Electronic Data Reporter

G.W. de Bolster, Statistics Netherlands

1. Introduction

The Electronic Data Reporter is a 99.99% Blaise build data collection module. It's build in Blaise 4.6 using part of its new functionality created mainly because of this development. Only the launcher was created as a small C++ program. The module is multi data collector, multi survey, multi questionnaire, multi data communication, multi user, multi language, multi statements and whatever multi ... you can think of. It is built to be as user friendly as possible and it's operational for CBS since the beginning of this year.

The Electronic Data Reporter has been developed to replace the EDISENT[8] module. This module, also a development by Statistics Netherlands, was build for a 16-bits environment and therefore difficult to maintain and extend. Although it could handle numerous questionnaires it also had several restrictions. With the introduction of the Electronic Data Reporter most of these restrictions are eliminated. Especially the use of the Blaise language for the definition of the questionnaires extended the possibilities for questionnaire design enormously as all the flexibility of Blaise can be applied. Just like EDISENT the Electronic Data Reporter must be installed at the respondent. After installation surveys can be send by the data collector e.g. by e-mail and automatically imported in the Electronic Data Reporter environment. Every data collector can supply its own data communication software which can be added separately. Even the software itself can be extended and updated by e-mail.

Different data collectors can use the same installed Electronic Data Collector. A level called *survey* has been introduced to create the possibility of hierarchical questionnaires. Surveys are grouped by data collector. The questionnaires can use different code lists e.g. for the lookup of answer values. With the possibility of shared code lists voluminous code lists can be stored only once while used in the surveys of many data collectors. Tests with code lists containing over 35.000 elements had almost no influence on the high performance.

Every questionnaire can contain its own hierarchy of keys ten levels deep. All kind of key types (predefined using a lookup or just open) are supported. Predefined values for keys can be maintained by the data collector when sending questionnaires. Routing and checking is available for open keys. The Electronic Data Reporter provides storage facilities for statements.

2. History

To lower the administrative burden of the respondents Statistics Netherlands has been active in the field of electronic data collection since the beginning of the nineties. For this purpose several tools have been developed, always applying Blaise as much as possible.

As the borders in the EU opened for trade in 1993 the reporting burden related to this trade shifted from customs to the enterprises. To limit this burden as much as

[8] EDI between Statistics and ENTerprises.

possible IRIS[9], a dedicated tool for the collection of data for the Foreign Trade survey, was developed. The first version was built in Blaise 2.5 and operational since January 1, 1993. It was quite successful from the start and therefore it was further developed during the years. The current version is build with Blaise 4.5. For special functions a Delphi DLL is added. Besides manual data entry a semi-fixed import facility is available. I refer to it as "semi" because although the fields are fixed their place in the record can be defined by the user.

In 1993 a pilot project called EDI Pilot1 was started to develop a more generic process of collecting data from enterprises for all kind of surveys. This pilot project resulted in 1994 in the proto type of the EDISENT tool. This tool was based on the concept of flexible, tuneable import functionality and it supported multiple questionnaires. The proto type was build using Turbo Vision 6 in combination with Blaise 2.5. With this module a small pilot was performed involving 10 enterprises. As the results were positive the management gave the assignment to start a second pilot project, called EDI Pilot 2. The goal of this project was to develop a more robust version of EDISENT and test it out with a larger number of respondents. The project started in 1995 and the development of the new version of the EDISENT module was placed under the same responsibility as Blaise development because of the available knowledge of such tools in this department. It was build completely in Turbo Vision using several parts of the Blaise sources. In 1996 the pilot phase was ended with some 150 enterprises using this tool for reporting statistical data to Statistics Netherlands. In 1996 Statistics Netherlands introduced the EDISENT concept in the European funded TELER[10] project. In this project a Windows 3.1 version of EDISENT with improved functionality was built in Delhi 1 still using parts of the Blaise sources. It was tested out in 8 countries (Sweden, Finland, Germany, Italy, Spain, Portugal, Slovenia and The Netherlands). Today it is still in use in Slovenia and The Netherlands.

