Web Services for Recovery.gov

Erik Wilde, Eric Kansa, and Raymond Yee
School of Information, UC Berkeley

UC Berkeley School of Information Report 2009-035
October 2009

Available at http://repositories.cdlib.org/ischool/2009-035/

Abstract

One of the main goals of the `Recovery.gov` Web site is to provide information about how funds for the *American Recovery and Reinvestment Act (ARRA) of 2009* are allocated and spent. In this report, we propose a reporting architecture that would focus on the *reporting services* rather than the Web site and page design, and that uses these Web services to build the user-facing part of ARRA reporting. Our proposed architecture is based on simple and well-established Web technologies, and the main goal of this architecture is to provide citizens and watchdog groups simple and easy access to machine-readable data. Our architecture uses a more sophisticated framework than simple downloads of data files. Our proposed architecture is based on the principles of *Representational State Transfer (REST)* and uses established and widely supported Web technologies such as feeds and XML. We argue that such an architecture is easy to design and implement, easy to understand for users, and easy to work with for those who want to access ARRA reporting data in a machine-readable way.

Contents

1 Introduction **2**

2 Overview **2**

3 Discovery **4**

4 Feeds and Feed Contents **4**
- 4.1 Feeds . . . 5
- 4.2 Feed Paging . . . 5
- 4.3 Feed Contents . . . 5
- 4.4 Flagging Reports . . . 6

5 Linked Data **6**
- 5.1 URI Templates . . . 8

6 Query Services **8**

7 Conclusions **9**

8 Acknowledgements **9**

1 Introduction

This report is meant as a preliminary guide for the design and deployment of services offering public access to reporting data collected through the *American Recovery and Reinvestment Act (ARRA) of 2009*. Additional guidance will be needed, because, thus far, most publicly available information about ARRA data dissemination focuses on the redesign of `Recovery.gov` as a user-friendly Web site and hence most discussion is limited to just the *User Interface (UI)* design of that site. Unfortunately, this focus on presentation issues alone limits the usefulness of ARRA reporting data in supporting intended functions of greater disclosure, accountability, and economic and administrative efficiencies. Design priorities need to shift from focusing on how to deploy an attractive Web site toward the dissemination of ARRA reporting data in ways that support reuse of data in third-party applications [21]. ARRA data should not be shackled to one mode of presentation offered by `Recovery.gov`. Instead, public watch-dog organizations, journalists, other government agencies, and commercial interests should be able to build custom visualizations, analyses, and other services based on ARRA reporting data. Thus, the design of ARRA Web services[1] that can support such third-party applications needs to seen as a central priority and should command much greater attention.

This report highlights the need for well-designed *Web Services* provided for consumers of ARRA reporting data. These services should allow any party interested in ARRA reporting data to receive the complete set of ARRA reporting data in a timely and easily usable manner, so that at least in principle, the full functionality of `Recovery.gov` could be replicated by a third party. Specifically, we are focusing on capabilities beyond simplistic "download" features, which often provide only delayed data, in formats (such as *Excel* spreadsheets) not easily usable across a wide range of development platforms, with no support for incremental data dissemination. Bulk data downloads can be regarded as the most primitive way of making information available, harking back to the old days of FTP servers. Service orientation and Web services have come a long way since then, and we argue that the main focus of providing access to ARRA reporting data should be on Web services providing timely, comprehensive, and flexible access to data. UI features then can be regarded as a visualization of these services, but all services can also be consumed by machines.

The strategy we follow in this report follows the idea of the Plain Web [25], always attempting to use the simplest and most widely known and supported technology for any given task. In keeping with this general principle, this report recommends the feed-based dissemination of ARRA reporting data to support third-party applications, using the most widely used technologies on the Internet today: *HTTP* for service access, *Atom* for the service interface, and *XML* for the data provided by the service. This choice of widely supported technologies makes service access and consumption as open and easy as possible, allowing access from sophisticated server-based applications or from resource-constrained devices such as mobile phones. However, it is beyond the scope of this report to offer comprehensive guidance on data dissemination services. Instead, this report outlines a general set of recommendations and requirements for ARRA data dissemination. Furthermore, this document looks only at the data dissemination aspect, and not data collection, management, or long-term curation. While these are also important issues, they lie beyond the scope of this report, but are discussed in more detail in related publications [26, 27, 28].

