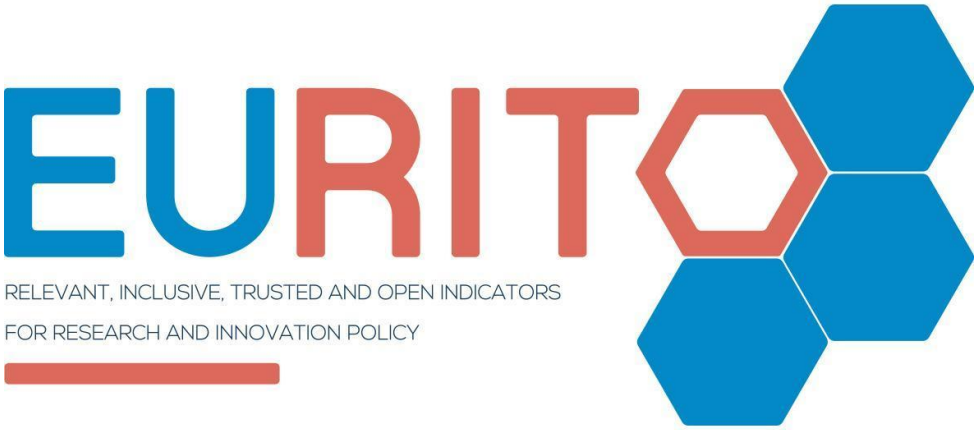

D3.1 Design of Data Collection Phase

Deliverable No.		D3.1	
Workpackage No.	3	**Workpackage Title**	Data Collection and Analysis
Lead beneficiary		DTU	
Dissemination level		Public	
Type		Report	
Due Date		M14 (28 February 2019)	
Version No.		0.5	
Submission Date		28 February 2019	
File Name		D3.1 Design of Data Collection Phase	
Project Duration		36 Months	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 770420.

Version Control

Version	Date	Author	Notes
0.1	2 April 2018	Nesta	Template Creation
0.2	13 Dec 2018	DTU	Definition of overall document structure
0.3	28 Jan 2019	DTU	Main points elaborated
0.4	18 Feb 2019	DTU, Nesta	Submitted for internal review with Partners
0.5	26 Feb 2019	DTU, Nesta	Internal review completed

Reviewers List

Version	Date	Reviewers	Notes
0.3.1		Alberto Abella	Contributions on phases for data ingestions and normalisation
0.3.2			

Disclaimer

This document has been produced with the assistance of the European Union. The contents of this publication are the sole responsibility of the author and can in no way be taken to reflect the views of the European Union.

Executive Summary

This report describes the data collection strategy for the EURITO project. Developed data collection strategy comprises of three main stages:

- data extraction
- data ingestion
- data enrichment

We first define requirements and tools for the whole data collection phase and then discuss specific requirements and tools for each of the stages of the data collection phase.

The inputs for this data structure were drawn upon the outputs generated from the results of the scoping report (T1.6), data and metadata infrastructure (T2.2) and the pilot implementation phase in general (WP2 “Exploration phase”).

The data collection phase strategy presented here provides a backbone for the subsequent phases of the EURITO project: pilot scale up implementation, quantitative analysis and visualisation prototyping.

Table of Contents

List of Figures

No table of figures entries found.

List of Tables

No table of figures entries found.

1. Introduction

With a variety of available data that could be used in data science projects to support policy-making in Europe, it is crucial to define a data collection strategy to minimize issues in the data pipeline during the scaling up phase of the project or during the actual usage of the developed systems. This is especially evident when such data analytics projects are performed by a team of data analysts, where each analyst has his or her own preferred methods and tools.

Present report builds on findings from European Commission report on data mining (Campbell et al., 2017), which discusses technological aspects of using data mining within private and public sectors. We then follow up on the additional considerations during the data collection phase that may arise during the implementation of data analytics projects in the public policy domain.

The strategy for the data collection phase outlined in this report addresses three stages of the overall data pipeline:

1. **Data extraction** process concerns methods of data retrieval from various data sources for the further preprocessing and analysis. Every data source has their own means of providing data for final users, so having requirements to data extraction methods will help to make sure that data is obtainable from a wide variety of sources. In most cases, techniques such as downloading data dumps, access through API and web-scraping are used to extract data.
2. **Data ingestion** deals with the further preprocessing of the extracted data and preparing it for the analysis. As the usability of the analysis is highly dependent on the quality of data, cleaning and normalisation methods are employed on the raw data. Extracted raw data might include personal data, so anonymisation techniques can be used to protect subjects behind the retrieved data.
3. **Data enrichment** covers the integration of different datasets to produce new and potentially useful data based on connections. Two datasets can be joined together through one field to derive new information about entities in both datasets.

