Towards a Vendor and Technology Independent Enterprise Asset & Resource Management Indexer for Integrated Infrastructure Projects

An open-source alternative for BIM’s current way of data linking and retrieval

Peter Paul Abraham Wiggers
1363387

Master Thesis
Construction Management and Engineering
Delft University of Technology

February 2015

Graduation Committee
Prof.dr.ir. A.R.M. Wolfert Delft University of Technology (CiTG)
Dr.ir. A.J.H. Hidders Delft University of Technology (EWI)
Dr.ir. H.K.M. van de Ruitenenbek Volker InfraDesign

Volker InfraDesign

TUDelft Delft University of Technology
ABSTRACT

The construction industry is rapidly implementing different types of new digital engineering assets information technologies. All these asset information systems generate lots of data and having these large amounts of data poses new challenges: recombining the data in useful ways to achieve a more efficient working environment. However, all these tools use their own data structures, formats and identifiers, making this recombination a complex and practically almost impossible task.

A solution for this problem, which is recently gaining popularity, is BIM: the Building Information Model. This model represents a 3D model of the construction to be built, which allows to connect data from various sources. Therefore, BIM prescribes specific standards, such as an object-based approach and the use of specific file formats and structures.

However, since all applications are designed for a specific purpose, they not all function object-based (e.g. Maximo, which works with linear assets). Enforcing an object-based approach to be able to work with BIM results in losing the power of an application, which is unfavourable. In practice it appeared to be very difficult to impose the users with this required level of standardizations. People and software lost their strength because they had to comply with all the standards BIM imposed. The fact that building consortia – who often execute these large infrastructure projects – consist of multiple companies makes this task even more difficult, since they all use their internal BIM standards and naming conventions.

There are plans to continue with BIM nD but that is expected to result in even more complications. Research approaches acknowledge this problem and have investigated the possibilities of methods to achieve the same results in a different way; the way of requiring less standardizations and connecting data on multiple identifying aspects. This emphasizes the need to continue a different way.

The research of Pauwels, De Meyer, & Van Campenhout (2011) uses a technology used from the world wide web: the semantic web technology. This technology enables machines to easily understand data and its relationships, enabling the user to ask difficult questions. However, this method has several requirements, such as the use of specific formats, that make it hard to implement.

This research deems practical (real world) functionality as more important than theoretical (ideal world) functionalities. Therefore it considers a solution that is more user-oriented and more flexible. Tech-
Technologies that allow such flexibility are JSON\textsuperscript{2} and NoSQL\textsuperscript{3}(CouchDB). A combination of these – and other – technologies might result in a system that is good at assets data retrieving and linking, while not imposing much standards to the user – allowing him to use methods, formats and applications he prefers.

To check if this solution can eventually be implemented, a theoretical set-up has been designed after an extensive analysis of the current situation and (future) requirements. A prototype of this set-up has been built and simulated in various tests.

The results of the tests were evaluated. The prototype functioned well and almost all requirements were satisfied. The combination of JSON and a NoSQL database proved to be a good combination and fits very well in such software environments; the unstructuredness of data can be handled very well by the flexibility of this solution. However, one requirement was not fully satisfied: the requirement of guaranteeing a minimum amount of identified data links.

However, this research proved the technical feasibility of this implementation and can therefore recommend to continue research on this approach. The main recommendation is to do more research on application integration (mappers) and integrating natural language processing\textsuperscript{4} techniques to make the system better capable of understanding data and thus identifying links between data.

It is a promising set-up which definitely deserves more research, which can eventually lead to application regaining their full power while its generated data is properly indexable and linkable.

---

\textsuperscript{2} a transport format especially for transporting all kinds of data
\textsuperscript{3} a schema-less database to save these JSON-files
\textsuperscript{4} technologies that improve a machines capability to understand text written by humans
My name is Peter Wiggers, student Construction Management and Engineering at Delft University of Technology. I have written this thesis as part of my graduation research. The research took place from May 2014 until January 2015.

I have always been interested in bringing my two favourite disciplines together: civil engineering (my study) and IT (my hobby/profession). In my bachelor thesis I researched the possibilities of an interactive highway, preventing rear-end collisions. This master thesis also combines these two disciplines. Recent history and predictions show that these disciplines keep getting tighter interwoven. This research also jumps in on this combination. Therefore, the graduation committee of this research also consists of people from both these disciplines.

My committee made it possible for me to have various perspectives and bring this research to a higher level. I would like to thank Rogier Wolfert (professor and chairman, CiTG), Jan Hidders (IT, EWI) and Martinus van de Ruitenbeek (Civil engineering and IT, Volker InfraDesign) for their support and for making this research possible. Second, it would like to thank everybody who helped me by during my research; from the people I interviewed to my family.
# CONTENTS

Abstract ii  
Preface v  
Table of Contents vi  
Acronyms ix  

1 INTRODUCTION 1  
1.1 History 1  
1.2 Current situation 2  
1.3 Other research 3  
1.4 This research 5  

2 RESEARCH FRAMEWORK 7  
2.1 Scope 7  
2.2 Problem Analysis 7  
2.3 Research Question 9  
2.4 Hypotheses 9  
2.5 Objective 9  
2.6 Research Methodology 9  

3 ANALYSIS 13  
3.1 BIM: Visualize 13  
3.2 BIM: Inform 15  
3.3 Software 16  

4 SYNTHESIS: THEORY DEVELOPMENT 19  
4.1 Requirement Analysis 19  
4.2 Scope-specific interpretation of semantic web principles 20  
4.3 Setup design of data extraction part 23  
4.4 Setup design of data linking part 26  
4.5 Setup design of comparing data part 33  

5 SIMULATION: THEORY TESTING 35  
5.1 Tests 35  

6 EVALUATION 43  
6.1 Requirement Evaluation 43  
6.2 Hypotheses Evaluation 48  

7 APPLICATION SCENARIOS 49  
7.1 Specific information extraction 49  
7.2 Search engine 49  
7.3 Revisioning and backups 50  
7.4 Analyses 50  
7.5 Case-based reasoning 50  
7.6 Exchange 50  
7.7 Custom dashboards 51  

8 CONCLUSIONS 53
Contents

9 RECOMMENDATIONS 55
References 57
APPENDICES 61
### ACRONYMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIM</td>
<td>Building Information Model</td>
</tr>
<tr>
<td>IFC</td>
<td>Industry Foundation Classes</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>BC</td>
<td>Building &amp; Construction</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>OTL</td>
<td>Object Type Library</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>PDF</td>
<td>Portable Document Format</td>
</tr>
<tr>
<td>NGO</td>
<td>Non-Governmental Organization</td>
</tr>
<tr>
<td>DBM</td>
<td>Design, Build and Maintain</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>NOSQL</td>
<td>Not Only Structured Query Language</td>
</tr>
<tr>
<td>DMS</td>
<td>Document Management System</td>
</tr>
<tr>
<td>GSA</td>
<td>Google Search Appliance</td>
</tr>
<tr>
<td>SE</td>
<td>Systems Engineering</td>
</tr>
<tr>
<td>AM</td>
<td>Asset Management</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographic Information System</td>
</tr>
<tr>
<td>HLM</td>
<td>Highway Location Marker</td>
</tr>
<tr>
<td>OTL</td>
<td>Object Type Library</td>
</tr>
<tr>
<td>OL</td>
<td>Object Library</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Values</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>WPA</td>
<td>Work Package Activity</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
</tbody>
</table>
INTRODUCTION

1.1 HISTORY

“Since the advent of the personal computer, the last 50 years witnessed a digital revolution which has evolved at such a phenomenal rate and shows no signs of slowing down.” (Underwood & Isikdag, 2011).

This digitalization enabled us to realize larger and complexer projects, with more stakeholders involved. However, this complexity resulted in the fact that the Building & Construction (BC) industry continually has been criticised for its poor performance and efficiency, and failure to deliver to the client, compared to other industries (such as the aviation industry) (Latham, 1994; Egan, 1998, 2002). 2D drawings were not sufficient any more because of the many interfaces between various disciplines. The main function of 3D drawings was just visualization and had no connection with information from other disciplines.

To solve this problem, the idea of Building Information Model (BIM) was born (Van Nederveen & Tolman, 1992). A BIM is a 3D model that represents the construction to be built including various aspects of the construction, for example walls, plumbing and electricity. This was set up as an object-based approach, with the idea to connect information to specific objects.

Developments lead to extensions to BIM: BIM 4D stands for connecting the 3D model with planning information gathered from the planning application. This way, a 4D visualization can be generated and can help identify clashes during construction. Plans are to continue with 5D (4D plus costs), 6D (5D plus facility management), 7D (6D plus sustainability), 8D (7D plus safety) and who knows what the future will bring (Kamardeen, 2010).

To use a BIM effectively, a high level of standardization is required. To make different applications able to exchange information with each other (and gather the data in the BIM), they all have to “speak” the same language. However, history learned that it is practically impossible to have one, all-encompassing standard in a free market that is accepted by all players (Pratt, 1915, Tassey, 2000). This is backed by just looking around: is there just one type of A/C charger, instant messaging service, language, operating system or driving orientation? No. The attempts to create this one standard often leads
1.2 Current Situation

According to various BIM-using companies, it is just a matter of time before software developers develop their software being generally interoperable. However, this can be argued since there are several advantages for software developers not to do so (e.g. vendor lock-in, development efficiency) and it is very hard to get consensus about that interoperability standard – as stated above.

