Linked Data Integration and Consumption: A Case of Open-Interconnected Application in Swiss Administration

Fabian Cretton, Zhan Liu and Anne Le Calvé
Institute of Information Systems
University of Applied Sciences and Arts Western Switzerland, Sierre, Switzerland
{fabian.cretton, zhan.liu, anne.lecalve}@hevs.ch

ABSTRACT

Building government services with an interconnected administration approach is a worldwide trend aimed to solve the complexity of government-related processes. The connection between people and services relies on technical tools that could benefit from improvements at the data level, by more easily sharing and having a better interlinking of data from different sources. Linked data technologies have been developed to fulfill this universal need at the World Wide Web scale, and evolve it into a web of data. In this study, we move along the data lifecycle from linked data access onto its consumption and up to the design of end-users applications. As technologies for linked data publication are already at an advanced stage, we focus on the features needed for linked data consumption and present the framework we implemented in the "OverLOD Surfer" project. We apply the solution to our former work in eGovernment and semantic business process management to strengthen our assertion that the web of data approach is effective in supporting the realization of applications for an interconnected administration.

KEYWORDS

Linked Data Application, Data Integration, eGovernment, Semantic Web, Linked Open Data

1. INTRODUCTION

eGovernment is a typical and necessary topic allowing new ideas and solutions to cope with the increasing complexity in government-related processes. It is a complex ecosystem where a tremendous number of users need to exchange and share information. These users are, in a very broad sense, an administration needing to share information internally or with another administration, an official trying to understand the different impacts of a new law, a citizen looking for information, or a computer programmer implementing a new system or functionalities to provide better tools. Therefore, it is important to address the interoperability challenges in order to achieve better services.

According to [1], problems arise when we, as human beings, are overwhelmed by data, and when the number of interlocutors or stakeholders increase. Machines can be very helpful for tracking, finding, organizing, and merging information in many areas. In this respect, The World Wide Web is a successful medium to share data and it is widely used by eGovernment services. However, that information is written in Web Pages using a natural human language such as French or English. The current drawback of the Web is that software programs still have difficulties to deal with such texts, despite the fact that this field of research, called Natural Language Processing, is continually making progress. The promising approach to improve the situation of the Web is to enrich its content with computer-friendly data, known as structured data, where each piece of data has a specified meaning.

During the past decade, new technologies have been designed to evolve the Web into a Web of data, understood a Web of structured data. Referred to as "the Semantic Web" in the early days, they are known now as "linked data" [2]. According to [3], the Semantic Web is "an extension of the current web in which information is given a well-defined meaning, better enabling computers and people to work in cooperation." A huge amount of open data is already available as linked data. Such "linked open data" (LOD) has been used in many domains such as media, publications and libraries, life sciences, geography, as well as eGovernment, in which case it is referred to as "linked open government data" (LOGD). The increasing usage of LOD was expected a few years ago in [4], and the regularly updated "Linking Open Data cloud diagram" [1] shows year by year the importance of that growing trend.

Compared to the traditional ways to exchange numeric data (e.g., files, Web services or APIs), the new technologies are conceived to more easily share data and their schemas on the Web. By giving each resource a universal identifier (a URI, and more specifically for the Web, a URL) and providing natively ways to link different datasets, it is possible for the machines to comprehensively access and merge the structured data. This highly promotes 5 stars open data [2] as well as good-quality documentation for data consumers (e.g., software developers). Based on accessible and compatible data from different sources, a good data structure will allow better connections at the application level and thus facilitate humans' interactions.

The early research about linked data has been mainly oriented towards data publication and the creation of a data space at the Web scale. But the present paper discusses the challenges regarding the next step of the linked data lifecycle, which is the consumption of those distributed data to implement enterprise-ready software solutions. Our approach is based on data integration in order to maintain a local unified view of data combined from different external sources. Linked data integration has its competitive advantages, but it also presents some challenges. On one hand, handling data uniformly modeled using the native language for linked data, the Resource Description Framework (RDF), provides a strong basis. This format ensures each piece of information has a universal identifier, preventing any conflict at the schema or data level. On the other hand, the flexibility of the RDF world is a convenience for data publishers that can turn into complications for data consumers. A minor one is the different technical ways available to publish RDF data resulting in different methods to access those data. A more important one is the difficulty to uniformly query a data set originating from different sources, given the openness of RDF data schemas and representation. The flexibility of this data structure, which is also error-prone, brings the need to implement data control and constraints checking for enterprise application integration. Therefore, our research question is: How can we effectively build an application based on linked data technologies to support an interconnected administration in the domain of eGovernment?

