Provenance in Web Feed Mash-Up Systems

Watsawee Sansrimahachai, School of Science and Technology, University of the Thai Chamber of Commerce, Bangkok, Thailand

ABSTRACT

The recent emergence of web 2.0 technologies and rich internet applications is driving the development of a new class of applications that combines data from diverse sources which we refer to as “mash-ups.” One of the most popular mash-ups comes in the form of web feed mash-ups relying on syndication technologies such as RSS and Atom. This kind of mash-ups aggregates web feeds derived from multiple news websites or blogs and then timely presents them in a single interface. In such systems, it is difficult to know exactly how feed results in data mash-ups are generated. In particular, it is difficult for users to make determinations about whether information is trusted. Therefore, it is necessary that web feed mash-ups have to support a mechanism that is capable of recording and querying provenance information - the information about the process that led to result data. In this paper, the author proposes a provenance tracking solution that enables provenance functionality to be facilitated in web feed mash-ups. He demonstrates how the provenance of feed mash-up results to be determined by means of a provenance query algorithm. To tackle the storage problem resulting from the persistence of intermediate web feeds, a novel storage optimization method is introduced. Finally, the author evaluates his provenance solution in terms of storage consumption for provenance collection, demonstrating significant reductions in storage size and achieving reasonable storage overheads.

KEYWORDS
Data Provenance, Mash-up, Provenance Tracking, Storage Optimization Method, Web Feed

INTRODUCTION

A new breed of applications and services that integrates data from several sources which we refer to as mash-ups has grown significantly all across the Internet. Mash-ups are web-based applications that combine data and content from more than one source into an aggregated service presented in a single display. Several types of mash-ups have been developed such as mapping and image mash-ups. One of the most popular mash-ups comes in the form of web feed mash-ups (news mash-ups) relying on syndication technologies (e.g. RSS (RSS Advisory Board, 2016) and Atom (IETF, 2016)). This kind of mash-ups aggregates feeds derived from multiple news websites and then presents them over the web. Visiting many separate websites can take a lot of time to find out if content has been updated. With web feed mash-ups, updated information from many sites can be consolidated into a single view, reducing the time and effort needed to regularly check websites for updates. Furthermore, as information is combined from multiple sites, feed mash-ups are utilized as tools that create personalized and unique information catering to the user’s particular interests.

Although using web feeds in mash-ups offers several advantages, there are still some limitations in existing applications. According to the nature of this kind of mash-ups, web feeds aggregated generally come from several diverse sources (e.g. news website, blogs and web services). Furthermore, they are usually executed by a number of transformations or feed processing units in real-time. However,
the description of the source of data that are aggregated in mash-ups is not usually apparent. The
details pertaining to the feed items and the processing operations involved in the production of feed
mash-up results are not often provided as well. Therefore, it is difficult to know exactly how feed
results in data mash-ups are generated. In particular, it is difficult for users to make determinations
about whether information is trusted or not, considering that source feeds have to pass through several
transformations before the feed results are presented.

As a result, it is crucial to ascertain automatically whether or not to trust each individual feed
mash-up result by examining the process that created, aggregated, and delivered it. We derive a strong
requirement for precisely tracing the sFource feeds that caused a given result feed, which we refer
to as “provenance tracking in web feed mash-ups.” The existence of such functionality would allow
feed mash-up results to be verified and validated, so that users can have confidence and trust in the
results provided by web feed mash-ups.

To address provenance tracking in web feed mash-ups, we propose a novel provenance mechanism
that supports dynamic provenance collection and provenance query in web feed mash-ups. We introduce
a provenance architecture for web feed mash-ups extending the PASOA provenance architecture
(Groth et al., 2006). This provenance architecture applies the service-orientated architecture (SOA)
as a core principle. To allow provenance information to be recorded and interchanged with other
systems, PROV (Gil & Miles, 2016) - W3C data model for provenance interchange on the web - is
utilized as the provenance model for our provenance solution. The use of PROV enables an implement
of our provenance architecture to dynamically record provenance assertions - assertions pertaining
to provenance recorded by feed processing units during the execution of data mash-ups. Therefore,
dynamic provenance tracking in web feed mash-ups can be achieved.

This paper will make the following key contributions:

- It presents a novel provenance tracking mechanism which can precisely express relationships for
every feed mash-up result in a web feed mash-up system.
- It introduces a provenance architecture for feed mash-ups that describes the structure of the
provenance system, system components and interactions between components.
- It proposes a provenance query algorithm for feed mash-ups which utilizes a graph traversal
 technique to obtain the provenance of a particular feed mash-up result.
- It introduces a provenance storage optimization method for web feed mash-ups
- It presents the performance characteristics of our provenance solution in terms of the storage
consumption for provenance collection.

The rest of this paper is organized as follows. First, a review of previous work relevant to
provenance tracking mechanism in the area of the web of data is detailed. Second, a provenance
tracking mechanism for web feed mash-ups is presented. Third, the details of a provenance query
algorithm are described. After that, a provenance storage optimization method and a novel replay
execution algorithm are presented. This is followed by the presentation of the experimental evaluation
in terms of the storage consumption for provenance collection. Finally, conclusions are drawn.

RELATED WORK

Much of the research into provenance has come in the context of database systems. Tan (2007)
classifies research studies on data provenance (sometimes called lineage) into two distinct approaches:
in the lazy approach, provenance is generated on demand, by means of a query, only when requested
(Woodruff & Stonebraker, 1997; Cui & Widom, 2000, 2003) whereas in the eager approach information
is propagated at runtime (Buneman et al., 2002; Chiticariu et al., 2005; Green et al., 2007). We now
discuss them in turn.
A Location-Aware Access Control Model for Mobile Workflow Systems
www.igi-global.com/article/location-aware-access-control-model/4030?camid=4v1a

Examining Design Pattern Strategies as a Means to Achieve Social Presence in the Online Classroom
www.igi-global.com/chapter/examining-design-pattern-strategies-as-a-means-to-achieve-social-presence-in-the-online-classroom/137412?camid=4v1a