2012

A Perspective on Resource Synchronization

Herbert Van de Sompel

Robert Sanderson

Martin Klein

Michael L. Nelson
Old Dominion University

Bernhard Haslhofer

See next page for additional authors

Repository Citation

Van de Sompel, Herbert; Sanderson, Robert; Klein, Martin; Nelson, Michael L.; Haslhofer, Bernhard; Warner, Simeon; and Lagoze, Carl, "A Perspective on Resource Synchronization" (2012). *Computer Science Faculty Publications*. 135.
https://digitalcommons.odu.edu/computerscience_fac_pubs/135

Original Publication Citation

Van de Sompel, H., Sanderson, R., Klein, M., Nelson, M. L., Haslhofer, B., Warner, S., & Lagoze, C. (2012). A perspective on resource synchronization. *D-Lib Magazine, 18*(9/10), 3 (1-7). doi:10.1045/september2012-vandesompel

Search D-Lib: Go!

D-Lib® Magazine

The Magazine of Digital Library Research

HOME | ABOUT D-LIB | CURRENT ISSUE | ARCHIVE | INDEXES | CALENDAR | AUTHOR GUIDELINES | SUBSCRIBE | CONTACT D-LIB

D-Lib Magazine

September/October 2012
Volume 18, Number 9/10
Table of Contents

A Perspective on Resource Synchronization

Herbert Van de Sompel, Robert Sanderson, Martin Klein
Los Alamos National Laboratory
{herbertv, rsanderson, mklein}@lanl.gov

Michael L. Nelson
Old Dominion University
mln@cs.odu.edu

Bernhard Haslhofer, Simeon Warner
Cornell University
{bernhard.haslhofer, simeon.warner}@cornell.edu

Carl Lagoze
University of Michigan
clagoze@umich.edu

doi:10.1045/september2012-vandesompel

Printer-friendly Version

Abstract

Web applications frequently leverage resources made available by remote web servers. As resources are created, updated, deleted, or moved, these applications face challenges to remain in lockstep with changes on the server. Several approaches exist to help meet this challenge for use cases where "good enough" synchronization is acceptable. But when strict resource coverage or low synchronization latency is required, commonly accepted Web-based solutions remain illusive. This paper provides a perspective on the resource synchronization problem that results from inspiration gained from prior work, and initial insights resulting from the recently launched NISO/OAI ResourceSync effort.

1 Introduction

The Web is highly dynamic [2,4,9], with resources continuously being created, updated, deleted, and moved. Web applications that leverage third party resources face the challenge of keeping in step with this rate of change. Many such applications are not concerned with accurate coverage of a server's resources or consider delays in reflecting changes acceptable. In these cases, alignment with the dynamics of a remote server is commonly achieved by optimizing web crawling and resource discovery mechanisms, for example through scheduling crawls based on change frequency prediction. However, there are significant use cases that require more real-time and accurate synchronization. In many cases, this need is addressed through ad-hoc technical approaches implemented within a small group of collaborating systems. Proposals for more generic, Web-scale, synchronization approaches have been suggested but have not been widely adopted. Initially motivated by the need to synchronize resources for applications in the realm of cultural heritage and research communication, the National Information Standardization

Organization (NISO) and the Open Archives Initiative (OAI) have recently launched the ResourceSync project that aims at designing an approach for resource synchronization that is aligned with the Web Architecture [5] and that is targeted at the needs of different communities. Types of use cases that are considered within the scope of the effort are exemplified by, but not limited to, the following:

- The arXiv.org collection of scientific articles is mirrored to a number of sites on a daily basis. The current mirroring approach uses a homegrown notification and audit mechanism that can only be used to replicate content to trusted associated repositories that share the same storage structure. An open approach that allows any third party to mirror the content is desired.
- Applications based on Linked Data integrate resources from various datasets, with resources likely changing at a different pace. The BBC Linked Data applications that integrate data from, among others, Last.FM, DBpedia, MusicBrainz, and GeoNames serve as examples. The accuracy of services based on such an integrated resource collection depends on the contributing resources being up-to-date.
- The Europeana project provides a central portal based on cultural heritage collections of institutions throughout Europe. While the effort uses Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [7] to transfer metadata about cultural artifacts from these institutions to the portal, a need is anticipated for a mechanism to also transfer actual content (e.g., images, videos, audio) and to keep such content permanently in sync.

