Citable by Design
A Model for Making Data in Dynamic Environments Citable

Stefan Stefan Pröll[1] and Andreas Rauber[1,2]
[1]*SBA Research, Vienna, Austria*
[2]*Technical University of Vienna, Austria*
sproell@sba-research.org, rauber@ifs.tuwien.ac.at

Keywords: Dynamic Data Citation, Relational Databases, SQL, Persistent Identifiers.

Abstract: Data forms the basis for research publications. But still the focus of researchers is a paper based publication, data is rather seen as a supplement that could be offered as a download, often without further comments. Yet validation, verification, reproduction and re-usage of existing knowledge can only be applied when the research data is accessible and identifiable. For this reason, precise data citation mechanisms are required, that allow reproducing experiments with exactly the same data basis. In this paper, we propose a model that enables to cite, identify and reference specific data sets within their dynamic environments. Our model allows the selection of subsets that support experiment verification and result re-utilisation in different contexts. The approach is based on assigning persistent identifiers to timestamped queries which are executed against timestamped and versioned databases. This facilitates transparent implementation and scalable means to ensure identical result sets being delivered upon re-invocation of the query.

1 INTRODUCTION

Scientific research has fully arrived in the digital age, where researchers have powerful infrastructures at their fingertips (Hey et al., 2009). Within the area of eScience, increasingly complex experiments are based on large data sets. Many scientists still have their primary focus on the paper based publication. In principle, these publications remained conceptually the same as they have been since decades. Despite the fact that it has never been so easy to publish not only the results in a written format, but also the underlying data that were the foundation of these results, little attention is attributed towards the research data. Many funding bodies, such as the FWF[1] in Austria or the European Union[2], but also governments and Journals e.g. Nature[3] demand or at least recommend the availability of data and other material, that is required for the re-execution of an experiment. So far data is often considered as a supplement or metadata to the publication, that has to be cited in its entirety. Thus although several approaches address the data citation problem, there are open issues specifically concerning the scalable and machine-readable citation of subsets of potentially dynamically changing and evolving and growing data sets. If data is deposited, it often is submitted in large, indivisible units and often offered as a download. Data only will be reused if it can be utilised within different scientific contexts. Hence a more flexible way of citing also specific subsets is required.

Data sets need to be identifiable in order to foster reuse, enable validation, re-production and re-execution of scientific experiments. We propose a model for citing subsets of large scale research data. In this paper our focus is specifically on relational database management systems (RDBMS), which allow to define precise subsets with the SQL language. We concentrate on the queries and their results, not on the large, indivisible data dumps as a basis for reference. Our model increases the scalability of data citation by assigning unique identifiers only to queries used for selecting the data used in subsequent experiments. Being based upon temporal database aspects and unambiguous result presentation, citing only the query persistently is sufficient for our model. It guarantees not only consistent result sets across time, but also consistent result lists.

[1] http://www.fwf.ac.at/en/downloads/pdf/free-research-needs-the-free-circulation-of-ideas.pdf

[2] http://ec.europa.eu/research/science-society/document_library/pdf_06/ recommendation-access-and-preservation- scientific-information_en.pdf

[3] http://www.nature.com/authors/policies/availability.html

The remainder of this paper is structured as follows: Section 2 provides an overview of current data citation practises and motivates the need for a new model for data citation. Section 3 introduces a model for citing data in dynamic environments. The model is described for relational databases and generalised for generic data sources. Section 4 concludes the paper and provides an outlook of our future work.

2 HOW DATA IS CITED TODAY

Publications increasingly contain references to data that was used or generated during the research that substantiates the work. However, research data sets are often treated as one entity, i.e. indivisible, static and referencable as one unit. In many cases, data is referenced bibliographically. As a minimum (Brase, 2009), the following metadata about a data set are required (Australian National Data Service, 2011): author, title, date, publisher, identifier and access information. The data itself is then often deposited at an institutional site and referenced by providing an URL. Obviously this mechanism is not suitable for sustainable data citation for several reasons. Uniform Resource Locators (URL)[4] have not been designed to be stable for the long term. As their name implies, URLs refer to a location, not the object itself. As a result, many URLs that served as data citation reference are not accessible any more. Either because the author of the data set left the institution and the Web page was taken down, or because the server moved and the location changed.

