ1*), (2) providing a consistent background of provenance concepts and fundamental techniques in provenance which provides the necessary knowledge to understand the features identified in our taxonomy (*RQ2*), and (3) exhaustively analysing 25 provenance systems attending to our taxonomy, showing in detail how provenance is supported in these systems (*RQ3*).

It is our belief that this work can provide several benefits. First, this work can help inform and enhance the understanding of provenance. Second, it can help general users distinguish between different perspectives of provenance and guide them in their decision about the most suitable solution according to their needs. Third, this work can inform about up-to-date approaches and technologies available, as well as open problems being seen in practice. Finally, the results provided can help potential researchers aimed at identifying provenance research issues that have been already tackled or directions for future research.

Acknowledgements.

This work has been partially supported by the Spanish Ministerio de Economía y Competitividad (project MTM2014-54151-P and Project EDU2016-79838-P), and by the University of La Rioja (grant FPI-UR-2015).

References

[1] L. Carata, Sherif Akoush, Nikilesh Balakrishnan, et al. A Primer on Provenance. *Communications of the ACM*, 12(3):10:10–10:23, March 2014.

[2] Boris Glavic and Klaus R. Dittrich. Data Provenance: A Categorization of Existing Approaches. In *Proc. of Datenbanksysteme in Business, Technologie und Web (BTW'07)*, pages 227–241, 2007.

[3] Wang Chiew Tan. Provenance in Databases: Past, Current, and Future. *IEEE Data Eng. Bull.*, 30(4):3–12, 2007.

[4] Luc Moreau. The Foundations for Provenance on the Web. *Foundations and Trends in Web Science*, 2(2–3):99–241, 2010.

[5] P. Buneman and S. B. Davidson . Data provenance – the foundation of data quality. Available at: www.sei.cmu.edu/measurement/research/upload/Davidson.pdf. Last visited on December 2017.

[6] Juliana Freire, David Koop, Emanuele Santos, and Cláudio T. Silva. Provenance for Computational Tasks: A Survey. *Comput. Sci. Eng.*, 10(3):11–21, May 2008.

[7] Susan B. Davidson and Juliana Freire. Provenance and scientific workflows: challenges and opportunities. In *Proc. of MOD'08*, pages 1345–1350, 2008.

[8] Reagan Moore, Arun Jagatheesan, Arcot Rajasekar, et al. Data grid management systems. In *Proc. of the 21st IEEE Conference on Mass Storage Systems and Technologies (MSST04)*, pages 1–15. IEEE, 2004.

[9] Boris Glavic. *Perm: Efficient Provenance Support for Relational Databases*. PhD thesis, University of Zurich, 2010. Available at: http://www.zora.uzh.ch/44573/1/dissGlavic.pdf. Last visited on December 2017.

[10] Paul Groth, Michael Luck, and Luc Moreau. A protocol for recording provenance in service-oriented grids. In *Proc. of the 8th International Conference on Principles of Distributed Systems (OPODIS'04)*, pages 124–139. Springer-Verlag, 2005.

[11] Susan B. Davidson, Sarah Cohen-Boulakia, Anat Eyal, et al. Provenance in Scientific Workflow Systems. *IEEE Data Eng. Bull.*, 30(4):44–50, 2007.

[12] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A Survey of Data Provenance in e-Science. *SIGMOD Rec.*, 34(3):31–36, 2005.

[13] Sérgio Manuel Serra da Cruz, Maria Luiza Machado Campos, and Marta

Mattoso. t. In *Proc. of the IEEE Congress on Services, Part I, SERVICES I*, pages 259–266, 2009.

[14] Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. Why and Where: A Characterization of Data Provenance. In *Proc. of the 8th Intl. Conf. on Database Theory (ICDT)*, pages 316–330, 2001.

[15] Peter Buneman and Wang Chiew Tan. Provenance in Databases. In *Proc. of MOD'07*, pages 1171–1173. ACM, 2007.

[16] James Cheney, Laura Chiticariu, and Wang Chiew Tan. Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases*, 1(4):379–474, April 2009.

[17] Rajendra Bose and James Frew. Lineage retrieval for scientific data processing: a survey. *ACM Comput. Surv.*, 37(1):1–28, 2005.

[18] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A Survey of Data Provenance Techniques. Technical Report 612. Extended version of SIGMOD Record 2005. Available at: http://www.cs.indiana.edu/pub/techreports/TR618.pdf. Last visited on December 2017.