2 Overview

The feed-based publishing of data initially proposed by the *Office of Management and Budget (OMB)* represented an exciting first step in making government transparency on the Web a reality [18]. However, subsequent guidance has moved away from this plan, first by not refining the rough sketches of the first

[1] For the purpose of this report, the term *Web services* is used for machine-usable services based on Web technologies. Specifically, it is not meant in the narrower sense of the SOAP/WS-* family of specifications, which often are summarized as "Web services" even though they are only a subset of the design possibilities when designing and implementing services based on Web technologies.

guidance [19], and then by removing the open feed-based architecture completely [17]. Statements from public officials and documents made available by federal agencies now paint a much more muddled, muted, and confusing picture for how ARRA reporting data will be published in machine-readable formats that can support third-party applications, and the focus has entirely shifted to building a UI instead of providing services.

Nevertheless, publicly available sources outline two major components of the ARRA systems architecture, `FederalReporting.gov` and `Recovery.gov`. `FederalReporting.gov` will collect and validate ARRA reports as well as offer federal officials tools to review and verify reports submitted from ARRA grantees, contractors, and subcontractors. `Recovery.gov` has the primary purpose of publishing ARRA reporting data to the public. Available documents place a heavy emphasis on the presentation aspects of `Recovery.gov`, and not on services providing access to machine-readable data.

The current ARRA systems architecture (as far as information about it is publicly available) centralizes many important functions on the `FederalReporting.gov` and `Recovery.gov` systems. However, there is no publicly available information that describes how these two systems will interact with each other. Since `Recovery.gov` will supposedly publicly disseminate data collected by `FederalReporting.gov`, the manner by which data flows from `FederalReporting.gov` to `Recovery.gov` is of critical importance. Ideally, `Recovery.gov` should use Web services offered by `FederalReporting.gov`. In this sense, the connection between `FederalReporting.gov` and `Recovery.gov` offers a logical context in which to deploy data dissemination services for the public. Why go through the trouble and expense of creating some customized method for moving data from `FederalReporting.gov` to `Recovery.gov` and then duplicate this effort to create additional Web services for public consumption? A truly open architecture would allow no back channels between `FederalReporting.gov` and `Recovery.gov` and thus would completely expose all `FederalReporting.gov` services used by `Recovery.gov` to the public as well (of course with the usual options of imposing access and usage control based on mechanisms such as access rights and API keys).

The feed-based dissemination architecture described in this report can support data transfers from `FederalReporting.gov` to `Recovery.gov`. It can also support dissemination of machine-readable data for the public. This approach has the advantage of allowing `Recovery.gov` designers to focus on public presentation and user-friendly access without making access to machine-readable data a poorly conceived afterthought. Secondly, requiring `Recovery.gov` to use the same Web services offered to the public offers enhanced transparency, because this approach avoids creation of a back channel inaccessible and invisible to the public. Finally, this approach will help gather specific requirements for the design of the public Web services. At a minimum, the public Web services need to support the capabilities needed to power public dissemination and presentation features offered by the redesigned `Recovery.gov` site. If the developers of the `Recovery.gov` site have to rely upon the same Web services offered to the public, these Web services will receive appropriate design attention and investment. From a functional point of view,[2] we cannot think of a single reason why `Recovery.gov` should have more privileged support to the services provided by `FederalReporting.gov` than any other ARRA reporting consumer.

While we recommend that the feed-based dissemination architecture described in this report supports data exchanges between `Recovery.gov` and `FederalReporting.gov`, we recognize that these systems are in active development and that other options may be implemented. Nevertheless, even if `Recovery.gov` obtains data from `FederalReporting.gov` via some other methods, the feed-based dissemination methods described here can still be deployed as the public Web services provided by one of the two sites (it is currently unclear whether `Recovery.gov` or `FederalReporting.gov` would have the primary responsibility of providing Web services). However, if feed-based dissemination is not elevated in importance so that feeds help bridge the `Recovery.gov` and `FederalReporting.gov` systems, the ARRA's overall reporting architecture will, in all likelihood, be more complex, more expensive, and more difficult to maintain and extend. In the end, feeds

[2] This functional point of view specifically excludes aspects such as access and usage control, which should be viewed as being independent of functionality and service design.

can be deployed in a variety of ways, but some ways offer better levels of transparency than others.