Aforementioned three-stage description of the data collection phase aims to ensure a structured approach to organising appropriate methods and at the same time, to allow sufficient flexibility through modularisation of the methods for accommodating a range of data sources and potential applications. Then, the data collection strategy defined here will serve as a backbone for the implementation of the data collection methods and tools.

Therefore, the goals of the data collection strategy presented here as defined as follows:

- coordinate the data collection phase in the EURITO project and guide the building of the common data infrastructure during the scale up phase
- guide data analysts in the public policy domain that would like to perform similar data science projects or further extend on the findings reported in the EURITO project in setting up sustainable and robust data pipelines in the long term

In this report the definition of term “data collection” differs from the one from traditional data collection methods, such as surveys, experiments, interviews, i.e. when data is directly

collected from data subjects. Data collection here refers to automated data retrieval and preprocessing before the analysis phase, usually from available web sources, i.e. without direct participation of data subjects (Liang and Zhu, 2017).

2. Data Collection Phase

In this section, we outline general requirements for data collection methods when designing data analytics methods and tools for policy makers. These requirements are relevant for the whole data pipeline.

Robustness. Ensuring perfect data quality is impossible. While there are data preprocessing methods that detect and handle errors and inconsistencies in data, methods in data pipeline have to be able to produce relevant results even when the data quality decreases.

Modular architecture. Methods and algorithms used in the data pipeline have to have clear and separate functions, which allows breaking complex processes into smaller, reusable operations. Proper modularisation enables standardization of methods across the data pipeline, alleviates debugging, maintaining and further developing the data infrastructure (Michell, 2018).
Tools: Python modules, unit testing, code review ("pull requests")

Scalability. As the amount of generated data grows, developed methods and tools have to accommodate these changes in volume and velocity of data. Moreover, it is critical that such data growth does not affect the overall performance of the system (e.g. time for queries, computation time for quantitative analysis, rendering time for interactive visualisations). Following the modularity principle, parameterisation of processes have to be in place, for scalable operations with data.
Tools: AWS (EC2, Batch, RDS)

Reproducibility and reusability. Developed data collection methods have to be reproducible by other stakeholders of the project, i.e. policy makers, researchers, private companies. To achieve this, adequate modularisation and transparency of data collection methods through thorough documentation of the produced code have to be ensured. Open access to raw and intermediately processed data, as well as to data collection methods themselves must be realised.

Tools: Python modules, Git, integrated unit testing (Travis), documentation (sphinx, readthedocs)

Versioning. During the development and maintaining the data pipeline tools, developers often need to switch between multiple versions of code to implement new features or fix bugs. Versioning software allows concurrent and coordinated development of data pipeline tools.
Tools: Git

Logging and monitoring. Documenting execution of data pipeline processes improves transparency and maintainability of the data infrastructure. Important events, such as start and finish of the data retrieval process, as well as information about the amount of loaded

data, operations performed on data, errors during the execution have to be logged and ready for the subsequent analysis.
Tools: Python modules

Cost. It is preferred that the software used in the project is free to use for research purposes. However, in case when there is no free option of a software with the critical functionality, the cost of proprietary software should not be prohibitive to the general audience.

Licensing requirements. When using third-party software, one needs to be aware of the necessary licensing requirements. Priority should be given to more permissive open source licenses. An overview of open source licensing requirements is provided in (Open Source Initiative, 2019)

Privacy and ethics requirements. Given the GDPR regulations that came into effect on May 25, 2018, "privacy by design" and "privacy by default" paradigms should be maintained throughout the project (Danon, 2019). "Privacy by design" paradigm means that when dealing with personal data, designers of data services have to consider privacy from the earlier phases of development. "Privacy by default" principle states that the default configuration for the services dealing with personal data have to be set to the level that implies maximum privacy.

On a more general perspective, Floridi and Taddeo (2016) define **data ethics** as "a *new branch of ethics that studies and evaluates moral problems related to data (including generation, recording, curation, processing, dissemination, sharing and use), algorithms (including artificial intelligence, artificial agents, machine learning and robots) and corresponding practices (including responsible innovation, programming, hacking and professional codes)*". Therefore, besides privacy considerations that arise when dealing with access to personal data, authors mention the threat of re-identification of data subjects (both individual and as a group) through data analysis. Such re-identification can happen when using data analysis techniques even the data that was classified initially as non-personal, can lead to the identification of individuals or groups.

