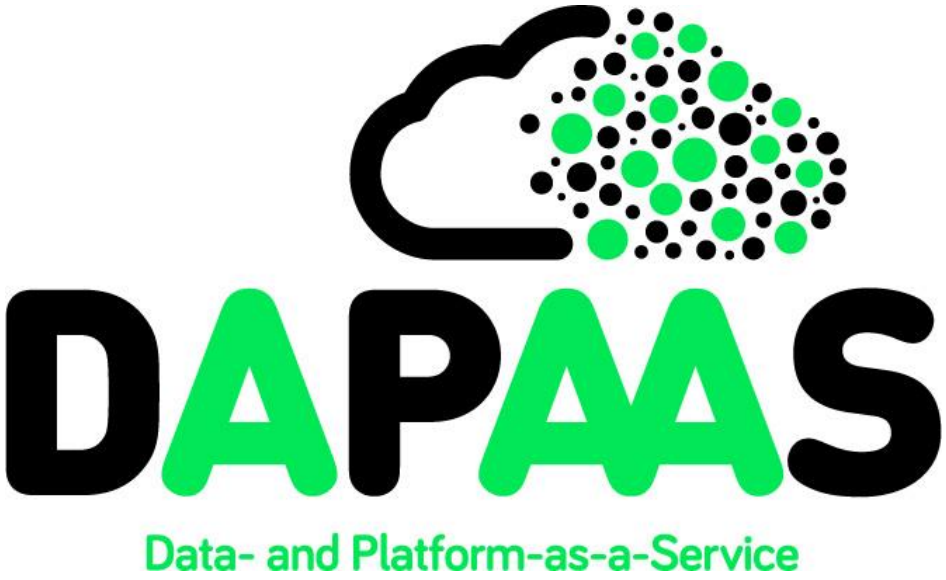
Small or medium-scale focused research project (STREP)

ICT SME-DCA Call 2013
FP7-ICT-2013-SME-DCA

Data Publishing through the Cloud: A Data- and Platform-as-a-Service Approach to Efficient Open Data Publication and Consumption

DaPaaS



Deliverable D4.3:

Evaluation of tools in realistic cases

Date:	30.10.2015
Author(s):	Bill Roberts (Swirrl), Rick Moynihan (Swirrl), Nikolay Nikolov (SINTEF)
Dissemination level:	PU
WP:	WP4
Version:	1.0

Document metadata

Quality assurors and contributors

Quality assuror(s)	Dina Suhobok (SINTEF), Alex Simov (Ontotext)
Contributor(s)	DaPaaS consortium members.

Version history

Version	Date	Description
0.1	26.10.2015	Complete draft for review
0.2	28.10.2015	Updated after review
0.3	29.19.2015	Final comments from project coordinator
1.0	30.10.2015	Final version

Executive Summary

The DaPaaS project has developed a methodology and supporting tools for transforming data from a variety of sources into Linked Data (see project deliverables D4.1 and D4.2).

The main software tool for supporting the DaPaaS data publishing methodology is the Grafter data transformation framework. Grafter has been applied and tested as a standalone software library and also via the Grafterizer front-end framework, which integrates Grafter into the overall DataGraft platform, via an intermediate service and API software layer, known as Graftwerk.

This document presents the experience of applying these tools in practical situations, to transform data into Linked Data suitable for hosting in DataGraft. This has verified that the DaPaaS Methodology captures the most important aspects of the Linked Data publishing process, and that the tools created in DaPaaS are appropriate to supporting this workflow.

Grafter and Grafterizer have been applied successfully to transforming large quantities of data from a variety of sources into Linked Data, suitable for publication through DataGraft.

The core design concept has been verified in practice: that data transformation generally requires developers and data domain experts to work in partnership and that providing tools suitable for both groups is the best solution for efficiency.

The methodology and tools are an important part of realising the potential of the DaPaaS approach to data publishing and re-use.