3 Discovery

Regardless of the specific context of deployment, feed-based dissemination must offer clear means by which services can be discovered. Unfortunately, feed autodiscovery currently is not standardized anywhere. There are developments worth noting however, including best practices, a draft for setting up an IANA registry for link relations [13], and also a proposed "feed" link relation in the draft for HTML 5 [10]. While not formally standardized, feed autodiscovery is well supported by current browsers (most browsers indicate discovered feeds by displaying an icon in the browser address bar), and could be implemented reliably with a well-defined set of implementation guidelines for Web pages offered by `Recovery.gov`.

Feed discovery is the key mechanism that connects the human-oriented Web with the machine-oriented Web. Typically, users of a Web site expect that they can "subscribe" to updated information provided on a Web page by using that page's feed. While simple in principle, paired with mechanisms such as Web forms and thus mechanisms for specifically requesting information based on a user's inputs, feeds become a flexible and powerful mechanism by which services can provide machine-readable access to information sources [28]. It is important to note that by using appropriate design (as described in Section 4), these feeds can be consumed in normal human-oriented feed readers such as the popular *Google Reader*, and yet they do provide all the necessary information (in the form of links) to get access to machine-readable versions of the information carried in the feed. This is a marked difference to the more detached vision of the *Semantic Web* [2], which uses a data representation and access method which has very little common with the human-oriented Web, and thus does not provide the same seamless integration of human- and machine-readable data that can be achieved by using a more lightweight set of Web technologies [25].

4 Feeds and Feed Contents

Feeds are machine-readable lists of information items, that in principle can carry any kind of content, but in practice often carry HTML so that the information can be viewed in a browser.[3] We strongly recommend that Recovery reporting systems adopt the Atom syndication format [14] for feeds, as we have demonstrated with simulated Recovery reporting data at `http://recovery.berkeley.edu/feeds/reports1.atom`. Atom is open, enjoys very wide use, and has several important extensions useful for implementation in the Recovery Web site. Atom also requires unique identifiers to be assigned for each feed entry. These identifiers provide a critical component of a data dissemination infrastructure. These identifiers can be used to disambiguate specific Recovery reports or even specific Recovery Web services.

The feed entries should contain links to different representations of Recovery reports. Some of these representations should be optimized for display on Web browsers and presented at `Recovery.gov`. Other representations should express structured data using XML so that these data can be processed by different applications supporting visualization, analyses, notification, and other functions. The simulated Recovery reporting service accompanying this report illustrates this point by linking to XML data such as in this example: `http://recovery.berkeley.edu/feeds/reports/18.xml` from this feed entry: `http://recovery.berkeley.edu/tech/#entries`. Based on popular demand and capacity, alternative representations (such as RDF) could be supported as well. Alternatively, comma seperated value (`CSV`) representations can make data convenient for some consumers confortable with spreadsheet applications. There is no limit to the variety of alternate representations that might be available for a feed entry.

[3] The popular *podcast* is an example in which feeds provide access to other information as well, in most cases, to video or audio clips.

4.1 Feeds

Each item in a feed is represented by an entry in the feed, and each entry will typically have a number of different properties (Atom supports a number of standardized data fields). These properties describe when the data was updated, who published and created the data, and where the complete record of the data can be obtained. Such descriptions make timely information easy to find.

4.2 Feed Paging

Based on the service a user is accessing, ARRA feed-based services may offer many dozens, hundreds, or even thousands of reports collected over several weeks and months. Publishing this volume of reporting data via feeds will become unwieldy without methods to "page" through entries. Atom *Feed Paging and Archiving* [12] offers a standards-based approach for representing feeds with large numbers of entries. We strongly recommend making feed paging and archiving mandatory, so that the feeds are not just a temporary way of communicating that information has become available. Instead, the feed pages should be available as persistent and permanent access points, so that accessing information via feeds can be done robustly and reliably. To make feed paging and archiving more predictable and easier to implement, we also recommend that the guidelines should at least recommend, or maybe even require, a naming scheme for paged and archived feeds. Feed paging for simulated Recovery reporting is demonstrated here: `http://recovery.berkeley.edu/tech/#paging`. In such a publishing scenario of feeds as well as information items in XML being available via persistent URIs, it becomes significantly easier to build RESTful services[4] based on these resources.

