WP. 2
ENGLISH ONLY

UNITED NATIONS STATISTICAL COMMISSION and ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Bonn, Germany, 25-27 September 2006)

Topic (i): Editing nearer the source

eSTATISTIK.core: COLLECTING RAW DATA FROM ERP SYSTEMS

Invited paper

Prepared by Michael Schäfer, Federal Statistical Office, Germany[1]

INTRODUCTION

1. This paper describes the data collection system eSTATISTIK.core and its generic facilities for validating raw data.

2. eSTATISTIK.core is an internet-based data collection system for collecting raw data directly from Enterprise Resource Planning (ERP) or other software systems managing business data. As a machine-to-machine procedure, it is complementary to on-line data collection systems using web forms. Its primary objectives are to minimise the burden of responding companies and to increase the efficiency of the statistical system. The project includes the development of survey-independent data collection procedures. This two-fold approach enables users to set up seamless and fully automated reporting and collection workflows.

3. eSTATISTIK.core consists of several infrastructure and software components:

- a single point of delivery on the web on a central data collection server, *CORE.server*
- standardised survey-independent XML[2] document types
- free software, namely the software library *CORE.connect* and the stand-alone application *CORE.reporter*

4. Raw data are validated against electronic survey definitions and then automatically forwarded to the collecting statistical office. While validation takes place automatically on the data collection server, it is optional on the client machine. In either case, validation is a truly generic process driven by electronic survey definitions in XML format and performed by the same software component.

MOTIVATION, OBJECTIVES, REQUIREMENTS

5. In recent years, national statistical offices have made considerable efforts to reduce the burden of respondents, including standardising and simplifying questionnaires and terminology, cutting down on questions and frequency, and using previously collected data. For facilitating data provision, internet-based on-line data collection has become a standard procedure that is especially suitable for surveys with

[1] Prepared by Michael Schäfer, michael.schaefer@destatis.de.
[2] Extensible Markup Language; see http://www.w3.org/XML

reasonably-sized questionnaires and modest data transmission volumes. However, its most common interfaces – data entry into web forms and file upload – are designed for manual use and lend little or no support in helping respondents to improve on their internal reporting workflow. Where raw data is transferred from machine to machine, specific software, validation reports and formats are applied. Such solutions still leave it to the respondent to organise data retrieval.

6. In 2002, the German statistical offices developed DatML/RAW, an XML-based cross-survey document type for raw data messages. DatML/RAW is used as a standard raw data interface on which automated generic server-side data *collection* procedures have been built. The question came up if DatML/RAW offered possibilities to standardise client-side data *reporting* procedures as well.

7. In the area of business surveys, a relatively small number of sources supply large volumes of raw data. Typically, these sources are either companies responding on their own behalf or service providers reporting as a third party on behalf of customers. Many of these sources use sophisticated ERP systems, mostly from one of the major vendors. Here, a chance was seen to create standardised data reporting procedures, using DatML/RAW as a common format and allowing full automation. Also, the need to report to various statistical offices simultaneously – a normal situation in the German federal statistical system – should be eliminated by taking advantage of DatML/RAW’s capability to transport any number of statistical messages with arbitrary reporting contexts.

8. These considerations matched very well with proposals made in late 2003 by the German industry to develop standard internet-based reporting procedures that would make automatic compilation and transmission of statistical reports from ERP systems possible. Several months earlier, the German statistical offices had already begun a discussion with the user community. In March 2003, a working group was established with the objective to discuss improvements of data collection procedures and to prepare a pilot for the statistics of wages and salaries. The working group represented the German statistical offices and partners from the industry, mainly through the *AWV - Arbeitsgemeinschaft für wirtschaftliche Verwaltung* (Working Party for Economical Administration). The AWV works towards improving the relationship between the economy and the public service. Its members are respondents, service providers and a total of more than 70 software producers, many of which are renowned companies like Lufthansa, Datev, SAP, Oracle and UBM.

