XBRL 2005: Taxonomy and instance creation and control

Report 06001

*Piet Daas and Alex Stroom**

Statistics Netherlands

Voorburg/Heerlen, May 2006

Explanation of symbols

.	= data not available
*	= provisional figure
x	= publication prohibited (confidential figure)
–	= nil or less than half of unit concerned
–	= (between two figures) inclusive
0 (0,0)	= less than half of unit concerned
blank	= not applicable
2003–2004	= 2003 to 2004 inclusive
2003/2004	= average of 2003 up to and including 2004
2003/’04	= crop year, financial year, school year etc. beginning in 2003 and ending in 2004

Due to rounding, some totals may not correspond with the sum of the separate figures.

Publisher
Statistics Netherlands
Prinses Beatrixlaan 428
2273 XZ Voorburg
The Netherlands

Printed by
Statistics Netherlands - Facility Services

Cover design
WAT ontwerpers, Utrecht

Information
E-mail: infoservice@cbs.nl

Where to order
E-mail: verkoop@cbs.nl

Internet
http://www.cbs.nl

ISSN: 1574-4698
Key figure: X-14
Production code: 6009504001

Statistics Netherlands

Contents

XBRL 2005: TAXONOMY AND INSTANCE CREATION AND CONTROL

This report provides an overview of the results of the XBRL 2005 project that was conducted at Statistics Netherlands in cooperation with Atos Origin. The XBRL 2005 project studied the various ways that data checks of Statistics Netherlands could be applied to instances in the XBRL format. The Structural Business Statistics questionnaire for the wholesale was used as an example.

Keywords: XBRL, validation, instance checks, XSLT, structural business statistics.

1. Introduction

XBRL (eXtensible Business Reporting Language) is an XML-based (eXtensible Markup Language) open standard for the compilation and electronic exchange of business reports and data on the basis of a common dictionary (taxonomy) of terms and their interrelationships (1). Examples are the periodic compilation of reports from various sources and/or systems and the exchange of data between systems. XBRL is very suitable for data exchange via the Internet, but is also useful for applications within organisations. This makes XBRL very interesting for Statistics Netherlands as an exchange standard for the receipt and processing of business data (2).

The initiative for the development of XBRL was taken in 1998 by the American Institute of Certified Public Accountants (AICPA) (3). According to AICPA, using an XML-based standard for financial reporting has the following advantages:

- It provides for a global standard, independent of software or platform
- It improves data quality, as extra operations, such as data re-input, are no longer required.
- It makes it possible to electronically exchange, analyse and where necessary process data selected and reported on the basis of the same taxonomy (definition of terms, structure, interrelationships and content of the data elements to be reported).
- It makes it possible to realise considerable cost savings, as data in XBRL format is based on taxonomies from which it can be selected depending on necessity or demand.

The basis of XBRL is the taxonomy (definitions of the terms to be reported) and the instance (the data to be reported). One XBRL instance consists of a collection of elements (called ‘concepts’ in XBRL) and attributes which are related to a certain area of reporting, e.g. the area of accounting, finance or other business-related information. A typical XBRL taxonomy contains a very extensive set of attributes. In addition, an XBRL instance also contains so-called context information, e.g. the reporting unit and the period to which the instance data refer. The data in the instance cannot be interpreted correctly without context.

An instance document is based on -and useless without- a taxonomy, as it is the taxonomy that indicates which concepts can be reported and what their interrelationships and structure are; it is the taxonomy that contains the metadata. As XBRL is not a reporting standard in a traditional sense, (such as the Generally Accepted Accounting Principles of the US-GAAP or the International Accounting Standards) but a technology, various taxonomies can be created for various regions and different types of reports. These could be based on GAAP or IAS definitions. The Dutch taxonomy is currently being developed (4). With the aid of XML-functionalities is possible to use concepts from different taxonomies in a single instance.

A taxonomy consists of an XBRL schemafile and files describing the relationships of concepts in the schemafile. Such relationships are called hierarchies and are recorded in so-called linkbase files (5). Linkbase files contain information on the relations between concepts and the relationships between concepts and documentation. Examples of the former are the definition, calculation and presentation linkbase. The definition linkbase describes the logical relationships between concepts, such as their mutual dependency. A calculation linkbase describes the arithmetic relationship between concepts; e.g. from which concepts the value of the total concept is composed. The presentation linkbase records the presentation structure between the concepts. The relationship between concepts and the documentation is described in the label and reference linkbases. The label linkbase contains the labels to be used for the various concepts. Each concept may be associated with several labels. These will often be labels in different languages. Lastly, the reference linkbase defines references to external information sources, such as documents, publications or statute books. A future addition to the linkbase family will be the formula linkbase (6). This file will allow the recording of complex arithmetic relationships between concepts in the taxonomy, and between concepts and constants. Such relationships are much more extensive and complex than the simple calculation relations recorded in the calculation linkbase. At the time of writing, the formula linkbase has not yet been included definitely in the XBRL specifications.

Because of mutual dependencies, a taxonomy may take on a very complex structure. Fortunately, this can be checked for errors with the aid of software. This is also possible for an instance. Such a check is called a validation. Special XBRL software is available for the creation and validation of taxonomies and instances (section 2.2.2 and ref. 7).

1.1 Purpose of the XBRL 2005 project

Statistics Netherlands' XBRL 2005 project is intended to contribute to knowledge and experience of XBRL, so that it can be applied in development projects. The project examined the extent to which XBRL is applicable at Statistics Netherlands, and which requirements had to be met to realize this. The underlying principle is that application of XBRL must improve the efficiency and effectiveness of the complete information supply chain; i.e. reduction of the administrative burden for companies as well as improvement of data processing at Statistics Netherlands. The XBRL 2005 project aims at supplying the knowledge to do this, and at establishing whether - and if so how - this can be done. The study focused on the following aspects:

- To what extent can concrete survey questions be conveyed in XBRL.

  To find this out, an annual business inquiry, i.e. the Structural Business Statistics questionnaire for the wholesale, was converted to XRBL, including process and validation rules.

- Validation with XBRL.

  Which data checks are possible in XBRL and, equally important, which checks are not possible in XBRL? For the latter category of checks it is important to know whether there is an alternative way to carry out such checks in an XBRL/XML based environment.

- The construction of taxonomies in accordance with current XBRL (5) and Financial Reporting Taxonomy Architecture (FRTA) specifications (8).

  Is it possible to make the taxonomy of the Structural Business Statistics questionnaire satisfy these specifications? How much work is involved in adapting an XBRL valid taxonomy to the FRTA specifications?

- The use of tools that Statistics Netherlands chooses to process XBRL.

  Which software tools are required to create, validate and/or process taxonomies and instances? Are these tools available at Statistics Netherlands?

- Preparation for (pilot) chain tests for the Dutch Taxonomy Project (NTP).

  For this purpose the taxonomy of the Structural Business Statistics questionnaire must be based on (the Statistics Netherlands' part of) the Dutch taxonomy. It is important to know which consequences this has for the construction of the questionnaire's taxonomy and the creation of instances.

The XBRL 2005 project does not focus on the development of an architecture for the processing of XBRL instances, nor does it focus on the design aspects of the use of XBRL in electronic questionnaires. The project was carried out at Statistics Netherlands in collaboration with Alex Stroom of Atos Origin.

1.2 Outline of this report

This report gives an overview of the results of the various stages of the XBRL 2005 project (9). More detailed descriptions of the results are available (10-12). Chapter 2 describes the steps leading to the construction of the XBRL taxonomy for the Structural Business Statistics questionnaire for the wholesale. This involves the inclusion of the Statistics Netherlands part of the Dutch taxonomy, the construction of the questionnaire's taxonomy and the validations steps that where subsequently carried out. Chapter 3 describes the creation of instances, the result of the collection of validation and processing rules and the various control methods to could be used. This chapter also clarifies the organisation of the classification and validation checks and the methods used to study these checks. Chapter 4 elaborates on the classification and discusses extensively the results of the tests. The separate test reports can be found under reference 12. In chapter 5 the results of the classification are linked to the validation rules. Lastly, chapter 6 reports on the findings and recommendations following the work described in the previous chapters. The annex to this document includes a list with abbreviations and a list of references.

2. Taxonomy

2.1 The Dutch taxonomy

The taxonomy constructed in this project is based on the Structural Business Statistics questionnaire for the wholesale (9). As the Dutch taxonomy is the basis for the architecture of the questionnaire's taxonomy, we took the structure of the Dutch taxonomy into account (4). When the project started, version 0.1 of the Dutch taxonomy was available. The structure of this version is given in figure 1.

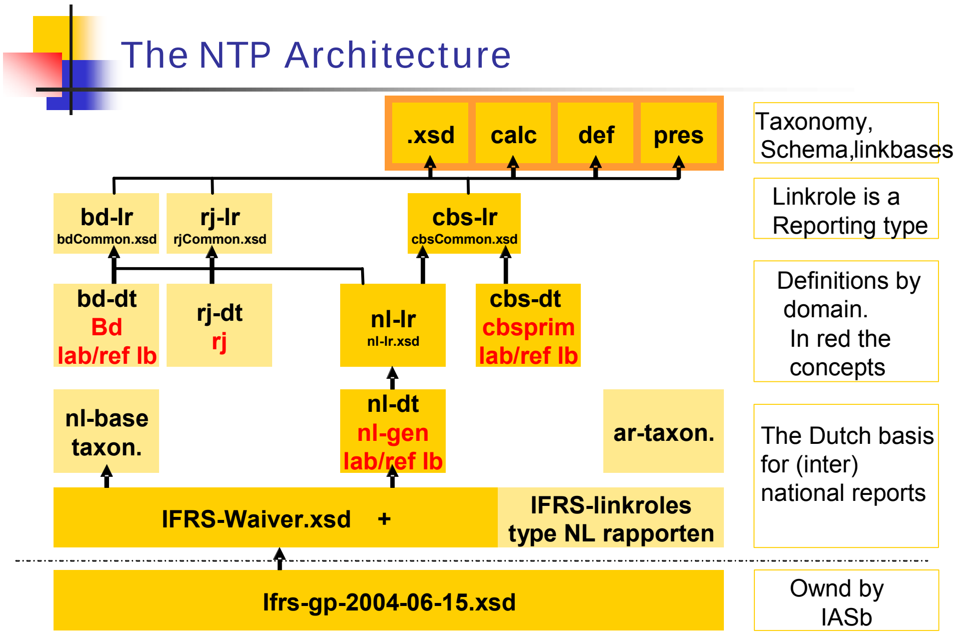


Figure 1. Overview of the Dutch taxonomy (version 0.1).

The Dutch Taxonomy Project (abbreviated in Dutch as NTP) is based on the International Financial Reporting Standards (IFRS) (16). Indeed the Dutch taxonomy physically includes its own IFRS waiver. Typically Dutch concepts have been added to the IFRS concepts by means of the nl-gen schema (xsd-file). Data types are stored in the nl-dt schema. These two schema files constitute the basis from which the tax department (bd), the Foundation for Annual Reporting (rj) and Statistics Netherlands (CBS) draw for their own reporting. These reports are called link roles in the NTP, and also in this document. The abbreviations 'Bd', 'rj' and 'cbsprim' in red in figure 1 are domain specific concepts used by the various link roles which are gathered in a 'domain'-lr taxonomy. Each link role consists of a specific schema, a number of linkbases and further uses (links to) concepts from the generic taxonomies, such as the nl-gen.

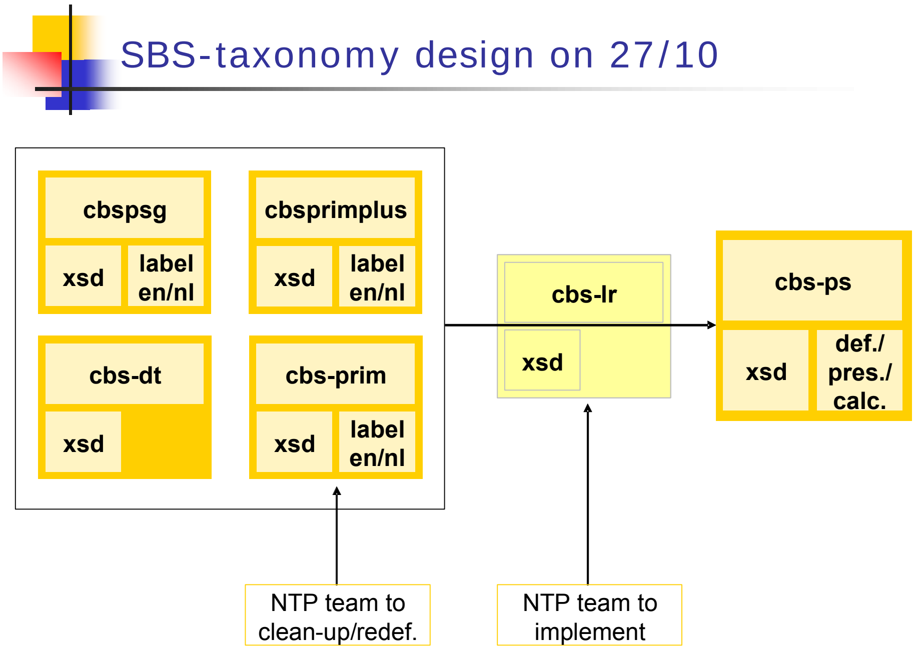
2.2 The Structural Business Statistics taxonomy

2.2.1 Design of the Structural Business Statistics taxonomy

The taxonomy of the Structural Business Statistics (SBS-) questionnaire had to comply with the XBRL version 2.1 specifications (5). At the same time it had to take into account the design of the Dutch taxonomy; i.e. make maximum use of basic concepts already present in the Dutch taxonomy. The concepts of Statistics Netherlands are located in the cbs-prim schema (figure 1). However, the Dutch taxonomy underwent a number of changes during the project (4). As such changes had a direct effect on the design of the SBS-taxonomy it was decided to base the prototype of the SBS-taxonomy on the version of the Dutch taxonomy available at the start of the project: version 0.1. The SBS-taxonomy was subsequently developed independently of the Dutch taxonomy. To this end, the cbs-prim schema of the Dutch taxonomy was incorporated integrally (as a copy) in the SBS-taxonomy. This also made it much easier to include any corrections in the cbs-prim. The cbs-prim file contained 506 concepts, a Dutch and an English label linkbase, a reference linkbase and an associated (imported) data type file (cds-dt).

