The Foundations for Provenance on the Web

Luc Moreau

November 11, 2010

Foundations and Trends in Web Science
© 2010 L. Moreau
DOI: 10.1561/1800000010
Abstract

Provenance, i.e., the origin or source of something, is becoming an important concern, since it offers the means to verify data products, to infer their quality, to analyse the processes that led to them, and to decide whether they can be trusted. For instance, provenance enables the reproducibility of scientific results; provenance is necessary to track attribution and credit in curated databases; and, it is essential for reasoners to make trust judgements about the information they use over the Semantic Web.

As the Web allows information sharing, discovery, aggregation, filtering and flow in an unprecedented manner, it also becomes very difficult to identify, reliably, the original source that produced an information item on the Web. Since the emerging use of provenance in niche applications is undoubtedly demonstrating the benefits of provenance, this survey contends that provenance can and should reliably be tracked and exploited on the Web, and investigates the necessary foundations to achieve such a vision.

Multiple data sources have been used to compile the largest bibliographical database on provenance so far. This large corpus permits the analysis of emerging trends in the research community. Specifically, the CiteSpace tool identifies clusters of papers that constitute research fronts, from which characteristics are extracted to structure a foundational framework for provenance on the Web. Such an endeavour requires a multi-disciplinary approach, since it requires contributions from many computer science sub-disciplines, but also other non-technical fields given the human challenge that is anticipated.

To develop such a vision, it is necessary to provide a definition of provenance that applies to the Web context. A conceptual definition of provenance is expressed in terms of processes, and is shown to generalise various definitions of provenance commonly encountered. Furthermore, by bringing realistic distributed systems assumptions, this definition is refined as a query over assertions made by applications.

Given that the majority of work on provenance has been undertaken by the database, workflow and e-science communities, some of their work is reviewed, contrasting approaches, and focusing on important topics believed to be crucial for bringing provenance to the Web, such as abstraction, collections, storage, queries, workflow evolution, semantics and activities involving human interactions.

However, provenance approaches developed in the context of databases and workflows essentially deal with closed systems. By that, it is meant that workflow or database management systems are in full control of the data they manage, and track their provenance within their own scope, but not beyond. In the context
of the Web, a broader approach is required by which chunks of provenance representation can be brought together to describe the provenance of information flowing across multiple systems. For this purpose, this survey puts forward the Open Provenance Vision, which is an approach that consists of controlled vocabulary, serialization formats and interfaces to allow the provenance of individual systems to be expressed, connected in a coherent fashion, and queried seamlessly. In this context, the Open Provenance Model is an emerging community-driven representation of provenance, which has been actively used by some twenty teams to exchange provenance information, in line with the Open Provenance Vision.

After identifying an open approach and a model for provenance, techniques to expose provenance over the Web are investigated. In particular, Semantic Web technologies are discussed since they have been successfully exploited to express, query and reason over provenance. Symmetrically, Semantic Web technologies such as RDF, underpinning the Linked Data effort, are analysed since they offer their own difficulties with respect to provenance.

A powerful argument for provenance is that it can help make systems transparent, so that it becomes possible to determine whether a particular use of information is appropriate under a set of rules. Such capability helps make systems and information accountable. To offer accountability, provenance itself must be authentic, and rely on security approaches, which are described in the survey. This is then followed by systems where provenance is the basis of an auditing mechanism to check past processes against rules or regulations. In practice, not all users want to check and audit provenance, instead, they may rely on measures of quality or trust; hence, emerging provenance-based approaches to compute trust and quality of data are reviewed.
## Contents

1 Introduction ........................................... 3
  1.1 Drivers for Provenance ............................... 3
  1.2 Provenance for Web Science .......................... 5
  1.3 A Web Science View of Provenance .................... 6

2 Analysis of the Provenance Literature .................. 7
  2.1 The Provenance Bibliography .......................... 7
  2.2 New Research Fronts ................................. 9
  2.3 Analysis of Research Trends ........................... 10
  2.4 Summary ............................................. 16

3 Definition of Provenance ................................ 18
  3.1 Dictionary Definition ................................. 18
  3.2 Definition of Provenance in Computer Systems .......... 19
  3.3 Mashup Exemplar Application .......................... 20
  3.4 Alternative Definitions of Provenance ................ 22
  3.5 Assumptions ........................................... 24
  3.6 Provenance: a Query over Process Assertions ........ 27
  3.7 Summary ............................................. 28

4 Provenance in Workflows and Databases ................ 29
  4.1 Views and Abstraction ................................ 31
  4.2 Data Collections and Streams ........................ 33
  4.3 Efficient Storage of Provenance ..................... 35
  4.4 Querying Provenance .................................. 36
  4.5 Workflow Evolution ................................. 38
  4.6 Provenance Semantics ................................ 39
  4.7 Human-Driven Workflows .............................. 41
  4.8 Summary ............................................. 41

5 The Open Provenance Vision ............................. 42
  5.1 Broadening the Scope of Provenance ................. 43
  5.2 Provenance-Aware Monolithic Application .......... 44
Chapter 1

Introduction

Provenance, i.e., the origin or source of something, is becoming an important concern for several research communities in computer science, since it offers the means to verify data products, to infer their quality, to analyse the processes that led to them, and to decide whether they can be trusted. In fact, provenance is an intrinsic property of data, which gives data value, when accurately captured. To motivate the need for provenance, its potential benefits are reviewed in several contexts: e-science, curated databases and Semantic Web. Furthermore, the provenance philosophy is showed not to be restricted to data in computer systems, but also to apply to real-life artifacts, such as ingredients in the food industry, parts in manufacturing and works of art (Section 1.1). Building on theoretical and practical results related to provenance, a new, multi-disciplinary perspective of provenance is proposed, so that it can be developed on the Web (Section 1.2). A bibliography-based methodology, capable of identifying trends in the provenance research community, is outlined; the results of this analysis are used to structure a vision of provenance on the Web (Section 1.3).

1.1 Drivers for Provenance

As the e-science vision becomes reality [445, 446], researchers in the scientific community are increasingly perceived as providers of online data, which take the form of raw data sets from sensors and instruments, data products produced by workflow-based intensive computations [175], or databases resulting from sophisticated curation [53]. While science is becoming computation and data intensive, the fundamental tenet of the scientific method remains unchanged: experimental results need to be reproducible. In contrast to a workflow, which can be viewed as a recipe that can be applied in the future, provenance is regarded as the equivalent of a logbook, capturing all the steps that were involved in the actual derivation of a result, and which could be used to replay the execution that led to that result so as to validate it.
Curated databases typically represent the efforts of a dedicated group of people to produce a definitive description of some subject area [53]. They cover a vast range of application domains from Swiss-Prot\(^1\), which is a protein knowledgebase that is manually annotated and reviewed, to Wikipedia\(^2\), a crowd-sourced encyclopaedia, with increasingly sophisticated editorial processes. Such databases are generally published on the Web; they are heavily cross-referenced with, and include data from, other databases. Curated databases therefore bring out the problem of attribution (who initially created a description), and raise questions about the source, or *provenance*, of such descriptions (where were descriptions initially published).

Meanwhile, the Web has evolved into a network of blogs, information portals, and social bookmarking services which provide automated feeds between subscribers. As the Web allows information sharing, discovery, aggregation, filtering and flow in an unprecedented manner, it also becomes very difficult to identify, reliably, the original source that produced an information item on the Web. Without knowing the provenance of information, information services may not be able to undertake the necessary due diligence about their content, they may be the subject of fraud or spam, and overall they may be judged as unreliable. Provenance is in fact identified as one of the many salient factors that affect how users determine trust in content provided by Web information sources [443]. This view is echoed by Lynch [257], who argues that among the consequences of this shift to new highly distributed dissemination systems, will be a new emphasis on the provenance of data and metadata, and the need for information retrieval systems to permit users to factor in trust preferences about this information.

Where the Web originally consisted largely of documents intended to be read by humans, the Semantic Web [430] envisions a Web of information and knowledge processable by computer systems which undertake automated reasoning. Central to this effort are RDF [448] and OWL [437], the frameworks in which to express metadata, vocabularies and perform associated reasoning. This vision is being deployed by means of Linked Data [431, 458], an information space in which data is being enriched by typed links expressed in RDF [448], cross-referencing data sets, in a machine-processable fashion. Given the possibility for anybody (or system) to publish sets of Linked Data that refer to others, reasoners will need explicit representations of provenance information in order to make trust judgements about the information they use [426].

The issue of provenance is in no way limited to data, information or knowledge. It also applies to physical artifacts, for example in the food industry. From wine to meat, from dairy products to whisky, from coffee to vegetables, the food industry is very keen to be able to demonstrate the origin of the ingredients consumers purchase and eat. Understanding the provenance of food, i.e. its ori-

\(^1\)www.uniprot.org  
\(^2\)www.wikipedia.org
gin, how it is produced, transported, and delivered to consumers, is turned into a competitive advantage by the food industry, since it allows it to demonstrate quality (in taste, in carbon footprint, or in ethics). Furthermore, across the world, governments and associated regulatory authorities are interested in food safety, and typically require the traceability of food. Likewise, manufacturers focus on compliance and traceability initiatives for a variety of reasons. Understanding past processes is critical to discover bottlenecks, inefficiencies, wastage, and learn how to improve them. Exact traceability is essential to manage product recalls efficiently and minimise their economic impact. Similarly to the food industry, provenance of products is used to build customer trust. And of course, in the context of art, the provenance of art objects is so important that available evidence is typically produced before auctions in order to maximize the price obtained for these objects.

1.2 Provenance for Web Science

Web science is the emerging interdisciplinary field that aims to understand the Web, engineer its future and ensure its social benefit [429]. In the context of Web science, trust is recognised as one of the important concerns associated with the Web [182]: there is a broad consensus that trust in content could be derived if the transformations and derivations that resulted in such content are known. Hence, given that the Web currently provides little support for provenance, the topic of provenance is becoming recognized as an important subject of investigation [458] in this context.

Like Web science, there is a multi-disciplinary facet to provenance. First, within computer science, multiple sub-disciplines are involved including database, systems, escience, grid, Semantic Web, and security. Second, provenance can be exploited to provide new services to the scientific community, businesses, and all Web users. It has the potential to make systems more transparent, and therefore auditable. As a result, it is a strong contender technology to underpin information accountability [402]. While it can be used to perform compliance checks (such as conformance to process or checking that terms of data licensing are met), it also raises issues related to privacy. Thus, societal, legal, and business perspectives on provenance could potentially have a wide impact on its use on the Web.

The aim of this document is to survey the technical aspects of provenance that are relevant to Web science, but also to draw attention to the potential multi-disciplinary opportunities that they bring. Provenance, as a technical subject of study, is by no means a green field. The oldest publications discussed in this survey dates back to the late eighties. Importantly, the interest of provenance has been growing dramatically, as illustrated by the number of publications on the topic (see Figure 2.1, page 8, to be discussed in the next chapter). Over 400 publications on provenance have been identified, 200 of which have been
published over the last two years.

Several surveys already exist, but, to some extent, work has so far been broadly surveyed on a per discipline basis, as illustrated by Simmhan et al.’s review of provenance in e-Science [360], Bose and Frew’s survey of provenance for scientific processing [36], Cheney et al.’s survey of database provenance [91], and Glavic and Dittrich’s classification of approaches [179]. This article aims to break such silos, and tries to investigate cross-cutting concerns that are relevant to providing provenance of information on the Web.

In fact, society is now at a turning point since it is presented with a unique opportunity, which will require social and technical changes: it is the author’s belief that society can and should reliably track and exploit the provenance of information on the Web. To achieve this vision, the research output from all disciplines investigating provenance should be integrated into a coherent approach, for which a foundational framework is proposed here. For instance, the work undertaken by the workflow community on provenance is very relevant to the flow of information that is becoming common place on the Web, since it can help track provenance as information flows through distributed services. Given that much of the data available on the Web is actually stored in databases, provenance research in the context of databases is essential, since it tracks provenance as data changes within databases. The work focusing on making provenance secure and non forgeable is also relevant to the goal of reliably tracking information on the Web.

1.3 A Web Science View of Provenance

Having compiled the most extensive bibliography on provenance so far, developments in Web science provide tools and techniques to analyse this research topic. Simple metrics such as citation count can help identify the most popular papers. However, citation analysis can help gain a deeper insight in the different subfields of this subject of study. Using clustering techniques, emerging research fronts dealing with different concerns can be identified; using tag clouds [427], these concerns can be summarized in a visual manner.

This survey is structured as follows. Chapter 2 undertakes an analysis of the provenance literature, discusses key topics of interest and identifies landmark papers. Chapter 3 discusses a broad definition of provenance that would apply to the Web and compares it with various alternative definitions that are recast in a Web context. Chapter 4 contrasts the work undertaken by the workflows and databases, which have traditionally studied provenance. By some measures, these approaches can be regarded as closed; hence, Chapter 5 introduces a vision of provenance for open systems such as the Web. Chapter 6 then discusses issues of provenance related to Web technologies. Finally, the problem of accountability is tackled in Chapter 7, before some concluding remarks in Chapter 8.
Chapter 2

Analysis of the Provenance Literature

Multiple data sources have been used to compile the largest bibliographical database on provenance (in a technical sense) so far. The database\(^1\) is made available online by the journal publisher\(^2\). This reasonably large corpus is useful to identify landmark papers and observe an acceleration of activities (Section 2.1), but more importantly, to analyse emerging trends in the research community. Specifically, CiteSpace [434, 435] allowed the discovery of clusters of papers that constitute research fronts (Section 2.2). Six clusters have been identified and positioned in time, covering topics as varied as database, workflows, eScience, “Provenance Challenge”, Open Provenance Model, Semantic Web and electronic notebooks. The key characteristics of these research fronts are extracted, and used to structure a foundational framework for provenance on the Web, as well as the rest of the paper (Section 2.3).

2.1 The Provenance Bibliography

The provenance bibliographical database was compiled using multiple sources: the author’s own original database, the ACM, IEEE, and Springer digital libraries, the DBLP computer science bibliography\(^3\), and some programmes of provenance-specific events such as the International Provenance and Annotations workshops (IPAW’06, IPAW’08), and the Workshop on Theory and Practice of Provenance (TAPP’09). Each publication is maintained with the explicit list of publications it cites and its abstract. A transitive closure of citations is applied

---

\(^1\)Note that this article’s bibliography is divided in two sections: the first part consists of the provenance bibliography, whereas the second part starting page 130 refers to papers that do not have provenance as a specific focus.

\(^2\)http://www.nowpublishers.com/web/tocompletebypublisher

\(^3\)http://dblp.uni-trier.de/
so as to ensure that each cited paper that contains the words ‘provenance’ or ‘lineage’ in its title is included in the database (provided this is a Computer Science paper).

Since the analysis of publications relies on temporal information, citation counts are kept separate for conference and journal versions. Likewise, technical reports can represent significant contributions by a community; when a technical report is superseded by a published paper, the latter is preferred (and the published paper was assigned all its citations). The bibliography contains papers that were known to the author up to summer 2009. More recent publications can be found in provenance specific workshops such as TAPP, IPAW, and at specialist conferences such as ISWC, eScience, WWW, SIGMOD, ICDE, and VLDB.

Figure 2.1 contains a histogram displaying the number of publications on provenance per year. A total of 425 papers have been identified. The first publication dates back from 1986 [22] and describes an auditing technique to assist analysts in understanding and validating data results. The histogram shows a definite trend in the research activity related to provenance, with about half the papers published in the last two years.

![Number of Provenance Publications](image)

**Figure 2.1: Number of Provenance Publications**

Several publication peaks can be observed in Figure 2.1; they coincide with events organised by the “provenance community” itself: in 2002, the first provenance workshop organised by Foster and Buneman; in 2006, the International Provenance and Annotation Workshop\(^4\) (IPAW) workshop organised by Foster and Moreau; and, in 2008, the second IPAW workshop organised by Freire and

---

\(^4\)www.ipaw.info
Moreau and the first Provenance Challenge special issue edited by Moreau and Ludaescher [297].

The provenance bibliography database was condensed into the list of the most cited papers (within the database). The top of the list is occupied by Buneman et al.’s seminal paper on Where and Why provenance [59]. Two surveys [360, 36] appear in the top five. The reader will note that this analysis does not reflect the impact of a publication outside the provenance community. For instance, Buneman et al.’s paper attracts 387 citations in Google Scholar (in August 2009), but is cited only 116 times in this database.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Citations</th>
<th>Paper (first author:venue)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>116</td>
<td>Buneman:ICDT01 [59]</td>
</tr>
<tr>
<td>2</td>
<td>97</td>
<td>Simmhan:SIGMOD05 (*)</td>
</tr>
<tr>
<td>3</td>
<td>65</td>
<td>Foster:SSDBM02 [147]</td>
</tr>
<tr>
<td>4</td>
<td>62</td>
<td>Cui:TODS00 [108]</td>
</tr>
<tr>
<td>5</td>
<td>56</td>
<td>Bose:ACMCS05 (*)</td>
</tr>
<tr>
<td>6</td>
<td>49</td>
<td>Moreau:CCPE08 [298]</td>
</tr>
<tr>
<td>7</td>
<td>47</td>
<td>Woodruff:ICDE07 [406]</td>
</tr>
<tr>
<td>8</td>
<td>47</td>
<td>Miles:JOC07 [279]</td>
</tr>
<tr>
<td>9</td>
<td>44</td>
<td>Groth:DD1.1 [196]</td>
</tr>
<tr>
<td>10</td>
<td>49</td>
<td>Bhagwat:VLDB04 [28]</td>
</tr>
<tr>
<td>11</td>
<td>39</td>
<td>Muniswamy-Reddy:USENIX06 [301]</td>
</tr>
<tr>
<td>12</td>
<td>38</td>
<td>Widom:CIDR05 [403]</td>
</tr>
<tr>
<td>13</td>
<td>38</td>
<td>Buneman:SIGMOD06 [51]</td>
</tr>
<tr>
<td>14</td>
<td>37</td>
<td>Freire:IPAW06 [157]</td>
</tr>
<tr>
<td>15</td>
<td>36</td>
<td>Groth:OPDIS04 [198]</td>
</tr>
<tr>
<td>16</td>
<td>36</td>
<td>Cui:VLDB03 [107]</td>
</tr>
<tr>
<td>17</td>
<td>33</td>
<td>Fosterb:PROV02 [146]</td>
</tr>
<tr>
<td>18</td>
<td>31</td>
<td>Zhao:CCPE08 [419]</td>
</tr>
<tr>
<td>19</td>
<td>31</td>
<td>Groth:HPDC05 [199]</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>Szomszor:ODBASE03 [375]</td>
</tr>
<tr>
<td>21</td>
<td>30</td>
<td>Cui:ICDE00 [105]</td>
</tr>
<tr>
<td>22</td>
<td>29</td>
<td>Zhao:ISWC04 [421]</td>
</tr>
<tr>
<td>23</td>
<td>29</td>
<td>Altintas:IPAW06 [5]</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>Moreau:IPAW06 (*)</td>
</tr>
<tr>
<td>25</td>
<td>27</td>
<td>Green:PODS07 [189]</td>
</tr>
</tbody>
</table>

Figure 2.2: Fifty most-cited publications, where starred items denote surveys or proceedings

2.2 New Research Fronts

Emergent trends and abrupt changes in the scientific literature can be associated with internal as well as external causes [434]: typical internal causes include new discoveries and scientific breakthroughs; external ones may provoke scientists to study a matter from new perspectives. In the case of provenance, it could be conjectured that the development of the Grid as a technology for running scientific applications [440] and the UK e-science programme [446] have been two significant external triggering factors that have caused increasing number of
researchers to focus on the provenance problem. This conjecture is based on an analysis of the participants to the first provenance workshop and IPAW’06, who were predominantly from a grid computing and e-science background.

