Evolution of Interrelated Data: Users, Problems, and Systems

Yannis Stavrakas, Theodore Dalamagas, and Timos Sellis
{yannis,dalamag,timos}@imis.athena-innovation.gr

Institute for the Management of Information Systems – R.C. “Athena”, Greece
http://www.imis.athena-innovation.gr

A user perspective

Artemis is a biologist conducting research on micro-RNAs. With her team, they try to determine the behavior patterns of micro-RNAs, the mechanisms of their involvement and the consequences of their presence in animals, hoping to ultimately draw conclusions about humans. To this end, they form hypotheses based on experiments and previous conclusions, and check to what extend those hypotheses can explain observations. Sometimes, new hypotheses or new evidence lead to re-evaluate previous conclusions. This in turn may lead to assess older experimental data under a new light. Moreover, experiments may be repeated using more advanced methods, and old data may be substituted by more precise, new data. This is similar to the research process in many scientific fields, where iterative refinements of the theory or complete paradigm shifts lead to closer approximations of the reality.

In Artemis’ domain of research, a main objective is to determine the relationships and interdependencies between entities such as micro-RNAs, proteins, diseases, etc. Those interdependencies are frequently revised as new theories and experimental data come forward. However in many occasions subverted beliefs prove to hold as much validity as their revisions. In this highly dynamic and evolving field, it is important for Artemis not only to record and access the community’s current perception of the involved entities and interdependencies, but also to review past states, and be able to follow their evolution backwards and forwards.

In particular, Artemis needs to deal with the following (families of) problems:

Poor management of evolving entities
The fundamental properties of an entity may evolve over time into something that constitutes an entity completely different than the original. However, there is only one name referring to all different entity perceptions, making it hard for Artemis to isolate the data that is really relevant. On the other hand, Artemis may take advantage of the single name identifier to access information relevant to the evolution history of the entity. Moreover, what is originally perceived as a single entity may prove to constitute a family of related but distinct entities, aggravating the problem of entity identification and access.

Evolution and Change not a first class citizen
Every so often Artemis needs to examine the steps that led to the current state. There are numerous reasons for doing that: (a) Observed inconsistencies with theory may trigger a revision of the steps taken so far. In this case Artemis would examine the sequence of previous modifications, re-evaluating the reasons for every change. (b) Previous versions of an entity may have pointed to information that was subsequently though as irrelevant, but which, in its current state, becomes again relevant. In this case, discredited links obtain new value. (c) Understanding how an entity reached its current state, why the changes took place, and who contributed, enhances the comprehension of the field. Moreover, Artemis would like to be able to access a specific transition an entity went through, in order to (a) document it (specify reasons for the change, related papers, who suggested the change, etc.), and (b) refer to it in an unambiguous way.

Infrastructure lacks support for evolution

In rapidly evolving fields it is common practice to use Web databanks for publishing results and for citing the work of others. The Web provides a direct means of publishing that avoids the delays associated with traditional publishing mediums, such as printed journals and conference proceedings. However, there is no guarantee that HTML links will continue to point to their original contents. It is possible that links to a remote databank will at some instance become broken or even worse, that their "contents" will be replaced, in which case they will point to something different than intended. Artemis would like a way to ensure that for some links, the "contents" will persist irrespectively of what change took place at the remote site pointed by the link.

Alternative names and implications they carry

An entity may be given different names by the research community. The reason for this name diversity is twofold. First, different research groups may assign different names to the same entity. Second, a name can change as the result of a deeper understanding of the entity. Therefore, alternative names may indicate (a) differences in the perception of the entity, and (b) focus on different aspects of the entity by the respective research groups. When Artemis is looking for information about an entity, she has to be aware of its alternative names, perform a number of distinct searches and integrate the results. The background implications associated with each name are very useful, since they may help Artemis prioritize the search results and focus her attention only to some of them.

A system perspective

Unfortunately, current Information Systems do not support such tasks adequately. Research work in temporal databases, version management, schema evolution, data provenance, and data preservation has provided elements of an "ideal" solution, but what is lacking, in our view, is a focus on the notion of evolution per se. A system that would view data and schema evolution as a first class citizen would allow not only recording and reviewing past states of information entities, but would also allow:

- Treating changes as entities: annotating them with data according to the type of the change, and being able to refer to a change like any other entity.
- Exploring and reasoning about a specific change: retrieving what was the change about, what evidence provoked the change, why it took place, who suggested it, who performed it, and so on.
- Expressing complex queries on changes: finding all modifications possibly provoked by the same cause, retrieving changes involving similar alterations, finding what other entities a modification affected, and more.