Analysis of the pdf-version of the SBS-questionnaire for the wholesale resulted in an overview of all questionnaire variables and (if mentioned) their system code (10). Overall the questionnaire contained 164 different variables. Of these, 79 were present in the cbs-prim. The other variables were included in a questionnaire specific schema (cbs-psg) and a schema with variables of a more general nature (cbsprimplus). The latter schema contained 16 concepts which, in the view of the authors, were not questionnaire specific and therefore should have to be included in cbs-prim at a later point in time. In the cbs-psg file 69 questionnaire specific concepts and one abstract concept were defined. Annex B gives an overview of the SBS-concepts, their system code, Dutch labels and schemafiles in which the concepts are defined. A Dutch and an English label linkbase were prepared for each of the three schemafiles.

Together with the Statistics Netherlands data type schema (cbs-dt), The cbs-psg, cbs-prim and cbsprimplus files form the basis of the SBS-taxonomy. The concepts in these files (including their labels and data types) are generated from an umbrella cbs-ps schema. *No* concepts are defined in the cbs-ps schema file itself. At the level of the cbs-ps schema, the presentation, calculation and definition relations between the concepts are recorded. Figure 2 gives an overview of the layout of the SBS-taxonomy. An added advantage of this approach is that it will be very simple to include the cbs-ps as a separate linkrole -at a later stage- in the cbs-lr schema of the NTP. This is illustrated in figure 2 by the remark ‘NTP team to implement’. The latter remark is the consequence of the many problems occurring during the validation of the SBS-taxonomy (10,11). The majority of validation errors were caused by the original cbs-prim schema (see section 2.2.4).



SBS-taxonomy design on 27/10

Figure 2. Design of the Structural Business Statistics taxonomy.

2.2.2 *Software used*

The taxonomy was built and the instances were created with the aid of the Semansys XBRL Composer, the UBmatrix Automator and Fujitsu's Taxonomy Editor/Instance creator. Altova's XmlSpy was used to check the schema references at XML level. Validation of XBRL files was performed with the Semansys XBRL Composer, the UBmatrix Automator, Fujitsu's Taxonomy Editor/Instance creator, Fujitsu's Validator and DecisionSoft's TrueNorth validator. Small changes to XBRL files were predominantly made with a text editor and/or Altova's XmlSpy. For formula linkbase tests the alpha version of Fujitsu's Rule Base Editor and the UBmatrix Automator were used. Comparison of instance files was done with Altova's DiffDog. To test the effect of XSL(T) on XBRL instances Altova's XmlSpy and Internet Explorer were used.

2.2.3 *Linkbase files*

The schema files of the SBS-taxonomy in which the concepts are defined each contain a Dutch and an English label linkbase file. At that level no presentation, calculation and definition relations are recorded. A presentation linkbase for a questionnaire must be recorded at the level of the file, in which all concepts in the questionnaire are available. For the SBS-taxonomy this is the umbrella cbs-ps file. In principle, the presentation linkbase shows the questionnaire concepts in the order in which they appear in the questionnaire, except when calculation relations exist between the various concepts (10). The abstract questionnaire concept (annex B) acts as the ‘parent’ concept to which all other concepts are linked, and which serves as a basis for the numbering in rising order. When, at a later stage, the decision was made to adjust the taxonomy to the FRTA specifications (section 2.2.4), it turned out that it was necessary to create a separate presentation linkbase for the tuples defined in the cbs-prim. (12). For this reason alone a presentation linkbase was added to the cbs-prim (see figure 3).

The calculation relations mentioned on the questionnaire nearly all (with two exceptions) refer to calculations between (subtotal) concepts in the cbs-psg schema and (total) concepts in the cbs-prim or the cbs-primplus. Therefore it was decided to record all calculation relations at the level of the cbs-ps.

Relations described in the definition linkbase are exclusively ‘essence-alias’ relations between concepts (section 4.2.1.3). Such relations lay down that one concept in essence equals the other. As concepts involved in such relations were defined in different schema files, here, too, it was decided to record these in the definition linkbase of the cbs-ps.

During the development and checking of the SBS-taxonomy it turned out that the reference linkbase file of the cbs-prim was not required for the validity of the SBS-taxonomy. Therefore it was decided to remove this linkbase file (11, 12).

2.2.4 Taxonomy validation

During the construction of the taxonomy the separate schema files (cbs-psg, cbs-prim and cbsprimplus) and their accompanying linkbase files were always first validated at individual schema level. In the beginning of the project, validation took place at the (standard) level of the XBRL 2.1 specifications (5). Once the separate files were error-free, including ‘warning’ level errors, the cbs-ps schemafile was constructed. When the cbs-ps was faultless, the presentation, calculation and definition linkbase files were made. Subsequently, the entire taxonomy was validated. This brought to light errors which were the consequence of the combination of the separate schema files that were previously found without error. The cbs-prim file in particular turned out to be the source of many of such errors (10,11). After correcting these errors and rigours testing, the SBS-taxonomy was validated with all XBRL validation programs available. All programs found it without errors at the XBRL 2.1 level of checking.

After the taxonomy was found to be without error according to the above-mentioned step-by-step procedure, an instance was created. According to the authors, this is a very important step in the ultimate validation of a taxonomy. It is essential that such an instance contains all concepts included in the presentation linkbase of the taxonomy. Only when such an instance, after validation, is found to be without error, can we assume that the taxonomy, according to the validation level used during the checks, contains no errors.

Later on in the project it was decided to additionally validate the SBS-taxonomy according to the FRTA 1.0 specifications (8). These specifications are supplementary to the standard XBRL 2.1 specifications. The approach followed was essentially same as that described in the previous section. First the cbs-psg, cbs-prim and cbsprimplus files (and their linkbase files) were validated at FRTA level. Subsequently the whole SBS-taxonomy was checked. Next, an instance was made and the instance and taxonomy was checked. Errors found have been described in detail elsewhere (12). On the basis of these results, it is recommended that all future taxonomies of Statistics Netherlands be constructed and checked according to the procedure described in this chapter. It is also recommended that taxonomies shall only be published after they have been validated and found error-free at FRTA level.

2.2.5 Definite structure of the SBS-taxonomy

The construction of the final XBRL 2.1 and FRTA 1.0 compliant SBS-taxonomy is shown in figure 3. The figure shows that the cbs-dt schema, which contains the data types of Statistics Netherlands, is used by both the cbs-prim and the cbs-psg. This makes it possible to define the questionnaire specific and the cbs-generic data types in a single file. The use of data types is discussed in section 4.2.1.4.2.

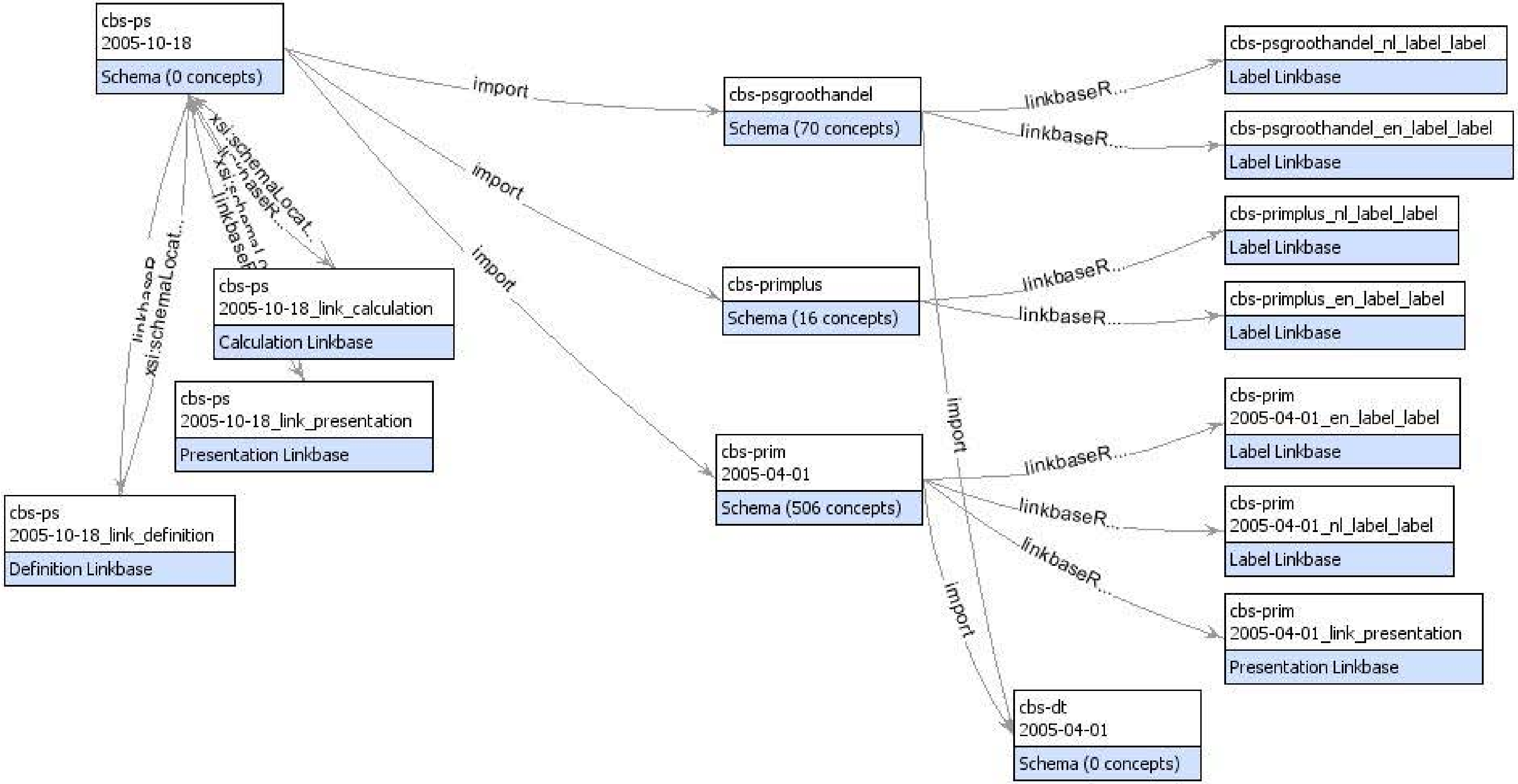


Figure 3. Structure of the XBRL 2.1 and FRTA-valid SBS-taxonomy.

3. Checking instances

3.1 Instances

When a taxonomy is error-free, an instance is created (section 2.2.4). To create a good instance, i.e. an instance that supplements the taxonomy validations, it is important that all concepts included in the presentation linkbase of the taxonomy occur. The user has to check this himself. Instances were validated with the aid of the XBRL validation programs available. A text editor was used to adjust and/or remove data in instances. This generated the many different correct and incorrect instances used in the tests listed in chapter 4.

3.2 Checks

For the SBS-questionnaire validation rules can be distinguished at various stages of processing. First of all on the questionnaire itself various relations between variables are mentioned. These are indicated as questionnaire checks in this document. Such checks include:

i) calculations (calculation relations)

ii) statements that one field equals another (alias field relations)

iii) descriptions that state that certain questionnaire variables constitute a subset of other variables (subset relations).

iv) instructions for entries in a field (format restrictions)

In addition, during the processing of the SBS-questionnaire data at Statistics Netherlands, all sorts of other rules are used. These consist of so-called validation (detection) and correction rules. A questionnaire must comply with the rules of the first group to be eligible for further (automatic) processing at Statistics Netherlands. After that step, in the data editing process, correction rules are applied. When that's the case, missing questionnaire data is filled in and/or certain values are adjusted. An overview of the validation and correction rules applied at Statistics Netherlands in the processing of the SBS-questionnaires has been drawn up by the project members. These rules are taken from:

i) workflow agreements which apply before a completed form is returned

ii) validation and correction rules of the processing system for statistical data

iii) checks related to the calculation of the plausibility index

iv) desired future comparisons with data from other questionnaires and surveys.

A total of 1054 different validation rules were collected. This group was compressed to a set of 58 representative validation rules. A total of 85 correction rules were collected. This collection was regrouped into six distinct subsets.

3.3 Technical methods

Data in an XBRL instance can be checked in various technical ways. The majority of these are XBRL and/or XML based. For an XBRL instance the following technical methods or techniques can be applied:

i) XBRL 2.1 specifications

ii) formula specifications

iii) using XSL-based techniques (XSLT)

iv) using script and a XBRL software program

v) custom-made software programs.

