Distr.
GENERAL

WP.2
29 April 2011

ENGLISH ONLY

UNITED NATIONS ECONOMIC COMMISSION FOR EUROPE (UNECE)
CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE EUROPEAN UNION (EUROSTAT)

ORGANISATION FOR ECONOMIC COOPERATION AND DEVELOPMENT (OECD)
STATISTICS DIRECTORATE

Meeting on the Management of Statistical Information Systems (MSIS 2011)
(Luxembourg, 23-25 May 2011)

Topic (i): Architectures, models and standards

Standard Process Steps in Statistics[1]

Invited Paper

Prepared by Robbert Renssen and Astrea Camstra, Statistics Netherlands, the Netherlands

I. Introduction

Statistics Netherlands (SN) is facing a number of major challenges. On the one hand, budget cuts and the IT maintenance burden call for increasing efficiency. On the other hand, the statistical landscape is changing rapidly. This applies both to the input side, where attention shifts from primary data collection to the use of administrative registers, and to the output side, where users ask for increasing flexibility while maintaining high quality standards. In order to manage these competing challenges an ambitious redesign program, the Masterplan, was started in 2005.

The general ideas of the Masterplan are represented in a comprehensive enterprise architecture. Some key elements are

- the identification of steady states in statistical processes, consisting of data sets with guaranteed quality to promote re-use of data and supported by a Data Service Centre
- the introduction of a series of standard statistical methods and a box of standard tools, both supporting the core production processes.

More detailed background information about the first key element can be found in Renssen and Goossens (2010). They describe a number of underlying ideas and concepts of the Data Service Centre (DSC) at SN, including the notion of a steady state.

This paper touches upon the second key element. It deals with so-called standard process steps, which can be considered as a linking pin between a general statistical method and its specific use in a process, taking into account all kinds of complications one may encounter in practice.

[1] The views expressed in this paper are those of the author and do not necessarily reflect the policies of Statistics Netherlands. The research project is still in progress. This paper is in concept and shows some intermediate results.

Briefly, a standard process step is an application of a statistical function (automated as a component or building block) that can be (re)used in a statistical process. The working of the function is usually based on a statistical method. By identifying these statistical functions and providing guidelines for their use, this paper aims to close the gap between the high-level view taken in the business architecture and the usual way to design statistical processes in practice.

There are several reasons why the gap between the high level view of the business architecture and the usual way to design processes in practice is huge. We mention a few:

- Applications of statistical methods are not recognized, especially when these applications are trivial. For example, when estimating population totals based on register data, these data are implicitly weighted using weights that are equal to 1.
- Applications of statistical methods may be very complex and need preparatory activities. For example, when using *t*-1 data in an editing process, these *t*-1 data should be matched beforehand. In addition, there are procedures in case the *t*-1 data is not complete.
- Applications of specific tools may need preparatory (non-statistical) activities, like a transformation of format, reshuffling data columns or renaming variables.
- Re-use of data and mixed mode strategies complicate the activities in a process, because several data sources should be combined and processes should be mutually linked.

The idea of standard process steps anticipates two future situations that are not mutually exclusive. In both situations a library of automated statistical functions (building blocks)[2] constitutes the starting point to design and build production processes.

- In the first future situation the library is used to design and build specific processes, by selecting, specifying and putting together the required components. As soon as a component is specified for its specific use in a statistical process, we call its application a standard process step. The resulting processes are slightly flexible by adjusting the specification of the components through their fixed interfaces. It is also possible to replace one component by another component provided that the functionality and interface of the component is similar. However, the flexibility is limited as no more components are implemented than are needed for the specific statistical process.
- The second future situation is more ambitious. The same library is used to design and build generic processes that can be used for several (similar) statistical processes. For a particular statistical process, the generic process can be configured by disconnecting one or more components.