The remainder of the paper is organized as follows: in section 2, we review previous related works on linked data access and integration. In section 3, we describe the implemented solution from data integration to end-user applications. We then present an eGovernment use case study to evaluate our proposed solution in section 4, and finally conclude and discuss directions for future research in section 5.

[1] http://lod-cloud.net/
[2] http://5stardata.info/en/

2. RELATED WORK

2.1 Linked Data Consumption

The first step towards obtaining the data is to access and consume the original source according to the way it is made available. As presented in [5], RDF data can be published in a number of ways as RDF files, RDF in HTML known as RDFa, and query services known as SPARQL end-points. Serving big RDF Dumps allows consumers to access a large amount of data from a provider. But real-life applications will often need only parts of data from a specific source, and loading a huge dataset into a local triple store can be very demanding in terms of time and resources. Therefore, [6] proposed RDFSlice for extracting a subset of a dataset dump. Such a process could scale well in a scenario where the original data does not change often and when dumps are frequently released. However, this solution for consuming data relies on an intermediary dump of the original source and does not fulfill our need to maintain an accurate and up-to-date local copy of the original source without much delay. Instead, we chose to consume data published in a more direct way that allows fine-grain consumption of specific subsets. SPARQL end-points are Web services that accept queries in the SPARQL language—the native query language for RDF data—and return the results in standard formats such as JSON, CSV, or XML. Despite allowing very powerful queries on the datasets, this architecture does not scale well as it may have to handle high demand for expensive resources. In response, [7] proposed Linked Data Fragments (LDF) to move the burden of complex queries to the client and give a rest to the server by asking it to serve only chunks of data. Currently gaining a strong interest, this architecture is based on data accessed through truly RESTful services enriched by semantic description based, for instance, on VoID [8] or Hydra [9]. Aware of the current functional requirements and starting from a very practical list of use cases,[3] the W3C is working on a specification for a Linked Data Platform (LDP) that allows it to create, read, update, and delete (CRUD) operations on RDF and non-RDF data by using standard RESTful HTTP. [10] describes how LDP will bring linked data technologies to enterprise application integration, with the main advantage of solving the data silos problem encountered by traditional Web service approaches.

The SPARQL end-point and LDP solutions have the common drawback for data publishers in maintaining a powerful server to answer user queries. We thus opted to handle the consumption of simpler and more common RDF data published in the form of small data subsets (e.g., a single resource from a dataset) that can be identified and accessed by a specific URL. This includes at least three well-adopted methods for publishing RDF data. The first one is about RDF files accessible from a URL, for instance, GeoNames resources [11] or the numerous FOAF files [12]. The second one is a standard way for publishing a Linked Data resource by making the URL of a resource dereferenceable and handling URI resolution [13]. The third one is a mixed structured content inside the traditional HTML pages using RDFa (RDF in attributes), which has become the W3C standard for "Rich Structured Data Markup for Web Documents" in HTML 5. It is noted that, as shown by [14], HTML mark-up has been widely adopted, especially since 2011 when search engines finally expressed their interests in structured contents with the launch of their own vocabulary "schema.org."[4]

[3] http://www.w3.org/TR/ldp-ucr/
[4] http://schema.org/

2.2 Linked Data Integration

As foreseen by [15], the current norm for scalable linked data application is to cache data locally by integrating multiple sources. The disadvantage of maintaining an up-to-date copy of the original sources is not a major technical issue. The effectiveness of such a solution has been proven by search engines that index and maintain a copy of the Web to answer users' queries efficiently in real-time. In [15]'s proposed architecture for linked data application, data accessed from the Web might undergo different operations to be well integrated: vocabulary mapping in order to convert data to the schema used in the data store and identity resolution to give a unique identifier to resources that have different URIs in different sources. Both of these functionalities have been implemented and validated by [16].

While vocabulary mapping and identity resolution could be interesting features for certain data sources, we consider data quality as a mandatory feature as well, with the need for a clear and precise data validation and constraint checking step. One issue with RDF, when it comes to enterprise-level data, is its schema-less data model which makes it hard to validate a set of data against a schema definition. The problem can be tackled after the data are aggregated, and the proposed algorithm of [17] assesses the quality of the integrated data. This cannot be used in our solution as we want to validate the data before its import and update to avoid obvious maintenance issues. However, [18]'s proposition of Resource Shape to address that problem has lead to the creation of the W3C RDF Data Shapes working group [5] to deliver recommendations in the coming months.