Each of the above-mentioned techniques has its own pros and cons. The order in which the solutions are listed, shows the approach followed by the authors in the test process (see chapter 4). If a method is successful in executing the desired check, it is no longer necessary to try a subsequent method. Obviously, it is possible to deviate from this in the test process.

The last method in the above list is only included as an option and will not be used in the XBRL 2005 project. It is the method currently used at Statistics Netherlands for the processing of SBS-questionnaire data.

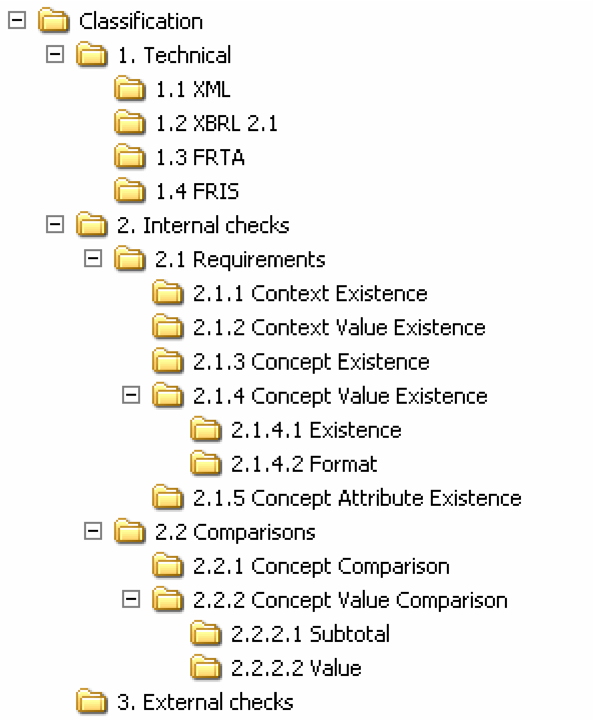
3.4 Classification approach

An issue often raised during the XBRL 2005 project was how the combination of rules and solution techniques could best be approached and discussed. (14). A precondition for this was the fact that the rules had to be executed by standard XBRL/XML technologies as much as possible. In the end a classification was chosen whereby first the rules (and probable) XBRL data characteristics of an instance were related to each other. This resulted in a classification in which XBRL instance components and/or characteristics (derived from the data characteristics and related to the rules) were linked to one or more of the technical solutions methods (12). The next chapter describes the design of the final classification and the results of the various tests carried out.

4. Classification

The classification schema of the rules and data characteristics has numbered main divisions. The subdivisions are also numbered. The classification consists of three main divisions: 1) Technology, 2) Internal checks and 3) External checks. This schema distinguishes clearly between checks according to technical specifications, checks relating to data of a single instance, and checks requiring data from more than one source.

The complete classification schema is illustrated in figure 4, which also shows the subdivision. This will be discussed in the present chapter, including the relation with the solution methods (section 3.3). For the classification of the checks, it is imperative that they – in the case of automatic checks and processing of instances – take place in the numbered order. To test the suitability of the possible technical methods one or more tests were designed and executed for each of the various classes. The methods and results are described in this chapter, which follows the structure of the classification schema.



- Classification
  - 1. Technical
    - 1.1 XML
    - 1.2 XBRL 2.1
    - 1.3 FRTA
    - 1.4 FRIS
  - 2. Internal checks
    - 2.1 Requirements
      - 2.1.1 Context Existence
      - 2.1.2 Context Value Existence
      - 2.1.3 Concept Existence
      - 2.1.4 Concept Value Existence
        - 2.1.4.1 Existence
        - 2.1.4.2 Format
      - 2.1.5 Concept Attribute Existence
    - 2.2 Comparisons
      - 2.2.1 Concept Comparison
      - 2.2.2 Concept Value Comparison
        - 2.2.2.1 Subtotal
        - 2.2.2.2 Value
  - 3. External checks

Figure 4. Overview of the classification schema.

4.1 Division 1: Technology

Division 1 includes all syntactical and architectural checks of an XBRL instance on the basis of internationally standardised specifications. A data file that does not pass a test from this division is not a technically correct XBRL instance (12).

4.1.1 XML

The complete file is based on data identification with the aid of the XML 1.0 notation. Instances do not have to contain valid XML; i.e. validated at XML level by a referred XML schema. So-called well-formed XML is sufficient; this implies that the instance XML is correct syntactically. An instance which does not contain well-formed XML is not a valid instance. XBRL validators and XML programs, such as Altova's XMLSpy, can carry out such checks.

4.1.2 XBRL 2.1

The XBRL 2.1 standard (5) describes how an XBRL instance, using W3C standards such as XML, XMLschema, XPath and XLink, must be constructed. This concerns both syntactical and semantic recommendations. An instance that does not comply to those recommendations is not a valid XBRL instance. Checks of the XBRL 2.1 standard can only by carried out by XBRL 2.1 validators (15).

4.1.3 FRTA

The Financial Reporting Taxonomies Architecture (FRTA) is a set of rules for taxonomies in the area of financial reporting (8), but is often also suitable for other fields. The FRTA contains a set of additional rules for XBRL 2.1 taxonomies. The reason that the FRTA is mentioned in this classification (which is based on checking instances) is that in the course of the project it turned out that the (additional) FRTA rules may affect the composition (and validity) of instances based on previous, non-FRTA compliant, versions of the taxonomy (section 2.2.4 and reference 12). The FRTA test is recommended as a consistency check for instances based on an FRTA valid taxonomy. Not all XBRL 2.1 validators, however, have the possibility of executing FRTA validation.

4.1.4 FRIS

The Financial Reporting Instance Standards (FRIS) describe a number of guidelines which can be applied to instances in the field of financial reporting (16). However, it was not possible in the project to validate instances with the aid of software on FRIS aspects. The XBRL validators used in the project had no, or no operational, FRIS control option. Neither was it possible to perform the FRIS conformance tests (17) reliably with these programs. Also, no other programs were found on the Internet that could perform such a check.

4.2 Division 2: Internal checks

This division contains all checks related to the relations of data within a single instance. It does not matter whether these relations are laid down in a taxonomy. The subdivision of this division cost the authors the most worries. Two sub-categories are distinguished, namely requirements and comparisons.

4.2.1 Requirements

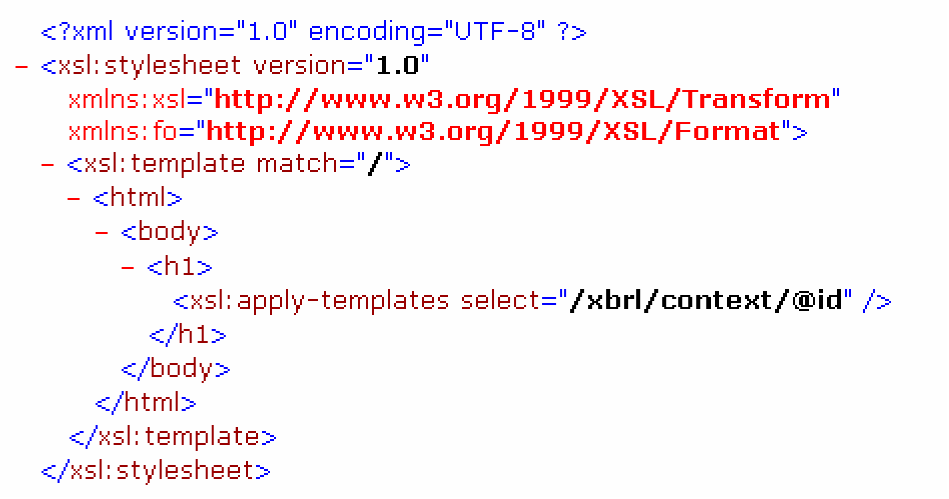
This group contains the checks which refer to the absolute requirements that an instance must fulfil. Only instances that comply with these checks are admitted for further processing.

4.2.1.1 Context existence

For most of the validators, checking an instance in which all contexts have been removed, produces an error for each missing concept in the instance. For the SBS-instance, with 165 concepts (section 2.2.1), this means a total of 165 error messages. Only one validator generates a single error message. The error description, however, does not make it very clear that the problem is a lacking context. Addition of extra contexts, that are not used by any of the concepts in the instance is reported by only two of the validators.

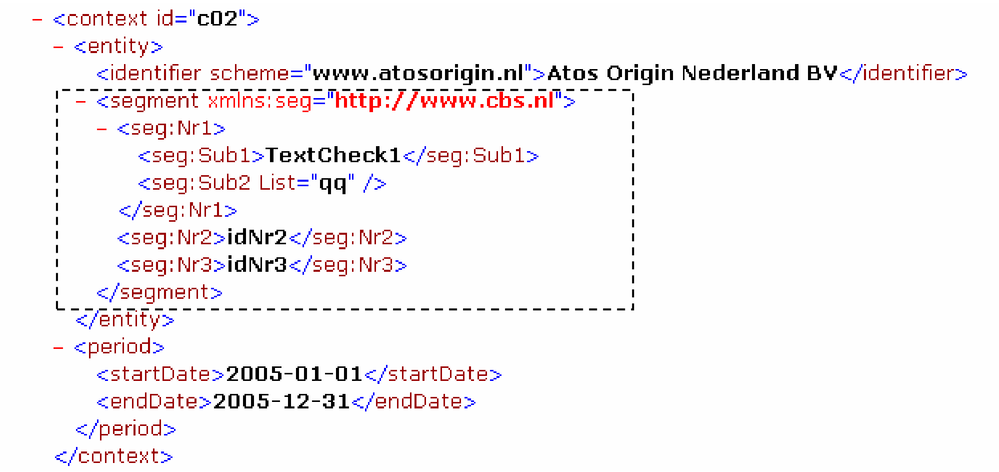
XSL(T) techniques are another way to check an instance for the presence or absence of context items. Figure 5 shows the code of an XSLT file. The effect of the file on an XBRL instance was tested with Altova's XmlSpy and Internet Explorer.

In the context part of an instance, a segment and a scenario part can be included. The XBRL specifications allow additional elements to be included in the segment and/or the scenario of an instance (5). To illustrate this possibility, the segment of a SBS-instance is extended with (XML) elements which are not defined in the SBS-schema files. This is done by including a piece of 'valid' XML code in the segment of an instance. The code refers to the XML schema definition of the extra included



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fo="http://www.w3.org/1999/XSL/Format">
  - <xsl:template match="/">
    - <html>
      - <body>
        - <h1>
            <xsl:apply-templates select="/xbrl/context/@id" />
          </h1>
        </body>
      </html>
    </xsl:template>
  </xsl:stylesheet>
```

Figure 5. XSLT which gives as result all id's of the contexts in an instance.



