Social Mining & Big Data Ecosystem

SoBigData

RESEARCH INFRASTRUCTURE

Project Acronym	***SoBigData***
Project Title	***SoBigData Research Infrastructure*** ***Social Mining & Big Data Ecosystem***
Project Number	***654024***
Deliverable Title	***Data processing workflow specification language***
Deliverable No.	***D10.11***
Delivery Date	***01 September 2016***
Authors	***Leonardo Candela (CNR), Fosca Giannotti (CNR), Valerio Grossi (CNR), Paolo Manghi (CNR), Roberto Trasarti (CNR)***

SoBigData receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 654024

DOCUMENT INFORMATION

PROJECT	
Project Acronym	SoBigData
Project Title	SoBigData Research Infrastructure Social Mining & Big Data Ecosystem
Project Start	1st September 2015
Project Duration	48 months
Funding	H2020-INFRAIA-2014-2015
Grant Agreement No.	654024
DOCUMENT	
Deliverable No.	D10.11
Deliverable Title	Data processing workflow specification language
Contractual Delivery Date	31 August 2016
Actual Delivery Date	01 September 2016
Author(s)	Leonardo Candela (CNR), Fosca Giannotti (CNR), Valerio Grossi (CNR), Paolo Manghi (CNR), Roberto Trasarti (CNR)
Editor(s)	Roberto Trasarti (CNR), Valerio Grossi (CNR)
Reviewer(s)	Paolo Manghi (CNR), Roberto Trasarti (CNR)
Contributor(s)	
Work Package No.	WP 10
Work Package Title	JRA3_SoBigData e-Infrastructure
Work Package Leader	CNR
Work Package Participants	CNR, USFD, UNIPI, FRH, UT, IMT, LUH, KCL, SNS, AALTO, ETHZ
Dissemination	PU
Nature	Websites, patents filling, etc.
Version / Revision	V1.0
Draft / Final	Final
Total No. Pages (including cover)	20
Keywords	Workflow Language, Processing, Systems Integration

DISCLAIMER

SoBigData (654024) is a Research and Innovation Action (RIA) funded by the European Commission under the Horizon 2020 research and innovation programme.

SoBigData proposes to create the Social Mining & Big Data Ecosystem: a research infrastructure (RI) providing an integrated ecosystem for ethic-sensitive scientific discoveries and advanced applications of social data mining on the various dimensions of social life, as recorded by "big data". Building on several established national infrastructures, SoBigData will open up new research avenues in multiple research fields, including mathematics, ICT, and human, social and economic sciences, by enabling easy comparison, re-use and integration of state-of-the-art big social data, methods, and services, into new research.

This document contains information on SoBigData core activities, findings and outcomes and it may also contain contributions from distinguished experts who contribute as SoBigData Board members. Any reference to content in this document should clearly indicate the authors, source, organisation and publication date.

The document has been produced with the funding of the European Commission. The content of this publication is the sole responsibility of the SoBigData Consortium and its experts, and it cannot be considered to reflect the views of the European Commission. The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

The European Union (EU) was established in accordance with the Treaty on the European Union (Maastricht). There are currently 27 member states of the European Union. It is based on the European Communities and the member states' cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice, and the Court of Auditors (http://europa.eu.int/).

TABLE OF CONTENT

DELIVERABLE SUMMARY

This document contains a general overview of the workflow language definition status. All the content can be found in the public webpage at the following addess: https://support.d4science.org/projects/sobigdata-eu/wiki/workflowlanguage

EXECUTIVE SUMMARY

Research e-infrastructures are “systems of systems”, patchworks of tools, services and data sources, evolving over time to address the needs of the scientific process. Scientist implement their processes by workflows that include usage of web services, download of shared software libraries, or tools, access to workflow execution engines. In such scenarios, scientists may obtain interesting results by reusing common tools, but the underpinning heterogeneity hinders their ability to represent, share and reproduce such workflows. The work here presented is FLARE, a workflow language addressing the problem of supporting the specification of a scientific process in highly-heterogeneous e-Infrastructures. FLARE lays in between *business process modeling languages*, which offer a formal and high-level description of a reasoning, protocol, or procedure, and *workflow execution languages*, which enable the fully automated execution of a sequence of computational steps via dedicated engines.

The works is presented also at Repscience 2016 workshop (http://repscience2016.research-infrastructures.eu/) in order to receive useful feedbacks from the scientific community since the first steps of definition and development.