4.3 Feed Contents

The links found in feed entry inform applications where different representations of the entry's content can be accessed. In the case of the Recovery, links can point to human readable versions, XML versions as reported to `FederalReporting.gov`, or XML versions that conform to the *Extensible Business Reporting Language (XBRL)* standard. Representations in *JavaScript Object Notation (JSON)* [4] format may also be offered as a convenience for developers seeking to build browser-based UIs and visualizations. Thus, feeds provide clear methods to guide applications in retrieval of available representations of a given resource.

ARRA reporting data are complex and often express information about many different entities (agencies, contractors, sub-contractors, funding sources, jobs creation). In our example implementation, we demonstrated a simple mapping visualization of a geo-coded set of sample data available here: `http://recovery.berkeley.edu/demos/`. However, this sample dataset could be mapped in several different ways. One could map the locations of contractors, or subcontractors, or both contractors and subcontractors (for example using lines to represent a graph of relationships). Alternatively, one could also map the locations where work was performed. Because of this complexity, ARRA data can be explored through many forms of analysis and visualization. Thus, as noted by many transparency advocates, publishing machine-readable XML representations of ARRA reports is a key requirement. Access to these XML data will enable citizen developers to easily transform and repurpose data for alternative visualizations and analyses. For example, this `http://recovery.berkeley.edu/demos/timemap/` demonstrates a visualization of simulated Recovery reporting data in time and space. It is based on another set of simulated Recovery data, transformed into the KML format.

Because the reporting data is complex, there should be published and documented schemas for the XML, based on standardized schema languages such as *XML Schema (XSD)* [23, 3] or RELAX NG [11]. More importantly, developers should have access to additional information resources explaining the reporting

[4] Named after the architectural principle underlying the Web, which is called *Representational State Transfer (REST)* [7].

XML with documentation, examples, tool sets, and any other information that might be helpful to use the XML [24].

4.4 Flagging Reports

Most of this report is concerned with "read-only" data dissemination. However, if successful, data dissemination services will support many different uses for these data across different Web-based applications. As people work with these different applications, they will likely discover errors, discrepancies, and other faults in Recovery reports, or they might just be interested in sharing observations and annotations. They may also discover indications of waste or even fraud. Services should be deployed to help harness the fruits of this distributed, community-driven scrutiny. These services should not be application specific, but should instead allow reporting and flagging of errors and other discrepancies via a common interface.

The *Atom Publishing Protocol (AtomPub)* [8] provides a standards-based basis for enabling different applications to publish community contributed annotations, flags, and comments about Recovery reports. AtomPub allows services to become write-enabled, in the sense that the read-oriented model of Atom is extended to support writing to the service as well. Is is entirely up to the service to decide under which circumstances and what to expect from clients, but the main issue is that AtomPub is a standardized format, well-integrated with Atom and its model of feed-based access, and thus it will be easier for developers to understand and use services that allow write-based access, allowing commenting and annotation features.

5 Linked Data

The approach described in this report, driven by a desire for openness and accessibility, uses the most widely established technologies and data formats to ensure that access to reporting data is as easy as possible. Recently, the idea of openly accessible data has been promoted under the term of "linked data", with recent recommendations [1] being centered around a very specific choice of technologies and data models (all centered around *Semantic Web* [2, 22] approaches focusing on RDF for data representation and centralized data storage). While it is possible to use these approaches for building Web applications [9], our recommendation is to use better established and more widely supported technologies, thereby lowering the barrier-to-entry and choosing a simpler toolset for achieving the same goals as with the more sophisticated technologies envisioned for the Semantic Web. For the remainder of this report, we use the term "linked data" as the general concept of publishing interlinked data representations, without referring to the one specific way of implementing it that is often associated with that term as well.[5]

Data publication involves more than exposure of data on the Web. In the case of ARRA reporting, datasets reference many different codes and coding systems. Unfortunately, public resources that describe such codes and coding systems are generally available only in the form of difficult-to-find and difficult-to-parse PDF documents (if at all). Such explanatory data needs to be made available in more usable and accessible formats and services that help provide contextual information that can help make Recovery data more useful and usable.