```
- <context id="c02">
  - <entity>
      <identifier scheme="www.atosorigin.nl">Atos Origin Nederland BV</identifier>
    - <segment xmlns:seg="http://www.cbs.nl">
      - <seg:Nr1>
          <seg:Sub1>TextCheck1</seg:Sub1>
          <seg:Sub2 List="qq" />
        </seg:Nr1>
        <seg:Nr2>idNr2</seg:Nr2>
        <seg:Nr3>idNr3</seg:Nr3>
      </segment>
    </entity>
  - <period>
      <startDate>2005-01-01</startDate>
      <endDate>2005-12-31</endDate>
    </period>
  </context>
```

Figure 6. Illustration of additional data in the segment of the context of an instance. The inserted part is boxed.

elements with an own unique name-space prefix (figure 6). Correct validation by an XBRL validator is only possible if the name space and reference to the schema definition are included in the cbs-ps-2005-10-18.xsd (taxonomy) file. It is therefore not completely independent of taxonomy. To check the correct treatment by XBRL validation programs an enumeration list is defined in the added xsd-schema for the element ‘Sub2’ (see figure 6). As a result, only a limited set of values is allowed for this element. The results of the validation of an instance with a non-permitted and a permitted value for the Sub2-element by XBRL validation programs vary. The majority of the programs validate the instances correctly. One program, however, does not accept the enumeration list of the segment component of the adjusted taxonomy, and correct instance validation is therefore not possible.

4.2.1.2 Context value existence

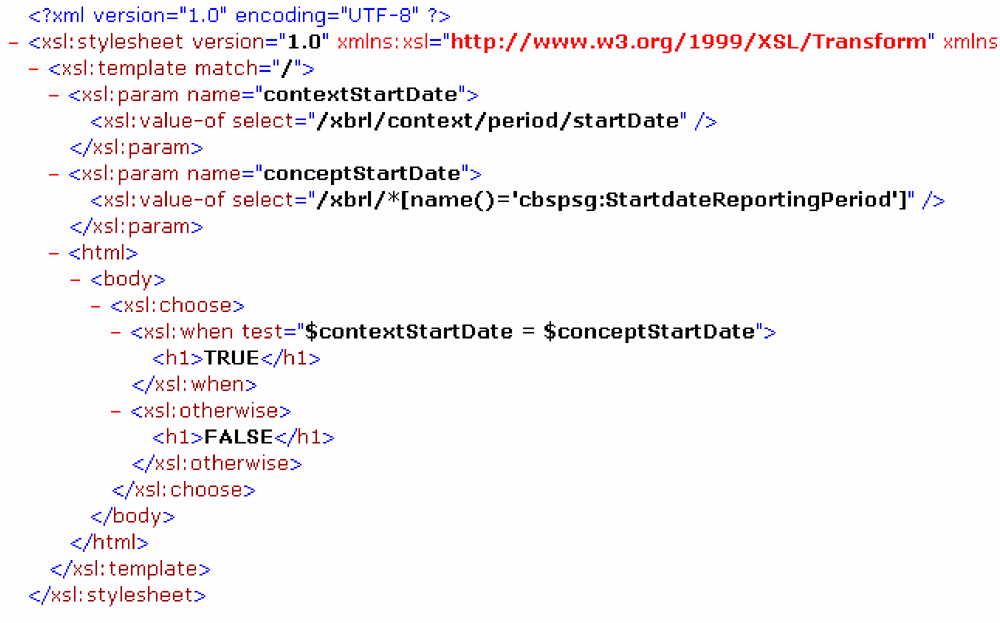
This section describes how XBRL software copes with (unknown) values in the context and unit part of an instance. This concerns both the values of attributes and those of the underlying elements. Both the context and the unit parts comprise an id-attribute which must be referred to by a concept. None of the XBRL validation programs accepts the lack of such id’s.

In a simple instance the context element contains at least an entity and a period element. The identifier element is always present within the ‘entity’, and a segment-part – described in the previous section - may also be present. The identifier element is used to identify the owner of the instance; this can be done in either the text of the element or in the schema attribute. According to two of the validators the identifier element does not have to contain text, the schema attribute does. The other software programs require a value for both.

The period element in the context is related to the period type of a concept defined in the taxonomy. Each concept has a periodType (5). A period type of a concept may be 'instant' or 'duration' (explained below). When an instance is created and validated, the period type definitions of the concepts are compared to the content of the period element in the context(s). The period element of a context can be 'instant', containing a real (actual) date, or can comprise a period ('duration'), with a starting date ('startDate') and an end date ('endDate'). The start and end date elements must both contain real dates, and the end date must be later than the starting date. Combinations of a concept and a context with different period types give errors in all XBRL software. Erroneous dates, such as 30 February and 29 February in a non-leap year are not spotted some of the programs, but are by others. Missing dates and wrongly dated starting and end dates are spotted by all programs. Some programs replace a missing date by a default date, e.g. '1899-12-30'.

A questionnaire nearly always contains a field which states the period to which the survey refers. In an XBRL instance it is possible to include such dates in the context or in concepts. To test both possibilities, concepts are included in the SBS-taxonomy which contain the starting and end dates of the period under review; namely 'StartdateReportingPeriod' and 'EnddateReportingPeriod'. This makes it possible to include start and end dates of the period under review in both concepts and in a context. These dates cannot be compared with the XBRL validation programs. This is possible using an XSLT. Figure 7 shows an XSLT in which the starting date in the context is compared with the reported starting date in the concept.

The unit component in an instance is used for the univocal definition of the unit of a value. The unitRef attribute in a concept indicates with which unit a value must be interpreted. The data type of a concept is defined in the taxonomy. A concept may



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns
  - <xsl:template match="/">
    - <xsl:param name="contextStartDate">
        <xsl:value-of select="/xbrl/context/period/startDate" />
      </xsl:param>
    - <xsl:param name="conceptStartDate">
        <xsl:value-of select="/xbrl/*[name()='cbspsg:StartdateReportingPeriod']" />
      </xsl:param>
    - <html>
      - <body>
        - <xsl:choose>
          - <xsl:when test="$contextStartDate = $conceptStartDate">
              <h1>TRUE</h1>
            </xsl:when>
          - <xsl:otherwise>
              <h1>FALSE</h1>
            </xsl:otherwise>
          </xsl:choose>
        </body>
      </html>
    </xsl:template>
  </xsl:stylesheet>
```

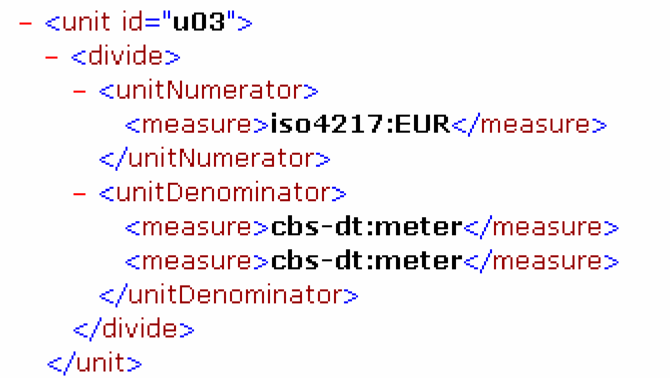
Figure 7. XSLT which compared with starting date in the context and the reported starting date in the concept.

be data type ‘Monetary’, for example; i.e. the concept refers to a sum of money. In the instance, the unit component indicates in which currency the value of the concept is recorded by means of the unitRef attribute. For example: dollars or euros. The FRIS standards propose the use of standard units only for ‘unit’, and not the derivates (16). So metres may be used for a unit, but not millimetres, centimetres etc. as they are a scale factor of metres (16). The FRIS recommendations entail that all concept values must be fully written numbers of the right unit and processing applications must use the concept attributes ‘decimals’ or ‘precision’ in forms or reports.

The unit component consists of one or more measure elements or the element divide. The latter comprises a ‘unitNumerator’ and a ‘unitDenominator’, which in turn consist of one or more measure elements. Units with more than one measure element indicate that the units they refer to have to be multiplied. In this way it is possible to define complex units. Figure 8 demonstrates this by defining the unit ‘euros per square metre’ in an XBRL instance. All XBRL software programs used spot the difference between the data type of a concept and the corresponding unit defined in the ‘measure’ element. The lack of a correct unitRef and the lack of the required ‘decimals’ or ‘precision’ attribute following from the data type are spotted (see also section 4.2.1.5).

4.2.1.3 Concept existence

Generally, when an XBRL instance is created, only those concepts are included to which a value has been assigned (by the user). A concept to which no value has been assigned will, in principle, not occur in the XBRL instance. However, the XBRL 2.1 specifications contain a possibility to require the user to include a certain concept in an instance (5). In the definition linkbase a so-called ‘requires-element’ arcrole can be assigned to a concept. This obliges the user to assign a value to that concept. The ‘requires-element’ arcrole lays down a relation between two concepts: a ‘child’ and a ‘parent’ respectively. If the parent concept is present in the instance, the child

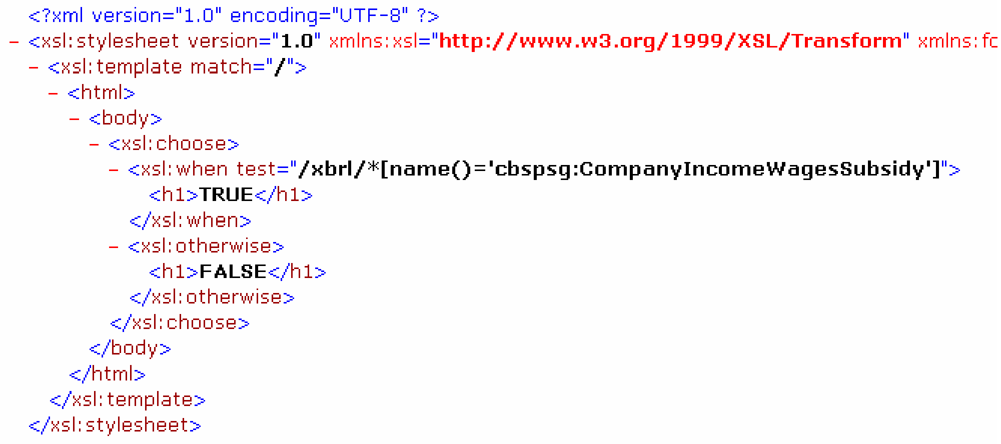


```
- <unit id="u03">
  - <divide>
    - <unitNumerator>
        <measure>iso4217:EUR</measure>
      </unitNumerator>
    - <unitDenominator>
        <measure>cbs-dt:meter</measure>
        <measure>cbs-dt:meter</measure>
      </unitDenominator>
    </divide>
  </unit>
```

Figure 8. Notation of the unit Euros per square metre (€/m^2) in an XBRL instance. The name space prefix ‘cbs-dt’ refers to the origin of the unit ‘metre’ (in Dutch: meter).

concept must also be present. All validation programs check this correctly. The great disadvantage of the 'requires-element' approach is that in the absence of the parent concept, the child concept is no longer required to be present. This makes the 'requires-element' arcrole only suitable for laying down relations between two concepts for which it can be said that if one occurs (parent) the other must also occur (child). The 'requires-element' arcrole cannot be used to demand the presence (under all circumstances) of a specific concept in an instance. This is not possible in the present version of the XBRL specifications. At a first glance, a practical solution for that problem could be to send a partly pre-filled instance to respondents. However, under those circumstances it is still possible to return a valid instance in which a specific previously pre-filled concept (this could be the parent concept) has been removed. A more practical solution, would be to include the attribute value 'required' in the 'use' attribute in the XBRL specification (identical to the XML schema specification).

