

**UNITED NATIONS STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE**

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Ottawa, Canada, 16-18 May 2005)

Topic (ii): Implementing editing strategies and links to other parts of processing

LINKING DATA EDITING PROCESSES BY IT-TOOLS

Supporting Paper

Submitted by the Federal Statistical Office, Germany¹

I. Introduction

1. In 2002 the Federal Statistical Office of Germany (Destatis) started to re-structure its data editing process. The focus of its ambitious new data editing concept is on enhancing both the planning and the execution of the underlying work steps. In a prior evaluation of the previous practice, a variety of sub-processes was identified. By re-arranging these sub-processes, the crucial role of tailor-made software in linking related sub-processes became apparent, and the provision of software tools was considered the tie between realising methodological improvements on the one hand and optimising work flows on the other (Wein 2004, 2005, Kuchler 2003).

2. The integration of data editing processes by means of appropriate software is primarily based on the provision and accessibility of metadata in the entire survey process. Advanced survey methods generally depend on metadata from previous steps of processing and at the same time produce metadata on their part. Since most of these methods are implemented in specific software tools, it is obvious to establish the required metadata streams by providing well defined interfaces and routines for activating them. Engstrom and Granquist (1999) were the first to propose so called Process Data Subsystems that are intended to allocate information about methods applied to a survey at hand and their particular performance. Since then several approaches and implementations extended this starting point towards the provision of general metadata information systems which are available in any step of the survey process (for recent examples see the description of the SIDI system at ISTAT by Brancato et. al. 2004 or the CORD and CORM system of the ONS briefly introduced by Tate 2003).

3. The focus of this paper is rather on the integration of survey process on the level of application software than on the description of general depository systems. Point of reference is the so called PL-Editor that is a key element of the software strategy flanking the new data editing concept at destatis. It is intended to support almost any specification task to be accomplished in preparing the data editing process of a particular survey. By this means the PL-Editor is one of the central gates through which metadata enter the survey process. In order to make the PL-Editor satisfy existing and future information claims raised by related processes of data production, much attention was paid on the specification of its interfaces with respect to an office-wide software standard. This paper treats the PL-Editor with respect to problems of linking processes on both the data editing level and the general data production level, as introduced by examples in section three and four. In preparing this, the following section provides an outline of basic concepts of the PL-Editor. Finally the fifth section reports some experiences made in launching the PL-Editor in different surveys, and gives an outline of its future prospects.

¹ Prepared by Carsten Kuchler and Corina Teichmann; {carsten.kuchler, corina.teichmann}@destatis.de

II. Basic Concepts of the PL-Editor

4. The PL-Editor is intended to support almost any specification task concerning the physical and logical description of data structures and processes related to data editing like particularly the compilation of data set descriptions, data checks and sequences of data checks. Specifications are stored in an office wide depository system and are made available by direct data base access, specific XML-based interfaces common to any software tool recently developed by destatis or automatically generated source code to be processed by other software tools. By this means specifications of an ongoing survey are utilised throughout the entire survey process, and specifications of completed surveys can easily be included in current application flows. Thus the PL-Editor as part of the integrated office-wide software standard supports the reuse and harmonisation of specifications and increases the efficiency of the executing department.

5. The PL-Editor mainly consists of four interrelated components, as shown in figure 1, each of them conceptually independent from the others: (a) the actual editing interface as illustrated in detail by a screenshot in figure 2 provides input fields for the specification tasks, (b) the data base interface transfers metadata entered by the PL-Editor to an office wide depository system, (c) a collection of source code generators allows for generating metadata driven source modules and finally (d) an interface layer organises the complete data exchange between these components. This section mainly provides an outline of the basic concepts of the PL-Editor. Embedding and transferring metadata and the functionality of the source code generators is discussed in the subsequent sections in their context of application.

