Data Provenance: A Categorization of Existing Approaches

Boris Glavic, Klaus Dittrich
University of Zurich
Database Technology Research Group
glavic@ifi.unizh.ch, dittrich@ifi.unizh.ch

Abstract:

In many application areas like e-science and data-warehousing detailed information about the origin of data is required. This kind of information is often referred to as data provenance or data lineage. The provenance of a data item includes information about the processes and source data items that lead to its creation and current representation. The diversity of data representation models and application domains has lead to a number of more or less formal definitions of provenance. Most of them are limited to a special application domain, data representation model or data processing facility. Not surprisingly, the associated implementations are also restricted to some application domain and depend on a special data model. In this paper we give a survey of data provenance models and prototypes, present a general categorization scheme for provenance models and use this categorization scheme to study the properties of the existing approaches. This categorization enables us to distinguish between different kinds of provenance information and could lead to a better understanding of provenance in general. Besides the categorization of provenance types, it is important to include the storage, transformation and query requirements for the different kinds of provenance information and application domains in our considerations. The analysis of existing approaches will assist us in revealing open research problems in the area of data provenance.

1 Introduction

With the increasing amount of available storage space and the acceleration of information flow induced by the internet, a growing interest in information about the creation process and sources of data has developed.

Scientists in the fields of biology, chemistry or physics use data from so-called curated databases. Most of the data stored in a curated database is a result of manual transformations and derivations. Researchers that use data from a curated database are interested in information about the sources and transformations that were applied to this data. This information can be used to assess the data quality, examine or reconsider derivation processes or re-run a specific experiment.

Data warehouses are used to integrate data from different sources and with different data representations, and to analyze the integrated data. Such analyses could benefit from information about the original data sources and transformations used to create the data in the data warehouse.

In workflow management systems and more generally in systems based on SOA (service oriented architecture), several, probably distributed services are used to accomplish complex computational tasks. For the user it would be interesting to understand how a result from such a computation was created.

GIS-systems manage spatial data. Many applications e.g. in geography and engineering hydrology use GIS-systems to store data with spatial dimensions. Additional to the type of provenance information mentioned already, these applications would be interested in spatial provenance properties, such as information about the origin of merged or split objects.

When combined with timestamped secure certificates, data provenance could be used to investigate copyright issues. For example, a scientist could prove that he was the first one to perform a certain experiment or that he is the creator of a specific piece of information.

In business applications, document management systems are used to manage the company documents and permit multiple users to work on the same document. A supervisor can use provenance information to gain a better understanding of the workflow in his company or find the origin of erroneous documents or document parts. The same ideas apply to distributed software development tools.

Other application domains that could benefit from provenance information include interactive statistical environments, visualization and KDD (knowledge discovery in databases).

While this broad range of application areas would benefit from provenance information, the type of provenance data, manipulation and querying facilities needed differ from application to application. Therefore, we try to find out the differences and similarities between the various application and data model provenance needs and present a general scheme for the categorization of provenance. By defining this scheme and applying it to existing work we hope to reveal open questions in the area of data provenance.

The remainder of this paper is organized as follows. In section 2 we discuss existing research approaches for managing provenance and introduce a consistent terminology. In section 3 we present our categorization scheme based on a generic view of provenance. This scheme is used in section 4 to study the properties of some of the approaches discussed in section 2. Finally in section 5 we cover open research questions and conclude in section 6.

2 Overview of Existing Approaches

A broad diversity of terms is used in the literature for provenance related concepts. To prevent confusion we introduce a consistent terminology for the most important concepts in data provenance and relate them to the terminology used by other researchers. A number of synonyms like *lineage* or *pedigree* are used for data provenance. We decided to use the term *provenance*, because it is short and intuitive. The terms *provenance model* and *provenance management system* are used to distinguish between a conceptual model for provenance and a system for the management of provenance information. Provenance was studied for different data models and levels of detail. Each data model has its own terminology for data and hierarchical structures. We use the term *data item* for a structural unit of data, which is the target of provenance management and the notion *level of detail* for the granularity of a data item. For example a data item could be an XML-document, a database tuple, a database relation, a database schema construct or a a file in a file system. Tuple and relation are two possible levels of detail for a data item in a relational database. To abstract from different systems for storing data like relational databases, object oriented databases and file systems, we refer to these storage systems as *data repositories*. The term *hierarchy position* is used for the position of a data item in the structural hierarchy of a data model. For example, for database tuples the hierarchy position could be represented by the database and relation the tuple is stored in.

There are two basic views of provenance. The first one describes the provenance of a data item as the processes that lead to its creation and the other one focuses on the source data from which the data item is derived from. In contrast to [Tan04], where these concepts are called *provenance of data* and *provenance of a data product*, we use the terms *source provenance* and *transformation provenance* because these notions seem to be more intuitive. The term *transformation* refers to the creation process itself and the terms *source* and *result* refer to the input and output of a transformation.

Most of the existing research can be classified by their approach to provenance recording. One research direction focuses on computing provenance information when data is created, while the other computes provenance data when it is requested. Tan [Tan04] refers to these approaches as *lazy* and *eager*. Most of the eager approaches are based on annotations about source data items and transformations, and most of the lazy approaches rely on inversion or input tracing of transformations.

Buneman et al. distinguish in [BKT01] between *Why-* and *Where-provenance*. While Why-Provenance captures all source data items that contributed to the creation of a result data item, Where-provenance captures the concrete origin of a result. We use the terms *contributing source* and *original source* instead of Why- and Where-Provenance. In section 3 these concepts are discussed more in detail.

Several surveys on provenance have been published [BF05, SPG05a, SPG05b], but most of them cover only a specific type of provenance or separate different kinds of provenance according to their application domain. In [SPG05b] a taxonomy of provenance systems is introduced. While forming a valuable overview of existing approaches, this taxonomy fails in seeing provenance from a more abstract, conceptual point of view.

As mentioned before the diversity of data representation models and application domains has lead to a number of application or data model dependent provenance models and prototype implementations. While some publications present a provenance model for a specific application domain like visualization, GIS or interactive statistic systems, others address a specific data representation or transformation model.