9. The working group identified the following objectives, taking into account the needs of both respondents and the statistical system:

- technical improvements in the client-side handling of raw data, standardisation of formats, and transfer of data and metadata between respondents and statistical offices
- maximum reduction of the burden of respondents through elimination of manual work
- higher efficiency of server-side data collection procedures
- timely delivery of pre-checked high-quality raw data
- wide acceptance of the solution

10. Also, some requirements and limitations were formulated:

- respondents shall be able to pack any number of reports, in any combination of survey and addressee, into a single file, and send it to a single point of delivery (*one-stop reporting*)
- the statistical offices provide the metadata that controls eSTATISTIK.core. They must guarantee the metadata’s accuracy and validity and their ability to sustain the data collection infrastructure
- the solution has a potential to produce significant savings, but if it does so for an individual respondent depends on many factors and is difficult to predict

KEY FEATURES

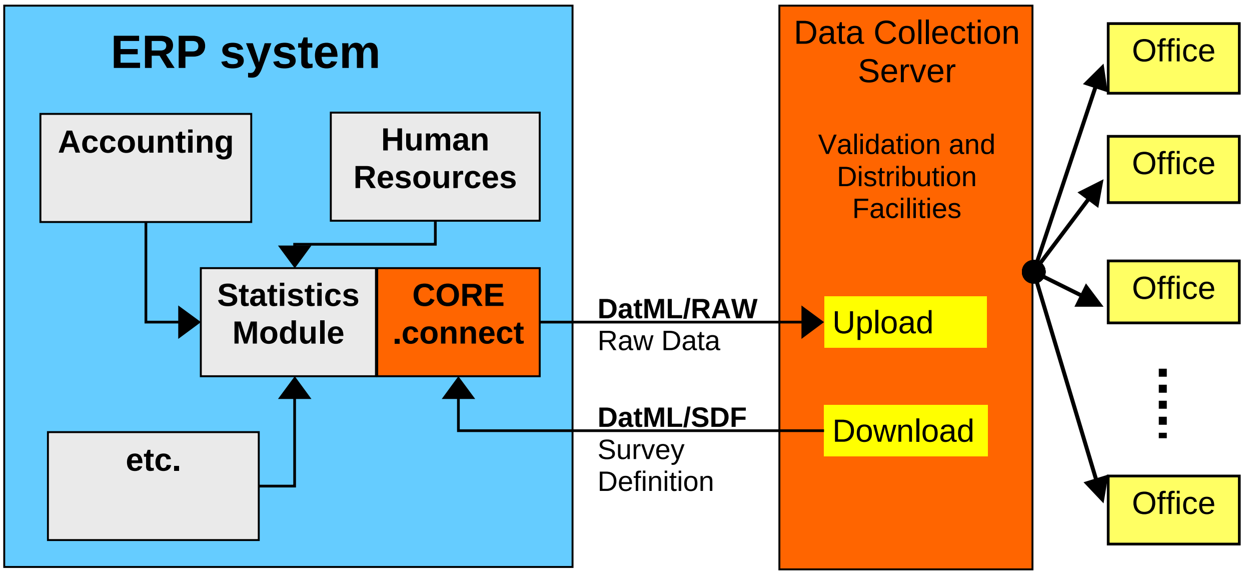
11. Bearing the above-mentioned requirements and limitations in mind, the partners involved in the project agreed that eSTATISTIK.core should centre on improvements of data and software interfaces and have the following key features:

- XML-based, generic document types for raw data messages (DatML/RAW), survey definitions (DatML/SDF) and validation reports (DatML/RES)
- a software library, CORE.connect, implemented in a variety of common programming languages, for controlling the communication between reporting and collecting systems
- CORE.reporter, a stand-alone PC application with data mapping facilities for respondents without ERP or equivalent systems
- so-called *statistics modules*, implemented by the vendors of ERP systems; a statistics module compiles a raw data message for a specific survey and uploads it to the data collection server using the CORE.connect library
- a central internet server for uploading raw data messages to a single internet address and for downloading survey definitions and validation reports
- suitability for fully automated operation

12. In extension of the technological approach described above, it has also been agreed that terminologies used in statistics and business will be harmonised to achieve a higher degree of conceptual congruence and to minimise the necessity of adapting business data to statistical concepts.