A clear distinction in BIM functionalities can be made: to visualize and to inform. 3D and 4D represent visualization functionalities since the visualization helps the user to identify clashes. Users have become enthusiastic about this approach and its functionalities and wanted the continue this development to BIM nD. However, BIM 5D (costs) has for example no advantages of this visualization and thus might not require the standardizations needed for BIM 4D, making it easier to implement and use.

Taking into account the current and expected problems of BIM’s current way-to-go and the fact that BIM’s informing part maybe requires less standardization, the following question arises:

“Should we continue this way – and wait until the disadvantages of BIM catch up with the advantages of BIM – or should we go an other route?”
It might be wise to use BIM just for visualization purposes and stop BIM at 4D, and come up with an other implementation for connecting data from different sources and achieve the original BIM nD ideas. Maybe we have to accept the ironclad fact that the required standardization is impossible in practice and that generated data during the entire lifetime of a project will remain flowing in in an unstructured way. Adding D’s to BIM just makes this problem even bigger. Let’s give the application its strength back. Let’s give the users freedom in their work-flows. Let’s not misuse a 3D model as a central database-like model, but just as one of the ordinary data sources. This implies that it is the other route we have to go. But what can that route look like?

1.3 OTHER RESEARCH

Inspiration for this route might be found elsewhere. The main goal is to have the data from various sources relate to each other. A place where this works very well – without demanding a specific structure or data format – is the world wide web. Since everybody has the ability to put information on the web according to his preferred structure, naming conventions and format, the web can be seen as an unstructured, loosely coupled information source (Newcomer & Lomow, 2004). However, this information still has no relations to other information. This is where the semantic web comes into play. This concept enables the web to easily define relationships between entities and also make it easy to map differently structured data sources to each other (Berners-Lee et al., 2001). It seems the situation and goals on the web are quite in line with those in the software environment of a Design, Build and Maintain (DBM) infrastructure project, where information management is important during the entire lifecycle of the project.

Currently there are various researches emphasizing the need to take the other route. They all acknowledge the problems when trying to create an optimal software environment by forcing a high level of standardization (vendors, technology, workflows). These researches are elaborated in the next paragraphs.

Pauwels, De Meyer, & Van Campenhout (2011) not only use the concept of the semantic web, but also want to use its technology to connect information. With semantic web, computers can understand the data, making it easier to automatically link objects, resulting in new information. However, this approach appeared to be very complex and requires software developers to upgrade their applications to work semantic web formats like Resource Description Framework (RDF). The idea of using semantic web in this context (Ibrahim et al., 2004) originates a couple of years after the introduction of semantic web on the world wide web (2001) and still is not ready for production use.
Figure 2: Is it already possible to make computers understand information without explicitly telling it what it is? Can the use of semantic web technologies be skipped?

Secondly, the developments in Natural Language Processing (NLP) technologies and artificial intelligence might make this explicit formulation to make computers understand the data superfluous, as shown in Figure 2. Secondly, using semantic web technology in this context imposes two unpractical consequences: files have to be machine readable, i.e. not Portable Document Format (PDF) and files have to be saved in a non-proprietary format, i.e. not Excel (van Nederveen et al., 2014).

Federated BIM (van Berlo, 2012) and distributed BIM (Ibrahim et al., 2004) acknowledge the current and future problems of BIM and try to solve it with a decentralized approach, which is currently quite common.

Ibrahim et al. (2004) describe the need for a distributed BIM, where the model is a referential model, keeping the data located at the original source, which can also be the web. However, this solution requires the information to be put on the web and in an open format like JSON. The advantage of this approach is that data automatically stays up-to-date; if the manufacturer makes a modification in the specifications, it will automatically be updated in the BIM. However, the disadvantage of this approach is that it requires manufacturers to provide such a service, which is not reasonable to expect.

van Berlo (2012) states that there is no such thing as a central BIM: every project consists of one big BIM and for every discipline a little BIM. The advantage is that these little BIM’s only have to work with some selected software package (packages within that discipline or
little BIM). This makes it possible to have multiple (internal) highly exchangeable little BIM’s. However, the difficult part is to make these little BIM’s understand each other. This still is a big issue if you require such intercommunication. Second, these BIM’s use the same approach as BIM currently does – this approach is only easier to implement within the little BIM’s (for the reason mentioned before).

Second, both approaches do not make a distinction between BIM Visualize and BIM Inform, making these approaches suboptimal in this research’s vision.

All three researches have one important thing in common with this research: they all go the way of requiring less standardizations and solving the problems that gives in a different way. However, the key difference with this research is the technology used. This research considers the technologies used as non-optimal and tries to prove that it is easier to go this route with different technologies, such as JSON and NoSQL.

Google has a generic system that is more in line with this approach. It’s Google Search Appliance (GSA) (Google, 2014) is a loosely-coupled system that is able to gather data in all kinds of formats, connecting it and making it available for the end-user. It can be placed in most IT environments and “just does its job”: indexing a lot of data, no matter its source our format. However, a reason that the GSA will not fit in this situation is that it is closed-source, so you cannot entirely modify it to your needs or extend specific functionalities (although it has an extensive API). Second, the GSA is not built for this specific purpose: it is a generally applicable search engine. Being able to find information quickly is a part of the solution.

This research investigates if and how it is possible to fill in this other way using different technologies. It is well underpinned to go the other route, and this research investigates the possibilities with JSON and NoSQL as technologies used within that route. It proposes to shift away from current low-level BIM developments (the unmeritorious attempt to create a Blocks World), towards addressing real problems in the real world. In order to prove the aforementioned examples and principles, they have been implemented in a rudimentary form for a real DBFM project with multi-sources engineering assets information.
RESEARCH FRAMEWORK

In this chapter the aspects leading to a valid and comprehensive research question will be elaborated. It includes a scope definition, a more extensive problem analysis, the resulting research question, the hypotheses, the objective and the way to achieve that: the research methodology.

2.1 SCOPE

This research checks the technical feasibility of a (for this industry) new data linking and retrieval set-up, used by (a consortium of) client(s) in integrated infrastructure projects. It does not focus on usability, legal consequences – just on the technical feasibility.

The data to be linked is data that concerns applications used for Enterprise Asset & Resource Management in these complex infrastructure projects.

Third, it focuses a different approach for textual links – it does not include a different approach for BIM 4D’s visualisations.

Setting this scope also enables this research define its borders and to select commonly used software and applications. It also emphasizes that this research tries to find a solution for projects that has multiple contractors and produce a lot of data. The solution will not be designed for small, single-contractor projects, although it might still appear to be useful.

2.2 PROBLEM ANALYSIS

BIM is theoretically a great solution for the communication problem mentioned in the Introduction. For all aspects applies: it is easier to achieve the required level of standardization within a company, but it is very difficult to achieve the same result within a building consortium, where each company has its own internal standards. Unfortunately, complex infrastructure projects are nearly always executed by building consortia.
Sub-problems

Software and interoperability
Given the fact that data from various applications in various disciplines has to be imported to one, central BIM, there is a limitation on the software to use.

Data structure
Since it has to be able to import data from one application to an other, the data format, structure and contents have to be standardized (for example, IFC). During this exchange in a standard format, information is lost, which is of course not preferable. In order to be able to choose any application one prefers means that one cannot have high restrictions or requirements to the application’s data format and structure (a software developing company should define the format and structure that fits best for their product).

Workflow and skills
Within a building consortium, companies have various levels of BIM adoption and various years of experience, making their workflow and skills differ. To work together in one BIM, the workflow and skills have to be aligned. Company A can have a different workflow regarding creating/maintaining a BIM than company B, which can be a result of personal preference or a difference in skill level. This leads to the problem that employees might not know or understand how to work with the model or how to interpret others’ modifications: a chain is as strong as its weakest link.

Naming convention
Naming problems can apply within building consortia or with the client. Within a building consortium a central Object Type Library (OTL) is needed. Every piece of information has to be connected to an object type from this OTL to have it properly connected in the BIM. This means that employees from a company might work with different naming conventions for every project. For example, a corridor can be named a corridor by Company A and a hallway by Company B, while they mean the exact same thing. One of the goals of this research is to avoid that, and let companies define their own standards (and thus use their own, internal OTL).

Problem Statement
These sub-problems produce the following general Problem Statement:
2.3 Research Question

Currently, using BIM requires a specific workflow and technology in order to achieve linked data for information retrieval purposes, which appeared to be hard to realise.

2.3 Research Question

The Introduction and problem description led to the definition of the following research question:

*Can a vendor and technology independent indexer provide data linking and retrieval in EAM software in a less standard-requiring way?*

2.4 Hypotheses

To answer the research question, this research tests the following hypothesis:

\[ H_1: \text{A JSON-based data crawler can achieve linked data within software environments of large Dutch infrastructure without requiring predefined workflows.} \]

And the following null hypothesis:

\[ H_0: \text{A json-based data crawler cannot achieve linked data within software environments of large Dutch infrastructure without requiring predefined workflows.} \]

2.5 Objective

The main objective of this research is to investigate whether a combination of modern concepts and technologies used on the world wide web can be used to set-up a more loosely coupled alternative to BIM. To check if such a solution is possible, a prototype will be built and some tests will be conducted. With these results, the research question can be answered.

