Towards Early Prototyping of Services based on Open Transport Data: A Feasibility Study

Nicolas Ferry, Aida Omerovic and Marit Kjøsnes Natvig
SINTEF, Norway

Keywords: Service Prototyping, Open Transport Data, DevOps.

Abstract: Data under open licenses and in reusable formats, often referred to as ”open data”, is increasingly being made accessible by both public and private actors. Government institutions, municipalities, private companies and entrepreneurs are among the stakeholders either having visions of new open data-based services, or just looking for new ideas on potential innovations based on open data. It is, however, in both cases, often unclear to the service developers how the open data actually can be utilized. A main reason is that the data needs to be retrieved from multiple sources, understood, quality checked and processed. While gaining insights on possible services that can be created on the top of open data, a service developer has to undergo an iterative ”trying and failing” exercise of service prototyping. In order to be practically feasible, such a process needs to be agile and efficient. Open data from the transport sector is used as a case. The open transport data are characterized by many challenges common for open data in general, but also a few specific ones. One of those challenges is the need for combining (often real-time) data from rather many sources in order to create a new service. In this paper we propose an agile approach to early service prototyping and we try out the approach on an open transport data service. Finally, we propose the priorities for future work towards a comprehensive approach for agile prototyping of open transport data-based services.

1 INTRODUCTION

During the past several years, increasingly many private and public actors all over the world have been actively releasing data under open licenses and often in reusable formats (Barometer, 2015). The goal is to foster creation of new and innovative digital services. The innovation and economic potential is becoming more and more visible, as documented by a European study (Carrara et al., 2015), thus attracting governments, municipalities, companies and entrepreneurs to take part in the ecosystem of the data provision and creation of innovations on the top of open data. Once the data are released and announced through a public catalogue, a developer needs to understand its format and content, evaluate its quality and then (at least partially) create a new service through several iterations. This process is necessary in order to try out the ideas and evaluate feasibility of the envisioned service. Such a creative process of “trying and failing” to develop new services needs to be highly agile and efficient. The process is however slowed down since the data openly available online frequently consist of rather unstructured information (Kim et al., 2014), which makes service prototyping difficult and expensive (Rusu et al., 2013). It is also a challenge that the quality of the dataset descriptions and the meta data announced might not be good enough to give the developer the information needed (Martin et al., 2013)(Beno et al., 2017).

Some tools and methods facilitating open data-based service prototyping do exist. The Linked Data Stack (Auer et al., 2012) is a software stack consisting of a number of loosely coupled tools, each capable of performing certain sets of operations on linked data, such as data extraction, storage, querying, linking, classification, and search. The LinDA project (Hasapis et al., 2014) developed a set of tools for linked data publishing, packaged into the LinDA Workbench. In the cases of both Linked Data Stack and LinDA, the complexity of provisioning resources and managing the web application rests on the service developer who must install the tools and maintain the infrastructure. The COMSODE project (P. Hanečák, 2015) provided a set of software tools and methodology for open data processing and publishing. COSMODE is not available as an online service, but rather as a set of tools that need to be individually man-

Ferry, N., Omerovic, A. and Natvig, M.
Towards Early Prototyping of Services based on Open Transport Data: A Feasibility Study.
DOI: 10.5220/0007675402570262
In *Proceedings of the 9th International Conference on Cloud Computing and Services Science (CLOSER 2019)*, pages 257-262
ISBN: 978-989-758-365-0

aged, which implies additional burden on the developer. Datalift (Scharffe et al., 2012) is a software framework for linked data publishing. It is considered as an ”expert tool” (Scharffe et al., 2012). For example, it comes with no GUI to support data publishers in the data publication process. The Linked Data AppStore (Roman et al., 2014) is a Software-as-a-Service platform prototype for data integration on the web. Common for the mentioned tools and approaches is that they either only partially cover the prototyping process, or that they are too extensive and therefore unfit for a DevOps-driven agile approach.