The ResourceSync effort recognizes the challenge in devising an approach that can be applied across a variety of use cases that entail different types of resources, different types of changes, and differing requirements regarding the coverage and speed of synchronization. This effort is not the first to step into this problem domain. However, given the increased usage and convergence of content management systems, the time may be ripe for a concerted effort to add a component to the web infrastructure that adequately addresses resource synchronization needs. The explorations of the ResourceSync effort have only recently started, and this paper provides an insight into early thinking about the problem.

2 Existing Approaches to Resource Synchronization

Resource synchronization is not a new problem. It has been acknowledged as an issue by, among others, Tim Berners-Lee [1], Umbrich, *et al*. [11], and the related W3C Dataset Dynamics activity.

With web resources frequently changing, a common approach to determine whether a resource was updated or deleted consists of recurrently issuing an HTTP HEAD or conditional GET against its URI and leveraging the Last-Modified and ETag response headers. When large resource collections are involved, or when synchronization requirements regarding coverage and speed are strict, this straightforward approach becomes problematic. Moreover, Clausen [3], for example, has shown that the approach is not reliable because ETag headers are missing in 40% of responses. Umbrich, *et al*. [11] further found that almost 68% of documents returned neither the ETag nor the Last-Modified response headers. In addition, such requests do not help for the discovery of newly minted resources. However, other techniques such as observing Sitemaps can be successful for this purpose, even for large collections. Sitemaps are used to support resource discovery by crawlers, and communicate an inventory of resources that minimally conveys their URI and many times also their last modification date. It is at the content creator's discretion to generate and update Sitemaps, which may result in a delay between a resource changing and the ability to discover such change.

In previous work we have investigated the use of OAI-PMH for the transfer of digital assets [12]. The proposed approach allowed the incremental collection of updated, but also newly created, resources and their metadata. However, this application of OAI-PMH has never received broad adoption, despite the fact that the protocol is widely deployed. The Apache module "mod_oai" [8] followed this OAI-PMH approach for digital asset transfer and was developed to allow crawlers to easily discover and obtain new and changed web resources exposed by the Apache server. The approach turned out too complex and too difficult for servers to support, and the more lightweight Sitemap approach promoted by Google became widely adopted.

Determining whether a resource has moved from one URI to another remains a challenge for which, so far, heuristics-laden solutions such as discovery by means of the resource's lexical signature are used [6]. Such approaches turn link maintenance for an application's collection into a complex endeavor.

3 Problem Characterization

Informed by the significant body of prior work related to resource synchronization and by ideas resulting from initial discussions between the ResourceSync project partners, an initial perspective on the problem domain has emerged that is described here.

3.1 Resource Synchronization Basics

At the core of the resource synchronization problem is the need for one or more **Destination** servers to remain synchronized with (some of the) resources made available by a **Source** server. Three distinct needs are recognized:

1. **baseline synchronization:** An approach to allow a Destination that wants to start synchronizing with a Source to perform an initial synchronization operation.
2. **incremental synchronization:** An approach to allow a Destination to remain up-to-date regarding changes at the Source.
3. **audit:** An approach to allow a Destination to determine if it is in sync with a Source.

With regard to baseline synchronization, a distinction must be observed between:

1.1. **inventory:** A way to express which resources are available at the Source, minimally conveying each resource's URI but maybe also transmitting auxiliary information such as most recent update time, fixity information, etc.

1.2. **dump:** A way to make representations for all resources of the Source available in a bulk manner that does not require dereferencing each resource's URI.

With regard to incremental synchronization, a distinction must be observed between:

2.1. **change communication:** An approach to allow a Destination to understand that a resource has changed at the Source; and what the type of change is (e.g., update, delete).

