A Publishing Pipeline for Linked Government Data

Fadi Maali[1], Richard Cyganiak[1], and Vassilios Peristeras[2]

[1] Digital Enterprise Research Institute, NUI Galway, Ireland
{fadi.maali,richard.cyganiak}@deri.org
[2] European Commission, Interoperability Solutions for European Public Administrations
vassilios.peristeras@ec.europa.eu

Abstract. We tackle the challenges involved in converting raw government data into high-quality Linked Government Data (LGD). Our approach is centred around the idea of *self-service LGD* which shifts the burden of Linked Data conversion towards the data consumer. The self-service LGD is supported by a publishing pipeline that also enables sharing the results with sufficient provenance information. We describe how the publishing pipeline was applied to a local government catalogue in Ireland resulting in a significant amount of Linked Data published.

1 Introduction

Open data is an important part of the recent open government movement which aims towards more openness, transparency and efficiency in government. Government data catalogues, such as `data.gov` and `data.gov.uk`, constitute a corner stone in this movement as they serve as central one-stop portals where datasets can be found and accessed. However, working with this data can still be a challenge; often it is provided in a haphazard way, driven by practicalities within the producing government agency, and not by the needs of the information user. Formats are often inconvenient, (e.g. numerical tables as PDFs), there is little consistency across datasets, and documentation is often poor [6].

Linked Government Data (LGD) [2] is a promising technique to enable more efficient access to government data. LGD makes the data part of the web where it can be interlinked to other data that provides documentation, additional context or necessary background information. However, realizing this potential is costly. The pioneering LGD efforts in the U.S. and U.K. have shown that creating high-quality Linked Data from raw data files requires considerable investment into reverse-engineering, documenting data elements, data clean-up, schema mapping, and instance matching [8,16]. When `data.gov` started publishing RDF, large numbers of datasets were converted using a simple automatic algorithm, without much curation effort, which limits the practical value of the resulting RDF. In the U.K., RDF datasets published around `data.gov.uk` are carefully curated and of high quality, but due to limited availability of trained staff and contractors, only selected high-value datasets have been subjected to the Linked

E. Simperl et al. (Eds.): ESWC 2012, LNCS 7295, pp. 778–792, 2012.

Data treatment, while most data remains in raw form. In general, the Semantic Web standards are mature and powerful, but there is still a lack of practical approaches and patterns for the publishing of government data [16].

In a previous work, we presented a contribution towards supporting the production of high-quality LGD, the "self-service" approach [6]. It shifts the burden of Linked Data conversion towards the data consumer. We pursued this work to refine the self-service approach, fill in the missing pieces and realize the vision via a working implementation.

The Case for "Self-service LGD"

In a nutshell, the self-service approach enables consumers who need a Linked Data representation of a raw government dataset to produce the Linked Data themselves without waiting for the government to do so. Shifting the burden of Linked Data conversion towards the data consumer has several advantages [6]: (i) there are more of them; (ii) they have the necessary motivation for performing conversion and clean-up; (iii) they know which datasets they need, and don't have to rely on the government's data team to convert the right datasets.

It is worth mentioning that a self-service approach is aligned with civic-sourcing, a particular type of "crowd sourcing" being adopted as part of Government 2.0 to harness the wisdom of citizens [15].

Realizing the Self-service LGD

Working with the authoritative government data in a crowd-sourcing manner further emphasizes managing the tensioned balance between being easy to use and assuring quality results. A proper solution should enable producing useful results rather than merely "triple collection" and still be accessible to non-expert users. We argue that the following requirements are essential to realize the self-service approach:

Interactive approach it is vital that users have full control over the transformation process from cleaning and tidying up the raw data to controlling the shape and characteristics of the resulting RDF data. Full automatic approaches do not always guarantee good results, therefore human intervention, input and control are required.

Graphical user interface easy-to-use tools are essential to making the process swift, less demanding and approachable by non-expert users.

Reproducibility and traceability authoritative nature of government data is one of its main characteristics. Cleaning-up and converting the data, especially if done by a third party, might compromise this authoritative nature and adversely affect the data perceived value. To alleviate this, the original source of the data should be made clear along with full description of all the operations that were applied to the data. A determined user should be able to examine and re-produce all these operations starting from the original data and ending with an exact copy of the published converted data.

Flexibility the provided solution should not enforce a rigid workflow on the user. Components, tools and models should be independent from each other, yet working well together to fit in a specific workflow adopted by the user.

Decentralization there should be no requirement to register in a centralized repository, to use a single service or to coordinate with others.

Results sharing it should be possible to easily share results with others to avoid duplicating work and efforts.

