Choosing a language and tool for the Business and Information Architecture of Statistics Netherlands

A.R. Griffioen and F. Hofman

The views expressed in this paper are those of the author(s)
and do not necessarily reflect the policies of Statistics Netherlands

Discussion paper (201202)

Statistics Netherlands

The Hague/Heerlen, 2012

Explanation of symbols

.	data not available
*****	provisional figure
******	revised provisional figure (but not definite)
x	publication prohibited (confidential figure)
–	nil
–	(between two figures) inclusive
0 (0.0)	less than half of unit concerned
empty cell	not applicable
2011–2012	2011 to 2012 inclusive
2011/2012	average for 2011 up to and including 2012
2011/’12	crop year, financial year, school year etc. beginning in 2011 and ending in 2012
2009/’10– **2011/’12**	crop year, financial year, etc. 2009/’10 to 2011/’12 inclusive

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

Publisher
Statistics Netherlands
Henri Faasdreef 312
2492 JP The Hague

Prepress
Statistics Netherlands
Grafimedia

Cover
Teldesign, Rotterdam

Information
Telephone +31 88 570 70 70
Telefax +31 70 337 59 94
Via contact form:
www.cbs.nl/information

Where to order
E-mail: verkoop@cbs.nl
Telefax +31 45 570 62 68

Internet
www.cbs.nl

ISSN: 1572-0314

60083201202 X-10

Choosing a language and tool for the Business and Information Architecture of Statistics Netherlands

A.R. Griffioen and F. Hofman

Summary: This is a report for choosing a language and tool for the business and information architecture (BI-architecture) of Statistics Netherlands. This is a first step to improve the current way of BI-architecture development and maintenance. In the first phase of the project the business architects indicated the need to change the current language and tool for others. In the second phase of the project the business architects together with the IT Enterprise architects advised that Archimate and BizzDesign Architect are the best language and tool combination to support the new way of working.

Keywords: Business and Information architecture, architecture language, architecture tool

1. Introduction

1.1 Background and purpose

This is a report for choosing a language and tool for the business and information architecture (BI-architecture) of Statistics Netherlands. The BI-architecture is developed by the Department of Process development and Quality. The main reasons for making this report are to document how we came to the final decision and to inform other (business) architects of governmental institutions, in particular other National Statistical Institutes (NSI's).

Until now the BI-architecture has been developed within the IAF framework with Word and Visio as tools for development. The BI-architecture itself has been spread by email and Sharepoint. As there was no single repository or a main document, this way of working resulted in different, even sometimes contradicting, versions of a dozen of architecture documents. This was an important reason to change the way of working. Other reasons were:

- To make it more consistent and understandable for developers/administrators[1] and users;
- To make its maintenance more controllable;
- To improve the alignment with the IT (IS & TI) Architecture;

[1] Developers and administrators of the BI-architecture are the same persons at Statistic Netherlands.

- To improve user accessibility; For instance, having one version of the architecture and a tool that can visualise or show user-friendly views can contribute to the accessibility.

Thus, there are sufficient reasons to improve the current way of working. To achieve this, the first step we have taken is to choose a possible new language and tool (set) to develop and maintain the BI-architecture. Some of the different language and tool options represent our current way of working, while others represent a more formal or industrial way of working. For instance, one language option was IAF and one toolset choice was Word and Visio. These alternatives represent the current way of working. Hereby the remark that IAF is a framework and not a language. Natural language is then the language actually used.

Of course a new language and tool alone will not change the way of working such that it solves all our current problems. However, some languages and tools will facilitate the desired way of working more than others. For instance, a more professional tool like BizzDesign Architect has one common repository that synchronises the input from different developers. BizzDesign Architect thus lowers the chance of having different architecture versions compared with Word.

There was 100 hours reserved to carry out the project. These were hours only for the project executers, excluding the hours to do the workshops. This amount of hours limited the number of alternative languages and tools that could be investigated and the depth to investigate their scores to the requirements.