2.2. **resource transfer:** An approach to allow a Destination to update its holdings to reflect the change the resource underwent at the Source.

Deletions and moves of resources are obvious indications of the need for this distinction, as they don't require resource transfer. But considering change communication and resource transfer separately also allows synchronization strategies in which a Destination catches up with changes at its own pace rather than in lockstep with changes as they occur at the Source.

When it comes to the **types of events** that should be considered for change communications, **create**, **update**, and **delete** for an individual resource immediately come to mind. But also **move** and **copy**, which involve two resources, can be considered of importance although they raise trust issues, for example, when these resources are in different domains. For each of those events, change messages can identify resources by their URIs. Distinct representations could be further specified by means of protocol parameters and their respective values. Support for events that pertain to a range of resources may be considered, for example, moving all resources that reside in a certain URI path of a Source to another path. But this level of expressiveness may come at the expense of increased complexity as it would require a uniform URI *wildcarding* approach (e.g., regular expressions). It seems logical that all change messages should include the type of the change event, the event time, and the URI(s) of the resource(s) involved. But additional information could be considered, such as an indication of the significance of a change, fixity information, etc.

Selective synchronization refers to a Destination's need to only remain in sync with a subset of resources made available by a Source, or to only be aware of certain types of changes. The notion of a **channel** as a conduit for messages about a subset of a Source's changes follows quite naturally. Channels that have a reach across several Sources are also appealing. An example would be a channel that reflects changes pertaining to all digitized manuscript collections of all Europeana partners. While a Source can easily define channels according to criteria that make sense for its resource collection, Destinations may have different requirements, which raises the question of whether and how parties other than the Source can define channels. Approaches that have been used regarding this commonly rely on registration of queries [10] but these raise scalability concerns.

3.2 Interactions

The choice between a **push** or **pull** approach to handle change communication and resource transfer is an important design consideration that will be further explored in subsequent work. In a pull approach, a Destination would initiate a synchronization request to the Source, whereas in a push approach a Source would broadcast synchronization information to select Destinations.

Regarding resource transfer, a rather fundamental choice presents itself between solutions that rely on communicating **resource changes only** versus those in which the **entire changed resource** is transferred. The former is very likely to introduce optimizations regarding payload, but requires changes and re-assembly instructions to be expressed in ways that are specific to media or content types. Devising a generic changes-only approach (e.g., conveying byte-level deltas) seems out of reach, especially considering that, in a Web context, Source and Destination may very well store content differently. For example, a Destination may store a derivative of a Source's resource (e.g., a lower resolution image in another media format) but still have the need to update this derivative as the original resource changes. Or, Source and Destination may store RDF graphs differently. In contrast, an approach based on the transfer of entire changed resources generically applies across resource types but will likely be more costly in terms of data transfer size.

Another aspect pertaining to resource transfer deserves attention. In a Web-centric synchronization framework, it is fair to assume that changed resources would be obtained by dereferencing their HTTP URI. Nevertheless, the framework should support alternative mechanisms to do so. For example, a Source may prefer that agents obtain a changed resource from a store dedicated to synchronization rather than from the live Source itself. Also, protocols other than HTTP may be preferred to obtain changed content. And, change communication might pertain to "container" resources, for example, indicating that an OAI-PMH repository underwent changes. In this case, an agent would have to be informed about the specifics of the container resource in order to understand how to obtain its changes. For example, in the case of an OAI-PMH repository, that would be by issuing a ListRecords command with an appropriate "from" value.

3.3 Memory

To add robustness to a synchronization framework, a **memory** function must be considered that allows for synchronization in case a Destination has missed updates. Aligned with the above reasoning, two memory functions can be distinguished:

1. **change memory:** An approach to allow a Destination to catch up on missed change communications, e.g., by means of a summary of changes that were communicated during a timeframe that includes the Destination's down time.
2. **resource memory:** An approach to allow a Destination to catch up on resource versions that were created by the Source while the destination was unable to obtain them.