6. Due to the strictly object-oriented office-wide software standard (like any recent destatis software, the PL-Editor is implemented in Java), modelling a survey by means of the PL-Editor is an object-oriented task. Metadata entering the system via the editing interface are thus considered instances of objects. Figure 3 reveals the aggregation of the involved objects and the cardinality of their (non total!) part-whole-relationships.

7. The atomic object is a variable. The essential attributes of a variable are its type (integer, real, date, string, etc.), storage information like field length, its domain (set of legal values and missing codes), variable and value labels, etc. Data checks and associated correction instructions that solely refer to the modelled variable (like domain and coding checks) are considered methods of the variable object.

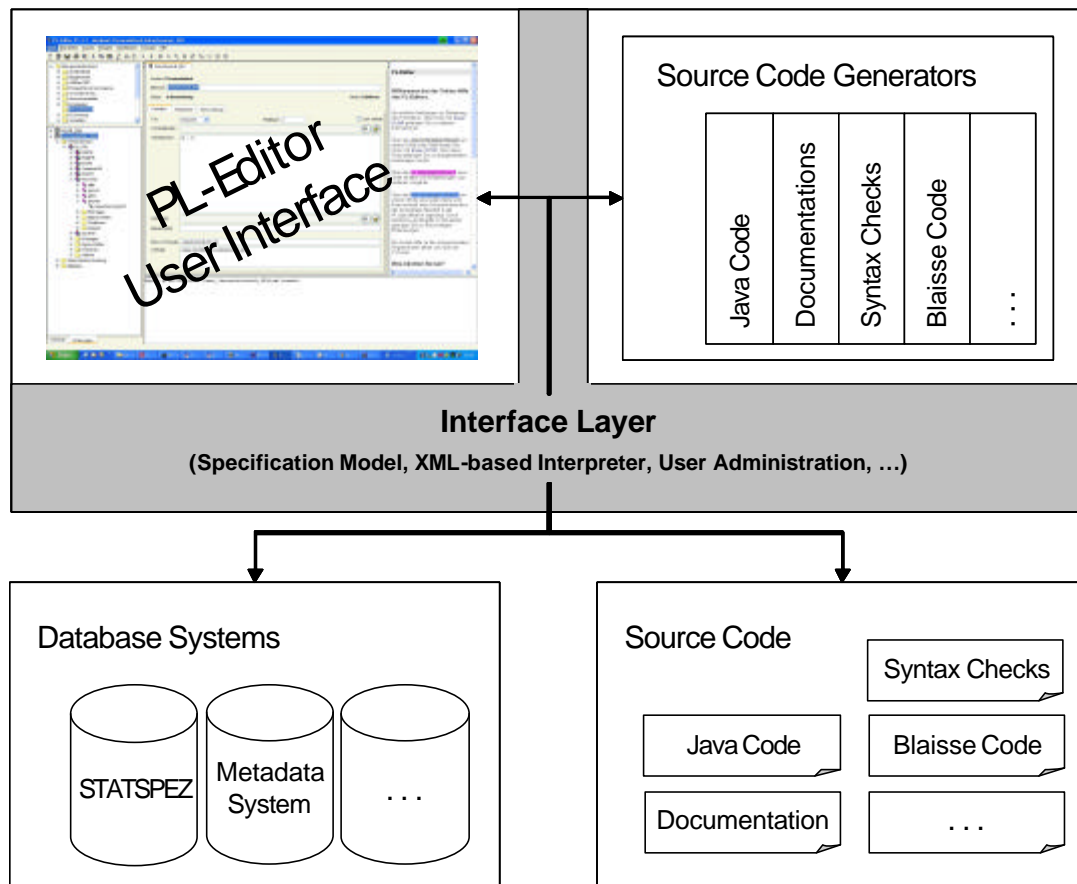


Figure 1: Components of the PL-Editor

8. On the next higher level of aggregation, so called topics combine variables and topics of lower aggregation. Topics are basically disjoint sets of variables sharing a context of content (like item blocks or sections in a questionnaire). Figure 4 shows an example of the aggregation of two topics by a superior topic. The resulting contexts induce relational data checks referring to at least two of the subsumed variables. Relational data checks and the associated correction instructions are considered methods of a topic. Combining topics thus allows for structured modelling of data checks of increasing complexity. Finally a major topic, i.e. the one that is not covered by another topic, refers to the entire survey.