ARCHITECTURE

13. Overview:



14. The overview graphic depicts the architectural concept of eSTATISTIK.core. In the course of a data collection process, the following steps are carried out:

- after optionally downloading a DatML/SDF survey definition document, a statistics module (of an ERP system) retrieves raw data from subsystems such as Human Resources and Accounting and compiles them into a DatML/RAW document
- optionally, the DatML/RAW document is validated against an XML schema and the survey definition, and – if valid – uploaded to CORE.server
- on CORE.server, it is validated automatically and a DatML/RES validation report is generated
- valid documents are decompiled into single-message document

- messages are transformed into survey-specific formats if required
- raw data messages are forwarded to the collecting statistical office in XML form (plus the output of optional transformations)

XML DOCUMENT TYPES

15. A common property of the XML document types used in the context of eSTATISTIK.core is that they are generic with respect to surveys. Any document instance is specific to a survey only in terms of content, but not in terms of structure. Thus, it is possible to build generic applications and procedures that create survey-specific results driven alone by metadata, and without the need of changing a single line of code when another survey is added.

16. DatML/RAW is a document type for raw data messages. Its flexible structure allows a document to accommodate any number of raw data messages in any combination of survey, reference period, respondent and collecting office. Metadata can be shared among reports. Raw data is stored in a generic structure of elements representing *records*, *variables* and *variable groups*. This structure varies from survey to survey alone in the number and nesting of these elements and in the names of the objects they represent.

17. DatML/RES is a document type for acknowledgements and validation reports. A validation report contains information about the input document and describes the validation parameters and the validation results at the document, message and report level, including the type and position of errors, and the affected variable, if any.

18. A DatML/SDF document describes a survey in a formal manner. The organisational metadata it carries enables procedures to identify a survey, test if the definition is out of date and determine the period between surveys. It contains a definition of the survey data model – that is variables and variable groups, and their characteristics and dependencies – and describes how it maps onto the structure of a raw data message. The data model definition is useful for both analysing and constructing raw data messages. DatML/SDF-based survey definitions drive the generic validation process of eSTATISTIK.core.

SURVEY DEFINITION

19. The survey’s data model forms the main content of a survey definition. DatML/SDF divides the data model into an *input data model* and an *output data model*.

20. The input data model defines a survey’s variables and how these variables are organised. Variables can be grouped into variable groups, and variable groups can be nested to model arbitrarily deep and complex hierarchical structures.

21. Variables and variable groups are defined at the top level of the input data model. Where they are *used* as part of another variable group, a reference to the definition is made using the name of the variable [group] which must be unique in the scope of the survey definition. In the context of DatML/SDF, this mechanism is termed *inclusion*. It provides for maximum flexibility and easy re-use. For example, a variable group can be declared optional in one context, and required in another, without redefining it.

22. A variable definition specifies a variable’s name and defines its data type and value space. If the value space is omitted, the legal value set is the implicit value set determined by the boundaries of the data type. Vice versa, if the legal value set is smaller than the implicit value set, a value space must be specified. DatML/SDF supports the following data types:

- Numeric data types: `decimal`, `integer`, `nonNegativeInteger`, `positivInteger`
- String data types:, `normalisedString`, `string`, `token` (differing in handling of white space)
- Date and time data types: `formattedDate` (date format controlled by meta characters)

23. The following XML snippet shows a sample definition of an integer variable with a value set of 1, 2 and 3. Please note that enumerated values can be assigned a name.

```
[1]     <sdf:variable name="MeldungArt">
            <sdf:simpleType>
[2]             <sdf:integer total-digits="1" />
            </sdf:simpleType>
[3]         <sdf:valueSpace>
[4]             <sdf:enumeration>
[5]                 <sdf:enumElement name="Anmeldung">
[6]                     <sdf:value>1</sdf:value>
                    </sdf:enumElement>
                    <sdf:enumElement name="Ummeldung">
                        <sdf:value>2</sdf:value>
                    </sdf:enumElement>
                    <sdf:enumElement name="Abmeldung">
                        <sdf:value>3</sdf:value>
                    </sdf:enumElement>
                </sdf:enumeration>
            </sdf:valueSpace>
[7]         <sdf:description>Art der Meldung</sdf:description>
        </sdf:variable>
```