Table of Contents

List of Acronyms

API	Application Programming Interface
DSL	Domain Specific Language
GIS	Geographical Information System
PaaS	Platform-as-a-Service
REST	Representational state transfer
RDF	Resource Description Framework
SKOS	Simple Knowledge Organization System
SOA	Service Oriented Architecture
SPARQL	SPARQL Protocol and RDF Query Language
UI	User Interface

List of Figures

1 Introduction

The DaPaaS project has developed a Methodology and accompanying guidelines on publishing data as Linked Data: with the objective of making the data accessible and re-usable in both human-readable and machine-readable forms. The Methodology is documented in Deliverable D4.1 “Documented Methodology and Guidelines”.

The Grafter software has been developed to support the Methodology, providing tools to enable automation of data transformation from a range of data sources to Linked Data. A front-end framework to support building, testing and executing Grafter transformations, ‘Grafterizer’, has been developed and integrated into the DataGraft platform.

Deliverable D4.2 (“Software tools, integrated into the platform”) gives details of Grafter and how it fits into the DataGraft architecture. The overall architecture of DataGraft has been described in Deliverable D2.2 (“Open Data PaaS prototype v1”).

This document:

- Summarises the capabilities of Grafter and Grafterizer,
- Presents our experience of how users have applied Grafter and Grafterizer in practice, for a range of data types and a range of applications,
- Describes the lessons learned from that, and
- Lists recommendations for future work that could enhance what has already been achieved.

2 Summary of capabilities

There are three main software components that have been created to support application of the DaPaaS methodology:

- The Grafter data transformation framework has been built to support transformations from data sources in a range of formats into Linked Data, allowing automation of the main steps in the DaPaaS methodology. Defining new transformations with Grafter requires programming skills.
- The Graftwerk system has been built to allow Grafter to be used as a service, supporting integration of Grafter data transformation pipelines into the rest of the DataGraft platform.
- The Grafterizer transformation builder software makes use of Grafter, via the Graftwerk service, and provides a user interface to help the user create and execute their own data transformation, without the need for programming skills.

As discussed in D4.2 (“Software tools integrated into platform”), the software tools need to support two main audiences in their application of the methodology: Software Developers and Knowledge Workers. The Grafter software library supports the requirements of Software Developers directly. Grafterizer provides support to Knowledge Workers to use the capabilities of Grafter, without requiring software development skills.

In order to allow an efficient solution to the requirements of our main user groups, a series of high level design criteria were developed. Grafter was designed according to the following requirements:

- Modular, re-usable components to carry out specific parts of the transformation process;
- The ability to manage the entire transformation tool-chain using standard software version control approaches;
- The ability to process large data collections (hundreds of millions of triples) at reasonable speed, without being limited by available system memory;
- The ability to incorporate transformations into other software systems, to enable automation of data transfer between various systems.
- Exploitation of existing software libraries where available, for example to parse and process input files of various types, or to generate RDF in different formats.

Grafter and Grafterizer are designed to work closely together, to allow programmers and domain experts or data publishers to work together to transform data into standard re-usable formats.