1 RELEVANCE TO SOBIGDATA

1.1 PURPOSE OF THIS DOCUMENT

This document contains a general overview of the workflow language definition status. All the content can be found in the public webpage at the following addess: https://support.d4science.org/projects/sobigdata-eu/wiki/workflowlanguage

Relevance to project objectives

The website will be updated with the progress of the task T10.3. The objective of the workfow specification language is to enable the definition and execution of data-driven workflows in terms of combination of SoBigData resources, e.g. pipelining of services, execution of algorithms, and selection of data for services and/or algorithms. By means of the language, SoBigData scientists will be able to create, share, discover and reproducetheir scientific workflows. Scientists will be encouraged to use SoBigData VREs and their applications to both expose their reasonings, i.e. expressed in terms of workflows, and reproduce or simply reuse such reasonings to develop their scientific process.

1.2 RELATION TO OTHER WORKPACKAGES

The workflow language is related to

- WP3: As soon as the language and the software infrastructure will be released to the final user the dissemination will take care of presenting examples in the various events.
- WP8 and WP9: The two workpages define the datasets and methods to be integrated and then supported by the language.

1.3 STRUCTURE OF THE DOCUMENT

The rest of the document is organised as follows. Section 2.1 introduces the context and the genearal idea, Section 2.2 describes the FLARE language and its constituents. Section 2.3 describes how the FLARE-based approach can be successfully implemented in the case of SoBigData.eu, a large scale Research Infrastructure intended to serve the Social Mining research community. Section 2.4 reports some consideration and the future works.

2 FLARE: A FLEXIBLE WORKFLOW LANGUAGE FOR RESEARCH INFRASTRUCTURE

2.1 INTRODUCTION

Over the past decade Europe has developed world-leading expertise in building and operating e-Infrastructures[1]. They are large scale, federated and distributed research environments in which researchers have shared access to unique scientific facilities (including data, instruments, computing and communications), regardless of their type and location in the world. They are meant to support unprecedented scales of international collaboration in science, both within and across disciplines, their aim is to realise a common environment where scientists can create, validate, assess, compare, and share their digital results of science, such as *research data*, intended as scientific data produced by a scientific effort, and *research methods*, intended as "actions" produced by a scientific effort (the terminology is discipline-specific, but examples include software, services, tools, workflows, scripts, algorithms, protocols).

In the last decade, all stakeholders of the research life-cycle (e.g. researchers, organizations, funders) have highlighted and endorsed the importance of applying Open Science publishing principles [Bartling et al. 2014]. According to such principles, researchers should "publish" their scientific results in order to enable transparent evaluation and reproducibility of science. According to such vision, the scientific article is only one of the possible publishable products, certainly required but insufficient at supporting Open Science principles. The Open Science movement encourages researchers to publish research data and methods valuable to their research (e.g. input and output data, a text-mining algorithm), the e-infrastructure tools and services they used to implement their research, and possibly their research *workflow*, intended as the sequence of steps they performed to reach their results. The availability of all the constituents parts and results of a scientific process, which is in turn described in an article, maximizes the chances to correctly evaluate the quality of the research and the chances to re-use results produced by others (reducing the cost of science). In particular, where available, the availability of workflows is of paramount importance in order to address automated execution of scientific experiments but, more in general, to enable reproducibility of science.

The implementation of Open Science principles is hindered by a multitude of problems. One of the most prominent is that e-Infrastructures available to research communities today are far from being well-designed and consistent environments embracing all needs of a research community, e.g. designed to share and reuse resources, be them datasets or research methods, according to common policies, standards, language platforms, and APIs. They are rather "systems of systems", patchworks of tools, services and data sources, evolving over time to address the needs of the scientific process; examples are web services devised to deliver discipline-specific functionalities, shared software libraries to be used to implement research methods, desktop tools, web-accessible execution engines (e.g. Taverna). As such, they are often

[1] https://ec.europa.eu/digital-single-market/en/news/e-infrastructures-making-europe-best-place-research-and-innovation , http://ec.europa.eu/research/infrastructures/index_en.cfm

equipped with resource catalogues, offering resource registration and discovery of research data and methods, but generally lack of e-infrastructure workflow languages. Subsystems of e-infrastructures may indeed support workflow languages and engines (e.g. Taverna, BPEL), but these require methods (and datasets) to meet minimal integration requirements, which are not generally addressed by all tools and services available to e-infrastructures.