[1] Variable definition giving the variable's name.
[2] Data type (here: one-digit integer)
[3] Value space
[4] Enumerated list of value elements
[5] Enumerated named value element
[6] Value of the value element
[7] Description of the variable

24. The output data model maps the variables and variable groups of the input data model onto the logical structure of a raw data message. This information is useful for both message construction and validation. There are two structural elements that basically govern the scope and occurrence of variables: *message data groups* and *case data groups*.

25. The definition of a message data group is optional. If present, there must not be more than one. As its name suggests, the scope of a message data group is the entire message. Consequently, a variable included into a message data group may not occur repeatedly. In a single raw data message, only one value can be supplied to such a variable. Message data groups are often used to provide information about respondents such as identifiers and geographical information that cannot be accommodated by the regular generic structures of DatML/RAW. Currently, variable groups cannot be included into a message data group.

26. A survey definition must contain at least one definition of a case data group. The idea behind case data groups is to allow sets of variables to occur as many times as needed, each representing a case or object that is observed individually. For example, in the intra-community trade statistics, each good traded between EEC countries represents a statistical object for which data must be reported separately. Both variables and variables groups may be included.

27. Like variable groups, message and case data groups are composed by inclusion. Each of these groups represents an *inclusion context* with one or more *inclusion nodes*. Variables and variable groups are *inclusion objects*. Since variable groups can include other variable groups, a tree-like *inclusion hierarchy* is created with a case data group as the root node. The path from the root node to an inclusion node is referred to as the *inclusion path*.

28. Inclusion can be *required*, *optional* or *conditional*. It can also be *forbidden*, if so specified for the case of a condition evaluating to false. A condition is defined in an inclusion context and can apply to all inclusion nodes visible in that context. These are the inclusion nodes local to the context and the ones directly or indirectly included, in other words, all inclusion nodes of the partial inclusion tree with the inclusion context node at its root. These nodes are also referred to as *target nodes* of a condition. If two or more conditions apply to the same inclusion node, only the one highest in hierarchical order is evaluated. By processing an inclusion node, it is determined if and how many instances of an inclusion object can be generated during message construction or must be present during message validation, at the minimum or maximum, respectively.

29. Conditions can be defined in the form of *preconditions* and *dependencies*. Preconditions are purely descriptive and refer to conditions that are subject to human evaluation rather than they are machine-processible. They are especially suitable for display to human users. Dependencies describe relations between inclusion objects and can be calculated by software to yield true or false. A dependency has a `notation` element that contains a condition expressed in the *Data Edit Specification Language* (see below). It is possible to test if an instance of an inclusion object (other than the one to include) exists and to compare its value against a constant or a value of another inclusion object. The most common dependencies are:

- An inclusion object must be present if another one is present or has a specific value.
- An inclusion object must not be present if another one is present or has a specific value.
- An inclusion object must be present if another one is not, and vice versa.

30. The maximum and minimum numbers of occurrences of a variable group is controlled separately at each inclusion node, permitting it to vary depending on the inclusion context. For case data groups, being top-level nodes and therefore having no explicit inclusion context, these properties are specified in the group’s definition.

31. When a set of variables and variable groups is processed and there are two or more definitions of message data groups, it must be unambiguously assigned to a case data group. This is done by means of *selectors* and *identifiers*. Each selector and each identifier references a variable. Identifiers must also specify a value that conforms to the variable’s data type and value space. Selectors are declared at the output data model level, identifiers at the group level. For each group, it is tested if the variables’ values match those of the identifiers. It is not permitted that more than one group meets the selection criteria.