2.6 Research Methodology

This research is a theory oriented prototype development. It suggests a theoretical solution which will be tested with the creation of a prototype. In a simulation this prototype will be tested and these results will be used to decide if the prototype worked as expected and can be considered a solution for the problem.

Therefore this research uses the – in this case – suitable *Iterative Design Process* methodology [Nielsen 1993]. This framework prescribes the steps showed in Figure 3.
2.6 RESEARCH METHODOLOGY

Please note that this theory also prescribes to test with end-users to find bugs, but since this research’s goal is not to create a final end-product but a minimal prototype to test the theory, testing is done by this research’s author.
2.6 RESEARCH METHODOLOGY

Figure 3: Iterative Design Process methodology (Nielsen 1993)
ANALYSIS

The distinction this research created between BIM functionality areas is to visualize and to inform. Paragraphs 3.1 and 3.2 describe the (expected) desired functionality for these parts. Common software environments within the scope are analyzed in paragraph 3.3 to determine the requirements for the new solution.

3.1 BIM: VISUALIZE

Currently BIM 4D is in production use. It enables users to prematurely detect two types of clashes:

1. time-bound physical clashes (dynamic). The clashes occur in the construction phase, for example two clashing beams that need to be placed on the same time.

2. non-time-bound physical clashes (static). These clashes can be a result of people from different disciplines designing the same construction, for example the plumbing clashes with the electricity cables.

These premature detections give a company more freedom to make adjustments to the design, which is cheaper and easier in an early stage. This is visualized in the MacLeamy curve, see Figure 4.

This section focuses on the time-bound clashes, since these can only be detected by connecting the 3D model and the planning. These visualizations can have two purposes: visualizations for external use and visualizations for internal use.

3.1.1 4D Visualization for external use

External parties, such as Non-Governmental Organization (NGO)’s and (nearby) residents often resist to construction works in their neighbourhood. To make them willing to cooperate easier, a 4D visualization can help. This way, residents know what and when to expect in their ‘backyard’. However, because of the difference in priorities, this visualization can not be the same as the internal 4D visualization, because it is not attractive to watch. Details such as the colour of the
Figure 4: The MacLeamy curve: BIM allows a contractor to shift personnel to an earlier stage, where modifications are easier and cheaper to execute (buildingSMART, 2012).

guardrail or the number of trees can be important to a resident. Since this information is often not available in the 3D model, the 4D visualization for external use has to be created manually. This is a costly and time-consuming process, but is often worth the efforts.

3.1.2 4D Visualization for internal use

With a connection between the 3D model software and the planning software, a 4D visualization can be automatically generated. However, this is only for internal use since it is quite a technical visualization, especially for clash control. The visualization can be understood by professionals working on the project and it can help them to give them an impression which can be easily regenerated (e.g. after planning changes).

For this to work, it is necessary to use a third-party application to combine the 3D model and the planning. A commonly used application within the scope is Autodesk Navisworks. This application supports a lot of file formats, for 3D modelling software as well as planning software (Autodesk Building Industry Solutions, 2014).

For such a visualization, this research considers this approach a good approach because:

1. A separate application is needed to create a visualization – this can not be handled by just database-driven information connections, it really needs to generate a video.

2. It requires a specific object-based approach. Every planning task needs to be connected with a specific object.
3. It requires a high level of accuracy. If the planning and the 3D model to not correspond well, you can be overwhelmed with false clashes or miss clashes.

However, one is still dependent of a third-party and its supporting formats, and that other companies within a consortium work also with supported software. This implies that this setup should only be used when it is the best option available. This research considers an alternative for BIM 5D and further to be a better option.

3.2 BIM: INFORM

The main difference with BIM Visualize is that these connections between information can be done on database-level. It is just about linking information and not about creating additional output such as a video. This is a key difference, enabling a different approach than necessary for BIM 4D.

In practice, BIM Inform is about linking information from different sources so:

1. information can be grouped in new ways (object-specific, author-specific, etc.)
2. new information can be generated, based on analyses of existing information and its relations.

Currently all information is located in discipline-specific islands: the planning information is located at location A in format X, cost information is located at location B in format Y, etc. Main focus is to make the information from these islands connect to each other. This is schematically visualized in Figure 5.

New information can be generated because of the identification of relationships between information located on different islands. Let’s say you want to know when lamppost ABC is due for maintenance. It is common that in a maintenance program, not every single object is mentioned, but a range of objects – in this case the range is a Highway Location Marker (HLM) range. So how can this link be defined? Currently this is mostly done by asking person A for the location of lamppost ABC, for example HLM 51.2, and then ask person B for the next maintenance of this lamppost. This is a devious process. Easier would be that, after indexing and linking all information, a connection between the information on different islands could be established, making it possible to ask a question that contains elements from multiple islands: “What is the production date of every lamppost on the A7 that cost more than 5000 euro and is due for maintenance this year?” Just this simple question already needs information from four of those islands. Currently, these questions are hard to answer.
To achieve this functionality in a loosely-coupled, adaptable way, a new setup has to be designed, which will be done in Chapter 4.

3.3 SOFTWARE

It is important that no adjustments to commonly used software (within the scope) have to be made to index its data. Therefore, two elements have to be investigated:

- Which applications are in use, what data format(s) do they use and where are those files stored?

- What export capabilities do the storage engines support?

In the table below, commonly used scope-related applications are listed including their data format(s).
### 3.3 Software

<table>
<thead>
<tr>
<th>Category</th>
<th>Application</th>
<th>File format</th>
<th>Data location</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D modelling</td>
<td>Autodesk AutoCAD</td>
<td>IFC</td>
<td>DMS</td>
</tr>
<tr>
<td></td>
<td>Autodesk Navisworks</td>
<td>IFC</td>
<td>DMS</td>
</tr>
<tr>
<td></td>
<td>Autodesk Civil3D</td>
<td>IFC</td>
<td>DMS</td>
</tr>
<tr>
<td></td>
<td>Autodesk Revit</td>
<td>IFC</td>
<td>DMS</td>
</tr>
<tr>
<td></td>
<td>Autodesk InfraWorks</td>
<td>IFC</td>
<td>DMS</td>
</tr>
<tr>
<td></td>
<td>Tekla Structures</td>
<td>IFC</td>
<td>DMS</td>
</tr>
<tr>
<td></td>
<td>Allplan Engineering</td>
<td>IFC</td>
<td>DMS</td>
</tr>
<tr>
<td></td>
<td>SolidWorks</td>
<td>IFC</td>
<td>DMS</td>
</tr>
<tr>
<td>Planning</td>
<td>Oracle Primavera</td>
<td>REST JSON[F]</td>
<td>Web service</td>
</tr>
<tr>
<td></td>
<td>Microsoft Project</td>
<td>REST JSON[F]</td>
<td>Web service</td>
</tr>
<tr>
<td>Systems Engineering</td>
<td>Relatics</td>
<td>REST JSON</td>
<td>Web service</td>
</tr>
<tr>
<td>ERP</td>
<td>Oracle ERP</td>
<td>REST JSON</td>
<td>Web service</td>
</tr>
<tr>
<td></td>
<td>SAP</td>
<td>REST XML</td>
<td>Web service</td>
</tr>
<tr>
<td>GIS</td>
<td>ArcGIS (KernGIS)</td>
<td>REST JSON</td>
<td>Web service</td>
</tr>
<tr>
<td>Asset Management</td>
<td>IBM Maximo</td>
<td>REST JSON</td>
<td>Web service</td>
</tr>
</tbody>
</table>

Of course typical file formats, like Microsoft Office document formats or HTML webpages, are also taken into account.

Most companies also have in-house developed software. This software is automatically supported if it supports one of the above-mentioned export methods. If it does not, it can be investigated if a converter is already available or has to be created. This also applies to potential cloud services that are used – but they almost always provide a REST API.

Document Management System (DMS), used for savings all kinds of documents, that are commonly in use are Microsoft Sharepoint (support REST JSON), OmniComm Promasys (supports REST XML) and Cadac Organice (supports REST ODM). Data that is not stored in such a DMS is assumed to be accessible via REST API's, as shown in Table 1.
With the current situation analysed and the semantic web principles identified, a set-up according to these results can be designed, which is described in this chapter.

4.1 REQUIREMENT ANALYSIS

To avoid problems that currently occur with BIM, the new setup has to be designed keeping the following project specific requirements in mind. The solution has to be able to ...:

- ... not rely on applied standards in the workflow and software-choice of the user;
- ... work pro-actively, meaning the system has to do its job entirely on the background, without demanding any action from the user;
- ... accept all data formats and try to read and index as much as possible;
- ... be able to link data on various identifiers;
- ... be able to create dimension bridging (inferred) links;
- ... map information from same-purpose applications to a uniform structure;
- ... be easily extendible and adjustable;
- ... guarantee a minimum accuracy when it comes to correctly identified links between data;
- ... be scalable;
- ... allow companies within a consortium to each use their own naming conventions;
- ... work independently next to a company’s current BIM environment;
- ... work guaranteed with the software listed in section 3.3.
4.2 SPECIFIC INTERPRETATION OF SEMANTIC WEB PRINCIPLES

The system also has to satisfy the general six main principles of the semantic web [Siegel, 2010], since these are taken as the basis for the system:

- Data will become easily found;
- Data will be reusable;
- Data will be interoperable;
- Devices will be ubiquitous;
- Systems will be flexible;
- Real time.