We divided the project in two phases. In Section Two, we will describe Phase I and report the results. We will do the same in Section Three for Phase II. Based on these results, of the previous sections, we will present the main conclusions and our advice for language and tool choice. We will, furthermore, discuss future work that has to establish a firmer basis for our final choices.

1.2 Definitions, acronyms and abbreviations

Acronym/Abbreviation	Descriptions
Archimate	Archimate is a language to develop an architecture. It is developed by the Telematica Institute and used by the software application BizzDesign Architect.
BI-architecture	The BI-architecture describes the Business- and Information-systems domain of the IAF-framework.
BizzDesign Architect	BizzDesign Architect is a software application developed by the company BizzDesign. The main purpose of the application is for developing architectures. For more information on the company and application

	see http://www.bizzdesign.com/.
Glossary+	The idea of the Glossary+ is that it will describe the concepts of the IAF-architecture of Statistics Netherlands and their relations.
IAF	IAF is an abbreviation for Integrated Architecture Framework [1]. It is developed by Capgemini. The architecture distinguishes between the Business (B), Information (I), Information systems (IS) and Technology-Infrastructure domain (TI). One of its main goals is to have alignment between business and IT.
IS & TI architecture	The IS & TI architecture describe the Information systems and Technology-Infrastructure domain of the IAF-framework.
Mavim	Mavim is a software application developed by the company Mavim. Its main purpose is to describe business processes, but it can also be used for other purposes. For more information see http://www.mavim.com/.
Sparx Systems Enterprise Architect	Sparx Systems Enterprise Architect is a software application. The main purpose of the application is to facilitate software development projects. For more information see http://www.sparxsystems.eu/.
SVN	SVN stands for subversion. It is a software application with the purpose of a backup and versioning system.

2. Phase I

In Section 2.1, we will explain how we organised Phase I. In Section 2.2, we will present the results.

2.1 Methodological approach and goal

Table 2.1 gives an overview of the project. This is followed by a detailed description of phase I.

Table 2.1. Overview Phase I.

Phase	Result and actions needed to attain results
Phase I	Result: A go/no go decision for going to Phase II of the project. Actions needed: 1. Determine a set of languages and tools from which to choose. a. Consultation of other NSI's by email. b. Consultation of language and tool experts within Statistics Netherlands 2. Estimate the need and opinions of the business architects to use a new language and toolset. a. A workshop among the business architects with the "thermometer" approach.

1a) Consultation of other NSI's

To keep the project small regarding the amount of hours we had for the project, it was decided that we would focus on languages and tools that were used within Statistics Netherlands or other National Statistical Institutes. We approached the following NSI's, since their architectural development is expected to be at the same stage or further than Statistics Netherlands.

This resulted in the following list (Table 2.2) of NSI's:

Table 2.2. List of Contact persons.

NSI	Contact person
Statistics Denmark	Helle Stender
Statistics Finland	Lehto Kari
Statistics Norway	Jenny Linnerud
Statistics Sweden	Hans Irebäck

1b) Consultation within Statitistics Netherlands

To obtain technical information about architectural languages and tools we consulted the following people (Table 2.3).

Table 2.3. List language and tool experts.

Language and tool	Expert(s)
Archimate	Hans Wings and Dick Woensdregt
Other issues concerning the language of an architecture	Max Booleman
BizzDesign Architect	Hans Wings and Dick Woensdregt
Enterprise Architect	Remco Paulussen

Mavim	Fong Yee Wong and Robert Griffioen
Wiki	Tjalling Gelsema and Robert Griffioen

Since the IT department had already made a language and tool choice for the IT-architecture, we used the IT enterprise architects as an important source of information. We, therefore, had an additional review with one of IT Enterprise Architects. Furthermore, the tool document with arguments that support their choices written by the other IT Enterprise Architect [2] was a valuable source of information.

2a) The "thermometer" workshop

To acquire information about the opinions of the languages and tools from the architects, we organised a small workshop of one hour. Participants were mainly business architects from the Department of Process development and Quality. One of the participants was an Enterprise Architect from the IT-department to guarantee business IT alignment.