32. A sample message data group definition:

```
[1]  <sdf:dataGroup name="Anmeldung" required="no">
         <sdf:identifiers>
[2]          <sdf:identifier variable="MeldungArt">
                 <sdf:setValue>1</sdf:setValue>
             </sdf:identifier>
         </sdf:identifiers>
         <sdf:conditions>
[3]          <sdf:precondition enumerated-value="if-available">
                 <sdf:description>If data are available</sdf:description>
                 <sdf:targetNode>GemeindeZusatz</sdf:targetNode>
                 <sdf:targetNode>FrueheresGewerbe</sdf:targetNode>
             </sdf:precondition>
[4]          <sdf:condition>
[5]              <sdf:notation class="PL">
                             MeldungArt /= 2
                 </sdf:notation>
[6]              <sdf:targetNode>
                     Gewerbe.BetriebArtIndustrie
                 </sdf:targetNode>
                 <sdf:targetNode>
```

```
                                 Gewerbe.BetriebArtHandwerk
                          </sdf:targetNode>
                          <sdf:targetNode>
                                 Gewerbe.BetriebArtHandel
                          </sdf:targetNode>
                          <sdf:targetNode>
                                 Gewerbe.BetriebArtSonstiges
                          </sdf:targetNode>
                    </sdf:condition>
             </sdf:conditions>
[7]          <sdf:includeVariable variable="MeldungArt"
                    required="yes"
                    />
             <sdf:includeVariable variable="IstKorrektur"
                    required="yes"
                    />

             [omitted inclusion nodes]

             <sdf:includeVariableGroup variable-group="FrueheresGewerbe"
                    minimum-occurrences="0"
                    maximum-occurrences="1"
                    />
      </sdf:dataGroup>
```

[1] Definition of a case data group
[2] Definition of an identifier; the referenced variable `MeldungArt` must have a value of 1.
[3] Definition of a precondition for variables `GemeindeZusatz` and `FrueheresGewerbe`.
[4] Definition of a dependency.
[5] Condition expressed in the Data Edit Specification Language.
[6] List of inclusion target nodes. Note that the target nodes are in the included variable group `Gewerbe`.
[7] Begin of the inclusion node list.

VALIDATION LEVELS

33. Validation of a DatML/RAW document takes place at two main levels: the structural, or document type level, and the subject matter, or content level.

34. At the structural level, a DatML/RAW document instance is validated against the DatML/RAW XML Schema[3] definition plus a set of a few structural constraints that are defined in the DatML/RAW specification but cannot be expressed in an XML Schema definition. At this stage, almost all of the validation work is done by the XML parser.

35. At the content level, each raw data message is validated against a formal survey definition supplied by a DatML/SDF document. A content-level validation yields errors if variables are missing, unknown or have illegal values, of if metadata such as the reference period are incorrect. Since a DatML/RAW document may contain any number of raw data messages varying in survey and reference period, more than one DatML/SDF document may be required for the validation of one document.

CLIENT AND SERVER-SIDE VALIDATION

36. On CORE.server, validation is performed automatically by a software module called *Inspector*. It is part of a server system called *KonVert – Konvertierung und Verteilung*, a term that roughly translates to “transformation and forwarding”. KonVert is written in the Java programming language.

[3] XML-based language for defining XML document types; see http://www.w3.org/XML/Schema

37. When processing a DatML/RAW document message by message, Inspector first finds the matching survey definition and then validates the raw data message against it. For efficiency, survey definitions are read once and compiled into Java objects, and re-read only if Inspector is signalled to do so. The validation method returns Java objects representing an *inspection report* with zero or more *inspection problems*. For practical reasons, a maximum of one hundred inspection problems are reported.

38. Inspector analyses the inspection report, but does not produce a validation report. Instead, it passes validation information to another module, *Reporter*. Reporter collects validation information and generates a DatML/RES document at the end of the KonVert processing pipeline, which is then passed to the on-line data collection system IDEV. IDEV is the central interface to respondent management and enables users to view and download validation reports.

39. Since the release of version 1.1., CORE.connect includes Inspector. Software systems that integrate CORE.connect can validate DatML/RAW documents by means of the Inspector API. Contrary to server-side validation, no DatML/RES document is generated. Instead, Inspector returns an inspection report object as described above.