Absolute checks of the presence or absence of certain concepts in an instance can be done by XSLT. With this an instance can be checked for occurrence or non-occurrence of specific concepts. Figure 9 shows an XSLT by which an instance can be checked for the presence or absence of the concept 'CompanyIncomeWagesSubsidy'.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:fc
  - <xsl:template match="/">
    - <html>
      - <body>
        - <xsl:choose>
          - <xsl:when test="/xbrl/*[name()='cbspsg:CompanyIncomeWagesSubsidy']">
              <h1>TRUE</h1>
            </xsl:when>
          - <xsl:otherwise>
              <h1>FALSE</h1>
            </xsl:otherwise>
          </xsl:choose>
        </body>
      </html>
    </xsl:template>
  </xsl:stylesheet>
```

Figure 9. XSLT which checks for presence of the cbs-psg concept CompanyIncomeWagesSubsidy.

4.2.1.4 Concept value existence

In essence, the value of a concept can be determined at two different levels. At the first level a check is done to see if a concept has a value at all. At the second level a check is done to see whether the format of the value corresponds to what was expected. Both possibilities were tested.

4.2.1.4.1 Existence

In the taxonomy, the attribute 'nillable' can be added to the definition of an XBRL concept. The value of this attribute can be set to 'false' or 'true' (5). Originally, the ‘nillable’ attribute was an XML-schema attribute (18) used to indicate whether an XML element may occur without a value in an instance. To enable its use in XBRL the following requirements must be met:

1) in the taxonomy the value of the nillable attribute for the concept must be set to ‘true’.
2) In the instance the namespace ‘http://www.w3.org/2001/XMLSchema-instance’ with the preferred prefix xsi must be added.
3) in the instance the attribute xsi:nil= “true” must be added to the concept concerned.
4) in the instance for that concept, if the concept is a decimal or a derived type, the attribute ‘decimal’ or ‘precision’ must be removed.

When these four requirements are fulfilled, it is possible to create a valid instance in which the value for the concept concerned is missing. In all other cases an error is reported by all validators. By assigning the value ‘false’ to the nillable attribute for a certain concept in the taxonomy, respondents will be prevented from supplying an instance which contains no value for that concept. However, as described earlier, when -during the creation of an instance in an XBRL software program- no value is assigned to a concept, the concept will not be included in the instance (see section 4.2.1.3).

For XBRL instance, the value of the nillable attribute is no longer relevant. This means that the existence of a value for a concept can only be enforced if the user is required to include that concept in the instance. This is exactly the problem already discussed in section 4.2.1.3. The nillable attribute serves absolutely no useful function in XBRL. Only if the user can be forced -in some way or another- to include a certain concept in an instance would it be possible to put the nillable attribute to use. Under those circumstances, setting the nillable attribute to ‘false’ for that concept in the taxonomy will prevent a respondent (with a lot of effort) from supplying a valid instance in which the value is missing for that concept.

4.2.1.4.2 Format

The format of the value of a concept can be enforced by means of a more detailed specification of the data type. In XBRL this can be done by means of a mask or an list of possible values. For the SBS-taxonomy examples of both possibilities are recorded in the cbs-dt schema (figure 3). This was tested for the correspondence number, the ‘jetstar codes’[1], the end date of the survey, the document reference number, the standard industrial classification (SBI), the statistical code and the survey period. Error values entered for these concepts were spotted by all validators.

[1] Jetstar codes are codes used to register the receipt of the questionnaire at Statistics Netherlands.

One program was able to check the format *during* the creation of the instance another. Another program showed, after a value was put in, the list of possible values.

4.2.1.5 Concept attribute existence

A concept in an instance should have at least the attribute ‘contextRef’. Without this attribute, or with a wrong value for it, all XBRL validators give an error message (section 4.2.1.2). All numerical (decimal) or numerically derived concepts must also contain the attribute ‘unitRef’. Without this attribute, or with a wrong value for it, all XBRL validators give an error message (section 4.2.1.2). In addition to ‘unitRef’ numerical or numerically derived concepts must also contain the attribute ‘decimals’ or ‘precision’. The only exception to this are concepts derived from fractionItemType (5) or concept that may be nillable and contain no value (section 4.2.1.4.1).

The attributes ‘decimals’ and ‘precision’ are used to indicate the precision of the reported value of a concept. A ‘precision’ value may be nil, a positive whole number or INF. INF expresses that the value of the concept is as precise as it is reported. Nil means that the precision of the value is not known. A positive whole number provides the number of digits that may be considered as reliable. For example, a turnover of 100 Euro with precision 2 means that the turnover is expected to be a value between 95 and 105 Euro. For ‘decimals’ the same applies in essence. However, this attribute can also take whole negative values. For ‘decimals’ the value nil means that the decimal point indicates the limit of precision. INF indicates that the number is as precise as it is recorded. Positive numbers are used to indicate the number of reliable digits after the point, while negative numbers give the number of accurate digits before the point. For exampled: turnover 100 with precision 2 is as precise as turnover 100.0 with decimals -1. The precision attribute is in fact a special case of decimals, as the latter attribute can be used to express the same and more.

If ‘decimals’ or ‘precision’ are missing in a numerical concept, or either is used for a non-numerical concept, all XBRL validators interpret this as incorrect. This also holds for the -non-permitted- combination of the two. However, as soon as different values are used for the attributes, interpretation by the validators differs. If the precision of a concept involved in a calculation relation diminishes, all validators -except one- report that the calculation is incorrect. For example: for the calculation relation: concept A + concept B = concept C, the addition of concept A with value 25 and a precision of 1 (or decimals -1) and concept B with value 30 (and precision 2 or decimals 0) will no longer equal the value 55 of concept C (precision 2 or decimals 0). The most striking difference between the validators was obtained when the precision attribute was assigned the value 0 in a numerical concept. All validators except one did not report an error in such as case. This is not correct. According to the XBRL specifications, the value of a numerical concept (e.g. 25) with precision=0 should be interpreted as being value nil (5). Only one validator interprets this correctly. If for the same concept the analogous decimals attribute is used (for value 25 this is decimals = -2), the majority of the validators interpret this correctly. One validator was unable to interpret negative decimal values. From the above it will be clear that the presence or absence of the decimals and precision attributes do not pose any problems. These are interpreted correctly in all circumstances by all XBRL programs. Differences in interpretation occur when the values of these attributes differ. Only a single validator is truly able to interpret the effects of various precision and/or decimals attributes on the concept value correctly in all circumstances. In view of the above findings, it is very important that XBRL instance creation programs are extremely careful in assigning precision and decimals attributes.

4.2.2 Comparisons

The checks in this group constitute the required set of rules to validate an instance according to taxonomy and business directives (logic).

4.2.2.1 Concept comparison

To compare concepts the essence-alias relation in the definition linkbase can be used (section 4.2.1.3). By doing this, it is established in the taxonomy that two concepts in fact describe the same variable. A consequence of the essence-alias relation is that the value, the data type, the context type and the unit must be the same for both concepts. The name must differ, however. Surprisingly, the context dates and the precision do not have to be identical (5). Errors in the obligatory points of correspondence are spotted by all XBRL validators. Only one validator attempts to compare the precision of both concepts with the aid of the precision and/or decimals attributes. The program however does not report an error when the value of the essence-alias concepts differs. All other validators do report an error when that is the case.

4.2.2.2 Concept value comparison

There are two sublevels of comparing values between concepts. The first group, the subtotals, include the calculation relations. Examples of such relations are: Concept A + Concept B = Concept C and Concept D – Concept E = Concept F. The other sublevels includes the more arithmetically complex relations such as ‘smaller and greater than’ comparisons. Examples of these are: Concept A + Concept B < Concept D and Concept E >= Concept F. The latter can be laid down in the formula linkbase. At the time of writing, however, the formula linkbase recommendations are not yet part of the official XBRL 2.1 specifications (5).

4.2.2.2.1 Subtotal

Simple calculation relations between concepts can be laid down in the calculation linkbase. The simplest calculation relation is Concept A = Concept B. The concepts are required to have identical data type, period type, context and unit. Otherwise no error message will be generated in the case of a calculation difference. For example, for concepts A and B in the above simple calculation relation, no error will be reported if the value of concept A is 25 and that of B 30, and one has euros (€) as unit and the other US dollars ($). If the concepts fulfil the above-mentioned requirements, an error will be reported by all validators when the value of the total concept does not correspond with that of the subtotal concepts (according to the relations laid down in the calculation linkbase). In view of the extensive possibilities of the formula linkbase, it is also be possible to lay down calculation relations in the formula linkbase.

4.2.2.2.2 Value comparison

Comparisons of concept values with more complex calculation relations or comparison of a concept value with a constant is possible in the formula linkbase (6). Examples of such comparisons are: Concept A <= Concept B, Concept C + Concept D > Concept E and Concept F < Constant A. At the time of writing the formula specifications were not yet official, however. They are still in the 'candidate' stage. At the start of the project we knew that a number of XBRL software designers were developing a program or adding support to existing programs that made it possible to create a formula linkbase. Two XBRL software programs with formula linkbase capabilities were tested. In the programs not all specifications of the formula linkbase recommendations had (yet) been implemented. This meant that only a limited number of checks could be tested.

By way of example, we shall discuss the results of the check of a subset relation (section 3.2). Such a relation indicates that one concept is a subset of another. Consequently the value of the former concept must be smaller or equal to that of the latter. According to the additional information on the SBS-questionnaire this is, for instance, the case for the concepts CompanyIncomeIntraConcernSevices and CompanyIncomeOtherNotMentionedElsewhere. To check this, two instance equations were entered in the formula linkbase files of both programs. The equations tested were:

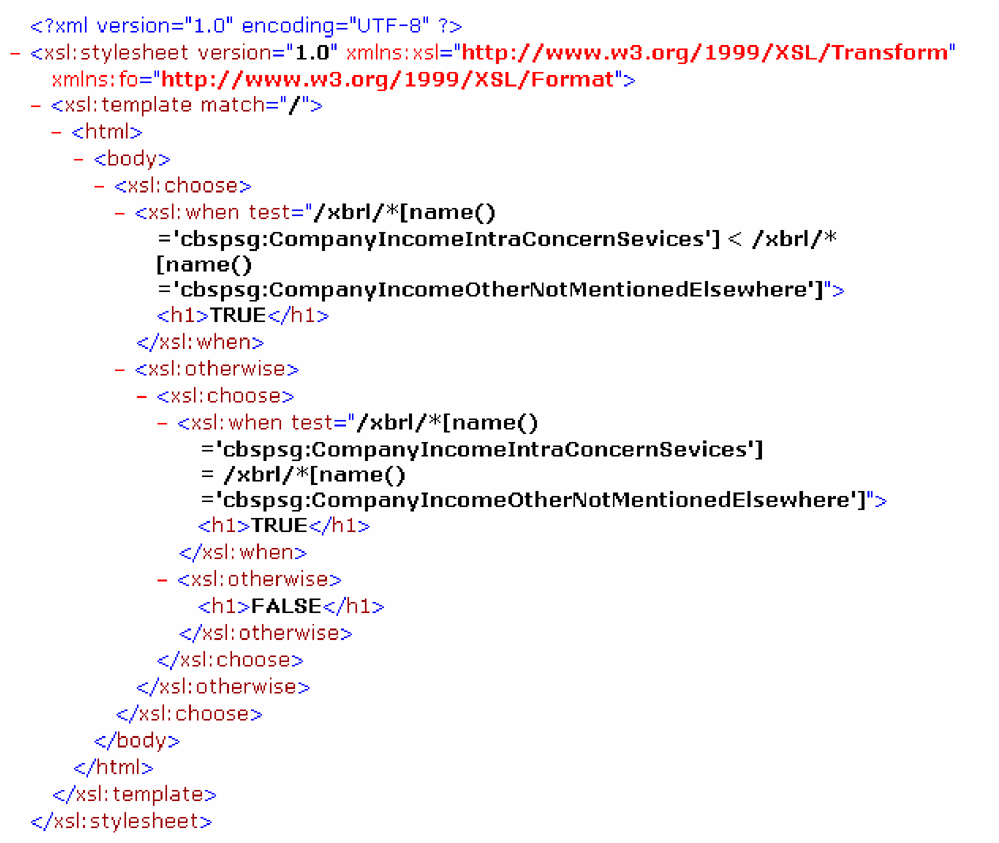
(1) CompanyIncomeIntraConcernSevices <= CompanyIncomeOtherNotMentionedElsewhere

(2) CompanyIncomeIntraConcernSevices > CompanyIncomeOtherNotMentionedElsewhere

If the values of the concepts in the instance satisfy the first relation, 'true' was shown and/or stored in a result concept created. Depending on the program used, this result could be stored in a separate of added to the original instance file. If the values satisfied the second relation, the result concept obtained the value 'false'. This turned out to work very well. In the same way the value of a concept in an instance could be compared with that of a constant, defined in the linkbase file.

Unfortunately, both programs did not allow the comparison of concepts of different periodType. According to the formula recommendation this is the intention (6). The latter is a very interesting option for Statistics Netherlands. For instance, this would make it possible to check whether the difference between the start and end value of a concept (both of the 'instant' context type) correspond with the total changes in that period (all 'duration' context type). The calculation linkbase does not allow this (5).

With an XSLT it is also possible to compare values of concepts within an instance. By way of illustration we made an XSLT which performs the same checks as the formula example discussed above. The content of the XSLT is shown in figure 10. With an XSLT it is also possible to perform calculations.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fo="http://www.w3.org/1999/XSL/Format">
  - <xsl:template match="/">
    - <html>
      - <body>
        - <xsl:choose>
          - <xsl:when test="/xbrl/*[name()
                ='cbspsg:CompanyIncomeIntraConcernSevices'] < /xbrl/*
                [name()
                ='cbspsg:CompanyIncomeOtherNotMentionedElsewhere']">
                <h1>TRUE</h1>
            </xsl:when>
          - <xsl:otherwise>
            - <xsl:choose>
              - <xsl:when test="/xbrl/*[name()
                    ='cbspsg:CompanyIncomeIntraConcernSevices']
                    = /xbrl/*[name()
                    ='cbspsg:CompanyIncomeOtherNotMentionedElsewhere']">
                    <h1>TRUE</h1>
                </xsl:when>
              - <xsl:otherwise>
                    <h1>FALSE</h1>
                </xsl:otherwise>
              </xsl:choose>
            </xsl:otherwise>
          </xsl:choose>
        </body>
      </html>
    </xsl:template>
  </xsl:stylesheet>
```

Figure 10. XSLT which compares the value of two concepts.

4.3 Division 3: External checks

This division includes checks where the data in an XBRL instance is compared with externally recorded information. Examples of the latter are another instance or a database. The rules in this division do not have any effect on the technical processability of an XBRL instance. They focus completely on business logic and may even deviate from the rules defined in the taxonomy. An example of rules that belong to this group are the plausibility indices of the Impect system of Statistics Netherlands (19).

Because of the short duration of the project and the time needed for working out the internal checks (section 4.2), after consultation it was decided to work out only a very limited set of external comparisons. This meant that comparisons of instance data with data in (external) databases were not studied. Such comparisons can be so complex, that it is questionable whether these checks can be tested with anything other than a custom-made software tool. What was tested was the comparison of concept values with a constant. The approach is exactly identical to that described in section 4.2.2.2.2. Comparisons of data between instances has also been examined.

An example of this is the comparison of values of the same concept in two different response periods. This can be done by comparing (the data of) two XML files. Standard XML solutions for this are using an XSLT or by means of an XML instance comparison program, such as Altova's DiffDog. Both can and have been applied to XBRL instances.

5. Rules and solutions

The XBRL 2005 project comprises three different rules:

1) Questionnaire rules, i.e. rules which describe the checks on the questionnaire of Statistics Netherlands.
2) Validation rules, i.e. rules used to check the data in an instance (at Statistics Netherlands)
3) Correction rules, i.e. rules used to check and correct the data in an instance (at Statistics Netherlands)

In this chapter the classification and the solution discussed in the previous chapter are run back through the various rules to verify the completeness of the classification. If it turned out that certain relations were missing, these would as yet be recorded in the classification schema. This was not the case.

5.1 Questionnaire rules

The four different rules on the questionnaire (section 3.2) can all be described with the aid of the found solution methods. The calculations listed on the questionnaire, for example, can all be recorded as calculation relations in the calculation linkbase (section 4.2.2.2.1). Alias fields on the questionnaire can be described with the aid of essence-alias relations in the definition linkbase (section 4.2.2.1). The notes on the questionnaire that one field is a subset of another field, the so-called subset relations, can be checked with formula linkbase equations (section 4.2.2.2.2). Format restrictions for entry in a field can be laid down by defining XBRL data types (section 4.2.1.4.2).

5.2 Validation rules

The rules in this group are used during the automatic processing of the questionnaire data at Statistics Netherlands. Instances which pass all checks are eligible for further (automatic) processing. A total of 1054 different validation rules were collected. This group was condensed to a set of 58 representative validation rules. For each of these validation rules it was examined which solution method was best suited for recording it.

In an attempt to give an overview, we have drawn up a table showing for each rule the actual check, a short description of the check, the classification of the rule, the recommended solution method and a reference to the section in which this method is discussed (table 1).

