

Multiple Researchers working at the Blaise Benchmark Services for the Disabled Act (WVG)

Carlo Vreugde & Mark Gremmen (SGBO, The Netherlands)

0. Introduction

Setting up a benchmark is a large operation and involves a complex procedure, involving many researchers and specialists. The definition of benchmark is: ‘A standard by which something can be measured or judged’ and ‘often bench mark A surveyor's mark made on a stationary object of previously determined position and elevation and used as a reference point in tidal observations and surveys.’, American Heritage Dictionary (2004). Benchmarks started in the computer industry.

The Fortran Whetstone programs were the first general purpose benchmarks that set industry standards of computer system performance. Whetstone programs also addressed the question of the efficiency of different programming languages, an important issue not covered by more contemporary standard benchmarks. Results are provided for computers produced during the 1960's to present day systems, including via different languages. (2005)

Benchmarks these days are used for a wide variety of topics. SGBO uses the benchmark for a variety of Municipalities policymakers each within their own special service.

The Services for the Disabled Act Benchmark of the Netherlands is a project for the Dutch Municipalities to learn and improve their method of applying Services for the Disabled programs. The Services for the Disabled Act Benchmark collects data from the Dutch Municipalities using the Blaise program to form a unique central database. The big challenge in 2004 and 2005 is that this National project has expended itself from a Monitor function to a combination of a Monitor and Benchmark functionality with multiple researchers.

The question was how could multiple researchers work together in one project with a combination of a Monitor and a benchmark that comes into one Blaise database?

The approach we took at SGBO was using the application that was most common knowledge among all the old and new researchers and easily to use. This was done by using and refining our Excel Survey Generator.

It is the development and implementation of this Generator that wil make the use of Blaise more manageable with two projects combining and involving multiple researchers. This paper addresses creating a Blaise Survey and Survey-results with multiple researchers through the use of a spreadsheet..

A Generator of Survey data collection formation can reward you in four key ways: (1) It simplifies the work for the researchers, because they have an easy overview of the whole survey, (2) Adjusting survey questions can be done by any researcher without knowing Blaise Syntax, (3) A Generator improves the quality of Blaise Syntax to perfect, because it is all automatically generated after foolproof testing, and (4) it considerably reduced time necessary to write Blaise Syntax code, and thus enhanced the efficiency and total costs of the benchmark.

1. Multiple researchers creating a Blaise Survey

1.1. Background

The combination of a Monitor and a Benchmark started with The Services for the Disabled Act. Data has been collected through paper surveys since 10 years. The data was entered in our own SGBO data entry program. Since SGBO has been using Blaise all the data entry procedures were rewritten in Blaise. The paper survey contains many numeric data and therefore all the paper surveys have to be entered twice in the data entry procedure. Blaise uses the first entry of a survey as base and the second data entry as a check on the base. In the end a final database in *.dbf format was produced and analysed in SAS. The project leader of The Services for the Disabled Act Monitor wanted to make a Benchmark and also a combination of the Monitor and Benchmark in 1 Blaise data entry program. Collecting all this data from the Dutch Municipalities using the Blaise program where multiple researchers could give their input on became a major challenge.

The first obstacle that arose were the frequent changes in the questions and answer possibilities. Monitor questions were different from the benchmark questions. Also the team of researchers of the benchmark changed their questions a lot more. At this point Excel was introduced and was setup in such a way that all the researchers could access this spreadsheet and change all their questions and answer possibilities.

The spreadsheet would generate all the syntax for the Blaise survey so whenever the metadata was changed new Blaise syntax could be generated automatically for the new survey.

1.2. Spreadsheet

In the year 2000 SGBO started with a complicated spreadsheet generator for creating Blaise Syntax. Everything was placed in spreadsheet cells and there were no macros or Visual Basic anywhere. Excel formulas generated the Blaise syntax roughly and little mistakes would produce errors in the Blaise generated syntax. Therefore a new approach was started with the use of Excel macros and Visual Basic and it created a stable and strong spreadsheet that produced accurate results even when mistakes were made in the input area.

The spreadsheet contains four major tab sheets. The first one contains an index of all the tab sheets plus the macro buttons to generate the desired Blaise- and SPSS syntax. The second tab sheet contains all the questions from the survey with all the answer possibilities. The third tab sheet consists of the tables setup for Blaise and the final fourth tab sheet is the input for SPSS formulas to analyse all the data into indicators for the benchmark.