In this paper, we describe how we addressed these requirements through the "LGD Publishing Pipeline". Furthermore, we report on a case study in which the pipeline was applied to publish the content of a local government catalogue in Ireland as Linked Data.
The contributions of this paper are:

1. An end-to-end publishing pipeline implementing the self-service approach. The publishing pipeline, centred around Google Refine[1], enables converting raw data available on government catalogues into interlinked RDF (section 2). The pipeline also enables sharing the results along with their provenance description on `CKAN.net`, a popular open data registry (section 2.5).
2. A formal machine-readable representation of full provenance information associated with the publishing pipeline. The LGD Publishing Pipeline is capable of capturing the provenance information, formally representing it according to the Open Provenance Model Vocabulary (OPMV)[2] and sharing it along with the data on `CKAN.net` (section 2.5).
3. A case study applying the publishing pipeline to a local government catalogue in Ireland. The resulting RDF, published as linked data as part of `data-gov.ie`, is linked to existing data in the LOD cloud. A number of widely-used vocabularies in the Linked Data community — such as VoiD[3], OPMV and Data Cube Vocabulary[4] — were utilised in the data representation. The intermix of these vocabularies enriches the data and enables powerful scenarios (section 3).

2 LGD Publishing Pipeline

The LGD Publishing Pipeline is outlined in figure 1. The proposed pipeline, governed by the requirements listed in the previous section, is in line with the process described in the seminal tutorial "How to publish Linked Data?" [4] and with various practices reported in literature [7,1].

We based the pipeline on Google Refine, a data workbench that has powerful capabilities for data massaging and tidying up. We extended Google Refine with Linked Data capabilities and enabled direct connection to government catalogues from within Google Refine. By adopting Google Refine as the basis of the pipeline we gain the following benefits:

[1] `http://code.google.com/p/google-refine/`
[2] `http://code.google.com/p/opmv/`
[3] `http://www.w3.org/TR/void/`
[4] `http://bit.ly/data-cube-vocabulary`

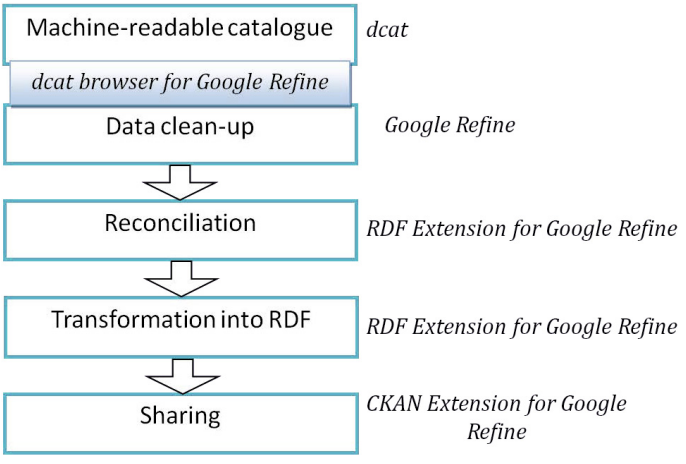

Fig. 1. Linked Data publishing pipeline (pertinent tool is shown next to each step)

- Powerful data editing, transforming and enriching capabilities.
- Rich import capabilities e.g. JSON, Excel, CSV, TSV, etc.
- Support of full and persistent undo/redo history.
- Popular in open data community.
- Extensible and under active development.
- Free and open source.

All the involved functionalities are available through a single workbench which not only supports transforming raw data into RDF; but also enables interlinking the data, capturing and formally representing all the applied operations (i.e. provenance information). The steps involved are independent from each other, yet seamlessly integrated from the user point of view. In the following subsections, we describe the involved steps outlined in figure 1.

2.1 Machine Readable Catalogues

Increasingly, governments are maintaining data catalogues listing the datasets they share with the public[5]. These catalogues play a vital role in enhancing the visibility and findability of the government datasets. However, catalogues' data is often only available through the catalogues web sites. Even when catalogues make their data available in a machine-readable format, they still use proprietary APIs and data formats. This heterogeneity hinders any effort to build tools that fully utilise and reliably access the available data.

We developed *Dcat*, an RDF vocabulary to represent government data catalogues [13]. Dcat defines terms to describe catalogues, datasets and their distributions (i.e. accessible forms through files, web services, etc.).

[5] `http://datacatalogs.org/` lists 200 catalogues as of 06/12/2011.