We have through the research and innovation project Open Transport Data, which gathers some of the major public and private actors from the transport sector in Norway, addressed service prototyping in the context of open data from the transport domain. The following list includes the main challenges that a developer faces when prototyping services on the top of open data:

- Discovery of relevant datasets through metadata search and visualisation of datasets to better understand the data content. Public catalogues and data portals are still not comprehensive and metadata for describing the contents are only to a limited degree standardized and available.
- Understanding and using varying application programming interfaces (APIs) for data retrieval. Even though API description standards exist (*e.g.*, OpenAPI), they are not commonly used, and APIs are not documented in a standardised way.
- Combining multiple sources of open data, in order to create value added services. Travel planners will for example need information on addresses, stop points, route plans and position data from several transport service operators, maps, etc.
- Accessing real-time data from IoT and sensors. The amount of such data will increase, and new services will use real-time data streams on, for example, the conditions at locations and the movement of people, vehicles and goods.
- Handling of large volumes of data, which is possibly unstructured.
- Handling proprietary data formats. For example, standards exists for data on public transport, but for other transport types (*e.g.*, car sharing, city bikes, ride sharing) there are no standards, and proprietary data formats are used.
- Understanding the data. In many cases, domain knowledge is required in order to sufficiently understand the data contents. This is a challenge due to lack of documentation and metadata, as described above.

Clearly, these characteristics impose requirements to the approach followed for prototyping the services based on open transport data. Our goal is that a service developer (*e.g.*, an entrepreneur with limited programming background) can incrementally explore the possibilities and ideas while creating a service prototype. To that end, the approach has to be highly iterative, comprehensible to non-expert developers and cost-efficient. To the best of our knowledge, there is currently no approach which sufficiently meets the above mentioned needs and challenges. In particular, the existing approaches fail to be sufficiently agile, scalable and comprehensible in order to fit for gradual prototyping through consolidation of many data sources through multiple iterations.

In this position paper we propose an initial agile approach to early service prototyping based on open transport data. The approach is novel in the sense that it is data-centric and focuses on how to develop an idea into a prototype rather than how to implement a solution. The approach is motivated by the above listed challenges as well as experiences gained from applying the data which has been harvested into an open catalogue by the Open Transport Data project. We exemplify our approach on an open transport data service and discuss the lessons learned so far. We also outline a roadmap for the forthcoming research towards a comprehensive approach for agile prototyping of open transport data-based services.

Section 2 gives an overview of the approach. Section 3 exemplifies the approach by prototyping a service based on real-life open transport data, and Section 4 summarizes the lessons learned in this trial and discusses the threats to validity of the results. We also propose the priorities for future work which aims to provide a comprehensive approach for agile prototyping of open transport data-based services.

2 OVERVIEW OF THE APPROACH

In this section we introduce our approach for the iterative prototyping of services based on open data. We propose the prototyping process for the development of services based on open transport data, as depicted in Figure 1 . Firstly, service developers need to search for the relevant data sets. Open data is typically released in a domain-dependent way when it comes to use of terminology and data structures. However, as stated in (Noy and Brickley, 2017): “*it can be difficult to determine not only the source of the dataset that has the information that you are looking for, but also the veracity or provenance of that information*”. In particular, datasets typically lack proper description and meta-data. Due to its open nature, the data is not prepared for a specific application and can be used in many different contexts which were not necessarily anticipated at release time.

Secondly, when data is found, developers need to access and understand the data. In many cases, only looking at the documentation of the data (when available) is not enough as documentation typically fails to represent aspects such as data missing, data accuracy, etc. As a result, in order to properly understand the data, developers need to manipulate and test it.

From this stage, the developer can identify the potential usage area for the data that enables new added value services. Once the capabilities of such service are identified, and before its implementation, the developers need to prepare the data (*e.g.*, pre-processing, cleaning).

In case additional data is required to deliver the service with the desired capabilities, developers can enter a new prototyping process. If not, the prototype can then be used in other stages of the product life cycle such as code and deployment stages, for instance when part of its implementation needs to be re-developed to meet the production requirements (*e.g.*, specific framework needs to be used), or to the testing stage.

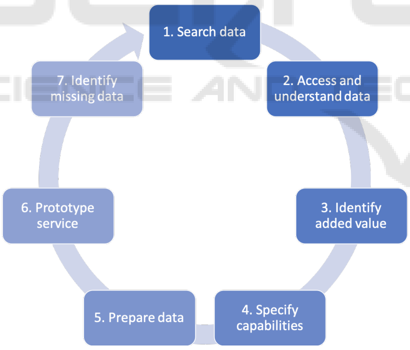


Figure 1: Data oriented early prototyping process.