Figure 1.2.1. shows the question 1.2 from The Services for the Disabled Act Monitor. In the spreadsheet the researcher can enter all the question information in a variety of different columns. The major strength of this spreadsheet is that it can be accessed by all the researchers who participate in this Monitor and Benchmark survey project. Chapters 1 through 6 are for the team of researchers of the Monitor project and chapters 7 through 14 are for the team of researchers of the Benchmark project. Each team can insert and delete questions and adapt their properties. The spreadsheet is placed on the network therefore it can be reached throughout the office and even when researchers are working at home through a secured VPN (Virtual Private Network) network connection. The spreadsheet will show a message of accessibility when one of the researchers is actively working in the spreadsheet by making changes in the metadata. Therefore no more than one person can change metadata at the same time. Simultaneously working in the

spreadsheet is out of the question but multiple researchers working together one after another is the preferred way. Version numbers in the spreadsheets name shown are required through time and show the progress in the survey. Also it is an easy way of making a backup of the data and go back steps to check on the changes that were made.

Figure 1.2.1. The spreadsheet question tab sheet for The Services for the Disabled Act monitor & benchmark

| | A | B | C | D | E | F | G |
|----|------------------------|-----------|-------|-------------|-------------|--------------------|--|
| 1 | B | | | | | | |
| 2 | | Hoofdstuk | vraag | vraagnummer | Blaise-SPSS | korte omschrijving | vragen |
| 3 | Boeking kosten | 0 | 1 | 0.1 | v0001 | Scanvraag A0. | Scanvraag A0. Kunt u hieronder aangeven hoe u de kosten van v |
| 4 | Kosten van individuele | 1 | 1 | 1.1 | v0101 | Scanvraag A1. | Scanvraag A1. Gemeentelijke bijdragen in de kosten van woonvoc |
| 5 | | 1 | 2 | 1.2 | v0102 | Scanvraag A2. | Scanvraag A2. Kunt u de gemeentelijke bijdragen in de totale kos |
| 6 | | 1 | 3 | 1.3 | v0103 | Scanvraag A2.0.1 | Scanvraag A2.0.1. Betreft het hier een schatting? |
| 7 | | 1 | 4 | 1.4 | v0104 | Scanvraag A2.1 | Scanvraag A2.1. Kunt u aangeven welke bijdragen in de kosten v |
| 8 | | 1 | 5 | 1.5 | v0105 | Scanvraag A2.2 | Scanvraag A2.2. Deze vraag is vervallen met ingang van Wvg-sca |
| 9 | | 1 | 6 | 1.6 | v0106 | Scanvraag A3. | Scanvraag A3. Deze vraag is vervallen met ingang van Wvg-sca |
| 10 | | 1 | 7 | 1.7 | v0107 | Scanvraag A4. | Scanvraag A4. Overige gemeentelijke bijdragen in kosten van woc |
| 11 | | 1 | 8 | 1.8 | v0108 | Scanvraag A5. | Scanvraag A5. Kunt u de totale bijdragen in kosten van woonwoo |
| 12 | | 1 | 9 | 1.9 | v0109 | Scanvraag A6. | Scanvraag A6. Wat waren in 2004 de uitgaven voor individuele |
| 13 | | 1 | 10 | 1.10 | v0110 | Scanvraag A7. | Scanvraag A7. Deze vraag is vervallen vanaf Wvg-scan 1995 |
| 14 | | 1 | 11 | 1.11 | v0111 | Scanvraag A8. | Scanvraag A8. Kunt u de totale uitgaven voor individuele vervoer |
| 15 | | 1 | 12 | 1.12 | v0112 | Scanvraag A9. | Scanvraag A9. Wat waren de uitgaven op kasbasis in 2004 voor r |

The spreadsheet contains a total of 18 columns and these are shown in Table 1.2.1. This is the metadata that is necessary for generating The Services for the Disabled Act Benchmark Blaise syntax. There are 14 chapters that are shown in the Blaise Survey as tab sheets. The variable name, which is used in Blaise as in SPSS, is a combination of the chapter and the question number. This is followed by a short description of the question so that multiple researchers can see in one instant what the question stands for. This helped on numerous occasions because multiple researchers working on this project could sometimes enter the same question without knowing that the other researcher did this already. Mostly this was done without looking through the survey and the short description helped perfectly because this was always printed before the meeting with all the benchmark participants.

