OPEN DATA SUPPORT

Assisting the publication of statistical Linked Data by the Digital Agenda Data tool

Contract No: 30-CE-0530965/00-17

Digital Agenda Data Tool on your desktop – How to

pwc

European Commission

Document Metadata

Property	Value
Release date	2015-10-23
Authors	Paul Massey – TenForce Bert Van Nuffelen – TenForce Nikolaos Loutas – PwC EU Services

Disclaimers

The views expressed in this report are purely those of the authors and may not, in any circumstances, be interpreted as stating an official position of the European Commission. The European Commission does not guarantee the accuracy of the information included in this presentation, nor does it accept any responsibility for any use thereof. Reference herein to any specific products, specifications, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favouring by the European Commission.
All care has been taken by the author to ensure that s/he has obtained, where necessary, permission to use any parts of manuscripts including illustrations, maps, and graphs, on which intellectual property rights already exist from the titular holder(s) of such rights or from her/his or their legal representative.

This report has been carefully compiled by PwC, but no representation is made or warranty given (either express or implied) as to the completeness or accuracy of the information it contains. PwC is not liable for the information in this presentation or any decision or consequence based on the use of it.. PwC will not be liable for any damages arising from the use of the information contained in this presentation. The information contained in this presentation is of a general nature and is solely for guidance on matters of general interest. This presentation is not a substitute for professional advice on any particular matter. No reader should act on the basis of any matter contained in this publication without considering appropriate professional advice.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Context and scope

The “Digital Agenda Data tool on your desktop” tool has been developed in the context of SMART 2012/0107 'Provision of services for the Publication, Access and Reuse of Open Public Data across the European Union, through existing open data portals' funded under Contract No. 30-CE-0530965/00-17 (henceforth referred to as Open Data Support).

Statistical data is a frequently produced to represent a state of a governmental, institutional or research investigation. In order to allow these figures to be compared or contrasted, they need to be consolidated and management in a common format, so that tools can be built to visualisation the information in a form which will make sense to the information consumer.

In Open Data Support, the need for providing a tool both for training and operational purposes was identified. To this end, we considered the reuse of the Digital Agenda Data tool (DAD) [1], which has been developed for the Digital Agenda Scoreboard[2] of DG CONNECT. The Digital Agenda Scoreboard publishes the European indicators on how digital Europe is. The DAD, which drives the Digital Agenda Scoreboard, is based up the Data Cube vocabulary. In this task, a locally deployable instance of the DAD was created to evaluate the tool.

The selection of the DAD for this task is motivated by the following:

- the DAD is being used in practice and supporting a working statistical data portal;
- the DAD is built upon open source technology and comes with an deployment manual;
- the locally deployable instance results in an easy to activate demo setup which makes the publishing process more tangible, which addresses the main challenge encountered in Task 1.3 *Publishing of Reference Datasets;*
- the locally deployable instance is also a positive outcome for the owners of the DAD:
    - the deployment instructions get validated by an external users they can better evaluate and test further directions.

[1] http://digital-agenda-data.eu/documentation
[2] http://digital-agenda-data.eu/

With this activity we strengthen the statistical Linked Data ecosystem. By means of the local deployment of the DAD tool, owners of statistical data are able to experiment and see in a practice how their data can be published as linked data cubes.

1.2 Structure

This technical report is structured as follows:

Chapter 2 presents the architecture and the components of the DAD;
Chapter 3 describes the installation process of the DAD;
Chapter 4 describes how to import a statistical dataset into DAD;
Chapter 5 describes how to create visualisations of datasets published on DAD;
Chapter 6 discusses experiences encountered during the deployment and use of DAD.

2 DAD architecture and components

2.1 Introduction

The Digital Agenda Scoreboard (DAS) is aimed at aggregating, storing and visualising European indicators on how digital Europe is. The system managing and publishing the data is called the Digital Agenda Data tool (DAD). It is made up of four main open-source software components, the installation of which is described in [**Error! Reference source not found.**]. The four main components being:

- *Content Repository*[3] (CR) for the data storage and browsing of the datasets and available metadata. The CR is the main component and this provides the functions required to maintain and browse the statistical data files. Import functions are also provided by the CR (via predefined excel files) which then converts the uploaded spreadsheets into RDF format.

- *Plone*[4] is a python based content management system (CMS) used to provide the front-end website for visualising the statistical data imported via the CR. Plone runs on top of Zope and has the facilities to theme the web-site, manage add-ons, user accounts, etc. via a web-based interface[5].

- *Elda*[6] providing an implementation of the Linked Data API. This is a java application which uses jetty as the application container.

- *Virtuoso*[7] is used as the RDF data store.

DAD is extensively documented[8]. The documents which are relevant for this work are:

- "The deployment guide" (PDF), which describes the current configuration and installation of the DAD,
- "Guide to preparing dataset for uploading" (PDF), with the associated:
    - Empty spreadsheet templates for the observations, metadata and the code lists (Excel files)

---

[3] https://github.com/tripledev/scoreboard.contreg

[4] https://plone.org

[5] There does appear to be a command line interface described for 5.0, but for 4.3 the documentation is missing. The documentation for 5.0 seems to suggest it will be difficult to use because of the expected interaction workflow ("plone code is ugly and expects a HTTP session, which needs to be mimicked…"). It's not clear from the documentation whether all the site modifications can be achieved via discrete commands which could be put in a source control system such as github (to be replayed as required). The document suggests that the database needs to be managed (and maintained and moved forward). Plone uses buildout which is a python facility used to download and install everything required in an isolated fashion.

[6] https://code.google.com/p/elda/

[7] http://virtuoso.openlinksw.com/

[8] http://digital-agenda-data.eu/documentation/

- “Guide for uploading datasets”, etc.
- Technical report[9] providing details of the chart making possibilities.

The main documentation of interest is to guide to “preparing a dataset for uploading”[10]. All the components are open-source ones and typically have much more extensive documentation available on the relevant community portals or websites.

2.2 System architecture

The DAD architecture consists of the common test and production environment setups. Replicating the complete environment was not required (although it could have been) for this test, so only the production environment was recreated (as it would have to be anyway). The main requirement here was to make the production environment look as close as possible to the original system, but on an isolated machine (see section 3.4 for details of the simple approach taken in this work).

