Identifying Shared Functionality using Latent Semantic Indexing

Master's Thesis

Dennis Learbuch

Identifying Shared Functionality using Latent Semantic Indexing

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Dennis Learbuch
born in Leidschendam, the Netherlands

Software Engineering Research Group
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Getronics PinkRoccade
Staalmeesterslaan 410
Amsterdam, the Netherlands
www.getronicspinkroccade.nl

Identifying Shared Functionality using Latent Semantic Indexing

Author:	Dennis Learbuch
Student id:	1150278
Email:	D.L.R.Learbuch@student.tudelft.nl

Abstract

There are numerous systems of which parts, at least in the minds of designers, have the same functionality. If the degree of this shared functionality can be measured some how, it would prove to be very valuable. This information can be used to modularize these systems and improve their maintainability. To identify this functionality a technique known as Latent Semantic Indexing (LSI), an information retrieval technique, is proposed. We applied LSI to two systems at Getronics PinkRoccade. Our main conclusion is that we can not recover 100% of the shared functionality automatically, however, the search space can be reduced. This means the experts, recovering the shared functionality, have less documents to evaluate.

Preface

I would like to thank the following people who helped and supported me in writing this thesis.

Gerd Gross for his guidance and advice on how to structure the thesis, he made sure quality came before quantity. Leon Moonen for his supervision during the first part of the project, after which he left for Scandinavia. Marco Lormans for his advice on how to best apply LSI in this context. Arie van Deursen for his oversight, and his ideas during a few preliminary brainstorming sessions.

At Getronics PinkRoccade, I would like to thank the following people. Jasper Sman for his day to day support in organizing the activities concerning the project, and for his feedback on my ideas. Jelle Gerbrandy for his critical questions, feedback and for guarding the scientific aspect of the work. Theo Thijssen for his organizational guidance and feedback on my work. Ad Vlaming for his advice in the bi-weekly sessions, discussing the progress of the project. Arthur Schafgans and Addie Tange for providing insight into WWO. Rolf van der Meer, Toos Vries, Christia Scholten and Ron ten Have for bringing the context to life by sharing their experience. Bert van der Velden, John van Heijningen, Henk Johan Kwakkel, Ruud Fennis and Peter Maas for helping me to create the expert traceability matrix.

I would also like to thank my parents, sister and friends, for their support.

Dennis Learbuch
Zoetermeer, the Netherlands
August 26, 2008

Contents

List of Figures

Chapter 1

Introduction

The project that this document describes is concerned with applying a software engineering technique. Such techniques are aimed at either improving the quality of software or decreasing the costs concerning software. In our case we want to decrease the costs by saving effort on maintenance by applying architectural reorganization to two monolithic systems and transforming them into component-based systems. This will be achieved by extracting shared functionality into reusable components. The idea behind this is, after the extraction, the shared functionality will only be present in one component rather than in two, and thus reducing the costs. The general expectation of applying architectural reorganisation by using reusable components, is that the maintainability is improved [20]. Our work is focused on a component based on gross-net calculations present in both systems. To be able to create a reusable component, the parts of the gross-net calculations which are actually, shared have to be identified. Traceability links are connections between, for example, requirements and source code, with the goal of tracing these requirements. In this case, traceability links between the calculations of both systems will be used to identify the shared functionality. In order to reverse engineer these links, an approach known as Latent Semantic Indexing (LSI) is applied.

LSI is an Information Retrieval (IR) technique which has several improvements over regular IR, like recovering that a word can have a different meaning in a different context. There are a lot of examples [11, 10, 21, 11, 4, 26, 29] where LSI is used for recovering traceability links in software engineering. The subject of the analysis by LSI is functional design documentation. In one of the systems, the documentation consists, for the most part, of pseudo code. LSI achieves its results by uncovering a latent semantic structure, it does this by finding statistical relations. This structure is present in a higher concentration in free text documentation than in source code [21], which is the reason why the functional design documentation is chosen as input.

The above described motivation and work, are the result of a project that was done at Getronics PinkRoccade, the project involved two systems and a certain amount of shared functionality in the form of gross-net calculations. These calculations are to be extracted into a reusable component. These components are intended for a bus-like architecture, this new architecture is according to a customer requirement.

1.1 Hypothesis

The context described above resulted in the following hypothesis:

> *LSI can be used to identify shared functionality between two systems on the basis of structured documentation.*

1.2 Research Questions

On the basis of the hypothesis above, the following research questions are identified, and are to be addressed in this thesis:

- How close are our results, the recall and precision, to previous work done in applying the LSI?
- Do our results prove that LSI can be applied to concepts of two different systems?
- What do the results say with respect to structured documentation, are they influenced in a positive or a negative way?
- Can LSI reduce the amount of effort needed to find shared functionality?

To research the validity of the hypothesis and to answer the research questions, a proof of concept has been set up. The work has been structured by a methodology known as MAREV, as described in [22], along with some additions and adjustments to suit our particular case. The results of our work are compared to those of previous work to check the quality relative to different contexts. By comparing the results, something can also be said with respect to validity of applying the technique to two systems. The observations that were made on how the specific input, structured documentation, effects the results, will say something on whether using this type of documentation was a good idea, or a bad one.

1.3 Thesis Organisation

The thesis is set up by order of abstraction, first the related work is presented to give the reader an idea of what LSI is used for in software engineering. After this, a theoretical background is given in the Chapter technologies, by describing reverse engineering and latent semantic indexing. After this is a Chapter on the method that was applied to the two systems described above, which is directly followed by the description of the actual work. The results from this work are discussed and interpreted in the next Chapter. Finally, the most important conclusions, along with their motivations, the results, which can be seen as the contributions of this document, and a selection of ideas for future work on the subject are presented.

Chapter 2

Related Work

In this Chapter the most used applications of LSI in the domain of software engineering are discussed. The main application of LSI is to recover traceability links, several examples of this application are presented, of these examples, the context in which it was applied are presented to give the reader a frame of reference with respect LSI. Since LSI is a specialized IR technique, some IR applications concerning traceability links are incorporated as well.

The work in this thesis draws heavily upon the work done by Lormans et el in [22, 21]. Their main contributions are the following. The method MAREV, which is used for automating the process of reconstructing traceability links and generating requirements views. A new two-dimensional vector filter strategy for selecting traceability links from an LSI generated similarity matrix. And they applied their approach to three case studies, one of which is an industrial strength case in the consumer electronics domain.

Since we are trying to find traceability links, several traceability recovery techniques are discussed below. Some are used to find relations between documentation and source code, others to map requirements between different levels of abstraction. Most literature on LSI, is research on recovering traceability links between free text documentation and source code [11, 10, 21, 4, 26, 29].

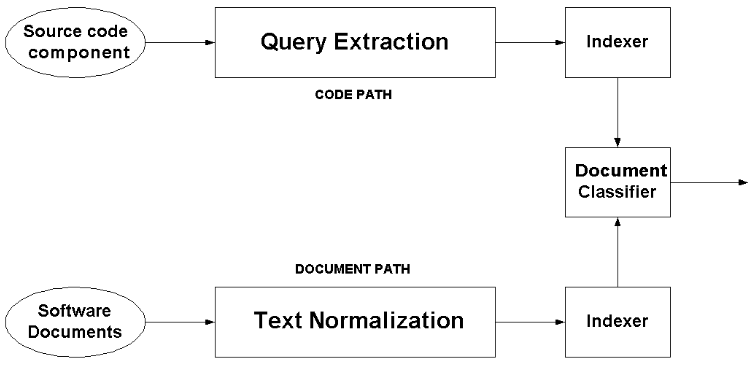


Figure 2.1: Traceability Link Recovery Method [4].

The process of LSI as applied to recovering traceability links between documentation and source code is depicted in Figure 2.1, it shows two paths, one to extract information from the source into queries and the other to prepare the documentation for retrieval. The text of the input documents is processed, this done by applying typical IR steps; lexical analysis, stop word elimination, stemming, index-term selection and index construction [21]. When all steps of the LSI process are executed, the similarities between the documents and the queries are calculated. The result is a list with ranks of documents for each source code component.

Antoniol et al. use information retrieval (IR) methods to recover the traceability relations from C++ code onto manual pages and from Java code to requirements [4]. Marcus and Maletic, and Di Penta et al. use latent semantic indexing for recovering the traceability relations between source code and documentation [30, 29, 14]. The IR methods in these cases are mostly applied to reverse engineer traceability links between source code and documentation in legacy systems.

De Lucia et al. present an artefact management system, which has been extended with traceability recovery features [10, 11]. This system manages all different artefacts produced during development such as requirements, designs, test cases, and source code modules. De Lucia et al. also use LSI for recovering the traceability links. In [12], they improved their traceability recovery process and propose an incremental approach.

Research conducted in applying information retrieval methods to source code and associated documentation is related to indexing reusable components [17, 19, 23, 24]. In [23, 24] Maarek uses an IR approach to construct software libraries automatically. IR techniques are researched because of their advantages over parsing and semantic analysis of natural text, the results might not be as accurate, but the process is relatively cheap to apply.

Another application of LSI is to use the similarity values between source elements to identify clusters. Marcus and Maletic [25, 27, 28] used clustering to help with the identification of abstract data types in procedural code and the identification of concept clones. They also used this information to asses the cohesion of components. To use IR methods for supporting the traceability recovery process, Antoniol et al. [2, 3, 5] used a probalistic method and the vector space model to recover links between source code and documentation, and between source code and requirements.