The next columns: question text and on the information is right below the question in the benchmark. This information is about the definition of the question, it usually involves a part of the law of the government. After that comes information about the law that would apply to the local government logistic administration process, for example: an administration code. The final part of information would be the source, for example: name of the law.

The final part is answering of the question; answer categories and type of question. A benchmark produces more types in metadata than in a standard survey. Calculation is done to check on answered data but also indicators variables are put in place. Questions in the monitor that were used in previous years are placed as an auxiliary variable because they are not needed in this years survey but are shown as discontinued questions. Also questions that are not applicable anymore can be disabled.

Table 1.2.1. The types of data that can be entered in the columns of the spreadsheet question tab sheet.

| # | Column | Code | Type | # | Column | Code | Type |
|-----|---------------------|------|---------------|-----|---------------------|------|----------------|
| 1. | Chapter text | | Text | 12. | Question type | Y= | Selection |
| 2. | Chapter number | | Number | | | Ym= | Multi response |
| 3. | Question | | Number | | | O= | Open |
| 4. | Question number | | Number | | | R= | Calculate |
| 5. | Blaise-SPSS | | Variable name | | | K= | Indicator |
| 6. | Short Description | | Text | | | B= | Save (keep) |
| 7. | Question text | | Text | | | U= | Off |
| 8. | Question definition | | Text | | | A= | Aux field |
| 9. | Quest. information | | Text | 13. | Column | | Number |
| 10. | Question source | | Text | 14. | Row | | Number |
| 11. | Answer categories | _1= | "No" | 15. | Blaise-SPSS year | | Variable name |
| | | _2= | "Yes" | 16. | Extra information | | Text |
| 12. | Question type | T= | Table | 17. | Extra information | | Variable name |
| | | N= | Numeric | 18. | Question older year | | Variable name |

Figure 1.2.2. contains the data entry screen for the Monitor paper surveys that are entered twice because of all the numeric data. As you can see the question 1.2 is a Blaise table function. I will take this question as an example of how the spreadsheet is build up.

Figure 1.2.2. A survey for The Services for the Disabled Act monitor & benchmark Blaise data entry for paper

In the spreadsheet the question is placed on the tab sheet questions and has the question type code of T. This means that the answer definition is on the tab sheet tables. Table 1.2.2. shows how the researcher can place the outline of a table in the spreadsheet. The table function starts with a table number, T1.2 and the first row shows the column text which consist of an integer starting with a 0 and ending with a 999999999. There is no row number at the second row and therefore this is also column text with the same type. When the row number starts in the third row Visual Basic knows this is text for the row that belongs to the row number.

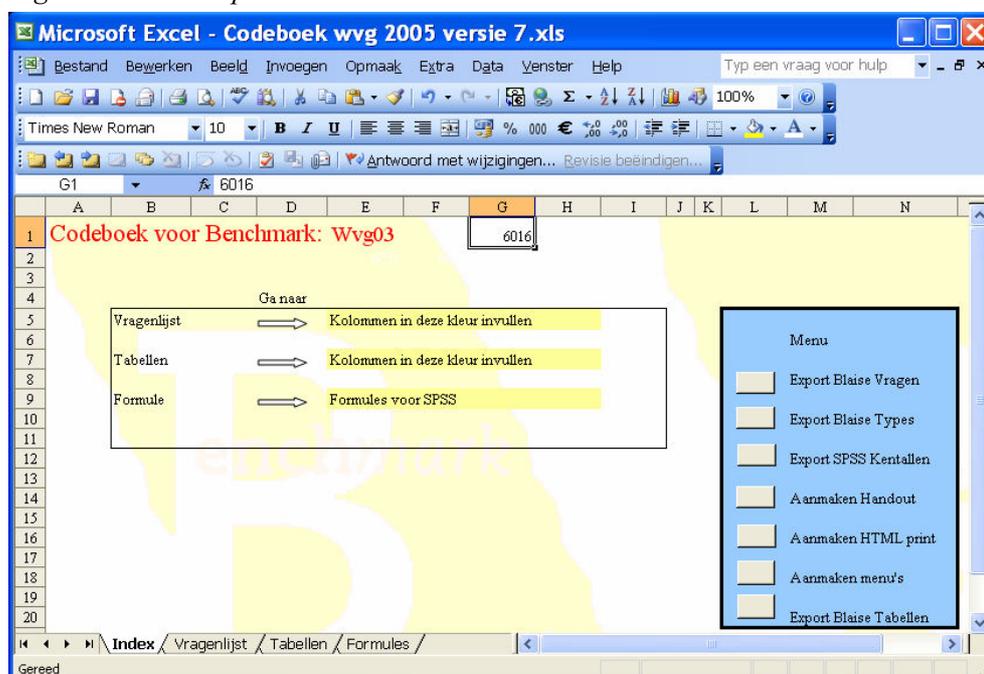
Table 1.2.2. The structure for the spreadsheet table tab sheet.