[9] http://digital-agenda-data.eu/documentation/technical-report-m30
[10] http://digital-agenda-data.eu/documentation/data-preparation-instructions

3 DAD Installation

3.1 Introduction

In order to run the DAD system, a potential user has to provision a local machine and follow the deployment instructions. This requires technical knowledge and experience, and often during the process problems arise. Since it is our intension to provide a data publishing setup to the business owners, the technical hurdles have to be as minimal as possible.

For this reason a **vagrant**[11]-based approach is taken. Vagrant is an approach to describe a complete machine setup using executable scripts. From a base machine (centos 6.3), the DAD is bootstrapped with everything necessary to perform the build and install. Vagrant usage is described more completely in the next section, and the description is available on github[12], along with some sample files.

3.2 Vagrant Installation

Vagrant is a container technology, which has a workflow for managing virtual machines. The "**Vagrantfile"** description indicates how the underlying (host) system will see the virtual machine (guest) in terms of windows, networking, access to the underlying host file system, as well as what base software should be installed on the guest system. This approach is heavier than some of the other container based approaches in that a complete VM is created. Typically, the *vagrantfile* description will indicate a script which will be used during provisioning to add software to the basic image which is being used, the **bootstrap script**. This can be such things as editors, compiler and desktop software such as web browsers (e.g. firefox).

The bootstrap scripts follow the deployment guide, but with the exception that *all* updates to configuration files as requested in the deployment guide are made on copied of the configuration file and then moved across to the correct place by the build process. The scripts were created based on the instructions given in the deployment guide. This required downloading, installing and configuring all the components indicated in the deployment manual[13].

These vagrant-based machines are aimed at development on a local machine and are explicitly not intended for deployment on an externally accessible machine (i.e. accessible via the Internet). The main advantage of this approach is that the vagrant provisioned machine is a live, complete and accurate description of a working deployment (anything missed in the bootstrap script will lead to a failing deployment). Additionally, because the

[11] https://www.vagrantup.com
[12] https://github.com/tenforce/vagrant-digital-agenda-scoreboard.git
[13] http://digital-agenda-data.eu/documentation/deployment-manual

machine description is textual rather than binary it can be stored in a source code system (i.e. github, etc.), Vagrant usage as a very simple basic workflow consisting of a number of simply commands, these being:

- **vagrant up -** which will use the *vagrantfile* in the current directory to start the machine if already provisions, or the first time to build and provision the VM,
- **vagrant halt -** which will stop the currently running VM, and
- **vagrant destroy -f** which will delete the VM which was previously created and provisioned (other files will be left are they where)

The vagrant description[14] will recreate the production environment (not the test environment which was not required for the testing being done here).

3.3 Basic VM requirements

DAD requires the following base main items be installed:

- Centos 6.3 (Initial vagrant box chosen)
- Tomcat 6.0.44
- Java 1.6
- Python 2.7.8

Some of these components have now been superseded by newer versions or are approaching end-of-support (Tomcat 6 is end of 2015[15]). CentOS 6.3[16] is supported till 2020, but has now been superseded by CentOS 7.0. Java 7 is no longer supported by Oracle for security updates but version 6 of the openjdk is still supported (even though it is now classed as legacy, fixes will be backported but possibly with an undefined delay). Python 3.0 will replace the 2.7.8 version. The DAD vagrant bootscript has been divided into three separate scripts (all controlled via **scripts/setup.sh**):

- **scripts/das-install.sh** will install the default expected software,
- **scripts/build.sh** will download, compile/build and install the main DAD software components,
- **scripts/processes.sh** which will (re)start the DAD services required.

[14] https://github.com/tenforce/vagrant-digital-agenda-scoreboard.git
[15] https://tomcat.apache.org/tomcat-60-eol.html
[16] https://www.centos.org/

Calling the scripts/setup.sh script with an -S option will startup only services required (i.e. tomcat, elda, virtuoso, etc.). This is the main expected route when the uploading of data files and visualisation testing is be used.

The main script **scripts/build.sh** follows the deployment guide and is divided into separate **bash** functions, once for each of the main components, thus:

- install_cr –installs all the CR software, overwriting the configuration files as needed. This also downloads the apache-tomcat version required and unpacking. The scoreboard java code is cloned from the github repository and built as needed (this will also download other software).

- install_virtuoso –installs and sets up virtuoso, the replacement database is downloaded[17], unpacked and placed in the correct location for virtuoso (which has to be stopped before and restarted after the copy operation).

- install_scoreboard – this installs the Plone 4.3/Zope/etc. parts of the system required for managing the user interaction with the web application. This also downloads the plone-storage tar file required[18] and places it in the buildout location indicated in the deployment guide.

- install_elda –downloads the standalone version of the elda server and will place the scoreboard specific files where required (updating the jetty configuration file, etc.).

It should be noted that the DAD software is installed, but the services are not installed and have to be restarted each time the machine is restarted (up). This was deliberate since the intention was also to attempt to change some of the look & feel.

3.4 Network setup

The ambition is to emulate a fully deployed system: without being connected to the internet the vagrant machine should act as if the system is running in production. This can be achieved by setting the configuration of the network emulating the production environment.

In network terminology, the *localhost*[19] (the name referring to this machine, having IP-address 127.0.0.1) is used to access web applications that are running on that machine. From the outside the same web-application is accessible using a public name. Resolving names to machine IP-addresses and the services on those machines is done via DNS[20] and proxies. Configuring these rules is an essential aspect in achieving a complete Linked Data experience.

---

[17] http://test.digital-agenda-data.eu/download/virtuoso_copy.db.gz
[18] http://digital-agenda-data.eu/download/plone-storage.tar.gz
[19] https://en.wikipedia.org/wiki/Localhost
[20] https://en.wikipedia.org/wiki/Domain_Name_System

For instance, to make the following URI – representing an observation - redirecting to the right service in the vagrant machine

http://semantic.digital-agenda-data.eu/data/CyprusTestData1/pureagr_holdings/cy_ammochostos/nbr_holding/CY/2013

a combination of domain name rewrite rules (the /etc/hosts/) and proxy rules (see /etc/httpd/conf.d/) forward it to

http://127.0.0.1:8080/data/CyprusTestData1/pureagr_holdings/cy_ammochostos/nbr_holding/CY/2013

The vagrant setup contains a network (re)configuration realizing the above. So browsing www.digital-agenda-data.eu and the others domains supported by DAD will be redirected to localhost, rather than the external system production system. The host redirection can be tested using such things as *ping* on the guest environment[21].