9. The selection and order of data checks applied to a particular data set may vary due to the intended product, i.e. releasing micro data requires another sequence of data checks than the computation of aggregates published in first release, etc. A natural way of dealing with this problem in terms of object orientation is to provide sequences of data checks as an additional method of topics.

10. Each specified instance refers to a one-to-one identifier according to an office wide catalogue of surveys within the program of official statistics. Following the aggregation arrows in figure 3 in reverse direction or de-referencing the corresponding inclusions, this key attribute allows for the access to the instances of the subsequent objects. Thus the PL-Editor may be considered a user interface for the convenient enter of editing rules and data set descriptions as well as a front end of the underlying data bases and metadata streams. By this means specifications from already completed data editing processes can easily be included in current application flows and thus support the reuse and harmonisation of specifications. Additionally the specifications can be made available throughout the entire process of data production and for the respective software applications.

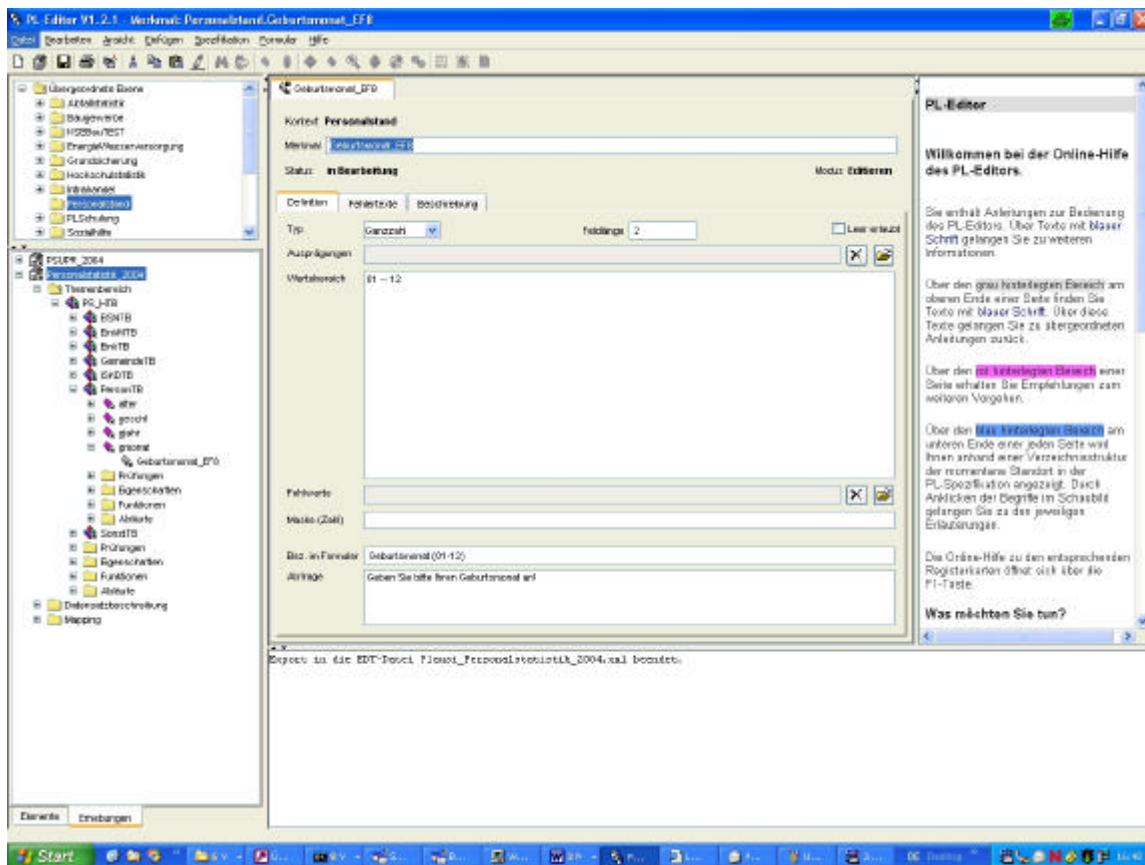