Antoniol et al. [4] see that establishing traceability links can be helpful in several activities concerning the maintenance cycle of a software system:

- **Program comprehension**. Be it, a top-down or bottom-up approach, traceability links between areas of code and related sections of free text documents, such as an application domain handbook, a specification document, a set of design documents, or manual pages, aid both types of program comprehension.

- **Maintenance**. Traceability links between code and other sources of information are helpful in performing analysis on documentation of different levels of abstraction.

- **Requirement tracing**. Traceability links between the requirements and source code provide clues to what areas of code contribute to implement specific functionality

[9, 32, 33]. They are helpful to assess the completeness of an implementation with respect to stated requirements and to devise complete and comprehensive test cases.

- **Impact analyses**. A major goal of impact analysis is the identification of the products affected by a proposed change [6], this is what traceability links do. For example, when a certain piece of functionality is altered, traceability links can provide a listing of a elements effected by this change.

- **Reuse of existing software**. Means to trace code to free text documents can help to locate reuse-candidate components. Often software has not been produced with reuse in mind, this means knowledge about its functionality is usually spread out, including requirement specification documents, manual pages, design documents, and the code itself.

Although, the work presented in this document does not map requirements or source code on documentation within one system, it still builds on the results described in this Chapter.

Chapter 3

Required Technologies for Identifying Shared Functionality

Unlike conventional reverse engineering, the process of recovering traceability links from documentation cannot simply be done by using compiler techniques, since it is difficult to apply syntactic analyses to natural language sentences. Instead an approach known as Latent Semantic Indexing (LSI), which is an Information Retrieval (IR) [8, 35] technique, is used.

Because the work done in this document is a reverse engineering exercise, the aspects that are related to reverse engineering are explained first. The technology LSI is explained in the next section, to give the reader an idea how it works, for when it is applied in the method described in the next Chapter.

3.1 Reverse Engineering

In [31] they state that: *the goal of reverse engineering is the identification and recovery of the design artefacts of a system, such as its requirements, specifications and architectures.* In work we have done, reverse engineering is present in the following aspects. There is an artefact we want to identify, this artefact is a traceability link matrix, which shows the information represented in Figure 3.1. In this diagram, A and B represent the functionality of the two systems. The intersection of the gross-net calculation functionality is marked by C, this is the part we are interested in and represents the information that is to be captured in the traceability link matrix.

Below a general approach for solving reverse engineering problems is presented, this approach is used by most reverse engineering tools and consists of three phases:

- **Extraction:** This phase always involves the collection of information from artefacts such as the source code, documentation, configuration files and makefiles. Runtime information can also be important, by using dynamic analysis, this information can be extracted as well. This extracted information is then stored in a repository.

- **Abstraction:** By querying and shaping this repository, new information can be ob-

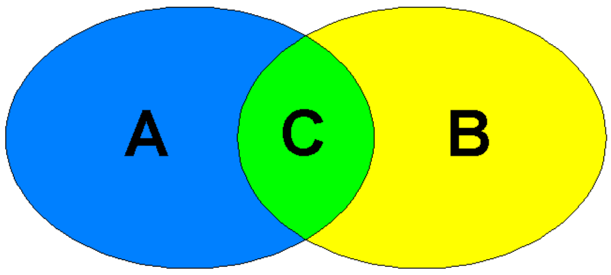


Figure 3.1: Two systems and their overlap

tained on the system. These results are, again, stored in a repository, so that one can iteratively apply this process.

- **Presentation:** In this phase the information gathered in the previous phases is represented in a clear fashion. The presentation format differs according to the purpose.

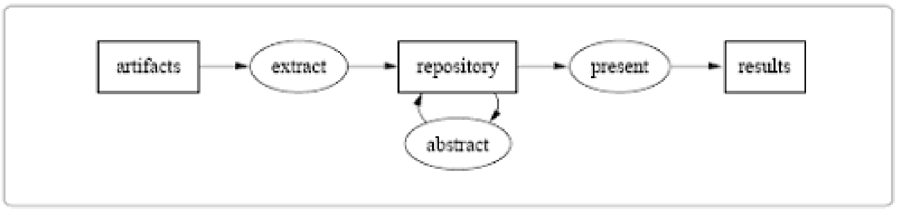


Figure 3.2: The layout of a general reverse engineering tool [31]

In our case, the extraction involves the acquirement of the relevant documentation, which is the functional design documentation of both systems with respect to the gross-net calculations. The abstraction is done by applying LSI to the extracted documentation. Finally, in the presentation phase the results of the abstraction are processed in such a way that the recall and precision can be calculated, on the basis of these results, interpretations can be made and conclusions drawn.

3.2 Latent Semantic Indexing

Latent Semantic Indexing (LSI) [13, 15] is a technique based on information retrieval. The key feature of LSI is that it assumes the presence of a latent semantic structure in its input, the input can be source code as well as free text documentation, or a combination of the two. A motivation for trying to find a latent semantic structure are IR issues such as synonymy and polysemy.

- Synonymy refers to a situation where the same object has different names. Users from different backgrounds or with different needs, knowledge or linguistic habits will describe the same information by using different terms.
- Polysemy refers to the phenomenon that most words have more than one meaning. This depends on the context the terms are used in [13].

An estimate of the latent structure, that is assumed to be present in the documents, is recovered using statistical techniques. This results in a representation of the relations between the terms and documents, this representation is then used for displaying and retrieving information. In this manner LSI overcomes, to some degree, the difficulties caused by synonymy and polysemy.

The process of LSI begins by applying Singular Value Decomposition (SVD) to the term by document matrix to derive the latent semantic structure model [8, 35]. The term by document matrix, which is a rectangular matrix for example, $t \times d$, can be decomposed into the product of three other matrices:

$$X = T_0 S_0 D_0^T$$

such that T_0 and D_0 have orthonormal columns and S_0 is diagonal. This is called singular value decomposition of X. T_0 and D_0 are matrices of left and right singular vectors and S_0 is the diagonal of singular values.

SVD allows for a simple strategy to facilitate the optimal approximate for smaller matrices. If the singular values in S_0 are ordered by size, the first k largest values may be kept and the remaining smaller ones set to zero. The product of the resulting matrices is a matrix X' which is only approximately equal to X, and is of rank k. Since zeros were introduced into S_0, the representation can be simplified by deleting the zero rows and columns of S_0 to obtain a new diagonal matrix S, and deleting the corresponding columns of T_0 and D_0 to obtain T and D respectively. The result is a reduced model:

$$X' = TSD^T \approx X$$

which is the rank-k model with the best possible least square fit to X [13]. This is how dimension reduction is applied in LSI.

When the matrix is created it can be compared with a vector, thus calculating a degree of similarity. One has to create a query which will be converted into a vector, this vector is then mapped onto the matrix, the result will be a rate of similarity between the query and the documents on which the matrix is based. The similarity metric is the cosine between the corresponding vector representations, the value of this metric lies between [-1,1], in which 1 indicated (almost) identical. The measures can be used to cluster similar documents, or identifying traceability links between documents [22].

When compared with other IR techniques [16, 34], LSI has several advantages over these approaches. One main advantage of LSI is it takes into account synonyms, another advantage is fact that LSI considers word order, syntactic relations and morphology [29]. In same way as other IR approaches, LSI does not use a grammar or vocabulary, this allows the technique to be used without large amounts of preprocessing, which significantly reduces the cost of traceability link recovery [10, 26]. LSI also has a prerequisite, programmers have to use meaningful names for program items [4].

Our work is different to the other applications of LSI described in the related work, in which LSI is used to recover traceability links between documents of the same system. In our work, documents of different systems are compared with the objective of finding shared functionality.

Chapter 4

Method

The work done in this project builds on that in [22] and solves one of their research issues, where they question whether the recall and precision that are achieved are sufficient for practical purposes. Our proof of concept was executed by using an approach inspired by the Methodology for Automating Requirements Evolution using Views (MAREV) they presented. Although, our goal is not to automate requirements evolution using views, the method, with some refinements to make it conform to our context, and its results, suits our purpose. Our work is different from theirs in several aspects. We have applied this method to reconstruct traceability links by using LSI, our context consists of two systems and our goal is to identify shared functionality, which is different from other applications of LSI [22], where the context involves only one system at a time.

In our context no traceability matrix created by experts, which is needed for the validation, was available, this meant one had to be created, section 4.2 discusses this. The final section describes how the traceability links of LSI were validated by comparing them with those of the experts.

4.1 The Steps

The following steps are executed in the MAREV [22] method to reconstruct traceability links:

1. Defining the underlying traceability model
2. Identifying the concepts from the traceability model in the available set of documents
3. Preprocessing the documents for automated analysis
4. Reconstructing the traceability links
5. Selecting the relevant links

4.1.1 Traceability Model Definition

To reconstruct traceability links, first the links that are to be recovered have to be established. The diagram 4.1 below shows the links we want to reconstruct, these links are represented by the connection between the desired functionality of both systems. The arrows from systems A and B, shows where the functionality originates from.