[19] Alfredo Cuzzocrea. Big data provenance: State-of-the-art analysis and emerging research challenges. In *Proc. of the Workshops of the EDBT/ICDT*, pages 797–800, 2016.

[20] Gulustan Dogan. A survey of provenance in wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 30(1-2):21–45, 2016.

[21] Yu Shyang Tan, Ryan K. L. Ko, and Geoff Holmes. Security and data accountability in distributed systems: A provenance survey. In *Proc. of HPCC'13*, pages 1571–1578. IEEE Computer Society, 2013.

[22] Changda Wang, Wen-yi Zheng, and Elisa Bertino. Provenance for wireless sensor networks: A survey. *Data Science and Engineering*, 1(3):189–200, 2016.

[23] B.A. Kitchenham. Procedures for Performing Systematic Reviews. Technical Report TR/SE-0401 (2004), Keele University. At: http://www.inf.ufsc.br/ aldo.vw/kitchenham.pdf. Last visited on December 2017.

[24] B. Kitchenham and S. Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. (EBSE 2007–01). At: http://pages.cpsc.ucalgary.ca/∼sillito/cpsc-601.23/readings/kitchenham-2007.pdf. Last visited on December 2017.

[25] B. Kitchenham, O. Pearl Brereton, D. Budgen, et al. Systematic Literature Reviews in Software Engineering - A Systematic Literature Review. *Inf. Softw. Technol.*, 51(1):7–15, January 2009.

[26] He (Jason) Zhang and Muhammad Ali Babar. An empirical investigation of systematic reviews in software engineering. In *Proc. of ESEM'11*, pages 87–96, Banff, Canada, sep 2011. IEEE.

[27] R. E. S. Santos and F. Q. B. da Silva. Motivation to perform systematic reviews and their impact on software engineering practice. In *Proc. of ESEM'13*, pages 292–295, 2013.

[28] Supplementary material of the Systematic Review. Available at: http://www.unirioja.es/cu/beperev/SupplementaryMaterial.html. Last visited on December 2017.

[29] Louis Bavoil, Steven P Callahan, Patricia J Crossno, et al. Vistrails: Enabling interactive multiple-view visualizations. In *Proc. of the IEEE Visualization (VIS'05)*, pages 135–142. IEEE, 2005.

[30] Juliana Freire, Cláudio T Silva, Steven P Callahan, et al. Managing rapidly-evolving scientific workflows. In *Proc. of IPAW'06*, pages 10–18. Springer, 2006.

[31] David Gammack, Steve Scott, and Adriane P Chapman. Modelling provenance collection points and their impact on provenance graphs. In *Proc. of IPAW'16*, pages 146–157, 2016.

[32] Fernando Chirigati, Juliana Freire, David Koop, and Cláudio Silva. Vistrails provenance traces for benchmarking. In *Proc. of the Joint EDBT/ICDT 2013 Workshops*, pages 323–324, 2013.

[33] Carlos E Scheidegger, Huy T Vo, et al. Querying and re-using workflows with vistrails. In *Proc. of MOD'08*, pages 1251–1254, 2008.

[34] P. Missier, Stian Soiland-Reyes, Stuart Owen, et al. Taverna, reloaded. In *Proc. of the International conference on scientific and statistical database management (SSDBM'10)*, pages 471–481, 2010.

[35] Katherine Wolstencroft, Robert Haines, Donal Fellows, et al. The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic acids research*, pages 557–561, 2013.

[36] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R. Pocock, Anil Wipat, and Peter Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.

[37] Pinar Alper, Khalid Belhajjame, and Carole A Goble. Static analysis of taverna workflows to predict provenance patterns. *Futur Gener Comput Syst*, 75:310–329, 2017.

[38] Ilkay Altintas, Oscar Barney, and Efrat Jaeger-Frank. Provenance Collection Support in the Kepler Scientific Workflow System. In *Proc. of IPAW'06*, pages 118–132, 2006.

[39] Daniel Crawl and Ilkay Altintas. A provenance-based fault tolerance mechanism for scientific workflows. In *Proc. of IPAW'08*, pages 152–159, 2008.

[40] Kiran Kumar Muniswamy-Reddy, David A Holland, Uri Braun, and Margo I Seltzer. Provenance-Aware Storage Systems. In *USENIX Annual Technical Conference, General Track*, pages 43–56, 2006.