Since these principles are quite general, they have to be interpreted for this scope.

4.2.1 Data will become easily found

Since the data is produced in various sources, it needs to be gathered to make it easily indexable. However, all data has their own structures and formats. In this research, one assumption is that data can have a structure that is unknown and therefore difficult to import to a Relational DataBase Management System (RDBMS). However, the last couple of years, there has been great development in the Not Only Structured Query Language (NoSQL) area, resulting in flexible, real-life databases for real-life data. According to Anderson et al. (2010), the CAP theorem applies to databases. This theorem describes a few different strategies for distributing application logic across networks. The CAP theorem identifies three distinct concerns:

- **Consistency**: All database clients see the same data, even with concurrent updates.
- **Availability**: All database clients are able to access some version of the data.
- **Partition tolerance**: The database can be split over multiple servers.

The problem is that you can only pick two, see Figure 3.

In this scope and according to the semantic web principles, Availability and Partition tolerance are the most important concerns. The importance of the partition tolerance may need some explanation. Since this scope’s projects can become very large and can take a long time,
4.2 Scope-specific interpretation of semantic web principles

Figure 6: The CAP theorem. Pick two. (Anderson et al., 2010)

lots of data will be produced. Therefore it can performance-wise become necessary to spread the data across multiple machines – something that is impossible for an RDBMS. Consistency is not possible because of the assumption that data can have an unknown structure and the contents can also be unknown.

This leads to a logical choice for a NoSQL database (CouchDB in Figure 6). CouchDB is a good choice because of its scalability, JSON support, document-oriented structure and flexibility (no fixed schema’s) (Apache Foundation, 2014). Another advantage of a NoSQL database is its speed, which is, amongst others, a direct result from the partition tolerance. The fact that the data can be spread across multiple machines also means that queries can be spread across multiple machines. This improves the performance.

4.2.2 Data will be reusable

Since all data will be imported and all revisions will be kept, data can be reused for different purposes or new projects. For example, data from new projects can easily be compared to data from old projects by discovering patterns. These patterns can help to make the right choices in new projects.

4.2.3 Data will be interoperable

There are some steps to make to achieve this. Various applications use various structures and various data formats, making them not interoperable. However, all data has to be indexed by the indexer (into the database) so at some point the information has to be converted to a format the indexer understands. To achieve this, the data has to
be processed, which is visualized in Figure 7. The data formats are identified in section 3.3.

Since the data will be saved in a flexible JSON NoSQL database, it would be wise to convert the file to JSON, regardless its contents. The Apache Tika toolkit detects and extracts metadata and text content from various documents - from PPT to CSV to PDF - using existing parser libraries and can output the results in JSON [The Apache Software Foundation, 2014]. The converters already exist and are publicly available [Wang, 2011; Glawischnig et al., 2012].

This approach makes it easy for the contractor to switch applications according to its needs – not according to the interoperability requirements of a BIM.

![Format convert process](image)

**Figure 7:** Format convert process

### 4.2.4 Devices will be ubiquitous

It should not matter which devices the user is using. Standalone applications require specific hardware, operating system or screen resolution, making this quite hard. The web however is supported on most devices, from smartphone to desktop PC, and thus this aspects underlines the need for a web-based approach. Future developments, such as smart watches and smart glasses, will all be able to understand the web language, making those devices also applicable.

### 4.2.5 Systems will be flexible

“Today’s procedure-driven software is already broken (most enterprises spend 80 percent of their IT budgets on maintenance). Tomorrow’s flexible systems will be adaptive. They’ll respond in real time to business events and change themselves daily as business environments change.” [Siegel, 2010].
The solution should be easily adaptable, emphasizing the needs for a custom made solution and not a closed-source universal application. This gives freedom to the consortium and makes it usable for all its unique projects.

Another important factor is the fluctuating needs for processing power and storage in DBM projects. Roughly said; the amount of data to be processed in the design/build phases will be a lot more than in the maintenance phase. The amount of data in storage will always grow since no data will be deleted. The fluctuating demands require a flexible system setup.

4.2.6 Real time

Data has to be real-time accessible, emphasizing the correct choice to use web concepts, where data is 24/7 available. An automatically and periodically running script ensures the data is always up to date (this depends on the interval of the runs).

4.3 SETUP DESIGN OF DATA EXTRACTION PART

Taking into account BIM’s current problems, the required/desired functionality, common software environments, the interpretation from the semantic web principle and the available (web) technologies, the theoretical setup has schematically been visualized in Figure 8.

In the next paragraphs the solution is elaborated from top to bottom.

The cloud with various applications represents a company’s software environment. This can be every combination of the application as listed in section 3.3. This is visualized as a cloud since it is not a static set of software, but can change between and within projects and can contain applications from various construction parties in the same project. All these applications need to be able to be “plugged in”.

There are to methods to “plug in” an application: it can have its own web service (making the data real-time accessible via a REST API) or it produces files, which can be saved on a DMS which provides a web service.

Crawlers sent from the indexer, check these REST API’s and gather new data from the source. This data is brought back to the indexer in various formats. Because the indexer’s database is a JSON NoSQL database, all information needs to be converted to JSON to index it properly. The original binary data can be attached for other purposes.

1 Source-specific scripts that speak the language of the web service of the source. They “ask” the source for (new) data.
Figure 8: Schematic design of a possible solution.
Once the data is converted to valid JSON, it needs to be processed by an application-specific mapper. Each application can have its own mapper if the structure is different than in other mappers. This allows one to easily add new applications which have a different export structure. For example, this setup does not prescribe an application to use TITLE as key for the title of a record, but it can be named (or structured) to whatever that application fits best (according to its developers). A mapper takes care of this conversion. If the structure of two applications is alike but not identical, a mapper can inherit functionalities from another mapper, overriding elements as needed. This way, no standardized language is needed, every application can be mapped but no redundant work has to be done, see Figure 12. Notice the differences in structure, field names and values in the figure.

An advantage of the use of mappers is that they are easily modifiable and can be done by non-programmers. An example of a simple mapper is shown in Figure 9.

![Figure 9: An example of a mapper (written in Python). This mapper maps fields names as well as field values. Unmapped fields are also saved (untouched).](image)

This mapper maps the name of one column and the value of another column. For example, if the input would be an Excel file as shown in Figure 10.

The output would be a JSON string as shown in Figure 11.

Once the data is mapped it can be saved in the database. Please note that, since data can be unstructured, a mapper does not map all parts of the data. These unmapped parts are not removed, but will
still be indexed in the unstructured way, so it can still be retrieved. To clarify, an example: a planning task that needs to be imported can have fields like title, start-date, end-date and responsible. Let’s assume a company wants all fields to be translated to Dutch, but was not aware of the field responsible. The mappers takes care is this translation, but just for the first three fields (and if necessary its values). The output of the mapper would be Titel, Begindatum, Einddatum and responsible, and that would be the way it is saved to the database. No information is lost.

4.4 Setup design of data linking part

Now the data from all islands is stored in one database, it is time to link the data keeping in mind that the data has to be easily retrievable.

To link the data is the main objective of this system and this is the part where the real power of this setup comes in. It needs to be able to link data from various sources to each other. The link subject can be – as opposed to the current BIM approach – diverse: it can be object-based, author-based or whatever common properties the pieces of data have. This section is divided into two sections: How to link, explaining the way data can be linked and When to link, explaining when in the process the links are established.

4.4.1 How to link

In the current BIM approach, everything is object-based and a piece of data can only be linked to an other piece of data if they share the same object id. This required standardization is very difficult to achieve in practice, because various applications do not work with object id’s. There are also applications or workflows that do not specify single objects, but ranges of objects. For example, IBM Maximo – an application often used to control maintenance – works with these ranges. Let’s assume we have four lampposts with object id’s A, B, C
Figure 12: Various mappers for various applications, resulting in a uniform output. This is elaborated in section 5.1.3.
and D that need its light bulb replaced every five years. A common workflow is that IBM Maximo has no clue about these object id’s, since there can be thousands of the same lamppost that needs the same maintenance and it would be inefficient to set all the object id’s. Therefore, it uses ranges. This can for example be a range defined in HLM e.g. from HLM 51.2 to HLM 55.6. This situation is drawn in Figure 13.

Figure 13: How to link maintenance tasks from IBM Maximo (HLM-range-based) to objects from the object library (object-id-based)?

What if someone wants to know the maintenance schedule for lamppost B? The current solution is that that person calls a person from the Geographic Information System (GIS) department, who can tell him where the lamppost is located. Then our person calls someone from the maintenance department to ask the scheduled maintenance for lamppost B, from which he knows its corresponding HLM. With the current developments of BIM, only the second step in this situation would become easier (if they would implement maintenance in BIM nD).