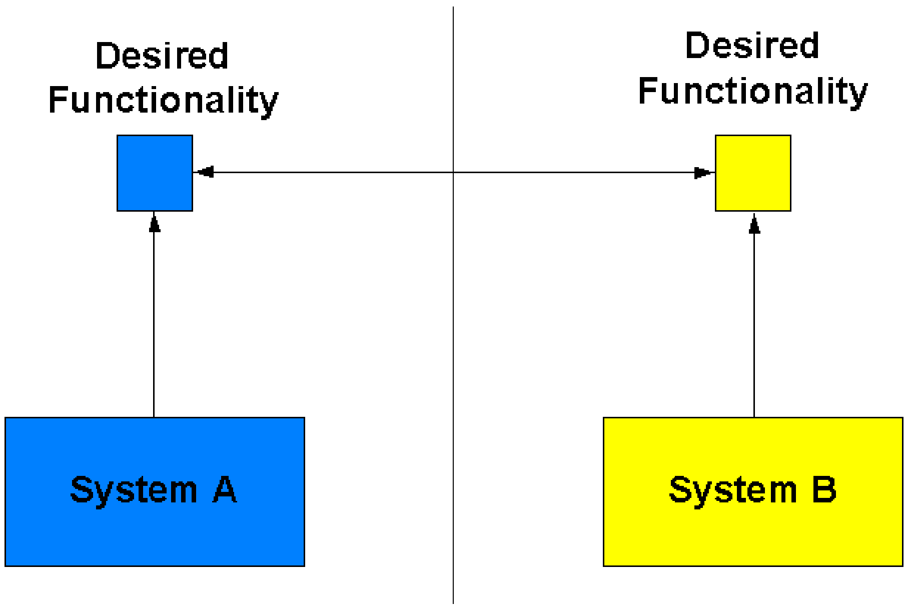


Figure 4.1: Mapping functionality

4.1.2 Concept Identification

Once the traceability model has been established, the concepts contained in the entities of this model are to be identified in the documentation. Since most documentation is not structured by using templates such as MIL-std 498 or IEEE-1233-1998 [22], identifying the concepts will require a certain amount of manual work. The templates structure the documents in such a way that it will be easier to distinguish concepts.

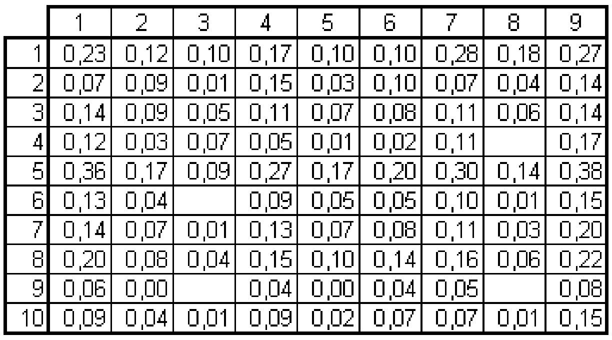
4.1.3 Text Processing

When the concepts are defined, the pre-processing of the text begins. First, the texts are exported into separate files, this can be done by hand, as in our case, or by using a script. The possibility for scripting depends on the structure of the documentation. The next step involves typical IR steps like lexical analysis, stop elimination, stemming, index-term selection and index construction. LSI will create a term-by-document matrix from the processed concepts.

4.1.4 Link Reconstruction

The term-by-document matrix from the previous section is used to reconstruct the traceability links by applying LSI. First, dimension reduction is applied by creating the rank-k model, which is used to calculate the similarities between the concepts. From these similarities a matrix is constructed, containing the similarities between all concepts.

Figure 4.2 shows an example of a similarity matrix.



	1	2	3	4	5	6	7	8	9
1	0,23	0,12	0,10	0,17	0,10	0,10	0,28	0,18	0,27
2	0,07	0,09	0,01	0,15	0,03	0,10	0,07	0,04	0,14
3	0,14	0,09	0,05	0,11	0,07	0,08	0,11	0,06	0,14
4	0,12	0,03	0,07	0,05	0,01	0,02	0,11		0,17
5	0,36	0,17	0,09	0,27	0,17	0,20	0,30	0,14	0,38
6	0,13	0,04		0,09	0,05	0,05	0,10	0,01	0,15
7	0,14	0,07	0,01	0,13	0,07	0,08	0,11	0,03	0,20
8	0,20	0,08	0,04	0,15	0,10	0,14	0,16	0,06	0,22
9	0,06	0,00		0,04	0,00	0,04	0,05		0,08
10	0,09	0,04	0,01	0,09	0,02	0,07	0,07	0,01	0,15

Figure 4.2: Example of an Similarity Matrix

4.1.5 Filtering

The similarity matrix resulting from the previous step, has to be evaluated. The reliability of every similarity value is to be determined. In [4, 10, 29] a number of strategies are described to evaluate the values, they are: the cut point, cut percentage, constant threshold, variable threshold strategy and scale threshold. We have chosen to apply to two filtering strategies row relative and column relative filtering.

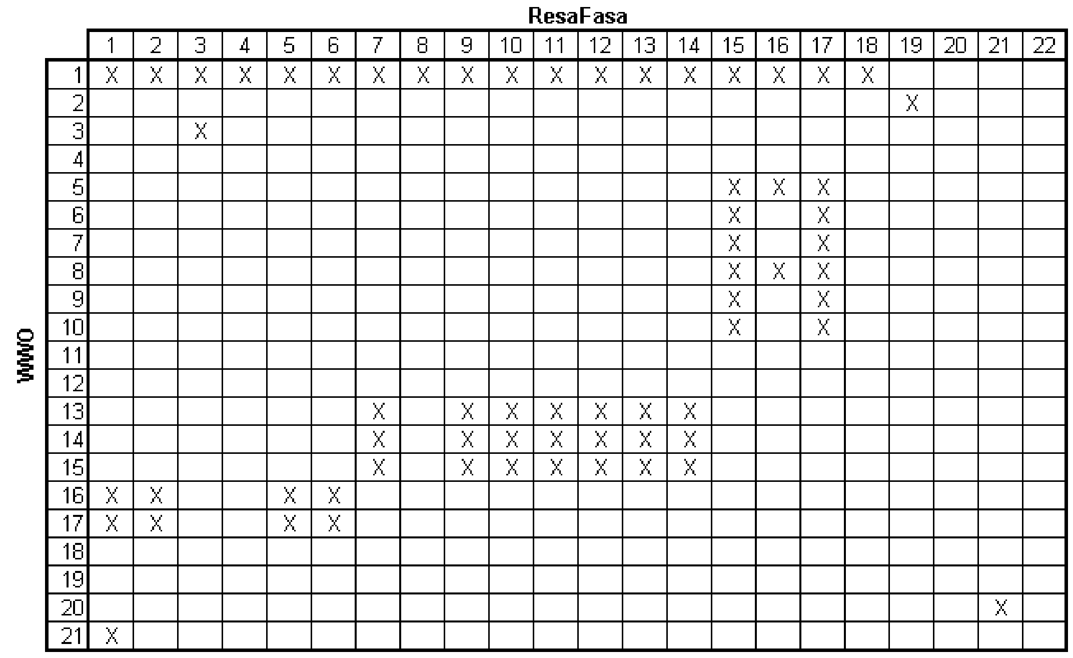
The row and column filtering strategies are similar to the one dimensional vector filter strategy presented in [22]. The maximum of a row or column is taken and every value that is within a certain percentage, the threshold q, of the maximum remains in the matrix the rest of the values is filtered out.

The two-dimensional filtering from [22] was not applied to our work, on recommendation of one of the authors, this because the similarities in our context, when looking at the entire matrix, have a relatively large spread, compared to the spread of the columns and rows. In other words, when comparing every value to all values in the matrix, a lot of relevant links would not have been reconstructed.

4.2 Expert Traceability Matrix

To be able to validate the results achieved by LSI, they need to be compared to the actual (or an approximation there of) traceability links. This aspect is only required for validation of the technology. In a typical application, the creation of a traceability matrix by experts is not necessary. If such information is part of the documentation, this is easy. However in most cases, like ours, there is not one available. When this is the case, they have to be created. Experts on the subject can help to supply this information, if there are no experts available, one has to create a traceability matrix their self. In our case, a bit of creativity was involved to recover the ”correct” traceability links from the experts.

Figure 4.3 shows a example of a traceability matrix created by experts, where the ”X” represents a link.



		ResaFasa																					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				
	2																			X			
	3			X																			
	4																						
	5															X	X	X					
	6															X		X					
	7															X		X					
	8															X	X	X					
	9															X		X					
	10															X		X					
WWO	11																						
	12																						
	13							X		X	X	X	X	X	X								
	14							X		X	X	X	X	X	X								
	15							X		X	X	X	X	X	X								
	16	X	X			X	X																
	17	X	X			X	X																
	18																						
	19																						
	20																					X	
	21	X																					

Figure 4.3: Example of an Expert Traceability Matrix

4.3 Validation

The results, achieved by applying LSI, are validated by comparing them to the traceability links created by the experts. This comparison will give information on the reliability and accuracy of the results of LSI. Two well known IR metrics, recall and precision [7, 18, 36, 35] are used to compare the links:

$$recall = \frac{|correct \cap retrieved|}{|correct|}$$

$$precision = \frac{|correct \cap retrieved|}{|retrieved|}$$

The term correct stands for the number of traceability links recovered by the experts, and retrieved, for those recovered by LSI. The values of recall and precision range from [0,1]. A precision of 1 means all links LSI found are correct, but there can be correct links that were not recovered. A recall of 1 implies all correct links were found, there can be incorrect links though. The way of filtering the links, the type of strategy and the parameters, influences these metrics as can be seen in the next Chapter.