Chen [434] defines notions of research front and intellectual base.

The concept of a research front was originally introduced by Price to characterize the transient nature of a research field. Price observed what he called the immediacy factor: There seems to be a tendency for scientists to cite the most recently published articles. In a given field, a research front refers to the body of articles that scientists actively cite.

The concept of an intellectual base is useful to further clarify the nature of a research front. Whereas a research front defines the state of the art of a specialty (i.e., a line of research), what is cited by the research front forms its intellectual base.

To investigate provenance research fronts, CiteSpace⁵, Chen’s citation analysis tool, and associated definitions are used:

- Research front: Emerging thematic trends and surges of new topics;
- Intellectual base: Co-citation network;
- Cluster: Hybrid networks of co-cited articles and terms citing these articles;
- Labeling: Terms from titles, abstract and descriptions of abrupt frequency increase.

2.3 Analysis of Research Trends

As per CiteSpace’s default configuration, the bibliography’s 1995 to 2009 time period is sliced in 1-yearly slices. For each slice, a maximum of the 30 most cited publications are selected. The two most cited publications [59, 360] are also excluded from the analysis, since their frequent citations resulted in a smaller, and not discriminating set of clusters. This section analyses and provides an interpretation of the clustering produced by CiteSpace.

Six clusters have been identified by CiteSpace. They are graphically displayed in Figure 2.3 and named as follows:

Cluster 0: Security

Cluster 1: Database 05–08, RDF and Open Provenance Model

Cluster 2: Workflow 05–08 and Database 08–09

⁵http://cluster.cis.drexel.edu/~cchen/citespace/
Cluster 3: Provenance Challenge

Cluster 4: eScience and Database 02–05

Cluster 5: Electronic notebook

Cluster name: First cluster label (See Figures 2.6 and 2.7)

<table>
<thead>
<tr>
<th>Cluster 0: Security</th>
<th>(2.2) access control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1: Database 05-08 and OPM</td>
<td>(7.17) nested data</td>
</tr>
<tr>
<td>Cluster 2: Workflow 05-08 and Database 06-08</td>
<td>(187.29) pipeline-centric provenance model</td>
</tr>
<tr>
<td>Cluster 3: Provenance Challenge</td>
<td>(6.24) multi-institutional scientific system</td>
</tr>
<tr>
<td>Cluster 4: eScience and DB 02-05</td>
<td>(12.54) data warehouse</td>
</tr>
<tr>
<td>Cluster 5: Electronic Logbook 02</td>
<td>(46.82) multi-scale science</td>
</tr>
</tbody>
</table>

Figure 2.3: Six Clusters in the Provenance Literature

Figures 2.4 and 2.5 contain tables summarising the intellectual base of these clusters, i.e., the papers cited by the research front; the tables are structured as follows. The first column identifies the cluster number. The second column is concerned with bursts, defined as follows: a burst is a set of fast-rising terms used by scientists in their latest publications. It is a measure of how often a paper is cited in the context of a burst in the citing paper; the higher the number, the
more the publication is cited in papers with these fast-rising terms. The third column contains the betweenness centrality (it quantifies the importance of the node’s position in a cluster): the higher the value, the more representative of the cluster. Sigma combines betweenness centrality and the burst rate to provide an indication of the transformative strength of a publication in a given network over a time interval $[435]$; again, the higher the number, the stronger its transformative potential. The following columns include the publication year, the author and publication venue, and the bibliographic reference.

The cluster description is complemented by Figures 2.6 and 2.7, which contain the cluster labels and the research front. The research front consists of the citers to a cluster, whereas the cluster labels are obtained from salient features selected from the titles and abstracts of the citers; labels are combined with descriptors of abrupt frequency increase.

The reader should note that this set of clusters is not a partition of the complete bibliography, since only the 30 most cited publications are selected for each year. Given the histogram of Figure 2.1, this means that a substantial number of publications from 2006–2009 do not appear in Figures 2.4 to 2.7. This is however not a concern since the aim is to identify trends, rather than exhaustively categorize papers. The figures are provided for completeness, in order to guide readers who wish to investigate these clusters. Each of them is now described.

**Cluster 0: Security**  Cluster 0 is the smallest and consists of two papers in its intellectual base $[47, 377]$, identifying key issues related to provenance and security. The research front is formed by publications dating from 2007–2009, discussing issues pertaining to provenance access control, provenance authenticity and scalable security.

The following two clusters identify research trends in the database and workflow communities.

**Cluster 1: Database 05–08, RDF and OPM**  This cluster’s intellectual base consists of a set of publications from the database community, essentially investigating theoretical aspects of provenance $[51, 189, 55, 89, 54, 25, 93, 53]$, as well as a survey $[380]$, which shows a high citation burst, and two tutorials $[62, 118]$. This cluster also includes the publications about the Open Provenance Model $[292, 299]$, a community-driven provenance model, where $[292]$ has a high burst of citation, and a frequently cited paper about provenance in RDF $[66]$. The research front is predominantly composed of papers from the database community. Key topics discussed include nested data collection, dependency analysis and database technology.
<table>
<thead>
<tr>
<th>Cluster</th>
<th>Burst</th>
<th>Centrality</th>
<th>Sigma</th>
<th>Year</th>
<th>Author:Venue</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 0: Security (2/2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0.01</td>
<td>0</td>
<td>2008</td>
<td>Braun:HOTSEC08</td>
<td>[47]</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2006</td>
<td>Tan:IPAW06</td>
<td>[377]</td>
<td></td>
</tr>
<tr>
<td>Cluster 1: Database 05-08, RDF and OPM (14/14)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6.14</td>
<td>0.01</td>
<td>0.06</td>
<td>2007</td>
<td>Moreau:OPM1.00</td>
<td>[292]</td>
</tr>
<tr>
<td>1</td>
<td>3.25</td>
<td>0</td>
<td>0.03</td>
<td>2007</td>
<td>Tan:DBBUL07</td>
<td>[380]</td>
</tr>
<tr>
<td>1</td>
<td>0.03</td>
<td>0</td>
<td>2006</td>
<td>Buneman:SIGMOD06</td>
<td>[51]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>0</td>
<td>2007</td>
<td>Green:PODS07</td>
<td>[189]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>0</td>
<td>2005</td>
<td>Carroll:WWW05</td>
<td>[66]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2008</td>
<td>Buneman:TODS08</td>
<td>[55]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2007</td>
<td>Cheney:DBPL07</td>
<td>[89]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2007</td>
<td>Buneman:ICDT07</td>
<td>[54]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2006</td>
<td>Benjelloun:VLDB06</td>
<td>[25]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2006</td>
<td>Chiticariu:VLDB06</td>
<td>[93]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2008</td>
<td>Buneman:PODS08</td>
<td>[53]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2008</td>
<td>Davidson-Ferreira:SIGMOD08</td>
<td>[118]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2007</td>
<td>Buneman:SIGMOD07</td>
<td>[62]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2008</td>
<td>Moreau:OPM1.01</td>
<td>[299]</td>
<td></td>
</tr>
<tr>
<td>Cluster 2: Workflow 05-08 and Database 06-08 (13/13)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4.99</td>
<td>0</td>
<td>0.01</td>
<td>2009</td>
<td>Levine:DFRWS09</td>
<td>[250]</td>
</tr>
<tr>
<td>2</td>
<td>4.34</td>
<td>0</td>
<td>0.01</td>
<td>2007</td>
<td>Gil-Deelman:IEEE07</td>
<td>[175]</td>
</tr>
<tr>
<td>2</td>
<td>4.34</td>
<td>0</td>
<td>0.01</td>
<td>2006</td>
<td>Agrawal:VLDB06</td>
<td>[1]</td>
</tr>
<tr>
<td>2</td>
<td>4.1</td>
<td>0</td>
<td>0.01</td>
<td>2009</td>
<td>Groth:TODT09</td>
<td>[202]</td>
</tr>
<tr>
<td>2</td>
<td>3.4</td>
<td>0.03</td>
<td>0.18</td>
<td>2008</td>
<td>Chapman:SIGMOD08</td>
<td>[75]</td>
</tr>
<tr>
<td>2</td>
<td>3.4</td>
<td>0.02</td>
<td>0.14</td>
<td>2008</td>
<td>Clifford:CCPE08</td>
<td>[98]</td>
</tr>
<tr>
<td>2</td>
<td>3.4</td>
<td>0.01</td>
<td>0.11</td>
<td>2008</td>
<td>Ludascher:CCPE08</td>
<td>[256]</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.02</td>
<td>0.17</td>
<td>2008</td>
<td>Moreau:CCM08</td>
<td>[294]</td>
</tr>
<tr>
<td>2</td>
<td>0.19</td>
<td>0</td>
<td>2005</td>
<td>Bos:ACMS05</td>
<td>[36]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.08</td>
<td>0</td>
<td>2006</td>
<td>Muniswamy-Reddy:USENIX06</td>
<td>[301]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.04</td>
<td>0</td>
<td>2008</td>
<td>Zhao:CCPE08</td>
<td>[419]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.02</td>
<td>0</td>
<td>2006</td>
<td>Bowers:IPAW06</td>
<td>[39]</td>
<td></td>
</tr>
<tr>
<td>Cluster 3: Provenance Challenge (11/11)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9.4</td>
<td>0.09</td>
<td>0.19</td>
<td>2008</td>
<td>Moreau:CCPE08</td>
<td>[298]</td>
</tr>
<tr>
<td>3</td>
<td>4.33</td>
<td>0.02</td>
<td>0.13</td>
<td>2008</td>
<td>Barga:CCPE08</td>
<td>[17]</td>
</tr>
<tr>
<td>3</td>
<td>4.31</td>
<td>0</td>
<td>0.04</td>
<td>2008</td>
<td>Kim:CCPE08</td>
<td>[232]</td>
</tr>
<tr>
<td>3</td>
<td>3.31</td>
<td>0</td>
<td>0.06</td>
<td>2008</td>
<td>Miles:CCPE08</td>
<td>[282]</td>
</tr>
<tr>
<td>3</td>
<td>3.29</td>
<td>0</td>
<td>0</td>
<td>2008</td>
<td>Holland:CCPE08</td>
<td>[219]</td>
</tr>
<tr>
<td>3</td>
<td>2.8</td>
<td>0</td>
<td>0.02</td>
<td>2008</td>
<td>Golbeck:CCPE08</td>
<td>[183]</td>
</tr>
<tr>
<td>3</td>
<td>0.03</td>
<td>0</td>
<td>2006</td>
<td>Altintas:IPAW06</td>
<td>[5]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0</td>
<td>2008</td>
<td>Cohen-Boulakia:CCPE08</td>
<td>[100]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0</td>
<td>2008</td>
<td>Frew:CCPE08</td>
<td>[160]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2008</td>
<td>Bowers:CCPE08</td>
<td>[41]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2005</td>
<td>Bavoil:VC05</td>
<td>[20]</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.4: Clusters 0, 1, 2 and 3: Intellectual Base
Figure 2.5: Clusters 4 and 5: Intellectual Base

Cluster 2: Workflow 05–08 and Database 06–08  Cluster 2 is the dual of Cluster 1. Its intellectual base consists of a vast majority of workflow papers [250, 175, 202, 98, 256, 294, 419, 39], a provenance-aware operating system [368], and a survey [36]; they are accompanied by database papers, which are also concerned with systems [1, 217, 75]. The research front also consists of papers that, by and large, are system-oriented. Semantic Web technology is also quite a common thread in this cluster.

Cluster 3: Provenance Challenge  Cluster 3’s intellectual base consists of papers published after the first provenance challenge [298], an interoperability exercise between provenance systems. Its research front broadly consists of papers concerned with practical considerations for provenance.

Cluster 4: eScience and Database 02–05  Cluster 4’s intellectual base is concerned with both eScience/grid and database research, with the following central papers: Foster et al.’s Chimera system [147], Woodruff and Stonebraker’s fine-grained lineage [406], and Cui and Widom’s lineage in data warehouses [105]. This period, which coincides with the first peak in the histogram of Figure 2.1, was very active, as illustrated by the cluster size of 59. In this cluster, several papers have significant publication burst, and have attracted substantial citations.
Cluster 0: Security (size: 2)

- (2.2) access control
- (1.79) grouping provenance information
- (1.79) fake picasso
- (1.79) preventing history forgery
- (1.79) new sea data-provenance framework
- (1.1) scalable access control
- (0.69) multi-institutional scientific system
- (0.69) secure provenance

Research front:
Chong:TAPP09 [95], daCruz:CS09 [113], Groth:thesis07 [207], Hassan:FAST09 [216], Rosenthal:TAPP09 [338], Syalim:ISA09 [374], Tsai:ISADS07 [388], Tsai:SOCA07 [387]

Cluster 1: database 05-08 and OPM (size: 14)

- (7.17) nested data
- (7.17) dependency analysis
- (7.17) sql query
- (5.38) data synchronization
- (3.58) provenance semiring
- (3.58) efficient provenance storage
- (3.58) exploring scientific workflow provenance
- (3.58) using hybrid query
- (3.58) nested data collection
- (2.43) scientific workflow
- (2.43) data provenance

Research front:
Anand:EDBT09 [7], Anand:SSDBM09 [8], Biton:VLDB07 [29], Buneman:ICDT07 [54], Chebotko:escience08 [77], Cheney:DBBUL07 [85], Cheney:DBPL07 [89], Cheney:PLANX09 [87], Cheney:TRENDDB09 [91], daCruz:CS09 [113], Davidson:DBBUL07 [117], Ding:SS05 [125], Ding:tech05 [124], Factor:TAPP09 [136], Foster:DBBUL07 [149], Gibson:TAPP09 [173], Glavic:BTW07 [179], Green:PODS07 [189], Green:VLDB07 [188], Groth:ESAW09 [200], Hartig:LDOW09 [213], Hasan:SSS07 [215], Moreau:FOPM09 [296], Mutsuzaki:CIDE07 [305], Rosenthal:TAPP09 [338], Sarma:tech07 [347], Silles:USER09 [355], Sun:SIGMOD09 [373], Tan:DBBUL07 [380], Vansummeren:DBBUL07 [389]

Cluster 2: Workflow 05-08 and Database 06-08 (size: 13)

- (187.29) pipeline-centric provenance model
- (8.32) provenance model
- (1.79) semantic annotation
- (1.62) scientific workflow
- (1.1) knowledge provenance
- (1.1) collection-oriented scientific workflow run
- (1.1) conceptual model

Research front:

Cluster 3: Provenance Challenge (size: 11)

- (6.24) multi-institutional scientific system
- (4.85) provenance framework
- (4.39) conceptual model
- (2.77) e-science provenance
- (2.43) scientific workflow
- (2.2) scientific data
- (2.2) managing data provenance
- (2.2) connecting scientific data
- (2.2) project history
- (1.79) tracking file
- (1.79) kepler provenance framework
- (1.79) provenance support
- (1.79) creating visualization
- (1.39) building practical provenance system
- (1.1) constructing scientific publication package
- (1.1) graphical interface
- (1.1) using provenance
- (1.1) provenance explorer-a

Research front:

Figure 2.6: Labels and Research Front for Clusters 0 to 3
Cluster 4: eScience and DB 02-05 (size: 59)

(12.54) data warehouse; (12.08) scientific data; (8.96) virtual data language; (8.96) provenance recording; (8.96) warehousing environment; (8.96) virtual data language; (8.96) warehousing environment; (8.96) practical lineage; (8.96) data grid environment; (8.96) scientific data processing; (8.96) conceptual framework; (8.96) using schema transformation pathway; (7.17) managing scientific data lineage; (7.17) conceptual model; (5.38) agent-oriented data curation; (5.38) semantic desktop application; (5.38) modeling provenance; (4.39) semantic web

Research front:
Balis:eScience07 [14], Balis:PPAM07 [13], Bose:ACMCS05 [36], Bose:SSDBM02 [33], Buneman:FSTTCS2000 [58], Buneman:IDM00 [50], Buneman:SIGMOD07 [62], Cavalcanti:PROV02 [67], Chapman:DBBUL07 [73], Chebotko:escience08 [84], Cui:ICDE00 [104], Cui:DMDW00 [106], Cui:ICED00 [105], Cui:thesis01 [103], Cui:TODS00 [108], Cui:VLDB03 [107], DaSilva:DEBULL03 [114], Davidson:DBBUL07 [117], Ding:SS05 [125], Ding:tech05 [124], Fan:AD02 [137], Fan:IOS03 [138], Feng:ICCS07 [140], Fosterb:PROV02 [146], Fox:IJPR05 [153], Frew:PROV02 [159], Frew:SSDBM01 [158], Glavic:BTW07 [179], Greenwood:AHM03 [190], Groth:AHM05 [201], Groth:HPDC05 [199], Groth:IPAW06 [203], Groth:thesis07 [207], Hao:DES05 [139], Hasan:SSS07 [215], Huang:DEXA05 [224], Hunter:JDL07 [225], Lawabnia:MSST05 [245], Ledlie:NETDB05 [247], Lord:SWL04 [255], Macleod:PA002 [260], Marines:SBC07 [265], Miles:AMAS07 [285], Miles:AOS07 [283], Miles:eScience07 [278], Miles:IPAW06 [276], Miles:MASBIOME2005 [275], Monroe:SEM06 [304], Ram:BO05 [331], Rios:ISVC07 [336], Ruth:ITRUST04 [340], Simmhan:SIGMOD05 [360], Simmhan:thesis07 [358], Spery:GEO01 [368], Stevens:BBIO07 [371], Tan:IPAW06 [377], Townend:AHM05 [385], Townend:ISORC05 [386], Tsai:SOCA07 [387], Vazquez:book07 [390], Wang:ICCS07 [395], Widom:CIDR05 [403], Wang:ISWC05 [404], Zhao:ICSNW04 [416], Zhao:IPAW06 [424], Zhao:ISWC04 [421], Zhao:thesis07 [417], Zhao:SWT03 [415], Zhao:thesis07 [422]

Cluster 5: Electronic Logbook 02 (size: 3)

(46.82) multi-scale science; (5.38) supporting emerging practice

Research front:
Myers:SWT03 [306]

Figure 2.7: Labels and Research Front for Clusters 4 and 5

as illustrated by Figure 2.2.