FLARE is a workflow language under design in SoBigData addressing the problem of supporting the specification of a scientific process in highly-heterogeneous e-Infrastructures. FLARE lays in between *business process modeling languages* [Rosemann et al. 2000], which offer a formal and high-level description of a reasoning, protocol, or procedure, and *workflow execution languages* (e.g. BPEL [Weerawarana et el. 2005]), which enable the fully automated execution of a sequence of computational steps via dedicated engines. Specifically, the language models a *scientific process workflow* as a sequence of *scientific steps* of four main kinds:

- *Tools (to be downloaded)*: the execution of the step requires the user to download and execute the tool on its own premises;
- *Web-accessible services* (SOAP or REST): the execution of the step requires a call to the service that is operated by a provider;
- *Web-accessible applications* (tools accessible via user interfaces from the web): the execution of the step requires accessing the web user interface;
- *Executable workflows*: the execution of the step requires invoking the respective workflow execution engine;
- *Scientific process workflows:* indeed workflows can be obtained by combining, i.e. nesting, other workflows.

FLARE defines a framework where such steps can be described and combined by researchers to specify a scientific process workflow and share it with others. The identified scenario is that of a scientist working with a highly heterogeneous e-Infrastructure, hence using its different tools and services to run her experiments, by reusing and generating research data and methods. Once the experiment is concluded, the scientists has identified the *workflow steps* she needs to use to reproduce it and she is now willing to materialize the relative FLARE workflow in order share it with others (and enable others to make use of it, e.g. repeat the experiment for validation purposes, repurpose the experiment by altering the settings). FLARE-based tools can be constructed, which offer user interfaces to (i) support the scientists at constructing a workflow, (ii) publish the workflow as an e-infrastructure resource to be discovered and reused by others, and (iii) execute (i.e. reproduce) the workflow, for example using a UI wizard. The wizard user interface instructs the researchers on which steps they should manually execute to repeat the experiments but, when a sequence of steps is based on components of executable workflows, execute automatically the relative subpart.

2.2 WORKFLOW LANGUAGE AND OPERATORS

One of the aim of a RI is to guarantee an integrated framework in order to provide the researchers of the RI with an homogeneous and powerful way for designing new experiments, performing and sharing them among the community. The need of a language that binds workflows and processes for orchestrating the

resources (data and methods) and the human interaction is required for managing all the use cases that the RI should support. This language has to orchestrate cross-disciplinary research results implying the execution of different software developed in different programming languages, the interoperability with existing web services and a way for capturing the human interaction. In literature there are several tentative to design such language, e.g. in [Ceri et al. 2013] a vision of how a mega-modeling language can integrate different tool is presented. Our philosophy is similar to the one proposed by JBPM for Business process. It makes the bridge between business analysts and developers offering management features for both business users and developers [jBMP2016].
In general in literature two main types of workflow modelling language exist, both with specific features and requirements:

- **Conceptual language**: in this scenario, the RI supports the coordination of softwares/datasets download and execution but does not support any actual execution. In this situation, the workflow represents only a set of steps that drives the user to reproduce the experiments locally. It shows the instructions on how and where data can be downloaded and what are the requirements of the software, where it can be downloaded, and executed. An example of this kind of languages are UML [UML 2005] and YAWL [Aalst et al. 2005].
  The limitation of this kind of languages is the fact that they are abstract and do not contains specific information for a practical implementations and executions.
- **Processing language** the instance of a process is fully described and all the nodes involved into the workflow are fully integrated by RI. In this situation the execution of the workflow is managed by the RI, that covers both the computation of the single node and the orchestration among the nodes, included datasets and software movement. This last case requires a detailed language for capturing all the node features, and typically requires much time for configuring all the interaction. The advantage of this situation is that the final user can reuse the same experiment and typically change some parameters before running the experiment. On the other, hand we can have less expressivity since the modification of the experiment can involve a process of rewriting. Examples of this kind are all the workflow languages in analytical tools such as KNIME [Berthold et. al 2007], SAS [Sas 2011] or Meandre [Llora et al. 2008]. The drawback of such tight integration is the fact that all the basic steps of each process must be adapted and framed into the system to be executed. This is not always possible because sometime the author is not willing to do it or simply by the nature of the process in terms of interaction, e.g. all the visual analytics tools.

