On the Need for Model-Driven Engineering for Data Harvesters based on Experiences from the German GovData.DE Portal

Nikolay Tcholtchev and Arun Prakash
Fraunhofer Institute for Open Communication Systems FOKUS, Berlin
{nikolay.tcholtchev|arun.prakash}@fokus.fraunhofer.de

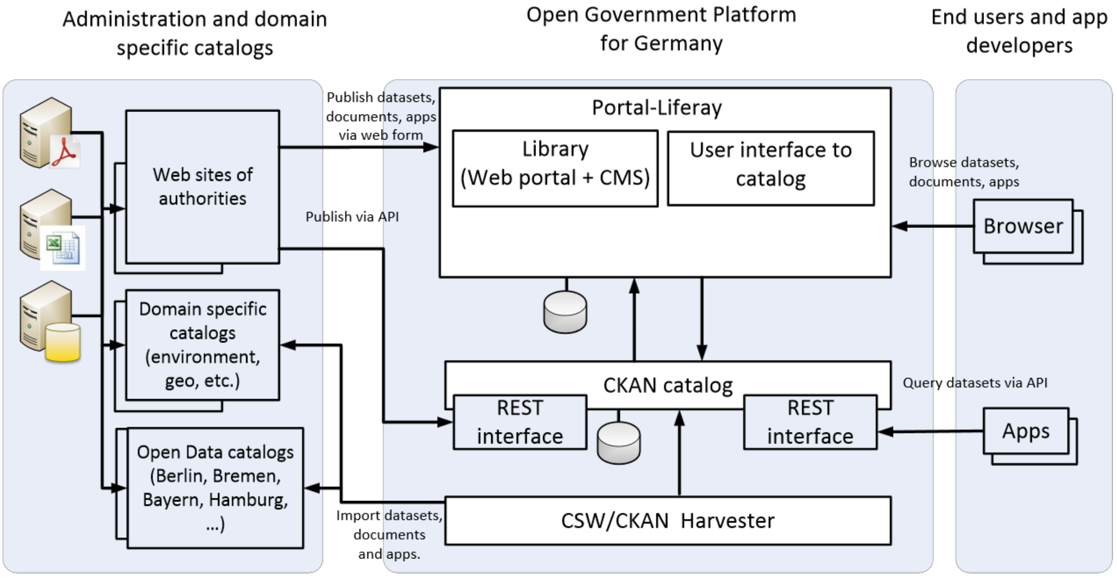
Abstract: The German governmental data platform GovData.DE has been launched in February 2013. Since then it has accommodated a large number of datasets which are made accessible over the belonging portal. Indeed, GovData.DE serves as a meta-data hub providing a single point of access to governmental data, whereby the data itself is available over the web portals of the belonging institutions (also denoted as data providers), e.g. municipalities, city councils, or federal institutions such as the Federal Statistical Office of Germany. The meta-data is regularly being obtained from the Internet platforms of the institutions in question. In order to achieve this, a large number of so-called data harvesters had to be developed, which are regularly updating the meta-data on GovData.DE, based on updates on the data providers' side. In this paper, we brief on our experiences in developing data harvesters and identify the need for a model-driven approach to the engineering of data harvesters, which at the same time constitutes a potential for various tool providers to sell and commercialize their MDE (model-driven engineering) tools. Furthermore, we argue that the use of MDE based harvesting will improve the quality and timeliness of the provided datasets (including their meta-data) and will correspondingly encourage the utilization of Open Data platforms for commercial developments.

Keywords: Open Data, Commercialization, Data Harvesting, GovData.DE

1. Introduction

The German governmental data portal (GovData.DE) is one of the key assets within the Open Data Strategy of the German Government. It was launched by the German Federal Ministry of the Interior in February 2013. Thereby, the preparation and development of the portal was worked out in close collaboration with the Fraunhofer Institute for Open Communication Systems FOKUS, which also took care of major aspects of the software development and integration. The development of GovData.DE [1] was proposed and analyzed in terms of feasibility and significance for the community in a study [8] that was completed by the Fraunhofer FOKUS institute [2] – responsible for the technical aspects, Lorenz-von-Stein-Institute [3] – responsible for the legal aspects, and Partnerschaften Deutschland [14] – in charge of analyzing the economic aspects of the emerging Open Data portal for Germany. Among others, the above mentioned study looked at potential data providers and identified the interfaces for harvesting data from the platforms of the institutions in question. The technological aspects of these processes were largely dependent on the choice of the technology for the GovData.DE portal.