The workshop counsellor had drawn a thermometer on a sheet of paper, with above a statement. In this case, the statements were "Do we have to use another language?" and "Do we have to use other tools?". Each statement had its own thermometer. The workshop participants had to sticker the thermometer indicating the level to which they agreed on the statement. The thermometer had values low, medium and high, but one should not value the marks too much. It is mainly meant to start up discussion. After the sticker phase, the workshop counsellor discussed the stickers and asked the participant who stickered and why it was placed there.

2.2 Results Phase I

Results of consulting other NSI's (1a)

The results in the table, Table 2.4, below show that all of the consulted NSI's are at the same stage as Statistic Netherlands and do not have a more formal architectural language or use a more professional tool for its development. Hereby, the remark that the person of Statistics Denmark we initially contacted was not working anymore for the institute.

Table 2.4. Language and tool use at NSI's.

Language and tool ***NSI***	Lanuage	Tools
Denmark	x	Unclear what they are using and want to use in the future
Finland	x	Visio and MS Office
Norway	x	MS Office
Sweden	x	Sybase Powerdesigner (mainly procesmodelling, but EA possible); Future: Sparx Enterprise Architect

Results of consulting CBS experts (1b)

The results of the consultation of the CBS experts were integrated in the language and tool matrices of the next section Section 3.2. As mentioned before (Section 2.2), the tool (set) choice was restricted to tools already used in Statistics Netherlands or used in another NSI. Since the IT-department is an important stakeholder with similar issues as ours, their reasons for choosing Archimate and BizzDesign were important to us. The main reason of IT for choosing Archimate above other formal languages as a language was because it is an open standard. The main reason for choosing BizzDesign Architect as a tool was that it was the best trade-off between number of desired features and cost.

Results of the "thermometer" workshop (2a)

The thermometer workshop showed that most of participants of the workshop wanted to use a more formal language than currently used. They were hesitant to use an existing formal language like Archimate, because of a high learning curve that may cost a lot of time. About the use of another toolset, the participants were unanimously: they all wanted another toolset. However, also in this case they did not want a fancy tool with a lot of features, because of the possible time cost of learning it.

Based on the outcome of the thermometer workshop, it had been decided between the project executers and the head of project that the project is allowed to carry on with a language selection and tool selection. It had also been decided that given the number of hours reserved for the project, there was no need to write a report for the first phase.

3. Phase II

In Section 3.1, we will explain how we organised Phase II. In Section 3.2, we will present the results.

3.1 Methodological approach and goal

Table 3.1 gives an overview of the project. This is followed by a detailed description of phase II.

Table 3.1. Overview of Phase II.

Phase	Result and actions needed to attain results
Phase II	Result: Language and toolset advice. Actions Needed: 1. Determine the set of language and tool set alternatives. 2. Determine two sets of requirements: one set to make a language choice and one for making a tool choice. Determining the scores for each alternative against a requirement. Determining the priority of each requirement. 3. Compare the alternatives.

	4. Give an advice about the choices. a. Consultation of language and tool experts within Statistics Netherlands. b. A one day workshop in which the language and tool requirements are discussed and possible extended with new ones.

To select the set of language and tool alternatives, we collected the options and requirements that we obtained from experts of Statistics Netherlands and other NSI's. From the requirements we selected knock out requirements. The effect of the knock out requirements on the tool selection was that in some cases instead of one tool a toolset was needed to satisfy them. The knock out requirements for the toolset demanded that the tool (set) could express the architecture in text format and visualise it. Furthermore, another knock out requirement demanded that it possessed a versioning system. The knock out requirements resulted, for example, in a toolset of Word, Visio and Subversion.

To determine further requirements and score them for the different language and tool options, we collected knowledge from experts of Statistics Netherlands and literature [3]. We mostly needed information for tools to score. We present the results in Table 3.2 and Table 3.3.

Table 3.2. Functional requirements for language selection.