We propose a workflow modeling language which is in the middle of the two: the execution is orchestrated by the RI and might requires a direct interaction with a user, in one or more phases of the workflow in order to complete the entire experiment. In this case, the workflow language should enable the definition of two kinds of nodes: (i) the ones that require the explicit human interaction, and (ii) the ones that are completely managed by the RI. The data and software flow should support both situations. In real domain, this is the most typical situation. The execution is partially performed by the RI with no direct human control, but the workflow contains steps that requires human feedback and the process cannot be completed without this interaction. This case happens when the information required is not available when the workflow is written since it has not produced yet or the process requires an external source of information which is not

integrable into RI or human interaction which cannot be coded into constraints or rules (i.e. visual analytics).

In this context, we think that the success of our workflow language is to cover the above two cases proposing a flexible and adaptable language without forcing the complete description of the execution, promoting a general way for enabling expiring reproducibility and knowledge sharing. We call it FLARE (a Flexible workflow LAnguage for Research infrastructurE).

The proposed language operators can be summarized in the follow categories:

- *Data management*: The super-category containing the operators dealing with data access, transformation, and storage. All the operators specify the data location, e.g. a database, a file system.
    - *Data Reader/Writer*: reads/writes the data from/to a repository location;
    - *Data Manipulation*: transforms the data in terms of format or values by using aggregations or transformations.
- *Execution*: An external algorithm is executed specifying the parameter and the data source (if required).
- *Workflow controls*: Define the execution flow of the process, i.e conditional, loops, variable, etc.

All the operators can be specified as *automatic* or *interactive*. In the first case specific information will be required in order to allow the system managing the operator without human intervention. In the other case a description must be specified by the system in order to explain, to future user, which are the steps that are required to proceed. This description may be generalized as an interactive webpage with all the materials and links needed.

An example of workflow is depicted in Figure 1. In this example composed by six steps it is described an analysis process to discover the possible vendors for some products which are part of interesting purchase patterns in a store.