The GovData.DE architecture is illustrated in Figure 1. Thereby, the CKAN (Comprehensive Knowledge Archive Network) [13] software was used as a key component in the emerging GovData.DE portal, since it constitutes a de-facto standard when it comes to managing meta-data for Open Data sets. CKAN is developed by the Open Knowledge Foundation [4] and builds the foundation for a set of Open Data portals across the globe, such as the Open Data Portal of UK [5] and the Open Data Portal of Berlin [6]. It was also used as a data hub in various research projects such as the EU-FP7 Open Cities project [7] or the Fraunhofer internal GeMo project [9] on collaborative electric mobility in Smart Cities. On top of CKAN, a Liferay [10] portal/portlet-container [11] was used to develop some components which enable the user friendly interactions between the community (data providers, app developers, data journalists etc.) and the CKAN registry in the backend.



Administration and domain specific catalogs
Open Government Platform for Germany
End users and app developers
Web sites of authorities
Domain specific catalogs (environment, geo, etc.)
Open Data catalogs (Berlin, Bremen, Bayern, Hamburg, ...)
Publish datasets, documents, apps via web form
Publish via API
Import datasets, documents and apps.
Portal-Liferay
Library (Web portal + CMS)
User interface to catalog
CKAN catalog
REST interface
REST interface
CSW/CKAN Harvester
Browse datasets, documents, apps
Browser
Query datasets via API
Apps

Figure 1: The Architecture of GovData.DE, establishing itself as a Key Component of the Open Government Platform for Germany [12]

In addition, a harvester component is running in parallel, which enables gathering meta-data from the Internet platforms of the involved data providers. The next section elucidates on the experiences that were obtained in the course of developing and operating the harvester component.

2. Experiences from the German GovData.DE Portal

Figure 1 and Figure 2 illustrate the overall harvesting architecture of GovData.DE with the CSW/CKAN Harvester as one of the key components that takes care of filling the CKAN repository with meta-data. Thereby, CSW stands for Common Service for the Web and constitutes a REST interface that allows the access to meta-data which is captured in the INSPIRE format for geographical data. In addition, a large number of CKAN based platforms are harvested over the belonging CKAN-REST interfaces. Finally, a number of data providers come up with REST services that output JSON strings, which represent the meta-data in their registries. Correspondingly, these JSON strings are also translated to the meta-data scheme used by the GovData.DE portal and imported as to be made available over the portal.

All above mentioned harvesters are implemented based on *CKAN harvesting* and *CKAN spatial* (for CSW and JSON) extensions[1] (see Figure 2). A dedicated harvester must be implemented for each data provider depending on the available interface. Thereby, most of the development work is spent in defining and implementing the mapping between the meta-data structure of the data provider

[1] *CKAN harvesting* and *CKAN spatial* constitute special CKAN extensions for meta-data harvesting

and the OGD (open government data) meta-data scheme [5], which constitutes the base for capturing meta-data within GovData.DE.