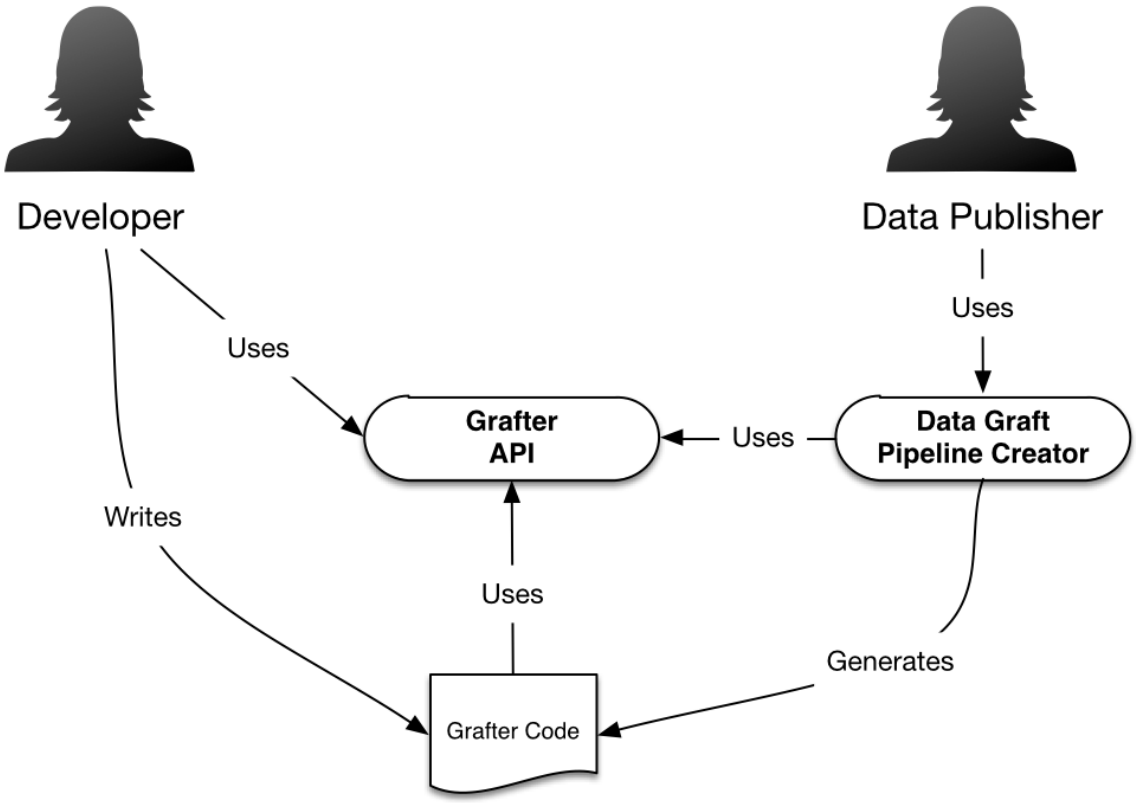


Figure 1 Collaboration between Developers and Data Publishers

3 Example applications

3.1 PLUQI

The 'PLUQI' application has been developed in Work Package 5 of DaPaaS as an example use case for how data can be applied to practical problems, via the DataGraft platform (See Deliverable D5.3 for details). PLUQI, which stands for 'Personalized and Localized Urban Quality Index' builds on a range of work on quality of life in cities, for example the OECD Better Life Index, but aims to illustrate how richer, more fine-grained data enables a more local approach to developing such indices.

In Year 1 of DaPaaS, a first version of PLUQI was developed and applied to data from South Korean cities. In Year 2, this was adapted to work with data about Scotland. In both cases, Grafter was used to prepare the necessary data.

For 'PLUQI Scotland', the data used comes from the Scottish Government, who make available data on many topics about a range of different geographical areas. The source data was provided in the form of CSV files, each containing information on a single 'indicator' for a range of areas and for one or more time periods.