Chapter 5

Proof of Concept

This chapter describes the proof of concept, the work done to support hypothesis stated in the Introduction: *LSI can be used to identify shared functionality between two systems on the basis of structured documentation*. The work was done on two systems, in both of which gross-net calculations are present, the objective of the work is to identify what steps of these calculations are done in both systems, with the goal to extract that functionality into a single reusable component.

First, the two systems are discussed to give the reader an idea of the domain pertaining the proof of concept, the size, age, function and programming language show what kind systems they are. The second section explains what tools were used to produce the results and the necessary processing. In the next section the steps of the process of applying LSI to the case are described in detail. After this the results are discussed. The fore last section explains and interprets the results achieved by applying the process. The final section discusses what can be done to improve the process in this context.

5.1 Context of the Work

The work done here entails two social security systems used by the Dutch social security agency (UWV), these systems were created and are currently maintained by the ***BAS - CAM*** division of ***Getronics PinkRoccade***. Each system is responsible for several social security laws. Below a short description of the systems WWO and ResaFasa is given along with a table of attributes, to give an idea of the domain of the work presented in this Chapter. As can be seen from the table 5.1, these very large legacy systems. Both systems have more than a million lines of code, the complexity is given by the number of function points [1] and the fact that they have been written in COBOL and running since the early nineties further emphasises the legacy aspect.

- **WWO**, this stands for "Werkloosheids Wet Ontslag". The beneficiaries for this system are those people who have become unemployed by means of dismissal, they are subject to the Unemployment Insurance Act law (WW). This system determines the amount the beneficiary will receive, this information is recorded and is eventually passed on, so the beneficiary is paid.

- **ResaFasa**, which stands for ”Registratie Financiele Afhandeling”. In principle ResaFasa does the same thing as WWO, be it in a different context. The beneficiaries in this case are people who are unable to work, the law responsible for them is the Sickness Benefits Act (WAO).

Attributes	**WWO**	**Resa/Fasa**
#MLOC	1,2	2
Function Points	4954	3534
Programming language	COBOL	COBOL
Database	Oracle RDB	Network DBMS
Interface	DECforms	DECforms
Operating system	OpenVMS	OpenVMS
Platform	Alpha GS160	Alpha GS160
Year of launch	1993	1991

Table 5.1: Short description of systems

5.2 Tools

During the proof of concept, two tools were used. The TMG matlab toolbox for applying LSI and MS Excel for processing the results from TMG and some other presentation purposes.

- **The Text to Matrix Generator** [37] (TMG) is a MATLAB Toolbox, it facilitates various Data Mining (DM) and Information Retrieval tasks. TMG is composed of five modules, these are shown in diagram 5.1 below.

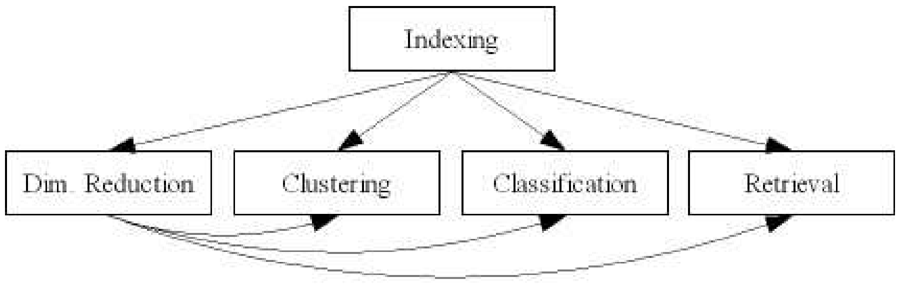


Figure 5.1: Structure and dependencies of modules of TMG [37]

From the five modules only the indexing and the retrieval were used extensively, and the dimension reduction on occasion. The indexing module takes the input and transforms it into a terms by document matrix. The retrieval module maps a query onto this matrix, which results in measures of similarity between the query and the input documents. The tool does this by applying lexical analysis, stop word elimination, stemming, index-term selection and index construction.

- **Microsoft Excel**, was extensively used to process the results of the TMG in order to create the similarity matrices. Excel was also used to filter these similarities and calculate the recall and precision. The experts used Excel as well, to fill out their traceability matrices. Some of the work done with the help of excel, can be further automated to reduce the amount of manual labour by writing a tool or a selection of scripts that facilitate this process.

5.3 The Process

This section gives a step by step description of the process which eventually lead to the results presented at the end of this Chapter. The subsections below describe and present all the intermediate results, along with an explanation of how they were achieved. First, the work involved with obtaining the input is discussed, followed by a description of how the input was prepared for processing, next up the actual processing, how the expert traceability matrix was created is the subject of the next subsection and finally the results of LSI are compared with the expert traceability matrix.

5.3.1 Obtaining the input

To obtain the input, first one has to establish what is needed and what is available, on the basis of this, the corpus and the nature of queries is established. The information needed in this case is functional design documentation regarding the gross-net module of both systems. Getting a hold of this documentation turned out to be relatively easy in this case, since the location of the functionality is known. Because the goal is to compare the systems, the queries will consist of the documents of the other system.

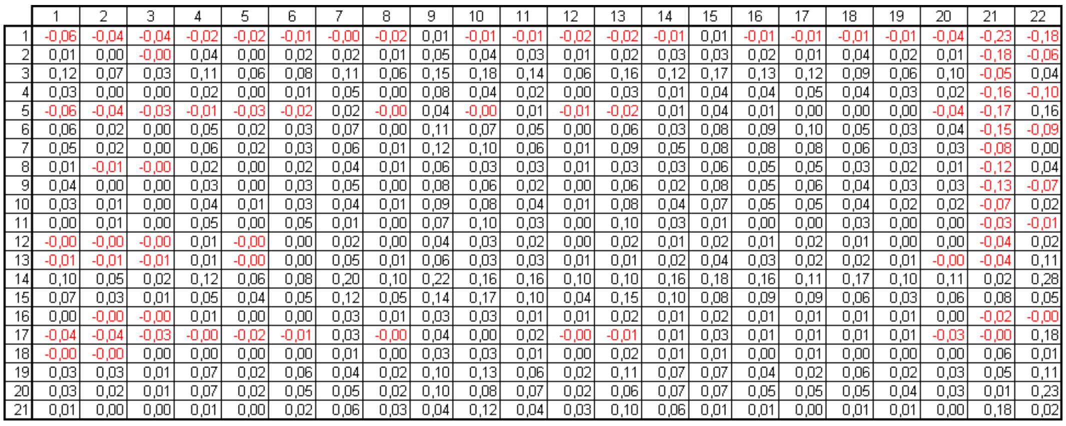
5.3.2 Finding the Concepts

To make sure the input is ready for processing by LSI, it has to be divided into separate documents if not already done so, these are the concepts described in the Chapter Method. To divide the functional design documentation we obtained of the systems WWO and ResaFasa, experts of both systems were consulted to help in determining how to divide the documentation. The experts had to be consulted because the documentation was not structured conform to certain standards. The goal is to find shared functionality between the systems, this resulted in a division on the basis of functionality. The input of WWO was divided in 21 documents, by looking at the submodules. The functional design documentation of ResaFasa is subdivided into paragraphs, these paragraphs were used as a guideline for dividing the functionality into 22 concepts.

5.3.3 Application of LSI

Now that the relevant input has been obtained and the concepts are identified from this input, it is time for the tool to calculate the similarities between the concepts. One concept is used as a query, the tool then returns a list of similarities between the query and the other concepts. These similarities are then, with the help of MS Excel, converted into a similarity matrix, as can be in Figure 5.3.

While applying retrieval, two important observations were made that have influenced the result. The first observations we made, was that dimension reduction had no effect on the result. We started with a rank-k model of 20% (in [22] they achieved good results with this value), with 1059 terms this means 212 factors, however there was no difference in between the results with of without dimension reduction, the same thing goes for 50 factors. With 10 factors however, we began to see changes, only to find out that, mapping a concept onto itself resulted in a similarity of 0,63 which means the results are useless.