Figure 2: Screenshot of the PL-Editor, showing a register card of a variable that provides facilities for specifying the data type, the field width, the domain and the missing declarations of the variable. The two boxes on the left allow for the navigation between surveys (upper box) and within the selected survey. The box on the right shows the online help for the current work step (specifying a variable). The footer contains a box with log information about the execution of functions.

11. Data checks are specified in a script language specific to the PL-Editor that provides advanced syntax constructs like for- and while-loops, the definition of functions and data structures and particularly facilities for de-referencing external data sources. The deposition of the specified instances takes place in an interface layer that connects the PL-Editor with the office wide depository system and the source code generators (see figure 1). Since the interface layer conceptually separates the specification syntax from the data structures the specified instances are processed in, the actual specification of instances remains unaffected by any change of the application flow. The price to be paid for this encapsulation is a rather advanced definition of the interface layer and the associated interfaces of counterpart software. This is done due to an office wide standard for data base structures and XML-based interfaces that is subject to the next sections.

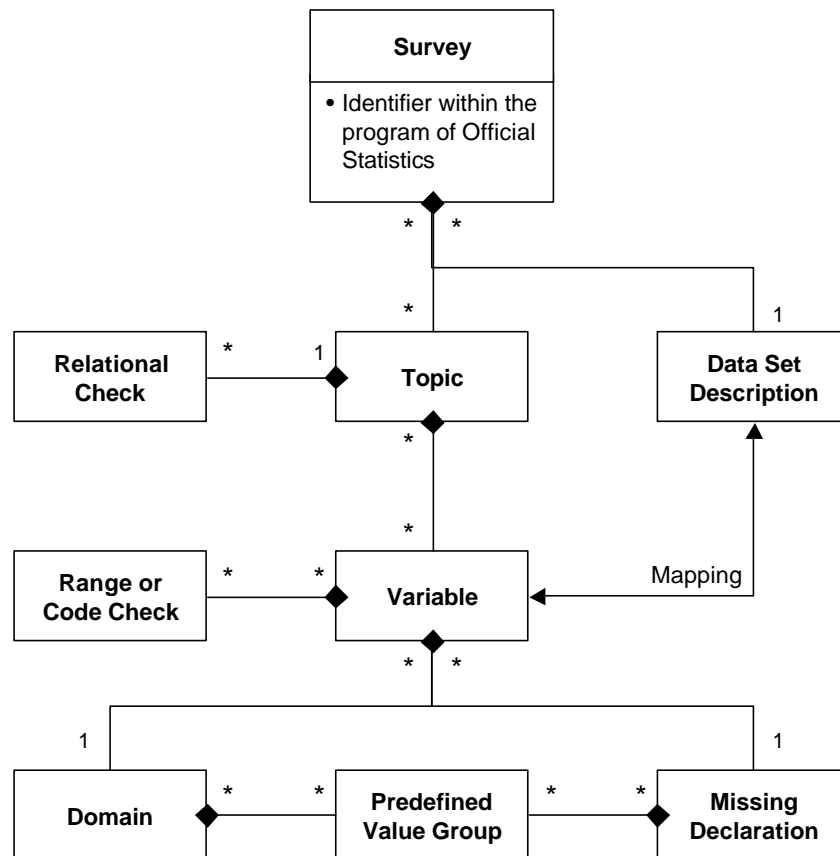


Figure 3: Aggregation of the objects provided by the PL-Editor (incl. cardinalities of the non total part-whole relationships, where “*” denotes an arbitrary natural or null).