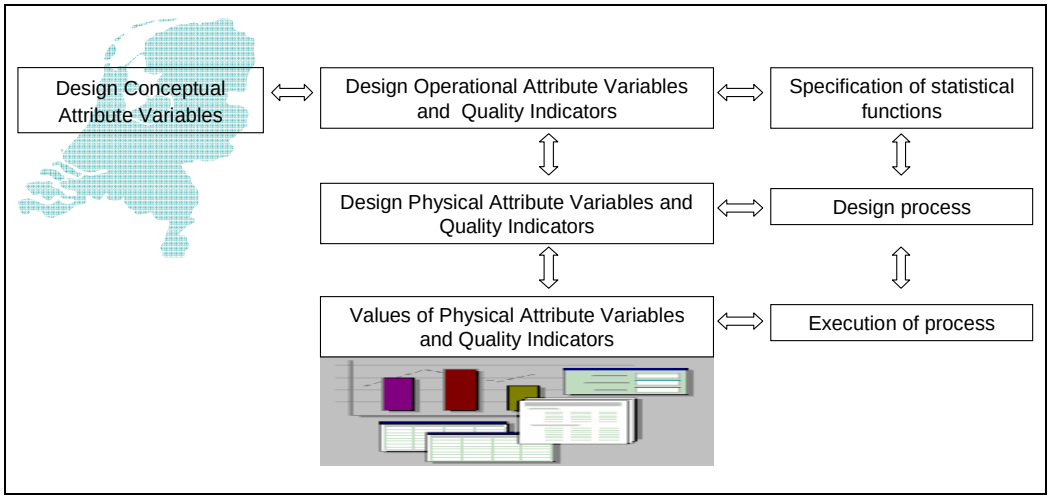
In this paper we will briefly discuss an integrated framework that underlies the concept of automated components and their use as standard process steps. This integrated framework relates data modelling and process modelling. Furthermore, we will give an initial outline of the content of the library of statistical functions and we will relate this library to the Generic Statistical Business Process Model, see Vale (2009).

II. The underlying integrated framework in a nutshell

The concept of a standard process step is not an isolated concept, but constitutes part of a framework. This framework is summarized in figure 1, which pictures the mapping of real-world properties onto a statistical dataset. The real-world properties and objects are placed on the upper left side (shape of the Netherlands) whereas the statistical data are depicted at bottom. Between the real-world objects and properties and the statistical data is a design process and an execution process.

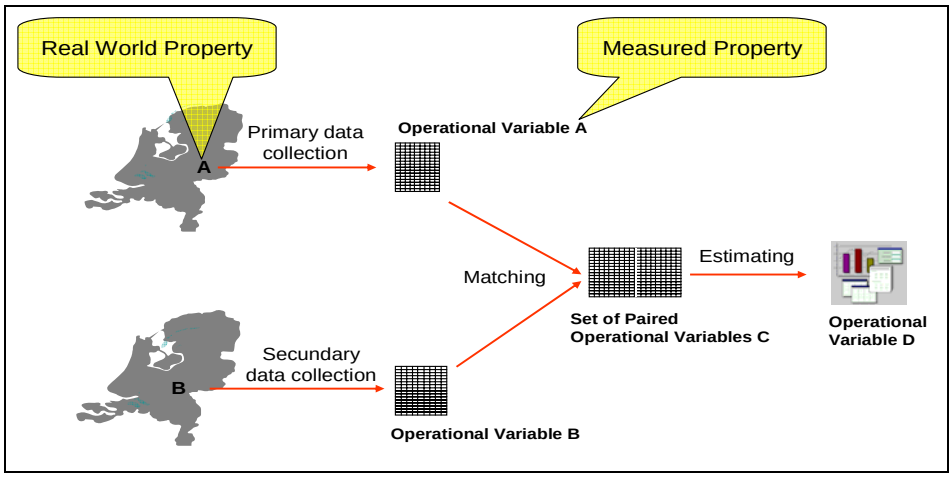
Figure 1: design and execution of a statistical process

[2] These components are automatized statistical functions and are made available as business services. These services have a standard in- and output and can only be adjusted by a well-defined interface.



1. The design process ideally starts with the delineation of the (relevant) real-world population and the properties to be considered (both with respect to a defined time period). This results into a set of so-called conceptual attribute variables. Gender with possible outcomes (male, female) is an example of such a variable. These conceptual attribute variables may also represent theoretical constructs.
2. The next step in the design process is to design the operational counterparts of these conceptual attribute variables. This results into a set of so-called operational attribute variables. Gender with possible outcomes (male, female, not observed) is an example of an operational variable. A conceptual variable, like unemployment, is usually operationalized in several stages. As an example, consider figure 2 in which a conceptual variable D is operationalized in three stages
   a. First, two conceptual variables (A and B) are observed during the collection stage, resulting in two operational variables (A and B). In contrary to their conceptual counterparts, both operational variables may have missing or erroneous outcomes.
   b. Second, these operational variables are matched. This results in a set of paired operational variables (C). Due to matching errors this paired set may suffer even more from missing or erroneous outcomes.
   c. Third, this set of paired variables is used to estimate the population total (D). Due to sampling errors, matching errors and/or data collection errors the resulting estimate may differ from the conceptual total that had to be estimated.