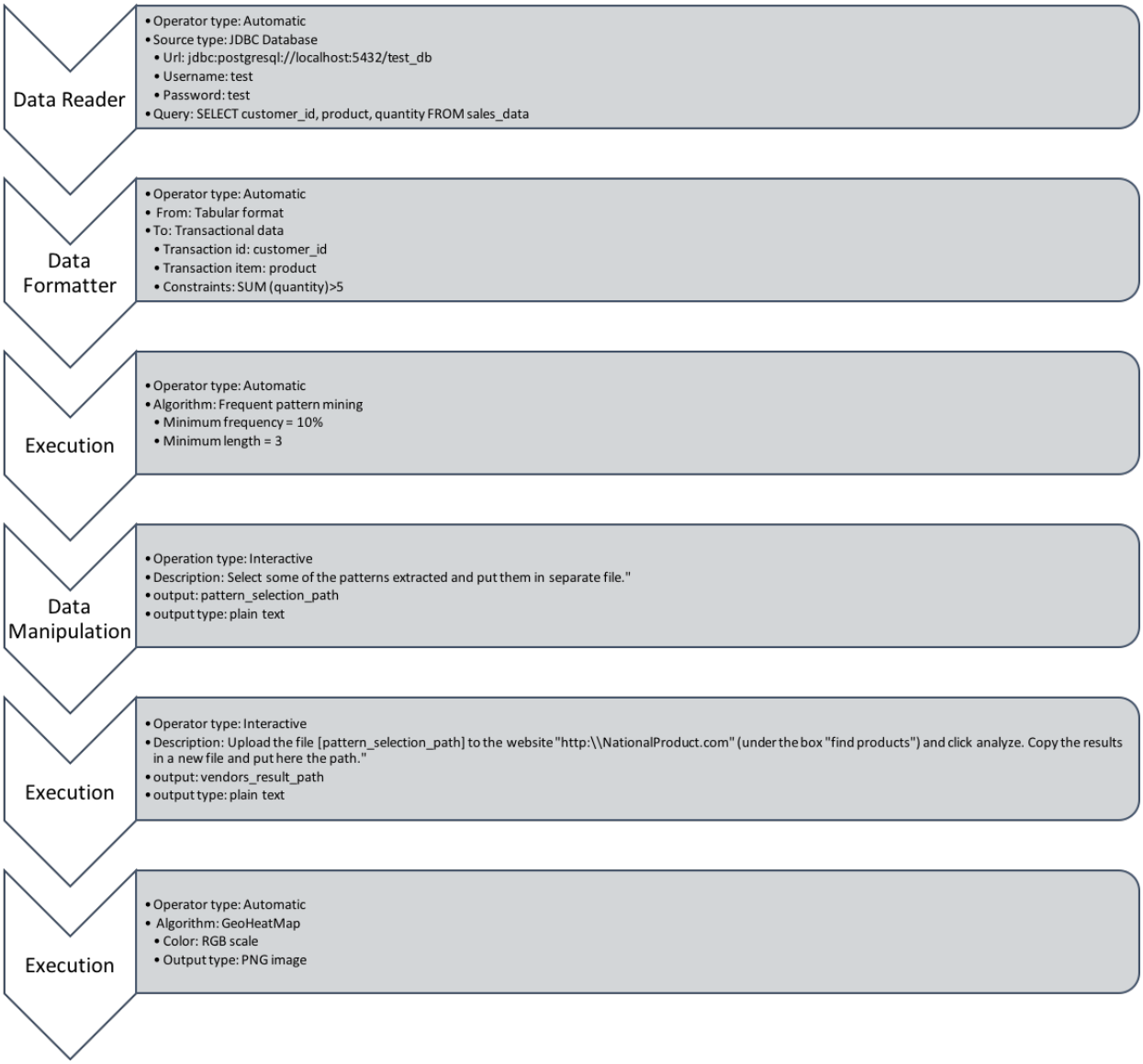

***Figure 1**: An example of process where both automatic and human interaction operators are integrated to represent a FLARE workflow.*

In the first step we connect to the database to extract the history of sales, than a data formatter operator will transform the data from a relational view to a transactional format in order to have for each line the set of products associated to a customer. This operator is able to apply some filter to removing all the products which are bought less than 5 times by the customer. The resulting dataset of transaction is passed to an algorithm of frequent pattern discovery extracting a set of interesting models. All the operators until this point of the analysis were automatic, this means that the user is able to configure them apriori and they will execute the process without any human intervention. The next operators is interactive, in fact the system cannot automatically execute the task due the fact that the user is required to select a set of patterns which are “interesting” from his point of view. The result is a new file where only the interesting pattern are

stored. Then another interactive operator is used to execute a search over national vendors for the selected items (the patterns selected). This operator is interactive because the search is done by an interactive web-page without any real API to access to the service uploading the file built in the previous step. Then after that all the system is able to finish the analysis by automatically executing the rendering of the vendors/products locations on a map so the user is able to see from where he can buy products to satisfy the frequent basket request of his customers.

Clearly this is a small example of analysis which show the interleaving of automatic and interactive operators. In the next section we will propose an implementation of this idea on a more complex scenario.

2.3 SOBIGDATA CASE STUDY

In this section we study a real case study of a framework containing a broad variety of tools: the European project SoBigData.eu[2]. SoBigData.eu proposes to create the Social Mining & Big Data Ecosystem: a research infrastructure (RI) providing an integrated ecosystem for ethic-sensitive scientific discoveries and advanced applications of social data mining on the various dimensions of social life, as recorded by "big data". SoBigData.eu is opening up new research avenues in multiple research fields, including mathematics, ICT, and human, social and economic sciences, by enabling easy comparison, reuse and integration of state-of-the-art big social data, methods, and services, into new research. In addition, as an open research infrastructure, SoBigData.eu promotes repeatable and open science. Although SoBigData.eu is primarily aimed at serving the needs of researchers, the openly available datasets and open source methods and services provided by the new research infrastructure will also impact industrial and other stakeholders (e.g. government bodies, non-profit organisations, funders, policy makers). Clearly this represents a challenge with the pre-requisites described in previous sections: complexity, impossibility to impose a standard, variety in technologies and kind of interaction with the operators. To show the feasibility of the idea we present the implementation of an analytical process illustrated in SoBigData called *City of Citizens*[3] using FLARE language.