The details of each of the steps of the iterative approach depicted in Figure 1 are described in the following:

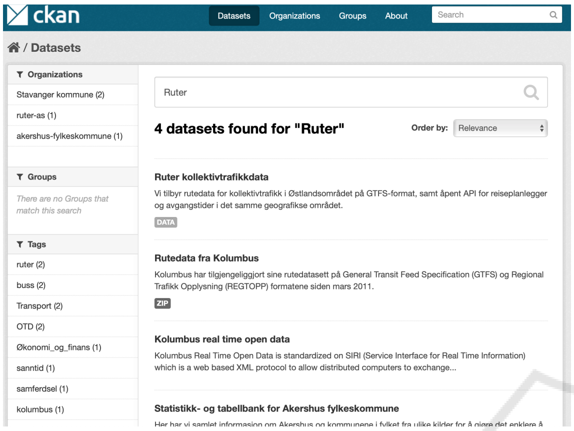
1. **Search Data:** The aim of this task is to identify the data sources and the datasets which the forthcoming prototyping iteration will be based upon. Catalogue, data repositories, and search engines can help finding the relevant datasets.
2. **Access and Understand Data:** This process consists first in understanding how the identified datasets or data streams can be accessed, second in actually accessing the data, and finally in looking at different samples of the data in order to properly understand its contents, structure, etc. These activities are often done in an ad-hoc manner as the APIs to retrieve data are typically not following API description standards.
3. **Identify Added Value:** This step requires looking into the details of the data in order to understand its contents and to identify which parts of it are relevant for our service. It is important at this stage to evaluate several samples of data in order to establish the overall quality of the data - e.g., data accuracy and the missing data.
4. **Specify Capabilities:** At this stage, the developer can start specifying the features that will be offered by the prototype. This activity will be affected by the availability of data and its identified added value.
5. **Prepare Data:** This stage consists in managing and preparing the data for further analysis and processing as part of the service business logic. This includes the following activities: data characterization, data organization, data filtering, restructuring and compression. At the end of this stage, the data should be ready to be consumed by the business logic of the service. In addition, it should fit its needs and requirements.
6. **Prototype Service:** This stage consists in the actual development, delivery and deployment of a prototype that implements the business logic of the service specified at step 4.
7. **Identify Missing Data:** At the end of a prototyping iteration, once a new set of features have been added, the developer identifies which features should be added to the prototype in the forthcoming iteration, as well as which data are required.

The cycle may be followed in several iterations, and terminates when a desired service prototype is in place, or when a stage fails in a manner that makes it impossible to proceed.

3 TRIAL OF THE APPROACH

We tried out our approach in the context of the X project, where we developed a service aiming at (i) counting all the ongoing deviations within the public transport (*e.g.*, tram delays, problems with a bus) and (ii) the average number of deviations over a week. In the following we detail the activities we performed in each of the step of our approach. The scope of the trial were open data available for the public transportation within the city of Oslo, Norway.

1. **Search Data:** We first searched for data in the Open Transport Data CKAN catalogue (see Figure 2) using "transport" and "Oslo" as keywords but we could not find relevant data. By contrast, when using the "Ruter" keyword (Ruter is the public transport authority for Oslo), we found the API of a "route planning" service.



/ Datasets

Organizations

- Stavanger kommune (2)
- ruter-as (1)
- akershus-fylkeskommune (1)

Groups

There are no Groups that match this search

Tags

- ruter (2)
- buss (2)
- Transport (2)
- OTD (2)
- Økonomi_og_finans (1)
- sanntid (1)
- samferdsel (1)
- kolumbus (1)

Ruter

4 datasets found for "Ruter"

Order by: Relevance

Ruter kollektivtrafikkdata

Vi tilbyr rutedata for kollektivtrafikk i Østlandsområdet på GTFS-format, samt åpent API for reiseplanlegger og avgangstider i det samme geografikse området.

DATA

Rutedata fra Kolumbus

Kolumbus har tilgjengeliggjort sine rutedatasett på General Transit Feed Specification (GTFS) og Regional Trafikk Opplysning (REGTOPP) formatene siden mars 2011.