As demonstrated, while the issues concerning the linked data lifecycle needed for application development are well understood, a fully integrated solution does not yet exist. Some solutions only handle part of the issues, though some further technical standards are currently being designed. We will thus present our work towards applying an effective and promising solution.

3. OVERLOD SURFER: A LINKED DATA APPLICATION PLATFORM

The goal of the proposed tool is to facilitate the development of end-user applications based on the web of data, and thus validate the answer to our research question. An OverLOD Surfer is configured for a specific purpose in order to manage a unified view of the required data as a cache. Data from different sources is integrated in the platform, performing mandatory operations such as data validity checks and optional ones such as vocabulary mapping and identity resolution. Data is then ready for consumption at the disposal for any number of end-user applications.

We present here under the OverLOD Surfer technical architecture and the way it is implemented as modules for the Apache Marmotta platform. We then give more details about external data sources referencement and its data integration process, and after that we explain how this data is made available for end-user applications. We finally present the available data usage statistics.

[5] http://www.w3.org/2014/data-shapes/charter

3.1 OverLOD Surfer Architecture

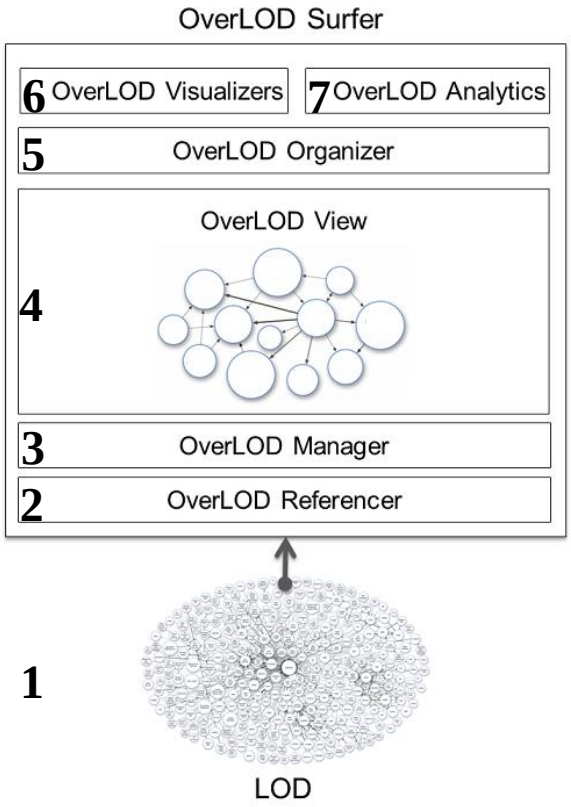

Figure 1. OverLOD Surfer architecture

Figure 1 describes how an instance of OverLOD Surfer can serve as a knowledge base for end-users tools, in a specific domain. Prior to the setup of the platform, administrators have to analyze the available data from the Web of Data (LOD) and define the trusted data sources that will serve as input for that instance. Original RDF data sources (1) are accessed through HTTP from the distributed environment of the Web. The platform's administrator (3) configures the OverLOD Referencer (2) to define the external data sources (1) that are cached in the platform. During this configuration process, the transformation and validation steps are tailored for each source. Those controls performed during each update ensure that the platform contains only valid data. The cached content is thus available as a single and reliable source of data (4). When creating end-user applications (6), developers can either access native RDF data or request specific portions of the data predefined by administrators (5) without any knowledge about RDF. Statistics are available to evaluate data consumption (7), which is also a tool for the administrator to estimate which data are effectively used and thus required in the platform.

3.2 Based on Apache Marmotta

The first step when analyzing the design of the platform was to look for existing tools. We then found an Apache top-level project with an active community: Apache Marmotta[6]. It is an open platform for linked data that offers all the basic building blocks needed for our platform, but it lacks the very feature we defined for data integration and consumption. Its open architecture allows implementing these features as separated modules that we make available for the community.

Marmotta includes a triple store and SPARQL end-point, two mandatory features to store and query RDF Data. Marmotta provides a native store called Kiwi, which is implemented on a configurable relational database back-end, but it offers the alternative to interact with a state of the art triple store if better performance is needed. This flexibility is an important feature. Besides, it supports basic authentication with the possibility to selectively restrict access to the data, a very useful feature to manage access rights for different applications. It has to be noted that Marmotta is one of the first implementations of the LDP specification that could become an interesting standard if it gains adoption in the near future.