Functional requirements for language selection
1. The language is suitable for the existing business concepts.
2. The language supports consistency checks
3. The language is sufficiently flexible to express the business concepts.
4. The language connects to the IT-Enterprise Architecture.
5. Ease of learning the new language by developers and administrators
6. Language is an open standard
7. Language is used by chain partners
8. Language is useable by other National Statistics Institutes (NSI`s).
9. Effect of the language on the ease of understanding the architecture.
10. Migration cost (in hours)
11. Used with other languages
12. Maintenance cost
13. Possibility of making automatic delta between different versions of the architecture.
14. Level arch. Communication

Table 3.3. Functional requirements for tool selection.

Functional requirements for tool selection
2. The tool has features to show the same information from different perspectives, without having to store them multiple times (views).
3. The ease of the new tool to learn it.
4. The ease of the tool to open up information to the intranet.
5. The possibilities of the tool for users (not developers/administrators) to respond to architecture.
6. The tool has sufficient export and import possibilities to connect to the IT-architecture: BizzDesign Architecture.
7. The tool has sufficient export and import possibilities to connect to the process development in case of redesign: Mavim.
8. Licence cost / administrator cost
9. Migration cost: One time cost to fill the new tool with knowledge
10. Different presentations
11. Impact analyses
12. Roadmaps
13. Open standard
14. Exchangeability
15. Central repository / consistency
16. Ease of use
17. Flexibility

To share the collected information with our fellow business architects and make a collective choice we did chose the workshop format. There were similar participants as in the small workshop: a number of business architects and one enterprise architects of IT.

To present the collected knowledge effectively for the workshop we prepared two matrices: one matrix for the language choice and one matrix for the toolset choice. In each matrix one axis represented the language/toolset options and the other axis represented the requirements[2]. There was one extra column that scored the priority of a requirement.

[2] To get clearer idea of a matrix and its content one can look in Section 3.2 in which the matrices are presented.

We prepared a third matrix that took into account the dependencies between a language and tool (set). To explain this with an example, working with a formal language as Archimate has the advantages of having facilities to work with this language, for instance a facility that checks the language grammar. Before the workshop, we excluded illogical language tool combinations. These were marked with an X in the matrix (see Section 3.2).

The procedure of the workshop was as follows.

1. Explaining alternatives and requirements: the different alternatives and requirements were explained by one of the project executers.
2. Adding additional requirements: the participants could propose new requirements and if accepted they were scored
3. Prioritising the requirements: the participants received a number of stickers, half the number of requirements, to grade the priority of the requirement by sticking them to their favourite requirement(s). There were no limitations: one was allowed to put all stickers to one requirement, but one could also give a requirement only one sticker. The sticker phase was followed by a brief discussion to check the chosen priorities.
4. Scoring the requirements for each alternative: one by one each requirement was scored for each alternative. Especially for the tools the project executers had prepared proposed scores for each alternative. The final scores however, were always determined by the group. To speed up this process we skipped the discussion of requirements with the lowest priority.
5. Discussing overall results: an quick indication of the overall score for each alternative was the weighted sum of each requirement score multiplied by the priority of the requirement. The participants then discussed these overall scores, for instance whether they corresponded to their expectations, and draw some conclusions e.g. the top two languages or top three tools.

We followed the above procedure first for the language matrix. This took all morning. In the afternoon we tackled the tool matrix. Later in the afternoon we had a discussion about the third matrix that integrated the results of the language and tool matrices.

3.2 Results Phase II

The first stage of Phase II resulted in the following list of architecture languages:

- Current language (IAF),
- Glossary+, and
- Archimate.

One of the language alternatives was the Glossary+. We meant by this an explanation or definition for each concept of the BI-architecture, which is the Glossary, and a description of relations between the concepts that is the plus part.