ZIP

Kolumbus real time open data

Kolumbus Real Time Open Data is standardized on SIRI (Service Interface for Real Time Information) which is a web based XML protocol to allow distributed computers to exchange...

Statistikk- og tabellbank for Akershus fylkeskommune

Her har vi samlet informasjon om Akershus og kommunene i fylket fra ulike kilder for å gjøre det enklere å

Figure 2: Open Transport Data CKAN.

2. **Access and Understand Data:** We first selected the Ruter Sirisx API[1] which allowed us to retrieve, for one stop (*i.e.*, buses, tram, and subway stops), the list of ongoing deviations in all the lines using this stop. It is exposed as a REST API and can be accessed using classical tools such as "curl" or a "REST console". However, the API is little documented and we identified that we could not use directly this service as it requires as input a JSON object containing the identifier of the stop of interest. We thus searched again in the catalogue for another API providing such information, and we selected the Ruter Reise API[2] as it provides details about all the public transportation stops in Oslo, regardless of the transportation mode. We verified that the information between the two services was matching semantically - *i.e.*, we stored identifiers of a few stops from the Ruter Reise API service and thereafter we called the Sirisx API using these identifiers.
3. **Identify Added Value:** We analyzed the data from both the Ruter Reise and the Sirisx APIs. We could easily find the relevant information and in general the data was accurate even though the textual description of a deviation was sometimes incomplete or missing.
4. **Specify Capabilities:** Using these APIs we could retrieve and provide users with live information about the deviations associated to one or several stops. We also decided to retrieve and store this information on a regular basis to compute the average number of deviations over a week in the whole city.
5. **Prepare Data:** We prepared the data in two ways. First, by filtering it to only manipulate the part relevant for our service. Second, we prepared the data for further analysis. The data from the Reise API describing the stops was obtained in the form of a JSON object stringified. Unfortunately, the JSON obtained was not properly formatted as it used single quotes instead of doubles. In addition, some Norwegian language characters where not properly encoded. We thus implemented a mechanism to fix this issue before transforming the string into a proper JSON object.
6. **Prototype Service:** We implemented our service using the Node-RED platform[3], an open source project by IBM that uses a visual dataflow programming model for building applications and services. Using Node-RED, an application takes the form of a set of *nodes* (*i.e.*, software components) wired with *links* that are encapsulated in a *flow*. A flow can easily be exposed as a service using specific Node-RED nodes. Thanks to the large community behind Node-RED, a large set of nodes are available off-the-shelf and for free, making it easy to implement new applications and services. We had to implement specific nodes for accessing the two APIs and for computing the average number of deviation over a week[4]. The final flow is depicted in Figure 3
7. **Identify Missing Data:** We did not find it necessary to implement this step in the trial, as the prototype already covered the intended functionality.

4 DISCUSSION

This section first summarizes the challenges we faced during the trial and thereafter discusses the threats to validity and reliability of the results.

4.1 Lessons Learned from the Trial

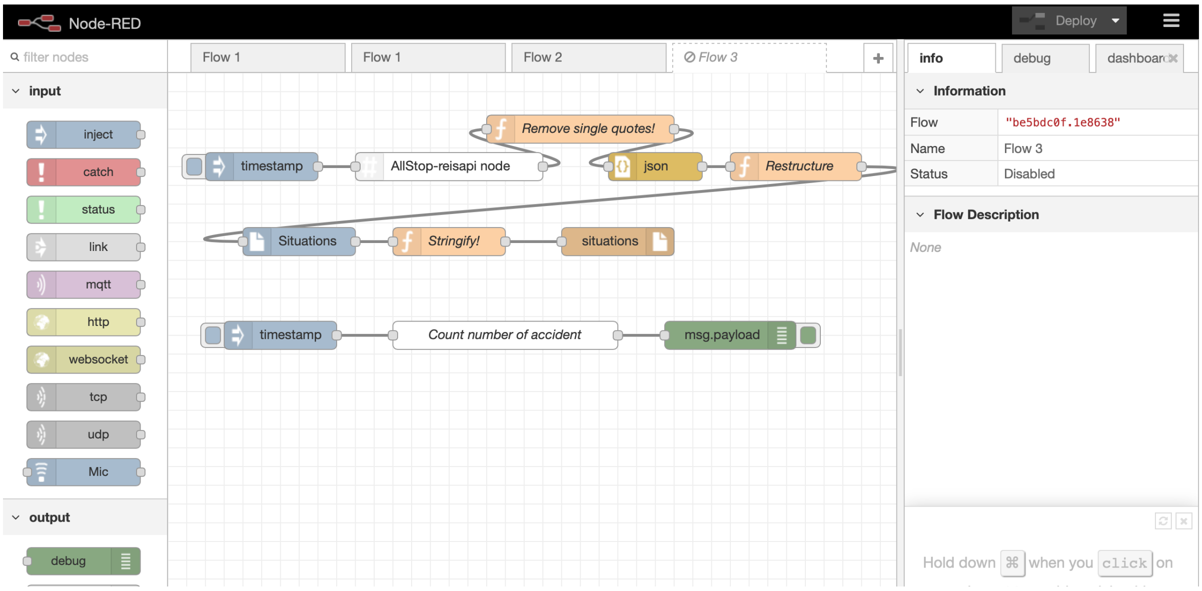
As already presented in Section 2, searching the most relevant datasets or data sources for building a specific service is challenging due to the lack of metadata about (i) the datasets (or data sources) and (ii)