3.5 Starting the VM

As previously indicate, the VM is started with:

vagrant up

This requires significant time (in the order of 30-40 minutes). In order to bootstrap the deployment of the DAD system a copy of the **virtuoso_db** is required. This database is downloaded as part of the installation process. Once the initialization has completed, there are still some permissions issues which need to be manually reconfigured in order for the DAD system to operate correctly. CR2USER and CR2ROUSER - have no database role assigned in the downloaded DB and these have to be reconfigured manually using the virtuoso conductor interface (http://localhost:8890/conductor). The following steps are required:

- Login with the dba user rights (password required) to the virtuoso conductor
- Got to the “System Admin > User Accouts” tab
- Select the CR2USER Edit link
- Change the “Primary Role” to “dba”
- In “Account Roles” move all rights, except for the “nogroup” to the selected list
- Finally, Save the updates and repeat this for the CR2ROUSER user account.

[21] This is the last thing the boot scripts do, since after this point it will no longer to easy to access the original sites.

Once the permissions have been updated, it is necessary to reboot the VM (so that the connections become stable again and the network connections are finalised). The next time the VM is started with *vagrant up* the DAD services will need to be restarted using (in a terminal with the –S operation to indicate **S**ervices):

(cd /vagrant ; sudo scripts/setup -S)

4 Publishing a dataset

4.1 Introduction

This section describes how to insert a statistical dataset into the DAD tool. We will illustrate the process with an example taken from the national statistical office of Cyprus.

A bird's eye view on the process's steps follows:

- first, the creation of a tabular representation of the data according to the DAD template, then
- the upload of the data in the semantic repository – this will transform the data into RDF Data Cube, and
- finally the visualization of the data.

This sequence is analogous to the one described in Deliverable D1.3.1-2. The key difference is the fixed tabular representation. Whereas the generic tools in Deliverable D1.3.1-2 impose few constraints on the structure of the table, those of the DAD have a well-defined format. The observations have well defined number of dimensions (year, country, variable, breakdown, unit) with a single measure value. For each of the dimensions a metadata sheet needs to be provided with all possible values. An additional sheet records the sources from where the data is being collected. The rigid format ensures a coherent approach which simplifies the publishing of the data and the user interface of the visualizations.

4.1.1 Example dataset

An existing statistical dataset from Cyprus for 2013[22] is chosen. The CSV form of the Excel file is:

Table 1: Example data from Cyprus

HOLDINGS AND UTILIZED AGRICULTURAL AREA BY TYPE AND								
HOLDERÂ´S DISTRICT OF RESIDENCE	2013							
DISTRICT	TOTAL	TOTAL	AGRICULTURALAND LIVESTOCK	AGRICULTURALAND LIVESTOCK	PURE AGRICULTURAL HOLDING	PURE AGRICULTURAL HOLDING	PURE LIVESTOCK HOLDING	PURE LIVESTOCK HOLDING
	Number of	Areas (decares)	Number of Holdings	Areas (decares)	Number of Holdings	Areas (decares)	Number of Holdings	Areas (decares)

[22] http://www.mof.gov.cy/mof/cystat/statistics.nsf/All/D16AEE8941F01177C2257132002C25B2/$file/EMISSIONS_OF_GREENHOUSE_GASES_A1990_12-EN-180914.xls?OpenElement

	Holdings							
TOTAL	35385	1093323	3708	361134	31403	732170	274	19
LEFKOSIA	12447	353703	877	101206	11512	252494	58	3
AMMOCHOSTOS	2167	84621	389	25409	1737	59208	41	4
LARNAKA	5276	302648	1002	122086	4186	180557	88	5
LEMESOS	9110	149640	701	36649	8365	112989	44	2
PAFOS	6385	202711	739	75784	5603	126922	43	5

4.2 Preparing the dataset

The preparation of the dataset for uploading it to the tool is a two-step process:

1. First the metadata description file for the observations needs to be created and needs to be uploaded as individual spreadsheets, and

2. The observation values need to be encoded in the template provided using the metadata encoded descriptions and uploaded.

4.3 Loading data via CSV files

The CR, provides the functionality to import data from CSV files, including Microsoft Excel, in a templated fashion. The CSV files have a pre-defined structure which must be followed as way as a defined way of loading the files[23].

4.4 Creating the Meta data excel file

There is a template provided for the metadata which has the general form:

- notation - the identifier to be used as a reference to this piece of meta data. The actual form can be recognised as having a pre-defined meaning (i.e. pc_x will reference observations which are expected to be percentages).

- prefLabel - label

- altLabel - along with the prefLabel these fields are used in the interface to show the values (inplace of the notation identifier).

[23] Loading data from a Microsoft Access Data base is also documented, but has not been considered further.

- definition - with notes, member-of, order and source indicate other items of information as required.

The template provides contains all the spreadsheets in one spreadsheet:

- Indicator - which details the variables in the observations and links the indicators together using the *indicator group*,
- Indicator Group - which is a grouping of the *indicators* used.
- Breakdown - country region, etc.
- Breakdown Group - grouping of regions, etc.
- Unit - details the unit type (no_holdings, decares, etc).
- Source - where the *indicator* or *breakdown group* came from.

These are all detailed in the guide "how to prepare the datasets"[24]. These provide a means of mapping variables, units and other code lists to their descriptions. There are specific 'notational forms' for some of the variables (e.g. pc_variable refers to a percentage value). When uploading the data files the following need to be considered:

1. When uploading a newly created meta-file the previous values will be *added* to the existing metadata descriptions, but there is a check box which will delete all other values before adding the new ones (don't do this[25]).
2. There does not appear to be a selective clean-up of the data when uploading the template files so updates need to be done carefully.

There is a link which allows the code lists to be browsed and which will also allow them to be exported as excel. The volume of the metadata is such that the lists are each exported as separate excel files (rather than the format using in the template).

4.5 Uploading the metadata file

This is a multi-step operation since each of the metadata excel template spread-sheets has to be loaded separately. The form to do this is found in the http://www.digital-agenda-data.eu/data under the Admin actions link.