A number of authors address provenance in the context of services and workflow management. The PReServ (Provenance Recording for Services) [GMM05, GJM$^+$06a, GJM$^+$06b] approach uses a central provenance management service. In [GMM05] this web service receives messages, called p-assertions, from the web services used for data transformation. The provenance data is stored by the provenance management service in a so-called provenance store. PReServ uses a common interface to enable different storage systems as a provenance store. The system relies on the user or service developer to modify existing services to support p-assertions. Simmhan et al. [SPGM06] expect services to post provenance information on a message board. A provenance service collects these messages and stores them in a provenance store. The myGrid system [SRG03] provides middleware for biological experiments represented as workflows. myGrid records all service invocations including parameters and data items. This log information can be used to derive the provenance of a data item produced by a workflow [GGS$^+$03, KTL$^+$03, ZWG$^+$04, ZGG$^+$03, ZGSB04].

Chimera [FVWZ02, Fos03] offers a Virtual Data Catalog for provenance information. A user registers transformations, data objects and derivations (an execution of a transformation) in the Chimera Virtual Data Catalog (VDC). The VDC is implemented as a relational database. VDL (Virtual Data Language) provides query and data definition facilities for the Chimera system. While the first prototype is limited to file system data objects and executable program transformations, the system is to be extended to support relational or object-oriented databases and SQL-like transformations. Chimera is a part of the GryPhyN project [AF00], a research project developing techniques for processing and managing large distributed data sets in data grids.

Cui et al. [CWW00, Cui02] focus on lazy computation of provenance for data warehouses. They study views in data warehouses and develop algorithms to generate queries for provenance tracing. These queries trace all source data tuples that lead to the creation of a tuple in a view. In [CW01] their approach is extended to general transformations that could not be expressed in SQL. While rather complete in the domain of transformations and views in data warehouses, their approach is limited to this domain and does not include other data models or user generated provenance. In the work of Fan and Poulovassilis [FP05], provenance data is recorded at schema level in the context of schema transformations in a data warehouse.

The topic of provenance for relational databases was first discussed in the context of visualization [WS97]. Here data transformations are represented as functions from one attribute domain to another. Provenance is traced by using inversions of these functions. Another approach for provenance management in a visualization environment is presented in [Gro04]. Groth et al. record user actions in an interactive visualization environment and present the whole user interaction as a DAG (directed acyclic graph). The user can navigate in this graph and jump back to previous states of the system.

In [BC88], the audit facility for the S-Language is presented. S is an interactive statistical environment. User actions are recorded in an audit file which offers the oportunity to undo user actions or investigate a user session. It is similar to the log files a database system uses for roll-back and crash recovery.

In [Wid05], Widom presents considerations for the Trio-system. Trio is a database system for handling uncertain data and provenance. The formal foundation is given in [BSHW06] and implementation details are presented in [ABS$^+$06]. Uncertain data has been intensely studied by the database community for more than two decades [BP82], but the combination of uncertainty and provenance introduce new challenges. Trio shows that provenance information can be used to solve some of the problems that arise from introducing uncertain data in a database system. While interesting because of the combination of provenance and uncertainty, the provenance computation of the Trio system based on earlier work for data warehouse views [CWW00, Cui02]. The provenance of the tuples in the database is stored in an auxiliary table. Trio records one level of provenance, meaning only the tuples from which a tuple was directly derived are stored in the provenance relation. A database-wide unique tuple ID is used to identify tuples. Halevy states in [HFM06] that the management of provenance in combination with uncertainty is needed for dealing with data spaces, a concept for next generation data management systems introduced in this publication.

Seltzer et al. [SMRH+05, LNH+05] created a prototype of a storage system that integrates provenance information. An extension of a Linux kernel and file system is used to manage provenance data. A new file system node type is used to store the provenance of a file. Unlike normal files, provenance information is never deleted and exists as long as the file system itself.

In the GIS research area, the importance of provenance for evaluating the quality of data items has been recognized early on. Most publications from this area focused on the development of metadata standards [HE97] which include provenance information. A well-designed metadata standard could provide a basis for provenance management system, but the proposed standards are limited to the GIS-domain and cannot be easily generalized. More important the metadata defined by these standards is meant to be provided by a user and may not be appropriate for automatic processing.

In [BCC06, BCCV06, BC05], data from various data sources is represented in a tree-structure. In this framework updates, insertions and deletes are copy, paste and insertion operations on these data trees. The authors present a query language that operates on data trees. Buneman et al. used this tree-representation of data for archiving [BKTT04]. They require unique keys for every data object. These keys can be used to associate different versions of a data item in different versions of a database (or data repository).

Provenance is related to data annotation. Annotation systems like DB-Notes [CTV05] and MONDRIAN [GKM05] enable a user to annotate a data item with an arbitrary number of notes. These notes are normally propagated when annotated data is transformed. Using annotations to represent information about a data item is a common approach in life sciences. The possibilities of using annotations to maintain provenance information were first discussed in [BKTT02, BS02]. This approach was also taken in [ZGG+03, MPL+06, CTV05]. The DB-Notes system introduced in [BCTV04, CTV05] enable a user to store annotations at the attribute level for data in a relational database. DB-Notes is based on a relational database and queries are represented in pSQL, an SQL-like language including constructs for managing annotations.

Data provenance is also related to temporal data management and versioning. Like in temporal data management, in provenance previous versions of a data item are queried and accessed. So provenance management systems may benefit from existing storage methods and query optimizations for temporal databases. Intelligent archiving techniques [BKTT04] need methods capable of identifying an object in different versions of a database or document. The identification methods used in this context may be also applicable to provenance management.

3 A categorization scheme for conceptual properties of data provenance

In this section we discuss data provenance from a conceptual point of view, extend the terminology introduced in section 1 and define a general categorization scheme for provenance management systems. We define several functionalities a provenance management system can provide and order these functionalities in a hierarchy of categories. The three main categories of our categorization scheme are *provenance model*, *query and manipulation functionality* and *storage model and recording strategy*. We present an overview figure for each main category (figures 1, 2 and 3). Categories and functionalities are represented by boxes and ellipses in these figures.

