Working Paper.
ENGLISH ONLY

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Paris, France, 28-30 April 2014)

Topic (v): International collaboration and processing tools

On Implementing CSPA Specifications for Editing and Imputation Services

Prepared by [Monica Scannapieco, Donato Summa, Diego Zardetto, Istat, Italy]

I. Introduction

1. CSPA (Common Statistical Production Architecture) is a template architecture for supporting the industrialization of Official Statistics production processes. CSPA includes some specifications intended to define interfaces of services in a standard way, with a focus on service inputs and outputs.

2. In the paper, we describe an experience related to the implementation of the CSPA specifications for an "Error Localization" service, realized in a Proof-of-Concept carried out within an international collaboration. The realized service wraps functions that are offered by the R package "editrules" developed at Statistics Netherlands.

3. The service has been deployed on the CORE platform used by Istat for process industrialization. We also show how such CSPA service can be combined to an error correction service in order to perform a full editing and imputation process.

II. CSPA: Basic Concepts

A. Motivations of the CSPA project

4. National Statistical Institutes (NSIs) produce Official Statistics having very similar goals, hence several activities related to their production processes are common. Nonetheless, such activities are currently carried on in an independent way, almost without relying on shared solutions.

5. Statistical organizations have attempted many times over the years to share their processes, methodologies and software solutions. However, most cases of sharing have involved significant work to integrate components into different processing and technology environments.

6. As part of the modernization effort in the Official Statistics field, the High Level Group for the Modernization of Statistical Production and Services (HLG) has taken action in order to address these issues by promoting the development of the Common Statistical Production Architecture (CSPA) and its implementation.

7. During 2013, the CSPA project resulted in the production of two major outputs: a document describing the CSPA Specification [CSPA Specification, 2013] and a Proof-Of-Concept (POC) consisting of the prototype implementation of several CSPA services carried by different NSIs.

8. The 2014 follow up of the project, CSPA Implementation, is on-going and has the general objective of providing production ready implementations of a set of CSPA services. Such CSPA services have been selected in order to cover different phases of GSBPM (Generic Statistical Business Process Model), and are: (i) Classification service, (ii) Sample selection service, (iii) Editing service, (iv) Error correction service, (v) Disclosure control service, (vi) Seasonal adjustment service, (vii) Data visualization service, (viii) SDMX dissemination service.

B. The CSPA Concept

9. CSPA provides a *template architecture* for official statistics, describing:

   (a) *What* the official statistical industry wants *to achieve*
   (b) *How* the industry can achieve this, i.e. principles that guide how statistics are produced
   (c) *What* the industry will have *to do*, compliance with the CSPA

10. The principal aims of this template architecture are: (i) Provide guidance for building reliable and high quality services to be shared and reused in a distributed environment (within and across statistical organizations); (ii) Enable international collaboration initiatives for building common infrastructures and services, and (iii) Foster alignment with existing industry standards such as the GSBPM and GSIM (Generic Statistical Information Model).

11. CSPA is based on Service Oriented Architecture (SOA). Statistical services are self-contained and can be reused by a number of business processes (either within or across statistical organizations). A statistical service will perform a task in the statistical process, at different levels of granularity: (i) an atomic or fine grained service may, for example, support the application of a methodological step within a GSBPM sub process, (ii) coarse grained or aggregate services may encapsulate a larger piece of functionality, for example, a whole GSBPM sub process.

III. The “Error Localization” CSPA Service

12. In the POC initiative of 2013 CSPA project, a number of functional roles within involved statistical organizations were adopted, namely: service designers, service builders and service assemblers.

13. For each statistical service, the designer specifies the needed functionality to meet requirements. The service builders is given the statistical service definition and a statistical service specification by the service designer. The assembler takes the statistical service and integrate it according to the understanding of the needed business process, as expressed in the design documentation (see Figure 1).

14. The identification of different roles enabled the possibility of collaboration among different NSIs, testing at the same time also the re-usability of CSPA services at conceptual and logical levels. For instance, the CSPA service “Autocoding 2”, mapping a field (e.g. the answer to a questionnaire variable) to a classification code, was designed and built by Statistics New Zealand (that hence played the role of designer and builder), and was deployed by Istat on its internal platform in the role of assembler.