III. Linkage of Data Editing Processes by the PL-Editor

12. The logical and relational structures of any instance specified with the PL-Editor are described in a specific XML-based format called DatML/EDT, that is assigned to the interface layer and thus separating specifications from the data structures they are subsequently processed in. The term DatML (Data Markup Language) denotes a family of XML-based formats and document-types for respectively describing any type of data and metadata occurring in the entire survey process (from surveying to archiving data). The provision of DatML formats based on a common XML core allows for a consistent and integrated (meta-)data flow. Beside DatML/EDT that was specifically developed for the interface layer of the PL-Editor, for instance DatML/RAW was implemented for handling and exchanging raw data or DatML/SDF is used for the description of survey properties with respect to automated processes.

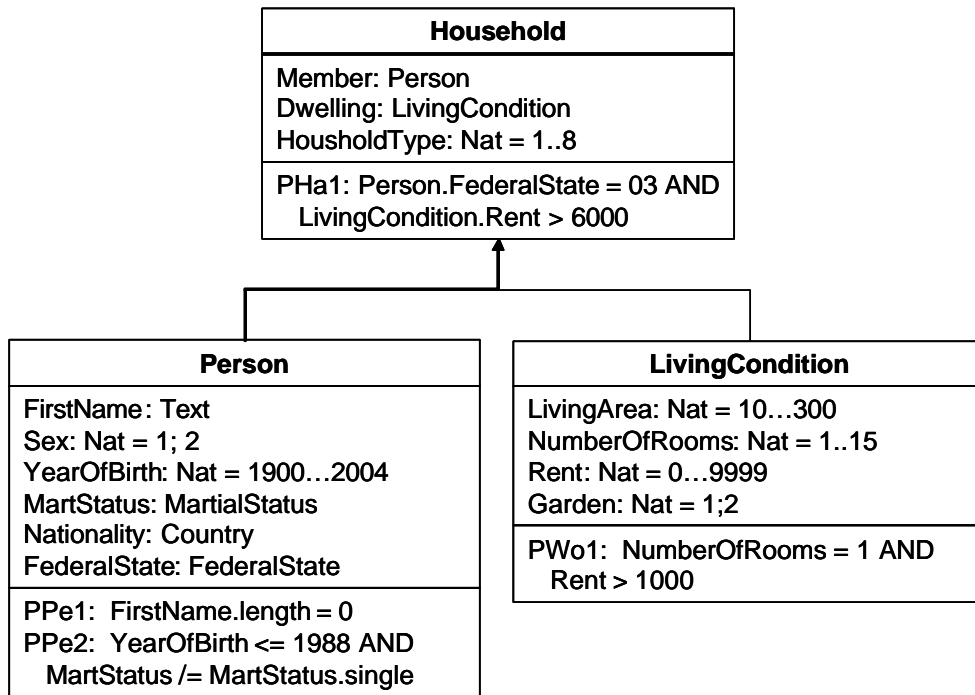


Figure 4: Imaginary example for the aggregation of topics (“Person” and “LivingCondition”) with the associated fields and data checks by a superior topic (“Household”). Note the de-referencing of fields by the data checks of the “Household” topic.

13. As illustrated in figure 1 the interface layer passes the DatML/EDT descriptions to office wide depository systems or to source code generators. Based on the DatML/EDT descriptions of the instances and some meta-information which is also available in XML format, the source code generators produce source files executable by target applications. A typical use case is the generation of Java classes that are simply included by other programs written in Java as well. But the generality of the XML-based interface layer in principle allows for arbitrary target programming languages and formats.

14. A basic example for the application of source code generators is the provision of documentation facilities for the PL-Editor. For this purpose a documentation-generator is available that transfers the descriptions of the instances from the DatML/EDT format into the csv-format common to Microsoft Office products. Another application within the PL-Editor is the generation of consistency and syntax checks applied to the specified data checks that are also implemented by an automatic code generator and then returned to internal check functions.