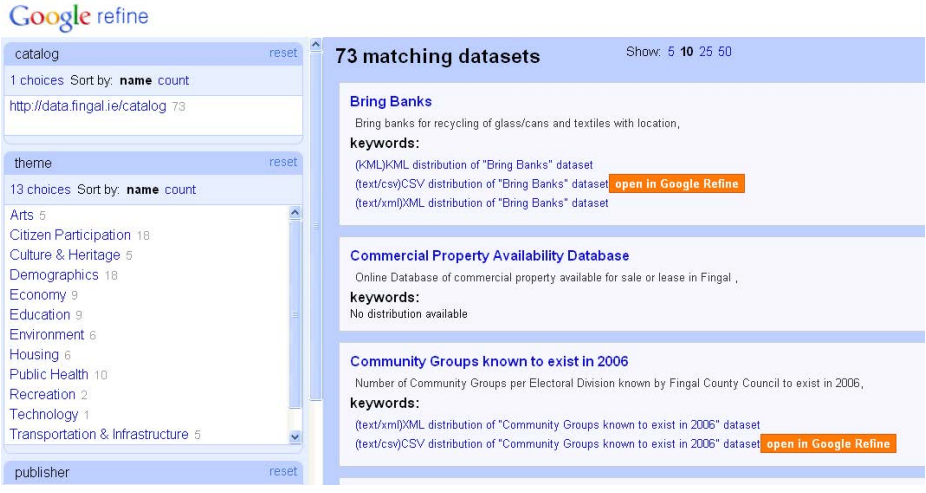

Fig. 2. Dcat Browser - navigating catalogues from within Google Refine

Dcat has been adopted by a number of government catalogues. Prominent examples of Dcat adopters include `data.gov.uk` and `semantic.ckan.net`. Currently, Dcat development is pursued under the W3C Government Linked Data Working Group[6]. Therefore, a growing adoption of it is plausible.

Our first extension to Google Refine, Dcat Browser, utilises Dcat to enable browsing government catalogues from within Google Refine. Feeding the Dcat Browser with Dcat data, via a SPARQL endpoint URL or an RDF dump, results in a faceted browser of the available datasets (figure 2). Datasets that have distributions understandable by Google Refine (e.g. CSV, Excel, TSV, etc.) can be directly opened as Google Refine project. The extension takes care of fetching files and opening them in Google Refine. Imported files can then be scrutinized and subjected to all Google Refine editing and transformation functionalities.

2.2 Data Clean-up

A stage of data preparation is necessary to fix errors, remove duplicates and prepare for transformation. Google Refine has powerful data cleaning and transformation capabilities. It also has an expressive expression language called GREL. The built-in clustering engine facilitates identifying duplicates. Additionally, facets, which are at the heart of Google Refine, help understanding the data and getting it into a proper shape before converting to RDF[7].

[6] `http://www.w3.org/egov/wiki/Data_Catalog_Vocabulary`

[7] Full documentation of Google Refine is available at: `http://code.google.com/p/google-refine/wiki/DocumentationForUsers`

2.3 Converting Raw Data into RDF

We developed the RDF Extension for Google Refine[8] to enable modelling and exporting tabular data in RDF format. The conversion of tabular data into RDF is guided through a template graph defined by the user. The template graph nodes represent resources, literals or blank nodes while edges are RDF properties (see figure 3). Nodes values are either constants or expressions based on the cells contents. Every row in the dataset generates a subgraph according to the template graph, and the whole RDF graph produced is the result of merging all rows subgraphs. Expressions that produce errors or evaluate to empty strings are ignored.

The main features of the extension are highlighted below (interested readers are encouraged to check [12]):

- RDF-centric mapping: From information integration point of view, mapping can be source-centric or target-centric. In our case it can be spreadsheet-centric or RDF-centric, respectively. RDF Extension uses the RDF-centric approach i.e. the translation process will be described in terms of the intended RDF data. RDF-centric is more expressive than the spreadsheet-centric approach [11]. Furthermore, it is closer to the conceptual model of the data rather than the representation model as expressed in the particular tabular structure of the spreadsheet.
- Expression language for custom expressions: Google Refine Expression Language GREL is used for defining custom values. GREL uses intuitive syntax and comes with a fairly rich set of functions. It also supports if-else expressions, which means that the exported RDF data can be customised based on cells' content (e.g. defining different classes based on cell content).
- Vocabularies/ontologies support: defining namespace prefixes and basic vocabulary management (add, delete and update) are supported. The RDF Extension is able to import vocabularies available on the web regardless of the format used (e.g. RDFa, RDF/XML and Turtle) as long as their deployment is compatible with the best practices recommended by the W3C in [3]. This makes it easier to reuse existing vocabularies. Such reuse not only saves effort and time but also assures that the data is more usable and not isolated. When no existing terms are suitable, users can forge their own.
- Graphical User Interface (GUI): The design of the template graph –the graph that defines the mapping– is supported by a graphical user interface where the graph is displayed as a node-link diagram. Autocomplete support for imported ontologies is also provided.
- Debugging: instant preview of the resulting RDF data is provided to enable quick debugging of the mapping. The preview is the RDF data generated from the first ten rows and serialised in Turtle syntax[9]. Turtle syntax is chosen because of its readability and compactness.