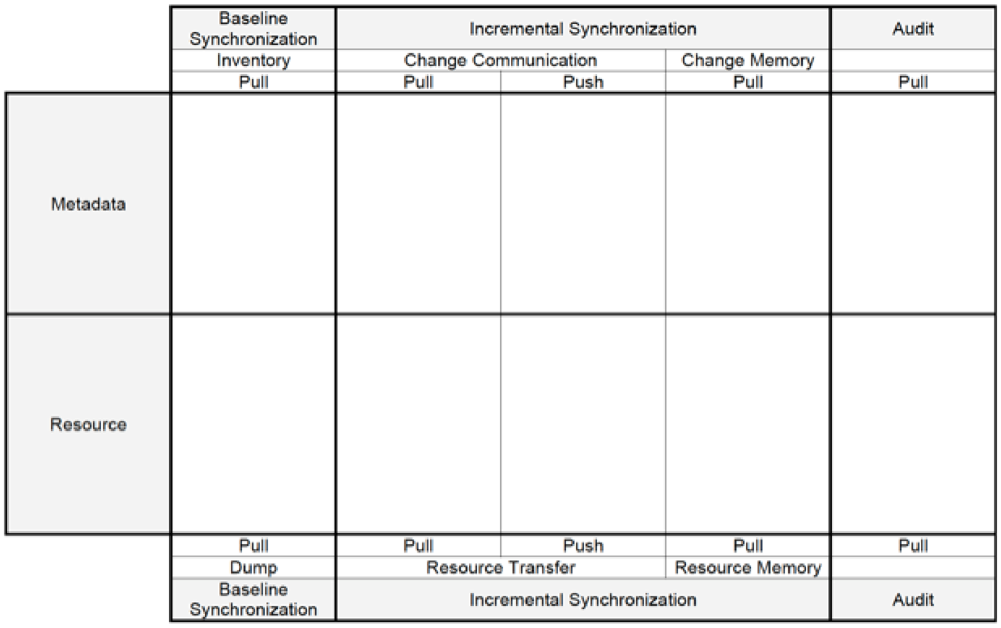
Recognizing that supporting a memory function is a costly activity raises the architectural question of where in a framework it can or should be provided: at the Source, by an intermediary? Also, it needs to be noted that not all use cases require such memories. For example, if a Destination is only interested in the current state of a Source's resources, resource memory is not needed. And, when less than perfect synchronization is acceptable, a change memory may not be essential. Therefore, a framework in which these memories can be added as optional components to a basic synchronization capability is appealing.

3.4 Discovery

Lastly, efficient Web **discovery** of several of the aforementioned functions is essential if a synchronization framework is to operate in a largely automated fashion. Discovery needs to include, but is not limited to: finding channels that provide change communications for a given Source; finding information about the nature of the resources and the types of events that a change communications channel provides messages for; being able to locate a recent dump, inventory, or message summary.

4 A ResourceSync Template

The above perspective on resource synchronization can be summarized in a template (Figure 1 below). The columns of the template list the various components of which a framework may consist; the rows indicate whether these components pertain to metadata about resources or to their actual representations. The template allows filling out components with specific technologies to design a concrete synchronization framework. Doing so allows the evaluation of characteristics of the composed framework, such as: do the various components logically fit together; is it possible to adopt the framework in a modular manner, i.e., to pick those components that meet a community's needs without the need to also adopt unnecessary components; how does the framework support discovery of components; which components fit which uses cases; etc. In subsequent work, this template will be used to illustrate and explore concrete technological choices to assemble a synchronization framework. Clearly, the evaluations enabled by the template do not address the necessity for quantitative evaluations of chosen technologies regarding synchronization latency and accuracy, but it is considered an important step towards getting a handle on the problem domain. Work regarding quantitative evaluations is also planned and will be reported on in due time.