Given the limitations within the project there were still a number of restrictions left in the functionality of EDISENT. One of them was the limited flexibility concerning the design of the questionnaires. The lay-out was rather fixed and only a basic set of elements could be applied to construct questions. There were hardly any rules possible, only range checks on numeric fields were allowed. The identification of the questionnaires consisted of *unit* and *reporting period*. To create questionnaires a program was developed using an ASCII definition as input. It appeared to be difficult to integrate this program in a data collection system. After the project has ended the EDISENT module was improved several times but the limitations of the programming language (Delphi 1) soon were felt. As Statistics Netherlands was in the possession of Blaise it was considered not opportune to put more effort in further development of EDISENT.

Once installed and tuned at the respondent EDISENT fulfilled its promise to reduce substantially the administrative burden. However, as the initial costs of tuning were to high the management of Statistics Netherlands decided to abandon the use of EDISENT. In stead a new tool was needed to replace EDISENT but limited to support only the data entry functionality. For this purpose the Electronic Data Reporter was developed. The challenge was to apply as much as Blaise as possible. Not only the integration of underlying parts would be as optimal as possible, also the tool will then be very light to install. Thanks to additions in Blaise 4.6 this goal was achieved: with the exception of the launcher the Electronic Data Reporter is completely built in Blaise. Besides the Electronic Data Reporter (and still IRIS and EDISENT) Statistics Netherlands is also using Blaise IS for its electronic data collection.

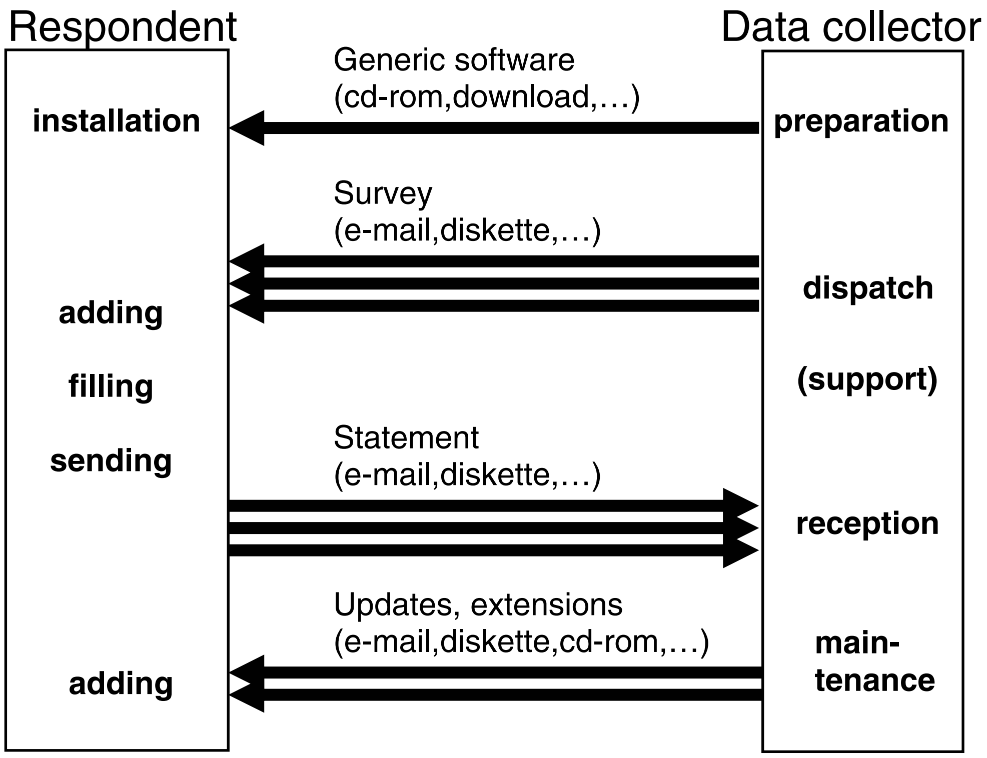
[9] Interactive Registration of International trade Statistics.
[10] TELematics for Enterprise Reporting

3. General process

Just like EDISENT the Electronic Data Reporter must be installed at the respondent. The installation is only necessary for a generic part of the tool. Although this is done by sending a installable version on a CD-rom the whole tool is light enough to be send by e-mail. Given the right compression it can be reduced to a little bit over 2 MB. As it does not contain any components, registration is not necessary. As a matter of fact it can even be distributed as a self-extracting executable or zip-file.

The Electronic Data Reporter has been developed to support the electronic data collection by a data collector (the organisation that is requesting the data like a NSI[11] or the Tax authority) at respondents (the organisations that are providing the data like enterprises and institutes). It has not been developed for typical statistical data collection, it can be used for the collection of all kind of data. An installed version of the Electronic Data Reporter can be used by different data collectors simultaneously.