Cluster 5: Electronic Logbook The last cluster consists of papers that were early advocate of nascent semantic web technologies, electronic logbook, and multi-scale science.

2.4 Summary

The use of a co-citation analysis tool has provided us with a new insight on research fronts related to provenance. Research fronts are not structured according to research communities, but take into account the ephemeral and evolving nature of citations, and cross-community citations. While Clusters 4 and 5 consist of early work in the database and workflow communities, Clusters 0 to 3 represent more recent trends. A strong theoretical interest underpins Cluster 1, whereas systems-related issues are the focus of Cluster 2. The explicit presence of the Provenance Challenge inter-operability exercise (in Cluster 3) and its subsequent Open Provenance Model (in Cluster 1) show a growing interest in tracking
provenance beyond a single system. The pervasive reference to Semantic Web technologies is also indicative of a growing interest for this type of technology in the community. Finally, while still very small, concerns for security show a growing interest in designing provenance technology that cannot be forged, and hence can be trusted.

The research fronts identified with the CiteSpace tool have inspired the structure of this survey. To some extent, both the database and workflow communities have similar concerns (though different approaches); thus, the discussion will not be structured on a community-basis. Both approaches have mostly looked at closed systems, but there is growing evidence that they are broadening their investigations to more open environments. Hence, the possibility of tracking and reasoning over provenance in open environments will be looked at. In particular, opportunities and challenges of deploying such a vision over the Web and the Semantic Web will be debated. Finally, building on the security approaches, ways of making systems accountable will be studied.

Interestingly, the analysis undertaken in this chapter is a typical example application that would benefit from provenance. Other scientists would like to be able to reproduce the plots and tables presented in this chapter, they would like to configure the tool differently, or they may even want to apply the same methodology to a totally different set of data. The variety and location of information sources, the number of technologies involved, the kind of processing required (from interactive Java based program, to shell and perl scripts, through manual curation of the database) are a typical driver for the Open Provenance Vision this survey presents.
Chapter 3

Definition of Provenance

Since the vision is to track and exploit the provenance of information on the Web, it is necessary to define what provenance means in this context. Luckily, as provenance is being studied by multiple sub-disciplines of computer science, various definitions of provenance have been proposed, but unfortunately, several of them are expressed in the context of specific technologies, or under specific assumptions, which do not directly apply to the Web. Instead, with a progressive approach, from dictionary definition to conceptual definition, and by incorporating realistic distributed systems assumptions, a definitional framework is being proposed which can help develop a vision of provenance on the Web.

This chapter is structured as follows. Starting with the dictionary definition (Section 3.1), a general and conceptual definition of provenance is expressed as a process (Section 3.2). To be more concrete, a typical Web Mashup application is adopted (Section 3.3) to illustrate the conceptual definition, and to review alternative definitions, which are cast in the context of the Web (Section 3.4). Some of the assumptions and considerations typically taken into account in the context of provenance are discussed (Section 3.5). Finally, by adopting a distributed systems perspective, a distinction is made between the assertion of information related to a process, and the extraction of provenance by means of queries (Section 3.6).

3.1 Dictionary Definition

This section introduces and discusses the dictionary definition of the word ‘provenance’. Its etymology is the French verb ‘provenir’, which means to come forth, originate. According to the Oxford English Dictionary\(^1\), provenance is defined as follows.

Definition 3.1 (OED Provenance Definition) (i) the fact of coming from some particular source or quarter; origin, derivation. (ii) the history or pedigree

\(^1\)www.oed.com
Likewise, the Merriam-Webster Online Dictionary\(^2\) defines provenance as follows.

**Definition 3.2 (MWO Provenance Definition)**  
(i) the origin, source;  
(ii) the history of ownership of a valued object or work of art or literature.  

Both definitions are compatible since they regard provenance as the derivation from a particular source to a specific state of an item. The nature of the derivation, or history, may take different forms, or may emphasise different properties according to interest. For instance, for a piece of art, provenance usually identifies its chain of ownership. Alternatively, the actual state of a painting may be understood better by studying the different restorations it underwent.

From Definitions 3.1 and 3.2, two different understandings of provenance can be identified: first, as a concept, it denotes the source or derivation of an object; second, more concretely, it is used to refer to a record of such a derivation. Such a distinction is revisited when the notion of provenance is defined in the next section.

### 3.2 Definition of Provenance in Computer Systems

This section focuses on data produced by computer systems, published and discovered on the Web, and seeks to define the provenance of a piece of data (also referred to as data item or data product). The two dictionary definitions consider provenance to be the derivation from a particular source to a specific state of an item. In computer systems, activities are carried out by executing programs that take input data, input state, input configuration, and produce output data and output state. Such programs are compositional by nature and can be the result of sophisticated compositions (sequential, parallel, conditional, etc) of simpler programs. In this context, a process is regarded as an instance of an execution. In computational terms, it is an instance of a running program, i.e. a computation.

If there is a description of the past process that resulted in a data item, then it can be explained how such a data item was derived. Hence, similarly to previous work [207, 196], this survey adopts the following definition of provenance, which makes the notion of process explicit.

**Definition 3.3 (Provenance as Process)** *The provenance of a piece of data is the process that led to that piece of data.*  

\(^2\)http://www.merriam-webster.com/dictionary/provenance
Definition 3.3 is concerned with provenance as a concept since potentially many things pertaining to execution may be captured under “process”, including the executed program, input data, configuration, computer, electricity powering it, users, etc. From a Computer Science perspective, the goal is to conceive a computer-based representation of provenance that permits useful analysis and reasoning to support the drivers for provenance identified in Section 1.1. This representation has to capture details about the process; Chapter 5 introduces a concrete example of such representation.

As an illustration, the provenance of the plot in Figure 2.1 (page 8) may include the steps involved in producing the plot (the filtering, the data transformation, the plotting program), its parameters, the data involved, the user who initiated the computation, the repositories from which some of the data were collected, their hosting institution, etc.

Such a definition of provenance is broad, since it allows a variety of activities and data that may have influenced the data item in question to be captured. Furthermore, this definition is technology-agnostic, and could apply to a plot that may be found in a file system or on the Web, that may have been produced by a workflow or a Java program, that may rely on data found in a public database or in a set of files, or that may have been derived by a single machine or multiple computers distributed across the Internet.

To deal with the complexity of today’s applications, which tend to be composed by assembling services and components together, possibly dynamically, systems are studied under specific assumptions. Given these assumptions, it is also not surprising that various definitions of provenance have been proposed in different contexts. To illustrate this and other definitions of provenance, an exemplar application is first introduced.

3.3 Mashup Exemplar Application

This section considers a Web 2.0 mashup application that includes a map displaying information extracted from feeds from multiple sources (e.g. RSS feeds, Twitter, Facebook). The user interface allows for information to be filtered according to multiple criteria: user trails, time intervals and geographical regions. The application caches (some) data locally. As the user makes a selection, information is obtained from the different caches according to the selection and mashed up on the map.

A variety of technologies are exploited in this application, whose architecture is illustrated in Figure 3.1. Indeed, the mashup service renders Web pages, whose contents are computed by a query engine (Q), operating over local caches, populated by RSS feeds from several information providers (Trails provider, Photos provider, and Blogs provider). Each provider may use their own technology: for instance, the blog contents may consist of XML documents, whereas the photo
repository consists of files, stored in a directory and exposed to the Web.

Figure 3.1: Provenance in a Mashup

After the user selects Alice (one of the three individuals tracked by the Trails and Blogs providers), the mashup displays a Southampton map with a picture of the cruise liner Queen Mary II in Southampton harbour, with the blog comment “in hythe ferry, next to the Queen Mary II”. Initially, the various information items displayed in the mashup are obtained by running the following query over the local cache:

```sql
select photos.url, blogs.comment
where trails.user='alice'
    and trails.location=photos.location
    and blogs.user=trails.user
```

The query outlined above is to be seen as a “continuous” query running over the local cache, so that whenever more data is pushed by information providers, the mashup is automatically refreshed.
3.4 Alternative Definitions of Provenance

Various definitions of provenance are now illustrated using this example application. It should be noted that the example application is simple, and not necessarily conducive to illustrate all nuances of provenance encountered in the literature. Furthermore, the presentation style adopted in this section is informal, so as to give the reader an *intuition* of the various definitions.

**Provenance as Process** Following Definition 3.3, the provenance of the mashup (whose content is depicted in Figure 3.1) is the computation that resulted in this content being displayed. The final mashup was produced by combining a map with the result produced by the query engine, itself running a query based on a user selection and extracting data from the Photos and Blogs caches. According to this definition, any data, event or user action that can be connected to the mashup through a computational process potentially belongs to the provenance of the mashup.

**Provenance as a Directed Acyclic Graph** An approach to provenance is that it can be expressed by a directed acyclic graph (DAG), explaining how a data product or event came to be produced in an execution [298]. In a first approximation, it can be assumed that in such a DAG, nodes represent data items and edges data derivations. An illustration of provenance as a directed acyclic graph is displayed in Figure 3.2 for the mashup application. The mashup instance appears at the bottom. Each node represents a data item that existed at some point in the computation. For instance, $b_2$ denotes the eponymous tuple in the Blogs provider, while $ba_2$ represents its copy in the local cache, and $bb_2$ represents the copy extracted by the query engine; the query itself is represented by node $q$, which causes copies of $p_3$ and $b_2$ to be incorporated in the mashup. Figure 3.2 exposes some modeling choices. To keep the size of the graph small enough, the granularity, i.e. smallest information for which provenance is tracked, was chosen to be the tuple; alternatively, it could have been decided to adopt tables or cells as the granularity. Likewise, details about processing steps, communications, and query engines could have been exposed. This highlights the need for techniques to deal with abstraction, a topic discussed in Section 4.1.

**Why-Provenance** Initially defined in the context of databases, data lineage [108] and why-provenance [59] identify tuples, whose presence justifies a query result. In the exemplar application, the image and the blog comment displayed in the

---

3http://twiki.ipaw.info/bin/view/Challenge/SecondWorkshopMinutes
4The reader is being referred to [91, Section 1.1.1] for a detailed example exposing the differences between [108] and [59], which have different interpretations of what a justification of query result consists of.
mashup are justified by the information records \( t2 \), \( p3 \) and \( b2 \), from the Trails, Photos and Blogs providers respectively. In Figure 3.2, why-provenance is illustrated by the information records obtained by computing the transitive closure of plain edges, representing data derivations.

**Where-Provenance** The user may realise that a spelling mistake occurs in the bubble displayed by the application (the word ‘hythe’ is not capitalised). To correct it, the user would like to know the field, record, and database in which this string originally occurred, so that it can be updated accordingly. In the application, this is in information record \( b2 \), in the Blogs system. Such a notion of provenance, referred to as where-provenance \([59, 55]\), was initially defined in the context of databases; it helps identify where information was copied from. In Figure 3.2, where-provenance is illustrated by the information records obtained by following \texttt{sameAs} edges, which represent data being copied or shared.
**How-Provenance** Why-provenance states which source tuples witness the existence of a result, but it does not explain how they are involved in the creation of this result, i.e. how their involvement proves the result. How-provenance consists of a polynomial representation that hints at the structure of the proof explaining how an output tuple is derived. How-provenance was defined in the context of relational algebra and recursive datalog. A very approximate representation of how provenance for the exemplar application is the polynomial \(p_3 \times b_2 \times t_2^2\), which explains that \(p_3\), \(b_2\) and \(t_2\) are all required to justify the presence of \(pb_4\) and \(bc_2\) in the mashup, but in addition \(t_2\) was used twice, to select both \(p_3\) and \(b_2\). How-provenance can also be derived from the process representation of Figure 3.2, since such a kind of polynomial can be composed from the graph representation by considering tuples as variables. Given that, in the most general case, the representation of Figure 3.2 is a directed acyclic graph, which would result in exponents (greater than one) in the polynomial whenever several distinct paths lead to a same tuple.

Glavic [177] introduces an alternative terminology for these notions: **copy-contribution** for where provenance, **input-contribution** for why-provenance, and **influence-contribution** for Cui and Widom’s lineage. Wang and Madnick [398] use the term **data source tagging** for a mechanism to refer to original sources, similar to where-provenance, and **data intermediate tagging** for a mechanism to refer to intermediate sources that helped in composing information.

**Provenance as Annotations** There exist ontologies, such as Dublin Core [455], to provide structure and semantics to metadata of resources. Aspects of these ontologies are provenance related, such as author, creation date, and version. Such information can also be seen as a specialisation of Definition 3.3, as it is concerned with specific properties of past processes. Miles is proposing a mapping\(^5\) of the Dublin Core to the Open Provenance Model.

**Other definitions** Hasan et al. [216] adopt an event-oriented view, according to which a provenance chain for document \(D\) is defined as a non-empty time-ordered sequence of provenance records, where a provenance record captures an access to a document \(D\). With such a definition, they do not explain how a given data product has been derived.

### 3.5 Assumptions

Given the broadness of Definition 3.3 and the universal appeal of provenance, work has independently been undertaken in multiple communities, using different assumptions. These assumptions are now discussed.

\(^5\)http://twiki.ipaw.info/bin/view/OPM/ChangeProposalDublinCoreMapping
**System Scope** Some approaches have a working hypothesis that a given system entirely manages the flow of information, and that provenance has to be tracked within the scope of that system. Examples of such systems include operating system [219][160], desktop [265], statistical packages [144, 21], databases [59, 406, 177], data warehouses [105], and workflow systems (Kepler [39, 41], Taverna [419], VisTrails [157], VDS [424], Pegasus [278]). Other approaches allow for varying degrees of open-ness: curated databases allow for user-edits [52], PASOA allows for service-oriented architectures [196] where the number of components can dynamically change and is not known at design-time, or some allow for data to be published on the Web, with data and provenance both accessible by simple browser navigation [452]. Others even aim at tracking the provenance of objects in the physical world [231].

**Program** Some approaches assume that both the programming language and the program that executed the process are known, and therefore can be used to identify provenance [59], to derive a reverse function that computes provenance [406], or to encode provenance efficiently [424][419]. In general, the benefits of such approaches is that provenance can be precisely defined in terms of the execution semantics, and that it can be efficiently monitored, inferred or represented. The downside is that it may require the reasoner that wishes to process this kind of provenance to have an understanding of the language used for this execution. As a result, this type of reasoning may not be portable across environments (for instance, a reasoner operating over VDS provenance will not be able to reason over Taverna’s). Other approaches do not make such an assumption, but instead rely on ontological descriptions of what happened, e.g., Provenir [342], Web Provenance [213], OPM [299], PASOA [202]. In that case, the reasoner needs to access the ontology definition to reason over provenance. The downside of such ontological descriptions is that it may not be sure that the system, seen as a black-box, has actually performed an execution compatible with such descriptions.

**Trusted base** Associated with the “program assumption” is the trusted base that underpins the execution environment that is assumed to be faithful and capable of reporting accurate provenance. For language-aware approaches, the trusted base is the compiler and runtime system compliant with a language definition (e.g. SQL or Java); for ontological-based approaches, the trusted base is the certified library, service or workflow, whose operations are described by ontologies.

**Granularity** In both business and science, applications have to manipulate collections of data, e.g., structured data, sets, hierarchies, tables, rows, nested collections, files, or directories. Approaches to tracking the provenance vary accord-
ing to their ability to track the provenance of collections and their members. Relational databases can track provenance to the level of tables [398], rows (Trio [403], Perm [177]) and cells [55]. In some operating systems and workflow systems, the trend is to deal with files (e.g., PASS [219], VDS [424], ES3 [162, 160]), but recent approaches are dealing with collections and their members explicitly (e.g., Kepler [39, 41], Taverna [419]).

**What is in the provenance?** The broad definition of provenance (Definition 3.3) allows for a vast range of information to be captured, including data derivation, libraries, hardware where the computation was run, and runtime information. Others are most prescriptive, and identify data items that are the original raw value as it entered a system [55], data items that were the cause of a given data element [108, 177, 59], and variants where a summary of how the data were used is incorporated [189].

**The provenance of what?** All the definitions above are concerned with the provenance of data products. Some authors generalize this notion to computational processes. For instance, Michlmayr et al. [274] propose the concept of service provenance. By this, they mean information such as past quality of service and past invocations as provided by a service monitor. Such service provenance can be framed in terms of Definition 3.3, by considering the provenance of the service state, given by all its previous interactions and initial configuration. Alternatively, Freire et al. [157] introduce the concept of workflow provenance (discussed in Section 4.5); workflows are a specific kind of data for which the process of derivation also needs to be tracked. Hence, for the rest of this survey, the provenance of data only considered.

**Time** It is generally recognized that a provenance model does not have to include time [292]. Indeed, considering the graph of Figure 3.2, data derivations are represented by edges explicitly. If creation time of data items was known, there would be an expectation that if a data item is derived from another, the former would be created after the latter. However, the converse does not hold: time precedence does not imply data derivation. While time is not mandatory, it is perceived that it is practical for users to be able to refer to time [298]. Therefore, many provenance approaches support a notion of time, so that users can refer to executions or data products according to the time at which they took place or were produced.
3.6 Provenance: a Query over Process Assertions

Let us again consider the mashup application of Figure 3.1. An observer could observe all activities in detail, and produce the mashup’s provenance as displayed in Figure 3.2. This observer would have to be distributed (since it has to inspect all activities taking places at the different sites across the Web involved in this computation) and it would have to be trusted to be able to inspect all systems. In fact, it would have to be omniscient to be able to observe every detail of the computation. While such an omniscient and ubiquitous observer could be conceived in specific cases, such as in a monolithic application, it is simply not possible for such an observer to exist across the Web.

For both social and technical reasons, it is impossible to observe the entire behaviour of arbitrary systems, without their cooperation. Instead, one would have to rely on assertions made by the systems’ distributed components, about their local actions and involvement in a computation. The description of a local computation by one component would have to be connected with other descriptions of local computations by other components according to the distributed flow of information, in order to formulate an overall meaningful description of a distributed computation. Such assertions will have to be trusted to reflect an accurate description of the computations the components perform.