[8] `http://lab.linkeddata.deri.ie/2010/grefine-rdf-extension/`
[9] `http://www.w3.org/TR/turtle/`

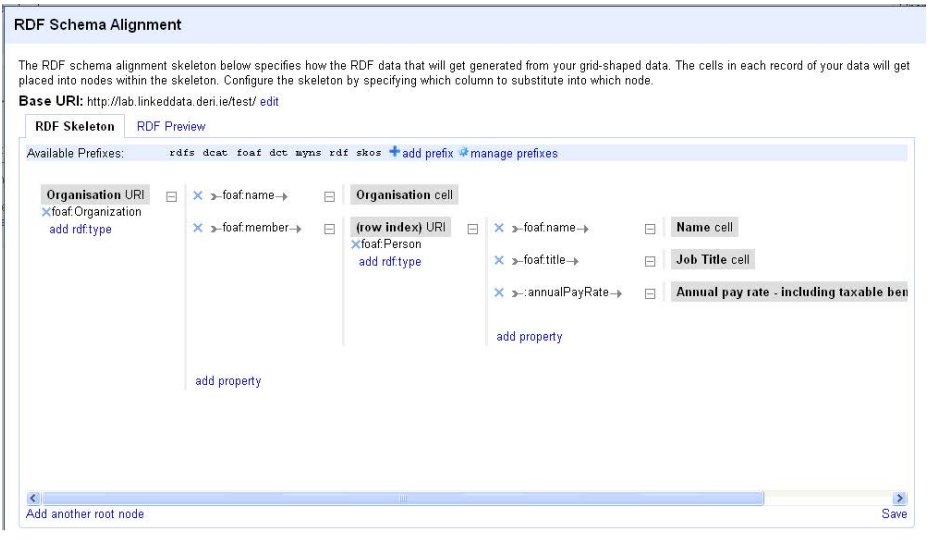

Fig. 3. RDF Extension user interface - graph template design

It is worth mentioning that in addition to the graphical representation of the mapping, users are able to access a text-based representation that can be reused, exchanged or directly edited by advanced users.

2.4 Interlinking

Linking across dataset boundaries turns the Web of Linked Data from a collection of data silos into a global data space [5]. RDF Links are established by using the same URIs across multiple datasets.

Google Refine supports data reconciliation i.e. matching a project's data against some external reference dataset. It comes with a built-in support to reconcile data against Freebase. Additional reconciliation services can be added via implementing a standard interface[10]. We extended Google Refine to reconcile against any RDF data available through a SPARQL endpoint or as a dump file. Reconciling against an RDF dataset makes URIs defined in that dataset usable in the RDF export process. As a result, interlinking is integrated as part of the publishing pipeline and enabled with a few clicks.

For example, to reconcile country names listed as part of a tabular data against DBpedia all is needed is providing Google Refine with DBpedia SPARQL endpoint URL. The reconciliation capability of the RDF Extension, will match the country names against labels in DBpedia. Restricting matching by type and adjacent properties (i.e. RDF graph neighbourhood) is also supported. In [14] we provided the full details and evaluated different matching approaches.

[10] `http://code.google.com/p/google-refine/wiki/ReconciliationServiceApi`

2.5 Sharing

The last step in the LGD Publishing Pipeline is sharing the RDF data so that others can reuse it. However, the authoritative nature of government data increases the importance of sharing a clear description of all the operations applied to the data. Ideally, provenance information is shared in a machine-readable format with a well-defined semantics to enable not only human users but also programs to access the information, process and utilise it.

We developed "CKAN Extension for Google Refine"[11] that captures the operations applied to the data, represents them according to the Open Provenance Model Vocabulary (OPMV) and enables sharing the data and its provenance on `CKAN.net`.

OPMV is a lightweight provenance vocabulary based on OPM [18]. It is used by `data.gov.uk` to track provenance of data published by the U.K. government. The core ontology of OPMV can be extended by defining supplementary modules. We defined an OPMV extension module to describe Google Refine *workflow* provenance in a machine-readable format. The extension is based on another OPMV extension developed by Jeni Tennison[12]. It is available and documented online at its namespace: `http://vocab.deri.ie/grefine#`

Google Refine logs all the operations applied to the data. It explicitly represents these operations in JSON and enables extracting and (re)applying them. The RDF related operations added to Google Refine are no exception. Both the RDF modelling and reconciling are recorded and saved in the project history. The JSON representation of the history in Google Refine is a full record of the information provenance. The extension OPMV module enables linking together the RDF data, the source data and the Google Refine operation history. Figure 4 shows an example representation of the provenance of RDF data exported using Google Refine RDF Extension. In the figure `ex:rdf_file` is an RDF file derived from `ex:csv_file` by applying operations represented in `ex:json_history_file`.