To make this process more efficient, this research uses inferred links, which this research calls dimension bridging links. Such a link uses multiple D’s (from BIM nD) (or islands or disciplines) to make a link between data from two different sources. How would this function in the situation described in the previous paragraph? It would use the information from more than two sources. The OTL contains all unique object types – e.g. a lamppost and a traffic barrier. The Object Library (OL) contains all instances of the object types – e.g. lamppost A, B, C and D as instances from the object type lamppost. Every OTL record contains information that is the same for every instance of that type – e.g. its energy consumption. Every OL record contains information that can be different for every instance – e.g. the object id. The GIS contains the location of the objects used – e.g. the location of the HLM’s and the lampposts. Finally the Asset Management (AM) application (e.g. IBM Maximo) contains the required maintenance
setup design of data linking part

Figure 14: Dimension bridging (using OTL and GIS in between) linking, visualized.

tasks for every object grouped by range – a range type that suits best for that object type. Now the link can be made. If we start with lamppost A, the GIS can be queried to retrieve its location (in HLM) according to its object id. Second, since the AM application does not use object instances but object types, we need to know the object type of lamppost A. Therefore the OTL can be queried to retrieve the corresponding object type id. Subsequently, with the location and object type, the AM application can be queried to return its maintenance information regarding the object type and satisfies the range bounds. This process is visualized in Figure 14.

4.4.2 When to link

An Structured Query Language (SQL)-query executing the process from Figure 14 could look like this (if all databases would be SQL databases and would be accessible):

This can be, especially on large relational databases, a time- and resource-consuming query. Therefore this setup uses – as motivated in section 4.2.1 – a NoSQL database: CouchDB, and indexes it. Every
piece of data coming from any source will be stored in this one big
database. This situation considered, a query (to ElasticSearch, see
paragraph after the list below) can look like this:

```
| 1 | "query": { |
| 2 |   "bool": { |
| 3 |     "must": { |
| 4 |       "query_string": { |
| 5 |         "default_field": "*all*", |
| 6 |         "query": "onderhoud AND 517a6cd30574814d AND wiggers" |
| 7 |     } |
| 8 |   } |
```

This query is way simpler than the SQL query, since it only contains
one variable: the search terms as a simple string (line 8). All other
lines are identical for every query. The computer does not have to
determine the semantic value of the terms, which makes the process
way easier (see the list of advantages of this approach below).

Note that this is only possible after mapping, see section 4.5.

A fundamental assumption of this research is that data that needs
to be indexed can be unstructured. For that reason, this research
takes a different approach than querying the indexed data: it takes a
keyword-based approach. These keywords will be added to the data
during the indexation of the data. This approach comes with huge
benefits:
• data can remain unstructured – keywords added during indexation can be queried;

• these keywords describe a record’s meaning and relations, making it for NLP technologies easier to understand the data;

• all data can be textually indexed by a specialized, making searches real-time, resource-efficient and very fast;

• keeps the system simple (no complicated queries have to be created. Relying on complicated queries is also a bad idea with unstructured data.);

• makes the system easily extendible, since no system modifications are needed for adding new data sources.

A popular, open-source technology is ElasticSearch, which will be used in this research. It shares the same advantages as CouchDB (scalable, etc.) but has a different purpose.

The process of indexing and adding keywords is visualized in Figure 17 and an example with fake data is shown in Figure 18. Consider the dynamic keywords block in the second figure. In this step, all object id’s of all instances of the lamppost object type are loaded and saved as keywords to the object type. After indexation by the ElasticSearch index, there is a keyword-based link between the object type and the object instance without executing any complex queries! The only input ElasticSearch needs is a keyword, and it will return all records that include that keyword (orderd by relevance). In this case, submitting ABCD1234 will return both the object type and the object itself (the instance of the object type), while originally there might have been no reference in the object type library of its instances. To see this in action, see Chapter 5 Simulation.

Second example: consider a meeting about a technical problem of the lamppost. The minutes of the meeting are saved in a Word document. There is no direct connection between the lamppost’s OTL record and the Word document. In reality, there is a connection since both records are about the same subject. With BIM’s current approach, a link between these two records should be defined explicitly, which is not plausible to expect. But here comes Apache Tika to help. This toolkit is able to read and export Word documents (and more!) to plain text, making it readable and thus indexable by ElasticSearch. Now both records contain the keyword lamppost, and therefore – analogous to the previous paragraph – are linked. Note: if deemed necessary, the same approach as in previous paragraph can be used: finding this Word document by object id, while the object id was never mentioned in the Word document.
4.4 Setup Design of Data Linking Part

Figure 17: Adding keywords in the process

Figure 18: Adding keywords in the process, an example
4.5 SETUP DESIGN OF COMPARING DATA PART

Since the import process runs periodically, data can change in the meantime. New data can be created and existing data can be modified or deleted. A possibility to refresh the data is to delete all indexed data and re-index all new data – every time. This is not a good solution, since:

- it is not resource-efficient;
- various application possibilities shown in Figure 8 become impossible, such as revisions, case based reasonings and analyses;
- identical data can be imported differently. If someone changes the functionality of a mapper, is does not necessarily mean that data created before the change should also be imported differently (consider a software upgrade that changes its exported file structure).

To avoid these problems, incoming data needs to be compared to the data that is already in the database. There are three types of data:

- unstructured files that do not need specific processing;
- (semi-)structured files that need specific processing;
- non-files, directly coming from an application’s web interface.

**unstructured files that do not need specific processing**

This category mostly contains manually created files coming from the DMS such as the Word document with the minutes of the meeting from previous section. Since the contents are unstructured, the only thing that can be done is trying to convert the file with Apache Tika to make it machine-readable, making it possible for ElasticSearch to index the contents.

For this type, it is easy to detect changes. It is possible to check the modification date in the document’s metadata, but that is not advisable since this setup really assumes unstructured data and that can also mean that there is no metadata. The best option is to calculate the MD5 hash2 of a file. This is a 32 character string, such as 857445967d4d168b3eb3b8a38d4b894. This string changes if just 1 bit of the file is changed. So if the system saves the MD5 hash on the first import and after that it compares the MD5 of the incoming file to the hash that is saved in the database, it knows whether the file has changed or not. This is a resource-efficient solution for comparing...

---

2 The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity. [Ciampa, 2009]
(semi-)structured files that need specific processing
These files are generally exports from applications that do not provide a web interface to extract its data from. Imagine a Comma Separated Values (CSV) export of the main task *Placing lampposts* with all its sub-tasks, coming from a planning application that does not provide this web interface. Since this data is according to a known structure, it can also be saved in a predefined structure. Consider the following example. The mapper receives the CSV of the main task. In the CSV the sub tasks are listed:

1. 29-10-2014: dig hole
2. 30-10-2014: place lamppost
3. 1-11-2014: connect wiring

Because we can prepare the mapper for this structure, it can read the lines and save it as sub tasks of the main task, which is the main record in the CouchDB database. CouchDB supports inline parent-child relations, which are useful for this situation. The main task (parent) has several sub tasks (childs).

All files in this category are structured and thus have a known structure. Therefore, the mappers that handles these sources have to be written accordingly so they can process these structures.

non-files, directly coming from an application’s web interface
Various applications provide web services (see section 3.3). This enables the crawler to ask specific questions to the application, such as (in human language) “Give me all modified planning tasks since yesterday”. Results are often returned in the JSON format, allowing the mapper to directly process the results. Since the crawler can ask for specific information and the results have a known structure, data from this category has the same advantage as the previous category: the system can understand the data.
SIMULATION: THEORY TESTING

With the prototype created according to the design in the Synthesis, tests can be performed to check its functionalities. The setup visualized in Figure 8 is created from the top until the NoSQL Indexer. To make the test as real-life as possible, the prototype has been extended with one of the end-use possibilities: the search engine (first circle) combined with specific extraction (second circle). This way all three elements (extracting, linking, retrieving) can be tested. Secondly, real-life situations will be used as test cases.

To check if the theoretical possibilities as described in the Synthesis do work, seven tests are performed. Every test checks specific requirement(s) which altogether check the complete set of requirements.

The following sources are emulated since no real access to the source was possible during this research. To download the data, please see Appendix B.

- Microsoft Sharepoint (emulated as the file system);
- Oracle Primavera (an exported Excel sheet from a real planning);
- IBM Maximo (emulated as a CSV);
- Relatics (SE) (emulated as an XML file).

5.1 TESTS

5.1.1 Keyword grouping

In construction projects, a common approach is to use a Work Package (WP), containing one or more Work Package Activity (WPA). A WP is a part of the project with one responsible company or person that will be realized with related WPA’s (Buijink 2013). A WPA consists of a generic activity, an object and a potential phase (Buijink 2013). A WPA contains the information needed to complete the task. An example of a WP can be “Realizing street lights” and WPA’s within that WP can be “Placing lampposts” and “Painting lampposts” – as long as the activity itself is generic. The idea is that the information in a WPA should be given as a package to the person responsible
and he should be capable to complete the task with the information. That means that the WPA has to contain information from multiple sources. For the WPA “Painting lampposts” – for which a painter is responsible – information like which paint to use (OTL), the painting guidelines (DMS), locations of the lampposts (GIS) and the planning (Planning) are crucial. To retrieve this information, one could search for “lamppost”, but then one would be flooded with (non-important) results (see Figure 19, left side). The second option is to perform multiple searches, like “lamppost planning”, “lamppost gis”, “lamppost otl” etc. However, that is also inefficient. Since the prototype supports the addition of keywords, this can be done easier. During the import, a mapper can add keywords. Each mapper knows its source and the type of document and can therefore decide to add the WPA keyword to the record. This is only done for data that might be important for a WPA. Now, a search for “wpa lamppost” returns all WPA-related information, see Figure 19, right side.