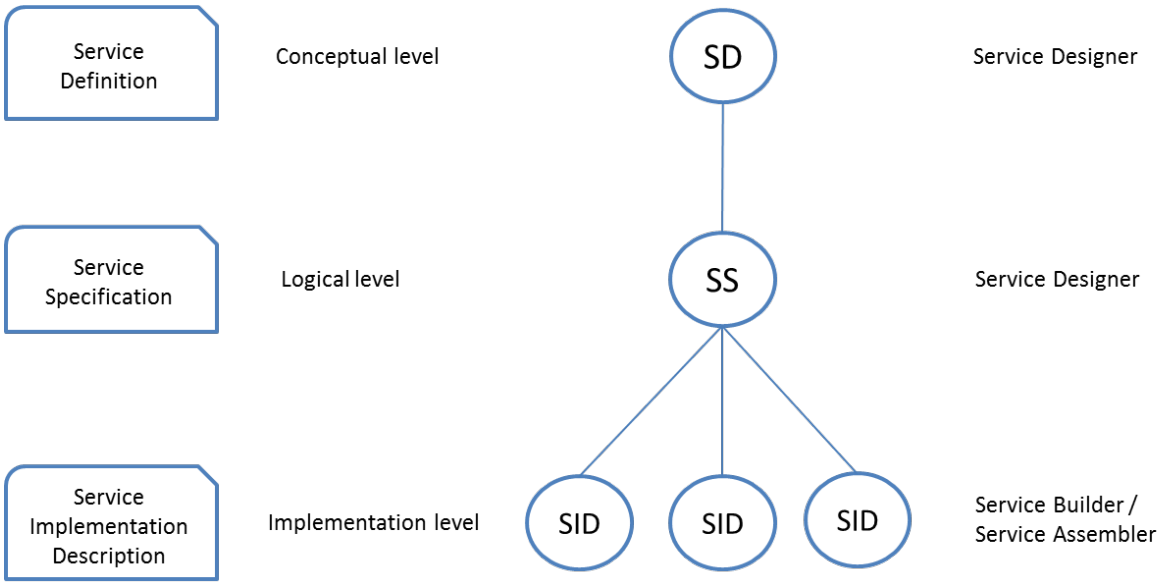

Figure 1: Different layers and roles for CSPA services [CSPA Specification, 2013]

15. Within the POC, Istat undertook the responsibility of developing the CSPA Error Localization service, with the roles of designer, builder and assembler.

16. During the CSPA Sprint held in Rome, the Service Definition and Service Specification of the service were defined in collaboration with other NSIs. Moreover, for the implementation, it was decided to wrap R package "editrules" [De Jonge, Van der Loo, 2013] developed at Statistics Netherlands.

17. The specific implementation of the service consists of a Java application that triggers a batch execution of an R script. The Java application:

    (a) Processes the input files that are conform to the Error Localization Service Specification.
    (b) Triggers a batch invocation of an R script that:
        - invokes a specific function of the editrules package named localizeErrors;
        - converts the output structure of the aforementioned function to the output structure specified by the Error Localization Service Specification.
    (c) Creates the output file on the file system

A. Issues

18. The principal issues that emerged during the implementation exercise are detailed in the following.

19. *Mixing logical and implementation layers*. The Service Specification was supposed to deal with logical level concepts. However, it also included some implementation level details, e.g.:

    (a) For the input: "The unit data set is a delimited ASCII file, using semi-colons as delimiters, using a period as the decimal separator, and as described in the attached metadata set."
    (b) For the output: "The unit data set is a semi-colon delimited ASCII file containing the case ID, followed by the Code."

    Some logical level structures should be defined for supporting the expression of implementation-independent specifications. An example of possible logical level structures is provided by: "The unit data set is a rectangular data structure whose columns can have different (logical) types". Logical types make sense because, for instance, a logical type "numeric" can be implemented by different physical types like "int" and "float".

20. *Handling Tool Parameters at Service Level.* The Error Localization Service Specification stated that input messages are expected for unit dataset, metadata set and rules. However, the function of the package editrules to be wrapped presented parameters that are not passed by the service but still have to be somehow provided. These can be either configuration parameters (set only once at assembly time) and statistical functional parameters (set on a single invocation basis: for instance, a general configuration for the labour forces survey can be invoked several times with different parameters). The solution adopted for the error localization service was to introduce a service configuration step. This is prepared by the builder that defines all the configuration properties that need to be set for the specific implementation. The assembler will provide the values for the properties related to environment configuration (e.g. file paths, working directories etc.) while the configurer and/or the user set the values for the properties related to statistical functional parameters. The additional parameters can be listed in the "Additional information" section of the Service Implementation Template.

21. *Dealing with Case ID at a Logical Level.* In the Service Specification, the GSIM message included a Case ID as Unit Identifier Component. However, we were not be able to find a way to specify in the DDI syntax which of the input dataset variables is prescribed to play the role of Case ID. As a consequence, when we had to generate the Output message with the Case ID, we assumed that it was the row number of the input unit data set.

B. Test Cases

22. Data used for test cases come from Istat's Structure of Earning Survey. Input unit data sets involve 20 variables. The Rules set consists of 44 edits involving 17 (numeric) variables appearing in the unit data sets. Three different test cases have been run adopting the same Rules set with the following features:

    (a) Unit data set with 1000 erroneous records;
    (b) Unit data set with 2000 exact records;
    (c) Unit data set with 3000 records, obtained by first appending data set 1 to data set 2 and then randomly reordering the rows.

23. Note that test case (b) has been intentionally designed to prove the robustness of the wrapper against a "degenerate" input case. The obtained output was an "empty" unit data set (i.e. zero rows), as expected.

24. In general, the three test cases were successful, thus proving the feasibility of the wrapping task with respect to:

    (a) Input and output formats
    (b) Correctness of the service behavior.