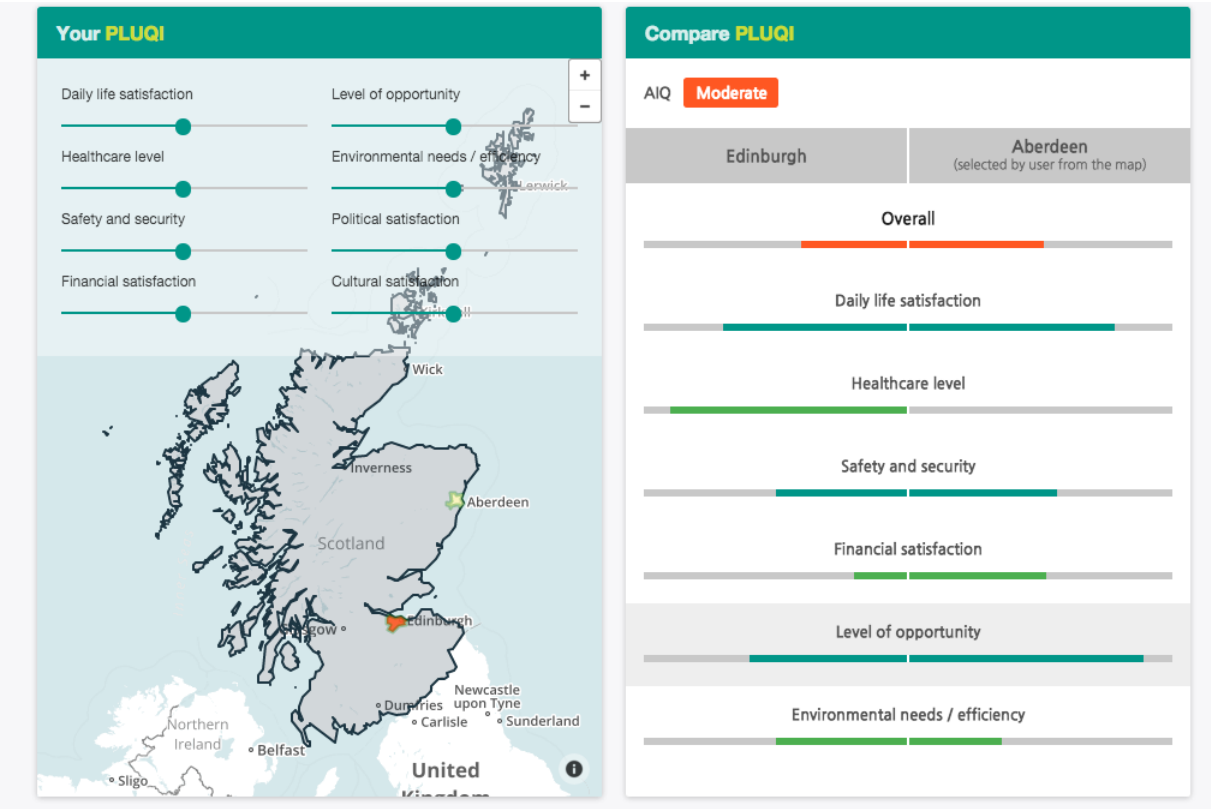


Figure 2 Comparing quality of life in two Scottish cities, using PLUQI

The Linked Data representation of the data made use of the RDF Data Cube Vocabulary[1], a W3C Recommendation designed for systematic representation of statistical data as Linked Data.

Although not all available data was made use of in the PLUQI application, the total quantity of data to be transformed was large. There were around 2500 CSV files to be transformed, resulting in approximately 1.2 billion triples. Therefore the process needed to be highly automated. Early versions of the transformation process exposed flaws in Grafter that led to excessive use of memory. Once corrected, it was possible to execute the transformation process against the full data collection in a reliable way. Due to the large quantity of data, it took a considerable amount of time to run: in the region of 24 hours on a medium power Amazon Web Services EC2 server, producing around 500GB of output when serialised to the RDF Turtle format.

3.2 Local government data

Grafter has been applied to generating a large number of Linked Data datasets for local government organisations in the UK, including Glasgow, Hampshire, Surrey and Manchester councils.

The source data for these datasets covered a range of file formats, including CSV, Excel, PostGIS (relational database with GIS extensions), and ESRI Shapefiles (a common GIS format).

The data incorporated reference data: for example geographical information, data on services, buildings, transport infrastructure and other fixed assets; and statistical or operational data, including population, economy, health, education, crime, spending and planning application data.

The purpose of the data publishing was partly for government transparency and accountability and partly in order to make the data more easily accessible for use in other applications and visualisations. Creation of 'area profiles' is a common requirement in local government: collecting and presenting data about a location of interest.

[1] http://www.w3.org/TR/vocab-data-cube/

Grafter proved itself capable of processing all the required input data types and as common patterns of data recurred frequently, it was possible to reduce significantly the overall development time by re-use and adaptation of existing Grafter transformations.

Some types of data appeared in very similar formats across more than one council organisation: including data on council spending and data on building planning applications.

In this case it was possible to take a transformation created for one council and apply it directly to data for another. This required that the starting CSV format was consistent between councils – the Local Government Association in the UK had carried out some work to define such CSV templates.