Figure 1. The general process



After installation surveys can be send by the data collector to the respondent e.g. by e-mail. These surveys are in reality Blaise files (bmi, bdm, bxi, bdv) and Maniplus set-up’s (msu) packed in Microsoft cabinet files (cab). If predefined data is included too (e.g. lookup files) it can be in the shape of Blaise databases (bdb+) or ASCII files. Finally a small XML-file is added containing information about the survey and the data collector such as identifiers and names. These so called supplement files are given the extension *esf* (**E**lectronic Data Reporter **S**upplement **F**iles). Using a registered file association the Electronic Data Reporter can be invoked by merely clicking on the supplement file. As a result the survey is automatically imported in the Electronic Data Reporter environment and, depending on a tag in the XML-file, the included questionnaire is opened and the

[11] National Statistical Institute

user is invited to fill it in and send the data (statement) directly to Statistics Netherlands using the included data communication software.

Every data collector can supply its own data communication software which can be added separately packing it in the a supplement file just like the surveys. Again the XML-file will inform the Electronic Data Reporter about the content. Even the software itself (including the Blaise run-time engine Manipula.exe) can be extended and updated using the same procedure. The concept of sending the surveys separately by mail was already implemented before in one of the last versions of EDISENT. As a data collector can control the whole functioning of the data collecting tool by just changing the meta data on a distant this concept is called Remote Meta Data Control or RMDC. This concept was once proposed as part of a EU-funded project (METER) but the project was eventually not accepted for organisational reasons.

4. Logical structure

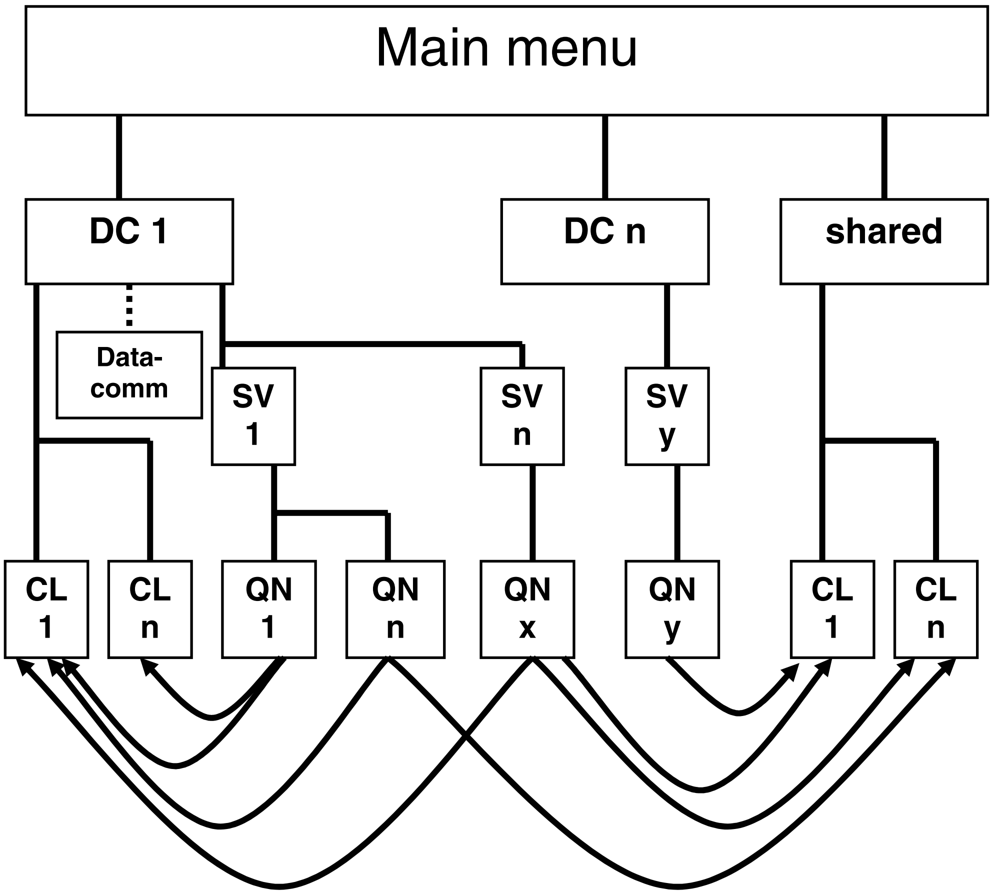
The general (and generic) access to the Electronic Data Collector is through the main menu. Below this main level the logical structure is divided per *data collector* as the tool has been developed for the simultaneous use by various data collectors. Each data collector should therefore have a unique identifier consisting of maximal 8 characters. The identifiers are also used as file names. To support still older environments as Novell 3.11 these names should be limited to 8 characters. The data collector level is parent for the next two levels: *surveys* and *codelists*. The survey level has been introduced to make it possible to apply a (hierarchical) set of questionnaires. For the survey on Road Transport a questionnaire set consisting of 4 linked questionnaires was created: *enterprise*, *vehicle*, *journey* and *load*. The survey is uniquely identified only within the set of the data collector. In the same line the identifiers of the questionnaires only have to be unique within the boundaries of a survey. Placing the code lists next to the surveys allows you to use the same code list in different surveys. Not only will it save disk space but it also guarantees more consistency. The code lists have there own set of unique identifiers within the domain of the data collector, separately from the surveys. Parallel to the data collector a special entry *SHARED* was introduced for the storage of code lists that can be used by all the data collectors.