[24] http://digital-agenda-data.eu/documentation/data-preparation-instructions

[25] All the meta data items for the category will be removed, not just the previously loaded meta data file (so if the meta data file is not globally complete, some labels, etc. will be lost.

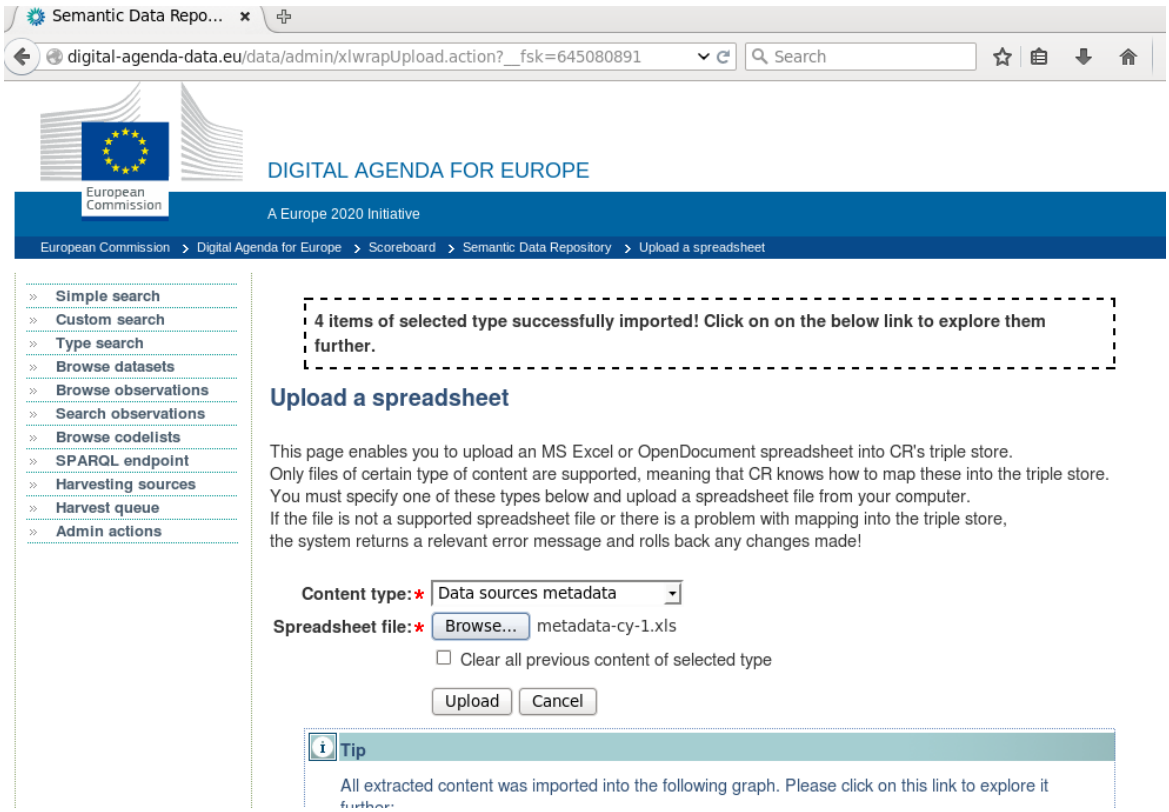

Figure 1: Uploading the Meta Data Sheets

Note: There really needs to be a link to load the *complete* excel template file as provided rather than having to edit the excel file and save it for each spreadsheet to be uploaded. Once the metadata has been loaded, it can be browsed (using the CR component).

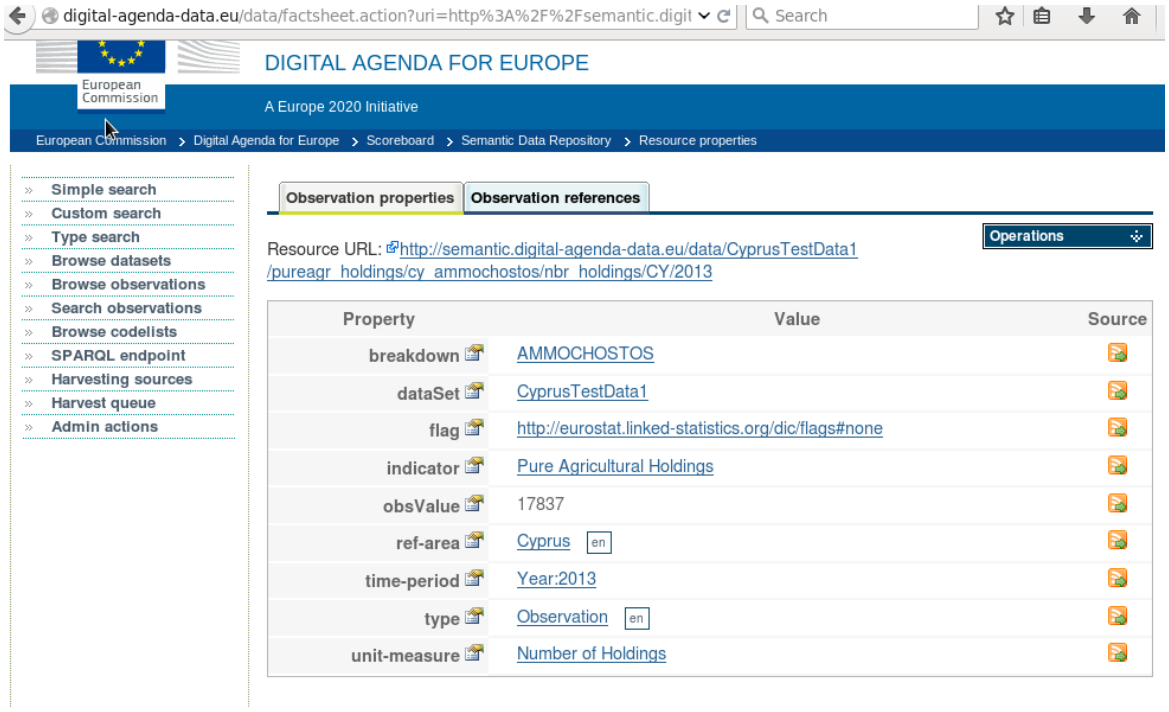

Figure 2: Browsing the Observations

4.6 The Observation Template

Once the metadata descriptions have been created and uploaded. The Observation description can be created, using a predefined observation template[26], which describes a table where each row will describe a single observation data point. The main columns are:

- Year - which is a code,
- Country - which will be the country code (CY in this case for Cyprus),
- Variable - the question which is being asked,
- Brkdown – breakdown of the variable according to aspect: e.g. age, profession, etc.
- Unit - the unit notation for indicating the type of the observation value,
- Value - the observation value which will be used,

The variable, brkdown, unit, and year must all be identifiers as found in the metadata excel files. The same form for uploading the metadata template file is used to upload the observations.