2.3.1 VIRTUAL RESEARCH ENVIRONMENTS

Virtual Research Environments are innovative, web-based, community-oriented, comprehensive, flexible, and secure working environments conceived to serve the needs of nowaday scientific investigations [Candela et al. 2013]. The implementation and operation of such challenging and evolving working environments largely benefits from and complements the offering of Research Infrastructures. In particular, in SoBigData an approach based on an open ecosystem of e-Infrastructure having d4science.org as pivot [Assante et al 2016, Candela et al 2014] is used. Via the pivotal infrastructure it is possible to provide scientists with innovative and enhanced services (including those offered by single Research Infrastructures)

[2] SoBigData Website: http://www.sobigdata.eu

[3] City of Citizens website: http://www.sobigdata.eu/exploratories/city-citizens

organized in dedicated VREs each tailored to satisfy the needs of a designated community. Such advanced services include (i) a rich array of services guaranteeing seamless data discovery and access both per-data typology and across data typologies and data sources, (ii) a data analytics platform benefitting from a distributed and multi-tenant computing infrastructure oriented to provide scientists a broad variety of algorithms and methods as well as other kind of web-based resources, (iii) collaboration oriented facilities enabling scientists to publish research results with the possibility to add comments on them in a social-network fashion. Equipping the VRE with FLARE allow the user to attach to a specific result the entire process used to obtain it. This enhance all the environment to a living laboratory which contains not only the methods and the results but also the experience of the researcher in using them in and compose analytical process with it.

2.3.2 RESOURCES AS OPERATORS

In this section we describe how the general operators described previously are implemented in the SoBigData.eu infrastructure and how it is supported in the VREs. As described above the VRE is a working environment tailored to serve the needs of a specific context, thus it is a selected "view" build upon the entire offering of a Research Infrastructure. The general term resource comprehends both data sources, algorithms, links to external services (e.g. web services) and computational nodes of the infrastructure (i.e. the machine where the algorithm can be executed). The operator of the language map these different kind of resources with diverse constructs. The data management, in particular the readers and writers, operators in FLARE are classified by different types of data sources:

- *JDBC Database*: using the JDBC interface is possible to link a generic database. The data is interpreted as tabular format equipped with some metadata information such as name of the columns, number of rows, etc.
- *Registered Database*: the infrastructure give the possibility to register a database, in that case the operator specify only its alias to access to it.
- *Workspace object*: the Research infrastructure might be equipped with several services offering access to stored objects, such as files and data streams. Such objects are each characterised by a unique actionable identifier, a specific format, and an access protocol. A series of specific operators are needed to both read and store objects from/into the RI objects stores.
- *External file*: using FTP (File Transfer Protocol) it is possible to link an external file specifying its URL. Besides the URL it is fundamental to be provided with mime-types to properly consume the content of the specific file.
- *Interactive insertion*: This operator allow the user to insert manually the data to be used in several formats, e.g. plain text, tabular, XML, etc.

For the data manipulation operators we distinguish between data format transformations changing a data format to another (e.g. from plain text to tabular format and vice versa as well as XML field extractors, etc.) and the operators which alter the content of the data applying joins, aggregations and filters. The execution operators handle the broad variety of research results and tools in the infrastructure. They include two types of operators:

- *Internal execution*: the operator specifies the name of the operator which is integrated in the system, this kind of algorithms can be directly executed and handled by the infrastructure which optimizes their execution choosing the most suitable computational resource able to execute them. The operator include also all the parameters of the algorithm.
- *External execution*: the operator specifies the external link to a service and how to interact with it. In the case of web-services this operator may be automatic specifying the request to send to the server. In case of interactive services such as a web-portal it will contain the description of how to use it and the steps to be done to obtain the expected output.

Clearly all those operators represents atomic task of an analytical workflow. The last set of operators include the workflow controls able to specify the sequence of operators to be processed as well as conditional branches or loops. The language allows the user to specify also variables to be used in the other operators to make them more general and flexible (e.g. the path or name of a file, the value of some constraints, etc.).

This mapping between operators and research infrastructure resources allows the user to use FLARE inside the VREs to explain the analytical processes and to make them reproducible by the community which will be able also to comment and discuss not only about the final results but also how they are extracted. In the next section we present an example of this case in the context of mobility data analysis.

2.3.3 IMPLEMENTING AN ANALYTICAL PROCESS OF CITY OF CITIZENS

As described above City of Citizens is a set of analytical processes of the SoBigData project which includes several inhomogeneous tools developed by different research groups. Some of them are developed to be used by an user which interacts with an interface, some others are algorithms which can be configured a priori to execute a specific task. The overall objective of the analysis is to generate a set of statistics and models describing a territory by means of data, statistics and models. In Table 1 an example of process involving tools of the City of Citizens VRE is shown. The workflow presented is a mockup of the language we want to implement, following we will describe step-by-step the effect of each operator.