Lastly, we enabled sharing the data on CKAN.net from within Google Refine with a few clicks. CKAN.net is an "open data hub" i.e. a registry where people can publicly share datasets by registering them along with their metadata and access information. CKAN.net can be seen as a platform for crowdsourcing a comprehensive list of available datasets. It enjoys an active community that is constantly improving and maintaining dataset descriptions. CKAN Storage[13], a recent extension of CKAN, allows files to be uploaded to and hosted by CKAN.net.

A typical workflow for a CKAN contributor who wants to share the results of transforming data into RDF using Google Refine might be: (i) exporting the data from Google Refine in CSV and in RDF (ii) extracting and saving Google Refine operation history (iii) preparing the provenance description (iv) uploading the files to CKAN Storage and keeping track of the files URLs (v)

[11] `http://lab.linkeddata.deri.ie/2011/grefine-ckan`
[12] `http://purl.org/net/opmv/types/google-refine#`
[13] `http://ckan.org/2011/05/16/storage-extension-for-ckan/`

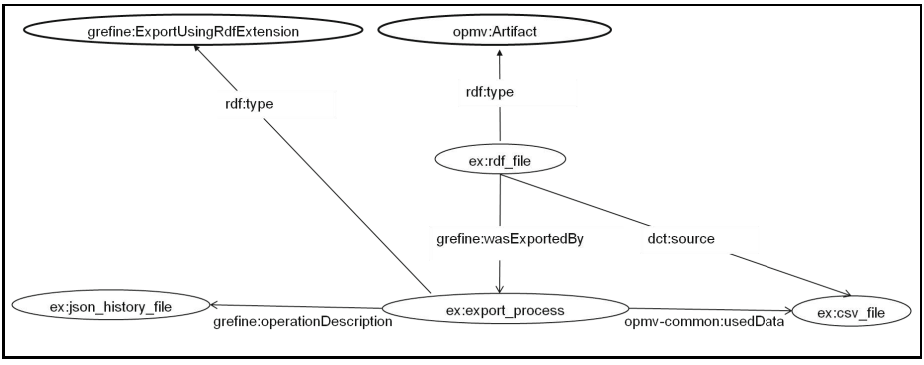

Fig. 4. RDF representation of provenance information of Google Refine RDF

updating the corresponding package on CKAN.net. CKAN Extension for Google Refine automates this tedious process to save time and efforts and to reduce errors (figure 5). In addition to uploading the files, the extension updates CKAN through its API accordingly by registering a new package or updating an existing one. The data uploaded from Google Refine can be any combination of the CSV data, RDF data, provenance description and Google Refine JSON representation of operations history.

Having the data on CKAN means that it is available online for others to use, its description can be enhanced and it can be programmatically accessed using CKAN API. Multiple RDF representations of a specific dataset can co-exist and the community aspects of CKAN.net, such as rating and tagging, can be harnessed to promote the best and spread good practices in RDF conversion.

3 Case Study - Fingal County Catalogue

Fingal is an administrative county in the Republic of Ireland. Its population is 239,992 according to the 2006 census[14]. Fingal County Council, the local authority for Fingal, is one of the four councils in the Dublin Region. It is the first council to run an open data catalogue in the Republic of Ireland.

Fingal Open Data Catalogue, available at `http://data.fingal.ie/`, enables free access to structured data relating to Fingal County. It aims to foster participation, collaboration and transparency in the county. Catalogue's datasets cover various domains from demographics to education and citizen participation. Most datasets are published by Fingal County Council and the Central Statistics Office in Ireland. Datasets are available, under Ireland PSI general license[15], in open formats such as CSV, XML and KML. In the light of Sir Tim Berners-Lee's star scheme[16], Fingal Catalogue is a 3-star one.