15. A rather advanced example for the use of source code generators is the generation of executable source code that implements the data checks related to a survey at hand. This functionality is particularly useful when specifications of data checks or their sequence of application are changed in the data editing process. Typical target applications are interactive correction programs and software for the automatic identification of erroneous values. Due to the object-oriented software standard of destatis, the recent interactive correction applications are written in Java. So it is natural to make available the data checks and their sequence of application by appropriate Java classes that are embedded into the target application. For this purpose the PL-Editor disposes of a Java code generator that receives the descriptions of the data checks and their sequence of application from the interface layer in DatML/EDT format as input and produces the according Java classes. The Java code generator already proved its reliability in several surveys (see the fifth section for an overview of use).

16. Due to the new data editing concept introduced at destatis, a software tool for the identification of erroneous values in an observation is currently implemented. According to an approach proposed by de Waal and Quere (2003) a minimal missing pattern is computed for any observation that violates at least one relational data check. Since the underlying algorithms are of high complexity, they need to be implemented in a programming language that allows for faster processing than Java. For this purpose the prototype implementation was written in C. Applying this programme in a data editing process, where data checks and their sequence of application are usually reviewed for several times, requires the provision of a C code generator that produces the according C modules at the push of a button.

IV. Linkage of Survey Processes by the PL-Editor

17. The focus of the previous section was on examples of integrating applications within the data editing sub process. This section widens the focus and deals with the linkage of the data editing with other survey processes by metadata streams provided by the PL-Editor. The exchange of metadata via the interface layer of the PL-Editor is in principle open for any counterpart software disposing of the required XML-based interfaces or having direct access to the underlying data base. Since the recent applications at destatis were programmed due to the office-wide software standard, various survey sub-processes are already supported by metadata entering the application flow via the PL-Editor. In this section we give two examples concerning exchange of metadata in XML format via the interface layer and a third example for direct data base access.

18. A typical field for re-using variable and data check specifications outside of data editing processes is the design of electronic questionnaires. Once the variables and data checks are entered, the basic structure of an electronic questionnaire can be considered fix. Even though the PL-Editor is intended to support data editing that does not mean that the respective specifications need to happen right before or within this process. In the majority of cases these specifications can be done prior to the process of questionnaire design. The IDEV software by destatis supports the design of electronic questionnaires by processing metadata entered by the PL-Editor. Of particular interest is the PL-Editor facility to treat sequences of data checks as methods of topics. Due to methodological considerations one may select a subset of data checks specific to the purposes of the electronic questionnaire which is then related to the associated fields of the electronic questionnaire. In subsequent stages of data editing one may apply other sequences of data checks. The metadata are released by the PL-Editor via the interface layer in the DatML/EDT format, and are entering the IDEV system via an XML interface also based on DatML/EDT. The resulting questionnaires are described in a XML-based format belonging to the DatML family as well which is called DatML/ASK.

19. This year the Federal Statistical Office and the Statistical Offices of the Länder released the so called .CORE software package that provides a variety of tools and transfer formats for surveying enterprise data straight from the accounting software used by a particular enterprise. The exchange of the questionnaire (which is a rather outmoded term in this context) and the data is carried out by save internet connections. The obvious effect of this approach is a significant increase of data quality both in terms of accuracy of the surveyed data and the efficiency and timeliness of subsequent steps of data production. In addition the respondents' burden is reduced to a minimum. In close collaboration with well-known suppliers of accounting software and representatives of involved enterprises, multiple XML-based exchange formats were agreed, each of them fitting specific purposes (one of them is the DatML/RAW format mentioned above). These formats became part of the DatML family and are supplemented by specific software tools realising the data exchange from the first request to a final acknowledgement of receipt. Any specification of variables and data checks necessary for the particular enterprise survey is done with the PL-Editor. Since each of the subsequent processes relies on members of the DatML family which are all based on the common XML-interface standard, these metadata are directly available via the interface layer of the PL-Editor. So changes of specifications only result in a re-integration of the associated metadata in subsequent applications. The .Core project is the 2005 winner of the 5th eGovernment competition in the field of "Economy and Labour", which is arranged by the international consulting firms BearingPoint and Cisco systems.