3.1 Provenance model

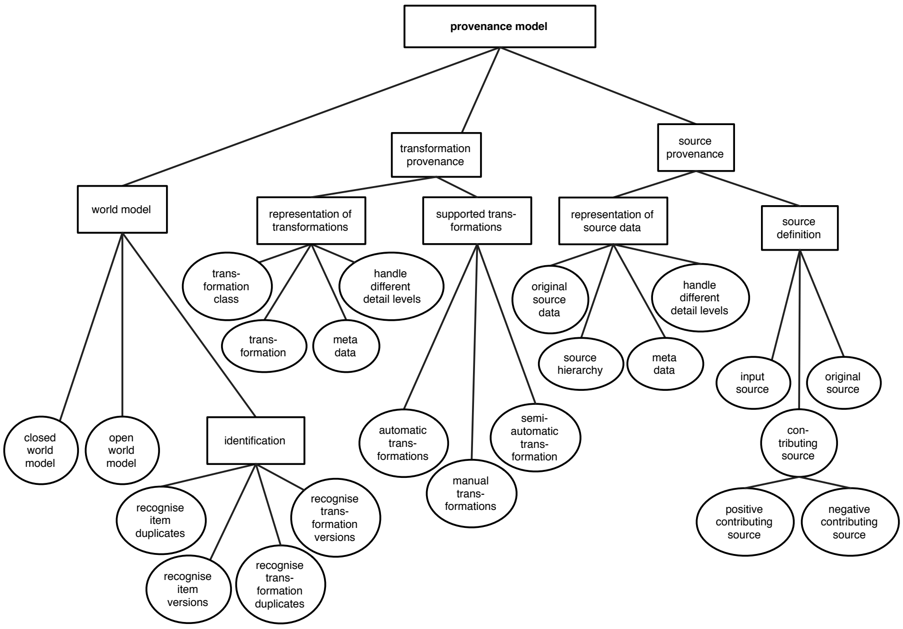
provenance model
transformation provenance
source provenance
world model
representation of transformations
supported transformations
representation of source data
source definition
trans-formation class
handle different detail levels
trans-formation
meta data
original source data
handle different detail levels
source hierarchy
meta data
input source
original source
closed world model
open world model
identification
automatic trans-formations
manual trans-formations
semi-automatic trans-formation
con-tributing source
positive contributing source
negative contributing source
recognise item duplicates
recognise trans-formation versions
recognise item versions
recognise trans-formation duplicates

Figure 1: Provenance model

The category *provenance model* describes the expressive power of the conceptual model used by a provenance management system to define the provenance of a data item. We define a number of functionalities and categorize a provenance system by means of the functionalities it supports. As stated before, the provenance of a data item can be divided into the two parts *transformation provenance* and *source provenance*. Source provenance is information about the data that was involved in the creation of a data item. There are three conceptual definitions for source provenance. These are *original source*, *contributing source* and *input source*. The input source of a data item includes all data items that were used in the creation of this data item. The *positive contributing source* of a data item includes data items that are essential for the creation of this data item. In formal terms the positive contributing source of a data item X is the union of all minimal subsets of the input source that, when used in the process that lead to the creation of X, would also lead to the creation of X. The original source of a data item consists of all data items which include data that is copied to the resulting data item.

For example, assume we manage the provenance of data in a relational database with two relations r_1 and r_2 and handle data items at tuple level of detail. When executing the SQL-query SELECT $r_1.name$ FROM r_1, r_2 WHERE $r_1.id = r_2.id$ against the a database including relations r_1 and r_2 the input source of a resulting tuple t includes all tuples in r_1 and r_2. The positive contributing source of t consists of all tuples t' from relation r_1 and t'' from relation r_2 with $t.name = t'.name$ and $t'.id = t''.id$. At last the original source of t includes all tuples t' from relation r_1 with $t.name = t'.name$.

The concepts original source and positive contributing source where first introduced in [BKT01] under names Why- and Where-provenance. Note that the following subset relationship holds:

input source $\supseteq$ positive contributing source $\supseteq$ original source

Some applications would benefit from information about data items that are not existent in the source, but would inhibit the creation of a resulting data item, if they were included in the source. This concept has been first introduced by Widom et. al. in [CWW00]. We use the term *negative contributing source* for this concept. In contrast to the concept positive contributing source, this definition of not straightforward. It seems feasible to include either all data items that would alone prohibit the creation of the result or include combinations of data items that would prohibit the creation of the result. In most data repositories the amount of data stored in the repository is only a very small fraction of the all possible data that could be stored in the repository. So in the general case, it is not possible to actually store the negative contributing source of a data item.

Additional to the considerations which kind of sources should be included in the source provenance, a provenance management system can record various information about each source data item. A source could be represented as the *original data*, *metadata* attached to the source, the *source hierarchy* structure or a combination of this representations.

A provenance system can either record source data items at one *level of detail* or be able to handle multiple levels of detail. For example, the source of a tuple data item in relational view could include all tuples from a relation r. If the provenance model handles multiple levels of detail the source could be represented as relation r instead of representing it by all tuples from r. Managing provenance information at different levels of detail is the more flexible approach and can result in smaller storage overhead of provenance information, but requires a more complex provenance model.

Transformation provenance is information about the transformations that were involved in the creation of a data item. To make a clear separation between a concrete execution of a process and the process itself we use the term *transformation class* for the first and *transformation* for the later. Foster et al. use the terms *transformation* and *invocation* for these concepts [Fos03]. In our concept a transformation is not limited to be an *automatic process*, but may be a *manual process* or a *semi-automatic process* with user interaction. The transformation provenance of a data item could include metadata like author of the transformation, the user who executed the transformation and the total execution time.

Examples for transformations are SQL statements used to create views in a database, the workflow descriptions of a workflow management system and executable files with command line parameters.

An important part of the provenance model is the *world model*, which could be either closed or open. In a *closed world model* the provenance management system controls transformations and data items. Contrary in an *open world model* the provenance management system has no or only limited control over the executed transformations and data items. Data items and transformations can be executed, manipulated, created or deleted without notification. From the view of the provenance management system, the world has a uncertain behavior which results in complex provenance recording or make it impossible to record exact provenance information. The closed world and open world model are extremes and there are many possible world models that are neither closed world nor open world models.

Besides the functionalities a provenance management system provides to handle source and transformation provenance, it should be able to recognize if two data items from two different data repositories represent the same real world object. For example the same data item could be stored in many databases or even in a database and as an XML-document. As real world object tend to change over time, it is important to have mechanisms for checking if two data items are different versions of one real world object. Identification is especially important, when updates to the data repositories are not controlled by the provenance management system. In this case the information about source data items recorded by the system might be incorrect, because these data items were changed or deleted by an update.