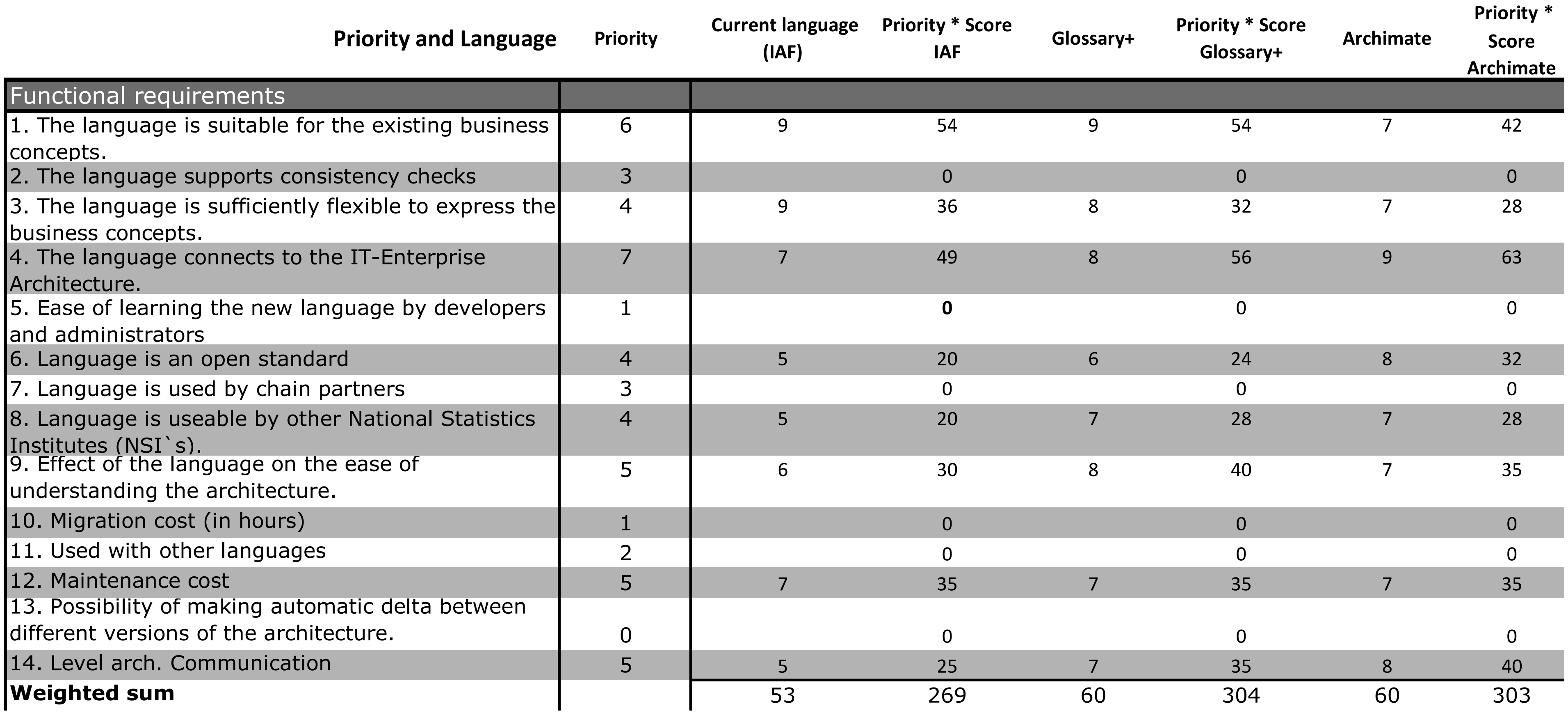
We did not exactly work out the Glossary+, but it was meant to indicate a more formal and strict architecture language than the current one**.**

Furthermore, we had the following list of architecture tool (set)s to choose from:

- BizzDesign Architect,
- Sparx Systems Enterprise Architect,
- Mavim,
- Wiki and Visio, and
- Word, Visio and SVN.