[14] `http://beyond2020.cso.ie/Census/TableViewer/tableView.aspx?ReportId=75467`

[15] `http://data.fingal.ie/licence/licence.pdf`

[16] `http://lab.linkeddata.deri.ie/2010/star-scheme-by-example/`

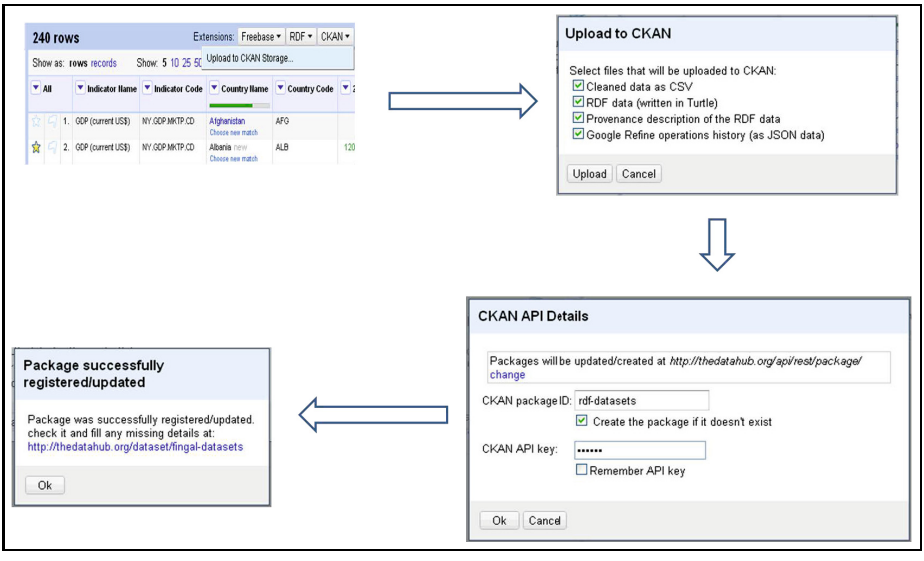

Fig. 5. User interaction flow with CKAN Extension

The catalogue provides fairly rich description of its datasets. Each dataset is categorized under one or more domain and described with a number of tags. Additionally, metadata describing spatial and temporal coverage, publisher and date of last update are also provided. Table 1 shows a quick summary of Fingal Catalogue at the time of writing.

Table 1. Fingal Catalogue summary

Number of datasets:	74 (68 available in CSV and 56 in XML)
Top publishers:	Fingal county Council (41), Central Statistics Office (17), Department of Education and Science (4)
Top domains:	Demographics(18), Citizen Participation(18), Education(9)

We applied the LGD Publishing Pipeline described in this paper to promote Fingal Catalogue to the five-star level i.e. to put the data in the interlinked RDF format. We briefly report on each of the involved steps in the following.

3.1 Machine-Readable Catalogue

Ideally, catalogues publishers make their catalogues available in some machine-readable format. Unfortunately, this is not the case with Fingal Catalogue. We had to write a scraper to get the catalogue in CSV format[17]. Then, using Google

[17] The scraper is on ScraperWiki
`http://scraperwiki.com/scrapers/fingaldata_catalogue/`

Refine with RDF Extension we converted the CSV data into RDF data adhering to the Dcat model.

Most catalogues organize their datasets by classifying them under one or more domain [13]. Dcat recommends using some standardised scheme for classification so that datasets from multiple catalogues can be related together. We used the Integrated Public Sector Vocabulary (ISPV) available from the UK government. RDF representation of ISPV (which uses SKOS) is available by the esd-toolkit as a dump file[18]. We used this file to define a reconciliation service in Google Refine and reconcile Fingal Catalogue domains against it.

3.2 Data Cleaning-up

Google Refine capabilities were very helpful with data cleaning. For example, Google Refine Expression Language (GREL) was intensively used to properly format dates and numbers to adhere to XML Schema data types syntax.

3.3 Interlinking

Electoral divisions are prevalent in the catalogue datasets especially those containing statistical information. There are no URIs defined for these electoral divisions, so we had to define new ones under `data-gov.ie`. We converted an authoritative list of electoral divisions available from Fingal County Council into RDF. The result was used to define a reconciliation service using Google Refine RDF Extension. This means that in each dataset containing electoral divisions, moving from textual names of the divisions to the URIs crafted under `data-gov.ie` is only few clicks away. A similar reconciliation was applied for councillor names. It is worth mentioning that names were sometimes spelled in different ways across datasets. For instance, Matt vs. Mathew and Robbie vs. Robert. Reconciling to URIs eliminates such mismatches.

RDF Extension for Google Refine also enabled reconciling councillor names against DBpedia and electoral divisions against Geonames.

3.4 RDF-izing

Google Refine clustering and facets were effective in giving a general understanding about the data. This is essential to anticipate and decide on appropriate RDF models for the data. Most of the datasets in the catalogue contain statistical information, we decided to use the Data Cube Vocabulary for representing this data. Data Cube model is compatible with SDMX – an ISO standard for sharing and exchanging statistical data and metadata. It extends SCOVO [10] with the ability to explicitly describe the structure of the data and distinguishes between dimensions, attributes and measures. Whenever applicable, We also used terms

[18] `http://doc.esd.org.uk/IPSV/2.00.html`

from SDMX extensions[19] which augment the Data Cube Vocabulary by defining URIs for common dimensions, attributes and measures.

For other datasets, we reused existing vocabularies whenever possible and defined small domain ontologies otherwise. We deployed new custom terms online using *vocab.deri.ie* which is a web-based vocabulary management tool facilitating vocabularies creation and deployment. As a result, all new terms are documented and dereferenceable. Newly defined terms can be checked at `http://vocab.deri.ie/fingal#`.