To overcome the problem of changing locations, the concept of persistent identifiers was introduced. Persistent identifiers (PIDs) provide unique identification of digital objects and reliable locations of Internet resources. PIDs require organisational effort for the management for the linking between the data and the identifier. Also, services for locating and accessing objects are necessary. The organisations providing these services are denoted Registration Authorities (RA). These RAs are responsible for the long term access, resolution and maintenance of the identifiers they issued for digital objects. There exist different solutions for the implementation of persistent identifiers. The authors of (Hans-Werner Hilse, 2006) provide an overview of the most common approaches. In (Bellini et al., 2008), six steps are identified for implementing a persistent identifier system:

1. Select the resource that needs persistent identification and define the granularity.
2. Decide which RA is trustworthy and suitable
3. Define resolution granularity and access rights
4. Assign a resource name register the object
5. Execute resolution service
6. Maintain the link between PID and the resource

Although persistent identifiers solve the problem with locations of digital objects, there are drawbacks for dynamic data. As stated in the enumeration above, the granularity of the identifiers can be adjusted to the requirements of the data set. Subsets require their own identification and metadata. Assigning persistent identifiers (PIDs) to data portions of finer granularity, i.e. database rows or even cells would require enormous numbers of unique identifiers and yield infeasible citations. PID approaches are suited very well for static data, which should only serve as reference point once it has been created. Using the identification and additional metadata is sufficient to search, identify, and retrieve data again. However, many settings require us to go beyond these limitations and introduce scalable and machine-actionable methods that can be used in dynamically changing, very large databases. Also, many data sets continue to grow and are updated as the data sets are used in experiments. In order to enable data citation in dynamic environments versioning support is required. Furthermore, different stakeholders may be interested in diverse portions of the data. Hence, clearly defined subsets of the data need to be identifiable and citable as well. These are some reasons why PIDs assigned to entire data sets or databases are not sufficient for several applications.

3 CITING DYNAMIC DATA

In many cases research data is not just static. It can change and evolve during the time, records can be updated or deleted. To understand which data actually was involved in an experiment and to reference that data, a new model is required. In order to be able to unambiguously and transparently cite subsets of data under such conditions, the following requirements need to be met:

1. Subsets of large data collections can be referenced
2. Dynamic data can be handled
3. Scalability is enabled
4. Implementation is transparent

The first requirement covers the reuse of data, which enables to perform new analysis on old data and therefore generate new knowledge. The second

[4] www.ietf.org/rfc/rfc1738.txt

requirement covers the capability of citing dynamically changing data. Data sources can potentially be huge in size. Citing individual attributes and cells would require enormous numbers of unique identifiers and yield infeasible citations. Hence the third requirement covers scalable solutions, feasible to deal with large data sources. The fourth requirement regards usability. Only if a solution is pragmatic and transparent, it will be accepted. The proposed requirements are valid for all kinds of research data formats. We demonstrate and motivate the model that we propose by uses relational databases for tackling these four requirements. Section 3.3 then introduces a generic model that can be used for other data such as flat files, streaming data or various other data formats.

3.1 Dynamic Data Citation Using Relational Databases

Research data is often stored in relational database management systems (RDBMS). The results that they deliver are the basis for further processing. We concentrate on the queries and their results, not on the large, indivisible data portions as a basis for reference. Our model increases the scalability of data citation by assigning unique identifiers only to the query itself. Furthermore our model increases the preservation awareness or readiness of research projects. Our model provides guidance on how to enhance the data model used for processing research data, in order to ensure it can be reliably cited and re-used in the future.

Relational database management systems (RDBMS) support many of our requirements off the shelf. These databases can be used to retrieve arbitrary subsets of data. Hence we concentrated on this database model for a first pilot study before discussing the general applicability. Our model is based on timestamped SELECT-Statements and versioned data. Queries can be used in order to persistently identify subsets of arbitrary complexity and size. The dynamic nature of research databases requires mechanisms that allow to trace and monitor all changes that occurred during time. Hence, temporal aspects have to be included in the model. This timing information needs to be stored on each UPDATE, INSERT or DELETE statement for the affected records, enabling to trace all changes that occurred. As relational database systems are set based, sorting is not an inherent criteria automatically. Therefore, we need to specify stable sorting criteria that are automatically applied to the subsets. Depending on the size of the data set, the schema and the complexity of the query, the retrieval of the result set can challenging. If these properties are met, citing only the query persistently is sufficient to meet our requirements. It guarantees not only consistent result sets across time, but also consistent result lists even in case of none or ambiguous result set sorting in the initial query, even in the case of migration to a different DBMS.