The user is accessing the different functionalities through the main menu. The two most important entries are *Statement* and *Survey*. Starting up the Electronic Data Reporter the user is more or less forced to select one of the surveys of one of the data collectors as the active one. The name and code of this survey is clearly visible in the main window so the user is constantly aware which survey he is working on. Every option within the *Statement* sub-menu is now related to actions on statements belonging to this survey, with the exception of *send* (all statements for the current active *data collector* are sent in one message if the option is activated).

Within the *Survey* sub-menu options to manipulate surveys are available like selecting another survey. The level of questionnaires is not available to the user. A user is expected to create statements for a survey. Questionnaires are only considered as the means to do so. As it can differ from survey to survey the routing between different questionnaires within a survey must be controlled by the survey itself. Therefore this functionality is included in the processes that are typical to a survey and not in the generic menu. In most of the cases a survey will consist of a single questionnaire so activating a survey is almost the same as activating a questionnaire.

Normally the code lists are fixed lists and accessible through a questionnaire, e.g. as a lookup. A data collector can also include code lists that can (or must) be edited by the user before using the resulting data in a questionnaire. This can be useful if local user codes have to be translated to standard data collector codes. In the case a data collector has defined a code list as editable it is accessible through the option *Code list* in the *Survey* sub-menu.

Figure 2: The logical structure (DC = data collector, SV = survey, QN = questionnaire, CL = code list)



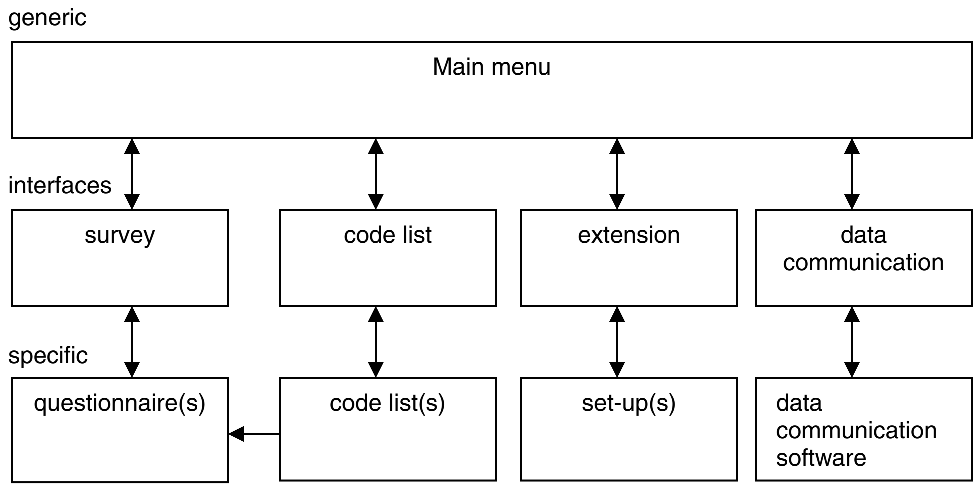
5. Technical structure

From a more technical point of view the architecture of he Electronic Data Reporter consists of three layers. The first layer is the generic main menu. The main menu is controlling the next layer which contains the interfaces with the specific data models of the questionnaires and the code lists. The latter one is only present if the code list is editable. Furthermore this layer contains the interfaces with the data collector dependant data communication software. Consequently the third layer contains the questionnaires, the code lists and the data communication software. The Electronic Data Reporter tool can be extended with extra functionality (like imports) using add-ons. They are also using an interface being part of the second layer.