Table 1. Overview of the validation rules and solution methods

Nr.	Controleregel vergelijking	Controleregel uitleg	Oplossingsclassificatie	Oplossingsmethode	Paragraaf verwijzing
1	IF EIND_BOEKJAAR <> 0 THEN YEAR(EIND_BOEKJAAR) = STATJR ENDIF	Periode eindigt niet in statistiek jaar	Context waarde vergelijking & Concept waarde vergelijking	XSLT	4.2.1.2
2	OPBRENG100000 >= 0	Negatieve waarde onwaarschijnlijk	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
3	PERSONS100000 = SUBTOTA100000 - PERSONS122000	Optelling totaal aantal werkzame personen klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
4	IF PERSONS110100 <> 0 THEN PERSONS111000 <> 0 ENDIF	Wel aantal werknemers op eigen loonlijst, maar niet herleid tot VTE's	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
5	IF WD_ID<>3697 THEN VERKOOP210000 >= OPBRENG100000 ENDIF	Totaal van overige bedrijfsopbrengsten is groter dan totaal van netto omzet	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
6	VERKOOP210000 = VERKOOP211000 + VERKOOP212000	Optelling netto omzet klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
7	VERKOOP211000 >= VERKOOP212000	Nevenactiviteit is groter dan hoofdactiviteit. Mogelijk typeringsfout	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
8	IF (LOONSOM110000 > 5 OR PERSONS110100 > 2) THEN LOONSOM121100 <> 0 ENDIF	Wel brutolonen en salarissen of werknemers, maar geen sociale lasten	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
9	IF LOONSOM121100 + LOONSOM121200 + LOONSOM122000 <> 0 THEN LOONSOM110000 > LOONSO	Sociale lasten zijn groter dan de brutolonen en salarissen	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
10	BEDRLST341000 = BEDRLST341100 + BEDRLST341200 + BEDRLST341300	Optelling totaal energiekosten klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
11	BEDRLST346000 = BEDRLST346100 + BEDRLST346200 + BEDRLST346300 + BEDRLST346400 + BEDR	Optelling van verkoopkosten klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
12	RESULTS130000 = RESULTS120000 + FINREST100000 + VOORZNG100000 + BUIBATN100000	Resultaat voor belasting klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
13	RESULTS120000 = OPBRENH000000 - BEDRLSH310000	Saldo bedrijfsresultaat klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
14	IF (PERSONS100000 <> 0 OR PERSONS110000 <> 0) THEN VERKOOP210000 <> 0 ENDIF	Wel werkzame personen per 30 september, maar geen netto omzet	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
15	IF OPBRENG112000 <> 0 THEN PERSONS122000 <> 0 ENDIF	Wel vergoedingen voor uitgeleend personeel ontvangen, maar geen uitgeleend personeel per 30 september	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
16	IF PERSONS111000 <> 0 THEN LOONSOM100000 <> 0 ENDIF	Wel werknemers op eigen loonlijst per 30 september, maar geen arbeidskosten	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
17	IF (PERSONS113000 + PERSONS130000 + PERSONS121000 <> 0 AND (PERSONS110100 = 0)) THEN L	Geen werknemers per 30 september, maar toch arbeidskosten	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
18	IF BEDRLST345000 <> 0 THEN (PERSONS100000 <> 0 OR PERSONS110000 <> 0) ENDIF	Wel andere personeelskosten, maar geen werkzame personen per 30 september	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
19	IF (PERSONS100000 <> 0 OR PERSONS110000 <> 0) THEN LOONSOM100000 + BEDRLST345000 <> 0	Wel werkzame personen per 30 september, maar geen arbeidskosten en andere personeelskosten	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
20	IF PERSONS121000 <> 0 THEN BEDRLST345200 <> 0 ENDIF	Wel overig ingeleend personeel per 30 september, maar geen betalingen i.v.m. overig ingeleend personeel	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
21	BEDRLST310000 = BEDRLSH310000	Totaal bedrijfslasten in bedrijfslastenblok ongelijk aan totaal bedrijfslasten in resultatenblok	Concept vergelijking	Definition linkbase	4.2.2.1
22	IF GK_COORDINATIE_ABR = 1 THEN PERSONS100000 < 10 ENDIF	Typeringsfout! GK en werkzame personen verschillen sterk van elkaar	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
23	IF GK_COORDINATIE_ABR = 2 THEN (PERSONS100000 > 0 AND PERSONS100000 < 15) ENDIF	Typeringsfout! GK en werkzame personen verschillen sterk van elkaar	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
24	IF GK_COORDINATIE_ABR = 9 THEN PERSONS100000 > 399 ENDIF	Typeringsfout! GK en werkzame personen verschillen sterk van elkaar	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
25	IF INKWRDE110000 >= INKWRDE130000 ENDIF	Bij deze SBI is de verwachting dat inkoopwaarde handelsgoederen groter is dan totaal van inkoopwaarde ov	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
26	PERSONS111000 >= PERSONS110120	Aantal parttimers is groter dan totaal aantal werknemers per 30 september	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
27	IF VERKOPH212000T.1 <> 0 THEN INKWRDE110000 <> 0 ENDIF	Wel handelsomzet, maar geen inkoopwaarde	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
28	INKWRDE110000 = INKWRDH110000	Inkoopwaarde handelsgoederen omslagvel komt niet overeen met inkoopwaarde inlegvel	Concept vergelijking	Definition linkbase	4.2.2.1
29	SUBSIDI100000 >= SUBSIDI130000 + SUBSIDI140000 + SUBSIDI150000	Som van specificatie subsidies groter dan totaal aan subsidies	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
30	OPBRENG100000 >= SUBSIDI130000 + SUBSIDI140000 + SUBSIDI150000 + OPBRENG111000	Som van specificaties overige bedrijfsopbrengsten inlegvel groter dan totaal overige bedrijfsopbrengsten oms	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
31	IF SUM(EXPORTS210100T) = 0 THEN EXPORTS210100 = SUM(EXPORTS210100T) ENDIF	Telling bedrijfsopbrengsten binnen EU klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
32	VERKOOP210000 >= EXPORTS210100 + EXPORTS210200	Netto omzet buitenland groter dan totale omzet	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
33	IF SUM(IMPORTS100100T) = 0 THEN IMPORTS100100 = SUM(IMPORTS100100T) ENDIF	Telling inkopen goederen en diensten binnen EU klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
34	IF SUM(IMPORTS100200T) = 0 THEN IMPORTS100200 = SUM(IMPORTS100200T) ENDIF	Telling inkopen goederen en diensten buiten EU klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
35	BEDRLST310000 >= IMPORTS100100 + IMPORTS100200	In het buitenland gekochte goederen en diensten zijn groter dan totale bedrijfslasten	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
36	VERKOPH212000 >= 0	Onterecht negatieve waarde	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
37	IF SUM(VERKOPH211300T) = 0 THEN VERKOPH211300 = SUM(VERKOPH211300T) ENDIF	Telling omzetspecificatie hoofdactiviteit klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
38	VERKOOP211000 >= VERKOPH211200	Netto omzet hoofdactiviteit inlegvel komt niet overeen met omslagvel	Concept vergelijking	Definition linkbase	4.2.2.1
39	VERKOOP211000 >= VERKOPH211300	Netto omzet hoofdactiviteit inlegvel komt niet overeen met omslagvel	Concept vergelijking	Definition linkbase	4.2.2.1
40	IF INKWRDH110000T.1 >= INKWRDH110000T.2 ENDIF	Inkoopwaarde detailhandel groter dan groothandel. Typeringsfout?	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
41	IF ((VOORRAH210100 <> 0) OR (VOORRAH210500 <> 0)) THEN INKWRDE110000 <> 0 ENDIF	Wel voorraden handelsgoederen, maar geen inkopen	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
42	IF ((TMIN1.MEDIAAN_PERSONS110000) <> 0) AND ((TMIN1.MEDIAAN_VERKOOP210000) <> 0) AND (((I	Mogelijke duizendfout op bedrijfsopbrengsten	Externe waarde vergelijking	Software programma	4.3
43	IF ((TMIN1.MEDIAAN_PERSONS110000) <> 0) AND ((TMIN1.MEDIAAN_LOONSOM100000) <> 0) AND ((T	Mogelijke duizendfout overige bedrijfslasten	Externe waarde vergelijking	Software programma	4.3
44	EIND_BOEKJAAR > BEGIN_BOEKJAAR	Begindatum boekjaar groter dan einddatum	Context waarde vergelijking of Concept waarde vergelijking, toetswaarde	XSLT of Formula linkbase	4.2.1.2 of 4.2.2.2.2
45	AFSCHRG110000 <> 0	Geen afschrijvingen op materiële en immateriële vaste activa	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
46	IF VERKOOP210000 <> 0 THEN (PERSONS100000 <> 0 OR PERSONS110000 <> 0) ENDIF	Wel netto omzet, maar geen werkzame personen per 30 september	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
47	IF TMIN1.EIND_BOEKJAAR <> EMPTY THEN MONTH(TMIN1.EIND_BOEKJAAR) = MONTH(EIND_BOEK	Boekjaarperiode wijkt af ten opzichte van het vorig jaar	Vergelijking tussen instanties	XSLT of DiffDog achtig programma	4.3
48	IF RECHTSVORM = 'NAAMLOZE VENNOOTSCHAP' OR RECHTSVORM = 'BESLOTEN VENNOOTSCHAF	Bij een rechtspersoonlijkheid bezittend bedrijf moeten altijd arbeidskosten voorkomen	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
49	IF ((BTW_JAAROMZET) <> 0) AND ((VERKOOP210000) <> 0) THEN VERKOOP210000 <= 300*(BTW_JA	Duizendfout bij vergelijking met BTW omzet	Externe waarde vergelijking	Software programma	4.3
50	IF VERKOOP210000 <> 0 THEN INKWRDE100000 <> 0 ENDIF	Wel omzet, maar geen inkoopwaarde. Typeringsfout?	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
51	IF SUM(EXPORTS210100T) <> 0 THEN EXPORTS210100 = SUM(EXPORTS210100T) ENDIF	Telling bedrijfsopbrengsten binnen EU klopt niet	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
52	BEGIN_BOEKJAAR <> EMPTY	Begin boekjaar niet ingevuld	Context waarde existentie of Concept existentie	XSLT of Definition linkbase	4.2.1.2 of 4.2.1.3
53	IF PERSONS110120 > 5 THEN PERSONS111000 > PERSONS110100 ENDIF	Doordat er parttimers werkzaam zijn op 30 september, dient het aantal werknemers groter te zijn dan het aar	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
54	IF (PERSONS130000 + PERSONS121000) <> 0 THEN BEDRLST345400 <> 0 ENDIF	Wel uitzendkrachten en/of overig ingeleend personeel per 30 september, maar geen betalingen i.v.m. uitzenc	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
55	IF PERSONS100000 >= PERSONS111000 THEN PERSONS110000 >= PERSONS110100 ENDIF	Als het totaal aantal werkzame personen groter of gelijk is aan de werknemers op de eigen loonlijst is de verv	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
56	BEDRLST348000 >= BEDRLST348600	Betaalde vrachtkosten aan derden inlegvel groter dan omslagvel	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
57	IF INKWRDE100000<>0 THEN VERKOOP210000<>0 ENDIF	Wel inkoopwaarde, maar geen omzet	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2
58	IF OPBRENG000000 <> 0 THEN VERKOOP210000 <> 0 ENDIF	opbrengsten zonder omzet	Concept waarde vergelijking, toetswaarde	Formula linkbase of XSLT	4.2.2.2.2

Table 2. Overview of the correction rules and solution methods for the validation part

Nr.	Correctieregel vergelijking	Correctieregel uitleg	Controledeel uitleg	Oplossingsclassificatie controledeel	Oplossingsmethode controledeel	Paragraaf verwijzing
1	If BEDRLST310000 < 0 then BEDRLST310000:= abs(BEDRLST310000) Endif	Positief maken van variabelen	Negatieve waarde controle	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2
2	If VERKOOP210000 = VERKOOP212000 and (VERKOOP210000 = VERKOOP211000) then VERKOOP212	Het corrigeren van dubbele waarden. (alleen binnen industrie)	Alias veld waarde vergelijking	Concept vergelijking	Definition linkbase	4.2.2.1
3	If SUBTOTA100000 = 0 or SUBTOTA100000 = empty and (PERSONS111000 <> empty or PERSONS1130(	Het vullen van lege (sub)totalen	Calculatie totaal waarde controle of Concept totaal existentie controle	Concept waarde vergelijking, toetswaarde of Concept existentie	Calculatie linkbase of XSLT	4.2.2.2.1 of 4.2.1.3
4	If (INKWRDE100000) <> 0 and INKWRDE110000 + INKWRDE130000 = 0 then INKWRDE110000:= INKWF	Vullen van lege specificaties met als doel betere startsituatie voor CherryPi te creere	Calculatie totaal waarde controle of Concept totaal existentie controle	Concept waarde vergelijking, toetswaarde of Concept existentie	Calculatie linkbase of XSLT	4.2.2.2.1 of 4.2.1.3
5	If ((Extern.BTW_Jaaromzet < 1) or (Extern.KS_Jaaromzet < 1)) and ((Plausibiliteit.PI_Actief.k1 = Ja) or (Exte	Herstellen van duizendfouten	Duizendfout controle	Externe waarde vergelijking	Software programma	4.3
6	If RESULTS130000 <> RESULTS120000 + FINREST100000 + VOORZNG100000 + BUIBATN100000 then	regels voor het resultatenblok	Calculatie vergelijking	Concept waarde vergelijking, subtotaal	Calculatie linkbase	4.2.2.2.1
7	If (-RESULTS120000 = OPBRENH000000 - BEDRLSH310000) and (RESULTS130000 <> 0) and (-RESULT	regels voor het resultatenblok	Complexe calculatie vergelijking	Concept waarde vergelijking, toetswaarde	Formula linkbase	4.2.2.2.2

From table 1 it can be seen that all validation rules are covered by the classification schema. If the first solution is consistently chosen for rules with more than one solution, it turns out that most rules can be laid down in a formula (i.e. 36) or in a calculation linkbase (i.e. 12). The remaining 10 checks contain 4 definition linkbase relations, 3 XSLT-solutions and 3 checks that can only be performed with the aid of a custom-made software program. The latter group consist of complex relations which check for the presence of so-called ‘multiple of a thousand’ errors. At this moment we think that, after some further study, it might also be possible to carry out a ‘multiple of a thousand’ check (or some of them) with an XSLT.