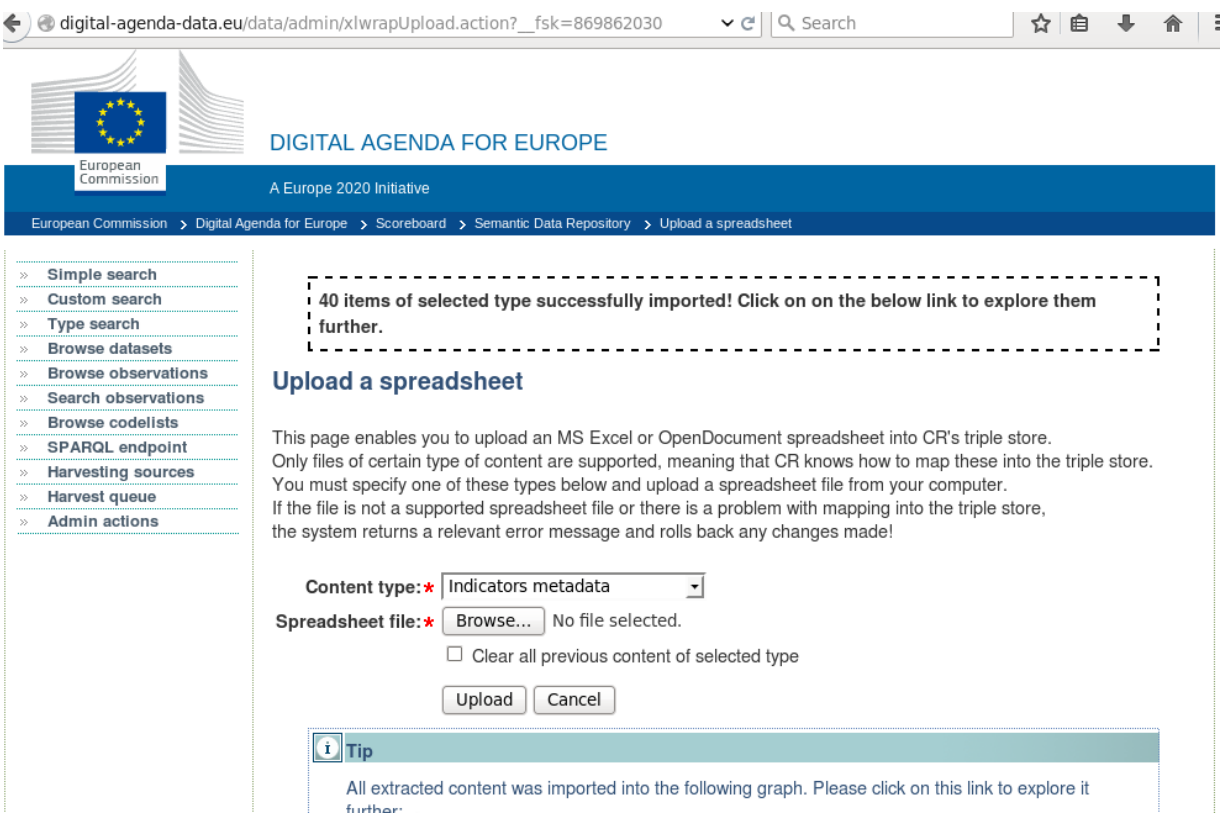

Figure 3: Observations Uploaded

[26] http://digital-agenda-data.eu/documentation/observations-template

Once the uploaded observations are available (the above figure shows that the 40 rows of observations in the sample have been uploaded correctly), then the Data Cube observations can also be browsed and this is shown in the next figure.

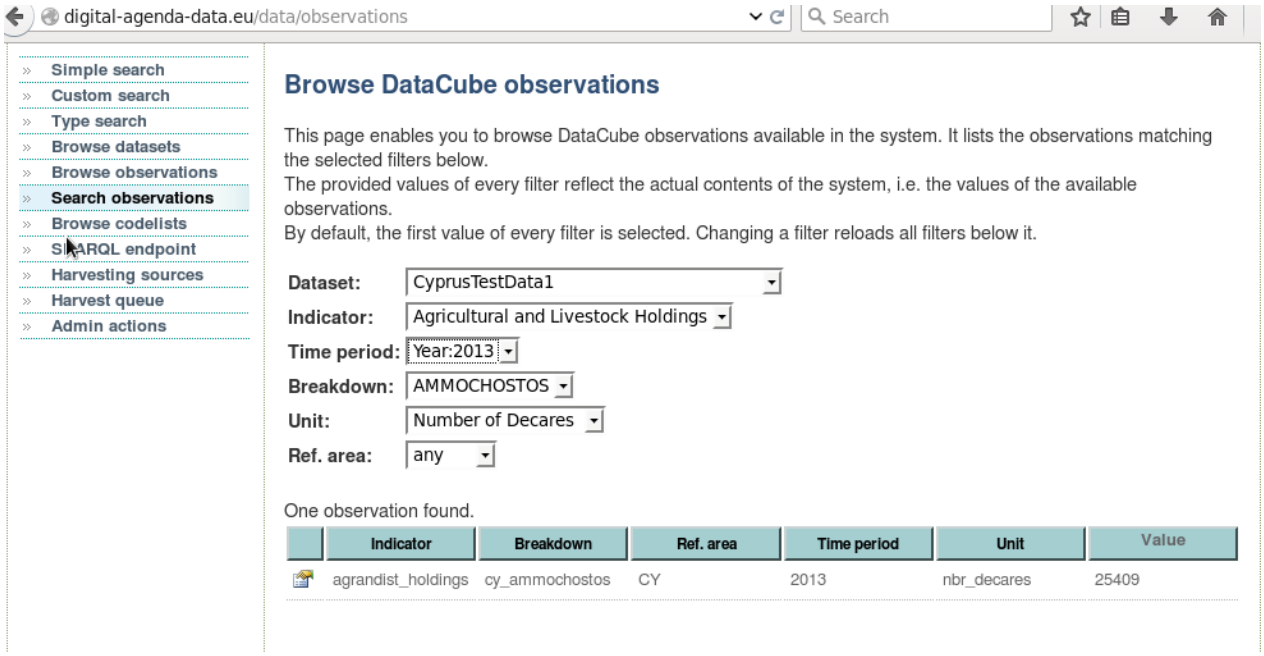

Figure 4: Data Cube Observation Browsing

Browsing the uploaded data allows the verification of whether the excel files are complete and uploading has been correctly done (fields show what is expected).