Workflow: City of Citizens - CarPooling for interesting area.
Begin

1. *city = interactive Insertion {Interactive, Description="Select a city to be analyzed"}*
2. *city_data = Data Reader {Automatic, Registered Database, alias="dataRepository", query="Select * from GPS_data where city='"+city+"'"}*
3. *If (city_data.size=0) End*
4. *city_trajectories = city + "_trajectories"*
5. *city_trajectory = External Execution {Automatic, Algorithm=Catalogue.get(TrajectoryBuilder), input_data=city_data, maxTime_gap=30min, maxSpace_gap=50m}*
6. *External Execution {Automatic, Web-service, url=Catalogue.get(UrbanMobilityAtlas), post=city_trajectories}*
7. *city_geometry = city + "_geometry"*
8. *External Execution {Interactive, Decription="Go to http://kdd.isti.cnr.it/uma2/?city=city to see the statistics generated by the Urban Mobility Atlas. Select from the toolbar 'in' and 'systematic' to see the traffic generated by commuters entering in "+ city + ". Do the same selecting 'out' and 'systematic'. Determine the areas you are interested in (e.g. the one with higher volume of traffic) and create a set of (postgres) geometry in a file representing them. Upload the file as "+city_geometry}*
9. *database_installed = Interactive Insertion {Interactive, Description= "Do you have access to a postgres database with postgis extension? (yes/no)"}*
10. *If (database_installed=='no')*

    External Executio {Interactive, Description="Go to https://www.postgresql.org/ download the software and install it. When the installation is done go to http://postgis.net/ and follow the instruction to install and enable postgis extension.)

11. *userDatabase = Interactive insertion {Interactive, Description="Specify your database url"}*
12. *userName = Interactive insertion {Interactive, Description="Specify your username"}*
13. *userPassw = Interactive insertion {Interactive, Description="Specify your password"}*
14. *Data Writer {Automatic, JDBC Database, url=userDatabase, username=userName, password=userPassw, import city_trajectory as "trajectory_data"}*
15. *Data Writer {Automatic, JDBC Database, url=userDatabase, username=userName, password=userPassw, import getFile(city_geometry) as "geometry_table"}*
16. *Data Manipulation {Automatic, JDBC Database, url=userDatabase, username=userName, password=userPassw, query="create table trajectories_instersection as select * from trajectory_data a, geometry_table b where st_intersect(a.trajectory, b.geometry)"}*
17. *External Executio {Interactive, Description="Download the software to compute the Mobility Profiles at http://www-kdd.isti.cnr.it/~trasarti/sobigdata.eu/mobilityprofile/index.html, configure the database.properties file in order to connect to your database. And execute it on trajectories_instersection table. As output_table parameter put profile_table)*
18. *External Executio {Interactive, Description="Download the software to compute the Car Pooling Matching at http://www-kdd.isti.cnr.it/~trasarti/sobigdata.eu/carpooling/index.html, configure the database.properties file in order to connect to your database. And execute it on profile_table)*

End

***Table 1**: The City of Citizens analytical process.*

At the beginning (1) the system will ask to the user to select a city which will be the focus of all the analysis. The line (2) retrieves the data from a registered public database called “dataRepository” where a set of GPS data is store. In case the data of the selected city is not available the process ends (3) otherwise the lines (4-5) build the trajectories (geometries composed by a sequence of GPS points according to specific spatio-temporal constraints) are created. The step (6) sends the data to a web-service called UrbanMobilityAtlas populating a webpage accessed in (7-8) by the user to select interesting areas considering the results obtained[4]. Steps (9-10) help the user to install a local database in case cannot be provided by the user. Steps (11-15) move the data from the “dataRepository” database to the local user’s database as well as importing the geometries created by the user in the step (8). Now all the data is in the local database can be filtered using a join (16) to take only the trajectories which influence the interesting areas. Finally the trajectories selected are processed using the Mobility Profiles algorithm (16) and then the Car Pooling Matching algorithm (17). The data is moved locally and the last two operators are interactive because the algorithms are not integrated in the system but are only available in download. The result describes the potential impact of the car-pooling in the areas of the city selected by the users. In this example several operators, automatic and interactive, interacts in order to describe the process followed by a researcher.