5.3 Correction rules

In total, 85 correction rules were collected. This collection was grouped into a set of 6 distinct rules. All correction rules contain an initial check, after which, depending on the outcome, values, whether or not determined, are edited in one way or another. Correction rules can be very complex. If we look at only the initial control part of a correction rule, it turns out that the 85 correction rules collected can be reduced to 6 types of relations. Taking into account the above-mentioned 6 distinct subgroups, this –in combination with the solutions– gives a total of 7 distinct rule and solution combinations. An example of each of these is given in table 2.

Table 2 shows that the control part of the correction rules is also completely covered by the classification. Compared with the validation rules in the previous section, this set of rules is mainly restricted to concept related rules. The latter can mostly be described with standard XBRL solutions and formula linkbase relations. Here, too, the only exception was a check for a “multiple of a thousand” error. Here, the same solutions as those suggested above apply.

6. Conclusions and recommendations

This chapter gives an overview of the most important conclusion and recommendations following from the XBRL 2005 project.

6.1 Conclusions

The work recorded in this document and in the project documentation (2,9-12,14) constitutes a sound basis for the design of an XBRL workflow system. It is important to realise that the results generalised here were realised by studying one statistical questionnaire and taxonomy. It is clear how all sorts of checks that can be done on an XBRL instance based on the SBS-taxonomy for the wholesale. A large part of the concept-related checks can be recorded within the XBRL specifications and the formula specifications. For context-related checks and concept-existence checks XSLT techniques must be used. An XBRL workflow system must therefore at least offer the facilities to apply both solutions.

The only sort of checks for which it seems necessary to use a specific software program are the rules that check for the presence of possible 'multiple-of-thousand' errors. Further studies will show whether it is possible to carry out such checks during the creation or analysis of XBRL instances in another way. Possible solutions are: checks during data entry, or a complex XSLT.

6.2 General recommendations

1) XBRL specifications are still subject to change. At the moment work is being done on the extension of the specifications with context dimensions (20) and the formula linkbase (6). Validation checks created during this project must be tested and adjusted or expanded in a new release of the XBRL specifications.

2) Because XBRL technology is still relatively young (see previous item), the XBRL software products cannot yet be expected to be fully fledged. This may partly be the result of the typical characteristics (sometimes peculiarities) of the various XBRL objects. Also, it is not always clear how they are used. For example the use of the definition of the attribute 'substitution group' in a schema with respect to an 'item' and 'tuple' (5).

3) Because of the relatively peculiar structure of XBRL instances[2], there is a risk that reporting parties cannot perform a suitable technical validation themselves. Assistance in the shape of validation products and/or validation certification is desirable for a smooth further processing of instances by the requesting party/parties.

[2] Compared to XML instances.

6.3 XBRL recommendations

The following recommendations are made to the XBRL specifications. The project team would like these to be put forward to the XBRL organisation.

1) Add an option to the XBRL specifications to make it compulsory for a concept to occur in an instance. This is not possible in the present XBRL specifications. One way to enable this is to add the option ‘required’ to the ‘use’ attribute in a linkbase arc. At present only the options ‘optional’ and ‘prohibited’ are allowed (5). In the XML specifications the ‘required’ option is also possible (21). The ‘required’ option can be added to the presentation or to the calculation linkbase.

2) Remove the balance attribute from the XBRL schema definition. According to the XBRL specifications it is not necessary to use the balance attribute in the definition of a concept (5). Its use leads to additional calculation restrictions, however, which, in our opinion, are unnecessary and confusing. For example: is the difference between debit and credit from the points of view of the payer or the receiver. Which view is used in the definition? Such confusion should certainly be prevented for complex calculations. It would therefore be better to remove the ‘balance’ attribute from the XBRL specifications.

3) Remove the precision attribute from the XBRL schema definition. According to the XBRL specifications (5) the precision attribute can be used to indicate the precision of a numerical value. In such cases it is also possible to use the decimals attribute. Precision is in fact a simplification of decimals. Everything precision can express, and more, can be expressed by the attribute decimals (section 4.2.1.5). Therefore use of the precision attribute is no longer necessary. The additional advantage is that the indication of the precision of the value of a concept is given in a single way. This could also result in a considerable improvement of the performance of the processing of instances. It is very likely that this will also benefit the inter-compatibility of XBRL validators.

4) Remove the nillable attribute, or assign it the standard the value ‘false’. Tests show that, for the SBS-taxonomy, absence of the nillable attribute was given the standard value ‘true’ (12). With some effort, this makes it possible to include a concept without value in a valid instance. In our opinion this could not have been the intention of this XML-derived attribute (18). It is therefore recommended that the nillable attribute be removed from the XBRL specifications. Should this result in insurmountable problems, the attribute should be set, by default, to ‘false’ in XBRL schemas. The latter will prevent instance processing from stopping when there is an (unexpected) empty concept in an instance.

5) Use the formula linkbase to record calculation relations. If the formula linkbase specifications are definitely accepted by the XBRL organisation, it is recommended that the calculation linkbase no longer be used. For new taxonomies it is better to record all calculation relations in a single linkbase, the formula linkbase. In this respect, it is uncertain whether the calculation linkbase has a role at all to play if the formula linkbase is operational.

6.4 NTP recommendations

The project team did not entirely understand some of the choices made in the 0.1 version of the NTP:

1) The minOccurs attribute has a different value in the ifrs-gp than in the nl-gen. In the ifrs-gp, the attribute minOccurs has the value ‘0’ for by far most of the concepts. In the nl-gen, all ifrs-gp concepts are standard imported with minOccurs= “1”.

2) The complete URL is not always used in imports of schemas in the NTP. It is important to state the whole URL, not only the name of the schema file. For example: in the cbs-dt file, the attribute schemaLocation="xbrl-linkbase-2003-12-31.xsd" is used. this should be replaced by schemaLocation=” http://www.xbrl.org/2003/xbrl-linkbase-2003-12-31.xsd”, otherwise the URL refers to an xsd-file on the local PC. This cannot be the intention.

6.5 Final remarks

In addition to the validation and control of XBRL instances it is important that in the future attention is also paid to the storage of XBRL instance data. If use is made of relational databases, such as SQL-server, it will not be completely trivial to store instance data in a set table structure. Especially if concepts are not always present.

7. Annexes

Annex A: Abbreviations/acronyms

Bd/bd	Dutch tax authority (In Dutch: Belastingdienst)
CBS	Statistics Netherlands (In Dutch: Centraal Bureau voor de Statistiek)
cbs-dt	Parts of the NTP in which the CBS data types are defined
cbs-prim	Parts of the NTP in which the CBS concepts are included
cbsprimplus	Parts of the SBS-taxonomy in which the questionnaire generic concepts are included which do not occur in the cbs-prim
cbs-ps	Umbrella parts of the SBS-taxonomy. Here the concepts from the cbs-prim, cbsprimplus, cbs-psg and the data type of the cbs-dt are combined.
cbs-psg	Parts of the SBS-taxonomy in which the questionnaire-specific concepts are included
dt	Data type
FRIS	Financial Reporting Instance Standards
FRTA	Financial Reporting Taxonomy Architecture
GAAP	Generally Accepted Accounting Principles
IAS	International Accounting Standards
IFRS	International Financial Reporting Standards
Impect	Implementation of the economic transformation process (In Dutch: Implementatie Economisch Transformatieproces)
Impect code	Code for a variable (concept) used in the Impect system
INF	Infinite precision (exact value)
jr	Council for annual accounts (In Dutch: Raad voor de jaarrekening)
lr	linkrole
NTP	Dutch Taxonomy Project (In Dutch: Nederlands Taxonomieproject)
pdf	Portable Document Format
SBI	Standard Industrial Classification (In Dutch: Standaard Bedrijfs-indeling)
SBS	Structural Business Statistics (in Dutch: Productie Statistiek)
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XBRL	eXtensible Business Reporting Language
XLink	XML Linking Language
XML	eXtensible Markup Language
XPath	XML Path Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations

Annex B: Overview of concepts in SBS-taxonomy

Impect Code	Nederlands label	Taxonomie bestandsnaam
	Vragenlijst ProductieStatistieken Groothandel groot	cbspsg (abstract)
	CBS Interne Identificatie van de betreffende Statistiek	cbspsg
	CBS Interne Identificatie Naam van de betreffende Statistiek	cbspsg
	Periode waarover geenquetteerd wordt.	cbspsg
	Standaard Bedrijfsindeling	cbspsg
	Correspondentie nummer	cbspsg
	Referentie documentnummer	cbspsg
	Uiterste inzenddatum	cbspsg
	Postadres controle code1	cbspsg
	Postadres controle code2	cbspsg
PERIODE100000	Begindatum van het boekjaar waarop deze opgave betrekking heeft	cbspsg
PERIODE200000	Einddatum van het boekjaar waarop deze opgave betrekking heeft	cbspsg
PERSONS111000	Totaal aantal werknemers op de eigen loonlijst per 30 september	cbsprimplus
PERSONS110100	Totaal aantal werknemers op de eigen loonlijst per 30 september omgerekend in voltijdequivalenten (VTE)	cbsprimplus
PERSONS113000	Overige werkzame personen per 30 september medewerkende eigenaren en gezinsleden	cbsprimplus
PERSONS130000	Overige werkzame personen per 30 september uitzendkrachten / gedetacheerd personeel (van uitzendbedrijven)	cbsprimplus
PERSONS121000	Overige werkzame personen per 30 september overig ingeleend personeel (van andere bedrijven)	cbsprimplus
SUBTOTA100000	Werkzame personen subtotaal aantal werknemers en overige werkzame personen	cbsprimplus
PERSONS122000	Aantal van het totaal aantal werknemers op de eigen loonlijst per 30 september dat was uitgeleend (aan andere bedrijve	cbsprimplus
PERSONS100000	Totaal aantal personen werkzaam in uw bedrijf	cbsprimplus
PERSONS110000	Totaal aantal personen werkzaam in uw bedrijf omgerekend in voltijdequivalenten (VTE)	cbs-prim
VERKOOP211000	Netto omzet uit de hoofdactiviteit van de onderneming	cbsprimplus
VERKOOP212000	Netto omzet uit overige activiteiten	cbsprimplus
VERKOOP210000	Totaal netto omzet	cbsprimplus
OPBRENG112000	Ontvangen voor uitgeleend personeel	cbspsg
INVESTN130000	Geactiveerde productie ten behoeve van eigen bedrijf	cbspsg
SUBSIDI100000	Ontvangen restituties en subsidies	cbspsg
ONTVANG100000	Uitkeringen verzekeringen i.v.m. bedrijfsschade en schade aan goederen	cbs-prim
OPBRENG110000	Overige opbrengsten n.e.g.	cbspsg
OPBRENG100000	Overige bedrijfsopbrengsten (Totaal)	cbs-prim
OPBRENG000000	Totale Bedrijfsopbrengsten	cbs-prim
INKWRDE110000	Inkoopwaarde handelsgoederen	cbsprimplus
INKWRDE130000	Overige Inkoopwaarde	cbsprimplus
INKWRDE100000	Inkoopwaarde van de omzet	cbsprimplus
LOONSOM110000	Bruto lonen en salarissen	cbs-prim
LOONSOM121100	Werkgeversdeel sociale verzekeringspremies	cbs-prim
LOONSOM121200	Pensioen VUT premies en inkoopsommen/dotaties	cbs-prim
LOONSOM122000	Niet wettelijk verplichte werkgeversbijdragen /uitkeringen	cbs-prim
LOONSOM100000	Arbeidskosten	cbs-prim
BEDRLST345100	Betaald aan uitzendbedrijven	cbs-prim
BEDRLST345200	Betaald voor ingeleend personeel	cbs-prim
BEDRLST345500	Kosten studie en opleiding	cbs-prim
BEDRLST345900	Overige personeelskosten	cbs-prim
BEDRLST345000	Overige kosten m.b.t. personeel	cbs-prim
BEDRLST342100	Huur en operationele lease van vervoermiddelen	cbs-prim
BEDRLST342200	Reparatie en onderhoud van vervoermiddelen	cbs-prim
BEDRLST342300	Brandstofkosten vervoermiddelen (LPG diesel benzine)	cbs-prim
BEDRLST342400	Motorrijtuigenbelasting	cbs-prim
BEDRLST342500	Kosten verzekering vervoermiddelen	cbs-prim
BEDRLST342900	Overige autokosten	cbs-prim
BEDRLST342000	Kosten van vervoermiddelen	cbs-prim
BEDRLST341200	Verbruik van electriciteit	cbs-prim
BEDRLST341100	Verbruik van gas	cbs-prim
BEDRLST341300	Verbruik van overige energiedragers	cbs-prim
BEDRLST341000	Energieverbruik	cbs-prim
BEDRLST343100	Huur en operationele lease van gebouwen en terreinen	cbs-prim
BEDRLST343200	Reparatie en onderhoud van gebouwen en terreinen	cbs-prim
BEDRLST343300	Schoonmaakkosten van gebouwen en terreinen	cbs-prim
BEDRLST343410	Milieuheffingen en zuiveringslasten	cbs-prim
BEDRLST343420	Onroerendezaakbelasting	cbs-prim
BEDRLST343500	Kosten verzekering gebouwen en terreinen	cbs-prim
BEDRLST343900	Overige huisvestingskosten	cbs-prim
BEDRLST343000	Kosten van huisvesting	cbs-prim
BEDRLST344100	Huur en operationele lease van mach. Invent. Install. Computers e.d.	cbs-prim
BEDRLST344200	Reparatie en onderhoud van machines installaties inventaris computers	cbs-prim
BEDRLST344900	Overige machine en inventariskosten	cbs-prim
BEDRLST344000	Kosten van machines inventaris installaties e.d.	cbs-prim
BEDRLST346100	Reclame beurs en representatiekosten	cbs-prim
BEDRLST346200	Agentenprovisie	cbs-prim
BEDRLST346300	Reis en verblijfskosten	cbs-prim
BEDRLST346400	Research en ontwikkelingskosten	cbs-prim
BEDRLST346900	Overige verkoopkosten	cbs-prim
BEDRLST346000	Verkoopkosten	cbs-prim