| Table row number | Table row text, (row text without row numbers is column text!) | Type | Closed, open, numeric, etc. |
|------------------|--|------|-----------------------------|
| T1.2 | Verplichtingenbasis kasbasis | int | 0..999999999 |
| 1 | subsidiabele kosten tot EUR 900,- | | |
| 2 | subsidiabele kosten van EUR 900,- tot 6.800,- | | |
| 3 | subsidiabele kosten van EUR 6.800 tot 20.400,- | | |
| 4 | subsidiabele kosten vanaf EUR 20.400,- | | |

1.3. Generating Blaise Survey Syntax

Once all the survey text is put in the spreadsheet the researcher / analyst and Blaise programmer uses the index tab sheet shown in Figure 1.3.1. He or she can use the white arrows to go directly to the tab sheets that of the questions, tables and formulas.

Figure 1.3.1. The spreadsheet index tab sheet.



On the right side there is the menu that gives the researcher / analyst and Blaise programmer seven options for generating special tasks from all the metadata in the spreadsheet.

Visual Basic is activated through these options. The Visual Basic code looks at specific locations in the spreadsheet at the specified tab sheet and generates syntax for Blaise or other programmes. A small part of the Visual Basic text for the Table function is shown in Syntax 1.3.1.

Syntax 1.3.1. Part of the Visual Basic Table code

...

```
'tabel beginnen – START A TABLE
Print #1, "TABLE " & VraagTabelNrT
'tabel maken – MAKE A TABLE
For r = 1 To rijnr
    pos = Positie + (kolnr) + (r - 1)
```

'nagaan of rij-kop totaal is – CHECK IF ROW HEADER IS A TOTAL

```

Rijkop = sh1.Cells(pos, 2)
rijnrTL = "geen"
If Rijkop = "Totaal" Then
rijnrTL = "TL"
Else
rijnrTL = "geen"
End If

```

'block openen – OPEN A BLOCK

```

blockNm = BlockR(VraagTabelNrB, r)
Print #1, "    BLOCK " & blockNm
Print #1, "    FIELDS"
sh1.Activate
Columns(1).Select

```

'Tabel zoeken in blad tabellen – SEARCH TABLE IN TAB SHEET

```

Selection.Find(What:=Tabelnr, After:=ActiveCell, LookIn:=xlValues, LookAt _
:=xlWhole, SearchOrder:=xlByColumns, searchDirection:=xlNext). Activate
Positie = ActiveCell.Row
pos = Positie
For k = 1 To kolnr
pos = Positie
pos = pos + (k - 1)
Kolkop = sh1.Cells(pos, 2)
...

```

The above Visual Basic syntax generates Blaise syntax for question 1.2 and the generated table syntax which is put in a separate include file, table.inc. The generated Blaise syntax is shown in Syntax 1.3.2.

*Syntax 1.3.2. Generated Blaise Syntax for the table.inc file***TABLE T0102**

```

BLOCK B0102R01
  FIELDS
    T0102011 " " / " verplichtingenbasis":0..999999999
    T0102012 " " / " kasbasis":0..999999999
  RULES
    T0102011
    T0102012
ENDBLOCK {B0102R01}
BLOCK B0102R02
  FIELDS
    T0102021 " " / " verplichtingenbasis":0..999999999
    T0102022 " " / " kasbasis":0..999999999
  RULES
    T0102021
    T0102022
ENDBLOCK {B0102R02}
BLOCK B0102R03
  FIELDS
    T0102031 " " / " verplichtingenbasis":0..999999999
    T0102032 " " / " kasbasis":0..999999999
  RULES
    T0102031
    T0102032
ENDBLOCK {B0102R03}
BLOCK B0102R04
  FIELDS