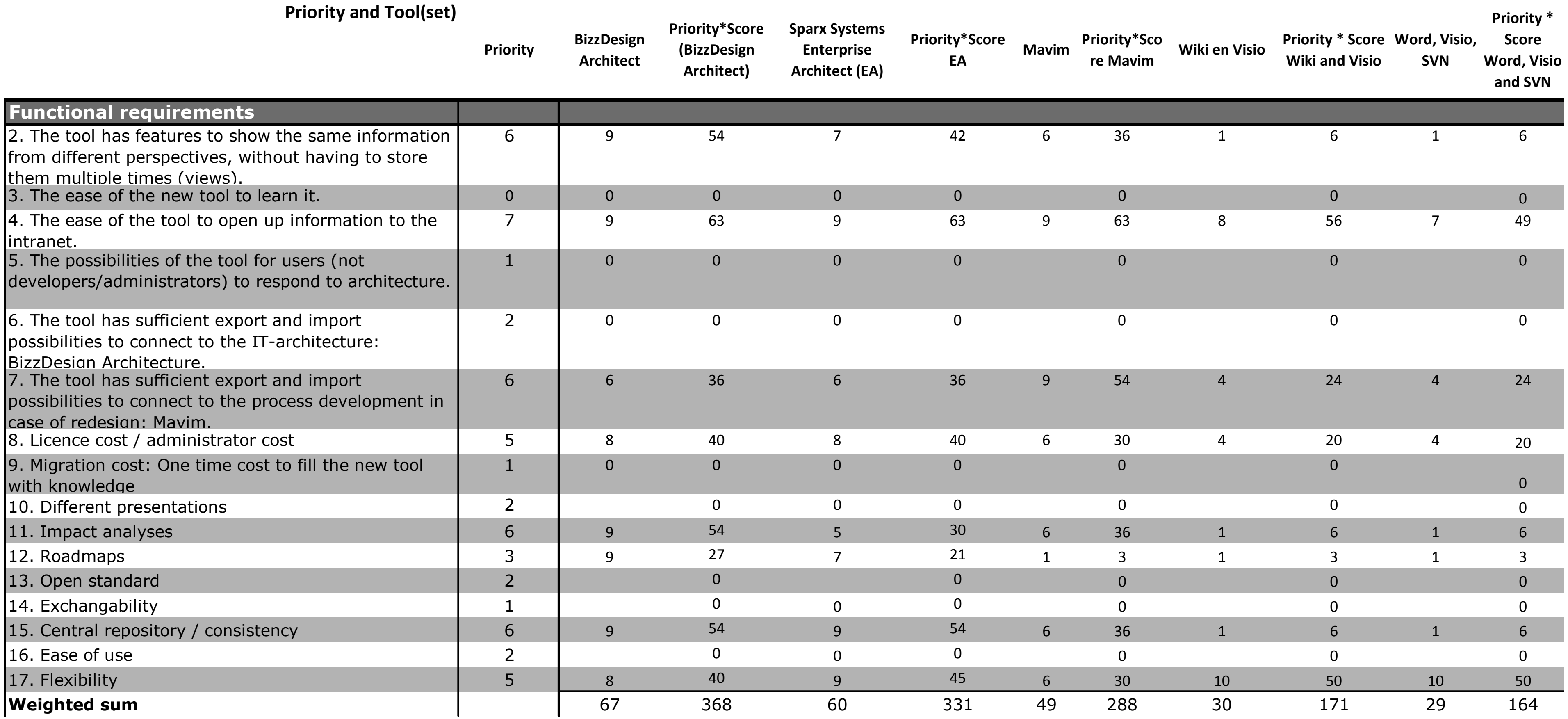
The matrices in Figure 1 and Figure 2 show the results of the comparison between the languages and the tools or toolsets.