[1] https://sirisx.ruter.no

[2] http://reisapi.ruter.no

[3] https://nodered.org

[4] https://github.com/SINTEF-9012/OTD-components



Flow	"be5bdc0f.1e8638"
Name	Flow 3
Status	Disabled

Figure 3: Data preparation using Node-RED.

the semantic overlaps between different datasets (or data sources). For example, it would be interesting to link datasets by means of automatic annotations with keywords that would form a domain specific ontology (Jiang and Li, 2019).

During our trial, in addition to these challenges, we first observed that many catalogues of datasets (and data sources) are available on the web, but it was difficult to make sure that we were using the best candidate. In particular, in terms of tooling, there may be a need for a cross-catalogue search engine (*i.e.*, an engine enabling searching on multiple catalogues). Similarly, once we selected our datasets or data sources, it was impossible to assess if these were the best candidates. However, in this case, it is worth noting that our agile approach, where we can start over again after trying to use the dataset, helps assessing the quality and value of different data sources.

Identifying the value of the datasets is also challenging as it can be difficult to evaluate the quality of the data. For instance, when dealing with large datasets or data streams, it is difficult to identify if some data is missing. As an example, in a large dataset with data recorded every second for a few months, it might be difficult to check if a few days or hours of recordings are missing. More generally, information about the reliability of a data source is typically not provided.

The preparation of the data does not necessarily involve complicated tasks. However, after a few steps of manipulation, it can be difficult to actually understand the status of the data being manipulated (*i.e.*, structure, format, or even the actual content of the data). In such a case, tools providing a means to visualize the data after each manipulation, would be highly beneficial. This applies not only to datasets but also to data streams.

Our approach is meant to be used during the prototyping phase of the overall life-cycle management of a service. However, it appears that this prototyping phase, by itself, would benefit from using classical tools for the continuous and agile development and operation of services. For instance, once a prototype has been implemented, it typically has to be deployed and tested in an sandbox environment. Similarly, more advanced prototypes could undergo a canary testing - *i.e.*, routing a subset of users or requests to the prototype. A deep analysis of how our approach fits within the main Agile and DevOps processes, is required.

4.2 Threats to Validity and Reliability

The validity of the results depends to a large extent on how well the threats to validity and reliability have been handled. This section discusses the essential aspects of such threats in our context.

In terms of validity, our example is only to a limited degree representative for the contexts intended to be within the scope of our approach. The trial has, however, given strong indications of feasibility of the approach. No particular customizations of the approach were needed for the trial. Thus, we have reason to believe that it should be possible to reapply our approach on new services.

Reliability is concerned with demonstrating that the empirical research can be repeated with the same results. Of course, a trial like the one we have conducted can not give solid repeatable evidence. There are several contextual factors influencing what happens, particularly the choices made by the researchers during the service development. As our main goal has been to propose an initial approach and test its feasibility through the example, performance evaluation of the approach was not addressed.

It is, in terms of evaluation, also a weakness that the researchers who tried out the approach also participated in design of the approach. As such, it is also a threat to reliability of the evaluation results, as we cannot know to what degree another service developer would have obtained the same results.