3.3 Data Integration

Marmotta's native LDCache module allows transparent caching of LOD data from its respective URL. It is built on top of the LDClient that manages the import of data from

[6] http://marmotta.apache.org/

different formats. However, LDCache does not fit our needs: none of the identified data integration features is implemented as it is a straightforward local retrieval of data. Furthermore, it is based on a very simple storage mechanism, without the explicit identification of provenance, which does not perfectly fit rich data sets' update handling.

We therefore built the OverLOD Referencer directly on top of the LDClient. The administrator configures "External Data Sources" (EDS) that are accessible from a URL (as a .rdf file or a linked data resource, for instance). Each EDS can be assigned scripts to validate the data before import, transform it, or import only part of it.

As we have seen, linked data can be published in a number of ways, and the LDClient is already able to import some data as RDFa. It can also RDFize data to transform different data formats to RDF. LDClient basically solves the problem of data access, and we only have to modify it a little to enable the import of small RDF files published on the Web. Another useful mechanism of LDClient is that external data is first loaded in a temporary store on which we can perform the next operations before effectively importing the data. Data validation standard methods are expected in the coming months from the result of the W3C RDF Data Shapes working group. But there already exists an implemented proposition based on SPARQL Inferencing Notation: SPIN [19]. We have successfully implement SPIN to define rules for the imported data, for example, defining that certain properties are mandatory for a type of resource. We did add new rules to the proposed one, enriching data validation and the detection of constraint violations for our requirements. After validation, we then based the next steps on SPARQL Construct queries. This interesting feature of SPARQL is used to extract only the part of data needed in the platform, and, last but not least, provide flexibility when applying data transformation or vocabulary mapping on the fly.

Finally, concerning the cache data update, we have identified that this functionality must be tailored according to the source. A time stamp might be available, for instance, in the HTTP header or in the RDF data itself. This being set-up, automatic checks are performed and new versions are transparently imported.

3.4 Data Consumption for End-User Applications

We took into account that most current software developers don't know about RDF and the well-known fact that the learning curve is steep. Therefore, the platform can provide predefined views of the data in well-known formats (as CSV or JSON), based on queries prepared by administrators. But this simplicity of use doesn't impose any limitation for developers who can write their own SPARQL queries for more specific data and require richer results as JSON for linking data (JSON-LD).

The OverLOD Organizer facilitates data access. Our "data view" module enables administrators to design and manage predefined queries. A basic mechanism is implemented to define parameterized queries that offer the possibility to pass specific parameters at run-time. The development of end-user applications is made simple and "classic". Programmers don't need to have knowledge of RDF or SPARQL, but simply call a Web service with a specific data view name, the expected result format, and eventually the other parameters, if needed. Data are made available with an optimized quality and links to other datasets, even to the whole Web of data.

3.5 Data Usage Statistics

We choose to implement Google Analytics [7] tracking inside the data view Web service calls.

[7] http://www.google.com/analytics/

This gives access to a wide variety of statistical tools that can be applied on specific portions of the whole dataset. The OverLOD administrator can thus adapt the EDS according to the information actually used by the end users. In addition, as linked data publishing involves some effort, we believe that data owner will find interesting value propositions and new revenue models [20] in such a feature in order to assess the real value of their data according to the usage.

4. E-GOVERNMENT USE CASE

Concerning Switzerland, Open Government Data and eGovernment topics, [21] shows that this country is more a follower than a pioneer. Nevertheless, Swiss institutions consider Linked Open Data as a recommended technology to publish their government data ([22]; [23]) to serve new software development architecture. On the practical level, the eCH association[8] is in charge of defining standards for the Swiss administrations. In our early work [23], we provided a semantic business process to show how a semantic layer could be added to the prioritized project B.1.13[9] "eCH Process Exchange Platform for Municipalities and Cantons" (PEP). Since then, as described in [24], the PEP evolved to natively provide metadata containing references to other eCH standards. We validated this environment as a very suitable case for building Linked Open Government Data. Our semantic layer includes the semantic translation of the processes, semantic representation of two eCH standards, and the corresponding linkage being part of the semantic version of the processes meta-data.

eCH standards are published in a distributed manner, coming from different working groups. We identified two of those standards suitable to our case: the inventory of public services (eCH-0070 [10]) and the national map of public administration processes (eCH-0145[11]), present in the meta-data. The original eCH-0070 was queried from a SOAP Web service, whereas eCH-0145 was an Excel file. We transformed both of them to Linked Open Data, the schemas,[12] and the data[13] made available in a common RDF format accessible from the Web.

