GPBS: statistical service implementation

F. Amato, M. Bruno, P. Francescangeli, R. Iannacone, S. Macone, G. Ruocco, M. Scannapieco

1. INTRODUCTION

In the last months, an initial version of a statistical service prototype was realized, according to the recommendations of the methodological committee. The team worked on the design and preliminary implementation of the core functionalities of a statistical service for selective editing.

In order to meet the needs of users with different skills (e.g.: survey staff, IT experts), a statistical service should, at least, provide the following set of functionalities:

- upload and analyse input/output data;
- data flow management;
- reporting output results;
- friendly graphical user interface.

The next paragraphs describe the general architecture resulting from the analysis performed by the different teams involved in the project (methodological, IT and domain experts).

2. STATISTICAL SERVICE ARCHITECTURE

According to Enterprise Architecture principles, the following analysis focuses on the main architectural layers (business, information and application), in order to:

- implement reusable components;
- harmonize the different viewpoints;
- define non-trivial executable workflows.

2.1. Business layer

The process steps needed to implement a statistical service, wrapping a specific method, are reported in Figure 1. The business process has been modelled according to Archimate standard. More precisely, *information objects* are displayed in orange, while *process steps* are yellow.

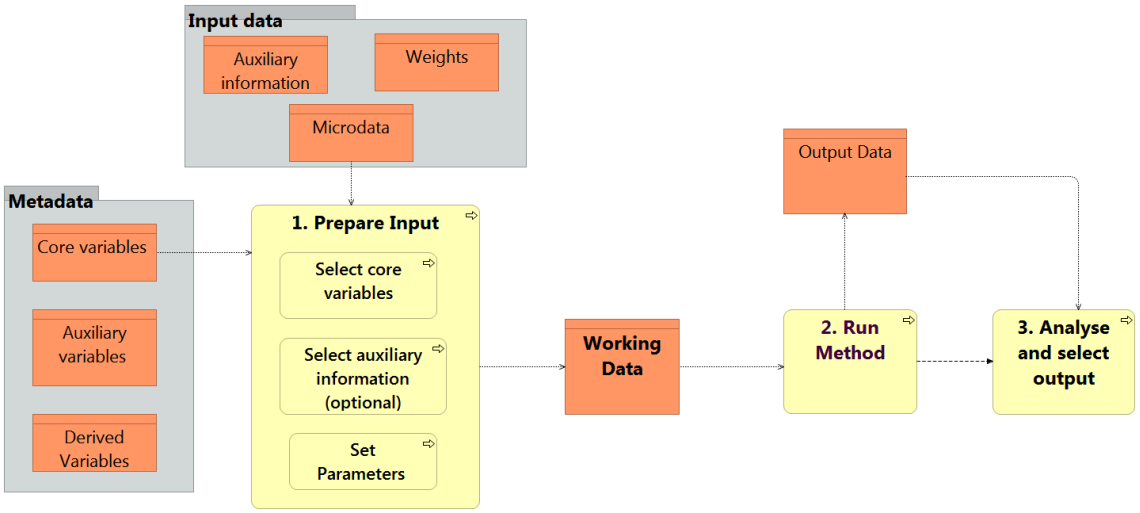

Figure 1 - Process flow of a statistical service

The model describes the main steps needed to run a statistical service:

1. **Prepare input**: create the input, by mapping initial data with standardized metadata to specify the meaning of the provided variables (e.g. identification variable, classification, core variables). The output of this step is the transformation of *input data* in *working data*.
2. **Run method**: *working data*, with their standardized data structures, can be processed by one or more iterations of the statistical method. The result of each iteration is stored in standardized data structures (*output data*).
3. **Analyse and select output**: this process step can be used to perform quality checks and/or calculate statistical indicators to assess the outcome of each iteration.

2.2. Information and Application architecture

In order to meet the above requirements, the Information architecture supporting the service implementation, should include the following components:

- **Data repository (input data)**: to store initial data, in order to avoid data uploading for each service iteration. Further, the data repository allows to run the service algorithm selecting a subset of units and/or variables (e.g. filter by territory level or business activity). Initial data are standardized by the step *prepare input*, that assigns a role and a meaning to the input dataset (e.g. independent variables, covariates, identifier variables, classification variables).
- **Metadata repository**: contains metadata to describe input/output data structures and to support generalized functionalities defined in the application layer (e.g. display input data, classify and select variables and/or units).
- **Working data**: standardized input and output data structures.