An alternative way of looking at the structure is to see the combinations of an interface with its specific parts (questionnaires, code lists, data communication software and extensions) as objects added to the generic main menu.

In the next paragraphs a small overview is given of the way the different parts of the Electronic Data Reporter are working and communicating with their environment.

Figure 3 – The 3 layers



5.1 The main menu

The main menu itself is a Maniplus set-up with a fixed set of Blaise files for administrative purposes. In these Blaise files the codes and names of data collectors, surveys and code lists are registered including some attributes. The menu itself and all display texts (including button texts) are read from an external ASCII file. For different languages different ASCII files are used, the language code is part of the file name. This language code is stored in the registry. Under Win2000 and XP the language setting (and other settings) is even user dependant.

To communicate with the registry the main menu simply ‘runs’ the standard Windows program *Regedit.exe* with command line options. Depending on the command line options it can be used to write to or to export from the registry. The data that is interchanged between the *Regedit.exe* program and the main menu is written to and read from a so-called *.reg* file.

In many other cases standard Windows or ‘user’ programs are used to perform special functions. In stead of supporting all kind of printers the Electronic Data Reporting produces reports as RTF or ASCII files and invokes then the associated text editor by calling the standard Windows program *RunDLL.exe* with the right set of parameters. It is left to the text editor to provide the printing facilities. Another standard Windows program called *Extrac32.exe* is used to extract files from Microsoft cabinet files. These cabinet files are used as supplement files mentioned in chapter 3. To create these cabinet files a free Microsoft program called *CabArc.exe* is available on MSDN[12]. This program is used at the data collector’ to create the supplement files.

The main menu is using the standard defined temporary map to store all kind of working files. The path of the standard temporary map is obtained from the environment variables TEMP or TMP or the equivalent registry entries. A user setting is available to select another temporary map.

The Electronic Data Reporter is working with a fixed folder structure relative to the main installation folder. The unique codes identifying the data collector, the survey and the code list are used as folder names. The launcher, the only C++ program within the Electronic Data Reporter called *EDataRep.exe*, is making the main installation folder the current folder. By so doing the set-ups and data models can use relative file paths.

[12] Microsoft Developer Network

Figure 4 – The directory structure (dc-code = data collector code, sv-code = survey code, cl-code = code list code)

```
<main installation folder>
          |-Data
          |          |-$Shared
          |          |          |-CL
          |          |
          |          |-<dc-code>
          |                     |-CL
          |                     |          |-<cl-code>
          |                     |
          |                     |-QN
          |                                |-<sv-code>
          |
          |-Datacomm
          |          |-<dc-code>
          |
          |-Log
          |          |-<dc-code>
          |
          |-Util
```

5.2 The interfaces

The interfaces are Maniplus set-ups just like the main menu. Through a standard set of commands using the CALL function the interface set-up is invoked to perform one of the available actions. For each command a set of additional parameters is defined. To report back to the main menu a message file in the temporary map is used. Currently a new feature in Blaise 4.6 is introduced using the HALT command with a user defined code. This code (an integer) is stored in the resulting field of the CALL command.

One of the temporary files created by the main menu is a Blaise file containing all kind of text strings to be used by the interfaces. The main menu is selecting the text strings from an ASCII file depending on the language setting. By so doing the interface set-up does not have to support functionality to select the right text file. The interface set-up simply selects the necessary text strings from this file using the GET command with the related key.

An interface can be created simply by applying a template available for the interface set-up of a survey with one questionnaire. For different actions or more complex multi questionnaire surveys one should create its own set-up.

5.3 The questionnaires and code lists

The questionnaires are just Blaise data models with a small administrative block and a field to store a version number added. Every type of Blaise data model can be applied with every type of lay-out. (Of course it is recommended to harmonise the look of questionnaires.) Menu configuration files (bwm) can be added as well. Normally two of these files are present: one with options used when editing forms and one used when viewing forms. With a language code included in the file names of the configuration files the interface can select the right one depending on the current language setting.