This highlighted the need for clear error reporting from Grafter. A Grafter transformation can validate that its inputs meet specified criteria: if the input is not in the 'shape' expected, then the transformation will generally fail and it is important to give clear feedback to the user to explain what was wrong with the input and allow them to correct it.

There may be possibilities in this context for Grafter to work alongside CSVLint[2], a tool developed by the Open Data Institute for validating a CSV file against a specified schema.

3.3 Scottish Government Statistics import pipeline

The Scottish Government is involved with Swirrl in a project to publish large quantities of statistical data as Linked Data. It is desired that statisticians with no knowledge of Linked Data or RDF are able to create Linked Data while complying with a number of agreed conventions for the choice of identifiers and vocabularies.

To enable this, an Excel template was developed, with a standard set of columns, allowing the values of statistical dimensions and associated measures to be created. The target user group for this tool are experienced with using Excel. They are easily capable of pre-processing the input data to match a defined set of columns in an Excel spreadsheet.

Grafter was used to create a transformation pipeline, starting from this generic Excel template, and generating data in RDF Data Cube format, together with supporting codelists represented using SKOS[3].

The pipeline makes use of separate data on the hierarchy of geographical areas represented in the statistics, in order to automatically aggregate data from small areas to the larger areas that contain them.

	A	B	C	D	E	F
1	GeographyCode	DateCode	Measuremer	Units	Value	Gender
2	S12000033	2014M06	Count	People	2058	All
3	S12000034	2014M06	Count	People	1203	All
4	S12000041	2014M06	Count	People	1408	All
5	S12000035	2014M06	Count	People	1068	All
6	S12000036	2014M06	Count	People	7989	All
7	S12000005	2014M06	Count	People	1337	All
8	S12000013	2014M06	Count	People	364	All
9	S12000006	2014M06	Count	People	2243	All
10	S12000042	2014M06	Count	People	4314	All
11	S12000008	2014M06	Count	People	3303	All
12	S12000045	2014M06	Count	People	934	All
13	S12000010	2014M06	Count	People	1428	All
14	S12000011	2014M06	Count	People	807	All

Figure 3 Example of spreadsheet following standard input format

3.4 Flemish Government Statistics

ProXML, a Belgian SME involved in the EU FP7 funded OpenCube project has applied Grafter to generating Linked Data on behalf of the Flemish Government statistics office. The process involved a

[2] http://csvlint.io/
[3] http://www.w3.org/2004/02/skos/

programmer and a Linked Data domain expert working together. The programmer developed a text based template where the domain expert could define the precise required form of the Linked Data to be generated. The starting point for the transformations was a collection of CSV format files that had been exported from a relational database.

The flexibility of Grafter and the ease with which transformations could be adapted and re-used meant that Grafter was evaluated favourably in comparison with alternative approaches for generating Linked Data, including R2RML[4] and TARQL[5].

3.5 Environmental data for TRAGSA

The EU FP7 project SmartOpenData[6] has created a linked data infrastructure for biodiversity and environment protection. The Spanish company, TRAGSA, one of the partners and project coordinator of the SmartOpenData project, with the help of SINTEF, tested Grafterizer as part of DataGraft to assess its applicability to their data transformation requirements.

They noted the following positive aspects of Grafterizer, as implemented in DataGraft:

- The ability to 'fork' (copy, then adapt) existing transformations easily allowed them to re-use existing transformations and so save time in the process of creating a new transformation
- The ability to edit the parameters of each transformation step interactively, and to change the order of steps, helped them to:
    - create transformations in general
    - detect and correct mistakes
    - experiment with different parameters for transformation steps
    - the ability to add utility functions with custom code provided flexibility in transformation design and the ability to re-use functions across different transformations

They identified some features not currently available in Grafterizer that would have made the tools more useful for them, notably the ability to join more than one input dataset and the ability to sort datasets. To overcome these limitations, it was necessary to carry out some pre-processing of the input files (e.g., for one transformation, 27 of the 43 files tested required some pre-processing).