The data item equality needed for provenance management systems with open world models is a semantic equality, which has been studied in depth by the data integration community [Ken91, New88, PAGM96, BN05]. Semantic equality in general is not solvable for open world models, but there are several heuristical approaches to this problem (e.g. [Zia99]). The problem of identifying different versions of the same object also applies to archiving and is discussed in [BKTT02, BKTT04].

There are several possibilities to identify duplicates. A straightforward approach would be to check if the data item and the duplicate represent exactly the same information. We refer to this approach as value based duplicate identification. If data items have a key property, then another approach is to identify duplicates by their key property. For example, if data items are tuples in a relational database, two tuples could be defined to be duplicates if they have the same attribute values or if they have the same key attribute values. Using the primary key constraints of a relational database for identification could be problematic when no further restrictions are introduced, because the primary key uniqueness is restricted to one relation and primary keys can be changed by updates.

Many data models have an explicit or implicit hierarchical structure. This hierarchy in combination with a key property or value equivalance could be used to identify a data item. For example, if the provenance of tags in XML-documents is recorded, duplicates could be defined by the name of the tag and the position of the tag in the hierarchy of the document.

3.2 Query and manipulation funcionalities

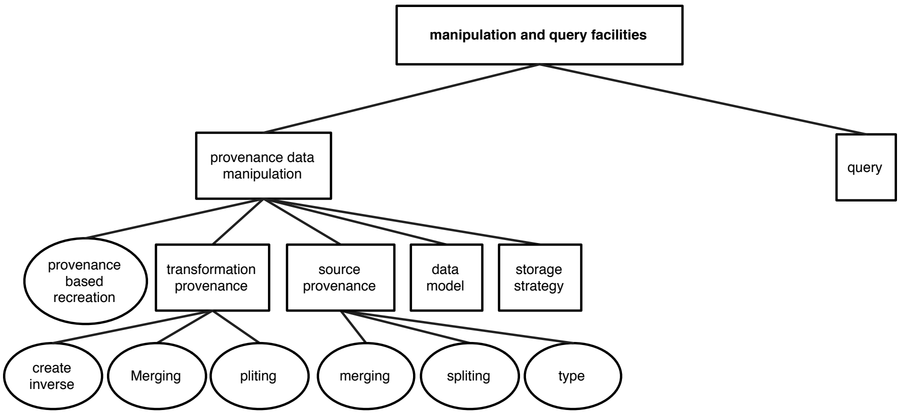
manipulation and query facilities
provenance data manipulation
query
provenance based recreation
transformation provenance
source provenance
data model
storage strategy
create inverse
Merging
pliting
merging
spliting
type

Figure 2: Query and manipulation funcionalities

To be useful for a real world application, a provenance management system should provide facilities to manipulate and query provenance information and data items. We do not discuss manipulation and querying of data items without integration of provenance information, because these query and manipulation facilities have been extensively studied and are well understood.

If a provenance management system handles transformations at various levels of detail, it should provenance mechanisms for *merging* multiple transformations into one transformation and *split* a complex transformation into a sequence or graph of simpler transformations. This functionality is similar to the composition of processes implemented by workflow management systems [Moh96]. Provenance data can be used to recreate result data items [Fos03], which cannot be accessed or are expensive to access, by executing the transformations that were used to create the result data item. If a provenance management system is able to compute the inverse of a transformation, then the inversion can be used to recreate source data items from result data items. For example Widom et al. [CWW00] compute queries for tracing the contributing source tuples for tuples in a materialized view based on the view definition statement.

Split and *merge* operations could also be applied to the data item dimension. Split divides a higher-level data item into its lower-level parts and merge combines lower-level data items into a higher-level data item. While the split operation is quite clear, the merge operation raises some questions. For example, what is the result of the merge operation on a subset of the lower-level data items that form a higher-level data item, and how can this result be distinguished from the result of a merge operation on the whole set. A provenance management system that records provenance information for different data models should provide facilities for converting the representation of a data item form one data model to another.

So far we here omit the aspect of provenance storage strategy. Provenance information may be attached to the physical representation of a data item or stored in a separated data repository. We discuss this topic in detail in the next subsection. At this point we are only interested in the fact that a provenance management system may support more than one storage strategy and might offer mechanisms for changing the storage strategy for data items.

Depending on the properties of the provenance model and world model it may be difficult or even impossible to implement the postulated manipulation operations.

3.3 Storage and recording

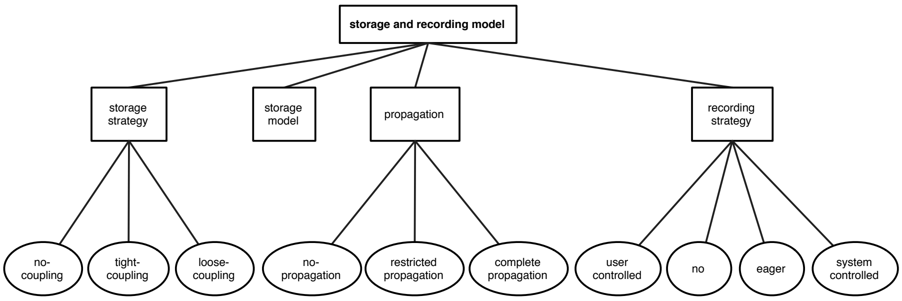

Figure 3: Storage and recording

The category storage and recording includes the approaches a provenance management system uses to store provenance information, to record provenance information and to propagate provenance information recorded for source data items.

Storage strategy describes the relationship between the provenance data and the data which is the target of provenance recording. There are three principal storage strategies: the *no-coupling*, the *tight-coupling* and the *loose-coupling* recording strategy. The no-coupling strategy stores provenance information in one or many provenance repositories. These repositories are dedicated to storing only provenance data. The second option, tight-coupling recording, stores provenance directly associated with the data for which provenance is recorded. The loose-coupling strategy uses a mixed storage scheme where provenance and data are stored in one storage system but logically separated.

Most approaches based on annotation use a tight-coupling or loose-coupling strategy by attaching provenance annotations to data items or storing annotations in the same data repository, but separated from the data items. Service based approaches in general record provenance for several data repositories in a distributed environment. These approaches normally deal with a very heterogeneous environment with limited control over the execution of processes and manipulation of data item, which make it difficult to record provenance information. Some control over this environment and especially over the provenance information can be gained by using a closed world model and a no-coupling storage strategy.