We need to further evaluate the approach in more realistic settings. There is also a need for a baseline for comparing this approach with the alternative ones, in order to assess its characteristics such as usability, usefulness and cost-effectiveness. It should be a part of the future work. Further empirical evaluation is also needed for assessing scalability of our approach with respect to complexity and size of the services to be developed.

Overall, we have drawn useful experiences from developing and instantiating the approach in the example. Although the mentioned threats to validity and reliability are present in the study, we argue that the results indicate feasibility and suggest strengths and weaknesses of the approach.

5 CONCLUSIONS

In this paper we propose an approach to early and continuous service prototyping based on open data We have also tried out the approach on an open transport data service. The results indicate feasibility and suggest strengths and weaknesses of the approach. In particular we argue for an iterative ”trying and failing” approach, as developers building services on top of open data typically need to play and understand the data while implementing a service. For this, automation should also be provided, in particular to facilitate the access to the data. Automation would also support deployment of the mechanisms and tools for (i) the prototyping and (ii) the execution of the prototype itself.

ACKNOWLEDGEMENT

This work has been funded by the Open Transport Data Project under Norwegian Research Council grant no. 257153 and by the H2020 programme under grant agreement no 780351 (ENACT).

REFERENCES

Auer, S., Bühmann, L., Dirschl, C., Erling, O., Hausenblas, M., Isele, R., Lehmann, J., Martin, M., Mendes, P. N., Van Nuffelen, B., et al. (2012). Managing the life-cycle of linked data with the lod2 stack. In *International semantic Web conference*, pages 1–16. Springer.

Barometer, O. D. (2015). Open data barometer global report. *WWW Foundation*.

Beno, M., Figl, K., Umbrich, J., and Polleres, A. (2017). Open data hopes and fears: determining the barriers of open data. In *E-Democracy and Open Government (CeDEM), 2017 Conference for*, pages 69–81. IEEE.

Carrara, W., Chan, W., Fische, S., and Steenbergen, E. v. (2015). Creating value through open data: Study on the impact of re-use of public data resources. *European Commission*.

Hasapis, P., Fotopoulou, E., Zafeiropoulos, A., Mouzakitis, S., Koussouris, S., Petychakis, M., Kapourani, B., Zanetti, N., Molinari, F., Virtuoso, S., et al. (2014). Business value creation from linked data analytics: The linda approach. In *eChallenges e-2014, 2014 Conference*, pages 1–10. IEEE.

Jiang, S., H. T. F. N. M. and Li, J. (2019). Ontology-based semantic search for open government data. In *In the proceedings of the IEEE 13th International Conference on Semantic Computing (ICSC)*. IEEE.

Kim, G.-H., Trimi, S., and Chung, J.-H. (2014). Big-data applications in the government sector. *Communications of the ACM*, 57(3):78–85.

Martin, S., Foulonneau, M., Turki, S., and Ihadjadene, M. (2013). Open data: Barriers, risks and opportunities. In *Proceedings of the 13th European Conference on eGovernment (ECEG 2013), Academic Conferences and Publishing International Limited, Reading*, pages 301–309.

Noy, N. and Brickley, D. (2017). Facilitating the discovery of public datasets.

P. Hanečák, S. Krchnavý, I. H. (2015). Comsode publication platform – open data node – final. Technical report.

Roman, D., Pop, C. D., Roman, R. I., Mathisen, B. M., Wienhofen, L., Elvesæter, B., and Berre, A. J. (2014). The linked data appstore. In *Mining Intelligence and Knowledge Exploration*, pages 382–396. Springer.

Rusu, O., Halcu, I., Grigoriu, O., Neculoiu, G., Sandulescu, V., Marinescu, M., and Marinescu, V. (2013). Converting unstructured and semi-structured data into knowledge. In *Roedunet International Conference (RoEduNet), 2013 11th*, pages 1–4. IEEE.

Scharffe, F., Atemezing, G., Troncy, R., Gandon, F., Villata, S., Bucher, B., Hamdi, F., Bihanic, L., Képéklian, G., Cotton, F., et al. (2012). Enabling linked data publication with the datalift platform. In *AAAI workshop on semantic cities*.