3.2 A Basic Model for Citing Data Sets in Relational Databases

In timestamped RDBMS, timestamps are provided for all records. This ensures that specific versions of data can be retrieved without having to stall the database tables for additional data. As records can change, they need to be versioned, i.e. all changes that affect the data need to be traceable. This entails that statements such as DELETE or UPDATE must not to destroy the data, but rather set markers that indicate that a record has been marked for deletion or that it as has been updated by a more recent version.

The construction of subsets of complex databases can be easily be achieved by issuing SELECT-Queries against the RDBMS. To enable the data citation facilities, the SQL-Query has also to be augmented with a timestamp. This timestamp maps the subset to a specific state of the data. As the records in the database can be altered individually, it needs to be ensured that the correct version that was valid at the query's timestamp is selected for inclusion in the subset. Hence the timestamp of the query can be used to retrieve arbitrary subsets of a specific version of the data.

There are several possibilities how this version information can be implemented (Snodgrass, 1999). The temporal timestamp contains the explicit date at which the data has been changed. Suitable timestamps are dates that are granular enough to capture the point in time that enables to differentiate between two versions of data. The actual chronon to be picked depends on the potential frequency of changes in data, which is not a trivial task(Jensen and Lomet, 2001). Thus granularity can range from days to milliseconds. Snodgrass et al. differentiate between valid time and transaction time (Jensen et al., 1993). Valid time refers to the period until the data was considered a true fact in the database. Transaction time refers to the time when the change occurred on the system, independent of its temporal meaning for the actual data. The valid time concept is a reference to the real world, the transaction time only refers to the system time, at which a change of data was manifested. Both concepts could be used for managing versions in our model. As we are interested in the state of the database at a given point in time, the transaction time concept is clearly better suited.

It is essential being able to identify all records uniquely. This property can also be handled by any RDBMS with the concept of primary keys. Hence our citable database schema requires each table to be equipped with a primary key. Primary keys are by definition unique, hence it allows to specify a unique sorting of the records to be included into the subset. To achieve this stable sorting, each query needs to specify a standard sorting order based on the primary keys.

These queries themselves need to be stored and augmented with a timestamp that reflects the time when the query was issued. The query's timestamp defines what versions of the records are included in the subset. A hash function over the SELECT-Query allows to identify queries that have already been issued against the system. Then a mechanism to identify the queries and the subsets they produced is required. In this case, PIDs become very useful, as a query that identifies a precise subset is static. If no changes of the records have occurred between two runs of the identical query, the same PID needs to be assigned to both runs of the query.

Figure 1 illustrates the interaction of the components of the framework. The database contains all records of a data set and maintains their versions. Queries are stored with a timestamp of their issuing in the Query Store. This ensures that a subset can be reproduced by knowing the query and the time of its execution. The citation is done by using PIDs. The PID Store enables to identify queries again and reuse the subsets created by the query.

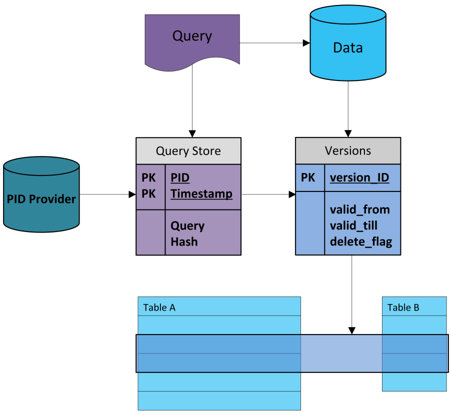

Figure 1: Data Citation Model for Relational Databases with PIDs and Versioned Data

It is easily possible to automate the creation of timestamps for data altering events as well as for queries. This allows to implement dynamic data citation transparently, i.e. no specific action is required on the user side: whenever a researcher selects subset of data for an experiment, the data is returned with a PID. This ensures that upon re-invoking the query, the PID is identified and an identical set is returned, even in identical order.

The model we introduced in this section describes how arbitrary subsets of data in potentially large relational databases can be created and retrieved at a later point of time.

3.3 Generalisation: Expanding the Model to Data Sources

The model that enables dynamic data citation introduced in Section 3.2 is not limited to relational databases. As nicely generalised from (Pröll and Rauber, 2013) in (Moore, 2013), the core concepts themselves can be mapped to other data models as well. The following requirements enable dynamic data citation on a generic level:

1. Uniquely identifiable data records
2. Time stamps of data
3. Versioned data, considering markings of deleted, altered or inserted data records
4. Query language for constructing subsets
5. Persistent query store that keeps queries and the timestamp of their issuing
6. An identification mechanism for queries, that enables access