BEDRLST347000	Communicatiekosten	cbs-prim
BEDRLST348100	Bank en incassokosten	cbs-prim
BEDRLST348200	Overige verzekeringspremies	cbs-prim
BEDRLST348300	Accountants en advieskosten	cbs-prim
BEDRLST348400	Automatiseringskosten	cbs-prim
BEDRLST348500	Vuilafvoer/verwerking	cbs-prim
BEDRLST348900	Overige kosten van dienstverlening door derden	cbs-prim
BEDRLST348000	Kosten van dienstverlening door derden	cbs-prim
BEDRLST349100	Kosten van octrooi en licentierechten	cbs-prim
BEDRLST349200	Betaling voor intraconcerndiensten / beheerkosten	cbs-prim
BEDRLST349300	Kantoorbehoeften abonnementen contributies	cbs-prim
BEDRLST349400	Overige huur en operationele lease	cbs-prim
BEDRLST349500	Reparatie en onderhoud van overige activa	cbs-prim
BEDRLST349600	Overige kostprijsverhogende belastingen	cbs-prim
BEDRLST349900	Overige algemene kosten n.e.g.	cbs-prim
BEDRLST349000	Overige bedrijfslasten	cbs-prim
AFSCHRG110000	Afschrijvingen op vaste activa	cbs-prim
BEDRLST310000	Totale bedrijfslasten	cbs-prim
RESULTS120000	Totaal Bedrijfsresultaat	cbs-prim
FINREST110000	Financieele baten	cbs-prim
FINREST120000	Financieele lasten	cbs-prim
FINREST100000	Financieel resultaat	cbs-prim
VOORZNG140000	Onttrekkingen en vrijval van voorzieningen	cbs-prim
VOORZNG110000	Toevoegingen aan voorzieningen	cbs-prim
VOORZNG100000	Saldo voorzieningen	cbs-prim
BUIBATN110000	Buitengewone baten	cbs-prim
BUIBATN120000	Buitengewone lasten	cbs-prim
BUIBATN100000	Saldo Buitengewone baten/lasten	cbs-prim
RESULTS130000	Resultaat voor belastingen	cbs-prim
PERSONS110120	Deeltijdwerkers	cbspsg
VERKOPH211200	Netto omzet uit groothandel in schoolmeubelen	cbspsg
VERKOPH211200	Netto omzet uit groothandel in bedrijfsmeubelen	cbspsg
VERKOPH211200	Netto omzet uit groothandel in scheepsbenodigdheden en visserijartikelen	cbspsg
VERKOPH211200	Netto omzet uit groothandel in vakbenodigdheden tuinbenodigdheden en dergelijke	cbspsg
VERKOPH211200	Netto omzet uit groothandel in verpakkingsmateriaal van papier en karton	cbspsg
VERKOPH211200	netto omzet uit groothandel in verpakkingsmateriaal van hout	cbspsg
VERKOPH211200	Netto omzet uit groothandel in verpakkingsmateriaal van glas	cbspsg
VERKOPH211200	Netto omzet uit groothandel in verpakkingsmateriaal van kunststof	cbspsg
VERKOPH211200	Netto omzet uit groothandel in overig verpakkingsmateriaal	cbspsg
VERKOPH211200	Netto omzet uit groothandel in overige goederen	cbspsg
VERKOPH211200	Netto omzet uit de hoofdactiviteit van de onderneming naar artikelgroepen	cbspsg
VERKOPH211300	Netto omzet naar afnemerscategorieën industriële ondernemingen	cbspsg
VERKOPH211300	Netto omzet naar afnemerscategorieën bouwondernemingen	cbspsg
VERKOPH211300	Netto omzet naar afnemerscategorieën groothandel inclusief exporteurs en grossiers	cbspsg
VERKOPH211300	Netto omzet naar afnemerscategorieën detailhandel	cbspsg
VERKOPH211300	Netto omzet naar afnemerscategorieën horecaondernemingen	cbspsg
VERKOPH211300	Netto omzet naar afnemerscategorieën andere ondernemingen en bedrijfsmatige gebruikers	cbspsg
VERKOPH211300	Netto omzet uit de hoofdactiviteit naar afnemerscategorieën	cbspsg
VERKOPH211400	Netto omzet naar verkoopkanalen verkopen via internet of andere elektronische netwerken	cbspsg
VERKOPH211400	Netto omzet naar verkoopkanalen verkopen via andere verkoopkanalen	cbspsg
VERKOPH211400	Netto omzet uit de hoofdactiviteit van de onderneming naar verkoopkanalen	cbspsg
VERKOPH212000	Netto omzet uit overige activiteiten detailhandel	cbspsg
VERKOPH212000	Netto omzet uit overige activiteiten handelsbemiddeling	cbspsg
VERKOPH212000	Netto omzet uit overige activiteiten industriele activiteiten	cbspsg
VERKOPH212000	Netto omzet uit overige activiteiten verhuur onroerende zaken	cbspsg
VERKOPH212000	Netto omzet uit overige activiteiten overige dienstverlening	cbspsg
VERKOPH212000	Netto omzet uit overige activiteiten overige activiteiten	cbspsg
VERKOPH212000	Netto omzet uit overige activiteiten	cbspsg
SUBSIDI140000	Loon(-kosten) subsidies	cbspsg
SUBSIDI150000	Export- en overige restituties en subsidies ingevolge EU-regelingen	cbspsg
SUBSIDI130000	Overige subsidies	cbspsg
OPBRENG111000	Doorberekende beheerskosten / Intraconcerndiensten	cbspsg
INKWRDH110000	Inkoopwaarde handelsgoederen groothandelsgoederen	cbspsg
INKWRDH110000	Inkoopwaarde handelsgoederen	cbspsg
INKWRDH110000	Totaal inkoopwaarde handelsgoederen	cbsprimplus
INKWRDH130000	Overige inkoopwaarde loondiensten werk door derden enzovoort	cbspsg
INKWRDH130000	Overige inkoopwaarde inkoopwaarde grond en hulpstoffen	cbspsg
INKWRDH130000	Overige inkoopwaarde overige inkoopwaarde niet eerder genoemd	cbspsg
INKWRDH130000	Totaal overige inkoopwaarde	cbsprimplus
BEDRLST348600	Vrachtkosten op de verkopen	cbs-prim
EXPORTS210100	Omzet binnen de EU m.u.v. Nld geëxporteerde goederen	cbspsg
EXPORTS210100	Omzet binnen de EU m.u.v. Nld transitohandel	cbspsg
EXPORTS210100	Omzet binnen de EU m.u.v. Nld overige	cbspsg
EXPORTS210100	Totalen netto omzet binnen de EU m.u.v. Nld	cbs-prim
EXPORTS210200	Omzet buiten de EU geëxporteerde goederen	cbspsg

EXPORTS210200	Omzet buiten de EU transitohandel	cbspsg
EXPORTS210200	Omzet buiten de EU overige	cbspsg
EXPORTS210200	Totalen netto omzet buiten de EU	cbs-prim
IMPORTS100100	Bedrijfslasten gemaakt binnen de EU (excl. Nld) geïmporteerde handelsgoederen	cbspsg
IMPORTS100100	Bedrijfslasten gemaakt binnen de EU (excl. Nld) transitohandel	cbspsg
IMPORTS100100	Bedrijfslasten gemaakt binnen de EU (excl. Nld) overige	cbspsg
IMPORTS100100	Bedrijfslasten gemaakt binnen de EU (excl. Nld)	cbs-prim
IMPORTS100200	Bedrijfslasten gemaakt buiten de EU geïmporteerde handelsgoederen	cbspsg
IMPORTS100200	Bedrijfslasten gemaakt buiten de EU transitohandel	cbspsg
IMPORTS100200	Bedrijfslasten gemaakt buiten de EU overige	cbspsg
IMPORTS100200	Bedrijfslasten gemaakt buiten de EU	cbs-prim
VOORRAH210100	Beginvoorraden handelsgoederen groothandelsgoederen	cbspsg
VOORRAH210100	Beginvoorraden handelsgoederen detailhandelsgoederen	cbspsg
VOORRAH210100	Totalen beginvoorraden handelsgoederen	cbs-prim
VOORRAH210500	Eindvoorraden handelsgoederen groothandelsgoederen	cbspsg
VOORRAH210500	Eindvoorraden handelsgoederen detailhandelsgoederen	cbspsg
VOORRAH210500	Totalen eindvoorraden handelsgoederen	cbs-prim

References

1. Website XBRL international, http://www.xbrl.org/.
2. Lammers, J. (2005), *ProjectBrief XBRL 2005.* CBS prince2 document, version 1.1p1, Aug. 24, Methods and Informatics Department, Statistics Netherlands, Heerlen.
3. Website XBRL international, XBRL's History (2006), retrieved Jan. 2, http://www.xbrl.org/history.aspx.
4. Website of the Dutch Taxonomy Project, http://www.XBRL ntp.nl/.
5. Extensible Business Reporting Language 2.1, Recommendations (2005), http://www.xbrl.org/Specification/XBRL RECOMMENDATION-2003-12-31+Corrected-Errata-2005-11-07.htm.
6. XBRL Formula Requirements, Candidate Recommendation (2005), http://www.xbrl.org/technical/requirements/Formula-Req-CR-2005-06-21.htm.
7. Daas, P. (2005), *XBRL programma's vergeleken.* CBS-nota, May 17, Methods and Informatics Department, Statistics Netherlands, Heerlen.
8. Financial Reporting Taxonomy Architecture 1.0, Recommendations (2005), http://www.xbrl.org/technical/guidance/FRTA-RECOMMENDATION-2005-04-25.htm.
9. Daas, P. (2005), *Project Initiatie Document XBRL 2005*, CBS prince2 document, version 2.0.p2 1.1, Sept. 6, Methods and Informatics Department, Statistics Netherlands, Heerlen.
10. Daas, P., Stroom, A. (2005), *Productbeschrijving A2*, CBS prince2 document, version 1.0.p2 1.1, Nov. 10, Methods and Informatics Department, Statistics Netherlands, Heerlen.
11. Daas, P., Stroom, A. (2005), *Productbeschrijving A3*, CBS prince2 document, version 1.0.p2 1.1, Nov. 17, Methods and Informatics Department, Statistics Netherlands, Heerlen.
12. Daas, P., Stroom, A. (2005), *Productbeschrijving A5*, CBS prince2 document, version 1.0, Dec. 1, Methods and Informatics Department, Statistics Netherlands, Heerlen.
13. International Financial Reporting Standards, General Purpose Financial Reporting for Profit-Oriented Entities, Incorporating Additional Requirements for Banks and Similar Financial Institutions, Exposure Draft (2005), http://xbrl.iasb.org/int/fr/ifrs/gp/2005-01-15/ifrs-gp-2005-01-15.xsd.
14. Daas, P., Stroom, A. (2005), *Project Faseplan 2*, CBS prince2 document, version 0.1, Dec. 6, Methods and Informatics Department, Statistics Netherlands, Heerlen.
15. Holland, L. (2004), *XML flattened: the lessons to be learned from XBRL.* XML Conference & Exposition 2004, Washington D.C., USA, Nov. 15-19.
16. Financial Reporting Instance Standards 1.0, Public Working Draft (2004), http://www.xbrl.org/technical/guidance/FRIS-PWD-2004-11-14.htm.
17. FRIS 1.0 Conformance Suite, Public Working Draft (2004), http://www.xbrl.org/technical/guidance/FRIS-CONF-PWD-2004-11-14.htm.
18. XML Schema Part 1: Structures Second Edition, W3C Recommendation (2004), http://www.w3.org/TR/xmlschema-1/.
19. Eman, I. (2002), Plausibiliteitsindex PS2000, Methods and Informatics Department, Statistics Netherlands, Voorburg.
20. Dimensional Taxonomy Requirements, Public Working Draft (2005), http://www.xbrl.org/technical/requirements/DIM-REQ-PWD-2005-06-21.htm.
21. XML Schema Part 0: Primer Second Edition, W3C Recommendation (2004), http://www.w3.org/TR/xmlschema-0/.