SUPPLYING THE METADATA

40. Because it is fully metadata-driven, eSTATISTIK.core is a flexible and easily extensible system. On the other hand, running it requires a metadata management that ensures that metadata can be supplied without undue effort and in high quality. Therefore, several tools have been – and are – developed to create metadata for eSTATISTIK.core, and a new metadata management system is being built up. Their design is based on the common approach that they shall enable subject-matter specialists to work with objects of their domain and in ways they are conceptually familiar with, creating at the same time re-usable metadata that can drive IT procedures.

41. *STATSPEZ*[4] was originally designed for specifying, creating and executing tabulation programs and has been recently extended to support data collection as well. In this process, two central components have been added that supply most of the metadata for eSTATISTIK.core: *Data Edit Designer* ("PL-Editor") and *Survey Definition Editor* ("SDF-Editor"). Another important source of metadata is *Data Set Designer* ("DSB-Editor") that supplies definitions of data set structures in the conventional form of records and fields.

42. Data Edit Designer has been developed for specifying data edits. The necessary data model, including the definition of statistical variables and their arrangement into structures, is also described with this tool. It forms the basis of the data model defined in survey definitions. The Data Edit Specification Language is used in survey definitions to express dependencies between variables.

43. Survey Definition Editor has been designed for creating survey definitions. It uses data model definitions created with Data Edit Designer. In many cases, only little re-modelling is necessary, if any. Sometimes, variables are added that are collected for organisational, informational or other purposes only but are not subject to data editing. Using data set definitions from STATSPEZ, users can create mappings of survey data models onto flat file structures. These mappings drive transformations of raw data messages into survey-specific data formats.

44. In the context of eSTATISTIK.core, survey definitions and data set definitions (for converting DatML/RAW documents into flat files) are the primary metadata resources. In the context of a complete survey, however, many more metadata resources have to be dealt with. For this reason, a new metadata management system is being built up to make metadata management feasible through the whole life cycle of a survey, and beyond. This system uses three conceptual elements for organising metadata: statistics, survey and resource. When an instance of any of these elements is released, it is assigned a unique

[4] Statistische Tabellenspezifikation; see http://www.statspez.de

identifier. This identifier allows managing resources in the context of a survey, and includes version control.

FURTHER DEVELOPMENT

45. It is planned to extend CORE.connect with programming interfaces that eliminate the need to deal with document type issues. Instead of creating an entire DatML/RAW document and then passing it to CORE.connect, applications shall be able to pass data only and can rely on CORE.connect to construct a valid document.

46. In DatML/SDF, the Data Edit Specification Language will be replaced by an XML equivalent so that dependencies can be formally processed with standard XML tools. Currently, the evaluation of a dependency yields true or false, resulting in a variable or variable group being required or not. A logical extension would be to return a value or a value range so variable’s value space could be controlled as well.

47. An XML input data base is under development. It will serve as a repository of raw data messages, providing interfaces to both human users and applications for managing and accessing raw data. The main focus is on supplying services, metadata and raw data to procedures like respondent management, data editing and archiving. The XML input data base has a great potential to develop into a central cross-survey raw data management facility connecting data collecting with subsequent production processes.

48. To ensure the success of eSTATISTIK.core, further marketing and promotion activities will be conducted by the statistical offices, the AVW, users, business organisations, and software vendors.

CONCLUSION

49. eSTATISTIK.core is a new approach to improving data collection from businesses. It relies on standardised, XML-based raw data interfaces and metadata objects. They make it possible to set up generic, automated reporting and collection procedures that considerably reduce the burden of respondents in terms of both time and costs, and at the same time improve the efficiency of the statistical system and the quality of its products. However, its success depends as well on contributions and support of the user community and software vendors.

50. As a procedure connecting machines, its data editing capabilities are (currently) limited to data validation. However, eSTATISTIK.core performs validation in a new manner: survey-independently, fully automated and controlled by metadata, and integrated into a consistent and complete metadata management.