There are multiple storage models for storing provenance. In principle every data model could be used to store provenance information, but not every combination of storage model and storage strategy is reasonable.

If provenance is recorded for a transformation which uses source data items with attached provenance information, how is this information propagated to the result data items? The three possible answers to this question are *no-propagation*, *restricted propagation* and *complete propagation*. With no-propagation the provenance of source data items of a transformation is ignored when creating provenance data for result data items of the transformation. Contrary under complete propagation the result data items of a transformation inherit all provenance data from source data items according to the kind of source used in the provenance model. Under restricted propagation a result data item inherits a part of provenance from the source data items, e.g. data provenance that was created during the last n transformations.

The *provenance recording strategy* specifies when provenance data is recorded. We consider *user controlled recording*, *eager recording*, *no recording* and *system controlled recording*. Obviously, with user controlled recording the user decides when and for which data item he like to record provenance information. Eager recording records provenance simultaneously with every transformation. The no recording approach generates provenance at query time. Under system propagation recording data creation is controlled by the provenance management system. This system could use strategies like record the provenance data once a day or record the provenance after every n transformations. Eager recording and no recording are related to the eager and lazy provenance tracking approaches introduced in [Tan04].

4 Categorization of existing approaches according to our categorization scheme

In this section we categorize existing models according to the scheme presented in the last section. A categorization of all existing approaches is beyond the scope of this paper, so we limit the discussion to the publications that seem to be most important, innovative or influential. As regards we decided to use the following models: Provenance of views in a data warehouse environment, Chimera, Trio, PRoServ, DBnotes and the copy-paste-model introduced by Bunemann.

4.1 Provenance of views in a data warehouse

provenance model	
world model	closed world model in [CWW00]extended by openness in [CW01] for transformations
identification	no data item versions supported and no duplicate recognition
representation of transformations	concrete execution (relational algebra expression) with handling of different levels of detail
supported transformations	SQL view definitions. In [CW01] extended with general data warehouse transformations (e.g. ETL)
representation of source data	original source data: database tuples
source definition	contributing source for views. Only input source for some of the general transformations introduced in [CW01]
query and manipulation facilities	
provenance based recreation	no
transformation provenance manipulation	concept relies on computation of inverses
source provenance manipulation	no
data model manipulation	no
storage strategy manipulation	no
query	receive provenance
storage and recording model	
storage strategy	no storage
storage model	no explicit storage of provenance information
propagation	no-propagation
recording strategy	lazy: computed when queried

In [CWW00], Widom et al. study the provenance of materialized views in a data warehouse. They present algorithms to create inverses of a view definition for tracing the positive contributing source of tuples in the view, based on the view definition. Transformations are represented by the relational algebra view definition and are handled at different levels of detail (by defining provenance for the basic relational operators and for the concatenation of these operators thus allowing views based on views). The provenance data is computed lazily and is not stored for further use. In the original paper the closed world model of a data warehouse is used and extended in [CW01] to allow certain openness and new transformations not representable in relational algebra by including general data warehouse transformations.

4.2 Trio

provenance model	
world model	closed world model: a database that is controled by the system, but limited support for data inserted from a external datasource

identification	no data item versions supported and no duplicate recognition
representation of transformations	no representation
supported transformations	restricted SQL queries with extended semantics for uncertainty
representation of source data	original source data: database tuples (alternatives)
source definition	contributing source
query and manipulation facilities	
provenance based recreation	no
transformation provenance manipulation	no
source provenance manipulation	no
data model manipulation	no
storage strategy manipulation	no
query	receive provenance
storage and recording model	
storage strategy	loose coupling
storage model	provenance information is stored in an axiliary relation
propagation	no-propagation
recording strategy	eager: provenance computation at query time

Trio [BSHW06, SBHW06, ABS$^+$06] is a database system for handling uncertain data and provenance, called ULDB. A ULDB presents a set of possible database instances by attaching a probability to every possible alternative for a tuple in the database. The positive contributing source of tuple alternatives, which present all possibilities of a tuple, is recorded at query execution time. Trio uses a loose coupling storage strategy where source provenance is stored in an auxiliary relation. Every tuple alternative has a unique identifier consisting of a tuple identifier and an identifier for the alternative. If a tuple is not derived from other tuples alternatives in the database, but inserted from an external source, this tuple is called a base tuple and special identifiers are used to refer to the external source in the provenance of the tuple. The authors also present TriQL, a query language based on a subset of SQL (currently supporting union, join, selection and projection) extended by predicates for querying provenance and uncertainty. The restriction to this subset is based on the observation that the provenance produced by this operations is "well behaved". A ULDB with well-behaved provenance has the advantage, that all possible instances of the database are determined by the possible instances of the base tuples in the database.

4.3 Chimera

provenance model	
world model	closed world model: distributed datasets and transformations.
identification	identification of duplicates through identifiers for data items and provenance based duplicates. data item versions not jet supported
representation of transformations	transformations and transformation classes: represented by metadata, handles different levels of detail
supported transformations	automated, semi-automated, manual
representation of source data	meta data: hierarchical position, type of data item, limited handling of multiple detail levels
source definition	input source
query and manipulation facilities	
provenance based recreation	possible
transformation provenance manipulation	change level of detail
source provenance manipulation	no
data model manipulation	no
storage strategy manipulation	no
query	receive provenance
storage and recording model	
storage strategy	no-coupling: provenance and data item metadata is stored in the VDC (Virtual Data Catalog)

storage model	relational database VDC for storing data item and provenance metadata developed for Chimera
propagation	no-propagation
recording strategy	user-controlled or eager: in general relies on the user to register new data items/transformations, but can be automated, if a transformation execution and data item generation is controlled by the system

Chimera [FVWZ02, Fos03, AZV$^+$02] offers a Virtual Data Catalog (VDC) for storing provenance data and data item identification information. Various kinds of transformations and data items are supported by the system through a general metadata representation. Chimera differentiates between transformation and transformation classes and refers to these concepts as invocations and transformations. Transformations and data items are handled at different levels of detail. The user is responsible for registering transformations, transformation classes, data items (called data sets) and data item types in the Chimera Virtual Data Catalog. The VDC is implemented as a relational database. VDL (Virtual Data Language) provides query and data definition facilities for the Chimera system. While the first prototype was limited to simple file data items and executable program transformations, the system is to be extended to support relational or object-oriented databases and more transformation types. Chimera supports data item recreation and changing the transformations level of detail.