3.5 Sharing

With the CKAN Extension, each RDF dataset published is linked to its source file and annotated with provenance information using the OPMV extension. By linking the RDF data to its source and to Google Refine operations history, a determined user is able to examine and (automatically) reproduce all these operations starting from the original data and ending with an exact copy of the published converted data.

In total, 60 datasets were published in RDF resulting in about 300K triples[20] (a list of all datasets that were converted and the vocabularies used is available in [12]). By utilising reconciliation, the published RDF data used the same URIs for common entities (i.e. no URI aliases) and were linked to DBpedia and Geonames. Based on our previous experience in converting legacy data into RDF, we found that the pipeline significantly lowers the required time and effort. It also helps reducing errors usually inadvertently introduced when using manual conversion or custom scripts. However, issues related to URI construction, RDF data modelling and vocabulary selection are not supported and need to be tackled based on previous experience or external services.

The RDF data were then loaded into a SPARQL endpoint. We used Fuseki to run the endopint. We used the Linked Data Pages framework[21] to make the data available in RDF and HTML based on content negotiation[22]. Resolving the URI of an electoral division, as the one for the city of Howth for example, gives all the facts about Howth which were previously scattered across multiple CSV files.

The combination of Dcat, VoiD and Data Cube vocabularies helped providing a fine-grained description of the datasets and each data item. Figure 6 shows how these vocabularies were used together. Listing 1.1 shows a SPARQL query that given a URI of a data item (a.k.a fact) locates the source CSV file from which the fact was extracted. This SPARQL query enables a user who finds a particular fact in the RDF data doubtful to download the original authoritative CSV file in which the fact was originally stated.

[19] `http://publishing-statistical-data.googlecode.com/svn/trunk/specs/src/main/vocab/`

[20] The conversion required approximately two weeks effort of one of the authors.

[21] `https://github.com/csarven/linked-data-pages`

[22] The data is available online as part of http://data-gov.ie

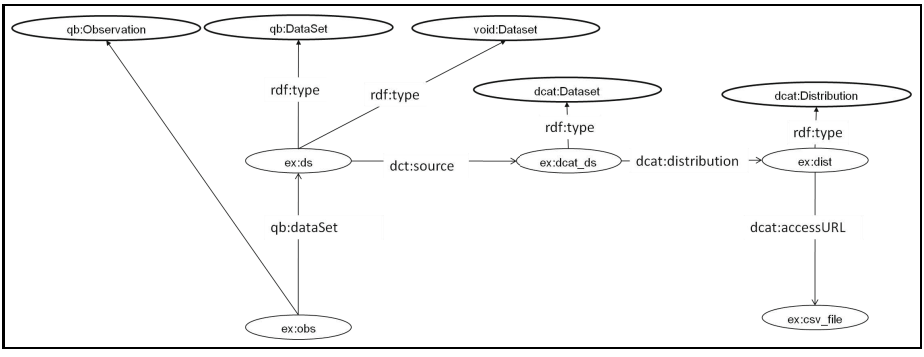

Fig. 6. The combination of Dcat, VoiD and Data Cube vocabularies to describe Fingal data

Listing 1.1. Getting the source CSV file for a particular fact (given as ex:obs)

```
SELECT ?dcat_ds ?csv_file
WHERE {
  ex:obs qb:dataSet ?qb_ds.
  ?qb_ds dct:source ?dcat_ds.
  ?dcat_ds dcat:distribution ?dist.
  ?dist dcat:accessURL ?csv_file;
        dct:format ?f.
  ?f rdfs:label 'text/csv' .
  }
```

Thanks to the RDF flexibility, the data now can also be organised and sliced in ways not possible with the previous rigid table formats.

4 Related Work

A number of tools for converting tabular data into RDF exist, most notably XLWrap [11] and RDF123 [9]. Both support rich conversion and full control over the shape of the produced RDF data. These tools focus only on the RDF conversion and do not support a full publishing process. Nevertheless, they can be integrated in a bigger publishing framework. Both RDF123 and XLWrap use RDF to describe the conversion process without providing a graphical user interface which makes them difficult for non-expert users.

Methodological guidelines for publishing Linked Government Data are presented in [17]. Similar to our work, a set of tools and guidelines were recommended. However, the tools described are not integrated into a single workbench and do not incorporate provenance description. The data-gov Wiki[23] adopts a wiki-based approach to enhance automatically-generated LGD. Their work and ours both tackle the LGD creation with a crowd-sourcing approach though in significantly different ways.

[23] `http://data-gov.tw.rpi.edu/wiki`

5 Future Work and Conclusion

In this paper, we presented a self-service approach to produce LGD. The approach enables data consumers to do the LGD conversion themselves without waiting for the government to do so. It can be seen as a civic-sourcing approach to LGD creation. To this end, we defined a publishing pipeline that supports an end-to-end conversion of raw government data into LGD. The pipeline was centred around Google Refine to employ its powerful capabilities.