Beyond access to structured data, transparency advocates have great interest in processing and understanding relationships between different sets of related structured data. In keeping this requirement in mind, ARRA data dissemination services should be more *resource-oriented* than *service-oriented*. XML representations should contain links (in the form of URIs) to related data resources, thereby representing the relationships between the different concepts which are relevant for reporting. Two such relevant concepts include identifiers for agencies and companies. For example, instead of putting in just the DUNS identifier

[5]This means that both "Web services" and "linked data" in this report do not refer to any specific technology choices, and our recommendations are based on pragmatic considerations of how to choose the simplest and most widely available tools to implement these abstract concepts.

in an XML report, the report should also put in a link to some DUNS URI so that additional information from another Web resource can be retrieved about this entity.[6] The simulated Recovery reporting service accompanying this report illustrates publication of "linked data" relating to DUNS and agencies here: http://recovery.berkeley.edu/tech/#linked.

Adopting a *Resource Oriented Architecture (ROA)* [20] helps to add context and intelligibility to ARRA reporting data. Supplemental information published as additional resources can help explain the use of various coding systems and concrete codes found in Recovery reports. In addition, various representations can be offered for related resources. To expand on the above example, the DUNS URI itself should return machine-readable data about the entity identified by the DUNS number, and one of the resources linked from that resource at a DUNS URI could be a feed, providing additional information about that entity. That feed would probably have entries about various reports that may relate to the organization described at that DUNS URI. Thus, a resource oriented architecture can provide a simple means to enable data retrieval that "pivots" off of different identifiers.

Providing URIs to additional Web resources for different entities described in ARRA reports requires careful consideration of what entities should be resolvable on the Web. The Recovery reporting schema uses many different coding systems and identifiers. Publication of resources related to some of these identifiers will be of great value. Below we offer a preliminary set of identifiers for which Web resources should be available in machine-readable and linked representations:

- OMB agency, bureau, and account codes (to identify US federal government departments, agencies, and bureaus in the federal budget and their respective accounts) [5, 6, 15].
- Program Source Treasury Account Symbol (TAS) (to identify the treasury account funding the program). TAS are composed of a Treasury agency code to represent a department or agency and Treasury account code to represent the account [6, 16]. There is also 3 digit optional subaccount number. There are 309 ARRA-related Program Source TAS as of the writing of this report.[7]
- Federal Agency Code and Agency Awarding Code (NIST Special Publication 800-87) (used in the recipient reporting, e.g. 6800 is the Environmental Protection Agency), related to Treasury codes.
- DUNS identifiers (a unique nine-character number for identifier organizations that receive federal money) to indicate primary recipients and sub-recipients of ARRA funds.[8]
- Catalog of Federal Domestic Assistance (CFDA) Number for grants and loans.[9]
- Activity codes to indicate the type of activity undertaken by the contract, grant, or loan: NAICS (North American Industry Classification System)[10] or NTEE-NPC (National Taxonomy of Exempt Entities / Nonprofit Program Classification)[11].
- Congressional District (state and district number).
- Government Contracting Office Code (this element is listed in the Recipient Reporting Data Model.)[12]
- Award Number and Order Number.

[6] Strictly speaking, such a link would not have to be there because it could be inferred from the fact that something is supposed to be a DUNS number, but the important aspect is that there should be a URI that can be used to access information that is available about each individual DUNS number.

[7] https://www.federalreporting.gov/federalreporting/documentation/program%20source%20(TAS)%20reference.xls

[8] http://www.dnb.com/US/duns_update/

[9] https://www.cfda.gov/index?cck=1\&au=\&ck=

[10] http://www.census.gov/eos/www/naics/

[11] http://nccsdataweb.urban.org/PubApps/nteeSearch.php?gQry=all-core\&codeType=NPC

[12] http://www.recovery.gov/sites/default/files/FedRptgDataModel+8-18-09.pdf

- PIID (Procurement Instrument Identifier), found for instance, in ARRA-spending reports from FPDS-NG.[13]

5.1 URI Templates

RESTful and Web-friendly architecture also requires consideration of URI templates. Hierarchic relationships are part of many coding relationships. These hierarchies should be represented in the addressing system used to identify specific codes. By representing codes in this way, one can find sets of related codes in a hierarchy, and even browse the hierarchy based on a clear and simply URI structure. In most cases, the Federal Government does not offer an authoritative Web address for many entities defined in federal coding systems. Therefore, `Recovery.gov` should meet this need.