Note: Given that the data to be converted into the Observations Excel file is often already in a tabular format which just needs mapping. The user of text based tools such as ***awk***[27] should be considered – the alternative could well be a lot of error-prone typing just to convert one table into another.

[27] http://www.gnu.org/software/gawk/manual/gawk.html

5 Visualisation of the data

5.1 Introduction

Once the sample data was converted into the required format and uploaded, the next test is to visualise it using the built-in functionality of DAD.

5.2 Chart Visualisations

Given the type of the data, regions and values in a single year, bar charts were used to visualise the regions and the associated values.

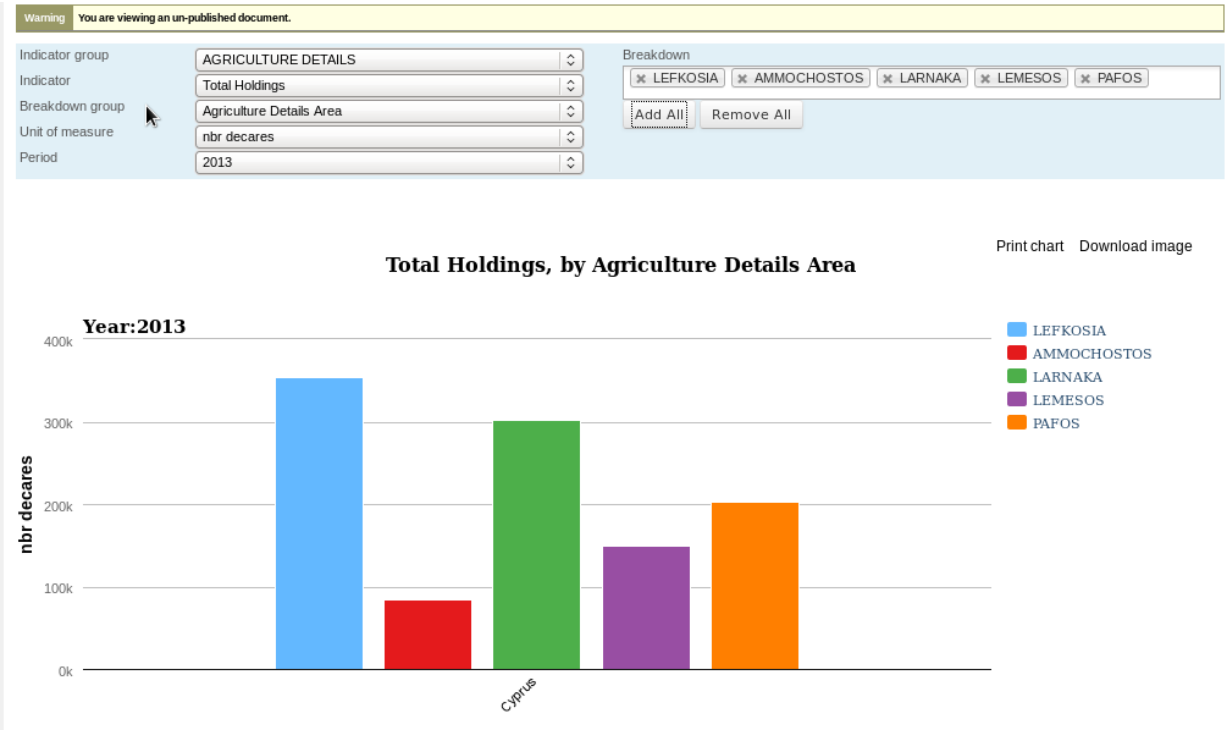

Figure 5: Chart Visualisation Example 1

The figure below is a variant of the above, but selecting livestock holdings rather than looking at the total holding values.