This grouping can be done for any purpose a consortium considers useful. For example, groups per job function can be created. An engineer might be just interested in (3D) models, calculation sheets and object type information. Adding the keyword engineer to these corresponding records makes the engineer retrieve more accurate results with less non-relevant information.

5.1.2 Revisions

In section 4.2.2 it is explained why revisioning functionality is important. This simulation tests that functionality. It starts with an empty database, to which the prototype imports a planning with several tasks and a Word document with meeting minutes. The first import,
it should add the items to the database as new items. The second import is ran immediately after that, without any changes. It should ignore all data because the MD5 hashes (see section 4.5) are equal to those calculated at the first import. After this, the following modifications have been made:

- One planning task has a later start and end date;
- Someone else is responsible for two planning tasks;
- Some grammar corrections have been done in the meeting minutes.

In Figure 20, the results of the three imports are shown. The modified records should be at revision 2, see Figure 21.

### Figure 20: Import results

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>_id</td>
<td>69e8e0f2a2c97c5050222de46415c253380</td>
</tr>
<tr>
<td>_rev</td>
<td>2-09e712e702f8b86d477e077e07d20f4e8</td>
</tr>
</tbody>
</table>

### Figure 21: Modified file is at second revision

5.1.3 Extendibility and adjustability

The mappers are the heart of the import process: they convert data to make it fit in the database. This simulation is to prove this setup’s extendibility and adjustability: two very important factors in an environment of unstructured data and changing players and software. Since each source can have its own mapper (or multiple) it is easy to adjust its working or add mappers for new software. For this test, consider the following situation. A company uses Microsoft Project 2007 for its plannings. Microsoft Project 2007 exports a date in the following notation: YYYY-MM-DD. Since this is the notation the database can read, it does not have to be mapped to a different notation. Meanwhile, the company upgrades to Microsoft Project 2013, which uses the notation D-M-YY. The database does not recognize the date format and leaves it empty. The company notices the empty dates in the database and adds two lines of code that reformat the date to

---

1 this is just as an imaginary example
5.1 Tests

YYYY-MM-DD, see Figure 22. The next import, the dates will correctly be formatted to the desired notation. Previous imports (with the old notation) will not be modified, and that is preferred. The second advantage of the fact that the company can edit its own mappers is that they are not dependent of a third-party with a closed-source application.

![Date formatting function in the mapper](image)

**Figure 22:** Date formatting function in the mapper

---

5.1.4 Dimension bridging links

Another important requirement is the ability to link data while not using the same type of identifier (e.g. object id and range). To establish links between such data, data from other sources is needed as explained in section 4.4.1.

In this simulation the test case is as follows. The manufacturer of the lampposts called and told that lamppost with object id 61aa5b979121bf0e (they have been provided access to the OL) has a faulty bulb (production error), causing it to flicker. The company wants to check if the bulb will be replaced anytime soon, so I can decide whether or not to go to the lamppost and fix the single bulb.

The maintenance of the bulb can be found in the AM application; in this case IBM Maximo with an emulated CSV export. Under the hood it followed the process from Figure 14. That means every lamppost instance should now be connected to their corresponding maintenance tasks in IBM Maximo. This is done by adding all queried object id’s as keywords to their corresponding task. While the original object id was never mentioned in IBM Maximo, Maximo’s planning task now can be found by object id, see Figure 24.
5.1 Tests

Figure 24: Dimension bridging linking works with the prototype

The company sees that the bulbs have not yet been replaced, but that they will be replaced on 1 February 2015 and therefore can decide not to fix it immediately, but wait until all bulbs in that range will be replaced, which is cheaper.

5.1.5 Textual file indexation

While the semantic web technology based VCon solution prohibits the usage of files like Word and PDF (van Nederveen et al., 2014), this setup allows such files, and can even index its contents as long as Apache Tika is able to read it. This simulation is to test whether the prototype can actually read and index a PDF and a Word file. The PDF used is an instruction manual of a solar lamppost, containing (amongst others) the word “wrench”, on which it will be tested. The Word document is the file with the meeting minutes about the technical problems of the lamppost. This file contains the word “investigate”. Both results pop up when searching for those keywords, see Figure 25.

5.1.6 Different (same-purpose applications) inputs to uniform output

Within a consortium, it is possible that Company A uses application A to manage its planning, while Company B uses application B for the same purpose.

Since the conversion from various file formats to JSON has already been simulated, this simulation focuses on two differently structured plannings exported to CSV. The goal is to convert both to a uniform structure, so both plannings can be queried in the same way.

In Figure 26 both exported CSV’s are shown, while Figure 27 shows the uniform results after mapping.

Note: the mappers also supports value-mappings, but that is already simulated in Simulation 3 (varying date notations).
5.1 Tests

Consortium's Indexer

wrench

1 results (2ms)

- Bestand: gamasonic_gs04.pdf
  Sharepoint - PDF - administrative/gamasonic_gs04.pdf

investigate

1 results (2ms)

- Bestand: Constructeurnoverleg 13 november 2014.docx
  Sharepoint - Word - administrative/Constructeurnoverleg 13 november 2014.docx

Figure 25: Textual indexation of the contents of files

Export structure of application A

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Actief</td>
<td>Naam</td>
<td>Duur</td>
<td>Begindatum</td>
<td>Einddatum</td>
<td>Voorafgaande taken</td>
<td>Overzicht/Niveau</td>
<td>Notities</td>
</tr>
<tr>
<td>2</td>
<td>Ja</td>
<td>Contracts</td>
<td>0 dagen</td>
<td>1-6-2008 00:00</td>
<td>1-6-2008 00:00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Ja</td>
<td>Supply Let Sale Agreement</td>
<td>0 dagen</td>
<td>1-6-2008 00:00</td>
<td>1-6-2008 00:00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Ja</td>
<td>Supply Construction Agreement</td>
<td>0 dagen</td>
<td>1-6-2008 00:00</td>
<td>1-6-2008 00:00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Ja</td>
<td>Supply Contract Plans</td>
<td>0 dagen</td>
<td>1-6-2008 00:00</td>
<td>1-6-2008 00:00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Export structure of application B

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK_ID</td>
<td>ACTIVITIES</td>
<td>TASK_NODE</td>
<td>TITLE</td>
<td>START</td>
<td>FINISH</td>
<td>DURATION</td>
<td>PREVIOUS_TASKS</td>
<td>RESPONSIBLE</td>
</tr>
<tr>
<td>1</td>
<td>Ja</td>
<td>Automatisch gepland</td>
<td>Contracts</td>
<td>1-6-2008 00:00</td>
<td>1-6-2008 00:00</td>
<td>0 dagen</td>
<td>Peter</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Ja</td>
<td>Automatisch gepland</td>
<td>Supply Let Sale Agreement</td>
<td>1-6-2008 00:00</td>
<td>1-6-2008 00:00</td>
<td>0 dagen</td>
<td>Bob</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Ja</td>
<td>Automatisch gepland</td>
<td>Supply Construction Agreement</td>
<td>1-6-2008 00:00</td>
<td>1-6-2008 00:00</td>
<td>0 dagen</td>
<td>Jan</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Ja</td>
<td>Automatisch gepland</td>
<td>Supply Contract Plans</td>
<td>1-6-2008 00:00</td>
<td>1-6-2008 00:00</td>
<td>0 dagen</td>
<td>Matthius</td>
<td></td>
</tr>
</tbody>
</table>

Figure 26: Two different exports from two different (same-purpose) applications

Figure 27: Both exports are mapped to this uniform output.
5.1 TESTS

5.1.7 Speed

To make sure the data is real-time accessible, data not only has to be up-to-date, but also has to be quickly retrievable. The indexes of ElasticSearch make sure the information is very quickly retrievable, see the results in Figure 19 (12 ms, 16 ms) and Figure 24 (7 ms).

However, the simulation databases’ contents are quite small, the horizontal scalability also assures this kind of speeds for way larger data sets, see the example visualized in Figure 29, Appendix F. Please note the side notes for this claimed scalability in section.\[1\]
The null hypothesis will only be rejected (the solution satisfies) if all predefined requirements (section 4.1) are fulfilled. The next section checks every single requirement with the results from the simulation.

6.1 REQUIREMENT EVALUATION

Both the project-specific requirements as the semantic web principles are summed up and checked below.

6.1.1 ... not rely on applied standards in the workflow and software-choice of the user

The user is free to use any document format he prefers; however, if he wants the file’s textual contents to be indexed, the format has to be supported by Apache Tika. This has been proven in Simulation 5. The user also has the freedom to save the data wherever he wants; it is up to the persons responsible for the implementation of this system whether the connect that source to the system by adding the source to the crawler and writing a mapper. This has been proven in Simulation 3.

Note: of course the system is better able to index (semi-)structured files than unstructured files, meaning that for the user it can still be beneficial to follow some standards the mappers are aware of.

Since mappers can be created for each application, the user is also free to use the software he considers best for his purpose. This has been proven in Simulation 3.

Requirement satisfied.

6.1.2 ... work pro-actively, meaning the system has to do its job entirely on the background without demanding any action from the user

For testing purposes, the simulations did not run automatically via a cron job. However, it was invoked manually by executing a single command, which set all processes (data extraction, data mapping,
data linking, data import, data synchronization to ElasticSearch) in motion (sequentially):

```
python ./run.py
```

This applies to all simulations.