We started by defining Dcat, an RDF vocabulary to describe government catalogues. Dcat was utilised to enable browsing catalogues from within Google Refine through a faceted interface. Google Refine was extended with RDF export and reconciliation functionality. Additionally, all the operations applied to the data are captured and formally represented without involving the user in the tedious and verbose provenance description. Finally, results can be shared on CKAN.net along with its provenance.

The publishing pipeline was applied to a local government catalogue in Ireland, Fingal County Catalogue. This results in a significant amount of Linked Data published. Data Cube vocabulary was used to model statistical data in the catalogue. Google Refine editing features and the added RDF capabilities were successfully applied to properly shape the data and interlink it.

Further work on the community and collaboration aspects of the publishing process would add a great value. Additionally, the problem of choosing a proper RDF model for the data is an important aspect that was not tackled in this work and cannot be considered solved in general.

Acknowledgments. The work presented in this paper has been funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2) and the European Union under Grant No. 238900 (Rural Inclusion).

References

1. Alani, H., Dupplaw, D., Sheridan, J., O'Hara, K., Darlington, J., Shadbolt, N., Tullo, C.: Unlocking the Potential of Public Sector Information with Semantic Web Technology. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 708–721. Springer, Heidelberg (2007)
2. Berners-Lee, T.: Putting Government Data Online. WWW Design Issues (2009)
3. Berrueta, D., Phipps, J.: Best Practice Recipes for Publishing RDF Vocabularies. World Wide Web Consortium, Note (August 2008)
4. Bizer, C., Cyganiak, R., Heath, T.: How to Publish Linked Data on the Web. Web page (2007) (revised 2008)
5. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems (IJSWIS) (2009)
6. Cyganiak, R., Maali, F., Peristeras, V.: Self-service Linked Government Data with dcat and Gridworks. In: Proceedings of the 6th International Conference on Semantic Systems, I-SEMANTICS 2010. ACM (2010)

7. de León, A., Saquicela, V., Vilches, L.M., Villazón-Terrazas, B., Priyatna, F., Corcho, O.: Geographical Linked Data: a Spanish Use Case. In: Proceedings of the 6th International Conference on Semantic Systems, I-SEMANTICS 2010. ACM (2010)
8. Ding, L., Lebo, T., Erickson, J.S., DiFranzo, D., Williams, G.T., Li, X., Michaelis, J., Graves, A., Zheng, J.G., Shangguan, Z., Flores, J., McGuinness, D.L., Hendler, J.: TWC LOGD: A Portal for Linked Open Government Data Ecosystems. Web Semantics: Science, Services and Agents on the World Wide Web (2011)
9. Han, L., Finin, T., Parr, C., Sachs, J., Joshi, A.: RDF123: From Spreadsheets to RDF. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 451–466. Springer, Heidelberg (2008)
10. Hausenblas, M., Halb, W., Raimond, Y., Feigenbaum, L., Ayers, D.: SCOVO: Using Statistics on the Web of Data. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 708–722. Springer, Heidelberg (2009)
11. Langegger, A., Wöß, W.: XLWrap – Querying and Integrating Arbitrary Spreadsheets with SPARQL. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 359–374. Springer, Heidelberg (2009)
12. Maali, F.: Getting to the Five-Star: From Raw Data to Linked Government Data. Master's thesis, National University of Ireland, Galway, Ireland (2011)
13. Maali, F., Cyganiak, R., Peristeras, V.: Enabling Interoperability of Government Data Catalogues. In: Wimmer, M.A., Chappelet, J.-L., Janssen, M., Scholl, H.J. (eds.) EGOV 2010. LNCS, vol. 6228, pp. 339–350. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-14799-9_29
14. Maali, F., Cyganiak, R., Peristeras, V.: Re-using Cool URIs: Entity Reconciliation Against LOD Hubs. In: Proceedings of the Linked Data on the Web Workshop 2011, LDOW 2011 (March 2011)
15. Nam, T.: The Wisdom of Crowds in Government 2.0: Information Paradigm Evolution toward Wiki-Government. In: AMCIS 2010 Proceedings (2010)
16. Sheridan, J., Tennison, J.: Linking UK Government Data. In: Proceedings of the WWW 2010 Workshop on Linked Data on the Web (LDOW 2010) (2010)
17. Villazón-Terrazas, B., Vilches-Blázquez, L.M., Corcho, O., Gómez-Pérez, A.: Methodological Guidelines for Publishing Government Linked Data. In: Wood, D. (ed.) Linking Government Data, ch. 2. Springer (2011)
18. Zhao, J.: The Open Provenance Model Vocabulary Specification. Technical Report, University of Oxford (2010)