An instance of OverLOD Surfer is then set up to aggregate all data: Processes and their meta-data are accessed as RDF files from the PEP, and then the corresponding eCH standards are referenced by their URLs and downloaded from their URLs.

This framework clearly shows one advantage of the semantic layer: In the non-RDF version, process meta-data make reference to an eCH-0070 service identified by a simple key value as "10." It is the duty of the data consumer to know the meaning of that key, find the corresponding Web service or Excel file, and then program the match in the calling code. All consumers will have to go through that strenuous process which is a somewhat problematic situation as the number of data sources increases. This is especially so if each one of them has a specific access method and format. In the RDF version, the key of the data, a URL, directly provides its location. All the information is readily available for any software agent, as the semantic of the data has moved from the calling code, which can't be shared among consumers, to the data itself, which is available to all consumers.

We used the data integration features to perform a validity check to import only processes with meta-data links to eCH-0070 or eCH-0145, and then extracted only the part of

[8] http://www.ech.ch

[9] http://www.egovernment.ch/umsetzung/00847/index.html?lang=en

[10] http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0070

[11] http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0145

[12] http://logd.ch/voc/service

[13] http://logd.ch/eCH-0070/id/10

the process needed for the use case, involving the participants, the exchanged messages, and some other activities.

An end-user application can show which participants are involved in processes created for a specific public service and display information about the public service. This application is based on information from three data sources: the processes, their meta-data, and the eCH standards. A query is prepared by an administrator for that data view, accepting as a parameter an eCH-0070 key or a URL. The programmer of the user interface doesn't need to know about RDF and can simply make a Web service call to retrieve JSON, thereby passing the specific service ID as the parameter.

To complete the use case, modifications were performed on the original data in the processes or in the eCH standards, which were updated automatically in the cache and made available in the end-user application.

5. CONCLUSION AND FUTUR WORK

The web of data and its semantic technologies are perceived as mature enough to play a key role in large-scale data exchange, enhancing data quality, accessibility, and shareability. Linked data technologies have been around for more than a decade now, but most of the effort so far has been put on data publication. Basic consumption of linked data is an easy process, but it is not enough to fulfill the needs of ready-to-use software. Data integration is necessary to maintain a unified view of data combined from different sources. Much more than trivial data aggregation functionalities, effectively linked data integration needs to perform data validation and constraint checking, as well as other optional steps, including vocabulary mapping and identity resolution.

In this paper, we presented OverLOD Surfer, a tool for linked data integration that enables the creation of powerful end-user applications. OverLOD Surfer has been built as specific modules of the Apache Marmotta platform, adding data integration facilities to that open platform for linked data. For the consumption of data by the applications, an abstraction of the RDF technologies was developed to take into account the fact that most of today's programmers have little or no knowledge of RDF or the added fact that the learning curve is quite steep.

A practical use case has been described around a prioritized Swiss eGovernment project to show that the framework can effectively support the vision of an open and interconnected administration. The end-user application was based on information from three data sources accessible as linked data in a distributed manner. Those external data sources go through a data integration process, including validation and constraints checking, and are then consumed by the application that doesn't need to be aware of the underlying RDF model.

In our future work, we plan to closely follow the evolution and acceptation of W3Cs recommendations concerning the Linked Data Platform and RDFShape and adapt our tool accordingly. We will also enrich the data source update mechanism. Firstly, this will better handle the various existing solutions to provide data versions time stamps. Secondly, this will evaluate the "push" mechanism where the data publishers can automatically push their updated data to the platform.

6. ACKNOWLEDGEMENTS

The work presented in this paper was supported by an RCSO funding at the University of Applied Sciences and Arts Western Switzerland (HES-SO) under grant number 40160, and the name of the project is OverLOD Surfer.

7. REFERENCES

[1] R. Klischewski, "Semantic web for e-government," In R. Traunmuller (Ed.) Proceedings of EGOV, Berlin, Germany, pp. 288-295, 2003.

[2] T. Berners-Lee, "Linked Data Design Issues," W3C-Internal Document, 2006. Retrieved Mai 2015 from: http://www.w3.org/DesignIssues/LinkedData.html

[3] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," Scientific american, 284(5), 2001, pp. 28-37.