It is possible to perform joins and to sort inside a Grafter pipeline, but the Grafter library itself does not provide support for these operations explicitly in a way that can be used by Grafterizer.

3.6 Sensor data from CITI-SENSE

The EU funded CITI-SENSE[7] project involves enabling citizens to set up large quantities of air quality sensors in a number of cities, thus creating a Citizen's Observatory.

The data collected from these sensors is shared in the form of one CSV file per sensor per hour, and is stored on a FTP server. Each file contained several measurements: concentration of various pollutants (Carbon Monoxide, Nitrogen Dioxide, Ozone) as well as temperature, pressure and noise measurements.

Grafter was used to access the FTP site and check for new measurements that had appeared since the previous check, then to transform those into Linked Data, using the RDF Data Cube Vocabulary. The data was hosted on DataGraft.

This was a useful proof-of-concept for use of Grafter in a Smart City and/or Internet of Things context. This is not truly 'streaming' data, but is a typical approach for distributing frequently updated sensor measurements. Grafter was able to retrieve and process the data quickly and this approach could be scaled up to larger numbers of sensors or more frequent updates.

[4] http://www.w3.org/TR/r2rml/
[5] https://tarql.github.io/
[6] http://www.smartopendata.eu
[7] http://www.citi-sense.eu/

3.7 Integration in other applications

For a commercial project for the UK Department for Communities and Local Government, Swirrl used the Grafter libraries to post-process tabular data retrieved from a linked data store in order to create Excel format downloads with a specific structure. This was required to support relatively large result sets – with up to 30 columns and up to 30,000 rows. Grafter proved very useful in this context, supporting the necessary methods and demonstrating good performance.

Activity on the Grafter mailing list is growing, showing that a number of people unconnected to DaPaaS are starting to use Grafter for their own projects. Examples of applications discussed on this list include integrating Grafter with Natural Language Processing tools for extracting structured data from text documents; and processing large quantities of environmental observations, centred around pollutants in the neighbourhood of an Italian steel plant.