Figure 1: Comparing Languages



Priority and Language	Priority	Current language (IAF)	Priority * Score IAF	Glossary+	Priority * Score Glossary+	Archimate	Priority * Score Archimate
Functional requirements							
1. The language is suitable for the existing business concepts.	6	9	54	9	54	7	42
2. The language supports consistency checks	3		0		0		0
3. The language is sufficiently flexible to express the business concepts.	4	9	36	8	32	7	28
4. The language connects to the IT-Enterprise Architecture.	7	7	49	8	56	9	63
5. Ease of learning the new language by developers and administrators	1		**0**		0		0
6. Language is an open standard	4	5	20	6	24	8	32
7. Language is used by chain partners	3		0		0		0
8. Language is useable by other National Statistics Institutes (NSI`s).	4	5	20	7	28	7	28
9. Effect of the language on the ease of understanding the architecture.	5	6	30	8	40	7	35
10. Migration cost (in hours)	1		0		0		0
11. Used with other languages	2		0		0		0
12. Maintenance cost	5	7	35	7	35	7	35
13. Possibility of making automatic delta between different versions of the architecture.	0		0		0		0
14. Level arch. Communication	5	5	25	7	35	8	40
Weighted sum		53	269	60	304	60	303

Figure 2: Comparing Tool (set)s

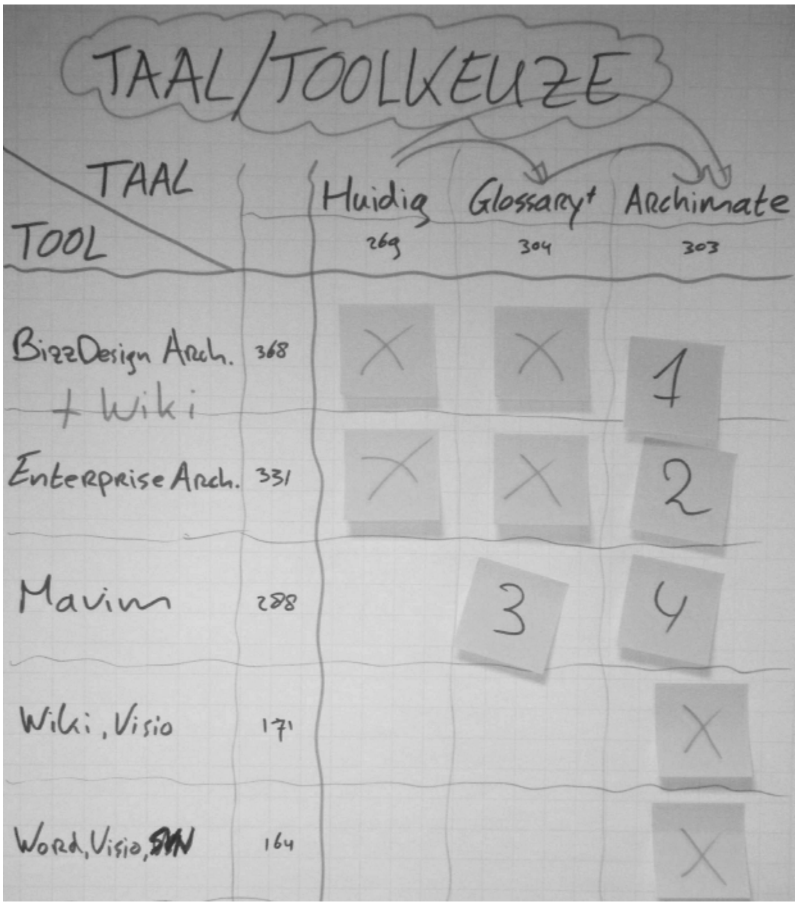


Priority and Tool(set)	Priority	BizzDesign Architect	Priority*Score (BizzDesign Architect)	Sparx Systems Enterprise Architect (EA)	Priority*Score EA	Mavim	Priority*Score Mavim	Wiki en Visio	Priority * Score Wiki and Visio	Word, Visio, SVN	Priority * Score Word, Visio and SVN
Functional requirements											
2. The tool has features to show the same information from different perspectives, without having to store them multiple times (views).	6	9	54	7	42	6	36	1	6	1	6
3. The ease of the new tool to learn it.	0	0	0	0	0		0		0		0
4. The ease of the tool to open up information to the intranet.	7	9	63	9	63	9	63	8	56	7	49
5. The possibilities of the tool for users (not developers/administrators) to respond to architecture.	1	0	0	0	0		0		0		0
6. The tool has sufficient export and import possibilities to connect to the IT-architecture: BizzDesign Architecture.	2	0	0	0	0		0		0		0
7. The tool has sufficient export and import possibilities to connect to the process development in case of redesign: Mavim.	6	6	36	6	36	9	54	4	24	4	24
8. Licence cost / administrator cost	5	8	40	8	40	6	30	4	20	4	20
9. Migration cost: One time cost to fill the new tool with knowledge	1	0	0	0	0		0		0		0
10. Different presentations	2		0	0	0		0		0		0
11. Impact analyses	6	9	54	5	30	6	36	1	6	1	6
12. Roadmaps	3	9	27	7	21	1	3	1	3	1	3
13. Open standard	2		0		0		0		0		0
14. Exchangability	1		0	0	0		0		0		0
15. Central repository / consistency	6	9	54	9	54	6	36	1	6	1	6
16. Ease of use	2		0	0	0		0		0		0
17. Flexibility	5	8	40	9	45	6	30	10	50	10	50
Weighted sum		67	368	60	331	49	288	30	171	29	164

The language results in Figure 1 (the weighted sum below the matrix) showed that there was a probably insignificant difference between the winners Glossary+ and Archimate. The results clearly showed the desire *not* to work with the current language.