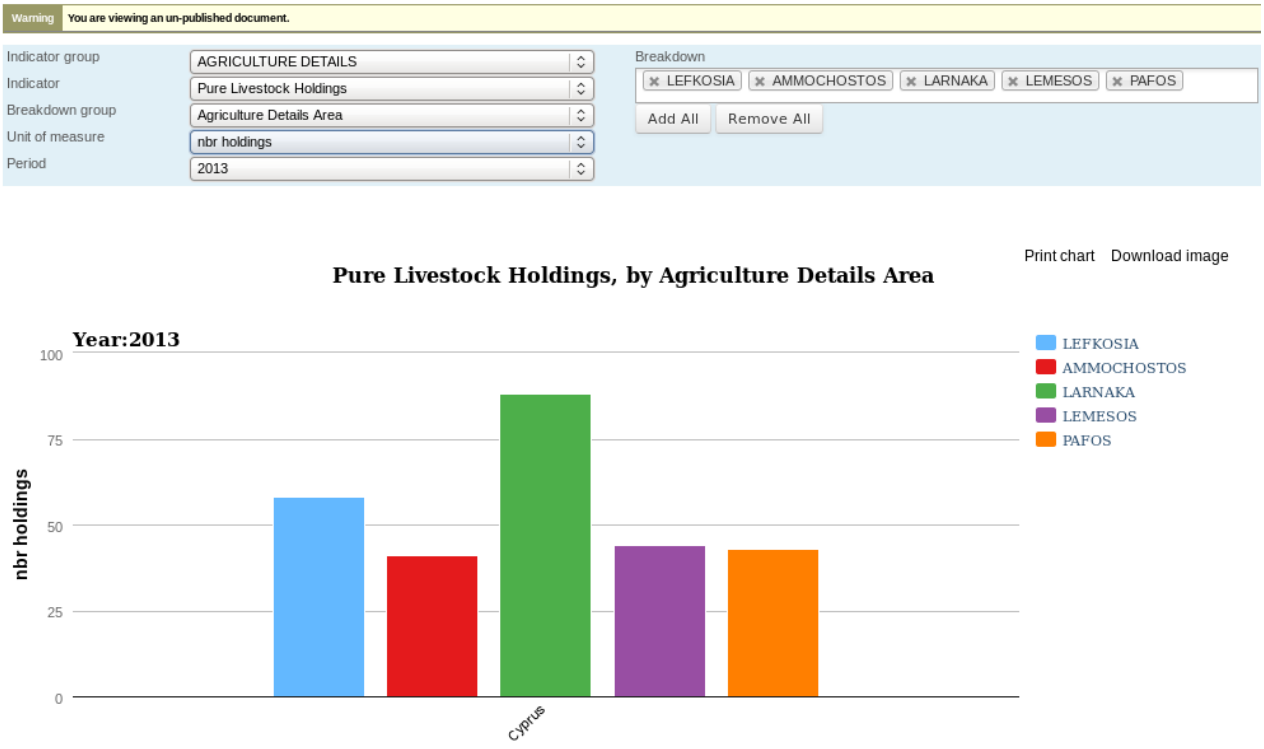

Figure 6: Chart Visualisation Example 2

5.3 Other Visualisations

The following example is taken from the Cyprus statistical data website concerning emissions of greenhouse gases[28]. This example was selected because it had observations spanning across a number of years and because it had a larger number of individual observation values[29].

The example shows a time series based chart, where the output of type of greenhouse gases is the item of interest. The listing of the items in the table and the selection boxes can be specified in the metadata description of the question variables.

[28] http://www.mof.gov.cy/mof/cystat/statistics.nsf/All/D16AEE8941F01177C2257132002C25B2/$file/EMISSIONS_OF_GREENHOUSE_GASES_A1990_12-EN-180914.xls?OpenElement

[29] The conversion of the original CSV file to the DAD observation CSV format was a 2-3 line call to *awk* (each row becoming several rows in the DAD output). Awk is the perfect tool for this sort of text based record conversion.

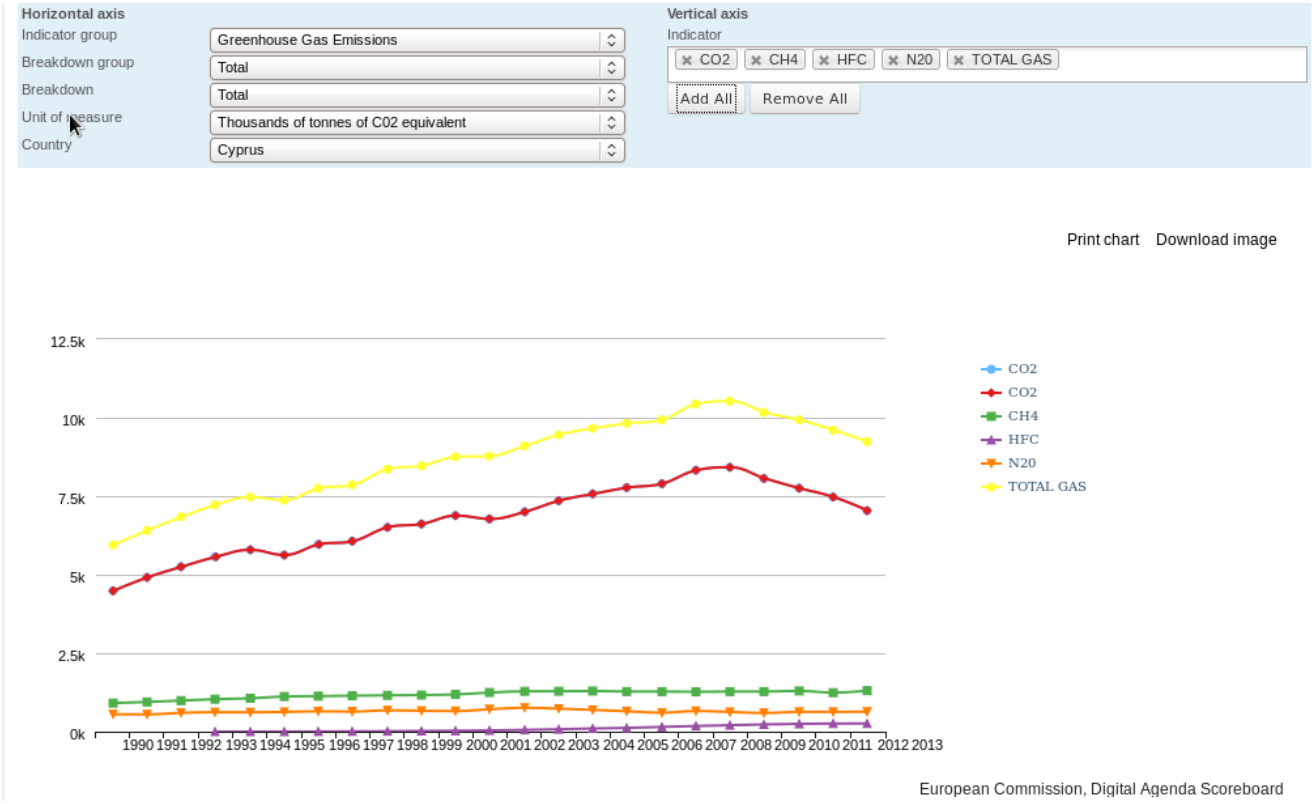

Figure 7: Cyprus Greenhouse Gas Emissions

6 DAD as publishing platform

This section details the key lessons which have been learnt during this work on turning an existing system into a locally deployable instance, complemented with lessons learnt about its use, acquired during the creation of a demo scenario. These lessons concern:

- the installation of the DAD software in a separate environment,
- how the DAD software operates for importing example data,
- how the DAD visualisation component works.