The list below shows a few examples illustrating possible URI templates for sets of resources that should be exposed by a RESTful Recovery reporting API. These examples are intended to illustrate methods for exposing sets of related data. However, designing the representations, and the links between them is not within the scope of this report. RESTful design still requires that all representations are still linked explicitly, even if URI templates such as the ones below are used to make the access points for Web services on `Recovery.gov` easier to find.

- `http://recovery.gov/symbols/omb` (feed of all OMB codes)
- `http://recovery.gov/symbols/omb/{AGENCY}` (feed of all OMB codes for that agency)
- `http://recovery.gov/symbols/omb/{AGENCY}/{BUREAU}` (feed of all budget accounts for that agency)
- `http://recovery.gov/symbols/omb/{AGENCY}/{BUREAU}/{BUDGET ACCOUNT}` (description or placeholder for the budget item, perhaps link to program descriptions)

The Treasury has its own code system and hierarchy. This also needs to be represented:

- `http://recovery.gov/symbols/treasury/` (feed of all Treasury agency codes)
- `http://recovery.gov/symbols/treasury/{AGENCY}` (feed of all treasury accounts)
- `http://recovery.gov/symbols/treasury/{AGENCY}/{Account}` (feed of all fiscal years associated with a treasury account)
- `http://recovery.gov/symbols/treasury/{AGENCY}/{Account}/{Fiscal Year}` (specific fiscal year for a specific treasury account)

Representations served at these URIs would use Atom as the container format, support features such as feed paging, and would contain representations of machine-readable data for these entities, complete with links to all related entities.

6 Query Services

Query services are closely related to the discussion of linked data above. The key design goal for querying services should be data accessibility and reuse. Queries that summarize data or perform other analyses are of secondary importance. There are many possible analyses that people may wish to perform on Recovery data, and it will be difficult or impossible to accommodate them all. Therefore, querying services should be more

[13] https://www.fpds.gov/dbsight/search.do?indexName=initiative\&templateName=awardSearch\&s=FPDSNG\&q=%20SPENDING_CATEGORY%3A%22ARRA%22

oriented toward making machine-readable representations of data available, so that third party developers can easily populate their own analysis engines and run their own specialized algorithms on that data.

Given the volume of data expected to be generated from Recovery reporting, it will be important to get data by identifiers described in Section 5. Additional querying services may include:

- By Time period (when grant/contract opportunities were announced, when awards were made, when spending took place).
- By Region (either defined by state, or some other geographic reference).
- By Taxonomy (if a budgetary taxonomy exists, classifications like "education" or "administrative costs"; could be useful querying criteria).

For each subset of requested data, the various types of reporting data should be made available through Atom feeds, so that these queries essentially would become "bookmarkable".

Such services may make Recovery reporting useful for applications besides catching waste fraud and abuse. The services can also help make Recovery data directly relevant for the intended purpose of the Recovery Act, jobs creation. For instance, a job placement service may use Recovery reporting services to identify contractors that recently won Recovery funding. The job placement service could then send resumes to organizations that are more likely to hire.

7 Conclusions

Feeds represent a major positive development in making government data more open to citizen review and reuse, and they provide a unique ability to do so my merging utility for humans as well as machines. However, effective use of feeds, or any other measure for data dissemination, requires careful consideration of the interdependencies of reporting data, and the dependencies on outside data. Therefore, we recommend the design of a *Resource Oriented Architecture* in line with the architectural principles of the Web to better express contextual relationships and to better support services that publish sets of related data. In that sense, Recovery data dissemination needs to be more far reaching than a narrow focus on publishing individual reports in a simplistic "download" style. Instead, Recovery data dissemination should also publish resources of related information to help put those reports in context.

8 Acknowledgements

We thank the Sunlight Foundation for its support through a grant.

References

[1] Daniel Bennett and Adam Harvey. Publishing Open Government Data. World Wide Web Consortium, Working Draft WD-gov-data-20090908, September 2009.