Therefore, the graph displayed in Figure 3.2 can be regarded as the result of a query over a set of assertions made by the different applications about their involvement in the computation; the query reconciles and composes assertions according to the flow of information. The PASOA approach [207, 196] makes the distinction between assertions about a process (referred to as p-assertions) and provenance obtained by a query over process assertions.

This conception of provenance as the result of a query is further justified by the following. The mashup’s provenance consists of a graph that traces back to the tuples t2, p3, and b2. These information items also have a provenance: the trail information could be derived from a GPS log, the photo provider could be a curated database. There is no natural reason to terminate the provenance graph at t2, p3, and b2. If the cache contents are the focus of study, it would be perfectly valid to terminate the graph with tuples stored in the cache. If the concern is about raw data capture, one would have to trace back to the various sensors involved in producing the data (GPS and camera). In fact, provenance needs to be scoped according to the user’s interest; otherwise, by default, the provenance of any item would conceptually trace back to the Big Bang, marking the origin of the Universe. The scope can identify the systems in which the tracing back should terminate, or the type of source data the user is interested in identifying. Miles [276] proposes such a scoping mechanism to allow queriers to delineate the provenance of data items.
Interestingly, with this novel understanding of provenance, the alternate definitions of provenance of Section 3.4 can be revised. These definitions can be seen as predefined queries over process assertions; for instance, where-provenance as a query that always follows the “sameAs” edges in a provenance representation.

3.7 Summary

By adopting a general definition of provenance as a process, a conceptual definitional framework for provenance has been proposed and was shown to apply to information flows over the Web. Concretely, provenance needs to be represented explicitly in a computer-processable format to allow for reasoning, so as to support the drivers discussed in Section 1.1. The proposed definition can be seen as a generalisation of alternative definitions encountered in the literature.

Assertions about processes have been distinguished from provenance resulting from a user-specific scoped query over such assertions, with respect to a specific data product. This general view of provenance is instantiated and optimised in different systems. Assertions can be optimised for specific execution environments, and queries over assertions can be pre-defined in systems; furthermore, queries can be eagerly or lazily computed, and query results can even be published as “provenance-metadata”. The distinction between process assertions and queries entails a lifecycle associated with provenance: process assertions need to be collected and accumulated as computations proceed, possibly without knowing which data product is ultimately to be derived. Once accumulated, these assertions can be queried to provide novel functionality to users, regarding the provenance of data.
Chapter 4

Provenance in Workflows and Databases

The majority of work on provenance has been undertaken by the database, workflow and e-science communities. Since several good surveys already exist on provenance in databases [91, 380, 379, 179] and in workflows and e-science [360, 297, 36, 117, 113], it is not intended to repeat these here. Instead, the purpose of this section is to identify cross-cutting concerns tackled by both communities and their respective assumptions. These concerns are research issues that need to be tackled in order to realise a vision of provenance on the Web.

To understand the distinct approaches by these communities, it is crucial to appreciate how differently workflow and database technologies are exploited. The survey begins with the scientific context, and then considers business. Workflow technology is increasingly considered a rapid experiment development tool, with workflow modifications, frequent runs, and parameter tuning [175]; workflow languages are a mechanism to rapidly glue libraries and services, easily transform data, and rapidly automate computational activities. Hence, a primary driver for provenance is reproducibility of scientific analyses and processes. Furthermore, provenance is not only used for interpreting data and providing reproducible results, but also for troubleshooting and optimization [278]. For an extensive analysis of user requirements for provenance in e-Science, the reader is invited to refer to Miles et al. [279]. The use of workflows in business differs substantially: business workflows are less of an iterative development tool, but are used to implement business processes; in such a context, traceability and accountability are important concerns [109].

In the scientific context, databases have traditionally been used for archiving data [49, 57]. Some of these databases undergo frequent updates, and new versions are released regularly. In particular, curated databases are constructed by the “sweat of the brow” of scientists who manually assimilate information from several sources [51, 10]; they may be the result of a great deal of manual anno-
tation, correction and transfer of data from multiple sources. In that context, provenance information concerning the creation, attribution, or version history of such data is crucial for assessing its integrity and scientific value\(^1\). A series of technical requirements, applying to both workflow and database technologies can be found in [73].

In the most general cases, workflows are used to compose services whose detailed behaviour is not necessarily known. Given that provenance in such a context involves components regarded as black boxes [99], Tan [380] categorizes such a kind of provenance as workflow and coarse-grained; in contrast, fine-grained provenance provides a detailed explanation of how data is actually derived. However, it is not clear that a partition between coarse and fine-grained provenance is the right approach: more or less details about services can be exposed, making them grey boxes [117]. Thus, it seems to follow from the above that there are not two competing solutions to provenance, fine-grained vs coarse-grained, but there is a continuum of solutions where all parts of the systems in which information flows can be exposed in a fine or coarse manner.

Figures 4.1 and 4.2 display tag clouds for database and workflow publications from the provenance bibliography. They share a common focus on scientific activities, workflows, provenance queries, and building and evaluating systems.

---

\(^1\)This kind of requirement is evidenced by journals, such as http://www.earth-system-science-data.net/, which adopt metadata schemas to annotate data in order to ensure their quality and potential reuse.
The rest of this chapter compares and contrasts approaches, from various perspectives, which are crucial to establishing the foundations of provenance on the Web. (i) For users to deal with the amount of information contained in provenance, mechanisms are required to abstract and synthesize information in views customized to users (Section 4.1). (ii) A specific aspect of abstraction is concerned with collections of data (Section 4.2). (iii) If the provenance of everything is to be tracked, consideration should be given to storage requirements (Section 4.3). (iv) The means to actually query provenance need to be provided (Section 4.4). (v) Tracking the evolution of workflows is a special kind of provenance tracking (Section 4.5). (vi) Formal properties of provenance are now emerging (Section 4.6). (vii) Finally, many activities involve humans in the loop, who impact on decisions and processes, and therefore need to be made explicit in provenance representations (Section 4.7).

### 4.1 Views and Abstraction

The provenance of a data product may be large, in particular, when the data product is the result of a long and complex computation. This presents challenges to users since it becomes very difficult for them to understand such provenance and make sense of it. Hence, novel techniques have been devised to allow users to deal with the complexity and size of provenance information. Some of these mechanisms consist of groupings of data, which are discussed in Section 4.2. This section focuses on approaches that have been proposed to structure historical information. They consist of layering, workflow-induced views, tracers, and accounts.
Redux [17] offers a four-layer provenance model. The first level consists of an abstract description of the experiment that captures abstract activities in the workflows and relations among them. The second level represents an instance of the abstract model, which captures instances of activities and additional relationships, as classes of activities are instantiated. The third level captures information to trace the execution of the workflow, including input data, parameters supplied at runtime, branches taken, and activities inserted or skipped during execution. The final level represents runtime-specific information, such as the start and end time of workflow execution, start and end time of individual activity execution, status codes and intermediate results, information about the internal state of each activity, along with information about the machines where activities were allocated.

A similar layering is adopted by VisTrails [348]: the workflow evolution (to be further discussed in Section 4.5) layer captures the relationships among the series of workflows edited by the user; the workflow layer consists of specifications of individual workflows; and the execution layer stores run-time information about the execution of workflow modules (e.g., execution time, machine name, date, etc.). The Wings/Pegasus workflow system [232] introduces the notion of reusable workflow template that is instantiated into a workflow instance, containing execution details. Provenance is structured in a similar manner, and, in essence, follows a layering that resembles that of provenance in Redux and VisTrails.

ZOOM [100, 99, 30] builds on the concept of composite step-classes - or sub-workflows - which is present in many scientific workflow systems to develop a notion of user views. There are several reasons why composite step-classes are useful in workflows. First, they can be used to hide complexity and allow users to focus on a higher level of abstraction. Second, composite step-classes can represent authorization levels; users without the appropriate clearance level would not be allowed to see the details of a composite step-class. A partial ordering on user views can be defined using the containment of step classes. Such views can be referred to when querying provenance: a user view determines what level of sub-workflow the user can see, and thus what data and tasks are visible in provenance queries [30]. The challenge is to construct such user views dynamically: a bottom-up approach to constructing well-formed user views is described in [29, 30].

Both VDL [98] and Karma [363] rely on a notion of nested workflows, which allows provenance to be grouped, and retrieved according to its depth. Section 6.1 discusses the technique that Hunter and Cheung [92, 225] propose to create user views by relying on Semantic Web techniques.

Tracers [282, 202] are unique tokens propagated by services at execution time through interactions, very similar to transactional contexts passed in distributed transaction systems. Tracers are typed and can be given different semantics: tracers can be used for instance to delimit a workflow run or to capture the dynamic nesting of workflow execution. As tracers are communicated by the application,
they are also documented among the assertions made about execution. By this mechanism, a tracer can be used to delimit a (sub-)process or activities with some properties, and so can help bundle all provenance regarding that (sub-)process or activities. Provenance query interfaces also take tracers as input to ensure that provenance information that belongs to the scope of a tracer is returned. Tracer-based views need not be hierarchical (as opposed to the workflow-induced views); for instance, they could be location-based.

The Open Provenance Model [299] (whose details are provided in Chapter 5) offers the notion of an account. Accounts are a workflow-independent mechanism 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. Such accounts can overlap (meaning they are related to a same execution), they can be hierarchical and linked by a notion of refinement\(^2\), or non-hierarchical\(^3\). In the latter case, they may even be offering conflicting views about a same execution observed by two different observers. The interest of accounts is that they are independent of the technology used to run the application, and therefore apply to workflow and non-workflow based systems.

4.2 Data Collections and Streams

Users very frequently have to deal with collections of data, as opposed to individual data items. Such collections may be more or less structured; they can be sets (e.g., file directories), relations (e.g., SQL relational tables), hierarchies (e.g., XML documents), or arrays (e.g., two-dimensional matrices of numbers). A collection of data is a group of data items, generally of the same type, which may be ordered or non-ordered. The grouping tends to reflect some properties of its elements: all the results produced by an experiment, all the results returned by a query, all the photos taken by a given camera during a period of time, all the simulations results regarding a given project, or a set of URIs returned by a search. For end users, collections become first-class entities that can be annotated, manipulated, transformed, or archived. As far as provenance is concerned, it is therefore important to distinguish the provenance of a collection from the provenance of its individual members. The provenance of a collection constitutes a form of abstraction, similar to the ones discussed in Section 4.1, where the collection’s provenance abstracts away from the details of its members’ provenance. Representing the provenance of collections and their members is challenging, given all the potential dimensions of the problem: collection mutability, granularity and efficient representation. Several approaches tackling these problems are now presented.

\(^2\)http://twiki.ipaw.info/bin/view/OPM/ChangeProposalMultipleHierarchicalRefinement
\(^3\)http://twiki.ipaw.info/bin/view/OPM/ChangeProposalRemoveOverlaps
Many workflow systems tend to create new data, without ever updating existing ones; this also goes for collections, where such systems, process input collections, mapping operations on their individual elements, and producing new collections. Examples of such systems include VDL [98] capable of manipulating directories of files, or Taverna/myGrid offering mapping functions operating over collections. Kepler [7] also allows collections to be manipulated, but such collections are stateful since Kepler provides operations to delete and insert members of a collection.

Database technology naturally deals with collections: for instance, relational tables, rows and cells are the constituents of the relational model, whereas hierarchies are at the core of XML databases. Provenance in databases has been dealing with data collections at various levels of data granularity: the seminal paper on why- and where-provenance [59] deals with semi-structured data (XML databases), whereas [55] deals with the provenance of cell contents in the presence of updates, and [108, 177] deal with tuples in SQL databases. Hence, approaches can be found for provenance to be tracked at all levels of data granularity: for instance, in [55], annotations to tables, rows and cells are propagated.

A challenge for systems constructing an expressive representation of provenance is the size of the provenance of a collections and its members. In the context of the Open Provenance Model, the collections profile\(^1\) allows for the provenance of members to be derived from the provenance of collections, by applying inference rules that are specific to the operation performed on the collection, such as map or filter. Instead, Anand et al. [7] prefer recording changes performed to data structures; a motivation for their choice of representation is storage efficiency, which is further discussed in Section 4.3.

So far, this section has discussed the concept of collection seen as a first-class aggregation of data, which may vary over time as elements are added or removed, but is persistent: at any point in time, it is possible to retrieve a collection and obtain all its members. On the other hand, streams are collections that are typically ephemeral (i.e., not made persistent or archived) and temporal. Specific techniques for handling the provenance of streams are now surveyed.

Sensors and data streaming techniques are increasingly used in a wide range of applications, from science (weather forecast [393]) to health care (remote health monitoring [287]). Such streamed data are used by sophisticated simulation, modelling and analysis tools. Streamed data which are of interest to both workflow and database communities, carry specific problems related to provenance. Examples of provenance queries are: which sensor was a piece of data produced by? What transformations were involved in deriving a stream? Which events were ancestors of a given event in a stream?

Vijayakumar and Plale [393, 392] adopt a stream as the unit of data for which

\(^1\)http://mailman.ecs.soton.ac.uk/pipermail/provenance-challenge-ipaw-info/2009-June/000120.html
they track provenance, so as to ensure a lightweight and low overhead approach to provenance recording in distributed stream applications. The downside is that the granularity of their provenance model is so coarse that it is unable to answer queries related to individual stream elements.

The goal of the Kepler workflow system is to build streaming applications; the provenance model conceived by Anand et al. [7] consists of changes performed to data structures; it is itself a stream, which is embedded in application streams.

Misra et al.'s approach [287] is fine-grained since, given an output (e.g., a medical alert) generated by a stream processing application, their system not only recreates the processing graph that generated the output, but also provides all the elements of the intermediate data streams that generated it. To this end, they introduce the Time-Value-Centric (TVC) model [287, 396, 228] which is an algebraic approach to compute all the ancestor events an event depends on.

While this section focused on streamed applications typically produced by sensors, work has also been undertaken on provenance in the context of video streaming applications. Gehani [171] designed an algorithm for in-band encoding of lineage metadata in video streams.

4.3 Efficient Storage of Provenance

Provenance can become huge: in the public database Gene Ontology, the provenance of a single tuple has been observed to be 10Mb [332]; likewise, a 250Mb database of biological data is associated with 6Gb of provenance [73]. The size of provenance matters; because this is a multi-dimensional challenge, it has to be acknowledged that there is a trade-off between compact representation (reducing recording/upload time), compact storage (reducing storage requirements) and query time.

Barga and Digiampietri [17] observe that a layered model can present opportunities to store provenance traces efficiently in a storage manager. This point is also noted by Scheidegger et al. [348], since structuring the provenance information into layers leads to a normalized representation that avoids the storage of redundant information.

Anand et al. [7] observe that in typical workflow steps, it is the case that not all outputs directly depend on every input. Therefore, to accurately trace the provenance of data, fine-grained descriptions of data dependencies need to be asserted. In the presence of collections and nested collections, the size of such descriptions can be considerable, resulting in poor recording performance and high storage requirements. They therefore propose a compact representation of provenance, which essentially tracks the changes performed to data structures, assuming that any other element remains identical. This compact representation is associated with inference rules, allowing all dependencies to be explicitly derived. Such representation techniques are combined with a series of storage
representation optimizations. Through a range of real and synthetic benchmarks they compare the efficiency of their representation techniques, from recording, storage and querying perspectives.

Buneman et al. [49] discuss the overhead of maintaining multiple versions of a data record in scientific databases, in the presence and absence of compression. More recently, a similar overhead study is undertaken in the context of curated databases, where Buneman et al. [51] investigate the storage requirement and associated overhead for their copy-and-paste provenance model.

Following the investigation of storage overhead in the copy-and-paste provenance model, Chapman et al. [75, 72] investigate the problem of increasing storage requirements for provenance, in the context of workflows. They identify two families of techniques to decrease the storage needs for provenance: factorization processes and inheritance-based. Factorization techniques factor out common "sub-expressions" in the provenance of different items, allowing them to be stored once for each item. Alternatively, an orthogonal optimization is based on similarities in a local portion of data collections (Structural Inheritance) or between the provenance associated with data items of a particular type (Predicate Inheritance). When provenance can be inherited by an item, there is no need to record any provenance with that item; instead, the inheritance mechanism can correctly instantiate what is required. Such techniques can reduce storage requirement by a factor of 20, while provenance remains queryable.

Ré et al. [332] note that is is often unnecessary for systems to track all derivations. Indeed, sometimes, an approximation returned quickly is more valuable than the accurate complete provenance returned after a long time. Furthermore, complete provenance does not identify the most influential steps in that derivation. As a result, they propose approximate lineage as an alternative to complete provenance. It compresses the provenance by tracking only the most influential facts in the derivation. They introduce two forms of approximate lineage. In their database approach, complete lineage is represented as a boolean formula over a set of boolean variables. Sufficient lineage is a smaller formula that logically implies the original. The second approach is polynomial lineage, consisting of a real-valued polynomial over boolean variables. (Interestingly, some form of polynomials is also used in how-provenance [189].)

Groth et al. [195] take advantage of the properties of a specific class of scientific workflows to derive an optimized provenance representation. They observe gains of over 70% for pipeline-centric workflows [195].

### 4.4 Querying Provenance

Section 3.6 discussed the idea that provenance is the result of a query over a set of assertions about execution. This approach is adopted by many systems (Taverna [419], PASS [219], ES3 [162, 160], PASOA [276], VDL [98], Kepler [39,
which accumulate information about processes during their execution, and offer query interfaces to retrieve provenance. The first Provenance Challenge shows that a wide variety of standard querying technologies are used: SQL, XQuery, Xpath, SPARQL. The downside is that implementers expose their implementation schema to their users, which makes it difficult to change it in the future. As an alternative, a series of domain specific languages are being designed to retrieve provenance (and are generally implemented as a translation to standard query languages). These domain-specific languages aim to improve expressiveness by offering new constructs and new abstractions to facilitate the writing of complex queries. The rest of this section reviews some of the characteristics of these provenance-oriented query languages.

The PASOA query interface for provenance has two constituents. First, it requires queriers to identify the data item for which they want to retrieve the provenance. Second, it mandates a specification of the part of the process for which queriers are interested in obtaining a description.

As far as data products are concerned, there are generally two approaches. Several systems, typically integrating workflow execution and provenance, name all intermediary results with a unique identifier, which can then be used to obtain their provenance. For instance, Taverna/myGrid uses Life Science IDentifiers (LSIDs), Swift uses tag URIs, and VDL uses filenames. This kind of identification can be seen as extensional, since data items are explicitly enumerated and named when issuing a provenance query. Alternatively, PASOA identifies objects intensionally, with respect to workflow steps: for instance, the object contained in a collection passed as input to a workflow step, carried out by a specific service. The PASOA provenance query interface allows for such extensional descriptions to be specified as an XPath expression, over the set of process assertions.