20. Finally the example of the so called STATSPEZ software illustrates the reuse of specifications by direct data base access. STATSPEZ is a client/server application for specifying, generating and presenting results of statistical analysis in a unique manner and due to the standards of the publication program at destatis. These tasks require specifications of the variables to be analysed like data set descriptions, storage information, domains, etc., which are all provided by the PL-Editor. As shown in figure 1, the interface layer of the PL-Editor directly feeds a data base underlying the STATSPEZ application. The data formats and structures required by the data base are due to a predefined standard and can thus be generated by the interface layer of the PL-Editor.

V. Introducing the PL-Editor at Destatis

21. The PL-Editor was released for use at destatis in July 2004. About three month earlier a preparatory training program started that still continues. Since the PL-Editor is a rather complex application, only staff members are trained that are concerned with surveys that are to be restructured with respect to the application of new software tools and editing methods. At the moment the PL-Editor is applied in about twenty surveys, with twelve of them generating an electronic questionnaire. Since the reorganisation of surveys is going step-by-step, the gain of efficiency achieved by sharing and re-using specifications in multiple surveys related by a similar context of content is not yet the major improvement realised by the PL-Editor. At the current stage of launching, departments chiefly benefit from the facilities for entering and administrating specifications with the PL-Editor and their high-grade availability in other survey processes. In particular the departments appreciate that changes in the specifications of data checks do not any longer require a re-programming of highly nested if-else constructions but only re-generating and re-embedding of source files into a target applications of subsequent processes which is regularly done within minutes.

22. However, with an annual survey, there may be periods of several months where staff does not work with the PL-Editor. In order to meet this “once-in-a-blue-moon” problem, the PL-Editor provides multiple help and documentation facilities. Of particular importance is the context sensitive online help in the right third of figure 2. It provides suggestions for work flows that can be seized by hyperlinks calling the demanded views and functions. In addition the F1-key is associated with a context sensitive help function providing information about currently activated input boxes and links to the intranet pages describing the method or work step in question.

23. Further developments of the PL-Editor are primarily related to the improvement of internal functionalities and the ergonomics of the application. Due to the strict separation of specifications and data structures realised by the PL-Editor, changes and enhancements in the linkage of processes and their associated software tools do not affect the actual PL-Editor, but only the interface layer and the underlying DatML formats. The primary enhancement to be released in next update versions of the PL-Editor is the provision of a modular testing environment for performing extended syntax and semantic checks on specifications in general and on data checks and their sequences of application in particular.

References

- Brancato, Giovanna, Concetta Pellegrini, Marina Signore and Giorgia Simeoni (2004): Standardising Evaluating and Documenting Quality: The Implementation of Istat Information System for Survey Documentation – SIDI. Proceedings of the European Conference on Quality and Methodology in Official Statistics (Q2004), Mainz.
- De Waal, Ton and Ronan Quere (2003): A Fast and Simple Algorithm for Automatic Editing of Mixed Data. In: *Journal of Official Statistics*, Vol. 19 (4), 2003, 383-402.
- Engström, Per and Leopold Granquist (1999): Improving Quality by modern Editing. UNECE, Work Session on Statistical Data Editing 1999, Rome, WP.23.
- Kuchler, Carsten (2003): IT Tools for an integrated Data Editing Concept. UNECE, Work Session on Statistical Data Editing 2003, Madrid, WP.19.

Tate, Pam (2003): The Data Editing Process within the new Statistical Infrastructure of the Office for National Statistics. UNECE, Work Session on Statistical Data Editing 2003, Madrid, WP. 9.

Wein, Elmar (2004): Improvement of Data Editing Processes. Proceedings of the European Conference on Quality and Methodology in Official Statistics (Q2004), Mainz.

Wein, Elmar (2005): Concepts, Materials, and IT Modules for Data Editing of German Statistics. UNECE, Work Session on Statistical Data Editing 2005, Ottawa, WP.37.