[4] C. Bizer, "The emerging web of linked data," Intelligent Systems, IEEE, 24(5), 2009, pp. 87-92.

[5] C. Bizer, R. Cyganiak, and T. Heath, "How to publish linked data on the web", 2007.

[6] E. Marx, S. Shekarpour, S. Auer, and A. C. N. Ngomo, "Large-scale RDF Dataset Slicing," In Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on, pp. 228-235. IEEE, 2013.

[7] R. Verborgh, M. Vander Sande, P. Colpaert, S. Coppens, E. Mannens, and R. Van de Walle, "Web-Scale Querying through Linked Data Fragments," In Proceedings of the 7th Workshop on Linked Data on the Web, 2014.

[8] A. Keith, and M. Hausenblas, "Describing linked datasets-on the design and usage of void, the vocabulary of interlinked datasets," In Linked Data on the Web Workshops (LDOW 09), in conjunction with 18th International World Wide Web Conference, 2009.

[9] M. Lanthaler, C. Gütl, "Hydra: A Vocabulary for Hypermedia-Driven Web APIs," In In Linked Data on the Web Workshops, 2013.

[10] N. Mihindukulasooriya, R. García-Castro, and M. Esteban-Gutiérrez, "Linked Data Platform as a novel approach for Enterprise Application Integration," In Proceedings of the 4th International Workshop on Consuming Linked Data (COLD2013), 2013.

[11] M. Wick, , and B. Vatant, "The geonames geographical database," Available from World Wide Web: http://geonames. org, 2012

[12] J. C. Paolillo, and E. Wright. "Social network analysis on the semantic web: Techniques and challenges for visualizing FOAF," In Visualizing the semantic web, pp. 229-241. Springer London, 2006.

[13] World Wide Web Consortium, "Best practices for publishing linked data," 2014.

[14] C. Bizer, K. Eckert, R. Meusel, H. Mühleisen, M. Schuhmacher, and J. Völker. "Deployment of rdfa, microdata, and microformats on the web–a quantitative analysis," In The Semantic Web–ISWC 2013, pp. 17-32. Springer Berlin Heidelberg, 2013.

[15] T. Heath, and C. Bizer. "Linked data: Evolving the web into a global data space," Synthesis lectures on the semantic web: theory and technology 1(1), 2011, pp. 1-136.

[16] A. Schultz, A. Matteini, R. Isele, C. Bizer, and C. Becker, "Ldif-linked data integration framework," In In 2nd International Workshop on Consuming Linked Data, 2011.

[17] T. Knap, J. Michelfeit, and M. Nečaský. "Linked open data aggregation: conflict resolution and aggregate quality," In Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual, pp. 106-111. IEEE, 2012.

[18] A. G. Ryman, L. H. Arnaud, and S. Speicher, "OSLC Resource Shape: A language for defining constraints on Linked Data," In Linked Data on the Web Workshops, 2013.

[19] H. Knublauch, J. Hendler, and K. Idehen, "SPIN–SPARQL inferencing notation," 2009.

[20] R. Bonazzi and Z. Liu, "Two birds with one stone. An economically viable solution for linked open data platforms," In proceedings of the 28th Bled eConference, Bled, Slovenia, pp. 77-85, June 7-10, 2015.

[21] A. C. Neuroni, R. Riedl, and J. Brugger. "Swiss Executive Authorities on Open Government Data--Policy Making beyond Transparency and Participation," In System Sciences (HICSS), 2013 46th Hawaii International Conference on, pp. 1911-1920, IEEE, 2013.

[22] C. Aschwanden, C. Bretscher, A. Bernstein, P. Farago, S. Krügel, F. Frei, C. Laux, B. Bucher, A. Neuroni, and R. Riedl, "Open Government Data Studie Schweiz," Berner Fachhochschule, 2012.

[23] Z. Liu, A. Le Calvé, F. Cretton, F. Evéquoz, and E. Mugellini, "A framework semantic business process management in e-government," In Proceedings of the IADIS International Conference WWW/INTERNET 2013, pp. 259-267, 2013.

[24] R. Schumann, S. Delafontaine, C. Taramarcaz, F. Evéquoz, "Effective Business process documentation in federal structures," In: Conference Proceeding of Informatik (Workshop BPM im Öffentlichen Sektor), pp. 1043-1058, GI, 2014.