```

```

T0102041" " / " verplichtingenbasis":0..999999999
T0102042" " / " kasbasis":0..999999999
RULES
T0102041
T0102042
ENDBLOCK {B0102R04}
FIELDS
TB0102R01" " / " subsidiabele kosten tot EUR 900,-":B0102R01
TB0102R02" " / " subsidiabele kosten van EUR 900,- tot 6.800,-":B0102R02
TB0102R03" " / " subsidiabele kosten van EUR 6.800 tot 20.400,-":B0102R03
TB0102R04" " / " subsidiabele kosten vanaf EUR 20.400,-":B0102R04
RULES
TB0102R01
TB0102R02
TB0102R03
TB0102R04
ENDTABLE {v0102}

```

The same method is applied for the question text. If the table function is very complex the spreadsheet is not sufficient anymore. In that case we make use of our Blaise Survey Generator (2003). This generator can be used to make complex surveys with basic routing throughout the survey. It is generally used for quick and small surveys. It is made in Blaise and asks the researcher simple questions from which detailed questions are asked and only answers are needed to make a survey. The Blaise Generator uses Blaise Manipula to generate the Blaise final survey syntax.

2. Multiple researchers creating Survey Results

2.1. Background

The spreadsheet became a very useful tool for the researchers and therefore we thought lets expand the spreadsheet with more than just survey questions. The Services for the Disabled Act Monitor consisted of questions but the Benchmark also involved formulas for indicators. The researchers all had a good idea what the formulas would be made of but had no way of sharing this in one system with one another. Therefore we added the formula tab sheet in the spreadsheet and this was a easy platform for exchanging views, data and make improvements. At this point all the meta data was combined in one spreadsheet.

2.2. Spreadsheet

Indicators are the main event of the benchmark. These figures are used for comparison and have to be visible and easy accessible to every benchmark researcher. In figure 2.2.1. is shown the following information that is visible in the formula tab sheet. The formula tab sheet consists of a chapter number, indicator text, special indicator conditions, the indicator variable name and the formula for the indicator. For internal use there is a formula information text and the old formula from previous years. All of this is used for the calculation of the indicators and for information towards the benchmark participants.

These formulas have to be tested thoroughly and this is a time consuming process. Especially with tables that consist of many variables, figures have to be checked more times and the calculations have to be double checked. Participants like to know the story behind the results of the indicators and they will also check there input with the results of the benchmark. Trust in the benchmark indicators is what makes the benchmark successful. It does not feel comfortable for a municipality if

they score low in the benchmark. Therefore they will first check their input and after that they will check the benchmark indicator. Once those areas are covered the municipality will work on their internal policy to make a better score on the next benchmark. This does not imply that their policy does not work but that they have to adapt by learning from the other municipalities. That is the positive course of the SGB0 benchmark, learn from one another. The moment that one municipality learns from an other municipality is a success factor and therefore the indicators have to be absolutely correct.

2.3. Generated SPSS Syntax

Once all the formulas are entered in the spreadsheet the researcher / analyst and Blaise programmer uses the index tab sheet shown in Figure 1.3.1. On the right side there is the menu that gives the researcher / analyst and Blaise programmer seven options for generating special tasks from all the metadata in the spreadsheet.

Figure 2.2.1. The spreadsheet formulas tab sheet for the Disabled Act monitor & benchmark