[41] K. K. Muniswamy-Reddy, Uri Braun, David A. Holland, et al. Layering in Provenance Systems. In *USENIX Annual Technical Conference*, 2009.

[42] David A Holland, Margo I Seltzer, Uri Braun, and Kiran Kumar Muniswamy-Reddy. PASSing the provenance challenge. *Concurr Comput*, 20(5):531–540, 2008.

[43] Jennifer Widom. Trio: A System for Integrated Management of Data, Accuracy, and Lineage. Technical Report 2004-40, Stanford InfoLab. Available at: http://ilpubs.stanford.edu:8090/658/. Last visited on December 2017.

[44] Parag Agrawal, Omar Benjelloun, Anish Das Sarma, et al. Trio: A System for Data, Uncertainty, and Lineage. In *Proc. of VLDB'06*, pages 1151–1154, September 2006.

[45] Omar Benjelloun, Anish Das Sarma, Chris Hayworth, and Jennifer Widom.

An Introduction to ULDBs and the Trio System. Technical Report 2006-7, Stanford InfoLab. Avail. at: http:// ilpubs.stanford.edu:8090/793/. Last visited on December 2017.

[46] Michi Mutsuzaki, Martin Theobald, and et al. Trio-One: Layering Uncertainty and Lineage on a Conventional DBMS. In *Proc. of CIDR'07*, pages 269–274, 2007.

[47] Jennifer Widom. Trio: A System for Data, Uncertainty, and Lineage. In *Managing and Mining Uncertain Data*. Springer, 2008.

[48] Parag Agrawal, Robert Ikeda, Hyunjung Park, and Jennifer Widom. Trio-ER: The Trio System as a Workbench for Entity-Resolution. Technical report (march 2009), Stanford University. Avail. at: http:// ilpubs.stanford.edu:8090/912/. Last visited on December 2017.

[49] Yogesh L Simmhan, Beth Plale, Dennis Gannon, and Suresh Marru. Performance evaluation of the karma provenance framework for scientific workflows. In *Proc. of IPAW'06*, pages 222–236, 2006.

[50] Yogesh L Simmhan, Beth Plale, and Dennis Gannon. A framework for collecting provenance in data-centric scientific workflows. In *Proc. of the International Conference on Web Services (ICWS'06)*, pages 427–436. IEEE, 2006.

[51] Mohammad Rezwanul Huq, Andreas Wombacher, and Peter MG Apers. Inferring fine-grained data provenance in stream data processing: reduced storage cost, high accuracy. In *Proc. of DEXA'11*, pages 118–127, 2011.

[52] Yogesh L Simmhan, Beth Plale, and Dennis Gannon. Karma2: Provenance management for data-driven workflows. *Web Services Research for Emerging Applications: Discoveries and Trends: Discoveries and Trends*, 317, 2010.

[53] Ian Foster, Jens Vöckler, Michael Wilde, and Yong Zhao. Chimera: A virtual data system for representing, querying, and automating data derivation. In *Proc. of SSDBM'02*, pages 37–46. IEEE, 2002.

[54] Ben Clifford, Ian Foster, Jens-S Voeckler, et al. Tracking provenance in a virtual data grid. *Concurr Compute*, 20(5):565–575, 2008.

[55] Yong Zhao, Michael Wilde, and Ian Foster. Applying the virtual data provenance model. In *Proc. of IPAW'06*, volume 4145 of *LNCS*, pages 148–161. Springer, 2006.

[56] Olivier Biton, Sarah Cohen-Boulakia, and Susan B Davidson. Zoom*UserViews: Querying relevant provenance in workflow systems. In *Proc. of VLDB'07*, pages 1366–1369. VLDB Endowment, 2007.

[57] Sarah Cohen-Boulakia, Olivier Biton, Shirley Cohen, and Susan Davidson. Addressing the Provenance Challenge using ZOOM. *Concurr Comput*, 20 (5):497–506, 2008.

[58] Olivier Biton, Sarah Cohen-Boulakia, Susan B Davidson, and Carmem S Hara. Querying and managing provenance through user views in scientific workflows. In *Proc. of the IEEE 24th International Conference on Data Engineering (ICDE'08)*, pages 1072–1081. IEEE, 2008.

[59] James Cheney and Roly Perera. An analytical survey of provenance sanitization. In *Proc. of IPAW'14*, pages 113–126, 2014.

[60] Yingwei Cui and Jennifer Widom. Lineage tracing in data warehouses.