Figure 2: Operationalization of a variable in several stages



3. Each of the stages involves statistical activities, which are modelled as applications of statistical functions. In our framework each application of a statistical function is managed through an interface by a product rule with respect to the input (according to some data model), a product rule with respect to the output (according to some data model) and

a method rule with respect to the (internal) process transforming the input into the output. This method rule is based on a statistical method and/or subject matter knowledge. The three stages mentioned above can be elaborated as follows.

a. In case of primary data collection (we will not elaborate on the secondary data collection), the first stage could be modelled by (a series of) successive applications of a matching function, a sampling function and an interviewing function, the latter based on some (mixed mode) collection strategy.

   i. The inputs of the matching function are two datasets (e.g. a population frame as a recipient dataset and an external register as a donor dataset). The output is an enriched population frame which will serve as the sampling frame. The choice of the matching variables and a matching criterion (often based on a distance measure) performs the role of the method rule.

   ii. The input of the sampling function is a sampling frame and the output is a sample. The sampling design constitutes the method rule.

   iii. The input of the interviewing function is a sampled unit and the output is the response (and non-response). The questionnaire design fulfils the role of both the method rule and the product rule with respect to the output.

b. The second stage could be modelled by one ore more applications of a matching function.

   i. The inputs of the matching function are two datasets (e.g. the dataset obtained through primary data collection serves as recipient and the dataset from secondary data collection as donor) and the output is a matched dataset (including the units that cannot be matched). The choice of the matching variables and the matching criterion (often based on a distance measure) fulfils the role of method rule.

c. The third stage could be modelled by an application of one or more estimation functions (often, but not necessarily, based on a regression model).

   i. The input of the estimation function is a micro data set (some variables in this data set may serve as classification variables to delineate the subpopulations, while other variables in this set serve as target variables). The output of the estimation function is a set of estimated population totals with respect to the target variable. These estimated population totals are classified according to classification variables. The first order inclusion probabilities, the regression model (including the choice of the auxiliary variables) and the corresponding 'known' population totals are laid down in the method rule.

Now, the third step in the design process is to design the sequence of statistical functions that have to be applied, and to specify each function by specifying its product rules and method rule. We note that functions, such as the matching function, could be applied at different stages for different purposes. We note further that certain product rules correspond to the design of the desired (conceptual) output. The instances of these product rules can be released (or made accessible) as a statistical product (steady state), i.e. a well described statistical data set that satisfies certain quality standards.

4. Designing the set of operational attribute variables requires no knowledge about the technical formats in which the statistical data are stored. The fourth step in the design process is to design the physical (or technical) counterparts of the operational attribute variables. This results in a set of so-called physical attribute variables. Gender with possible outcomes (1, 2, 9999) is an example of a physical variable. The same variable can be alternatively expressed as a set of three (physical) dummy variables, each with possible outcomes (1, 0). The first dummy variable indicates whether the outcome of the operational variable is male or not, the second dummy variable indicates whether the outcome is female or not, and the third dummy variable whether it is missing or not.

5. The fifth step in the design process is to design the production process in which the statistical functions are applied. The design of the production process results in a set of process rules and decision rules.

   a. A process rule can be considered as an ordered set of statistical functions with (versions of) specifications of the product rules and method rules that have to be applied in the production process.

   b. A decision rule tells the production process when to start and stop the application of a function. A decision rule may be time driven, quality driven, cost driven or event driven.

6. The actual realisation of the production process finally results into a physical dataset, i.e. statistical data. These are the outcomes of the physical variables. These outcomes can be interpreted by relating them, through the operational variables, to the conceptual definitions of the conceptual variables.

The important difference between a conceptual attribute variable and an operational attribute variable is that the latter may suffer from time lags, missing data, measurement errors and so on.

In our framework we also consider quality data that describe the quality of the outcomes of a set of operational attribute variables (steady states). In common with statistical data, we distinguish between conceptual, operational and physical quality indicators. Furthermore, we also distinguish statistical functions that 'measure' quality according to some statistical method or subject matter knowledge.

III. An example of a statistical function

As an illustration we give a simplified example of a statistical function, namely the matching function. Consider two micro datasets with common population delineation and with a number of common variables, and assume that one dataset is the recipient dataset and the other dataset is the donor dataset. Based on the common variables the units in the donor dataset are compared with and matched to the units in the recipient dataset.