Having identified a data item (intensionally or extensionally), queriers then need to select the part of the process they are interested in. Since provenance usually takes the shape of a directed acyclic graph, a provenance query involves some form of transitive closure. Holland et al. review the suitability of query languages to address this kind of queries. The PASOA approach offers various ways of specifying process scope: it can be delimited by location (of activities or provenance stores), by types of derivation, or by the type of intermediary data involved in the computation.

The database community has identified several forms of provenance (why, where, how, lineage). When these notions are transposed to a broader context beyond databases, as in the mashup of Chapter 3, a scope also becomes useful. For instance, in Figure 3.2, it may be useful to identify where a tuple originally appears in the cache.

Anand et al. propose a Query Language for Provenance (QLP) designed to be independent of any particular physical representation, and that includes constructs tailored specifically for querying scientific workflow provenance. QLP
consists of two components: lineage queries (aimed at traversing the transitive closure of the graph), and structural queries (aimed at traversing nested collections); both can be combined into so-called hybrid queries.

The VisTrails provenance query language \([348]\) (vtPQL) is designed to take advantage of the structure of the VisTrails layered provenance. Each level of the query is a simple SQL-like expression with some additional functions, predicates, and attributes. Basic operations that are useful for common querying tasks over workflows, and that further simplify the query syntax have been identified.

Heinis and Alonso \([217]\) show that workflows with a tree structure produce lineage dependencies that can be efficiently stored and queried using interval encoding. They define a provenance query as the transitive closure over a DAG. By a series of benchmarks, they show that recursive queries require little space but can be slow, whereas storing all paths leads to faster queries but increases storage requirement significantly. By using intervals to represent trees, provenance of a node can be determined by finding all the intervals that enclose the interval of this node. They explore how arbitrary DAGs can be transformed into equivalent DAGs that can be encoded with one-dimensional intervals.

In contrast to why-, where-, and how- provenance, Chapman \([72]\) investigates a why not query.\(^5\) This query produces a series of statements about the potential reasons the data of interest to the user is missing from a result set. Two algorithms answering this query by traversing the provenance graph in a forward and backward manner respectively are proposed and compared in evaluations. Huang et al. \([221]\) tackle a similar problem in the context of databases. The answer to the question is what modifications to an existing database would be required for a non-answer (i.e. tuple missing from the query result) to become an answer.

Bao et al. \([15]\) introduce an algorithm for differencing provenance (due to workflow execution). The difference or edit distance between a pair of valid runs of the same specification is defined as a minimum cost sequence of edit operations that transform one run to the other. While the differencing problem is NP-hard for general graphs, a polynomial solution is proposed for series-parallel graphs (with nested loops), capturing a broad class of scientific and business workflows.

4.5 Workflow Evolution

The introduction of this chapter discusses how workflow technology is increasingly used for iterative experiment design by e-Scientists. Given that frequent tweaks to workflows and modifications of parameters can have a significant impact on experimental results, they necessarily need to be included in the provenance of the results. Thus, tracing the provenance of workflows has been an increasing

\(^5\)This form of query in the context of provenance was first brought to the author’s attention by Yolanda Gil in 2006.
concern. Since the workflow and its parameters are themselves a data set, Definition 3.3 still applies: the aim of this research is to track the process by which a workflow has been derived. This is the approach taken by VisTrails [157, 252] which maintains workflow provenance, by capturing modifications made to workflows through its integrated development environment. This allows scientists to easily navigate through the space of workflows and parameter settings used in a given scientific task. In particular, this gives them the ability to return to previous versions of a workflow and compare their results.

In some cases, the workflow does not evolve because of user modifications, but because of workflow compilers. The Pegasus workflow compiler [278] takes as input an abstract workflow (specified as a DAG) that is compiled into an instantiated workflow, directly executable by a workflow engine, where computation location, data transfers and libraries to invoke have all been made explicit. In this case, provenance can be used for troubleshooting and understanding runtime behaviour of the workflow system and of the application; this is achieved by connecting runtime information to the original abstract specification, as designed by the user [278, 280].

4.6 Provenance Semantics

A large proportion of the work on provenance in the database community has been of a theoretical nature. Some of these results have applications beyond the database world and should be considered as desirable approaches to provenance semantics in general. Buneman et al.’s implicit provenance approach [55] considers the semantics of a query language (nested relational calculus) where a value has been tagged by an annotation, referred to as a colour, denoting the origin of that value. As values are propagated, the language passes along annotations. It follows that for any result produced by a program, the associated annotations indicate where the value was derived from. Buneman et al. establish that a class of well-behaved provenance-aware program transformations is equivalent to regular program transformations under implicit provenance semantics.

A similar approach is also adopted by Souilah et al. [366], who introduce a formalism for provenance in distributed systems based on the $\pi$-calculus. Its main feature is that all data products are annotated with metadata representing their provenance. Here, annotations consist of sequences of send- and receive-events, that are extended whenever values are communicated by the application. Souilah et al.’s annotations are richer than Buneman et al.’s colouring scheme, which means that more sophisticated provenance queries can be answered. Souilah et al. define a notion of correctness of a provenance annotation if what it tells about the past of a value agrees with what actually took place.

Cheney et al. [88] introduce a notion of provenance trace, and investigate some of its properties. Two of them are worth noting: a trace is consistent if
it describes what happened during execution. It has the \textit{fidelity} property if it contains enough information to describe how the program would have executed with different inputs.

Whilst the above work \cite{55, 366, 88} characterizes provenance in the context of a programming language’s execution semantics, work is also ongoing to characterize the expressiveness of provenance data models. Moreau \textit{et al.} \cite{296} formalise the Open Provenance Model \cite{299} and investigate its expressivity. They formally specify the kind of inferences the model allows. They establish that the inferences that are permitted by OPM are sound and complete with respect to a temporal semantics of OPM.

Annotations (for correction or curation purpose) are crucial in data management. The problem of annotation propagation has long been studied in databases, and has been showed to be relevant to provenance. Wang and Madnick \cite{398} were the first to propagate source information through queries. Bhagwat \textit{et al.} \cite{28} extend the approach to arbitrary annotations being propagated according to where-provenance. DBNotes is a realization of such an approach \cite{94}. In the context of data warehousing, Cui and Widom \cite{108} define provenance of a relational tuple in a view as the tuples in the base tables that witness the existence of that tuple; to produce such a kind of provenance, they generate a “reverse” query to retrieve all combinations of base tuples that satisfy this definition. This kind of provenance was later referred to as why-provenance \cite{59}. This vast corpus of literature shows that provenance is a multi-dimensional problem: annotations can be propagated differently in the presence of complex operators (e.g nested queries); provenance can take multiple forms (e.g., set of tuples \cite{108}, colour \cite{55}), it can present different properties (minimal set \cite{59, 91} or not \cite{108, 91}), it can be implemented with a lazy \cite{380} approach (generated on demand, by means of a query, only when requested \cite{108}), or eagerly \cite{380} by propagating information at runtime \cite{28, 94}.

Glavic and Alonso \cite{178} demonstrate that the widely used definition of why-provenance (as defined by Cui and Widom \cite{108}) fails in the presence of nested subqueries. They show that in the presence of a subquery, provenance includes tuples that do not actually contribute to the result (false positives), and furthermore that there is ambiguity in the presence of multiple subqueries. They propose a revised definition of provenance that takes this limitation. Their solution is implemented in the Perm system, which rewrites SQL queries to propagate provenance alongside query results (similarly to \cite{55}). For a given query \cite{177}, Perm generates a single query that produces the same result as the original query but is extended with additional attributes used to store provenance data. The benefit is that the rewritten query is expressed in SQL and can be optimised by the DBMS, and the provenance information is represented in relational tables, not requiring the data model to be extended.
4.7 Human-Driven Workflows

There are a number of domains, where the workflow is not driven by an automatic workflow enactment engine, but directly by humans. Provenance in that context is also important. Benefits of provenance are discussed in forensic analysis, knowledge discovery and visualization.

Levine and Liberatore [250] seek to improve the reproducibility and comparison of digital forensic evidence. They propose a simple canonical description of digital evidence provenance that explicitly states the set of tools and transformations that led from acquired raw data to the resulting product. This provenance representation allows for the comparison and the reproduction of results.

The knowledge discovery process is indeed a process and the steps that a user takes to discover knowledge are as important as the knowledge itself [193]. Groth et al. [193] propose to recognize the user interactions and annotations as first-class objects, which can be exploited by other users to discover resources. Information captured in the visualization system allows for visualizations to be replayed and for previous discoveries to be found again.

Silva et al. [356, 20, 65] use VisTrails' action-based provenance model to capture changes to parameters and pipeline definitions to ensure that users are able to reproduce visualizations, and to let them easily navigate through the space of pipelines created for a given exploration. Likewise, Jankun-Kelly [227] derives a visualization process graph representing visualization activities. A classification of these graphs is introduced and metrics to analyse them are defined.

Gotz and Zhou [187] introduce the notion of insight provenance to refer to a historical record of the process and rationale by which an insight is derived during a visual analytic task. Instead of relying on an action-based provenance model as in VisTrails, they aim to capture automatically a semantic record of user activity, consisting of high-level descriptions of analytic actions. Such actions are themselves inferred from low-level user interactions.

4.8 Summary

The bulk of the work on provenance has been undertaken by the database and workflow communities, specifically in the context of scientific applications. While database and workflow technologies are used significantly differently by scientists, they share numerous provenance concerns. As important progress is being made on the theoretical front, with various semantics of provenance, there are still a number of challenging issues to consider. Various issues were further reviewed: the problems of creating user-adapted views that abstract away from tedious details of execution or of manipulated data sets, expressive provenance queries, and efficient provenance storage. Furthermore, processes can be driven by humans, and provenance should reflect this in its representation.
Chapter 5

The Open Provenance Vision

In the past, applications used to be monolithic, running within a single security domain, possibly on a single machine, and without having to inter-operate with any other software system. Today’s applications are substantially different: they consist of many components, typically involving multiple technologies, deployed in separate security domains, and architected according to service-oriented principles [453], promoting loose coupling and reuse. Furthermore, with the advent of cloud-computing [433], many applications are architected around the Web, publishing and discovering information over the Web, mashing it up, and republishing it. In this context, the challenge is to be able to track the provenance of data across multiple technologies, applications, and security domains, which are involved in their derivation.

By and large, provenance approaches developed in the context of databases and workflows, which are reviewed in Chapter 4, deal with closed systems. By that, it is meant that workflow or database management systems are in full control of the data they manage, and track their provenance within their own scope, but not beyond. A few exceptions, discussed in Section 5.1, begin to emerge with techniques to track provenance beyond their scope, but these tend to be ad-hoc solutions to specific problems. A broader perspective is required by which elements of provenance information, captured by individual systems, can be brought together to describe the provenance of information flowing across systems. This is the specific purpose of the Open Provenance Vision, which is introduced in this chapter, organised as follows.

First, this chapter reviews extant approaches that broaden their provenance tracking capabilities to other systems (Section 5.1). Next, it considers the architectural principles allowing a monolithic application to be made provenance-aware (Section 5.2), and then generalises them to multi-systems applications (Section 5.3). It then provides some background about the Provenance Challenge activity, a community project aiming at inter-operability of provenance technology (Section 5.4), which resulted in the Open Provenance Model described in Section 5.5. In Section 5.6, it discusses issues pertaining to openness and prove-
5.1 Broadening the Scope of Provenance

Chapter 4 has already discussed highlights of the literature regarding provenance in workflows and databases, which are typical technologies used in implementation of individual systems. This section surveys recent undertakings to make other technologies provenance-aware.

Bowers et al. [40] note that the scope of most scientific workflow systems (Triana, Taverna, Pegasus) is limited to single workflow runs. While VisTrails tracks workflow definitions, most implementations are largely ignorant of data management tasks carried out between workflow runs. By providing a structure to record project information and name collections, provenance can be traced across workflow executions. To some extent, the proposed solution remedies some shortcomings of the Kepler approach, which embeds provenance in the application data as an output stream; however, the problem is real and of concern to all workflow systems.

In curated databases, the curators are an important “component” of the system since they make decisions to create, delete, annotate and edit database records. Buneman et al. [51, 52] assume that changes to the database can be modeled using transactions comprising simple “copy-paste” updates. They derive a model of provenance that captures such update sequence, which allows them to reconstitute changes to the database. To achieve this vision, their implementation intercepts user’s actions. Archer et al. [10] also tackle the curation problem but in the context of a “DataSpace”. They introduce a history table that captures each user action with respect to a data relation. From it, they derive a notion of “provenance graph”, which is a directed acyclic graph, where vertices correspond the current state of a data value of interest.

Given that many systems no longer have a native user interfaces, but instead offer a Web 2.0 rich interface accessible through the Web browser, Margo and Seltzer [264] investigate “browser provenance”, in the context of which, they consider browser logs as process assertions, from which provenance can be derived.

Since user interactions typically take place over their desktop (for users working at their workstation), it is also natural to consider provenance of information on the desktop. Shah [353] tackles this problem to provide a provenance-enabled search technique for files on the desktop. They use a binary rewriting technique to trace all file system and interprocess communication calls. They build a dependence tree based on the “kinship” relation, where a file is said to be an ancestor of another file, if the former may have played a role in the origin of the latter. Ultimately, to be able to capture everything that occurs on the desktop, cooperation from the operating system would be required. In PASS [219], relevant kernel calls are trapped, and recorded by means of a logging interface.
from which provenance logs are formed: at that level, the challenge is that the business logic cannot always be reconstituted; to address this problem, in addition to the kernel-level logging interface, explicit assertions can be made about application-level dependencies.

Capturing provenance on the desktop, in the browser, at the operating system level, or at the user interface inevitably brings ethics concerns: should all user interactions with a computer be captured, to the point that all information can be traceable? Privacy is an important concern, and any log must remain the property of the user. Therefore, the necessary security mechanisms need to be put in place to protect such information. Security techniques related to provenance are discussed in Chapter 7.

5.2 Provenance-Aware Monolithic Application

The research community has now gained a fairly good understanding on how to make a single monolithic application provenance-aware [284], by this it is meant an application that tracks the provenance of its data and allows for such provenance to be queried.

It is recognized by most communities (whether workflow, database, service oriented, or others) that extra information needs to be asserted and recorded as the application proceeds. In the case of databases, such information may be referred to as annotations [94] or simply provenance data [177]. In the case of workflow systems, it is also referred to as provenance information (e.g., Kepler [39, 41] and Taverna [419]); others refer to it as process documentation [202].

Without loss of generality, the extra information to be captured will be referred to as process assertions. Process assertions are envisioned to be to electronic data what a record of ownership is to a work of art. Provenance-aware applications create process assertions and store them in a provenance store, the role of which is to offer a long-term persistent, secure storage of process assertions (cf. Figure 5.1).

Once process assertions have been recorded, provenance can be retrieved and analysed by querying the provenance store. Analysis can vary from extractions identifying which source data were copied in a result (where-provenance), to sophisticated rule-based checks to decide whether a process is compliant with a set of rules. Such checks can determine, for example, that source data are appropriately licensed, that computations are undertaken with the required precision, or that the process that was executed is following established practices. An exporting capability can be used to provide feedback to applications, by means of asynchronous notifications or alarms; continuous monitoring or audit functionality can also be programmed.

Whilst Figure 5.1 depicts the provenance store as a separate entity, it may be integrated with application data, in separate tables in the same database (cf.
PERM [177]). To date, there is no universal consensus on an internal format for process assertions, since it is often optimised for the technology used in the application. For instance, Woodruff and Stonebraker [406] propose to minimize the cost of storing process assertions by computing them lazily, as queries are issued to the provenance store, making use of an inverse function (for invertible functions).

5.3 Provenance Inter-Operability across Components

When data flows across multiple components, the technique described in Section 5.2 could be adopted to make each individual component provenance-aware. However, there is a challenge to tracking provenance across multiple applications, since there is no common provenance model to describe the execution across multiple technologies, there is no agreed mechanism to connect the provenance of a received data item and the provenance of its matching sent data, and there is no query language and mechanism to operate over multiple provenance stores.

The IEEE Standard Computer Dictionary [442] defines inter-operability as the ability of two or more systems or components to exchange information and to use the information that has been exchanged. This survey builds on this definition and defines provenance inter-operability as the ability of coupled components to express and exchange provenance information pertaining the data they produce.
or exchange, and to allow such provenance to be queried.

The Open Provenance Vision is this survey’s hypothesis that there exists a set of architectural guidelines to support provenance inter-operability, by means of open models, open serialization formats and open APIs (Application Programming Interfaces). With the Open Provenance Vision, the provenance from individual systems or components can be expressed, connected in a coherent fashion, and queried seamlessly. The constituents of the Open Provenance Vision are now discussed.

Figure 5.2: Provenance Across Systems

Figure 5.2 displays a flow of information across multiple applications or systems. Each system is individually made provenance-aware, and making use of its own provenance store. To track the provenance of data produced by such an application, one would need to traverse the contents of all these provenance stores. Such a task is essentially impossible if all systems adopt their own provenance representation, tailored to their execution technology. Instead, an inter-operability layer can be introduced, which allows the contents of individual stores to be exposed, and queries to be run uniformly across these stores.

To be uniformly queriable, provenance must be represented using ontological descriptions of what happened, so as to be execution technology independent; several such representations are emerging, e.g., Provenir [342], Web Provenance [213], OPM [299], PASOA [202], which are technology independent. One of them, the Open Provenance Model (OPM [299]) is based on a small set of provenance-related concepts and designed by a community of practitioners aiming at inter-operability. OPM is a lingua franca for provenance systems, since it allows provenance to be represented in a technology-agnostic manner and to
be serialised in various formats such as RDF and XML. During the third Provenance Challenge (which is discussed in Section 5.4), it has been demonstrated that many systems can export provenance in the OPM representation, import it, and successfully answer provenance queries about past computations in other systems. A model such as OPM offers the ability to exchange and exploit provenance information in an inter-operable manner (as per the definition given above). From this model, it is anticipated that query languages and APIs will be devised, and that query engines would be able to federate provenance information from multiple stores.

To sum up, to permit claims about data on the Web and its provenance to be evaluated retrospectively, the Open Provenance Vision postulates that all systems/components should be able to:

1. keep a record of provenance for any important data they produce (in their formats and repositories of choice);
2. follow conventions when exchanging data so that provenance can be traced across systems;
3. export provenance of such data using a common data model, such as the Open Provenance Model
4. answer provenance queries, structured over the common data model.

5.4 The Provenance Challenge Series

Over the years, a series of systems have been developed to track and exploit provenance in many different ways. Following a discussion session on standardisation at the International Provenance and Annotation Workshop (IPAW’06) [291, 34], a consensus began to emerge, whereby the provenance research community needed to understand better the capabilities of the different systems, the representations they used for provenance, their similarities, their differences, and the rationale that motivated their designs.