In *Proc. of the 16th International Conference on Data Engineering*, pages 367–378. IEEE, 2000.

[61] Yingwei Cui, Jennifer Widom, and Janet L Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Transactions on Database Systems (TODS)*, 25(2):179–227, 2000.

[62] Yingwei Cui and Jennifer Widom. Practical lineage tracing in data warehouses. In *Proc. of the 16th International Conference on Data Engineering (ICDE'00)*, pages 367–378. IEEE, 2000.

[63] Janet Wiener, Himanshu Gupta, Wilburt Labio, et al. A system prototype for warehouse view maintenance. In *Proc. of MOD'95*, pages 26–33, 1995.

[64] Yong Zhao, M Hategan, and et al. Swift: Fast, Reliable, Loosely Coupled Parallel Computation. In *IEEE International Conference on Services Computing - Workshops (SCW'07)*, pages 199–206, 2007.

[65] Luiz MR Gadelha Jr, Ben Clifford, Marta Mattoso, et al. Provenance management in Swift. *Futur Gener Comput Syst*, 27(6):775–780, 2011.

[66] Anderson Marinho, Daniel de Oliveira, Eduardo Ogasawara, et al. Deriving scientific workflows from algebraic experiment lines: A practical approach. *Futur Gener Comput Syst*, 68:111–127, 2017.

[67] Michael Wilde, Mihael Hategan, Justin M Wozniak, et al. Swift: A language for distributed parallel scripting. *Parallel Computing*, 37(9):633–652, 2011.

[68] Paul Groth, Simon Miles, and Luc Moreau. PReServ: Provenance recording for services. *UK e-Science All Hands Meeting*, 2005.

[69] Michael Stonebraker, Jolly Chen, Nobuko Nathan, et al. Tioga: Providing Data Management Support for Scientific Visualization Applications. In *Proc. of VLDB'93*, pages 25–38, 1993.

[70] Allison Woodruff and Michael Stonebraker. Supporting Fine-grained Data Lineage in a Database Visualization Environment. In *Proc. of ICDE'97*, pages 91–102, 1997.

[71] Alexander Aiken, Jolly Chen, Michael Stonebraker, and Allison Woodruff. Tioga-2: A Direct Manipulation Database Visualization Environment. In *Proc. of the Twelfth International Conference on Data Engineering (ICDE'96)*, pages 208–217, 1996.

[72] E. Deelman, Gurmeet Singh, Mei-Hui Su, et al. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, 13(3):219–237, 2005.

[73] Jihie Kim, Ewa Deelman, Yolanda Gil, et al. Provenance Trails in the Wings/Pegasus System. *Concurr Comput*, 20(5):587–597, 2008.

[74] Ewa Deelman, James Blythe, Yolanda Gil, et al. Pegasus: Mapping scientific workflows onto the grid. In *Proc. of the 2nd European Across Grids Conference (EAGC'04)*, pages 11–20. Springer, 2004.

[75] Yolanda Gil, Varun Ratnakar, Ewa Deelman, et al. Wings for pegasus: Creating large-scale scientific applications using semantic representations of computational workflows. In *Proc. of AAAI'07*, volume 22, page 1767, 2007.

[76] Ewa Deelman, Gaurang Mehta, Gurmeet Singh, et al. Pegasus: mapping large-scale workflows to distributed resources. In *Workflows for e-Science*, pages 376–394. Springer, 2007.

[77] Daniel Garijo, Yolanda Gil, and Oscar Corcho. Abstract, link, publish, exploit: An end to end framework for workflow sharing. *Futur Gener Comput Syst*, 2017.

[78] Yolanda Gil, Varun Ratnakar, Jihie Kim, et al. Wings: Intelligent workflow-based design of computational experiments. *IEEE Intelligent Systems*, 26 (1):62–72, 2011.

[79] Peter Buneman, Adriane Chapman, and James Cheney. Provenance Management in Curated Databases. In *Proc. of MOD'06*, pages 539–550, 2006.

[80] Ashish Gehani and Dawood Tariq. SPADE: support for provenance auditing in distributed environments. In *Proc. of the 13th International Middleware Conference*, pages 101–120, 2012.

[81] Ashish Gehani and Minyoung Kim. Mendel: Efficiently verifying the lineage of data modified in multiple trust domains. In *Proc. of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC'10)*, pages 227–239. ACM, 2010.