The forms within a questionnaire file are the statements to send back to the data collector. The primary keys of these forms are not filled in through the form itself.

In the interface set-up the key values are asked using Maniplus dialogs. In case of creating a new form the interface set-up determines if the key already exists and if it does it offers the user the possibility to create another version of the form. The right version number is automatically created. Using Maniplus dialogs makes it possible to create a very neat user interface and even a very sophisticated (hierarchical) selection of predefined key values.

5.4 The data communication software

The data communication software is data collector dependant and normally written in languages like Delphi or C++. It is invoked by a special interface with the fixed name *EDRsend.msu*, which deals with the special commands to invoke the different functions of this program and to interpret the feedback. Statistics Netherlands uses its own program (*CBScrypt.exe*) that applies Triple-DES and RSA for encryption (according to the PKI[13] concept). It uses the local mail client for sending the statements over the Internet by SMTP[14].

6. Blaise 4.6. features

The Electronic Data Reporter was more or less developed parallel with Blaise 4.6. Many of the new features in Blaise 4.6 (at least the ones in Maniplus) were added based on needs emerging from the development of the Electronic Data Reporter. Consequently Blaise 4.6 was tested automatically while testing the Electronic Data Reporter. This implicit co-operation appeared to be very fruitful for both developments.

The new features of Blaise 4.6 listed below are used in the Electronic Data Reporter:

- Main window display (text and logo). The name and the code of the current active survey are displayed in the main window and the logo of the related data collector is shown at the right bottom corner.
- Hourglass display in message boxes while an action is performed.
- Fills in dialogs (including the buttons) to make them multi-lingual applying text read from external files.
- Improved shared functions for Blaise files creating a stable multi-user environment.
- Integration of help files so the same help file is used in Maniplus set-ups (interfaces) as well as data models (questionnaires).
- User-defined name for help files to make them multi-lingual.
- System-text-file parameter (/#) so even the system messages are multi-lingual.
- Version information string used in the ‘about’ box and status report.
- Database views for lookups in data models creating customised and multi-lingual lookups.
- Common stop option (Alt-F4) providing a more consistent user interface.
- Extended RUN-commands avoiding the use of non-Blaise programs for file handling.
- User-defined HALT values for better and easier communication between set-ups.
- Select folder dialog for user-defined work folder setting.
- Save file dialog for interactive user controlled storage of output files.
- Default NO in Confirm dialog avoiding user errors.
- IN- and OUTPUTPATH function for better sharing of data between set-ups.

[13] Public Key Infrastructure
[14] Simple Mail Transport Protocol

Other new features of Blaise 4.6 like the dialog boxes will be included in future releases of the Electronic Data Reporter as well as a better use of the other features.

7. Current use

The Electronic Data Reporter was first put in production for the Traffic and Transport survey at January 2003. The tool was not disseminated by Statistics Netherlands itself but by SIEV, an organisation for the transport branch. They co-operate with Statistics Netherlands collecting the data for both data collectors at the same time. NIWO, another branch organisation will join soon. The independency of the Electronic Data Reporter from the data collector appeared to be useful from the beginning. As mentioned before, the survey consisted of 4 hierarchical questionnaires with large code lists for lookups. Some of these code lists contain about 40.000 records and even a trigram search gave no performance problems.

In March the survey of International Trade in Services started with the use of the Electronic Data Reporter. For this survey the import extension was added. It is now used for a dedicated ASCII record set but an XBRL alternative already has been developed.

Soon the Electronic Data Reporter will be put in action for the Short Term Production survey. Unlike the other surveys the Blaise questionnaires for this survey will be generated from a questionnaire server (part of the LogiQuest data collection system) and send automatically by e-mail to the respondents.

More surveys are testing the tool at this moment.

8. Future developments

The basic architecture of the Electronic Data Reporter in combination with the Blaise 4.6 language makes it very easy to extend its functionality. Some features that are on the list to be added:

- Extended and flexible import facilities
- Complete multi-lingual environment
- Archive facilities
- Export facilities
- XML messaging (ebXML[15], XML4DR[16]). This feature is developed in co-operation with members of the SOS[17]-group.
- Certified distribution packages

With the growing use of the Electronic Data Reporter more useful features are expected to be identified and thereupon added to the list.

[15] Electronic Business XML
[16] XML for Data Reporting
[17] Statistical Open Standards