2.4 CONSIDERATIONS AND FUTURE WORKS

In this document, we proposed a case study inherited from the SoBigData experience. The outlined scenario represents a starting point on which building a workflow language that allows the representation and reproducibility of a scientific process in a research e-infrastructure.

FLARE will be the scientific process workflow language of SoBigData RI, and will cover all the different aspects related to the development of a RI for social mining in the context of big data. Based on this vision, as future work the idea is to provide a complete description of the language process expanding the current case study to the three general case outlined in this context. Furthermore, another direction is to provide a solid base where different scenario can be easily described as a workflow, that represents a sort of template for driving the final user in writing new experiments easily with the aim of sharing them and making them reproducible.

[4] See an example in http://kdd.isti.cnr.it/uma2/?city=Pisa

REFERENCES

[Assante et al 2016] M. Assante, L. Candela, D. castelli, G. Coro, L. Lelii, P. Pagano. Virtual Research Environments as-a-Service by gCube. 8th International Workshop on Science Gateways (IWSG 2016)

[Berthold et al. 2007] Michael R. Berthold and Nicolas Cebron and Fabian Dill and Thomas R. Gabriel and Tobias Kotter and Thorsten Meinl and Peter Ohl and Christoph Sieb and Kilian Thiel and Bernd Wiswedel. *KNIME: The Konstanz Information Miner.* Book Springer, Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007). ISBN 978-3-540-78239-1

[SAS 2011] SAS Institute Inc. 2011. *Base SAS® 9.3 Procedures Guide*. Cary, NC: SAS Institute Inc. Base SAS® 9.3 Procedures Guide. Copyright © 2011, SAS Institute Inc., Cary, NC, USA. ISBN 978-1-60764-895-6.

[UML 2005]. *Unified Modeling Language User Guide, The (2 ed.)*. Addison-Wesley. 2005. p. 496. ISBN 0321267974. , See the sample content, look for history

[Aalst et al. 2005] W. M. P. van der Aalst and A. H. M. ter Hofstede. *YAWL: yet another workflow language*. Journal Information Systems archive. Volume 30 Issue 4, June 2005. Pages 245-275.

[Candela et al. 2013] L. Candela, D. Castelli & P. Pagano Virtual Research Environments: An Overview and a Research Agenda. Data Science Journal. 12, pp.GRDI75–GRDI81. DOI: http://doi.org/10.2481/dsj.GRDI-013

[Candela et al. 2014] L. Candela, D. Castelli, A. Manzi & P. Pagano. Realising Virtual Research Environments by Hybrid Data Infrastructures: the D4Science Experience. In International Symposium on Grids and Clouds (ISGC) 2014 23-28 March 2014, Academia Sinica, Taipei, Taiwan, PoS(ISGC2014)022, Proceedings of Science, 2014.

[Ceri et al. 2013] Stefano Ceri, Themis Palpanas, Emanuele Della Valle, Dino Pedreschi, Johann-Christoph Freytag, Roberto Trasarti: *Towards mega-modeling: a walk through data analysis experiences*. SIGMOD Record 42(3): 19-27 (2013)

[Llora et al. 2008] Xavier Llora, Bernie XavierLlora`,Bernie Acs, Loretta S.Auvil, Boris Capitanu, Michael E. Welge, and David E.Goldberg. 2008. *Meandre: Semantic-Driven Data-Intensive Flows in the Clouds.* In Proceedings of the 2008 Fourth IEEE International Conference on eScience (e-Science '08). IEEE Computer Society, Los Alamitos, CA, USA, 238–245. DOI:http://dx.doi.org/10.1109/eScience.2008.172

[jBMP2016] Javal Business Process Management (jBPM). JBoss Community, Red Hat, Inc. Cur. Vers 6.4.0., http://www.jbpm.org/

[Rosemann et al. 2000] Becker, Jörg, Michael Rosemann, and Christoph Von Uthmann. "Guidelines of business process modeling." Business Process Management. Springer Berlin Heidelberg, 2000. 30-49.

[Weerawarana et. al 2005] Weerawarana, S., Curbera, F., Leymann, F., Storey, T., & Ferguson, D. F. (2005). *Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more*. Prentice Hall PTR.

[Bartling et al. 2014] Bartling, S., & Friesike, S. (2014). Towards another scientific revolution. In*Opening Science* (pp. 3-15). Springer International Publishing.