Ideally, the common variables have identical definitions, are operationalized in a similar way, and provide unique physical keys for identification. Unfortunately, the common variables often differ in practice, at both the operational and physical level. For example, there may be slight differences due to time lags or different ways of rounding numbers. In addition, the common variables may contain errors or missing data.

> Suppose that two micro datasets referring to a common population of persons are matched using five matching variables. The recipient data set contains the variables Social-Security-Number, Name, Place-of-residence, Gender and Age. The donor data set contains the variables Social-Security-Number, Name, Place-of-residence, Gender and Date-of-birth. Suppose further that the variables Social-Security-Number and gender are identically defined but may contain missing values and errors. Finally suppose the variables Name and Place-of-residence are open text variables (these variables may also contain missing values and errors).
> Provided that the variable Social-Security-Number contains no missing values and/or errors in both data sets, this variable is ideally used to match the recipient and donor units. Therefore, the validity of the social security numbers is usually checked first. For valid social security numbers the recipient and donor units are actually matched, using as a matching criterion that the numbers be identical.
> Suppose the number of positive matches is insufficient because of too many missing values in the variable Social-Security-Number. The unmatched recipient and donor units may be matched in a second iteration, using the matching variables Name, Place-of-Residence, Gender and Age/Date-of-Birth. In order to use the open text variables name and Place-of-Residence, these variables have to be coded first. That is, by means of a so-called knowledge table these open text variables have to be transformed into new variables having a finite number of categories. In order to use Age and Date-of-Birth as matching variables, the variable Age in the donor data set has to be derived from the variable Data-of-Birth. Finally, it could be necessary to recode the categories of the variable Gender, e.g. from {male, female} into {0,1}.
> Provided these derived and (re)coded variables have no missing values and/or errors – these variables can be used as matching variables for the second iteration. An example of a matching criterion would be that the values of the variables Name and Place-of-Residence should be identical in both datasets and also that there is a positive match on either Age or Gender.

When different statisticians are asked to give their understanding of *matching datasets* they would probably give different answers. Some answers would be close to the iterative procedure described above, including the preparatory steps. Other answers would more closely resemble an elementary matching step, e.g. the matching step in the first or second iteration.

In our framework, the notion of a matching function corresponds to the elementary matching step, or more precisely, the elementary matching step is an application of our notion of a matching function.

A. Product rule with respect to the input

Both the recipient dataset and the donor dataset may have complex data models, i.e. both dataset may refer to more than one object type, as long as they have

- a common population delineation (and hence a common object type) at the conceptual level, and

- a common set of variables with respect to this object type at the conceptual level.
-

Figure 3a and 3b illustrate a recipient and donor dataset that satisfy these conditions. The variable *v*3 serves as the common variable. At the conceptual level, this variable is defined identical for both the recipient and donor dataset. However, there may be operational differences at the operational level, resulting in differences in the instances of the operational variables. For this reason, the operationalizations of *v*3 are denoted by *v*3 (recipient) and *v*3 (donor).

We note that the specification of the logical data models, including the definitions of the variables and population delineations are laid down in the product rule with respect to the input.

Figure 3a: Example of a logical data model of the recipient dataset

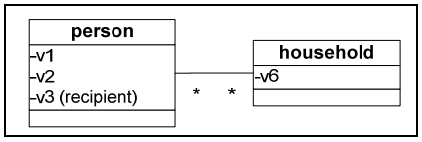
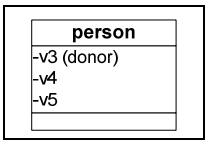


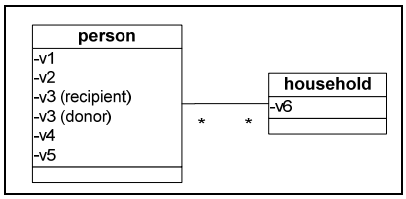
Figure 3b: Example of a logical data model of the donor dataset



B. Product rule with respect to the output

The specification of the logical data model of the matched dataset, as illustrated in figure 4, is laid down in the product rule with respect to the output. The population delineation of this dataset is determined by the common population delineation of the recipient and donor dataset. The set of variables with respect to this common population is a selection of variables from both the recipient and donor dataset.