The tool matrix in Figure 2 showed that BizzDesign Architect and Sparx Enterprise Architect were the winners with 368 and 328 point. They were followed at some distance by Mavim with 288 point. The old tools or the Wiki version lags more than 200 points behind.

Figure 3: Integrated results



TAAL/TOOLKEUZE

TAAL TOOL		Huidig 269	Glossary+ 304	Archimate 303
BizzDesign Arch. + Wiki	368	X	X	1
Enterprise Arch.	331	X	X	2
Mavim	288		3	4
Wiki, Visio	171			X
Word, Visio, SVN	164			X

The results of the matrix in Figure 3 was a synthesis of the other two matrices. They showed that the preference for BizzDesign Architect and Sparx EA, representing a more formal way of working, shift the preference to the most formal language Archimate. These tools have facilities to work with Archimate. Functionally, BizzDesign Architect was the winner, because it had most integrated facilities with Archimate and met requirements best. However, the IT department is thinking of exchanging BizzDesign Architect for Sparx EA, because the latter is cheaper and used by the project software architects. Moreover, there is a policy at Statistics Netherlands to have as few as possible different types of tools. Another detail that came out of the workshop was that the IT department was using a Wiki for the IT-architecture to accommodate users to comment and give advice on the existing architecture. It had been decided that the Department of Process development and Quality will use the Wiki for the development of the BI-architecture in the same way as the IT-department.

4. Conclusions and future work

The advice from this project is that the Department of Process development will use Archimate as a language for the BI-architecture. Furthermore, it will use the same tool as used by the IT-department for the development of the BI-architecture, which is BizzDesign Archictect or Sparx Enterprise Architect. A Wiki will be used as additional resource for the development of the BI-architecture.

Before the final decision of using Archimate and BizzDesign Architect or Sparx EA, a proof of concept (PoC) will be carried out to verify whether the use of Archimate and tool will not lead to insurmountable problems.

During the writing of this report the IT-department has decided to use BizzDesign Architect as an Enterprise Architect tool [4]. Mainly, because it has the most desired features compared to Sparx EA; That is worth the extra cost. For the Department of Process development and Quality this means that the PoC will be carried out with BizzDesign Architect. If the PoC succeeds, it will then be used as the tool for the development of the BI-architecture.

5. References

[1] Adriaanse, P. (2005). Architecture essentials, Capgemini.

[2] Wings, H. (2008). Modelleertool voor Enterprise Architectuur.

[3] Wilson, C. and Short, J. (2010). Magic Quadrant for Enterprise Architecture tools, Gartner RAS Core Research note G00207406.

[4] Wings, H. (2010).Vergelijking BizzDesign Architect en Enterprise Architect.