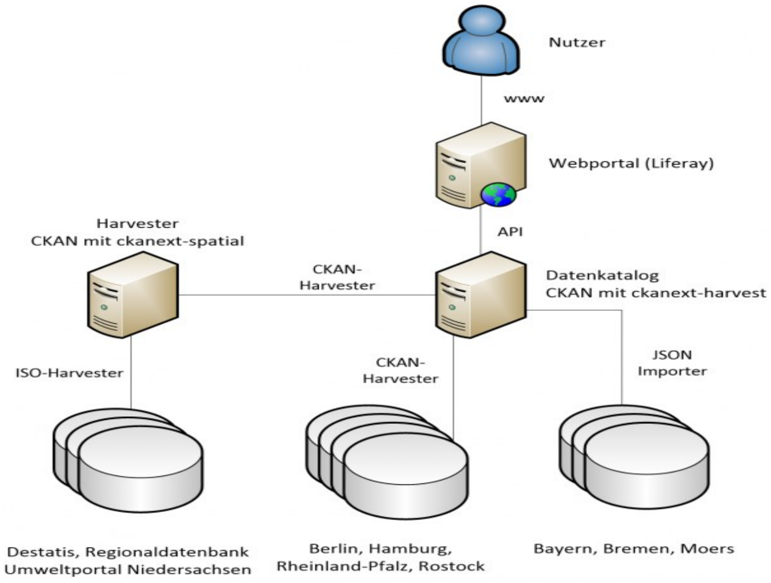


Figure 2: GovData.DE Harvesting Architecture [12]

In general, the handling of the CKAN provided harvesting platform proved to be not as straight forward as expected. This starts with the fact that all the extensions/harvesters have to be implemented in Python, which in the light of our experience is a very good language when it comes to automating processes and hacking down tools with a specific purpose, but bears a large numbers of pitfalls when it is used in a complex large scale project that requires the involvement of various developers with different coding styles. Furthermore, aspects such as quality assurance (code review) and code styling are more cumbersome in Python than in Java. In general, we believe that the maintainability of a large set of Python based CKAN harvesters becomes more difficult in parallel to the growth of functionality (e.g. number of harvesters) for the harvesting platform.

For the above reason, an effort was started to purely use the REST APIs of the involved parties – the belonging data provider on one hand and GovData.DE on the other hand – and implement some of the harvesters in Java. In that line of thoughts, Java comes with improved possibilities for code maintainability and with a larger community[2] (developer forums) which supports the resolution of coding issues.

However, first experiences indicate that the Java based harvesters do not scale as good as the CKAN based harvesters especially when it comes to harvesting a large amount of meta-data from other CKAN platforms. Thus, it is expected that in future the harvesting platform would consist of a set of Java based harvesters as well as of a set of CKAN extensions, i.e. Python based harvesters, in order to achieve optimal results with respect to speed and code maintainability at the same time.

3. The Need and Benefits of MDE based Harvesting

The discussion in the previous paragraph clearly indicates that various implementation technologies are likely to be used for the GovData.DE harvesting platform in the near future. On one hand we are

[2] As compared to pure CKAN harvesting based on Python

going to employ Java based harvesters, and on the other hand we will use traditional Python based CKAN extensions for efficient harvesting. In that line of thoughts, the following challenge emerges:

"How can the various types of harvesters, implemented based on different platforms/technologies, can be unified as to increase their overall maintenance and manageability?"

A practical answer to this challenge is given by the concept of Model-Driven Engineering, which follows the path of using abstract artefacts - called models (e.g. UML or SysML) – in order to design a system on an abstract level. Consequently, these abstract artefacts are automatically translated – by special transformations which need to be implemented – to the language of the execution platform, which in the current case is either given by Java or by Python for the CKAN extensions.

The Model-Driven Engineering of harvesters is expected to open a new field where so called tool providers can bring in their tools for MDE and that way achieve revenue. Typical high quality tools from that domain are given by Enterprise Architect, Rational Rose, and MagicDraw, to mention some. The belonging tool vendors might be selling such tools for the harvesting platforms of (Open) Data providers and that way benefit from the emerging eco-systems and business models. Thereby, the MDE providers would benefit directly or indirectly from the following aspects:

1) the fact that the quality of the data – provided by the Open Data providers – would drastically increase, given the improved harvesting processes due to the use of MDE
2) the time for the development of new harvesters will be drastically reduced since model based harvester engineering would allow a higher level abstraction and the involvement and collaborative harvester development by a larger set of collaborators – including people who are not pure developers and are more into the (governmental) data, its semantics and formatting
3) the above aspects would increase the quality of the overall set of data provided by data platforms and will facilitate and encourage the usage of (Open) Data by companies, since the provided (meta-)data would be more timely and from higher quality (data quality and trustworthiness is one of the key topics in the light of Open Data)
4) this would lead to higher competitiveness of companies and industry using (Open) Data and would for instance allow them to pay additional taxes
5) furthermore it is possible to come up with (industrial and public) fora and organizations, which would support the quality of the harvesting solutions thereby endorsing approaches such as MDE based harvesting towards establishing high quality Open Data provisioning
6) these fora might also bear a financial aspect and would be responsible for financing the MDE tool providers, e.g. by paying for licenses for the MDE tools and making these tools available to the (Open) Data providers, e.g. public institutions or non-governmental organizations