Moreover, such a system should take into account the following desiderata:

- Smooth integration with similar systems: In most occasions, there are many implicit and explicit links between different databanks of rapidly evolving fields. In the case of biology, Artemis often copies data from remote databanks to her local databank. This implicit link is lost as the two copies follow different evolution routes. Moreover, she often includes in her databank references to data in other databanks. Those references may break or point to wrong data in case the remote databanks update their contents. An "ideal" system would recognize such links and would allow Artemis to follow them through to remote sites, regardless of changes to the local or the remote databanks.
- Working as a supplement to existing systems: In our view it is not practical to provide a solution ignoring the current practice of user groups. A lot of data published through the Web lay in relational databases, in XML storage, even in structured text files. Support for data evolution

should come as an additional layer on existing systems, not interfering with their normal operation. This additional layer will interface with the user, and will keep additional information in its privately managed repository.

Building a system like the above requires answers to a number of research problems:

- What a conceptual model accommodating changes as first-class citizens looks like?
- What would be a suitable logical data model? The logical model will represent changes as they happen, keeping a record of previous states. What is more, the data model should allow to associate changes to entities that document the change. A lot of work exists on models that support evolution at schema and data levels [1,2,3,4,5]; however their focus was not to promote change at the information entity level.
- What are the basic change operations? If the basic change operations are too low-level, it may be difficult to recognize the user intent. For example, it may be hard to realize that the operations "add" and "remove" are used instead of "move".
- How change operations affect the data model so that changes are represented? The way changes are recorded in the data model may dictate the kind of queries that can be answered effectively.
- How to form more complex, user-defined operations? In many cases, users will perform specific sequences of basic operations to achieve well defined results. Do operations need to exhibit special properties to facilitate change representation and query evaluation?
- How to express queries that intuitively combine all the available data in meaningful ways? A query language should not only manage the structural information of the data, but also exploit (a) the intra-relationships of schema and data as they evolve over time, and (b) the inter-relationships of data that exist in different databanks.
- How to support persistent links? A persistent link is a method of reference which guarantees that the link leads always to the same object snapshot, regardless of changes that may have occurred to the object.

In conclusion, our objective is to build a system that supports change management on top of evolving interrelated databanks, in such a way that it is always possible to step back and examine why changes took place, and how changes relate to each other.

Ongoing work

Complementarily to the aforementioned directions, in IMIS we have been investigating web search methods focusing on personalization and support for research communities.

Web search engines are widely used for searching information on the Web. Their increased popularity is due to the simple and intuitive keyword search model, supported by fast text retrieval techniques that provide accurate results. However, there are use cases where the information need is complex. Consider a researcher that needs to set up her research agenda and generate innovative ideas. She often has the "big picture" for her search plan to start with, that is an abstraction that actually summarizes and classifies thoughts, ideas and concepts relevant to the plan. Based on this initial abstraction, she (a) gathers information from several data sources, (b) generates hypothesis, (c) refines their abstractions and their search plan, and (d) disseminates her results. Such a creativity cycle actually enables discovery and innovation.

New search models and techniques are necessary to support creativity and innovation [6]. The key objectives are:

- *Support creativity cycles for discovery and innovation.* In such cycle, users will be able to (a) define search plans for Web search using abstractions, (b) gather relevant information, (c) refine their abstractions and their search plans.
- Provide intelligent search services to support *metasearch orchestration* in several data sources/search engines. User queries are propagated to several search engines. Relevant resources are retrieved, merged, ranked and presented to user. Depending on the type of resources, certain search engines will be favored against other engines in order to answer the query.
- Improve the quality of the retrieved resources, by taking into consideration semantic information present in the abstractions.
- *Tailor the presenting result lists to user personal needs*, considering user search befaviour in the past.
- Provide *effective presentation and visualization capabilities* for the lists of retrieved resources that will guide users during their search and exploration by providing *compact visual cues* about resources attributes and various resource aspects, and supporting *rapid incremental and reversible exploration of retrieved resources*.

We have designed and started implementing a creativity support application based on *Freemind*, a popular open-source, *mind mapping* tool (see http://www.dblab.ece.ntua.gr/~dalamag/mm/ for an overview). Our application supports user creativity cycles, giving tools to (a) structure her topics of interest into mind mapping representations, (b) wrap data sources of her interest using a scraping wizard, and (c) search for papers or general web resources in these data sources, based on the mind mapping representations, in order to refine her representations.

References

[1] Hyun J. Moon, Carlo A. Curino, Alin Deutsch, Chien-Yi Hou, Carlo Zaniolo. Managing and Querying Transaction-time Databases under Schema Evolution. VLDB 2008.

[2] Amélie Marian, Serge Abiteboul, Laurent Mignet. Change-Centric Management of Versions in an XML Warehouse. VLDB 2001.

[3] Peter Buneman, Sanjeev Khanna, Keishi Tajima, Wang-Chiew Tan. Archiving Scientific Data. Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD), 2002.

[4] Chawathe, S. and Abiteboul, S. and Widom, J. Managing historical semistructured data. TAPOS 24(4), 1999.

[5] Shu-yao Chien, Vassilis J. Tsotras, Carlo Zaniolo. Efficient Schemes for Managing Multiversion XML Documents. VLDB Journal 11:332-353, 2002.

[6] Ben Shneiderman. Creativity support tools: accelerating discovery and innovation. Communications of the ACM, 50(12):20{32, 2007.