2.1 Data Extraction

The data extraction step refers to retrieval of data from relevant data sources for subsequent storage and processing. Depending on the degree of required amount of coding, there are three major methods of data extraction:

Bulk download. Range of official data sources (e.g. Eurostat, European Innovation Scoreboard, European open data portal) provide data dumps for download. Usually, the data dumps are provided in a form of archived CSV, XML or XLS files and available for the direct download. The benefit of providing data in this format is that no programming knowledge is required to open and process such downloaded data.

Web API access. Along with the bulk download option, an increasing number of public data sources provide access through application programming interfaces (APIs). REST (Representational State Transfer) paradigm is used in majority of public APIs and sends requests through HTTP protocols to access data. Response to requests are usually returned

in the form of JSON files. REST API provides a standardised and automated way to extract the data through predetermined set of commands.

When performing data extraction through API, it is critical to comply with terms of service of the data providers. Another consideration is to check for retrieval rates, as too rapid retrieval can sometimes lead to missing data without any notification.

Web scraping Another method of retrieving data from web sources is to save the respective web pages and parse the saved content according to the needed structure. Web scraping is often used when there are no bulk download or API options available (e.g. separate websites of numerous private companies). While web scraping allows to extract all the visible information from a website, this method required the most coding effort, especially when data is gathered across the variety of websites.

Some websites have specific requirements for automated data collection and in some cases unauthorised web scraping can be considered as a copyright violation. Therefore, when performing web scraping, having a clear understanding on what is allowed by each particular website is necessary. Typically, owners of a website create a robots.txt file that specifies which areas of the website could be scraped.

Tools Bulk download is normally performed with standard computer browsers. Web API access can be performed with python or php scripts. While web scraping can be done with Python modules (e.g. BeautifulSoup), there is a variety of web services that does not require knowledge of programming.

2.2 Data Ingestion

Data ingestion is an intermediate step between data gathering and data enrichment that improves data quality, handles anonymisation, storage, access and backup issues and provides a structure for subsequent enrichment and analysis.

Anonymisation. Anonymisation is the process of removing any information that could lead to the identification of the data subject. According to Recital 26 of GDPR (EU GDPR, 2018), anonymised data is not considered personal and therefore, can be used without the consent of the data subjects and for various purposes *(*Lubowicka, 2018). Data anonymization methods include attribute suppression, record suppression, character masking, pseudonymisation, generalisation, swapping, data perturbation, synthetic data, data aggregation *(*Lubowicka, 2018).

Tools: ARX, Amnesia, Python modules (FAKER)

Cleaning methods handle the following issues with data quality (Tan et al., 2015):

- missing values (eliminate record or ignore missing values)
- inconsistent values
- duplicate data
- data standardisation (e.g. units of measure)

Tools: Python modules (Pandas, NumPy), Google Refine

More sophisticated cleaning methods that require quantitative analysis (e.g. outlier and noise detection, estimation of missing values) are outside of the scope of this report and will be covered in the report for D3.2 “Quantitative methods”.

Data storage and access.
As both raw and processed data needs to be stored in a system, one of the major requirements for data analytics projects is to allow *shared access* between the users of the system and ensure *scalability* of the available storage space and computing power when the amount of data grows. *Versioning* is necessary feature not only for code, but for data itself as well. Whenever an issue in data is found, *data traceability* allows to determine the source of an error and trace back the origins of the whole dataset, if necessary. Typically, currently available cloud infrastructure technologies offer a wide range of solutions to satisfy these major requirements.

From the database standpoint, both raw and processed data can be stored in the form of flat files (e.g. CSV) or in *database management systems*. After data is processed and analysed, mechanisms to query and output the results of the analysis through a frontend application (including visualisation) should be setup. In addition, direct API access to the data could be implemented so that the authorized third parties could directly retrieve necessary data in an automated manner for their own subsequent analysis.

2.3 Data Enrichment
During the data enrichment phase, collected, cleaned and normalised datasets are integrated together or augmented with additional external information. This allows to link data together and create new attributes (features) within already extracted data to improve subsequent analysis. Data enrichment stage may include restructuring and cleaning of integrated datasets, as issues described in Section 2.2 may arise again after integration of several datasets.

2.3.1 Data linking
In this section, the reasoning and the overarching methodologies behind data linking are discussed. Any given type of data records (e.g. organisations, people, industries) might be initially stored in unrelated data sources. For example, a university offers courses for students whilst it also produces academic outputs, but the “footprint” of these two types of data are stored in different locations. In this case, to produce data about the academic publications and university courses per a given country, one only needs to connect universities to countries. This could be done by accessing geographical locations of universities and mapping these locations to the standard list of world countries. As with countries, similar joins can be performed with time, as data records typically have a time field. We define these types of join as “*natural*” joins.