Hence, the first, second and third Provenance Challenges\(^1\)\(^2\)\(^3\) were successively set up in order to provide a forum for the community to understand the capabilities of different provenance systems and the expressiveness of their provenance representations. In the first and second challenges, the participating teams ran an agreed Functional Magnetic Resonance Imaging workflow, exported provenance information, and implemented pre-identified “provenance queries” asking typical questions about past execution of the workflow [297]. Key themes related to the provenance challenge activity are summarised in the tag cloud of Figure 5.3.

\(^{1}\text{http://twiki.ipaw.info/bin/view/Challenge/FirstProvenanceChallenge}\)
\(^{2}\text{http://twiki.ipaw.info/bin/view/Challenge/SecondProvenanceChallenge}\)
\(^{3}\text{http://twiki.ipaw.info/bin/view/Challenge/ThirdProvenanceChallenge}\)
As discussions indicated that there was substantial agreement on a core representation of provenance, the Open Provenance Model (OPM) [292] (subsequently revised by a broader committee [299]) was put forward as a data model by which systems can exchange provenance information. This model was the focus of a Third Provenance Challenge, where its suitability was practically evaluated by using it as the agreed model for provenance information exchange.

Figure 5.4 illustrates the kind of setup adopted for the Provenance Challenge series (specific variants were designed for each challenge, according to its specific aims). An application is running across multiple systems (here two systems 1 and 2). System 1 sends some data D1 to System 2, and also passes along an OPM representation of the provenance of D1. As System 2 receives D1, it ingests its provenance. System 2 exports a final result D2. Provenance queries pertaining to the origin of D2, in both systems, have been identified, and need to be answered by System 2.

Systems 1 and 2 may adopt different internal representations of provenance.
In the absence of common representation of provenance, adhoc pairwise conversions would be required (potentially requiring a number of converters quadratic with the number of representations). A provenance representation is more than a syntactic framework to express provenance. Indeed, for System 2 to be able to answer queries about the past of D2 in System 1, a common provenance representation needs to support a notion of dependencies, and must have a clear specification of inferences that can be made from these. The Provenance Challenge series has shown that expressing provenance queries in natural language can be ambiguous, a common data model for provenance may help formalize such queries, though this has not been investigated yet. The problem is similar for deciding what a correct answer to a provenance query is. Finally, a common provenance representation should be extensible, to support domain-specific queries.

5.5 The Open Provenance Model

From the outset, because precision matters when systems have to inter-operate, OPM was described in a technology-agnostic manner, both in natural language and using a formal notation. The key structure defined in the Open Provenance Model is an *OPM graph*, a directed acyclic graph aimed at representing data and control dependencies of past computations. The specification also outlined the kind of inferences that are permitted over such graphs.

The primary aim of OPM is to be able to represent how “things”, whether digital data such as simulation results, physical objects such as cars, or immaterial entities such as decisions, came out to be in a given state, with a given set of characteristics, at a given moment. It is recognised that many of such “things” can be stateful: a car may be at various locations, it can contain different passengers, and it can have a tank full or empty; likewise, a file can contain different data at different moments of its existence. Hence, from the perspective of provenance, OPM introduces the concept of an *artifact* as an immutable piece of state; likewise, it introduces the concept of a *process* as computational activities resulting in new artifacts. The Open Provenance Model is a model of artifacts in the past, explaining how they were derived.

A process usually takes place in some context, which enables or facilitates its execution: examples of such contexts are varied and include a place where the process executes, an individual controlling the process, or an institution sponsoring the process. These entities are being referred to as *Agents*. They are a cause (like a catalyst) of a process taking place.

A provenance graph aims to capture the dependencies between the abovementioned entities, i.e. an explanation of how entities influence others. Therefore, a provenance graph is defined as a directed graph, whose nodes are artifacts, processes and agents, and whose edges belong to one of following categories depicted
in Figure 5.5. An edge represents a dependency, between its source, denoting the effect, and its destination, denoting the cause.

![Figure 5.5: OPM Nodes and Edges](image)

The first two edges express that a process *used* an artifact and that an artifact *was generated by* a process. Since a process may have used several artifacts, it is important to identify the *roles* under which these artifacts were used. (Roles are denoted by the letter *R* in Figure 5.5.) Likewise, a process may have generated many artifacts, and each would have a specific *role*. For instance, a division process used two numbers, with roles dividend and divisor, and generated two numbers, with roles quotient and remainder.

A process is caused by an agent, essentially acting as a catalyst or controller: this dependency is expressed by the *was controlled by* edge. Given that a process may have been catalyzed by several agents, their respective roles as catalysts are also identified. The dependency between an agent and a process represents a control relationship, and not a data derivation relationship. It is introduced in the model to more easily express how a user (or institution) controlled a process.

In some situations, it may not be known which process generated some artifact $A_2$, but that artifact $A_2$ *was derived from* another artifact $A_1$. Likewise, it may not be known which exact artifact a process $P_2$ used, but that there was some artifact generated by another process $P_1$. Process $P_2$ is then said to have been *triggered by* $P_1$. Edges *was derived from* and *was triggered by* are introduced, because they respectively allow dataflow or process oriented views of past executions to be adopted, according to the preference of system designers.

To illustrate the model, the application of Figure 3.1 is revisited, and the provenance of the mashup, represented in OPM, is displayed in Figure 5.6. (Figure 5.6 is a generalization of Figure 3.2, which only contained artifacts and *was derived from* edges.) The bottom of the figure displays the mashup artifact. Plain edges represent the *was derived from* relation, whereas dotted edges repre-
sent used and was generated by relations. Edges can be subtyped; the subtype is represented by a label alongside the arrow. In the middle of the graph, the User (represented by an Agent) selected tuple \( tb2 \), which was used to parametrise a query, that extracted copies of \( p3 \) and \( b2 \) to construct the mashup.

It is beyond the scope of this document to provide a complete description of the Open Provenance Model. A few salient features are being exposed here because they characterize its expressiveness. OPM introduces the concept of account, which can be regarded as a graph colouring, identifying a graph subset containing a description of a past execution (by one or more witnesses). Multiple accounts can co-exist in a graph, and relationships such as overlaps, alternate or refinements are being defined. Within the scope of an account, the chain of was derived from edges is expected to be acyclic. Finally, OPM graphs can be enriched with optional time information, which is expected to be consistent with data derivation order.

### 5.6 Provenance in Open Systems

The Open Provenance Model was designed to represent the provenance of artifacts produced in open systems, by this it is meant systems whose topology may not be known at design time, and whose components, location and identity may only be discovered at runtime. In such systems (even without assuming malicious behaviour), it is expected that components provide descriptions of execution that do not align exactly, since they are not omniscient and can only operate on the basis of their local observations and knowledge.

To allow for multiple descriptions of a same execution to co-exist in a same provenance graph, OPM’s notion of an account can be seen as a colouring of graph that identifies a consistent subset of provenance information. Novel reasoning techniques are required to be able to reason with conflicting information provided by different accounts.

In an ideal world, provenance would be consistent and complete, providing the one and only one description of execution, for all activities and data, at all levels of granularity, involved in an execution. By consistent, it is meant that provenance is not contradictory: for instance, if two descriptions of execution co-exist, there must exist a common interpretation (for instance, because one is a refinement of the other). By completeness, it is meant that all descriptions intended to be produced have been provided. Provenance consistency and completeness may unfortunately not hold. Some components may not be provenance-aware, and may not be able to capture provenance. In that case, the components they interact with or wrappers may simply capture the interactions between them, without providing explanations of how data were internally derived inside the non-provenance component. Tan [380] categorized this kind of provenance as workflow and coarse-grained. It has been argued before that it rather is an in-
Figure 5.6: OPM Provenance of Mashup
complete description in the continuum of possible provenance representations. Inconsistency in provenance can be introduced by malicious behaviour, but even with non-malicious behaviour, inconsistency can come from differences in observations, inferred behaviour rather than directly observed behaviour, or behavioural descriptions relying on ontological descriptions that are not compatible.

Re and Suciu’s approximate lineage [332] is proposed as an alternative to complete provenance. Approximate lineage is purposefully designed to be incomplete, with respect to a provenance representation, so as to compress the representation, while still representing the most influential facts of a derivation. In the Open Provenance Vision, however, a framework has been designed to capture as much process assertions as possible, but it is acknowledged that, due to systems’ imperfections, provenance may be incomplete. This view differs substantially from Gehani et al.’s [169], who assume provenance to be complete for security applications.

Caroll et al. [66] make an interesting observation on the open world assumptions of named graphs, which also apply to the Open Provenance Vision. In an open world, the provenance of an artifact is an open-ended collection of assertions about this artifact (similarly to RDF and OWL descriptions of a resource are considered to be open ended). Indeed, any component can always provide further information about a past execution, that is relevant to the provenance of an artifact. However, accounts can help structure such descriptions. Some approaches, such as PASOA [202, 205] offers an option for a component to indicate that it has completed the set of assertions that it intended to make about a given execution. Transposed to an OPM context, this would allow an account to be sealed, effectively, allowing a reasoner to infer that a faithful component has not observed some action (meaning either it did not occur, or that it could not be observed by the observer).

5.7 Summary

The Open Provenance Vision is motivated by the architecture of today’s information systems, which tend to be loosely-coupled dynamic assemblages of components, deployed in multiple security domains, and relying on multiple communication and execution technologies. The Open Provenance Model is the first community-driven provenance representation, allowing cross-systems information flows to be documented in a coherent manner.

Such a representation presents multiple research challenges, including those described in the previous chapters, such as design of query languages, optimised storage, account-aware inference techniques, and reasoning in the presence of partial information. A natural application of OPM is tracking the provenance of information over the Web, a topic discussed in the following chapter.

The Open Provenance Vision itself requires further investigation to become
reality. As with all systems that involve unknown components, mechanisms are needed to ascertain that provenance assertions made by applications reflect what they have actually performed. Certification techniques could be a way of determining and qualifying their trusted base.

Provenance models need to be extensible to allow for domain-specific customizations. It has to be anticipated that independently designed components may extend a common provenance model in conflicting directions. Ontology reconciliation [444] may be required to resolve such incompatibilities. Despite some domain-specific differences, it is believed that one can still reason on aspects pertaining to provenance using a common provenance model.

Finally, the Open Provenance Vision is not just about a model but an architecture with APIs, to record and query provenance and all necessary conventions to track provenance when exchanging data. Such architecture needs to be designed to be scalable, to support complex topological configurations and to be secure.
Chapter 6

Provenance, the Web and the Semantic Web

The ultimate driver for the Open Provenance Vision described in Chapter 5 is the World Wide Web. The Web has become a global information space where the contents of databases are increasingly exposed directly. The Semantic Web facilitates the annotations of these data sets with RDF metadata, forming a global web of Linked Data [431]. Technologies such as mashups, tweets and RSS feeds integrate data from multiple sources, providing users with information customized for their needs. In this context, tracking provenance is perceived as a critical issue [213, 124] since it helps determine the quality of and trust one can put into data.

Issues in this area can be categorized in the following separate strands. (i) Given the importance of provenance, it is to be regarded as first-class data, itself to be exposed on the Web (Section 6.1). (ii) Semantic Web technologies are themselves being used, not only to represent provenance information, but also to query and reason over it (Section 6.2). (iii) Given the importance of metadata in the information discovery process, and the ease by which such metadata can be published on the Web, tracking the provenance of RDF-based information has also become a focus of investigation (Section 6.3). (iv) In the Semantic Web, not only can triples be asserted, but also they can be inferred. In such case, special techniques need to be devised to track their provenance (Section 6.4). Issues of data quality and trust, which are crucial over the Web, are investigated in Chapter 7.

6.1 Publishing Provenance on the Web

The principles of exposing information on the Web are now well understood [447], namely the use of Uniform Resource Identifiers (URIs) — a system for identifying resources globally — and protocols such as HTTP to access resources. Different
approaches for exposing provenance have been proposed in the literature, namely hypertext generation, RDF [448] views, and Webdav [438], which this section now discusses.

Two major constituents of provenance are identified by Zhao et al. [415]: annotations attached to objects (in a structured, semi-structured, or free text form), and derivation paths (from a workflow, query or program). They describe [415, 416] a dynamically generated hypertext of provenance documents, data, services and workflows. Their aim is to support Hendler’s vision of a Web of science [445]. This Web is created dynamically by means of ontology reasoning, annotation processing and link insertion.

SAM [306, 309], the Scientific Annotation Middleware, is a precursor system, pioneering the use of emerging Semantic Web technologies to separate the initial capture and storage of data and metadata from its subsequent presentation to others, hereby shifting the focus from up-front standardization to on-demand mapping of the data and metadata. In particular, SAM offers an electronic notebook capturing provenance of scientific experiment. By adopting the Webdav approach and URI identifiers, it allows navigation of provenance information. The pedigree browser allows for provenance browsing, and a portlet allows for graph visualization.

Hunter and Cheung [225, 92] presents Provenance Explorer, a system able to generate personalized views of the provenance relationships automatically using a combination of user input, semantic reasoning and access policies. Provenance information is extracted from a system that generates it (such as Kepler or Taverna), and inferences are made to build user views.

6.2 Semantic Web Techniques for Provenance

The use of Semantic Web technologies has been advocated to facilitate provenance acquisition, representation, and reasoning. On the one hand, RDF allows for resources to be referred to by URIs and its triple structure simplifies graph representation; the associated query language SPARQL [456] easily expresses their querying. Finally, OWL can be used for ontological definitions and reasoning. The tag cloud of Figure 6.1, produced from papers of the bibliography with a focus on Semantic Web techniques, identifies key issues in this area.

Zhao et al. [416] view provenance information from four different levels: organisational (incorporates who ran the workflow), process (a kind of event log, capturing inputs, outputs), data (captures data derivation) and knowledge (annotation about all the previous). Zhao et al. [419, 421] advocate the use of RDF [448] to represent provenance information, LSIDs\footnote{lsids.sourceforge.net} to identify resources, and ontologies to deliver a common semantic view of such data. Such a Web of
data can be visualised by a Semantic Web visualization tool, or can simply be navigated with a browser.

Chen et al. [81, 79, 78] use the term “augmented provenance” to denote the provenance of a piece of data and related semantic metadata about the process that led to such a piece of data. They explain how such semantic metadata can be captured from the workflow construction environment and the workflow enactment engine.

Sahoo et al. [343, 342] use the term “semantic provenance” to denote provenance in which domain knowledge and ontological underpinning have been incorporated. Like Zhao et al. [415], they advocate a semantic service, which incorporates domain specific knowledge into representations. Such an approach, which is adopted by several systems, relies on specialising data dependencies for specific application domains.

Myers et al. [308] observe that the disconnect between processes and data, where scientists have to manually operate heterogeneous tools with little integration, preventing experiments to be reproduced easily, and the loss of the collaborative contexts (notes, discussions, emails) are such that by the time results are published, most traces of the original process and data are inaccessible to the reader. To address this concern, they advocate the use of a semantic content management system, of which Tupelo is a core constituent. Tupelo [308, 397] is a middleware that provides a Web access protocol and Java API that interface with an RDF mapping of the Open Provenance Model. For this system, Wang et al. [397] propose a specialisation of OPM for GIS applications.
Golbeck et al. [183] demonstrate the flexibility of Semantic Web technologies to implement the Provenance Challenge. A provenance ontology was specified in OWL, and the description of the execution expressed accordingly. SPARQL was then used to implement the challenge queries. While the approach was demonstrated to work with another system, it makes some strong assumptions about application data, which typically remain in the control of the application.

Miles et al. [286, 404] exploit OWL reasoning capabilities to determine the semantic validity of past experiments. Subsumption is used to determine that the actual inputs and outputs of past experiments have the expected types; actual performed operations are checked to be conformant to a plan; legal constraints associated with input data sets are checked to verify patentability of results.

Halaschek-Wiener et al. [210] present a Semantic Web portal to annotate digital images and track their provenance, in the form of annotations such as submitter name and email. Provenance is browsable and actively used to enrich the user’s browsing experience.

In the context of a Chemistry lab, Frey et al. [163] envisage an RDF-based semantically-described world, in which a policy of “annotation at source” is enforced to track all information, including provenance, of manipulated digital and physical artifacts. Whenever data is processed, it is annotated with a description of the processing, effectively making its provenance explicit.

In the context of a virology application, Balis et al. [13, 12, 14, 11] propose a Query Translation Tool (QUaTRo) which allows users to construct queries to underlying provenance and data repositories with wizards and using technology-independent concepts, expressed in the terms of the domain familiar to end users.

### 6.3 Provenance for RDF

Whilst many authors advocate the use of Semantic Web technologies to represent and query provenance, Carroll et al. [66] take the opposite view, and identify the problem of provenance of triples (and other issues such as versioning and signature) in RDF. They propose named graphs as an entity denoting a collection of triples, which can be annotated with relevant provenance information. The RDF triple is the atomic assertion permitted on the Semantic Web; attaching authorship and origin to such assertions is consistent with the PASOA approach [196, 377] which considers collections of assertions, cryptographically signed, as the foundation of provenance information. The named graph proposal follows a series of approaches to address the problem of provenance and signature of RDF triples [333, 339, 234], where triples are extended with some construct allowing provenance to be expressed. None of these approaches however specifies how provenance itself should be represented; instead, they simply offer a placeholder for its representation.

Pediaditis et al. [319] argue that named graphs alone cannot capture prove-
nance information in the presence of RDFS reasoning and updates. Given two triples belonging to separate named graphs, which graph does any tuple inferred from these belong to? In other words, shared origin cannot be captured by named graphs alone. To remedy this problem, they propose a new construct, a graph-set, which allows them to capture and query provenance information adequately. This construct bears a strong similarity with the notion of account in OPM [299] since OPM edges may be asserted to belong to multiple accounts, meaning they belong to different descriptions (potentially from different observers).

Watkins et al. [399] explain how named graphs allow them to define a partitioning of their graphs, which then can be signed, effectively creating a Warrant Graph [66], and track the provenance of documents in a software version control repository.

Zhao et al. [420] envision subject-specific data webs that integrate multiple sources of scientific data (published in separate databases) in a seamlessly integrated view across the web. To allow scientists to maintain their trust across this web of independently evolving databases, provenance metadata is produced. It relies on RDF named graphs, over which they provide evidence for links and traces of how links are updated and maintained.

Gibson et al. [173] propose an approach that leverages named graphs and extensions to the SPARQL query language to create and manage views as a server-side function, effectively customising the presentation of provenance to users. With their approach, multiple operations can be aggregated in a single operation, hereby hiding details that are not considered important to the user. The notion of account and associated refinement relation in OPM are mechanisms that offer a similar form of abstraction.