Figure 4: Example of a logical data model of the output



C. Method rule

Note that the instance of the population delineation of the matched dataset can be taken as

- The recipient data set (left join); donor variables have missing values if there is no unique match.
- The union between the recipient and donor data set (outer join); donor variables or recipient variables have missing data if there is no unique match.

The choice for one of these population instances is a design decision depending on e.g. the (expected) under coverage of the recipient dataset and the desired coverage of the output. The choice of the common variables, the choice of the distance measure to decide whether a match is positive (or negative) and the choice of the population instances to be taken are specified in the method rule. This rule tells us *how* the output is obtained from the input.

D. Data quality

Naturally, the matching function performs better if the quality of both the recipient and donor dataset is higher. Ideally, the instances with respect to the common population delineation of both datasets have perfect coverage and these instances have unique identifiable keys. In practice, however, both instances may suffer from e.g. coverage errors and duplicates. In addition the keys may suffer from missing data and non-uniqueness.

When applying a matching function to an imperfect recipient and donor dataset there is no guarantee for unique (positive) matches. Depending on the choice of the population instance (e.g. left join or outer join) this will result in an imperfect population instance as well as missing data with respect to the donor and/or recipient variables.

D. Event driven application

There are several ways to apply a matching function as a standard process step. We briefly discuss an event driven application.

The event driven application starts with the recipient dataset as an initial instance of the output of the matching function. Each time a donor becomes available the matching function is triggered, which results into a new instance of the output. This new instance could be the values of the donor variables that are added to a recipient unit. In case of an outer join, this new instance could also be the donor unit itself. The application stops when the new instance reaches a quality standard.

IV. Repository of statistical functions

We have globally described a statistical function in order to illustrate its main features. Without striving for completeness, in this section we will list a number of potential (statistical) functions in order to give a picture of the repository of functions.

Before listing these functions, a number of comments are in order. First we will make a distinction between a statistical and a non-statistical function. A statistical function changes an operational variable, while a non-statistical function only changes a physical variable (leaving its operational counterpart unaltered). When designing a process, both types of functions are needed. However, only the statistical functions ‘improve’ the statistical data.

Second, the internal process of a statistical function to process its input into output is often based on a statistical method. Often, there is more than one statistical method for the same function. Then there may be several methodological implementations with respect to the same function. For example, the estimation function could be implemented according to a calibration estimator, a general regression estimator, or a small domain estimator. The imputation function could be implemented according to hot deck imputation, regression imputation, deductive imputation, nearest neighbourhood imputation, and so on. Depending on the situation, one implementation could be more suitable to do the job than another implementation.

Third, like pieces of equipment, statistical functions in the library can be used for specific activities. The functions, however, do not have ‘knowledge’ about a specific use. If desired, they can be used throughout the statistical process and achieve different statistical goals. For example, the *matching function* can be used in an editing process to compare actual data to *t*-1 data. Alternatively, it can be used in the preparation stage of the data collection process to enrich a sampling design. Another example is given by the *estimation function,* which may be used to efficiently validate the micro-data in a macro-editing process or it could be used to estimate population totals for publication purposes. For both purposes the estimation function could have the same specification, e.g. the same specification of product rules and method rule, but this is not necessary.

Now, the following statistical functions are frequently used, either explicit or implicit, in many statistical production processes at Statistics Netherlands.

- Sampling function: draws samples from a sampling frame
- Interviewing function: obtains statistical data from respondents
- Data validation function: checks (combinations of) variables on missings, errors, implausible values, and so on
- Outlier detection function (special case of a data validation function): checks (combinations) of variables on outliers
- Error localisation function: determines the 'guilty' variables from detected combinations with errors
- Score function: assigns weights (scores) to units to indicate their (individual) influence on a population total
- Error correction function: corrects errors using the original erroneous values
- Imputation function: imputes missing or detected erroneous data without using the original erroneous values
- Variable and unit derivation function: derives new (operational) variables or units from existing (operational) variables.
- Coding function (special case of a variable derivation function): derives classification variables from open text variables.
- Matching function: compares and matches units from different data sources
- Estimation function (population totals): estimates population totals from micro-data
- Estimation function (variances): estimates sampling variances from micro-data
- Reconciliation function: adjusts estimated population totals to prior knowledge
- Disclosure control function: localises and handles (statistically) unsafe data

V. Relation to GSBPM

We have limited the list in IV to functions that are used in the production of statistical data. In a high level architecture, like GSBPM, a statistical production process is divided into a number of recognizable production activities. Again, without striving for completeness, we list the following typical (statistical) activities at the production stage of GSBPM:

- Collection stage:
  - select sample,
  - setup collection,
  - run collection
- Processing stage:
  - integrate data,
  - classify and code,
  - review, validate and edit,
  - impute,
  - derive variables and units,
  - calculate weights and aggregate
- Analysing stage:
  - validate, scrutinize and/or explain outputs,
  - apply disclosure control.