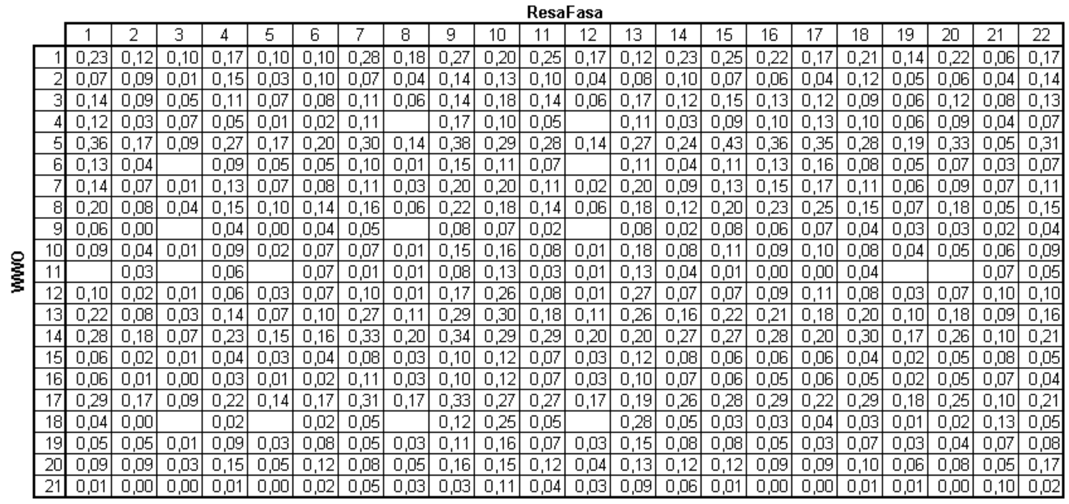


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	-0,06	-0,04	-0,04	-0,02	-0,02	-0,01	-0,00	-0,02	0,01	-0,01	-0,01	-0,02	-0,02	-0,01	0,01	-0,01	-0,01	-0,01	-0,01	-0,04	-0,23	-0,18
2	0,01	0,00	-0,00	0,04	0,00	0,02	0,02	0,01	0,05	0,04	0,03	0,01	0,02	0,03	0,03	0,02	0,01	0,04	0,02	0,01	-0,18	-0,06
3	0,12	0,07	0,03	0,11	0,06	0,08	0,11	0,06	0,15	0,18	0,14	0,06	0,16	0,12	0,17	0,13	0,12	0,09	0,06	0,10	-0,05	0,04
4	0,03	0,00	0,00	0,02	0,00	0,01	0,05	0,00	0,08	0,04	0,02	0,00	0,03	0,01	0,04	0,04	0,05	0,04	0,03	0,02	-0,16	-0,10
5	-0,06	-0,04	-0,03	-0,01	-0,03	-0,02	0,02	-0,00	0,04	-0,00	0,01	-0,01	-0,02	0,01	0,04	0,01	0,00	0,00	0,00	-0,04	-0,17	0,16
6	0,06	0,02	0,00	0,05	0,02	0,03	0,07	0,00	0,11	0,07	0,05	0,00	0,06	0,03	0,08	0,09	0,10	0,05	0,03	0,04	-0,15	-0,09
7	0,05	0,02	0,00	0,06	0,02	0,03	0,06	0,01	0,12	0,10	0,06	0,01	0,09	0,05	0,08	0,08	0,08	0,06	0,03	0,03	-0,08	0,00
8	0,01	-0,01	-0,00	0,02	0,00	0,02	0,04	0,01	0,06	0,03	0,03	0,01	0,03	0,03	0,06	0,05	0,05	0,03	0,02	0,01	-0,12	0,04
9	0,04	0,00	0,00	0,03	0,00	0,03	0,05	0,00	0,08	0,06	0,02	0,00	0,06	0,02	0,08	0,05	0,06	0,04	0,03	0,03	-0,13	-0,07
10	0,03	0,01	0,00	0,04	0,01	0,03	0,04	0,01	0,09	0,08	0,04	0,01	0,08	0,04	0,07	0,05	0,05	0,04	0,02	0,02	-0,07	0,02
11	0,00	0,01	0,00	0,05	0,00	0,05	0,01	0,00	0,07	0,10	0,03	0,00	0,10	0,03	0,01	0,00	0,00	0,03	0,00	0,00	-0,03	-0,01
12	-0,00	-0,00	-0,00	0,01	-0,00	0,00	0,02	0,00	0,04	0,03	0,02	0,00	0,02	0,01	0,02	0,01	0,02	0,01	0,00	0,00	-0,04	0,02
13	-0,01	-0,01	-0,01	0,01	-0,00	0,00	0,05	0,01	0,06	0,03	0,03	0,01	0,01	0,02	0,04	0,03	0,02	0,02	0,01	-0,00	-0,04	0,11
14	0,10	0,05	0,02	0,12	0,06	0,08	0,20	0,10	0,22	0,16	0,16	0,10	0,10	0,16	0,18	0,16	0,11	0,17	0,10	0,11	0,02	0,28
15	0,07	0,03	0,01	0,05	0,04	0,05	0,12	0,05	0,14	0,17	0,10	0,04	0,15	0,10	0,08	0,09	0,09	0,06	0,03	0,06	0,08	0,05
16	0,00	-0,00	-0,00	0,01	0,00	0,00	0,03	0,01	0,03	0,03	0,01	0,01	0,02	0,01	0,02	0,01	0,01	0,01	0,01	0,00	-0,02	-0,00
17	-0,04	-0,04	-0,03	-0,00	-0,02	-0,01	0,03	-0,00	0,04	0,00	0,02	-0,00	-0,01	0,01	0,03	0,01	0,01	0,01	0,01	-0,03	-0,00	0,18
18	-0,00	-0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,03	0,03	0,01	0,00	0,02	0,01	0,01	0,00	0,01	0,00	0,00	0,00	0,06	0,01
19	0,03	0,03	0,01	0,07	0,02	0,06	0,04	0,02	0,10	0,13	0,06	0,02	0,11	0,07	0,07	0,04	0,02	0,06	0,02	0,03	0,05	0,11
20	0,03	0,02	0,01	0,07	0,02	0,05	0,05	0,02	0,10	0,08	0,07	0,02	0,06	0,07	0,07	0,05	0,05	0,05	0,04	0,03	0,01	0,23
21	0,01	0,00	0,00	0,01	0,00	0,02	0,06	0,03	0,04	0,12	0,04	0,03	0,10	0,06	0,01	0,01	0,00	0,01	0,01	0,00	0,18	0,02

Figure 5.2: The difference between mapping one way or the other

The second observation has to do with the difference between mapping both ways, WWO on ResaFasa and vice versa. When the matrices of the resulting mappings were compared, rather large differences were seen, shown in Figure 5.2. After consulting with an expert on LSI, the cause for the differences was found, the dictionaries created when indexing the input were too different from each other, for example (14,22) the difference is 0,28, while the actual similarity is only 0,21, shown in Figure 5.3. The intersecting terms are 17% of the total number of terms, 180 of the 1059 terms are in both systems. To solve this problem, both systems were indexed together. In this way the queries were mapped on both systems at once, this resulted in the complete matrix A.3 with similarities of 1's on the diagonal, except for some minor errors. Figure 5.3 shows an excerpt, the relevant part.

The similarity values in Figure 5.3 are relatively low when compared to other studies [22]. Even though the similarity values are low, this does not necessarily mean that the results are useless, the underlying structure can still be present, as is demonstrated in the following sub-sections. The only repercussion is, it should be kept in mind while filtering,



	ResaFasa																					
OWW	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0,23	0,12	0,10	0,17	0,10	0,10	0,28	0,18	0,27	0,20	0,25	0,17	0,12	0,23	0,25	0,22	0,17	0,21	0,14	0,22	0,06	0,17
2	0,07	0,09	0,01	0,15	0,03	0,10	0,07	0,04	0,14	0,13	0,10	0,04	0,08	0,10	0,07	0,06	0,04	0,12	0,05	0,06	0,04	0,14
3	0,14	0,09	0,05	0,11	0,07	0,08	0,11	0,06	0,14	0,18	0,14	0,06	0,17	0,12	0,15	0,13	0,12	0,09	0,06	0,12	0,08	0,13
4	0,12	0,03	0,07	0,05	0,01	0,02	0,11		0,17	0,10	0,05		0,11	0,03	0,09	0,10	0,13	0,10	0,06	0,09	0,04	0,07
5	0,36	0,17	0,09	0,27	0,17	0,20	0,30	0,14	0,38	0,29	0,28	0,14	0,27	0,24	0,43	0,36	0,35	0,28	0,19	0,33	0,05	0,31
6	0,13	0,04		0,09	0,05	0,05	0,10	0,01	0,15	0,11	0,07		0,11	0,04	0,11	0,13	0,16	0,08	0,05	0,07	0,03	0,07
7	0,14	0,07	0,01	0,13	0,07	0,08	0,11	0,03	0,20	0,20	0,11	0,02	0,20	0,09	0,13	0,15	0,17	0,11	0,06	0,09	0,07	0,11
8	0,20	0,08	0,04	0,15	0,10	0,14	0,16	0,06	0,22	0,18	0,14	0,06	0,18	0,12	0,20	0,23	0,25	0,15	0,07	0,18	0,05	0,15
9	0,06	0,00		0,04	0,00	0,04	0,05		0,08	0,07	0,02		0,08	0,02	0,08	0,06	0,07	0,04	0,03	0,03	0,02	0,04
10	0,09	0,04	0,01	0,09	0,02	0,07	0,07	0,01	0,15	0,16	0,08	0,01	0,18	0,08	0,11	0,09	0,10	0,08	0,04	0,05	0,06	0,09
11		0,03		0,06		0,07	0,01	0,01	0,08	0,13	0,03	0,01	0,13	0,04	0,01	0,00	0,00	0,04			0,07	0,05
12	0,10	0,02	0,01	0,06	0,03	0,07	0,10	0,01	0,17	0,26	0,08	0,01	0,27	0,07	0,07	0,09	0,11	0,08	0,03	0,07	0,10	0,10
13	0,22	0,08	0,03	0,14	0,07	0,10	0,27	0,11	0,29	0,30	0,18	0,11	0,26	0,16	0,22	0,21	0,18	0,20	0,10	0,18	0,09	0,16
14	0,28	0,18	0,07	0,23	0,15	0,16	0,33	0,20	0,34	0,29	0,29	0,20	0,20	0,27	0,27	0,28	0,20	0,30	0,17	0,26	0,10	0,21
15	0,06	0,02	0,01	0,04	0,03	0,04	0,08	0,03	0,10	0,12	0,07	0,03	0,12	0,08	0,06	0,06	0,06	0,04	0,02	0,05	0,08	0,05
16	0,06	0,01	0,00	0,03	0,01	0,02	0,11	0,03	0,10	0,12	0,07	0,03	0,10	0,07	0,06	0,05	0,06	0,05	0,02	0,05	0,07	0,04
17	0,29	0,17	0,09	0,22	0,14	0,17	0,31	0,17	0,33	0,27	0,27	0,17	0,19	0,26	0,28	0,29	0,22	0,29	0,18	0,25	0,10	0,21
18	0,04	0,00		0,02		0,02	0,05		0,12	0,25	0,05		0,28	0,05	0,03	0,03	0,04	0,03	0,01	0,02	0,13	0,05
19	0,05	0,05	0,01	0,09	0,03	0,08	0,05	0,03	0,11	0,16	0,07	0,03	0,15	0,08	0,08	0,05	0,03	0,07	0,03	0,04	0,07	0,08
20	0,09	0,09	0,03	0,15	0,05	0,12	0,08	0,05	0,16	0,15	0,12	0,04	0,13	0,12	0,12	0,09	0,09	0,10	0,06	0,08	0,05	0,17
21	0,01	0,00	0,00	0,01	0,00	0,02	0,05	0,03	0,03	0,11	0,04	0,03	0,09	0,06	0,01	0,00	0,00	0,01	0,01	0,00	0,10	0,02