Futrelle [164] also proposes that attribution and timing information for each triple be represented using Dublin Core creator and date properties, using an actor URI for the value of the creator element and an ISO 8601 timestamp for the value of the date element. Ding et al [124] introduce the notion of RDF molecule, an RDF graph partition, which offers a level of granularity between graph and triple. The use of molecules is demonstrated to track provenance of a graph.

Hartig [213] proposes a specialisation of OPM, referred to as provenance vocabulary, to describe the provenance of Linked Data over the Web. His model accounts for the creation and access of RDF data.

Dividino et al. [126, 127] focus on the problem of querying data and at the same time querying associated meta-knowledge such as provenance, authorship, recency or certainty of data. Their approach consists of meta-knowledge in RDF, and specifically represents provenance using an RDF serialisation of HOW-provenance [189]. Their proposed query language is an extension of SPARQL that allows meta-knowledge queries also to be expressed.
6.4 Knowledge and Web Provenance

McGuinness and Pinheiro da Silva introduce Inference Web [270, 271, 114] an extension of the Semantic Web which aims to enable applications to generate portable explanations for any of their answers. A key component is PML (Proof Markup Language) which includes support for Knowledge Provenance and Reasoning information; PML includes metadata such as authorship and authoritativeness of a source, but also reasoner assumptions (e.g. closed vs open world, naming assumption) and a detailed trace of inference rules applied (with variable bindings). Relationships capture notions of Consequent and Antecedents to a proof step, the succession of which consists of a proof. Human-readable explanations are derived from the proof markup language, and browsable representations can also be exposed to the user. PML is shown to be convertible to OPM representations in the third Provenance Challenge\(^2\).

Rio et al. [337, 336] describe how the Inference Web’s knowledge provenance can be used to semantically annotate maps and how this semantic information can help scientists understand and evaluate map products.

Fox and Huang [153] also adopt the term Knowledge Provenance (KP), to address the problem of how to determine the validity and origin of Web information by means of modelling and maintaining information sources, information dependencies, and trust structures. They argue that given that the Web will always be a morass of uncertain and incomplete information, it is possible to annotate Web content to create islands of certainty. Knowledge Provenance consists four levels of provenance that range from strong provenance (corresponding to high certainty) to weak provenance (corresponding to high uncertainty). Static KP (level 1) focuses on provenance of static and certain information. Dynamic KP (level 2) considers how validity of information may change over time. Uncertainty-oriented KP (level 3) considers uncertain truth value and uncertain trust relationships. Judgement-based KP (level 4) focus on social processes to support KP.

Gomez-Perez and Corcho [185] investigate the use of Problem-Solving Methods as a mechanism to extract higher-level knowledge-oriented provenance from existing provenance traces. Such higher-level provenance is intended to be easier to understand by users, and allows them to better grasp vast amount of provenance information.

6.5 Summary

As information dynamically flows across the Web, users need to have reliable means to obtain its provenance to decide whether they can trust information they access. Furthermore, the emerging sets of Linked Data, a pragmatic route to a semantic Web, form a network of pointers that allows automated navigation

\(^2\)http://twiki.ipaw.info/bin/view/Challenge/TetherlessPC3
to data that is relevant to the user. In this context, reasoners need explicit representations of provenance in order to make trust judgements about the information they use.

This chapter has shown how useful Web and Semantic Web technologies can be exploited to represent, make accessible, query and reason over provenance information. Semantic Web technologies themselves are also susceptible to a provenance problem. Techniques are emerging to express both the relevant authorship of atomic assertions on the Semantic Web, and detailed reasoning process when these items of knowledge are being inferred by reasoners. This latter work, which has mostly been developed independently of provenance research in the workflow community, has recently been demonstrated to be compatible with it, in the third Provenance Challenge. All these elements are indicative of a convergence towards the Open Provenance Vision for the Web.
Chapter 7

Accountability

Complex organisations and systems are typically formed by assembling multiple autonomous entities that agree to cooperate in order to achieve overarching objectives. Such assembly of autonomous entities is often regulated by constraints in the form of norms, contracts or policies, specifying the responsibilities of entities, their obligations, permissions, and penalties incurred when failing to deliver.

This approach to organising complex systems existed well before the prevailing use of the Web, yet the pervasive use of the Web offers new opportunities for creating such complex organisations, quickly, dynamically, and for negotiating the rules governing them on the fly. For example, virtual organisations [441], in which autonomous agents collaborate to deliver composite services, provide a means of exploiting such possibilities.

However, this presents end-users with challenges, since they are now confronted with a very dynamic and fluid environment, where it is difficult to understand which entity is responsible and accountable for which action. Here, the term “end-user” must be understood in its broadest sense: end-users may be organisation customers, the organisation’s participants, or even regulatory authorities.

Given that such applications are formed by assembling components dynamically, static methods that analyse their source code to infer their properties are not suitable. Even in systems that seemed to have substantial design and analysis before deployment, public cases of breach typically require an investigation, after the fact, to understand the origin of the problem\(^1\). This strongly suggests the need for approaches where systems faithfully document their execution, for potential future investigation.

Weitzner et al. [402] argue that, for information, “accountability must become a primary means through which society addresses appropriate use”. For them, “information accountability means the use of information should be transparent so it is possible to determine whether a particular use is appropriate under a

\(^1\)http://news.bbc.co.uk/1/hi/uk/7103911.stm
given set of rules, and that the system enables individuals and institutions to be held accountable for misuse”. Dynamically assembled systems need to be made accountable for users to gain confidence in them, i.e., their past executions must be auditable so that compliance with, or violations of, policies can be asserted and explained.

Weitzner et al. [402] note the similarity between accountability and provenance in scientific experiments. Provenance is a key enabler for accountable systems since it consists of an explicit representation of past processes, which allows us to trace the origin of data, actions and decisions (whether automated or human-driven). It therefore provides the necessary logs to reason about compliance or violations. This chapter reviews the use of provenance to make systems accountable, or to derive trust in results and systems. Given the critical importance of provenance in such contexts, audit results or derived trust will be reliable only if provenance has been faithfully stored and has not been tampered with. Therefore, this chapter begins with a review of security in the context of provenance (Section 7.1). It then discusses novel approaches to check compliance, and make systems accountable, using provenance (Section 7.2). It then surveys various approaches that have emerged to determine quality and trust of data, based on provenance (Section 7.3).

7.1 Provenance and Security

Braun et al. [45] make the case that securing access to provenance is different to “traditional data”. First, given that provenance contains relationship between entities (such as artifacts, processes and agents), each relationship reveals information about the parties in the relationship, and therefore needs careful handling. Second, a data product and its provenance may have different sensitivity. In an employee’s performance review, the data product is the review itself, which is available to the employee; its provenance encompasses the authors of the review, who have to remain anonymous. Hence, the employee can see the data but not its provenance. Symmetrically, a University applicant typically provides the names of the referees (and sometimes the actual reference in a sealed enveloped); the reference is to remain invisible to the applicant, while its provenance is known to the applicant.

Figure 7.1 summarises the key themes pertaining to security and provenance, they relate to access control, provenance integrity, non-repudiation of provenance, and sensitivity of provenance information. These themes are discussed in the following sections.
7.1.1 Access Control

In many systems or approaches, it is often considered that all users can see all provenance information. Yet, as argued by Braun et al. [45], this is not realistic, and hence access control needs to be introduced to repositories of provenance data [31].

Following the spirit of Braun’s distinction of data access and provenance access, Chong [95] revisits Cheney’s traces in the context of secure provenance. He considers two kinds of policies for application data (confidential or public) and two kinds of policies for the associated provenance (confidential or public). A location is said to have confidential provenance if it should not be known whether this location affects the result of the computation. He proposes extensional security requirements that restrict what information is revealed by the provenance trace, and that ensure the program result respects the provenance security policies.

Rosenthal et al. [338] make the case that role-based access control is not easily extended to support the security requirements of multiple stakeholders related to a given provenance trace. They propose to structure distinct concerns in a modular fashion to facilitate maintenance: namely, security, legally mandated privacy, and organizationally mandated privacy. They annotate OPM entities with access control attributes. They promote the use of ABAC (attribute-based access control) over RBAC (role-based access control), as the latter suffers from scalability problems, when policy becomes finer-grained and more attributes are involved, and roles have to be created for each combination of attribute, making the management of user to role mapping challenging.

Syalim [374] considers different levels of granularity of access control over provenance information. Policies can be associated with artifacts, processes, or paths of dependencies. Access rights are granted to groups of users.
Chebotko et al. [76] extends user views [99, 30] with security considerations: specifically, the notion of security view consists of a portion of a provenance graph for a given user, according to the access control policy prescribed by the user’s role. Access control policies are specified by workflow designers (in terms of the workflow building blocks (task, port, data channel), and inherited by the derived provenance produced at execution time.

Nagappan et al. [310] investigate the problem of confidential provenance in the context of Kepler; they introduce the notion of query sharing, by which users can explicitly share queries over provenance with their collaborators.

7.1.2 Provenance Integrity

Provenance vouches for the origin and authenticity of the data it relates to. For such a guarantee to hold, provenance itself must be preserved in its original form without any falsification or tampering. Integrity, in Information Technology terms, means that data remains unchanged while stored or transmitted.

Hasan et al. [216] are concerned with undetected rewrites of history, which occur when malicious entities forge provenance chains, in order to fake the authenticity of a document or data set. They consider provenance of a document as a linear chain of the principals performing actions on that document. They target chain forgeries that maliciously add new chain entries and make after-the-fact modifications, and offer the following integrity assurances:

- An adversary acting alone cannot selectively remove other principals’ entries from, or add entries in the chain, without being detected by an auditor.
- Two colluding adversaries cannot add entries of other non-colluding users “between” them in the chain without being detected by the next audit.
- Once the chain contains subsequent entries by non-malicious parties, two colluding adversaries cannot selectively remove entries associated with other non-colluding users between them in the chain, without being detected by the next audit.

Their solution consists of propagating cryptographic checksums along the chain, allowing entries to be sequentially validated.

Factor et al. [136] consider the problem of long-term archiving of data, and note that in most cases, digital objects cannot be preserved without any change in the bit stream, and that digital librarians have to modify the original object to have the ability to make it available in the future. This leads to a paradox since preservation entails change, while authenticity needs fixity. To address this concern, they rely on provenance to track changes that occur to data during the preservation activities, and they preserve provenance alongside data.
Gadelha [167] considers some security properties of provenance — integrity, confidentiality and availability — in the context of intellectual property conflict claim resolution and of the reliable chronological reconstitution of scientific experiments. To address these, they secure data authorship and temporal information of provenance records. To this end, they use the time-stamping protocol in addition to cryptographic signatures. So, given a raw provenance record, they digitally sign the record, compute a hash value, which they send to the times-tamping server, which signs the hash and the current date.

Gehani and Lindqvist’s aim [169] is to reliably determine the lineage of a piece of data. Their model of provenance is one in which “lineage metadata” is communicated along application data; their concern is lineage space requirements increase, as data get processed. Their solution is not to relay provenance but to leave lineage details at the nodes where the operations occur and forward cryptographic commitment to prevent repudiation. Their solution [170, 171] provides operation chain non-repudiation since the signed lineage of each input is extracted and added to the metadata of each output, itself, being hashed and signed.

Zhang et al. [412] define the pedigree forgery attack, as the situation where an attacker presents to the data recipient a pedigree and data product, such that the pedigree does not accurately describe the data product’s authorship. To prevent an attack, they introduce a cryptographic proof in the audit records so that to the data recipient can check that the pedigree associated with a data product is correct. By a cryptographic protocol, they ensure that the output of a workflow step matches the inputs of its successor. Their approach is not as strong as chains discussed by Hasan et al. [216] since only input/output matches are protected and not the overall chain.

7.1.3 Liability and Accountability for Provenance

An important consideration in any provenance system is the accuracy or objectivity of the assertions recorded [377]. Most systems capture statements about some aspect of a process by some of its components. From a more abstract viewpoint, such statements are however only a subjective view of that aspect by a component. It can be difficult sometimes, if not impossible, to determine how closely this view tallies with actual reality. Therefore, it becomes paramount to establish a clear link between a component and an assertion that it is responsible for. Such a link, which can be provided through digital signatures [377, 196], ensures that responsibility and corresponding liability is attributable to the correct component.

Cryptographic signatures let one determine the source of a metadata assertion [257] (or, more precisely and more generally) the identity of the system, person or organization that stands behind the assertion, and to establish a level of trust in this identity. One can have near-absolute confidence that the source
possessed the requisite public/private key pair (assuming that the private key has not been compromised, and the key pair has not been revoked); the level of trust is in the binding of identity to possession of the key pair.

While it is important to be sure that provenance has not been tampered with, it is also crucial that provenance is faithful, i.e. consisting of an accurate description of execution. Introducing digital signatures on provenance assertions makes an explicit link to the trusted base, discussed in Section 3.5, since it allows provenance queriers to decide whether assertions are issued by components that are trusted. Unfortunately, making such a decision in an open environment is not as straightforward as it seems. An unforgeable proof of a component’s name does not mean this component faithfully asserted provenance; provenance certification techniques need to be developed to provide better insurance about the trusted base. Even in systems where a strong formal foundation links provenance to execution semantics, it is important to ensure that such systems have been made secured, so that provenance has not been tampered with. Likewise, where complex libraries whose execution is described by ontological descriptions are used, system administrators will have to leverage trusted library mechanisms offered by runtime systems and application servers, or approaches such as Trusted Computing [450]. The rest of this chapter assumes that the trusted base can be established by queriers, and that provenance contents faithfully reflects past executions.

Cryptographic signatures, as seen in this section, address several properties of provenance security: non-repudiation and liability for the actions performed by components, authentication of the individual assertions, and finally integrity and unforgeability of the assertions.

7.1.4 Sensitivity of Provenance Information

In a basic example, the assertions pertaining to a message exchange between two components would simply contain the contents of that message verbatim. Depending on application domain requirements, however, parts of the message may need to be obscured or transformed in some way when they appear in a provenance record. A good example of this is found in the electronic health care records domain [230], where privacy requirements mandate that patient identity on health care records be anonymized if the information on the record is being utilized for non-diagnostic reasons (for example, to answer statistical questions about medical processes).

If provenance is utilized in such a context, then certain data items (such as patient identifiers) that are transmitted in clear text in the original message exchange between actors must be obfuscated in some manner when stored as part of any provenance record. An approach, referred to as documentation style [196] has been proposed to address this problem: it applies a transformation to application messages before storing them in a provenance store.
7.2 Accountability

Figure 7.2 displays the tag cloud that summarises the key issues pertaining to accountability and provenance. In this context, a system is accountable if it can provide explanations for its actions, if its past actions are accountable, and if it can be demonstrated that its processes and decisions are compatible with rules, policies, or broadly regulations. The tenet of accountability is to keep a detailed record of past activities explaining how every data item is derived, and what triggered every action. In other words, with explicit representation of provenance, one can make systems accountable: provenance provides the necessary evidence which makes systems transparent and allows an auditor to determine whether policies are satisfied.

Figure 7.2: Accountability Tag Cloud

Groth et al. [203] derive, from philosophy and history, several principles that documentation of the past, in the form of process assertions, should implement, so that it can be seen as a proper evidence of past execution. (i) Assertions must be based on observations made by software components, according to the principle of data locality; (ii) Each assertion making up documentation of the past for a computer system must be attributable to a particular software component. Of course, in some cases, it is reasonable to expect a component may make inferences about the world, and events that are not directly observable, but such inferences have to be marked as such, so that an auditor can distinguish observation from inferences (or guesses).

Aldeco and Moreau [2] propose a provenance-based architecture for an ac-
countable system, and apply it to the data protection act, the UK implement-
tation of the EU directive on private data. The architecture identifies multiple
roles, such as data subject (the owner of private data), data controller and pro-
cessor (those that manage and process private data), and the auditor, who aims
to determine compliance of the system. Furthermore, the architecture relies on a
provenance store acting as a trusted, secure and persistent repository of evidence,
which the auditor can trawl to verify compliance.

Curbera et al. [109] distinguish business provenance from business activity
monitoring (BAM). The latter is mostly focused on real-time access to business
performance indicators, including interactive and real-time dashboards and
proactive alert generation, whereas the former adds a historical perspective to
BAM that enables root cause analysis and process discovery. With a focus on
business compliance, they consider a typical business workflow, which differs from
scientific workflows, since documents are exchanged by emails, published on and
downloaded from the Web, and submitted to specific applications. Their model
allows for data derivations to be inferred, by matching identities of business ar-
tifacts stored in a central provenance store.

Miles et al. [281] consider the problem of contract violation, i.e., where the
responsibilities of the different parties and expectations are not met. They rec-
ognize that long-term business relationships require flexibility in the face of miti-
gating circumstances, i.e. assumptions that do not conform to those of the
contract. Their purpose is to design a system that can handle such mitigating
circumstances automatically. To do so, they require a reliable documentation of
what has occurred and how it caused a violation. Determining the cause of the
violation is performed by executing provenance queries; after checking whether
mitigating circumstances prevails, they apply the appropriate policy to handle
the violation.

Likewise, Vazquez-Salceda and Avarez-Napagao [391] consider the problem of
runtime governance of service-oriented architectures. Their approach relies on
detecting violation states that agents may enter into, and the definition of the
sanctions that are related to the violations. An enforcement component relies on
a rule engine to reason about the evidence stored in a provenance store to take
decisions and plan actions whenever violations are observed.

Cirillo [97] propose a programming language with a type system that allows
the verification of claims regarding the provenance of objects. It relies on a
provenance-based semantics, which tracks the provenance of method parameters
and results. The kind of claims that can be verified with this approach include
the owner of some data, the path through which it transits, and access control
properties.

Philip et al. [320] argue that provenance could be used in e-Social science, to
track evidence-conclusion chains, so that decisions can be justified in terms of ev-
idence and applied reasoning. They discuss philosophical, ethical and legal issues
that such an approach would raise. Chorley et al. [96] discuss a representation of
provenance and its application to evidence-based policy assessment.

In the context of copyright management, Ockerbloom [312] notes that to reliably determine the rights to a work, one may have to understand and record the provenance of a work, the provenance of its rights, and the provenance of the information used in rights determination. The factual assertion chains can be complex in their structure, and involve varying degrees of uncertainty. Evaluating the reliability of such assertion chains and reasoning with incomplete provenance are important issues to be considered by the provenance community.