It is easy to have the command automatically and periodically executed as a cron job and the working of a cron job does not need any testing.

**Requirement satisfied**

### 6.1.3 ... accept all data formats and will try to read and index as much as possible

Already proven in section [6.1.1](#). The prototype accepts all formats, but if it cannot index its contents, it can still index its metadata and save the binary, making it usable for the user.

**Requirement satisfied**

### 6.1.4 ... be able to link data on various identifiers

Simulation 4 showed a link established via object id and via HLM range.

**Requirement satisfied**

### 6.1.5 ... be able to create dimension bridging (indirect) links

In simulation 4, a link from OL via OTL and GIS to planning has been established. It is an exact simulation of the process visualized in Figure 14 on page 29.

**Requirement satisfied**

### 6.1.6 ... map information from same-purpose applications to a uniform structure

This has been tested for changed field names in Simulation 6. Both different inputs were mapped to the same, uniform output.

**Requirement satisfied**

### 6.1.7 ... be easily extendible and adjustable

Simulation 3 is a test specifically for this requirement. The test produced the expected results.

**Requirement satisfied**
6.1 REQUIREMENT EVALUATION

6.1.8 ... guarantee a minimum accuracy when it comes to correctly identified links between data

Although the system automatically identifies links because of the keyword-based approach, it still misses links. Words can be typed differently or data can be unstructured that mappers cannot make up a link out of the context. Secondly, the prototype does not support NLP techniques, which could work very good because of all indexed content and added keywords.

Because the purpose of the system changes if it cannot guarantee a specific amount of links established, this requirement cannot be fulfilled. A potential solution for this is elaborated in Chapter 6 Recommendations.

Requirement not satisfied

6.1.9 ... be scalable

Scaling can be done horizontally and vertically. The traditional and easiest method is vertical scaling. With vertical scaling, one uses one machine and just upgrades its processor, memory or hard drive when necessary. This method however follows an exponential costs curve and has limits. Horizontal scaling is about using simple, cheap machines in clusters, over which the calculations can be spreaded using Map and Reduce algorithms (Dean & Ghemawat, 2008). Every modern cloud computing company, such as Google, Microsoft and Amazon, uses this method of scaling. The working of this process in visualized in Figure 29 (Appendix F).

Therefore this research considers horizontally scaling as the only way of responsible scaling. The system consists of three vital parts that need to be horizontally scalable for the entire system to be scalable:

- Processing part (crawlers and mappers). Sources can be individually crawled and mapped, making it possible to spread crawlers and mappers.

- Storage database. The prototype uses CouchDB, which is horizontally scalable (Apache Foundation, 2014).

- Search index. The prototype uses ElasticSearch, which is horizontally scalable (ElasticSearch BV, 2014).

The prototype is built accordingly. However, this does not necessarily mean that the entire system is scalable: there can still be a bottleneck in the connection between the parts. It is up to the implementer how to create this connection in a scalable way and this can not be tested in the scope of this simulation.

Requirement partially satisfied
6.1.10 ... allow companies within a consortium each use their own naming conventions

To allow each company within the consortium to use their own naming conventions, it has to be “translated” automatically within the system. This can be done via a “translation mapper”, which would function like the IFD Library [Bjørkhaug & Bell (2007)] or the Dutch version CB-NL [Bouw Informatie Raad (2014)]. These are sort of dictionaries with synonyms. Before importing the data into the database, this translation mapper finds the corresponding synonyms and adds them to they record. This way, the information can be found on all company’s naming conventions (if the dictionary contains a company’s naming conventions).

This would work with concept of multidimensional linking, making this requirements fulfilled since it has already successfully been tested in Simulation 4.

Requirement satisfied

6.1.11 ... work independently next to a company’s current BIM environment

The prototype does not require a change in a companies environment, it works as an independent addition to its software environment. Since it also does not require a change in a users workflow, current processes can continue as-is.

Requirement satisfied

6.1.12 ... work guaranteed with the software listed in section 3.3

However not all software exports and/or web services have been tested, the export possibilities of these software applications have been identified. To be sure the prototype supports an application, it needs to support two aspects. First of all, it needs to support the export method of the application. This can be a REST API or via the DMS. Second, it needs to support the output of an application, for example the file’s format.

It is not in the scope of this research to test if a REST API or DMS works – these are proven technologies.

All simulations used the input as described in the introduction of chapter 5 and the prototype proved in every simulation it was able to handle the input.

So, even the software’s web services have not been tested, literature [Maximilien et al. (2007)] and experience show that is should work fine.

Requirement satisfied
6.1.13  **Semantic web: data will become easily found**

The keyword-based approach from section 4.4.2 combined with a specialized text indexer (ElasticSearch) makes sure the information can be easily found. This has been proven with in simulations.

**Requirement satisfied**

6.1.14  **Semantic web: data will be reusable**

No data will be deleted, it uses revisioning for updates, see Simulation 2. Secondly, since all data is gathered in one database and can be queried (by source, author, project specifics etc.) it is easy to transfer data to the next project, making the data reusable. This is not simulated but can be done according to CouchDB’s wiki [Apache Foundation, 2013].

**Requirement satisfied**

6.1.15  **Semantic web: data will be interoperable**

In simulation 6 it has been proven that data with varying structures can be mapped to one uniform structure. This makes the data interoperable.

This is not a conversion tool for exporting and importing files between same-purpose applications. The mappers currently support one-way mapping and is created with the best effort idea in mind. This means that it can’t guarantee a 100% correct mapping with can be mapped back and forth.

**Requirement satisfied**

6.1.16  **Semantic web: devices will be ubiquitous**

The prototype’s frontend is web-based, making it available to all kinds of devices. For more information about the web interface of the prototype, see Appendix C.

**Requirement satisfied**

6.1.17  **Semantic web: systems will be flexible**

This is the same as the project-specific requirement in section 6.1.7

**Requirement satisfied**

6.1.18  **Semantic web: real-time**

Qualifying the system as being real-time concerns two aspects.

First, how long does it take for data to become available in the system after creation, and how long is considered real-time? One
could argue real-time is really real-time, but in practice a couple of seconds delay might also be considered real-time. The time it takes before new data becomes indexed depends on the amount of new data, the clusters’ power and the cron job’s interval. Since the set-up is horizontally scalable, the first two aspects negligible, because more computing power can easily be added if the amount of data grows. The last aspects can be configured to the wishes of the consortium, making them the one to decide how long real-time in practice is for them.

Second, when the data is in the system, is the data real-time retrievable? The use of ElasticSearch takes care of this. Because it indexes all data, searches are fast. Simulation 7 showed that searches take just a few milliseconds, making this part considered to be real-time.

**Requirement satisfied**

### 6.2 Hypotheses Evaluation

The null hypothesis from section 2.4 was:

\[ H_0: \text{A JSON-based data crawler cannot achieve linked data within software environments of large Dutch infrastructure without requiring predefined workflows.} \]

The successful simulations have proven that \( H_0 \) is rejected and therefore \( H_1 \) is supported, despite of the fact that not all requirements are satisfied. However, it might still be a good solution if the user would change his view and the system would get a slightly changed purpose. This is elaborated in Chapter 8 Conclusions.

Another reason why it still might be a good solution is the developments in the NLP area. This is elaborated in Chapter 9 Recommendations.
APPLICATION SCENARIOS

Assume the solution from this research fully functional in a live environment. The database stores a lot of JSON documents with lots of data that is linked to other data. This chapter describes some application scenarios this set-up could be used for.

7.1 SPECIFIC INFORMATION EXTRACTION

If a source exports structured data, such as Relatics used as an OTL, and the mapper is aware of this structure, the database can answer specific questions about the data. For example, consider a user who wants to know the weight of the lamppost with object type id ABC123. The database then can be asked "Give me the value of the property weight of a record with source Relatics and object type id ABC123. It would return – if a single result is found – a single value: the weight of the object. Of course, one could also directly find these values in Relatics. Therefore this only becomes useful if a user often wants to get specific information from multiple sources, and does not want, prefer or can continuously switch applications.

7.2 SEARCH ENGINE

A search engine is meant to quickly find information – not (necessarily) directly the right information, the user can browse through the results the search engine considers relevant. This is were it differs from the scenario described in previous section (Specific information extraction). It is really just a text based search engine, where an input field is provided for the search terms. Since the data can be unstructured and all types of data are aggregated, it is not useful to provide the search engine with specific input fields, e.g. for searches within a date or by author. The system is set up that it can answer these question just by a common keyword search. For example, if a mapper finds a date in the format dd-mm-yy it converts the date to a machine readable date and adds the day name (e.g. Saturday), the week number (e.g. w52), the quarter (e.g. 2014q4), etc. as keywords to the record. This makes searches on these criteria possible by keyword.
7.3 REVISIONING AND BACKUPS

In this set-up, no data will be overwritten. If some document has changed, it will be added as a new revision. This way, a user can easily check a file’s history and go back in time, if needed. It can also be used as a backup, since the database indexes data from another source, and thus all data is stored at two locations. CouchDB also supports automatic replication, making backups even more secure (data being stored at two different physical locations). Since all original data (binaries and API data) is stored, it should be possible to import the data back to the application, but this can depend on the application. Therefore, do not consider this system as a primary backup.