Behind these typical activities are applications of one or more of (methodological) implementations of statistical functions. For example, the review, validate and editing activity may exists of successive applications of the data validation function, the score function, the error localisation function and the error correction function, see Camstra and Renssen (in preparation). In addition, as preparatory activities one may also need a matching function, a variable derivation function, an outlier detection function and even an estimation function.

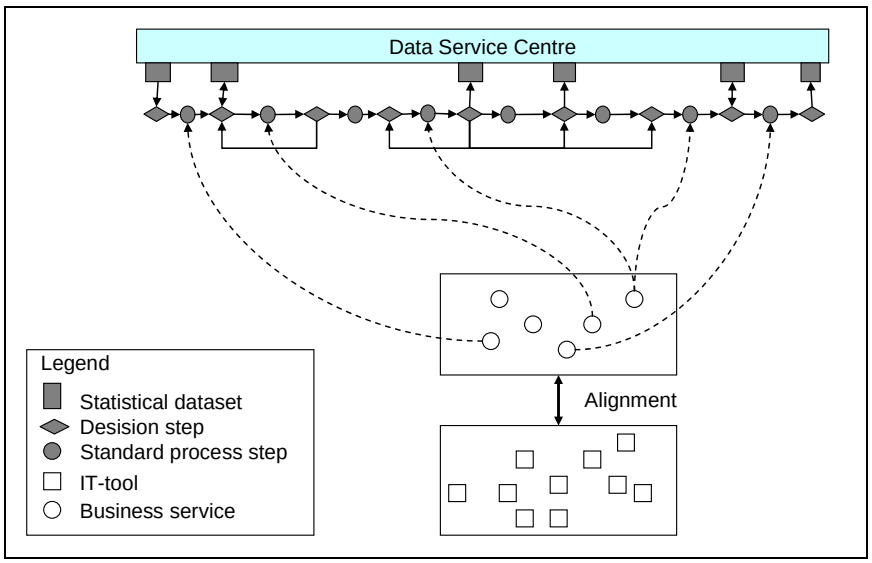
The applications of statistical functions to carry out typical statistical activities are called standard process steps. We consider them to be an important link between high level (business) architecture, like the GSBPM-model, and the usual way to design statistical processes in practice at Statistics Netherlands.

V. Processes, methods and tools

Statistical tools play a vital role in the production of statistics. About 3 years ago, the board of directors at SN has considered the diversity of tools required for our statistical production and has expressed the desire to limit this diversity as much as possible. However, as already stated in Renssen, Wings and Paulussen (2008), the discussion about tools should not begin with the diversity of tools but rather with the diversity of statistical functions. Once an agreement on the statistical functions has been reached, it is possible to take the next step and make a decision on the set of tools.

Although this paper is not about IT-tools, it may be instructive to give a glimpse of the relationship between a statistical function and its technical implementation. This relationship is reflected in figure 5 and could portray a future statistical production process at SN.

Figure 5: Future production process



This production process exchanges statistical data (shown as shaded rectangles) with other production processes through a Data Service Centre. The process consists of a series of applications of statistical functions and decision functions. These applications are the standard process steps and are shown as shaded circles. The methodology of each function is automated by means of IT-tools. These automated functions are available as business services. To that purpose the IT-tools and business services are aligned. The business services are shown as open circles and the IT-tools as open squares.

References

Camstra, A. and Renssen, R. (in preparation), *Standard process steps based on standard methods as part of the business architecture*. Paper to be presented at the 2011 ISI conference.

Renssen, R., Wings, H. en Paulussen, R. (2008), *Processes, methods and tools*, Statistics Netherlands, Heerlen.

Renssen, R. and Goossens, H. (2010), Underlying *Concepts of the Data Service Centre at Statistics Netherlands*, Statistics Netherlands, Heerlen.

Vale, S. (2009), *Generic Statistical Business Process Model*, Version 4.0, Joint UNECE/Eurostat/OECD Work Session on Statistical Metadata (METIS).