7.3 Data Quality and Trust

As users delegate important tasks to systems and endow them with private data, it is crucial that they can put their trust in such systems. Accountability as defined previously is a way by which trust can be built, since action transparency and audit help users gain trust in systems. However, users may not always want (or have the resources) to audit systems; instead, they would like to be given a measure of trust, which they can rely upon to decide whether to use a system or not. The topic of trust has been extensively reviewed [182, 443, 457, 460]. Trust is usually based on an agent’s own experience with respect to past interactions with other agents, whereas reputation draws upon information gathered from third-parties. This section reviews work that derives a notion of trust in data from the provenance of data. In some context, quality of data can be similarly derived from its provenance. Given a method to compute trust in data, it may then become possible to derive trust into systems by “aggregating” trust about all the data they produce.

Golbeck reviews trust issues on the World Wide Web [182] and identifies provenance as a key element necessary to derive trust. Golbeck et al. [181] propose a trust inference algorithm that operates over Semantic Web data. It is applied to the Friend of a Friend ontology, used in social networking [184]. Likewise, Harth et al. [212] also argue for the social provenance of data, identifying the people or groups of people who originated data.

One important issue in determining data integrity is the trusworthiness of source providers and intermediate agents [115]. Dai et al. [115] propose some recursive functions that computes trust scores for data, depending on the trust of the information used to generate it and the trustworthiness of parties that handle it. They rely on a very simple generation path with a linear topology. Whilst this seems to be a step in the right direction, this work suffers from some limitations, such as the nature of the computation being performed; more complex topologies, and potentially multiple accounts of execution should be considered to make a system usable in practice.

Rajbhandari et al. [323, 324, 328] use provenance information to evaluate whether an abstract workflow description has been adhered to, and to enable a
user executing a workflow-based application to establish trust in the outcome of a physical workflow. Their notion of trust is compositionally derived from the trust in processes (i.e. abstract workflow), the trust in services (i.e. physical workflow) and trust in source and intermediate data. A decision tree is used to decide whether a final data product can be trusted or not. Their original model offers a binary outcome (trusted or not) and was then extended to allow for a range of trust values [325].

Prat and Madnick [321] argue that believability is an essential characteristic of data quality and provide a precise approach to compute its measure using provenance. The measure is structured in terms of several building blocks: metrics for assessing the believability of data sources, metrics for assessing the believability of data resulting from process execution, and global assessment of believability.

Hartig and Zhao [214] define a notion of quality derived from the first author’s Web provenance vocabulary [213]. They apply their approach to timeliness of data.

When a contract has to be drawn with a new business partner, there may be very little prior interaction with the partner (and possibly no reputation) to derive a notion of trust. Hence, one needs the means to decide whether to trust the contract itself. Groth et al. [200] propose an approach to measuring the success of prior contract executions, and a notion of contract similarity, which they use to determine the trustworthiness of contract proposals.

A challenging aspect of the research presented so far in this section is in establishing that provenance-based trust is of value and can indeed be regarded as a measure of trust. The provenance research community has still to design a compelling and objective way to establish the suitability of the proposed techniques. Various forms of evaluation have been considered. In some cases, simulations over randomly generated data establish that the trust value approximates future behaviours [200]. Others simply consider the performance of their algorithms, without studying the quality of their outputs [115]. In other cases, trust values can be subjectively evaluated by users, very much like recommendations in recommender systems [184].

The ULDB model [25, 24, 26] underlying Trio [403] combines not only data, but also the accuracy and provenance of the data, in a database. A salient feature of the approach is a query language that handles data, accuracy and provenance in an integrated manner. One benefit is that provenance can be used for understanding and resolving uncertainty. To this end, strict properties of provenance are introduced that amount to considering a class of uncertainty that can be captured by (i) a finite set of base facts that are either mutually exclusive or independent, and (ii) possibly correlated data derived from these base facts in a way that propagates but does not affect uncertainty. These properties seem to differ from Ockerbloom’s (see Section 7.2) who also considers uncertainty on provenance chains. Further work is require to develop models that capture properties of provenance in the presence of uncertainty, whether of data
or of provenance.

### 7.4 Alternative Approaches

This section has surveyed how provenance can help in establishing accountability and trust. It has to be recognized that provenance itself may not be sufficient in that endeavour. Research in the database community tackles similar issues, and results in solutions that can complement extant provenance work. A line of investigation would be to leverage such database techniques for provenance.

Vaughan et al. [461] introduce the notion of evidence-based audit, which relies on a secure kernel that wraps critical primitives with wrappers that log the invocation of these primitives, with a proof, acting as evidence that the operation had been allowed.

Miklau et al. [451] define accountability in databases as the ability to analyze past events to detect breaches, maintain data quality, and audit compliance with policies. Yet, retaining historical information can pose a serious threat to privacy [459]. So, privacy and accountability are both legitimate goals, but they can be at odds. Miklau et al. [451] argue that data retained by databases should be accessible through a legitimate interface, to offer transparency. Hence they call for a redesign of traditional databases. Lu and Miklau [449] discuss the trade-off between full auditing of database transactions and privacy protection. They introduce two kinds of rules for selectively removing and obscuring sensitive data from the record of the past. Similar concerns would exist if a provenance-based approach is adopted to offer accountability.

Pavlou and Snodgrass [454] propose forensic techniques to analyse database tampering, once an intrusion has been detected. The analysis helps ascertain when the intrusion occurred, what data was altered, and ultimately who the intruder was.

### 7.5 Summary

Information transparency is a desirable design principle for next generation Web-based information systems. By giving users the means to understand how information is produced and used, systems can be made accountable for their actions and the information they offer. Provenance is therefore a crucial technology that can provide such a transparency. It will be effective in achieving such a goal only if its faithfulness can be established, if it cannot be forged, if its integrity can be ascertained, and if its source can be authenticated. Cryptographic techniques effectively applied to provenance can ensure strong properties, such as integrity and authentication. Certification techniques are also required to provide guarantees that systems assert provenance that is faithful to their actual execution.
This chapter has reviewed several approaches that adopt provenance as the foundational layer of accountable systems, which allow their actions and information flows to be audited, and their compliance or violation to rules and policies to be determined. Such strong capabilities — namely, information transparency, auditing capabilities, and compliance detection — provide users with the means to decide whether they can trust systems and information.

In practice, such a notion of trust needs to be derived for users, since they do not have the skills, time and will to audit systems and review their information flows. Several approaches have been proposed to infer measures of trust, quality or believability from provenance information. A research challenge that results from this kind of work is in establishing the soundness and suitability of such measures. Formal foundations on this topic are scarce to date, and empirical evaluations have so far been limited to very small scale simulations and user trials.
Chapter 8

Conclusion

This article has postulated that “society can and should reliably track and exploit the provenance of information on the Web”. By means of a journey through the provenance literature, arguments have been developed to establish this thesis. They are summarised as follows.

1. By means of blogs, social networking, news feeds, instant messaging, and collaborative tools, the Web has become a global and universal communication medium, exploited by businesses, governments and individuals. Concerns of privacy and security are being inevitably raised, as people lives become more dependent on the Web. To address these, information and systems available over the Web can be made accountable by introducing information transparency, auditing capabilities, and compliance checking tools; in turn, accountability allows trust networks to be developed. Chapter 7 has argued that provenance provides solid foundations for building accountable systems, and with the appropriate security techniques to ensure its security, provenance can help establish the quality of data.

2. However, provenance is non-existent in today’s Web applications, but is emerging in niche areas. Chapter 6 has reviewed state-of-the-art efforts to make provenance accessible on the Web, and integrate with Semantic Web technologies underpinning the Linked Data effort.

3. Given that information flows across multiple services over the Web, being transformed, filtered, processed and repackaged in many different ways, a representation of provenance has to be assembled by bringing evidence of local transformations and derivations into a coherent whole. This is the purpose of the Open Provenance Vision, and the community-driven Open Provenance Model, which were discussed in Chapter 5. Integrating the Open Provenance Vision with the Web architecture is therefore a critical step in ensuring that provenance of Web data can be tracked and used.
4. For information provenance to be traceable over the Web, each information system or service involved in a global information flow has to track provenance in its local activities. Given that the majority of research has been conducted by the workflow and database communities, Chapter 4 has surveyed the key issues these communities have investigated. A point worth noting is that attention has recently been given to systems that involve humans who affect decisions and information flows, and therefore are entities belonging to their history.

5. As research is being undertaken by different communities, multiple notions of provenance have emerged (even within a single community). Chapter 3 has introduced a general definition of provenance, which was shown to be compatible with the prevailing definitions, and could suitably serve as a definition for the origin of information over the Web.

6. The Citebase tool was exploited to analyse an extensive bibliographical database on provenance, and identify research fronts, in Chapter 2. These research fronts, which take the form of co-citation clusters, concern topics that are aligned with the argumentation developed in this article: by means of research in databases and workflows, in the Open Provenance Model and the Provenance Challenge activity, in Semantic Web, and in security, the research community has already begun laying out the foundations for provenance on the Web.

8.1 The Benefits of Provenance on the Web

Having built the case for the thesis “society can and should reliably track and exploit the provenance of information on the Web”, it is now time to revisit some of the drivers for provenance discussed in Section 1.1, and recast them in a Web context.

After discovering a workflow close to their needs, a scientist iteratively adapts and modifies it for their experiment, which relies on several data sets, imported from highly curated databases, and other online data sets available from their research colleagues. When satisfied with the latest run of the workflow, the scientist makes all the data available over the Web, with their provenance, which has been captured by the workflow engine and all invoked libraries and services. The edited workflow is also published on the Web, with its provenance, including credits that can be automatically generated from its provenance. The scientist writes an article about their experiment, which includes a series of plots (not dissimilar to the one of Figure 2.1).

A journal publishes the article online, including its full provenance, with reference the data sets used in the experiment and the workflow; the plots contained in the article have also their provenance on the Web, and it is possible to verify how
they were produced, with which data sets. Since the provenance of any data item published over the Web is now accessible, a series of services have been deployed for the scientific community. The service science-replay.org is capable of replaying executions to verify results, or replay the workflows against new data sets. The service science-quality.org analyses the actual flow of execution that was involved in the generation of data sets; from this, it infers error propagation and a measure of the quality of data. The service license-comply.org identifies all initial data sets, workflows, services and libraries invoked in an experiment, and checks their licensing conditions are respectively satisfied. Overall, by making provenance of all data available over the Web, the scientific process is being made more transparent, scientific results can be better verified, and reused; provenance is therefore a key enabler of Hendler’s vision of a Web of science [445].

Provenance is useful well beyond the world of science. Consider a shopper who finds some nice vine tomatoes in a supermarket, with a label indicating they were produced at a local farm. Using their mobile phone, they access the online information about that product, including its provenance. The supermarket, and its supply chain, including the transport company and the farm, believe that openness gives them a competitive edge, and therefore, expose all provenance information on the Web. Independent online services can exploit this information: www.yourcarbonfootprint.com computes the carbon footprint of these tomatoes and its actual food miles; e-organic.com is a service that provides an independent measure of the organic nature of products whose provenance is available online. (This measure itself is available online, and its provenance can be obtained and audited.) With explicit provenance, shoppers could be offered a wealth of services that help them assert the quality of the product they buy (online or offline). Of course, realistically, shoppers cannot be expected to undertake such extensive analysis themselves, for every item they buy, especially if low price. Instead, they may rely on online shopping assistants making recommendations for them, according to their preference.

8.2 Future Research

This survey has identified key building blocks that would be necessary to accomplish our vision of tracking and exploiting provenance of information on the Web. However, this vision is by no means implemented; it remains confronted to multiple conceptual and practical research challenges:

- Foundations: how can semantic frameworks and associated definitions be extended to a global information space such as the Web?

- Representation and Architecture: the Web is about standards, and these need to be agreed to represent, record and query provenance.
• Systems: systems need to be built to cope with the scale of information involved in this endeavour.

• Certification: certification approaches need to be developed to reliably identify the method by which provenance is generated, and its degree of faithfulness.

• Humans in the loop: non-intrusive methods need to be developed to better capture user’s actions and reasons for performing actions, and visualisations techniques for provenance need to be devised for users to easily understand and navigate such information.

• Reasoning: novel techniques need to be devised to deal with provenance that is incomplete, conflicting, or not authoritative.

• Accountability: new services need to be conceived that can exploit provenance to offer auditing capabilities, compliance checks, and ultimately help users decide whether they can trust information over the Web, using objective criteria.

Importantly, there is also a human challenge to achieving such a technical vision. While it is highly desirable to understand the origin of decisions and Web information, privacy is becoming an issue if all user’s actions on a computer are monitored and archived. The Web is recent and is being used as a quick dissemination tool. However, to make the Web provenance-aware, mentalities have to change: it is no longer sufficient to publish data, but associated provenance must also be made available. While tools may assist in this task, this inevitably increases the human effort involved. Hence, the cost-benefit of publishing provenance needs to be analysed. This survey has shown that when there is a strong requirement for accountable information, the benefits of provenance largely outweigh the cost of tracking and maintaining it.

Acknowledgements

Thank you to Chaomei Chen for his help with CiteSpace, to Danius Michaelides for his help with scripts for processing bibliographical data, and Ewa Deelman, Paul Groth and Simon Miles for providing feedback on this survey. Thanks to the anonymous reviewers for their detailed and constructive reviews.
Provenance Bibliography


Lawrence Berkeley Laboratory.

[23] Jeanne Behnke, John Moses, and James Byrnes. Archive issues associated
with nasa earth science datasets. In Eos Trans. American Geophysical
Union, Fall Meeting 2008, volume 89, 2008. abstract IN11C-1046, (url: 
http://www.agu.org/cgi-bin/wais?mm=IN11C-1046).

[24] Omar Benjelloun, Anish Das Sarma, Alon Halevy, Martin Theobald,
and Jennifer Widom. Databases with uncertainty and lineage. The
s00778-007-0080-z).

Uldbs: databases with uncertainty and lineage. In VLDB ’06: Proceedings
of the 32nd international conference on Very large data bases, pages 953–
703/).

An introduction to uldbs and the trio system. IEEE Data Engineering

[27] Dave Berry, Peter Buneman, Michael Wilde, and Yannis Ioannidis, editors.
(url: http://www.nesc.ac.uk/esi/events/304/).

[28] Deepavali Bhagwat, Laura Chiticariu, Wang-Chiew Tan, and Gaurav Vi-
jayvargiya. An annotation management system for relational databases.
In VLDB ’04: Proceedings of the Thirtieth international conference on
Very large data bases, pages 900–911. VLDB Endowment, 2004. (url: 

Zoom*userviews: Querying relevant provenance in workflow systems. In
Christoph Koch, Johannes Gehrke, Minos N. Garofalakis, Divesh Sriv-
astava, Karl Aberer, Anand Deshpande, Daniela Florescu, Chee Yong
Chan, Venkatesh Ganti, Carl-Christian Kanne, Wolfgang Klas, and Erich J.
Neuhold, editors, VLDB ’07: Proceedings of the 33rd international con-
ference on Very large data bases, pages 1366–1369. ACM, 2007. (url: 

[30] Olivier Biton, Sarah Cohen-Boulakia, Susan B. Davidson, and Carmem S.
Hara. Querying and managing provenance through user views in scientific
workflows. In International Conference Data Engineering (ICDE’08), pages


[113] Sergio Manuel Serra da Cruz, Maria Luiza M. Campos, and Marta Mattoso. Towards a taxonomy of provenance in scientific workflow management systems. volume 0, pages 259–266, Los Alamitos, CA, USA, 2009.


[267] Chris J Martin, Mohammed H Haji, Peter M Dew, Michael J Pilling, and Peter K Jimack. Semantically enhanced provenance capture for chamber model development with a master chemical mechanism. Philosophical


[291] Luc Moreau and Ian Foster, editors. *Provenance and Annotation of Data — International Provenance and Annotation Workshop, IPAW 2006*, volume


provenance challenge. *Concurrency and Computation: Practice and Ex-
cpe.1233).

[299] Luc Moreau (Editor), Beth Plale, Simon Miles, Carole Goble, Paolo Missier,
Roger Barga, Yogesh Simmhan, Joe Futrelle, Robert McGrath, Jim My-
ers, Patrick Paulson, Shawn Bowers, Bertram Ludaescher, Natalia Kwas-
nikowska, Jan Van den Bussche, Tommy Ellkvist, Juliana Freire, and Paul
Groth. The open provenance model (v1.01). Technical report, Univer-
uk/16148/1/opm-v1.01.pdf).

[300] Pierre Mouallem, Roselyne Barreto, Scott Klasky, Norbert Podhorszki,
and Maden Vouk. Tracking files in the kepler provenance framework.
In *Proceedings of 21st International Conference on Scientific and Statisti-
cal Database Management (SSDBM’09)*, pages 273–282, New Orleans, LA,
USA, 2009. (doi: http://dx.doi.org/10.1007/978-3-642-02279-1_21).

Provenance-aware storage systems. In *ATEC ’06: Proceedings of the an-
nual conference on USENIX ’06 Annual Technical Conference*, pages 43–
www.usenix.org/events/usenix06/tech/muniswamy-reddy.html).

[302] Kiran-Kumar Muniswamy-Reddy, Uri Braun, David A. Holland, Pe-
ter Macko, Diana Maclean, Daniel Margo, Margo Seltzer, and Robin
Smogor. Layering in provenance-aware storage systems. In *Procee-
dings of 2009 USENIX Annual Technical Conference*, San Diego, CA,

a cloud provenance-aware. In James Cheney, editor, *TAPP’09: First work-
shop on on Theory and practice of provenance*, San Francisco, CA, Febru-

[304] Steve Munroe, Simon Miles, Luc Moreau, and Javier Vázquez-Salceda.
PrIME: A software engineering methodology for developing provenance-
aware applications. In *ACM Digital Proceedings of the Software Engineering
and Middleware Workshop (SEM’06)*, pages 39–46, New York, NY, USA,

[305] Michi Mutsuzaki, Martin Theobald, Ander de Keijzer, Jennifer Widom,
Parag Agrawal, Omar Benjelloun, Anish Das Sarma, Raghotham Murthy,


[314] Leon J. Osterweil, Lori A. Clarke, Aaron M. Ellison, Rodion Podorozhny, Alexander Wise, Emery Boose, and Julian Hadley. Experience in using a process language to define scientific workflow and generate dataset


[331] Sudha Ram, Jun Liu, Nirav Merchant, Terrill Yuhas, and Patty Jansma. Toward developing a provenance ontology for biological images. In Eighth


[348] Carlos Scheidegger, David Koop, Emanuele Santos, Huy Vo, Steven Callahan, Juliana Freire, and Claudio Silva. Tackling the provenance challenge


Jing Zhang, Adriane Chapman, and Kristen LeFevre. Fine-grained tamper-evident data pedigree. Technical Report CSE-TR-548-08, University of


Bibliography