| | A | B | C | D | E |
|----|-----------------|---|----|-----------|---|
| 1 | B | | | | |
| 2 | Hoofdstuk Vraag | | IF | Variabele | Formules |
| 3 | | | | GEMNR | |
| 4 | 2 | 2 ALGEMEEN | | VH20 | |
| 5 | 2 | 2.1 Aantallen | | VH21 | |
| 6 | 2 | Totaal aantal cliënten | | T0503011 | |
| 7 | 2 | Totaal aantal cliënten per 1.000 inwoners | | C201 | T0503011/(inw2/1000) |
| 8 | 2 | Totaal aantal toegekende voorzieningen per 1.000 inwoners | | C202 | (T0409TL2+T0411012 + T0413012 +T0401TL1 + T041 |
| 9 | 2 | Aantal cliënten met nieuwe voorziening per 1.000 inwoners | | C203 | (T0506011 + T0506021) / (inw2/1000) |
| 10 | 2 | Aantal cliënten ouder dan 65 jaar per 1.000 ouderen | | C204 | T0503021 / (a3n65eo / 1000) |
| 11 | 2 | 2.2 Uitgaven | | VH22 | |
| 12 | 2 | Totale netto uitgaven | | C205 | (T0101TL2 + T0109091 + T0112061 + v0201 + T0202C |
| 13 | 2 | Totale netto uitgaven per inwoner | | C206 | ((T0101TL2 + T0109091 + T0112061 + v0201 + T0202C |
| 14 | 2 | Percentage netto uitgaven van de inkomsten | | C207 | ((T0101TL2 + T0109091 + T0112061 + v0201 + T0202C |
| 15 | 2 | Begrote uitgaven per inwoner in 2002 | | C208 | T0208TL1/inw2 |
| 16 | 2 | Begrote uitgaven per inwoner in 2003 | | C209 | T0208TL2/inw2 |
| 17 | 2 | Verwachte ontwikkeling in 2003 (in %) | | C210 | (T0208TL2/T0208TL1) |
| 18 | 2 | 2.3 Clientgerichtheid | | VH23 | |

Visual Basic is activated through these options. The Visual Basic code looks at specific locations in the spreadsheet at the specified tab sheets and generates syntax for SPSS or other programmes. A small part of the Visual Basic text for the formulas is shown in Syntax 2.3.1.

Syntax 2.3.1. Part of the Visual Basic SPSS code

```

...
Print #1, "GET FILE =" & projectnr & ".sav'."
Print #1,
Print #1, "/* Formules kentallen COMPUTE statements*/"
Print #1,
i = 3
For i = 3 To aantalvrg
  'Opschonen data
  sh.Cells(i, 1) = Trim(sh.Cells(i, 1))
  sh.Cells(i, 2) = Trim(sh.Cells(i, 2))
  sh.Cells(i, 3) = Trim(sh.Cells(i, 3))
  sh.Cells(i, 4) = Trim(sh.Cells(i, 4))

```

```

sh.Cells(i, 5) = Trim(sh.Cells(i, 5))
sh.Cells(i, 6) = Trim(sh.Cells(i, 6))
If Len(sh.Cells(i, 5)) <> 0 Then
  Print #1, "COMPUTE " & Trim(sh.Cells(i, 4)) & " = " & Trim(sh.Cells(i, 5))
  & "."
End If
Next i
Print #1, "EXECUTE."
Print #1,
Print #1, "/* Formules kentallen IF statements*/"
Print #1,
i = 3
For i = 3 To aantalvrg
  If UCase(Left(sh.Cells(i, 3), 2)) = "IF" Then
    Print #1, sh.Cells(i, 3) & " " & sh.Cells(i, 4) & " = " & sh.Cells(i, 5) & "."
  End If
Next i
Print #1, "EXECUTE."
...

```

The above Visual Basic syntax generates SPSS syntax for the formulas of the indicators and the label syntax which is put in a separate SPSS file, project number Export Excell.sps. The generated SPSS syntax is shown in Syntax 2.3.2.

Syntax 2.3.2. Generated SPSS Syntax for the Export Excell.sps file

```

/*****
/* BESTANDSNAAM: 091 Export Excell.sps      */
/* DATUM: 26/02/2006                       */
*****/

GET FILE ='6016.sav'.

/* Formules kentallen COMPUTE statments*/

COMPUTE C201 = T0503011/(inw2/1000).
COMPUTE C202 = (T0409TL2+T0411012 + T0413012 +T0401TL1 + T0413021)
/ (inw2/1000).
COMPUTE C203 = (T0506011 + T0506021) / (inw2/1000).
COMPUTE C204 = T0503021 / (a9n65eo / 1000).
COMPUTE C205 = (T0101TL2 + T0109091 + T0112061 + v0201 + T0202051 +
v0801 + t0203021+ v0205 + v0206 + v0207) - (v0301 - v0302 + T0303011 +
T0303021 + T0303031).
COMPUTE C206 = ((T0101TL2 + T0109091 + T0112061 + v0201 + T0202051 +
v0801 + t0203021 + v0205 + v0206 + v0207) - (v0301 - v0302 + T0303011 +
T0303021 + T0303031)) / inw2.
COMPUTE C207 = ((T0101TL2 + T0109091 + T0112061 + v0201 + T0202051 +
v0801 + t0203021 + v0205 + v0206 + v0207) - (v0301 - v0302 + T0303011 +
T0303021 + T0303031)) / ((FB02NW/2.20371)+(AWBZ02/2.20371)).
COMPUTE C208 = T0208TL1/inw2.
COMPUTE C209 = T0208TL2/inw2.
COMPUTE C210 = (T0208TL2/T0208TL1).
...
VARIABLE LABELS
  GEMNR      "
  VH20      '2 ALGEMEEN'
  VH21      '2.1 Aantallen'
  T0503011  'Totaal aantal cliënten'