4.4 PRoServ

provenance model	
world model	closed: recording provenance for services communication via a provenance protocol
identification	no data item versions supported and no duplicate recognition
representation of transformations	concrete execution, meta data
supported transformations	communication between services
representation of source data	original data send by the services and meta data about internal service states
source definition	input source
query and manipulation facilities	
provenance based recreation	no
transformation provenance manipulation	no
source provenance manipulation	no
data model manipulation	no
storage strategy manipulation	no
query	receive provenance
storage and recording model	
storage strategy	no-coupling
storage model	every storage model possible through implementation of an provenance store interface
propagation	no-propagation
recording strategy	eager: recoding of provenance messages send by actors services

Groth et al. have developed a protocol (PReP: P-assertion recording protocol) for recoding the provenance of interactions between services in a service-oriented architecture (SOA). This protocol requires the communicating services to send so called p-assertions, which are recorded by a provenance store web service. p-assertions present either inputs and outputs send by the services using a input source definition or metadata provided by the services about their internal state. Depending on the internal state data provided by the services it could by possible to identify the original or positive contributing source. The p-assertions transmitted by the services are stored by a provenance store service. To be able to correlate the multiple p-assertions sent by the communicating services, an identifier is attached to each p-assertion.

PReServ [GMM05] is a web service implementation of the PReP protocol. The provenance store service provided by this implementation uses a modular architecture thus enabling multiple storage models for provenance data storage. Provenance information can be queried through the query API included in PReServ.

4.5 Copy-paste-model

provenance model	
world model	closed world model: a database that is controled by the system with references to other data repositories
identification	no data item versions supported and no duplicate recognition, but identification based on hierarchy position
representation of transformations	concrete transformation: source location, result location, transformation class, handles different levels of detail
supported transformations	semi-automatic: copy, insert, delete operations for hierarchy position
representation of source data	source schema: hierarchy position of source data, handles different levels of detail
source definition	original source
query and manipulation facilities	
provenance based recreation	not implemented
transformation provenance manipulation	change level of detail at provenance recording time
source provenance manipulation	no
data model manipulation	no
storage strategy manipulation	no
query	receive provenance
storage and recording model	
storage strategy	no-coupling or loose-coupling
storage model	transformations and source hierarchy positions are stored in provenance repository
propagation	no-propagation
recording strategy	eager: computation when information is stored in the database

In[BCC06, BCCV06, BC05], Buneman et al. developed a provenance model for curated databases. They assume that a data item can be located by a path description in the hierarchy of the data repository it is stored in and represent data items from various data model as locations in a tree-structure. In this framework transformations are copy, paste and insertion operations on these data trees. The tree-structure enables their model to deal with source data items at different levels of detail. The authors defined four concepts of provenance which are in fact combinations of two levels of detail for source and transformation provenance respectively. In two of these concepts the user can control the transformation level of detail by manually defining the start and end of a transformation. A first prototype implementation of the model records the provenance for one database controlled by the system. The User can insert, delete and copy data in the database according to the defined transformation classes.

4.6 DB-Notes

provenance model	
world model	closed world model: a database that is controlled by the system
identification	no data item versions supported and no duplicate recognition
representation of transformations	no representation
supported transformations	pSQL-Statements
representation of source data	meta data: propagated annotations, path-to-source: an identifier for the position of the data item in the database
source definition	original source or user-controlled by the propagation scheme
query and manipulation facilities	

provenance based recreation	no
transformation provenance manipulation	no
source provenance manipulation	no
data model manipulation	no
storage strategy manipulation	no
query	receive provenance
storage and recording model	
storage strategy	tight-coupling: annotations attached to attribut values of database tuples
storage model	annotations stored in additional attributes in a relational database
propagation	complete or user-controlled restricted propagation. The pSQL language developed for DB-Notes includes operators for controlling the propagation strategy
recording strategy	eager: provenance annotations are generated when data is queried

The DB-Notes prototype is an annotation management system for relational databases that allows a user to attach an arbitrary number of annotations to data items at the attribute value level of detail. Annotated databases are queried using pSQL, an extension of SQL which includes constructs for manipulating and querying annotations. The system itself generates provenance annotations. These annotations represent hierarchical positions (see section 2) of data items. The propagation of annotations is controlled by the user through the PROPAGATE clause of pSQL. When the default propagation scheme is chosen, the system propagates provenance annotations form the source data items to a result data item according to the original source definition. The authors show that the default propagation scheme may create different annotations for equivalent queries and introduce a default-all propagation scheme, which is invariant with respect to query rewriting. The current prototype implementation uses a tight-coupling storage model, where relations are extended with additional attributes for annotation storage.

5 Open Questions and Future Research Directions

Our categorization scheme includes functionalities not jet included in provenance management systems. For example developing a provenance management system for open world models is a challenging problem. Furthermore many of the manipulation facilities present in the scheme are not included in the existing approaches. A formal model designed with the insight gained in this article could be the basis of a provenance management system that handles not only various storage models, but also different types of source and transformation provenance. Source and transformation provenance are not completely independent and it would be interesting to investigate under which circumstances it is possible to convert one into the other and study how much redundancy is introduced by storing source and transformation provenance. It seems also reasonable to investigate which of the functionalities included in our categorization scheme exclude or imply each other.

Some of the problems faced when dealing with provenance are related to data integration problems. For example the concept of semantic identity needed recognize duplicates or versions in an open world model was studied by various data integration publications [Ken91, New88, PAGM96, BN05]. A provenance management system handling different kind of data items stored in distributed repositories needs to integrate this data to gain a unified view on the data. Data integration systems might benefit by including provenance management. For example provenance data could be used to identify duplicate pieces of data or could help a user to assess the quality of integrated data.

It would be interesting to apply concepts developed in the area of temporal database and versioning in to the provenance management of updateable data repositories.

6 Conclusions

We have presented a categorization scheme for different types of provenance and categorized existing approaches according to this scheme. The categorization scheme helps us to gain a systematic overview of the capabilities and limitations of these approaches. Most categories used in our scheme are based on concepts developed by other researchers, but we investigated new combinations of these concepts and extended some of these concepts with new aspects.