On the other hand, we may be interested in the number of academic publications and the number of university courses in a given university. In this case, we would need to join the data based on the name of the university, so that the joint entity would be the university. We define this type of join as a “*artificial*” join, which is much more challenging, as there is no single standard dictionary of university names that we could relate to, opposed to the list of countries in the former example. When performing an artificial join, name disambiguation process can

be time consuming, if not impossible. Thus, one must design an algorithm to decide on the best possible match, and generally there is a (potentially unknown) degree of uncertainty in the joining procedure.

While challenging, this type of data enrichment can provide benefits to the data analysts, provided that the risks of such data linking are acknowledged.

2.3.2 Data management

FAIR data management principles (Wilkinson et al.,2016) are a set of principles proposed to improve data reusability, findability, accessibility and interoperability of academic data. Following these principles when building a data infrastructure will facilitate the transparency, reproducibility and the long-term sustainability of the produced data and tools. Moreover, once the R&I indicators and the respective analytics tools are built and released for the public use, it is crucial that FAIR principles are further continuously maintained, e.g. when adding new datasets or implementing additional quantitative methods.

The following definitions of principles are quoted from GoFAIR initiative portal (GoFAIR initiative, 2019):

Findable: (meta)data has to be searchable and identifiable by users and computers.

"*F1. (Meta)data are assigned a globally unique and persistent identifier*
F2. Data are described with rich metadata
F3. Metadata clearly and explicitly include the identifier of the data they describe
F4. (Meta)data are registered or indexed in a searchable resource"

Accessible: once (meta)data is found, users need to clearly understand how (meta)data can be accessed:

"A1. (Meta)data are retrievable by their identifier using a standardised communications protocol
A1.1 The protocol is open, free, and universally implementable
A1.2 The protocol allows for an authentication and authorisation procedure, where necessary
A2. Metadata are accessible, even when the data are no longer available"

Interoperable: (meta)data has to be able to be processed with a variety of processes in the overall data infrastructure without compatibility issues:

"I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2. (Meta)data use vocabularies that follow FAIR principles
I3. (Meta)data include qualified references to other (meta)data"

Reusable: enriched data has to be sufficiently documented allowing the further reuse:

"R1. Meta(data) are richly described with a plurality of accurate and relevant attributes
R1.1. (Meta)data are released with a clear and accessible data usage license

R1.2. (Meta)data are associated with detailed provenance
R1.3. (Meta)data meet domain-relevant community standards"

3. References:

1. Mitchell, T. (2018). “ETL Best Practices”. Accessed at https://www.timmitchell.net/etl-best-practices/.

2. Weber, B. (2018). “Data Science for Startups: Data Pipelines”. Accessed at https://towardsdatascience.com/data-science-for-startups-data-pipelines-786f6746a59a

3. Campbell, D., Tippett, C., Struck, DB., Lefebvre, C., Côté, G. and Archambault, É. 2017. ‘Data Mining on Key Innovation Policy Issues for the Private Sector: Technical Report’. Prepared by Science-Metrix for the European Commission.

4. GoFAIR initiative (2019). Accessed at https://www.go-fair.org/fair-principles/.

5. Tan, P. N., Steinbach, M., & Kumar, V. (2005). Introduction to data mining.

6. Lubowicka, K. (2018). “The Ultimate Guide to Data Anonymization in Analytics”. Accessed at https://piwik.pro/blog/the-ultimate-guide-to-data-anonymization-in-analytics/

7. Open Source Initiative. “Licenses by Name”. Accessed at https://opensource.org/licenses/alphabetical

8. EU GDPR (2018). Recital 26. Accessed at http://www.privacy-regulation.eu/en/recital-26-GDPR.htm

9. Wilkinson, M.D., Dumontier, M., Aalbersberg, Ij.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., et al. (2016), “The FAIR Guiding Principles for scientific data management and stewardship”, *Scientific Data*, Vol. 3, p. 160018, https://doi.org/10.1038/sdata.2016.18.

10. Floridi, L. and Taddeo, M. (2016), “What is data ethics?”, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, https://doi.org/10.1098/rsta.2016.0360.

11. Danon, S. (2019). “GDPR Top Ten #6: Privacy by Design and by Default”. Accessed at https://www2.deloitte.com/ch/en/pages/risk/articles/gdpr-privacy-by-design-and-by-default.html