```

| | |
|------|---|
| C201 | 'Totaal aantal cliënten per 1.000 inwoners' |
| C202 | 'Totaal aantal toegekende voorzieningen per 1.000 inwoners' |
| C203 | 'Aantal cliënten met nieuwe voorziening per 1.000 inwoners' |
| C204 | 'Aantal cliënten ouder dan 65 jaar per 1.000 ouderen' |
| VH22 | '2.2 Uitgaven' |
| C205 | 'Totale netto uitgaven' |
| C206 | 'Totale netto uitgaven per inwoner' |
| C207 | 'Percentage netto uitgaven van de inkomsten' |
| C208 | 'Begrote uitgaven per inwoner in 2002' |
| C209 | 'Begrote uitgaven per inwoner in 2003' |
| C210 | 'Verwachte ontwikkeling in 2003 (in %)' |
| VH23 | '2.3 Clientgerichtheid' |

...

2.4. More generated Syntax

Meta data can be used for Blaise and SPSS and other functions. Internet pages can be generated and give a nice overview of all the questions and variables. Also extra SPSS files could be generated. The benchmark participants found it difficult to recalculate the benchmark indicators and told SGB0 this was a time consuming task. SGB0 therefore created a spreadsheet through SPSS syntax with all the data of the municipalities and the data of all the indicators. Included were all the formulas that were necessary for calculating all the indicators. The spreadsheet could then be mailed to the municipality and it gave them all the information for the whole benchmark process. The spreadsheet became a powerful tool for the whole project.

3. Conclusion

Benchmarks for measuring a standard have been in use for quite a while. SGB0 has several benchmarks and SGB0 used a unique approach for The Services for the Disabled Act Benchmark where researchers combined a monitor and a benchmark. Multiple researchers had to work together in forming a unique survey and together produce the benchmark indicators for comparison.

With the use of a spreadsheet it became possible that multiple researchers start working on the same project. This spreadsheet has four tab sheets that consists of the following information. The first tab sheet is the index of the whole spreadsheet and will let you jump to the three other tab sheets. Also it consists of 7 functions that generate a wide variety of syntaxes from Blaise to SPSS all through Visual Basic code.

The second tab sheet is called 'Questions' and has all the questions of The Services for the Disabled Act Benchmark monitor and benchmark together. With 18 different columns definition it has all the meta data needed to generate a wide variety of syntaxes. The third tab sheet supports the second tab sheet by detailed information of the table questions. All column and row text is entered here with the type of answer possibilities.

The final tab sheet is the fourth one that has all the formulas for the benchmark indicators that show the benchmark results and let the municipalities compare one to one another. Setting up this benchmark was a large operation involved many researchers and specialists which was made possible with the help of a spreadsheet with Visual Basic code that could be changed at any moment and it generated in a second Blaise and SPSS error free syntax.

4. References

The American Heritage® Dictionary of the English Language, Fourth Edition (2004). Published by Houghton Mifflin Company, Wilmington.

Longbottom, Roy, 2004: Whetstone Benchmark History and Results, <http://homepage.virgin.net/roy.longbottom/whetstone.htm>, (TSU/CCTA 1960 to 1993), updated November (2005).

Vreugde, C.J.C. & De Groot, J.M.J & Van 't Hoft, C. (2003): Blaise Survey Generator, Any researcher can make a survey 'without writing Blaise code', Presentation at 8th International Blaise Users Conference, May 21-23, Copenhagen, pp. 1-10

Bethlehem, J.G. (1999): Cross-sectional Research. In: H.J. Adèr and G.J. Mellenbergh, Research Methodology in the Social, Behavioural & Life Science, Sage Publications, London, pp.110-142.