In future work we will investigate which of the functionalities included in our categorization scheme exclude or imply each other. Such an analysis would help us to gain a better understanding of provenance in general. This investigation could be extended to cover investigation of implementation problems and complexity analysis for different functionality combinations.

We will also define a formal language for the management of provenance data. This language should include generation, querying and manipulation of provenance data. Unlike existing approaches, this language should cover not only different data models, but also manage different types of provenance information. It will include language constructs for converting between different data models and kinds of provenance data. We plan to explore this languages computational complexity and implementation problems with the goal of creating a prototype implementation. Because of the complexity of the problem, the prototype will be limited to a specific kind of provenance and restricted manipulation options in the beginning.

References

[ABS$^+$06] Parag Agrawal, Omar Benjelloun, Anish Das Sarma, Chris Hayworth, Shubha Nabar, Tomoe Sugihara, and Jennifer Widom. An Introduction to ULDBs and the Trio System. *IEEE Data Engineering Bulletin*, 29(1):5–16, 2006.

[AF00] Paul Avery and Ian T. Foster. The GriPhyN Project: Towards Petascale Virtual Data Grids. *The 2000 NSF Information and Technology Research Program*, 2000.

[AZV$^+$02] James Annis, Yong Zhao, Jens Voeckler, Michael Wilde, Steve Kent, and Ian T. Foster. Applying Chimera virtual data concepts to cluster finding in the Sloan Sky Survey. In *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–14, Los Alamitos, CA, USA, 2002. IEEE Computer Society Press.

[BC88] Richard A. Becker and John M. Chambers. Auditing of data analyses. *Journal on Scientific and Statistical Computation*, pages 747–760, 1988.

[BC05] Peter Buneman and James Cheney. A Copy-and-Paste Model for Provenance in Curated Databases. 123:S123, 2005.

[BCC06] Peter Buneman, Adriane Chapman, and James Cheney. Provenance Management in Curated Databases. Technical Report EDIINFRR0769, The University of Edinburgh, June 2006.

[BCCV06] Peter Buneman, Adriane Chapman, James Cheney, and Stijn Vansummeren. A Provenance Model for Manually Curated Data. In *International Provenance and Annotation Workshop*. University of Edinburgh, 2006.

[BCTV04] Deepavali Bhagwat, Laura Chiticariu, Wang Chiew Tan, and Gaurav Vijayvargiya. An Annotation Management System for Relational Databases. In Mario A. Nascimento, M. Tamer Özsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and K. Bernhard Schiefer, editors, *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 900–911. Morgan Kaufmann, 2004.

[BF05] Rajendra Bose and James Frew. Lineage retrieval for scientific data processing: a survey. *ACM Comput. Surv.*, 37(1):1–28, 2005.

[BKT01] Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. Why and Where: A Characterization of Data Provenance. In *ICDT '01: Proceedings of the 8th International Conference on Database Theory*, pages 316–330, London, UK, 2001. Springer-Verlag.

[BKTT02] Peter Buneman, Sanjeev Khanna, Keishi Tajima, and Wang-Chiew Tan. Archiving scientific data. In *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 1–12, New York, NY, USA, 2002. ACM Press.

[BKTT04] Peter Buneman, Sanjeev Khanna, Keishi Tajima, and Wang-Chiew Tan. Archiving scientific data. *ACM Trans. Database Syst.*, 29(1):2–42, 2004.

[BN05] Jens Bleiholder and Felix Naumann. Declarative data fusion-syntax, semantics, and implementation. *Advances in Databases and Information Systems, Tallin, Estonia*, 2005.

[BP82] Billy P. Buckles and Frederick E. Petry. A Fuzzy Representation of Data for Relational Databases. *Fuzzy Sets and Systems*, 7:213–226, 1982.

[BS02] Peter Buneman and Mark Steedman. Annotation - the new medium of communication. *Extract of workshop*, 2002.

[BSHW06] Omar Benjelloun, Anish Das Sarma, Alon Y. Halevy, and Jennifer Widom. ULDBs: Databases with Uncertainty and Lineage. In *32nd International Conference on Very Large Databases*, Seoul, Korea, September 2006.

[CTV05] Laura Chiticariu, Wang-Chiew Tan, and Gaurav Vijayvargiya. DBNotes: a post-it system for relational databases based on provenance. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 942–944, New York, NY, USA, 2005. ACM Press.

[Cui02] Yingwei Cui. *Lineage tracing in data warehouses*. PhD thesis, 2002. Adviser-Jennifer Widom.

[CW01] Yingwei Cui and Jennifer Widom. Lineage Tracing for General Data Warehouse Transformations. In *Proceedings of the 27th International Conference on Very Large Data Bases(VLDB '01)*, pages 471–480, Orlando, September 2001. Morgan Kaufmann.

[CWW00] Yingwei Cui, Jennifer Widom, and Janet L. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Trans. Database Syst.*, 25(2):179–227, 2000.

[Fos03] Ian T. Foster. The virtual data grid: a new model and architecture for data-intensive collaboration. In *SSDBM'2003: Proceedings of the 15th international conference on Scientific and statistical database management*, pages 11–11, Washington, DC, USA, 2003. IEEE Computer Society.

[FP05] Hao Fan and Alexandra Poulovassilis. Using Schema Transformation Pathways for Data Lineage Tracing. In Mike Jackson, David Nelson, and Sue Stirk, editors, *Database: Enterprise, Skills and Innovation, 22nd British National Conference on Databases, BNCOD 22, Sunderland, UK, July 5-7, 2005, Proceedings*, volume 3567 of *Lecture Notes in Computer Science*, pages 133–144. Springer, 2005.

[FVWZ02] Ian T. Foster, Jens-S. Vöckler, Michael Wilde, and Yong Zhao. Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation. In *SSDBM '02: Proceedings of the 14th International Conference on Scientific and Statistical Database Management*, pages 37–46, Washington, DC, USA, 2002. IEEE Computer Society.

[GGS+03] Mark Greenwood, Carole Goble, Robert D. Stevens, Jun Zhao, Matthew Addis, Darren Marvin, Luc Moreau, and Tom Oinn. Provenance of e-Science Experiments - experience from Bioinformatics. In *Proceedings of the UK OST e-Science second All Hands Meeting 2003 (AHM'03)*, pages 223–226, Nottingham, UK, September 2003.