	Baseline Synchronization	Incremental Synchronization			Audit
	Inventory	Change Communication		Change Memory	
	Pull	Pull	Push	Pull	Pull
Metadata					
Resource					
	Pull	Pull	Push	Pull	Pull
	Dump	Resource Transfer		Resource Memory	
	Baseline Synchronization	Incremental Synchronization			Audit

Figure 1: A Template for Resource Synchronization

5 Conclusions

This paper discusses the resource synchronization problem and provides a perspective that breaks it down into several component problems, each of which could be addressed by distinct technologies that hopefully can be stitched together in a modular manner to construct concrete synchronization frameworks that meet communities' requirements. This problem analysis is being used to guide prototyping work and to evaluate specific technical solutions. The results will inform the work of the NISO/OAI ResourceSync effort that has set itself the challenging goal of delivering a resource synchronization specification by mid 2013. A forthcoming paper planned for the November 2012 issue of *D-Lib Magazine* will report on progress with this regard.

6 Acknowledgements

This work is partly funded by the Sloan Foundation. Thanks to members of the NISO/OAI ResourceSync committee for their contributions: Manuel Bernhardt, Todd Carpenter, Kevin Ford, Richard Jones, Graham Klyne, Nettie Lagace, Stuart Lewis, Peter Murray, David Rosenthal, Christian Sadilek, Shlomo Sanders, Sjoerd Siebinga, Ed Summers, and Jeff Young.

7 References

[1] T. Berners-Lee and D. Connolly. Delta: an ontology for the distribution of differences between RDF graphs. 2004.

[2] J. Cho and H. Garcia-Molina. Estimating Frequency of Change. *ACM Transactions on Internet Technology (TOIT)*, 3(3):256-290, August 2003. http://dx.doi.org/10.1145/857166.857170

[3] L. R. Clausen. Concerning Etags and Datestamps. In *Proceedings of IWAW '04*, 2004.

[4] D. Fetterly, M. Manasse, M. Najork, and J. Wiener. A Large-Scale Study of the Evolution of Web Pages. *SPE*, 34(2):213-237, 2004. http://dx.doi.org/10.1002/spe.577

[5] I. Jacobs and N. Walsh. Architecture of the world wide web.

[6] M. Klein and M. L. Nelson. Evaluating Methods to Rediscover Missing Web Pages from the Web Infrastructure. In *Proceedings of JCDL '10*, pages 59-68, 2010. http://dx.doi.org/10.1145/1816123.1816133

[7] C. Lagoze and H. Van de Sompel. The Open Archives Initiative: Building a Low-Barrier Interoperability Framework. In *Proceedings of JCDL '01*, pages 54-62, 2001. http://dx.doi.org/10.1145/379437.379449

[8] M. L. Nelson, J. A. Smith, I. Campo, H. Van de Sompel, and X. Liu. Efficient, automatic web resource harvesting. In *Proceedings of WIDM '06*, pages 43-50, 2006. htt0://dx.doi.org/10.1145/1183550.1183560

[9] A. Ntoulas, J. Cho, and C. Olston. What's New on the Web?: The Evolution of the Web from a Search Engine Perspective. In *Proceedings of WWW '04*, pages 1-12, 2004. http://dx.doi.org/10.1145/988672.988674

[10] A. Passant, and P. Mendes. sparqlPuSH: Proactive Notification of Data Updates in RDF Stores Using PubSubHubbub. In *Proceedings of SFSW'10*, 2010.

[11] J. Umbrich, M. Hausenblas, A. Hogan, A. Polleres, and S. Decker. Towards Dataset Dynamics: Change Frequency of Linked Open Data Sources. In *Proceedings of LDOW '10*, 2010. http://hdl.handle.net/10379/1120

[12] H. Van de Sompel, M. L. Nelson, C. Lagoze, and S. Warner. Resource Harvesting within the OAI-PMH Framework. *D-Lib Magazine*, Vol. 10, No. 12, 2004. http://dx.doi.org/10.1045/december2004-vandesompel

About the Authors

Herbert Van de Sompel is an information scientist at the Los Alamos National Laboratory and for 10 years has led the Digital Library Research & Prototyping Team. The Team does research regarding various aspects of scholarly communication in the digital age, including information infrastructure, interoperability, digital preservation and alt-metrics. Dr. Van de Sompel has played a major role in creating the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), the Open Archives Initiative Object Reuse & Exchange specification (OAI-ORE), the OpenURL Framework for Context-Sensitive Services, the SFX linking server, the bX scholarly recommender service, and info URI. Currently, he works with his team on the Open Annotation, Memento and ResourceSync projects. In recognition of his contributions, he has received numerous awards including the SPARC Innovator Award and the Frederick G. Kilgour Award for Research in Library and Information Technology. He graduated in Mathematics and Computer Science at Ghent University, Belgium, and holds a Ph.D. in Communication Science from the same university. Herbert's web site is at http://public.lanl.gov/herbertv/.