C. Building and Assembling Phases

25. The developed Error Localization service was deployed on Istat internal platform for statistical processes execution, namely the CORE (COmmon Reference Environment) platform [Bruno et al., 2013].

26. The prototype version of CORE resulted from the two ESSnets, namely CORA, carried out in 2009-2010, and CORE, carried out in 2011-2012. In 2013, Istat invested in the implementation of a production version of the CORE prototype, and the result was a fully engineered version of the platform. In particular, the CORE platform is a Web application consisting of (i) GUIs for processes, services and data design; (ii) application logic for making services interoperable, and for process execution, (iii) a back-end layer including an operational database and a metadata repository for describing services and processes define in the environment and for tracking processes execution.

27. The Error Localization service was successfully deployed on CORE platform, thus proving the fully compatibility of CSPA services with respect to a specific NSI's internal platform.

IV. Current and Future Work

28. Istat is currently involved in the 2014 CSPA Implementation project, with the role of developing the Error Correction service. Starting from the Error Localization service developed for the POC, the following activities are ongoing:

(a) study how to extend such a service in order to perform a full editing and imputation process;
(b) design a CSPA specification, to be shared and agreed among CSPA implementation project participants, for one or more services implementing such a process;
(c) implement the specifications provided at (b) by concrete CSPA services wrapping existing tools.

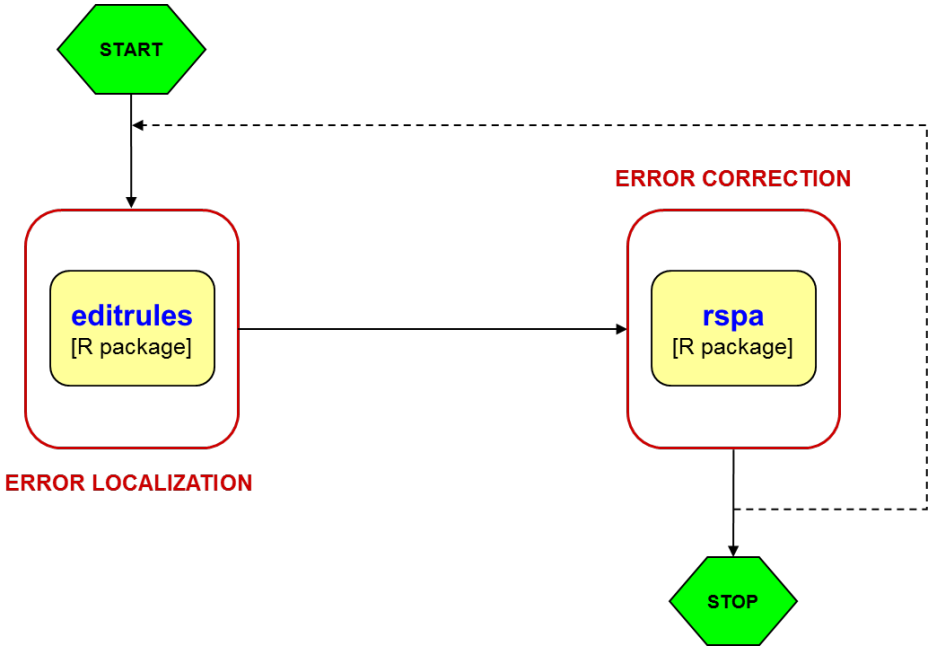

Figure 2: Composing CSPA services for Error Localization and Error Correction

29. With respect to point (c), we surveyed existing tools for error correction, and we chose as the most promising candidates the R packages "rspa" [Van der Loo, 2013] and "deducorrect" [Van der Loo, De Jonge, Scholtus, 2013], both made available by Statistics Netherlands. We have already tested "rspa" to the scope of selecting it as the inner tool to be wrapped in the CSPA Error Correction service. We have some evidence that the implementation we are working on for the two services CSPA Error Localization and CSPA Error Correction can successfully work together to perform a "full" editing and imputation subprocess.

V. References

CSPA Specification (2013): Common Statistical Production Architecture, version 1.0, available at URL: http://www1.unece.org/stat/platform/display/CSPA/CSPA+v1.0

Bruno, M., Scannapieco, M., Vaccari, C., Vaste, G., Virgillito, A., Zardetto, D. (2013). CORE: a Standard Platform for Statistical Production Processes. Proceedings of NTTS 2013, Brussels, 2013.

De Jonge, E., Van der Loo, M. (2013). editrules: R package for parsing, applying, and manipulating data cleaning rules. R package version 2.7.2., available at
URL: http://CRAN.R-project.org/package=editrules

Van der Loo, M. (2013). rspa: Adapt numerical records to fit (in)equality restrictions with the Successive Projection Algorithm. R package version 0.1-4., available at
URL: http://CRAN.R-project.org/package=rspa

Van der Loo, M., De Jonge, E., Scholtus, S.(2013). deducorrect: Deductive correction, deductive imputation, and deterministic correction. R package version 1.3-4., available at
URL: http://CRAN.R-project.org/package=deducorrect