Figure 5.3: The relevant excerpt of the similarity results of LSI

as it was.

The values in the similarity matrix were filtered in the two ways described in the previous Chapter, the relative row filter and the relative column filter. The results from the filtering are discussed in section 5.4

5.3.4 Getting a traceability matrix from the experts of the system

The results of the tool have been compared to a traceability matrix constructed by experts of both systems in order to show, to what degree the approach works in finding the shared functionality. Although, this is not 100% accurate, it will help to validate the results.

Since the systems in question are being developed by separate teams, there is no one that has intimate knowledge of the inner workings of both systems with respect to the gross-net calculations. To obtain a traceability matrix, experts from both systems were consulted.

The main activities that make up the process of creating a traceability matrix, in the context discussed above, were executed during a time span of a week. This description excludes the time that was needed for a supervisor of the meetings to get a general idea of the material, more on this below.

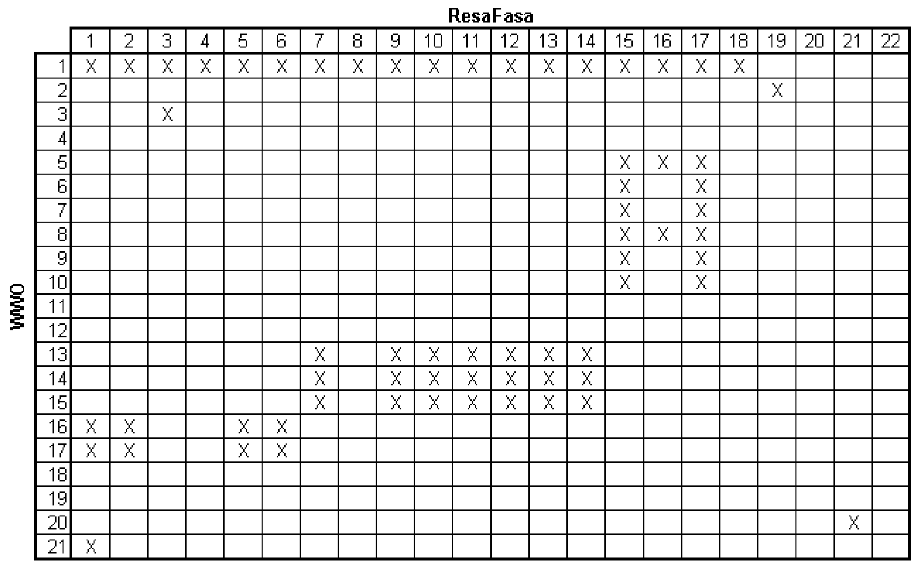
- A meeting with one, or more, experts from the systems was planned, during this meeting the experts were presented with functional design documents of the other system and asked to fill out a matrix. Since experts of the other system were present, it was efficient, in the way that they were able to ask questions if they did not understand something or the meaning of certain term.

- They continued working on the matrices on their own for two more days, in which they inspected the documentation of the other system.

- A final meeting was called in which the supervisor compared the matrices that were handed in, with one expert from each system present. In this session, every difference between the matrices was discussed until a consensus was achieved, this resulted in the Figure 5.4.

To be able to guide along the discussion of the final meeting, in which the conflicting links were discussed, the supervisor had superficial knowledge of the gross-net calculations and some idea of how the systems work. The supervisor obtained this knowledge gradually during the months we worked on this project, however most knowledge was obtained by conducting interviews with experts of both systems.

The intermediate results are included in the appendix, Diagram A.1 was created by the WWO expert and Diagram A.2 was created by the experts of ResaFasa.



ResaFasa

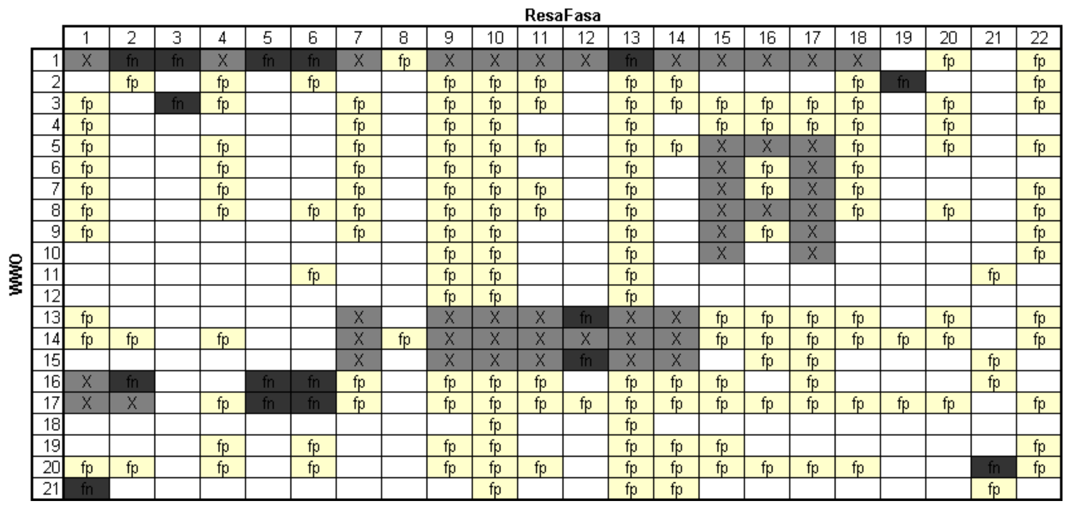
WWO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				
2																			X			
3			X																			
4																						
5															X	X	X					
6															X		X					
7															X		X					
8															X	X	X					
9															X		X					
10															X		X					
11																						
12																						
13							X		X	X	X	X	X	X								
14							X		X	X	X	X	X	X								
15							X		X	X	X	X	X	X								
16	X	X			X	X																
17	X	X			X	X																
18																						
19																						
20																					X	
21	X																					

Figure 5.4: The traceability matrix of the experts

5.3.5 Comparing the results of LSI with the matrix obtained from the experts

By comparing the results achieved with LSI and those found by the experts, the recall (R) and precision (P) were calculated. These results are shown in Table 5.7. In Figure 5.5 the reconstructed traceability matrix for the row relative filter is shown. In Figure 5.6 the reconstructed traceability matrix for the column relative filter is shown.

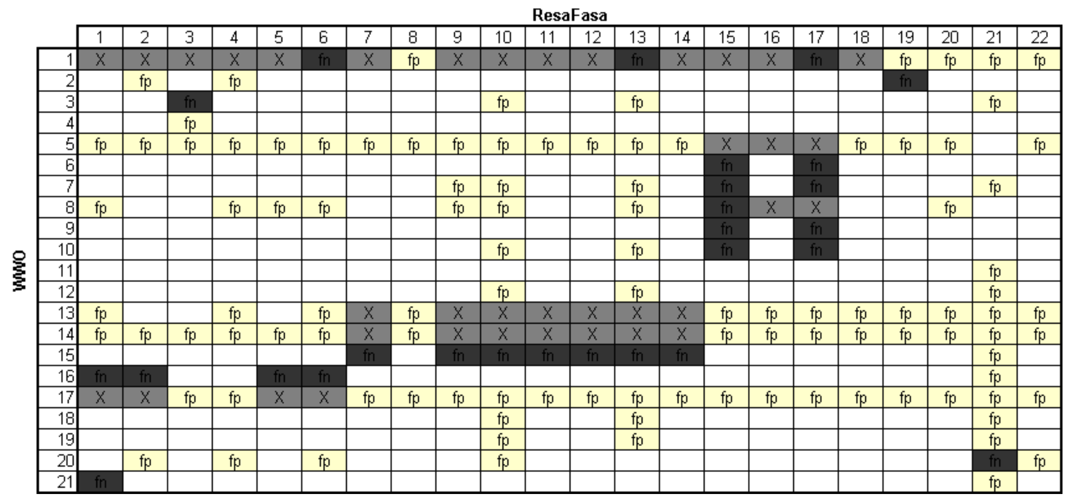
The correctly reconstructed links are marked by an ”X”. The empty cells have not been found by LSI or the experts. The cells marked ”fp” are false positives, this means they are invalid reconstructed links when compared to the links of the experts. There is a third category of cells, the false negatives marked by ”fn”, these links were found by the experts, but not by our approach.