6.1 Installation of DAD components

The "deployment guide" described the production/testing setup as used by the Commission. This is exactly what is expected in this situation in that the current deployment situation has to be maintained, databases managed, etc. The vagrant-based installation of DAD now works, but the following points should be noted:

1. When the deployment guide was initially followed, it indicated a possibility to rebuild the databases from scratch (for the virtuoso db and plone installations). This approach did not work and it was necessary to use a copy of the two main component databases:
    - The virtuoso database had to be downloaded (for the CR). The downloaded virtuoso database replaced the empty database initially created. This indicates a potential problem when deploying at second site – making sure the database is clean for a new instance.
    - Plone storage was also downloaded (for the Visualisation) and added to the scoreboard virtual environment.

    This need to use existing database(s) is at odds with the intended Vagrant intention of rebuilding the VM and installed applications from basic source building blocks (preferably in some form of longer term storage such as github).
2. Wrong git repository was used which indicated that the following changes might be of interest:
    - Log-files should have had a system/git origin line in them,
    - Version documentation should also have been included in the generated logs.
3. It should be possible to rebuild from scratch to DAD system (without any bootstrap binary databases),
    - All the default meta data should be loaded from the excel files during the build process,
    - The build system should use discrete calls to upload the required excel files.

4. Sometimes the installation fails due to a network accesses, this is not really gracefully handled (some parts give no results in configuration isn't just correct, e.g. Elda isn't configured properly).

The question of database scheme migration has been handled via the use of *Liquibase*[30], which is a source control system for the database schema.

6.2 Uploading or Importing Data

Uploading the metadata and observation files are necessarily an iterative operation (simply because the excel files will or can be edited by hand).

- When uploading the metadata files, if a mistake is made and the values which have been uploaded (or cleaned up) need to be cleaned up, there is no obvious way to do it (the form allows clean all metadata of a specific type, not just what the user has previously added). Database has to be backed-up before and that can be restored (not practical in some ways).

- The metadata template is distributed as a single file, but the uploading action is per sheet (which means multiple edits of the metadata file and possibly a missed sheet upload as a result).

- In some cases the observations file was uploaded with typo in the indicator reference (no2 rather than n20), which should have been signalled as an input error (the data set could check for unknown references).

Using the browsing functions for quality control of the input data would be a very time-consuming affair.

6.3 Visualising the imported data files

This section details lessons which have been learnt concerning the visualisation of the data cubes component of DAD:

- The creation of a chart is a multi-step operation, but it not so obvious from the interface what are the various steps required (first a data set is created, then a cube is selected, then a chart, etc.). It would be cleaner if this workflow was more visible (e.g. Edit/Configure buttons – what is the difference? Publish is normally part of configure…).

- Selecting the Chart is not an easy operation – multiple values can be selected and then typically an error message is returned when trying to view the chart. There is no indication of how to fix the error or

[30] http://www.liquibase.org

problem (maybe using a constraint or rule engine would allow more direct feedback when selecting the charts available).

- Sometimes the interface did not show anything when waiting for a response from one of the other components (it should indicate what is going on).

Most of these issues concern the feedback which is provided to the user of the system while creating the chart. However, once the user gained a better understanding of the chart creation part of the system, these problems will be reduced but not eliminated.

6.4 Extension Suggestions

- End-of-life components should be replaced with newer versions (this could be easily tested by updating the vagrant description to the latest CentOS and trying to rebuild). With some automated component level testing, the status of the port could be automatically checked by regularly rebuilding the VM.

- Each service might be better isolated by converting them into individual Docker[31] service descriptions[32]. This would mean:

    - Installation script would have to be update to the docker services at boot-time,

    - Updates for newer versions of the components would cause minimal interference (e.g. movement to Java8 would only impact on those components ready for such a move).

    - logging of all components should be centralised (otherwise the dockers would fail and might reinitialise the log files).

    - Each component could have an individual test facility, make sure that the component is correctly installed and returning the expected results for fixed queries.

- Any major fail points should record sufficient information to correct the configuration problem (if that is what it is).

- Errors at the interface should be more component based so the eventual user error report will mean something.

- ELDA includes jetty as a webserver, while tomcat is already used by the CR. It would reduce the system size if it was possible to reuse the container installations.

---

[31] Dockers are a light-weight container approach which runs with a GNU/Linux system.

[32] It should be noted that some of the components have docker descriptions already available (but these were not used in the deployment guide).

- In the interface there are references to hosts (which are not under the DAD control (ec.*). This makes it difficult to understand when transitioning from one application "part" to another (logins do not seem to be managed globally - maybe the configuration flag changes that, but the situation should be passed to the user).

- The creation of the selection form appears to be very computational intensive - it needs to be improved or recomputed at another point.

- When importing the metadata spreadsheet, the "whole" excel template should be processed as a unit (not just the first sub-sheet). The individual sheets should be named – the default template should consist of the current expected list.

6.5 Conclusions

DAD is a well-developed application, but as with many multi-component systems all the components have to be correctly functioning for it to be operationally stable. The tool deployment looks straight-forward in the deployment guide, but the versions have since moved and the deployment guide will always be out-of-date compared to the actual system.

Areas which would benefit from further work are those relating to indicating more clearly what the users should see when one of the components is taken offline or fails. Indicating what is not working, when the installation is not complete or there is a problem of some sort of other (Java exceptions indicate a problem, but not where to look for a solution) - e.g. a JDBC driver errors will often indicate "failed to connect" but will not always indicate why it failed to connect or to what it was trying to connect.

The vagrant-based installation works, but as it stands today, the setup will be short lived as the many of the used software versions are indicated end-of-life (e.g. centos, tomcat, also database access points move or disappear, etc.). The easiest way to reduce this fragility would be reduce the number of components required and update the components to the most recent versions. The uploading, maintenance and virtualisation compiled can probably be reused, but isolating the required parts and making them robust could take a considerable amount of time and diverse skill sets (given that the CR is written in python while the scoreboard/elda are written in java which will necessitate different development methods).

The work could only be realized by the support of the DAD support team. They provided assistance and clarifications to the setup process. The collaboration resulted in an even better documented system. This case confirms that a combination of open source software and professional support around it, can lead to a benefit for

the general public and the running platform itself. The work done in this task has been recorded in the Final Report of the Digital Agenda Scoreboard[33] (section 3.1, p. 6).

Before this task, the potential evaluation of the DAD by interested users required a serious amount of effort. Today, using the vagrant, the assessment can start within a day.

[33] http://digital-agenda-data.eu/documentation/r4_final_report