[82] Sheung Chi Chan, Ashish Gehani, James Cheney, et al. Expressiveness benchmarking for system-level provenance. In *Proc. of TaPP'17*. USENIX Association, 2017.

[83] Manolis Stamatogiannakis, Hasanat Kazmi, Hashim Sharif, et al. Trade-offs in automatic provenance capture. In *Proc. of IPAW'16*, pages 29–41, 2016.

[84] Ashish Gehani, Hasanat Kazmi, and Hassaan Irshad. Scaling spade to big provenance. In *Proc. of TaPP'16*, pages 26–33. USENIX Association, 2016.

[85] Zachary G Ives, Nitin Khandelwal, Aneesh Kapur, and Murat Cakir. ORCHESTRA: Rapid, Collaborative Sharing of Dynamic Data. In *Proc. of CIDR'05*, pages 107–118, 2005.

[86] Todd J. Green, Grigoris Karvounarakis, Zachary G. Ives, and Val Tannen. Update Exchange with Mappings and Provenance. In *Proc. of VLDB'07*, pages 675–686, 2007.

[87] Todd J. Green, Gregory Karvounarakis, Nicholas E. Taylor, et al. ORCHESTRA: facilitating collaborative data sharing. In *Proc. of MOD'07*, pages 1131–1133, 2007.

[88] Todd J Green and Val Tannen. The semiring framework for database provenance. In *Proc. of PODS'17*, pages 93–99. ACM, 2017.

[89] Zachary G Ives, Todd J Green, Grigoris Karvounarakis, et al. The orchestra collaborative data sharing system. *ACM SIGMOD Record*, 37(3):26–32, 2008.

[90] Boris Glavic, Renée J. Miller, and Gustavo Alonso. Using SQL for Efficient Generation and Querying of Provenance Information. In *In search of elegance in the theory and practice of computation: a Festschrift in honour of Peter Buneman*, pages 291–320, 2013.

[91] Boris Glavic and Gustavo Alonso. Perm: Processing Provenance and Data on the same Data Model through Query Rewriting. In *Proc. of the 25th IEEE International Conference on Data Engineering (ICDE'09)*, pages 174–185, 2009.

[92] Boris Glavic and Gustavo Alonso. Provenance for Nested Subqueries. In *Proc. of the 12th International Conference on Extending Database Technology (EDBT'09)*, pages 982–993, 2009.

[93] Boris Glavic and Gustavo Alonso. The Perm Provenance Management System in Action. In *Proc. of MOD'09 (Demonstration Track)*, pages 1055–1058, 2009.

[94] Bahareh Arab, Dieter Gawlick, Venkatesh Radhakrishnan, et al. A Generic Provenance Middleware for Database Queries, Updates, and Transactions. In *Proc. of TaPP'14*, 2014.

[95] Xing Niu, Raghav Kapoor, Boris Glavic, et al. Interoperability for Provenance-aware Databases using PROV and JSON. In *Proc. of TaPP'15*, 2015.

[96] Bahareh Arab, Dieter Gawlick, Vasudha Krishnaswamy, et al. Reenacting Transactions to Compute their ProvenA system prototype for warehouse view maintenanceance. Technical Report IIT/CS-DB-2014-02, Illinois Institute of Technology, 2014. Available at: http://cs.iit.edu/ dbgroup/pdfpubls/AD14.pdf. Last visited on December 2017.

[97] Seokki Lee, Yuchen Tang, Sven Köhler, et al. An Efficient Implementation Of Game Provenance In DBMS. Technical Report IIT/CS-DB-2015-02, Illinois Institute of Technology, 2015.

[98] Xing Niu, Raghav Kapoor, and Boris Glavic. Heuristic and Cost-based Optimization for Provenance Computation. In *Proc. of TaPP'15*, 2015.

[99] James Frew and Peter Slaughter. Es3: A demonstration of transparent provenance for scientific computation. In *Proc. of IPAW'08*, volume 5272 of *LNCS*, pages 200–207. Springer, 2008.

[100] James Frew, Dominic Metzger, and Peter Slaughter. Automatic capture and reconstruction of computational provenance. *Concurr Comput*, 20(5): 485–496, 2008.

[101] Shawn Bowers, Timothy M. McPhillips, and Bertram Ludäscher. Provenance in collection-oriented scientific workflows. *Concurr Comput*, 20(5): 519–529, 2008.

[102] Deepavali Bhagwat, Laura Chiticariu, Wang Chiew Tan, and Gaurav Vijayvargiya. An annotation management system for relational databases. *The VLDB Journal*, 14(4):373–396, 2005.