ResaFasa

WWO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	X	fn	fn	X	fn	fn	X	fp	X	X	X	X	fn	X	X	X	X	X		fp		fp
2		fp		fp		fp			fp	fp	fp		fp	fp				fp	fn			fp
3	fp		fn	fp			fp		fp	fp	fp		fp	fp	fp	fp	fp	fp		fp		fp
4	fp						fp		fp	fp			fp		fp	fp	fp	fp		fp		
5	fp			fp			fp		fp	fp	fp		fp	fp	X	X	X	fp		fp		fp
6	fp			fp			fp		fp	fp			fp		X	fp	X	fp				
7	fp			fp			fp		fp	fp	fp		fp		X	fp	X	fp				fp
8	fp			fp		fp	fp		fp	fp	fp		fp		X	X	X	fp		fp		fp
9	fp						fp		fp	fp			fp		X	fp	X					fp
10									fp	fp			fp		X		X					fp
11						fp			fp	fp			fp								fp	
12									fp	fp			fp									
13	fp						X		X	X	X	fn	X	X	fp	fp	fp	fp		fp		fp
14	fp	fp		fp			X	fp	X	X	X	X	X	X	fp	fp	fp	fp	fp	fp		fp
15							X		X	X	X	fn	X	X		fp	fp				fp	
16	X	fn			fn	fn	fp		fp	fp	fp		fp	fp	fp		fp				fp	
17	X	X		fp	fn	fn	fp		fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp		fp
18										fp			fp									
19				fp		fp			fp	fp			fp	fp	fp							fp
20	fp	fp		fp		fp			fp	fp	fp		fp	fp	fp	fp	fp	fp			fn	fp
21	fn									fp			fp	fp							fp	

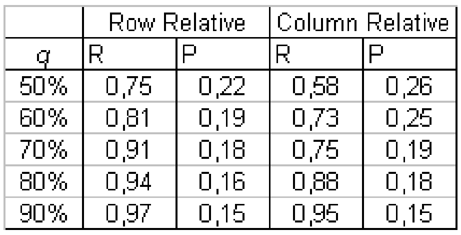
Figure 5.5: Reconstructed traceability matrix using row relative selection strategy, q=50%



ResaFasa

WWO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	X	X	X	X	X	fn	X	fp	X	X	X	X	fn	X	X	X	fn	X	fp	fp	fp	fp
2		fp		fp															fn			
3			fn							fp			fp								fp	
4			fp																			
5	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	X	X	X	fp	fp	fp		fp
6															fn		fn					
7									fp	fp			fp		fn		fn				fp	
8	fp			fp	fp	fp			fp	fp			fp		fn	X	X			fp		
9															fn		fn					
10										fp			fp		fn		fn					
11																					fp	
12										fp			fp								fp	
13	fp			fp		fp	X	fp	X	X	X	X	X	X	fp	fp	fp	fp	fp	fp	fp	fp
14	fp	fp	fp	fp	fp	fp	X	fp	X	X	X	X	X	X	fp	fp	fp	fp	fp	fp	fp	fp
15							fn		fn	fn	fn	fn	fn	fn							fp	
16	fn	fn			fn	fn															fp	
17	X	X	fp	fp	X	X	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp	fp
18										fp			fp								fp	
19										fp			fp								fp	
20		fp		fp		fp				fp											fn	fp
21	fn																				fp	

Figure 5.6: Reconstructed traceability matrix using column relative selection strategy, q=50%



	Row Relative		Column Relative	
q	R	P	R	P
50%	0,75	0,22	0,58	0,26
60%	0,81	0,19	0,73	0,25
70%	0,91	0,18	0,75	0,19
80%	0,94	0,16	0,88	0,18
90%	0,97	0,15	0,95	0,15

Figure 5.7: Recall and precision

5.4 Results of the Experiments

The results in Table 5.7 show a low precision, this is caused by many false positives. There are some differences between the row and column relative filtering, the row relative seems to be performing the best, a recall of 0,91 and a precision of 0,18 is achieved with a q of 70%. The column relative filtering has one particular strong result though, the recall is massively improved, from 0,58 to 0,73, when q goes from 50% to 60%, this is caused by the similarities that are included by changing q.

While looking at the results of the relative row filter with a q of 20%, three columns of false positives stand out; 9, 10 and 13. The same thing occurred with the column relative filter with a q of 20%, only this time it involves three rows; especially 5, 14 and 17. When looking in the documents which represent the columns, a possible explanation was found, in these documents, the laws relevant to the other system are named explicitly and they are not as frequent in the other columns, these laws are ”important” terms. When looking at the documents belonging to the rows, we noticed the following, 5 is a module from where a lot of other modules are called and 14 and 17 are part of a section of calculations where calculations of the other system are taken into account, which results in a lot of shared terms.

When looking at the Graph 5.8, the spread of the rows appear to be lower than that of the columns. This suspicion is conformed by the averages in Table 5.2, the average of the columns is significantly higher than the average of the rows. These results show that when WWO is mapped onto ResaFasa, the similarities are closer to the mean than the other way around.

5.5 Evaluation of the Experiments

In this section an interpretation of the results described above is given.

Dimension reduction, which is done in LSI by singular value decomposition, is the way the latent semantic structure is uncovered. We found that the results are not improved by applying this, we believe this means that LSI has not found a latent semantic structure in this context. This means that in our work multiple words are not mapped on a concept, but words are mapped on words. We believe this is caused by the low amount of intersecting terms discussed below.

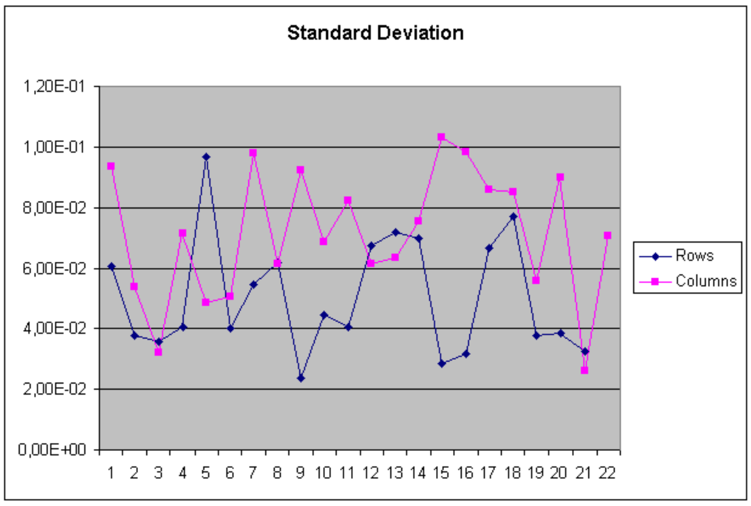


Figure 5.8: Spread of the results

	Rows	**Columns**
Average Spread	5,05E-02	7,13E-02

Table 5.2: Average spread of the similarity matrix

The amount of intersecting terms makes up 17% of the total amount of terms. This seems low, although, percentages of other studies are not available. As stated above, this low percentage could account for the failure of applying dimension reduction. When two subjects are compared and they have very little building blocks in common, it makes sense that the similarity between these will be low as well, as is discussed below. To conclusively proof that the low amount of shared terms is the cause for the failure to apply dimension reduction, further research has to be done, but it does point in that direction.

Because of the low number of intersecting terms, we believe the similarities are low as well. This because the number of terms that can be matched from one concept to another will be relatively low.

Although, the following claims have not been confirmed, we believe the low similarity values and the low number of intersecting terms can be attributed to the fact that a comparison was done between two separate systems instead of documentation describing the same system as in [22].

A factor which we believe contributes as well, is the distance in abstraction between the documents of the different systems, although, they are both labelled as functional design, one consists completely of free text, while the other consists for the most part of pseudo code.

By looking a Figure 5.8 and Table 5.2 we came up with the following. Although, not conclusively proven, we believe the difference in spread means that, compared to WWO the desired functionality is more spread out in ResaFasa, which we interpret as a sign that WWO is better decomposed than ResaFasa with respect to the gross-net calculations.

The recall and precision achieved in our experiments can be interpreted in several ways, it depends on the intended application of the results. When the objective is to achieve 100% recall, all expert links are retrieved by the tool, the row relative filter performs best with a recall of 0,97 and a precision of 0,15. The high recall suggests most expert links can be recovered, the low precision indicates a lot of false positives. Whether these results are useful depends on the intended application.