7.4 ANALYSES

Since all data is gathered in one database, the data can be easily analysed. Consider the situation that various sub-tasks experience delays, but you do not know the reason. An analysis of the planning tasks (and linked data) found a cross-reference between the tasks: they all contain welding and they all took place during high rainfall. This result can be used for case-based reasoning, see next section.

7.5 CASE-BASED REASONING

Consider the same example of previous section. It appeared that some tasks have experienced delays because of the high rainfall. Now a company can decide to investigate with this rainfall is causing delays. In the case of the combination with welding, a company can decide to use tents next time they need to weld to prevent these delays.

Note that the result of the analysis is quite obvious, but of course it can be more complex, consider a cause identified by multiple dimension bridged links.

7.6 EXCHANGE

As mentioned before, all data is gathered in one database and can be made accessible by a web interface. This makes the data accessible any time, anywhere. This is a great opportunity to give the client insight in the progress and other details of the project. However, the consortium probably does not want the client to see all information, so user access should be controlled.
7.7 CUSTOM DASHBOARDS

As explained in the simulation, a WPA contains information from various sources. It might be desirable to neatly and clearly show this information in one custom dashboard, designed by the user. Since all data is stored in one database, building such a dashboard is a quite simple task, giving the user the freedom to design it according to its needs. In Figure 28, an example of such a dashboard is shown.

Figure 28: An impression of a potential custom dashboard
CONCLUSIONS

The necessity of change within BIM nD developments is clear. Haphazardly adding D’s – because people want the theoretical possibilities – pays (and will pay even more) its price. This is evidenced by the current problems with BIM and the (partial) solutions and alternatives that are being developed. Standardization required to use BIM appeared to be hard to acquire, making it unwise to continue this way for extending BIM. However, this research acknowledges the added value by BIM 4D or BIM Visualize.

The theoretical set-up is designed to meet all drawn up requirements for BIM Inform. However, the prototype created according this design fails to fulfil one requirement: the guarantee that a specific percentage of the data will be linked properly. However, \( H_0 \) is still rejected and as a consequence \( H_1 \) is supported.

However, the fact that the prototype does not meet this single requirement, does not necessarily mean that the set-up will not meet this requirement too. The prototype lacks NLP technologies, which might solve the issue. Secondly, the degree of linking accuracy has a direct effect on the main role of the system: with a high accuracy the system can be considered a full replacement for BIM 5D+, with an unknown, lower accuracy it can be seen as a helpful tool to quickly find the right information in most cases.

Taking these two assumptions into account, the solution can satisfy. Acceptance and implementation of this solution increases developments, increasing the accuracy and overall functionality.

The research question of this research is:

Can a vendor and technology independent indexer provide data linking and retrieval in EAM software in a less standard-requiring way?

It is safe to say it technically can, although it depends on the consortium’s perspective. Does it expect a helpful tool or a full replacement? In the first case, the answer is yes. In the second case, the answer currently is no, but it can become yes in the future, see the next chapter.

The main difference of this set-up with others is that it has chosen usability over technical possibilities. Yes, technically seen a lot is possible, but due to restrictions in practice (mostly caused by humans)
it is often not the way to go. BIM nD needs a lot of standardization to acquire its goals, which are unreachable in practice. This research set slightly different goals and was therefore able to require less standardization in workflows.

To conclude: this research has proven there is an other way to go. It is the way of imposing (and needing) fewer standards, making it more enjoyable and efficient for the user to work, probably resulting in a more productive and efficient working environment.

Which way to go cannot be answered, but this research has proven the technology is no barrier to go this way with BIM Inform. It is a promising way that is not obstacle-free, but still contains way less obstacles than BIM nD’s current path.
RECOMMENDATIONS

Companies or consortia who recognize the problems from section 2.2 Problem Analysis can definitely be given the advice to look for alternatives for their current situation.

The solution of this research is one way. If a consortium decides to go this way, the best method is to develop it internally according to its needs. The flexible set-up allows a consortium to make modifications for its next project, so it does not have to be afraid to design the solution according to the needs of a specific project. It is advisable to develop with Scrum[1] techniques. This ensures a solid process with continuous deployments of fully functional software, piece by piece. The fact that from the beginning it might be unclear what the solution exactly will look like, makes it necessary to be able to continuously receive feedback from the user. This feedback allows the developers to decide how to continue and how to prioritize. Scrum is developed for such developing processes.

Although the final result might be a bit unclear and depends on a consortium’s needs, some aspects the solution definitely benefits from, are the following.

First, the solution does not include security and user permissions. Indexed data is for everyone available but this might not be preferred; a financial manager has access to financial documents, but an engineer might not. Handling user permissions well is a complex task that might endanger this solution since it is based on gathering and indexing all data. User permissions have to be imported properly to ensure the same security the application maintains.

Second, it should be investigated to what extend NLP technologies can help better identification of links between data. Implementing NLP technologies will improve the machine’s understanding of data, which probable will make it better in understanding relations between that data. The fact that the solutions uses the keyword-based approach is very useful for this automatic relationship identification. Note that the semantic web is very good at adding semantics to data and defining relationships a machine can understand.

[1] Scrum is an iterative and incremental agile software development framework for managing product development.
Third, it might be wise to investigate source-specific converters. Users do not want to save documents in a specific way in order to make it indexable by the system. If the system disposes of the right converter, this is not necessary. If it does not, please do some research, maybe the converter already exists. This research stumbled upon an IFC to JSON converter ASSIMP2JSON (Gessler 2014), making it possible to read and index entire IFC files.

Fourth, since this research’s simulation purposes were only to test the technical theory of the solution, no test cases with real users and data have been conducted. Before developing this solution for production use, it is wise to conduct these tests first and evaluate the results. This research has not investigated anything about user acceptance, however literature showed that less-standard demanding processes are more favourable to the user (so he can do what he considers best).

The last, but not least, recommendation is to always follow developments in the BIM nD area and developments at alternatives. In the end everybody can learn from eachother and it would be wise to do so.
REFERENCES


References


APPENDIX A: PROTOTYPE REQUIREMENTS

The prototype is written Python 2 and uses the following Python packages:

- couchdb-python==0.10
- Werkzeug==0.9.6
- argparse==1.2.1
- arrow==0.4.4
- beautifulsoup4==4.3.2
- chromelogger==0.4.1
- dicttoxml==1.5.6
- ecdsa==0.11
-jsonpickle==0.8.0
- lxml==3.4.1
- paramiko==1.15.1
- pycrypto==2.6.1
- python-dateutil==2.2
- six==1.8.0
- sqlparse==0.1.13
- wsgiref==0.1.2
- xlrd==0.9.3
- xmltodict==0.4.1
- xmldict==0.9.0

The packages are installable with pip.

Next to these packages it uses the following (open-source) software:

- Ubuntu Server 14.04 LTS, ubuntu.com
• ElasticSearch 1.3.5, elasticsearch.org
• Apache Tika 1.7, tika.apache.org
• CouchDB 1.5, couchdb.apache.org
• elasticsearch-river-couchdb, https://github.com/elasticsearch/elasticsearch-river-couchdb
APPENDIX B: PROTOTYPE SOURCE CODE

Because it is not useful and not efficient to put the actual source code in this document, it has been made available for download until 1 March 2015 via the following URL:

https://onedrive.live.com/redir?resid=D03E94CC000016E3!36009&authkey=!ADv0fLrjW67w&ithint=file%2czip

It includes the test content for the simulation.
APPENDIX C: SIMULATION FRONTEND

The frontend is based on an existing user Interface called Calaca (https://github.com/romansanchez/Calaca). It is a user interface based on AngularJS, especially designed for easy searches in Elastic-Search indexes.

The prototype is fully functional available on the URL below (until 1 March 2015):

http://research.peterwiggers.nl/

Username: peter
Password: theotherway
APPENDICES

APPENDIX D: APACHE Tika CONFIGURATION

Apache Tika handles two thins: it extracts metadata from a file and it tries to convert a file’s contents to plain text.

The importer requests the Apache Tika instances, which are running in server mode as follows:

For contents in plain text (at port 1234):

```bash
java −jar ./tika−app−1.6.jar −server 1234 −r
```

For metadata in JSON (at port 1235):

```bash
java −jar ./tika−app−1.6.jar −server 1235 −j
```
APPENDICES

APPENDIX E: ELASTICSEARCH CONFIGURATION

The ElasticSearch service can be started by simply executing the following binary (in the ElasticSearch directory):

```
./bin/elasticsearch
```

It has a automatic, live connection with CouchDB, made possible by the elasticsearch-river-couchdb plugin. The following HTTP PUT requests establishes the live connection:

```
curl -XPUT 'localhost:9200/_river/my_db/_meta' -d '{
  "type" : "couchdb",
  "couchdb" : {
    "host" : "localhost",
    "port" : 5984,
    "db" : "my_db",
    "filter" : null
  },
  "index" : {
    "index" : "my_db",
    "type" : "my_db",
    "bulk_size" : "100",
    "bulk_timeout" : "10ms"
  }
}
'
APPENDIX F: MAPREDUCE AND SCALING

RDBMS

Situation 1, 1GB of data  
Situation 2, 1000GB of data

NoSQL / BigQuery

Situation 1, 1GB of data  
Situation 2, 1000GB of data

Figure 29: Vertical scaling vs. horizontal scaling, made possible with Map and Reduce