[103] Laura Chiticariu, Wang Chiew Tan, and Gaurav Vijayvargiya. DBNotes: A Post-It System for Relational Databases based on provenance. In *Proc. of MOD'05*, pages 942–944. ACM, 2005.

[104] Y. Amsterdamer, Susan B. Davidson, Daniel Deutch, et al. Putting Lipstick on Pig: Enabling Database-style Workflow Provenance. *PVLDB*, 5 (4):346–357, 2011.

[105] Roger S Barga and Luciano A Digiampietri. Automatic capture and efficient storage of e-Science experiment provenance. *Concurr Comput*, 20(5): 419–429, 2008.

[106] Philip J. Guo and Margo Seltzer. BURRITO: Wrapping Your Lab Notebook in Computational Infrastructure. In *Proc. of TaPP'12*, 2012.

[107] Philip J. Guo. *Software Tools to Facilitate Research Programming.* PhD thesis, Stanford University, May 2012.

[108] Peter Macko and Margo Seltzer. Provenance map orbiter: Interactive exploration of large provenance graphs. In *Proc. of TaPP'11*, 2011.

[109] Adriane Chapman, Barbara T Blaustein, Len Seligman, and M David Allen. Plus: A provenance manager for integrated information. In *Proc. of*

the IEEE International Conference on Information Reuse and Integration (IRI'11), pages 269–275. IEEE, 2011.

[110] Adriane Chapman, M David Allen, Barbara Blaustein, et al. Plus: Provenance for life, the universe and stuff. *VLDB'10, VLDB Endowment*, pages 13–17, 2010.

[111] Barbara Blaustein, Len Seligman, Michael Morse, et al. Plus: Synthesizing privacy, lineage, uncertainty and security. In *Proc. of the 24th International Conference on Data Engineering Workshop (ICDEW'08)*, pages 242–245, 2008.

[112] Robert Ikeda, Hyunjung Park, and Jennifer Widom. Provenance for generalized map and reduce workflows. In *Proc. of the Fifth Biennial Conference on Innovative Data Systems (CIDR'11)*, pages 273–283, 2011.

[113] Hyunjung Park, Robert Ikeda, and Jennifer Widom. Ramp: A system for capturing and tracing provenance in mapreduce workflows. *PVLDB*, 4(12): 1351–1354, 2011.

[114] Emilio Rodriguez-Priego, Francisco José García-Izquierdo, and Ángel Luis Rubio. References-enriched Concept Map: a tool for collecting and comparing disparate definitions appearing in multiple references. *J. Information Science*, 39(6):789–804, 2013.

[115] Carlos Saénz-Adán, Francisco J. García-Izquierdo, Ángel Luis Rubio, Eduardo Sáenz de Cabezón Irigaray, Emilio Rodríguez-Priego, and Oscar Díaz. A Tool for Management of Knowledge Dispersed throughout Multiple References. In *Proc. of the 10th International Conference on Software Paradigm Trends (ICSOFT-PT'15)*, pages 79–86, 2015.

[116] D.A. Holland, U. Braun, D. Maclean, et al. Choosing a data model and query language for provenance. In *Proc. of IPAW'08*, pages 98–115, 2008.

[117] Provenance Challenge Series. Available at: http://twiki.ipaw.info/bin/view/Challenge/. Last visited on December 2017.

[118] Luc L. Moreau, B. Clifford, Juliana Freire, et al. The open provenance model core specification (v1. 1). *Futur Gener Comput Syst*, 27(6):743–756, 2011.

[119] PROV-Overview. An Overview of the PROV Family of Documents. Available at: https://www.w3.org/TR/prov-overview/. Last visited on December 2017.

[120] Sean Bechhofer, David De Roure, Matthew Gamble, Carole Goble, and Iain Buchan. Research objects Towards exchange and reuse of digital knowledge. *The Future of the Web for collaborative science. Raleigh: nature Precedings*, 2010.

[121] Jun Zhao, Alistair Miles, Graham Klyne, and David M. Shotton. Linked data and provenance in biological data webs. *Briefings in Bioinformatics*, 10(2):139–152, 2009.

[122] Olaf Hartig and Jun Zhao. Publishing and consuming provenance metadata on the web of linked data. In *Proc. of IPAW'10*, pages 78–90, 2010.