The basic requirements are uniquely identifiable data records, that can be included in subsets of data. These records that form a subset need to be identifiable on an individual level. Furthermore, a versioning scheme must be available. These versions should reflect events such as insertion, updates or deletion. Hence no change on the data must be lost, regardless what data model is used. The versioning mechanism should include timestamps that allow to derive the set of valid records at a given point of time. For constructing subsets, the data source must provide a query language, which is powerful enough to select specified records based on precise criteria. To enable citation of subsets, it is sufficient to store the queries that led to the subsets and combine them with the timestamp. This timestamp provides the mapping between the query and the different versions of the records. This query is the key to the subset. Hence the query needs to be identifiable in order to retrieve the subset at a later point in time. With the requirements introduced, arbitrary data sources can be cited. The model based on these requirements allows to cite data that is evolving within the data source.

4 CONCLUSIONS AND FUTURE WORK

Digitally driven research is a rather young discipline that evolves fast. As a result the tools and the data are rarely developed with a focus of long term awareness. What matters most to researchers is fast results and prompt publications. If the data they produce today can be understood, interpreted or even accessed in the future is not addressed with the same attention. We want to change this paradigm and highlight the need for preservation aware research data.

Therefore we introduced a model for citing data in dynamically changing environments. We described how the model can be applied to relational database management systems and extended the framework to generic data sources. We identified requirements that enable data sources to provide citable subsets of data. Once the framework has been applied, most parts can be automated, hence transparent data citation capabilities are easy to offer. The easier and transparently this citation process can be implemented, the higher is the acceptance among the target audience and the designated community.

The concepts are currently considered to be addressed as part of a larger working group within the Research Data Alliance (RDA[5]). Our goal is to provide proofs of concept, mock-ups and prototype implementations, that can be tested and used by the community within the near future. A first prototype will be implemented by inserting the query re-writing and time-stamped storage of the query in the JDBC layer and testing it on several data sources used for scientific experiments. Future work will focus on other data formats that are widely used within research. This includes specialized file formats from various disciplines and areas.

Besides these criteria introduced in 3.2, there are additional considerations that have to be made. The requirements mentioned so far only consider internal properties of the system the data resides in. It is clear that external influences that can alter data, but are not recognised by the data storage system, need to be prevented. Furthermore, side effects that depend on the query system, the query language or specific properties of the data sets need to be removed in order to enable reproducibility. If the query language provides functions that are based on non-deterministic calculations, they have to be treated. This includes all sorts of randomised functions (e.g. a random number generator) or relative time specifications (e.g. CURDATE()). Such operations hinder the re-execution of a query for retrieving the exact same result, as they depend on external influences. How this issue can be mitigated will be part of our future work. Schema or format changes are a challenge that needs to be addressed.

[5] http://forum.rd-alliance.org

ACKNOWLEDGEMENTS

Part of this work was supported by the projects APARSEN and TIMBUS, partially funded by the EU under the FP7 contracts 269977 and 269940.

REFERENCES

Australian National Data Service (2011). Data Citation Awareness. `http://ands.org.au/guides/data-citation-awareness.pdf`.

Bellini, E., Cirinn, C., and Lunghi, M. (2008). Persistent identifiers distributed system for cultural heritage digital objects. In *iPRES 2008: The Fifth International Conference on Preservation of Digital Objects*.

Brase, J. (2009). DataCite - A Global Registration Agency for Research Data. In *COINFO 2009: Proceedings of the Fourth International Conference on Cooperation and Promotion of Information Resources in Science and Technology*, Washington, DC, USA. IEEE Computer Society.

Hans-Werner Hilse, J. K. (2006). *Implementing Persistent Identifiers: Overview of concepts, guidelines and recommendations*. Consortium of European Research Libraries, London.

Hey, T., Tansley, S., and Tolle, K., editors (2009). *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research.

Jensen, C., Soo, M., and Snodgrass, R. (1993). Unifying temporal data models via a conceptual model. *Information Systems*, 19:513–547.

Jensen, C. S. and Lomet, D. B. (2001). Transaction timestamping in (temporal) databases. In *Proceedings of the International Conference on Very Large Data Bases*, pages 441–450.

Moore, R. (2013). Workflow virtualization. In (Pröll and Rauber, 2013). Research Data Alliance - Launch and First Plenary March 18-20, 2013, Gothenburg, Sweden.

Pröll, S. and Rauber, A. (2013). BoF-Session on Data Citation. Research Data Alliance - Launch and First Plenary March 18-20, 2013, Gothenburg, Sweden.

Snodgrass, R. (1999). *Developing Time-Oriented Database Applications in SQL*. Morgan Kaufmann.