[2] Tim Berners-Lee, James A. Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001.

[3] Paul V. Biron and Ashok Malhotra. XML Schema Part 2: Datatypes Second Edition. World Wide Web Consortium, Recommendation REC-xmlschema-2-20041028, October 2004.

[4] Douglas Crockford. The application/json Media Type for JavaScript Object Notation (JSON). Internet RFC 4627, July 2006.

[5] Office of Management Executive Office of the President and Budget. Appendix C-Listing of OMB Agency/Bureau and Treasury Codes, August 2009.

[6] Office of Management Executive Office of the President and Budget. Section 79: The Budget Data System, August 2009.

[7] Roy Thomas Fielding and Richard N. Taylor. Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology*, 2(2):115–150, May 2002.

[8] Joe Gregorio and Bill de Hóra. The Atom Publishing Protocol. Internet RFC 5023, October 2007.

[9] Michael Hausenblas. Exploiting Linked Data to Build Web Applications. *IEEE Internet Computing*, 13(4):68–73, 2009.

[10] Ian Hickson and David Hyatt. HTML 5 — A Vocabulary and Associated APIs for HTML and XHTML. World Wide Web Consortium, Working Draft WD-html5-20090825, August 2009.

[11] International Organization for Standardization. Information Technology — Document Schema Definition Languages (DSDL) — Part 2: Grammar-based Validation — RELAX NG. ISO/IEC 19757-2, November 2003.

[12] Mark Nottingham. Feed Paging and Archiving. Internet RFC 5005, September 2007.

[13] Mark Nottingham. Web Linking. Internet Draft draft-nottingham-http-link-header-06, July 2009.

[14] Mark Nottingham and Robert Sayre. The Atom Syndication Format. Internet RFC 4287, December 2005.

[15] Budget Analysis Branch Office of Management and Budget. Public Budget Database User's Guide: Budget of the United States Government Fiscal Year 2010, May 2009.

[16] US Department of Treasury. FAST Book (Federal Account Symbols and Titles). http://www.fms.treas.gov/fastbook/index.html.

[17] Peter R. Orszag. Implementing Guidance for the Reports on Use of Funds Pursuant to the American Recovery and Reinvestment Act of 2009. Office of Management and Budget Memorandum M-09-21, June 2009.

[18] Peter R. Orszag. Initial Implementing Guidance for the American Recovery and Reinvestment Act of 2009. Office of Management and Budget Memorandum M-09-10, February 2009.

[19] Peter R. Orszag. Updated Implementing Guidance for the American Recovery and Reinvestment Act of 2009. Office of Management and Budget Memorandum M-09-15, April 2009.

[20] Leonard Richardson and Sam Ruby. *RESTful Web Services*. O'Reilly & Associates, Sebastopol, California, May 2007.

[21] David Robinson, Harlan Yu, William P. Zeller, and Edward W. Felten. Government Data and the Invisible Hand. *Yale Journal of Law & Technology*, 11, 2008.

[22] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101, March 2006.

[23] Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn. XML Schema Part 1: Structures Second Edition. World Wide Web Consortium, Recommendation REC-xmlschema-1-20041028, October 2004.

[24] Erik Wilde. Structuring Namespace Descriptions. In *Poster Proceedings of the 15th International World Wide Web Conference*, Edinburgh, UK, May 2006. ACM Press.

[25] Erik Wilde. The Plain Web. In *Proceedings of the First International Workshop on Understanding Web Evolution (WebEvolve2008)*, pages 79–83, Beijing, China, April 2008.

[26] Erik Wilde. Feeds as Query Result Serializations. Technical Report 2009-030, School of Information, UC Berkeley, Berkeley, California, April 2009.

[27] Erik Wilde, Eric Kansa, and Raymond Yee. Proposed Guideline Clarifications for American Recovery and Reinvestment Act of 2009. Technical Report 2009-029, School of Information, UC Berkeley, Berkeley, California, March 2009.

[28] Erik Wilde and Alexandros Marinos. Feed Querying as a Proxy for Querying the Web. In *Proceedings of the Eighth International Conference on Flexible Query Answering Systems*, volume 5822 of *Lecture Notes in Artificial Intelligence*, pages 663–674, Roskilde, Denmark, October 2009. Springer-Verlag.