The application components needed to support the described process are:

- **Interpreter**: the core element of the architecture; it implements a generalized metadata driven model performing runtime data processing. More specifically,

the *interpreter* transforms *input data* in *working data*, by accessing the metadata repository, thus creating standardized data structures.

- **Bridge**: connects the interpreter with the algorithm/procedure implemented in different statistical languages (e.g. R, SAS), running in other environments.
- **Generalized application components**: set of generalized functionalities that allow to: visualize output reports or generic data structures (e.g. *input data*, *working data*, *metadata*); generate search filters to select units and/or variables in *input* and *working data*.

3. RESULTS: TECHNOLOGICAL ARCHITECTURE PROTOTYPE

In order to test the Information and Application architecture described above, a prototype web application is under development. The layers of this prototype are described in Figure 2:

- **Client layer**: composed by a web application, built as Single-Page Application (SPA) with Angular and Boostrap CSS.
- **Application layer**: composed by a set of microservices. Each microservice implements a specific functionality requested by the client layer, for example the visualization of *working data* or the invocation of a procedure. From a technical perspective, this component is built by using Java open source frameworks. A key element of this layer is the *interpreter*, that transforms *input data* in *working data*.
- **Engine layer**: this layer contains the runtime environments of the algorithm or procedures to be wrapped.

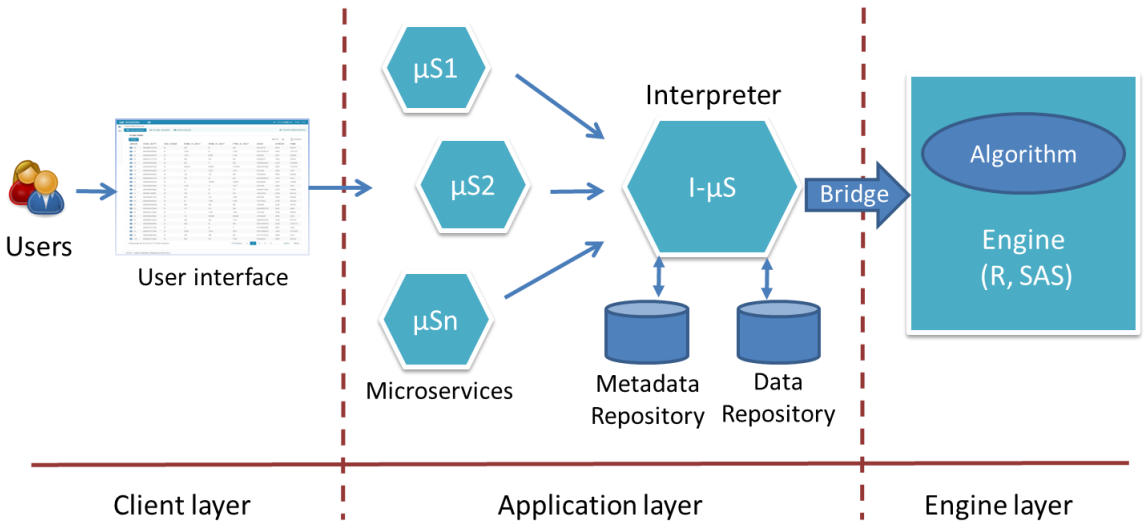

Figure 2 - Technological architecture

4. CONCLUSIONS

The architectural layers have been designed considering the following recommendations:

- Analyse what each step in the processes to implement requires from the legacy apps (see **'Data repository'** and **'Working data'**);
- Capture the relevant (configurable) data processing parameters (see **'Metadata repository'**);
- Develop adapters to make the legacy applications accessible through the selected APIs (see **'Bridge'** and **'Interpreter'**);
- Specification of the decision points and how the parameters affect the process (see **'Metadata repository'**);
- Lightweight solution to document the schema of inputs/outputs (see **'Metadata repository'**).