[GJM+06a] Paul Groth, Sheng Jiang, Simon Miles, Steve Munroe, Victor Tan, Sofia Tsasakou, and Luc Moreau. An Architecture for Provenance Systems — Executive Summary. Technical report, University of Southampton, February 2006.

[GJM+06b] Paul Groth, Sheng Jiang, Simon Miles, Steve Munroe, Victor Tan, Sofia Tsasakou, and Luc Moreau. D3.1.1: An Architecture for Provenance Systems. Technical report, University of Southampton, February 2006. Available from http://eprints.ecs.soton.ac.uk/12023/.

[GKM05] Floris Geerts, Anastasios Kementsietsidis, and Diego Milano. MONDRIAN: Annotating and querying databases through colors and blocks. Technical Report EDIIN-FRR0243, The University of Edinburgh, March 2005.

[GMM05] Paul Groth, Simon Miles, and Luc Moreau. PReServ: Provenance Recording for Services. In *Proceedings of the UK OST e-Science second All Hands Meeting 2005 (AHM'05)*, Nottingham,UK, September 2005.

[Gro04] Dennis P. Groth. Information Provenance and the Knowledge Rediscovery Problem. In *IV*, pages 345–351. IEEE Computer Society, 2004.

[HE97] Kathleen Hornsby and Max J. Egenhofer. Qualitative Representation of Change. In Stephen C. Hirtle and Andrew U. Frank, editors, *Spatial Information Theory: A Theoretical Basis for GIS, International Conference COSIT '97, Laurel Highlands, Pennsylvania, USA, October 15-18, 1997, Proceedings*, volume 1329 of *Lecture Notes in Computer Science*, pages 15–33. Springer, 1997.

[HFM06] Alon Halevy, Michael Franklin, and David Maier. Principles of dataspace systems. In *PODS '06: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–9, New York, NY, USA, 2006. ACM Press.

[Ken91] William Kent. The Breakdown of the Information Model in Multi-Database Systems. *SIGMOD Record*, 20(4):10–15, 1991.

[KTL+03] Ananth Krishna, Victor Tan, Richard Lawley, Simon Miles, and Luc Moreau. The myGrid Notification Service. In *Proceedings of the UK OST e-Science second All Hands Meeting 2003 (AHM'03)*, pages 475–482, Nottingham, UK, September 2003.

[LNH+05] Jonathan Ledlie, Chaki Ng, David A. Holland, Kiran-Kumar Muniswamy-Reddy, Uri Braun, and Margo Seltzer. Provenance-Aware Sensor Data Storage. In *ICDE Workshops*, page 1189, 2005.

[Moh96] Chilukuri K. Mohan. Tutorial: State of the Art in Workflow Management System Research and Products. *5th International Conference on Extending Database Technology, Avignon, France, March*, 1996.

[MPL+06] James D. Myers, Carmen M. Pancerella, Carina S. Lansing, Karen L. Schuchardt, Brett T. Didier, Naveen Ashish, and Carole A. Goble. Multi-scale Science: Supporting Emerging Practice with Semantically Derived Provenance, March 2006.

[New88] Howard B. Newcombe. *Handbook of Record Linkage*. Oxford University Press, 1988.

[PAGM96] Yannis Papakonstantinou, Serge Abiteboul, and Hector Garcia-Molina. Object fusion in mediator systems. *Proceedings of the 22th International Conference on Very Large Data Bases*, pages 413–424, 1996.

[SBHW06] Anish Das Sarma, Omar Benjelloun, Alon Y. Halevy, and Jennifer Widom. Working Models for Uncertain Data. In Ling Liu, Andreas Reuter, Kyu-Young Whang, and Jianjun Zhang, editors, *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3-8 April 2006, Atlanta, GA, USA*, page 7. IEEE Computer Society, 2006.

[SMRH$^+$05] Margo Seltzer, Kiran-Kumar Muniswamy-Reddy, David A. Holland, Uri Braun, and Jonathan Ledlie. Provenance-Aware Storage Systems. Technical report, Harvard University, 2005.

[SPG05a] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *SIGMOD Rec.*, 34(3):31–36, 2005.

[SPG05b] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A Survey of Data Provenance Techniques. Technical report, Computer Science Department, Indiana University, Bloomington, Bloomington IN 47405, 2005.

[SPGM06] Yogesh L. Simmhan, Beth Plale, Dennis Gannon, and Suresh Marru. A Framework for Collecting Provenance in Data-Centric Scientific Workflows. 2006.

[SRG03] Robert D. Stevens, Alan J. Robinson, and Carole A. Goble. myGrid: personalised bioinformatics on the information grid. *Bioinformatics*, 19(90001):302–304, 2003.

[Tan04] Wang-Chiew Tan. Research Problems in Data Provenance. *IEEE Data Engineering Bulletin*, 27(4):42–52, 2004.

[Wid05] Jennifer Widom. Trio: A System for Integrated Management of Data, Accuracy, and Lineage. In *CIDR*, pages 262–276, 2005.

[WS97] Allison Woodruff and Michael Stonebraker. Supporting Fine-grained Data Lineage in a Database Visualization Environment. In *ICDE '97: Proceedings of the Thirteenth International Conference on Data Engineering*, pages 91–102, Washington, DC, USA, 1997. IEEE Computer Society.

[ZGG$^+$03] Jun Zhao, Carole Goble, Mark Greenwood, Chris Wroe, and Robert Stevens. Annotating, linking and browsing provenance logs for e-Science, October 07 2003.

[ZGSB04] Jun Zhao, Carole A. Goble, Robert D. Stevens, and Sean Bechhofer. Semantically Linking and Browsing Provenance Logs for E-science. *First International Conference on Semantics of a Networked World*, pages 157–174, 2004.

[Zia99] Wojciech Ziarko. Discovery through rough set theory. *Communications of the ACM*, 42(11):54–57, 1999.

[ZWG$^+$04] Jun Zhao, Chris Wroe, Carole A. Goble, Robert Stevens, Dennis Quan, and R. Mark Greenwood. Using Semantic Web Technologies for Representing E-science Provenance. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *The Semantic Web - ISWC 2004: Third International Semantic Web Conference,Hiroshima, Japan, November 7-11, 2004. Proceedings*, volume 3298 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2004.