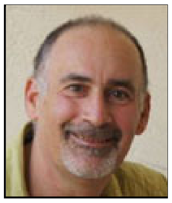
Robert Sanderson is an information scientist in the Research Library at Los Alamos National Laboratory and previously a Lecturer in Computer Science at the University of Liverpool. His areas of research are focused around scholarly communication, digital humanities and large scale data mining. He has won international awards for his research, including the 2010 Digital Preservation Award and both the Vannevar Bush Best Paper award at JCDL2011 and Best Poster Award at JCDL2012. Between 2009 and 2011, he was the UIUC GSLIS Honorary Research Fellow for his interdisciplinary work in digital humanities. He is an editor of several international specifications including, most recently, W3C Open Annotation Community Draft, IETF Memento Internet Draft, and the NISO Resource Synchronization specification.

Martin Klein received his Diploma in Computer Science from the University of Applied Sciences Berlin (2002) and his Ph.D. in Computer Science from Old Dominion University (2011). From 2002 to 2005, he was a scientist at the University of Applied Sciences in Berlin conducting research in the realm of e-Learning and mobile computing. At Old Dominion University, he was part of the Web Science and Digital Libraries Research Group led by Dr. Michael L. Nelson, and a part-time lecturer in the Computer Science Department. He currently is a Postdoctoral Research Associate at the Research Library of the Los Alamos National Laboratory. His research interests include digital preservation, the temporal aspect of web resources, and information retrieval and extraction.

Michael L. Nelson is an associate professor of computer science at Old Dominion University. Prior to joining ODU, he worked at NASA Langley Research Center from 1991-2002. He is a co-editor of the OAI-PMH, OAI-ORE, Memento, and ResourceSync specifications. His research interests include repository-object interaction and alternative approaches to digital preservation. More information about Dr. Nelson can be found at: http://www.cs.odu.edu/~mln/.

Bernhard Haslhofer is an EU Marie Curie Fellow Postdoc at Cornell University Information Science. His research interest lies in the area of global, decentralized data networks and their social, cultural, and technical contexts. Dr. Haslhofer designs, builds, and experiments with Web-based data infrastructures, works on solutions for assessing and maintaining data quality in open environments, and examines how we can (re-)use open data in application use cases. In his current research, he investigates how data networks and algorithms built on the principles of openness and decentralization could support scholarship and, vice versa, how knowledge produced in the scholarship cycle could flow into broader, possibly public data and knowledge networks. He earned a Ph.D. in Computer Science from University of Vienna.

Simeon Warner is Director of the Repositories Group at Cornell University Library. Current projects include development of an archival repository, the arXiv e-print archive (http://arxiv.org/), and Project Euclid (http://projecteuclid.org/). He was one of the developers of arXiv and his research interests include web information systems, interoperability, plagiarism detection, and open-access scholarly publishing. He has been actively involved with the Open Archives Initiative (OAI) since its inception and was one of the authors of the OAI-PMH and OAI-ORE specifications.



Carl Lagoze is an associate professor at the School of Information at the University of Michigan. Prior to this appointment, he spent a large part of his academic and research career at Cornell University, lastly in the information Science department. His research focuses on data and scholarly cyberinfrastructure, investigating both the technical aspects and the social/cultural/policy influences on cyberinfrastructure. He has been a longtime collaborator on Open Archives Initiative projects including OAI-PMH, ORE, and the present resourceSync project.

(On January 3, 2013, the misspelling of author Bernhard Haslhofer's name was corrected.)