[123] Boris Glavic. Big data provenance: Challenges and implications for benchmarking. In *First Workshop of Specifying Big Data Benchmarks, (WBDB'12)*, pages 72–80, 2012.

[124] Jianwu Wang, Daniel Crawl, Shweta Purawat, et al. Big data provenance:

Challenges, state of the art and opportunities. In *Proc. of the IEEE International Conference on Big Data, Big Data*, pages 2509–2516, 2015.

[125] Deniz Appelbaum. Securing big data provenance for auditors: The big data provenance black box as reliable evidence. *Journal of Emerging Technologies in Accounting*, 13(1):17–36, 2016.

[126] Sherif Akoush, Ripduman Sohan, and Andy Hopper. Hadoopprov: Towards provenance as a first class citizen in mapreduce. In *Proc. of TaPP'13*, 2013.

[127] Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance Semirings. In *Proc. of PODS'07*, pages 31–40. ACM, 2007.

[128] Grigoris Karvounarakis and Todd J Green. Semiring-annotated data: queries and provenance? *ACM SIGMOD Record*, 41(3):5–14, 2012.

[129] Sarawat Anam, Byeong Ho Kang, Yang Sok Kim, and Qing Liu. Linked data provenance: State of the art and challenges. In *Proc. of the Australasian Web Conference (AWC'15)*, volume 166 of *CRPIT*, pages 19–28. Australian Computer Society, 2015.

[130] Graphviz, Graph Visualization Software. Available at: www.graphviz.org. Last visited on December 2017.

[131] W3C PROV Implementation Survey. Available at: https://www.w3.org/2002/09/wbs/46974/prov-implementation-survey/results. Last visited on December 2017.

[132] Szymon Klarman, Stefan Schlobach, and Luciano Serafini. Formal Verification of Data Provenance Records. In *Proc. of the 11th International Semantic Web Conference (ISWC'12) Part I*, pages 215–230, 2012.

[133] Luc Moreau, Trung Dong Huynh, and Danius Michaelides. An online validator for provenance: algorithmic design, testing, and API. In *Proc. of the FASE'14*, pages 291–305. Springer, 2014.

[134] Hasnain Lakhani, Rashid Tahir, Azeem Aqil, et al. Optimized rollback and re-computation. In *Proc. of HICSS'13*, pages 4930–4937, 2013.

[135] D. Agrawal, Philip Bernstein, Elisa Bertino, et al. Challenges and opportunities with big data. a community white paper developed by leading researchers across the united states. *Computing Research Association*, 2012.

[136] Olaf Hartig. Provenance information in the web of data. In *Proc. of the Linked Data on the Web (LDOW)*, 2009.

[137] Sarah Cohen Boulakia, Khalid Belhajjame, et al. Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. *Futur Gener Comput Syst*, 75:284–298, 2017.

[138] Patrick McDaniel, Kevin RB Butler, Stephen E McLaughlin, et al. Towards a secure and efficient system for end-to-end provenance. In *Proc. of TaPP'10*, 2010.

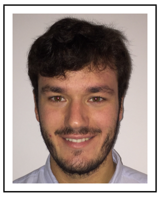
[139] Jiawei Han. *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. ISBN 1558609016.

[140] Wil M. P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer Publishing Company, Incorporated, 1st edition, 2011. ISBN 3642193447, 9783642193446.

Author Biographies

Beatriz Pérez is an Assistant Professor in Computer Science at the University of La Rioja, Spain. She received her B.Sc. degree in mathematics from the University of La Rioja, Spain, in 2003, and her Ph.D. in Computer Science from the University of Zaragoza in 2011. Her main research interests are in Meta-modelling, Model-Driven Development, Business process modeling and provenance.

Julio Rubio is a full professor of Computer Science and Artificial Intelligence at University of La Rioja. His main research topics are Symbolic Computation and Formal Methods.



Carlos Sáenz-Adán is a Ph.D. student in Computer Science at the University of La Rioja (UR), Spain. He received his B.Sc. degree in Computer Science from the University of La Rioja in 2013. In 2014, he received a Master degree in Advanced Computer Systems from the University of Pas Vasco (UPV/EHU). His research, which is partially supported by the University of La Rioja grant (FPI-UR-2015), is mainly focused on Business process modeling and provenance.

Correspondence and offprint requests to: Beatriz Pérez, Department of Mathematics and Computer Science, University of La Rioja, E-26004 – La Rioja, Spain. Email: beatriz.perez@unirioja.es