The above listed aspects constitute the various potentials, which would arise through the use of MDE tools for data harvesting. To sum up: on one hand the quality of the overall set of provided data would be increased, which would encourage the utilization of Open Data platforms by commercial developers, and on the other hand MDE tool providers would be able to bring in their products to the market of data harvesting.

4. Conclusions

In this paper, critical aspects of our meta-data harvesting experiences around the German governmental data portal (GovData.De) were presented. Based on these experiences, the need for a model-driven approach for the continuous design of harvesters was derived.

It was concluded that the utilization of tools for Model-Driven Engineering would provide tool vendors with the possibility to commercialize their tools and let them benefit from the eco-systems emerging around (Open) Data providers. In addition, we also summarized the benefits which would emerge for companies/industry as well as for MDE tool providers through the concept of model-driven harvester engineering. These include the improved quality and timeliness of available datasets, the possibility for commercial developers to rely on high quality data – which would in turn encourage the use of Open Data for commercial developments, and finally the possibility for MDE tool providers to benefit from the emerging eco-system around the topic of Open Data.

The commercialization in that context would be given by the MDE tool providers selling licenses to the data providers and indirectly benefiting from the revenue, which commercial developers achieve through the improved and timely amount of Open Data. In that context, it is possible to organize different industry fora which support the improved Open Data quality and finance indirectly the processes of MDE based harvesting.

References

1. GovData.DE: https://govdate.de, as of date 03.10.2014
2. Fraunhofer FOKUS: http://www.fokus.fraunhofer.de, as of date 03.10.2014
3. Lorenz-von-Stein Institut für Verwaltungswissenschaften: http://www.lvstein.uni-kiel.de, as of date 03.10.204
4. Open Knowledge Foundation: https://okfn.org/, as of date 03.10.2014
5. Data.gov.uk: data.gov.uk, as of date 03.10.2014
6. Berlin Open Data Portal: http://daten.berlin.de/, as of date 03.10.2014
7. Evanela Lapi, Nikolay Tcholtchev, Louay Bassbouss, Florian Marienfeld, Ina Schieferdecker: Identification and Utilization of Components for a Linked Open Data Platform. COMPSAC Workshops 2012: 112-115
8. J Klessmann, P Denker, I Schieferdecker, S Schulz: "Open government data Deutschland. Eine Studie zu Open Government in Deutschland im Auftrag des Bundesministerium des Innern", Deutschland <Bundesrepublik> / Bundesministerium, 2012
9. Nikolay Tcholtchev, Benjamin Dittwald, Thomas Scheel, Begum Ilke Zilci, Danilo Schmidt, Philipp Lammel, Jurma Jacobsen, Ina Schieferdecker, "The Concept of a Mobility Data Cloud: Design, Implementation and Trials", Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International, 192-198,2014/7/21
10. Jonas X. Yuan: Liferay Portal Enterprise Intranets - a Practical Guide to Building a Complete Corporate Intranet with Liferay. Packt Publishing, Birmingham 28. April 2008, ISBN 1847192726.
11. JSR 268: Java Specification Requests 286 - Portlet Specification 2.0
12. Florian Marienfeld, Ina Schieferdecker, Evanela Lapi, Nikolay Tcholtchev,"Metadata aggregation at GovData.de: an experience report",International Symposium on Open Collaboration (WikiSym + OpenSym) <9, 2013, Hong Kong>,638-642,ACM, 2013
13. CKAN: http://ckan.org/, as of date 03.10.2014
14. Partnerschaften Deutschland, http://www.partnerschaften-deutschland.de/, as of date 03.10.2014
15. OGDD meta-data scheme: https://github.com/fraunhoferfokus/ogd-metadata, as of date 03.10.2014