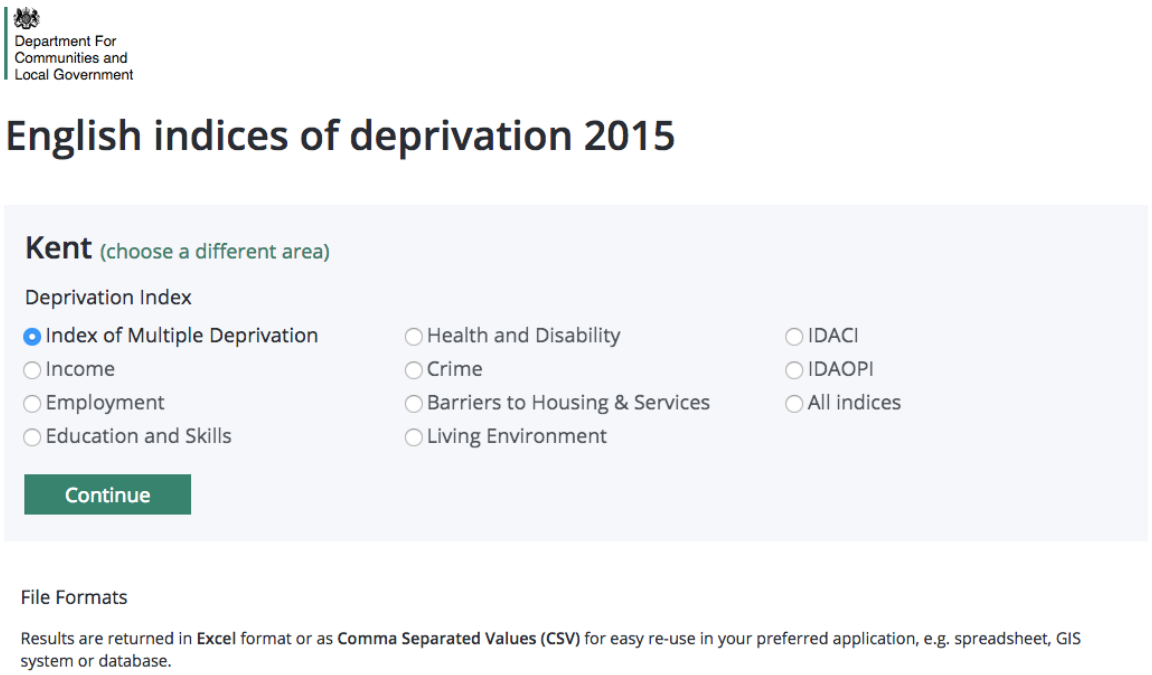


Figure 4 UK government application using Grafter to create user-defined extracts of a popular dataset

3.8 Lessons learned

The experience of applying Grafter and Grafterizer in practice has verified that the DaPaaS Methodology captures the most important aspects of the Linked Data publishing process, and that the tools created in DaPaaS are appropriate to supporting this workflow.

Grafter and Grafterizer have been applied successfully to transforming large quantities of data from a variety of sources into Linked Data, suitable for publication through DataGraft.

The core design concept has been verified in practice: that data transformation generally requires developers and data domain experts to work in partnership and that providing tools suitable for both groups is the best solution for efficiency.

Bulk processing of large quantities of data or frequent processing of rapidly changing data is often required and this needs automation and integration of data transformation processes into other software.

It has also been demonstrated that Grafter's design is suitable for supporting an interactive user interface, aimed at the 'knowledge worker' user group. Grafterizer has proved useful for a number of users and, as part of DataGraft, is currently being extended as part of the proDataMarket[8] project to support a sufficiently wide range of data functions to meet the needs of all users. Future enhancements

[8] http://prodatamarket.eu/

of Grafterizer will increase its capabilities in terms of 'built-in' functions, but the ability for a programmer to use the underlying Grafter libraries to create new custom functions is an important part of the overall solution.

Grafter demonstrates good performance for large data collections but the current deployment model of Grafterizer within DataGraft, where users can register and run arbitrary transformations, requires a 'sandboxed' security model, to ensure that untrusted code cannot compromise the security of the hosting environment. This currently places a limit on the maximum size of datasets that can be transformed via the hosted Grafterizer environment.

Defining a standard data input template and a corresponding Grafter transformation has proved another useful way to meet the needs of data owners without programming skills. This allows a knowledge worker to prepare their data using familiar tools such as Excel. Then the design choices made on how to represent the data as Linked Data are coded into the Grafter transformation process.

DataGraft supports the ability to 'fork' an existing Grafterizer transformation allowing a user to benefit from the work of others and adapt it for their own benefit. This has the potential to enable significant productivity increases for knowledge workers involved in data processing.

4 Recommendations for Further Work

The experience of testing Grafter and Grafterizer in practice has confirmed that the core concept and design is sound and has great potential. It has also identified areas where further work should be focused to enhance these tools. The most important areas are:

- Extend the library of standard functions available to Grafterizer to increase the range of transformations that can be created with it
- Work on the security sandbox model used in Grafterizer and DataGraft to increase the capacity to deal with large datasets while still ensuring security within a shared resources/'untrusted user' environment
- Improve the validation and error reporting mechanisms to assist users to understand the source of any problems in processing data, and precisely which part of the data needs attention
- Further development of the 'Domain Specific Language' supported by Grafter, essentially the set of functions that it enables out of the box: to extend the number of table manipulation functions available and to improve the overall structure and documentation of the API
- To standardise the metadata for a Grafter transformation pipeline to enable easier integration with other software
- To improve (and release as open source) the 'Grafter Server' software which supports the integration of Grafterizer and Grafter, enabling improved user interface support for previewing of data transformations.
- To enable import of ontologies into Grafter, reducing the effort required by a programmer to support standard ways of representing data.

SINTEF is currently extending Grafterizer, as part of DataGraft, to support the above mentioned features, and Swirrl will continue the development of Grafter as part of its commercial activities.