When compared to results from case studies where LSI was used as well, the following was found. High precisions can not be achieved, however, when the recall becomes higher, and this is desirable [22], the results are comparable to the upper bound of the pacman case study. During the comparison, something else is noticed, the same recall and precision values are achieved at a much higher q, for example our values at a threshold of 80%, were achieved by them with a threshold of 40% and the same thing goes for 60% of ours and 30% of theirs. This means our results have a higher spread, which indicates fewer terms used in all concepts.

5.6 Recommendations for improving the results

On the basis of the results and observations made during the process described above, below an advice is given on how to alter the input or the process so that the results of LSI may be improved or the process simplified.

By looking at our results, the recommendation is to use the row relative filtering in this context to achieve the best results. One thing that could be improved is the following, the documentation could structured to adhere to certain standards, MIL-std 498 or IEEE-1233-1998 [22], this will speed up the process of extracting the concepts from the input, however, if this will improve the eventual results (of recall and precision) remains a question.

Chapter 6

Conclusions and Future Work

In this chapter along with the conclusions, a listing of contributions is given and a number of research issues is addressed in the section Future Work.

6.1 Conclusions

This thesis begins with the following hypothesis: *LSI can be used to identify shared functionality between two systems on the basis of structured documentation.* This hypothesis is decomposed into several objectives, in the form of research questions. Whether these objectives were achieved is discussed below.

Although, our work is similar to that presented in [22], there are several differences. We apply LSI to the documentation of two systems, instead of different types of documentation of the same system. One of our systems has a documentation, which, for the most part, consists of pseudo code. This means, there is a difference in abstraction between the documentation of the two systems, since the other is free text.

From the results achieved by applying the method (described in Chapter 4) on the two systems, which were described and discussed in Chapter 5, we conclude the following:

> *LSI cannot fully recover the shared functionality from the two systems, as initially planned, because of the many false positives. However, we believe LSI can be used to reduce the search space. This means the experts, recovering the shared functionality, have less documents to evaluate.*

Below, conclusions regarding the other objectives are presented:

- When the theoretical side of the recall and precision values, that were achieved in Chapter 5, are explored and compared to other cases, the results are good. For example when compared to the pacman case in [22], which they think of as an upper bound, our results are similar. A high precision can not be achieved, however, when the recall becomes higher, and this is desirable [22], the results are comparable to the upper bound of the pacman case study. From this we conclude that, the results in our context, are comparable to other applications of LSI.

- Applying LSI to concepts of two different systems has resulted in a low number (17%) of intersecting terms, we believe, although, we have not proven it, this is the cause for the failure to apply dimension reduction. More on how to improve this situation in the section Future Work.
- We believe the results are influenced in a negative manner by applying LSI to structured documentation. We think, the distance in level of abstraction between the documentation, of the two systems, is too large, and has partially caused the low similarities. One system has a documentation, consisting, for the most part, of pseudo code, while the documentation of the other system is free text.

The vision of this project was to recover 100% of the shared functionality automatically, by applying LSI. However, the results of our experiments did not fully conform this vision. Instead, we conclude that LSI can help in recovering the shared functionality, by reducing the search space. How much this search is reduced depends on the value of the threshold, however, if this threshold is set too firm, there is a risk that links are not recovered, which should be recovered.

6.2 Contributions

This section explains in what way the work we have done is different from the current applications of LSI.

The objective of this document is to research if LSI can help in finding shared functionality in a large industrial context, namely the systems presented in the proof of concept. Even though, LSI did not completely solve the problem of recovering shared functionality, there are still some contributions that came out of the work, they are presented below:

- LSI can be used to reduce the search space, for experts, when recovering shared functionality.
- A detailed description of which steps to follow that lead to a traceability matrix by using LSI, which is a refinement of the MAREV method and tailored for this context.
- A traceability matrix of concerning both systems with respect to the gross-net calculations created by experts, and a description of how to create such a matrix.
- The recall and precision values with regard to finding shared functionality in the two systems, accompanied by an explanation of what their use is.

6.3 Future Work

This section covers, aspects of the project, given the current knowledge and experience, we would have done differently, or would have done if there was time.

The results of LSI can be further improved by a strategy inspired by the relatively low number of intersecting terms in the dictionaries of both systems. This strategy entails searching for differences in terminology between the systems, the synonyms can then be inserted in the documents of the other system. In this way the traceability links will be fortified and the similarities boosted, we also believe dimension reduction can be applied once the number of intersecting terms are improved and this could also improve the recall and precision.

The scalability of the approach can be researched. We have concluded, the approach reduces the amount of work the experts will have to do when recovering shared functionality. However, at which point, considering the size of the systems, does LSI significantly reduces the amount of work, remains a question.

Bibliography

[1] Nederlandse software metrieken associatie. http://www.nesma.nl/.

[2] G. Antoniol, G. Canfora, G. Casazza, and A. De Lucia. Information retrieval models for recovering traceability links between code and documentation. *Proceedings of the International Conference on Software Maintenance (ICSM'00)*, page 40, 2000.

[3] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E. Merlo. Tracing object-oriented code into functional requirements. *Program Comprehension, 2000. Proceedings. IWPC 2000. 8th International Workshop on*, pages 79–86, 2000.

[4] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E. Merlo. Recovering traceability links between code and documentation. *Software Engineering, IEEE Transactions on*, 28(10):970–983, 2002.

[5] G. Antoniol, G. Canfora, A. De Lucia, and E. Merlo. Recovering Code to Documentation Links in OO Systems. *Proc. of the Working Conference on Reverse Engineering*, pages 136–144, 1999.

[6] RS Arnold and SA Bohner. Impact analysis-Towards a framework for comparison. *Software Maintenance, 1993. CSM-93, Proceedings., Conference on*, pages 292–301, 1993.

[7] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*. Addison-Wesley Harlow, England, 1999.

[8] M.W. Berry, Z. Drmac, and E.R. Jessup. Matrices, Vector Spaces, and Information Retrieval. *SIAM Review*, 41:335, 1999.

[9] J. CONKLIN and M.L. BEGEMAN. glBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM Transactions on Office Information Systems*, 6(4):303–331, 1988.

[10] A. De Lucia, F. Fasano, R. Oliveto, and G. Tortora. Enhancing an artefact management system with traceability recovery features. *Software Maintenance, 2004. Proceedings. 20th IEEE International Conference on*, pages 306–315, 2004.

[11] A. De Lucia, F. Fasano, R. Oliveto, and G. Tortora. ADAMS Re-Trace: A Traceability Recovery Tool. *Proceedings of the Ninth European Conference on Software Maintenance and Reengineering*, pages 32–41, 2005.

[12] A. De Lucia, F. Fasano, R. Oliveto, and G. Tortora. Can Information Retrieval Techniques Effectively Support Traceability Link Recovery? *Proceedings of 14th IEEE International Conference on Program Comprehension (ICPC'06), Athens, Greece*, pages 307–316, 2006.

[13] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[14] M. Di Penta, S. Gradara, and G. Antoniol. Traceability Recovery in RAD Software Systems. *Proceedings of the 10th IEEE International Workshop on Program Comprehension (Paris, France). IEEE Computer Society Press, Los Alamitos, CA*, 2002.

[15] S.T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23(2):229–236, 1991.

[16] C. Faloutsos and D.W. Oard. A survey of information retrieval and filtering methods. *University of Maryland at College Park, Technical Report CS-TR-3514*, 1995.

[17] B. Fischer. Specification-based Browsing of Software Component Libraries. *Automated Software Engineering*, pages 74–83, 1998.

[18] W.B. Frakes and R. Baeza-Yates. Information Retrieval: Data Structures and Algorithms. *New Jersey*, 1992.

[19] WB Frakes and BA Nejmeh. Software reuse through information retrieval. *ACM SIGIR Forum*, 21(1-2):30–36, 1986.

[20] P. Klint and C. Verhoef. Evolutionary software engineering: A component-based approach. *IFIP WG*, 2:1–18, 1998.

[21] M. Lormans and A. van Deursen. Can lsi help reconstructing requirements traceability in design and test? In *Proceedings of the 10th European Conference on Software Maintenance and Reengineering (CSMR'06)*, pages 47–56. IEEE Computer Society, 2006.

[22] M. Lormans and A. van Deursen. Reconstructing requirements traceability in design and test using latent semantic indexing. *Technical University of Delft, Technical Report TUD-SERG-2007-007*, 2007.

[23] YS Maarek, DM Berry, and GE Kaiser. An information retrieval approach for automatically constructingsoftware libraries. *Software Engineering, IEEE Transactions on*, 17(8):800–813, 1991.

[24] YS Maarek and FZ Smadja. Full text indexing based on lexical relations an application: software libraries. *Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 198–206, 1989.

[25] J.I. Maletic and A. Marcus. Supporting Program Comprehension Using Semantic and Structural Information. *INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING*, 23:103–112, 2001.

[26] J.I. Maletic, E.V. Munson, A. Marcus, and T.N. Nguyen. Using a hypertext model for traceability link conformance analysis. *Proc. of the 2nd Int. Work. on Traceability in Emerging Forms of Software Engineering*, pages 47–54.

[27] JI Maletic and N. Valluri. Automatic software clustering via Latent Semantic Analysis. *Automated Software Engineering, 1999. 14th IEEE International Conference on.*, pages 251–254, 1999.

[28] A. Marcus and J.I. Maletic. Identification of High-Level Concept Clones in Source Code. *Proceedings Automated Software Engineering (ASE'01), San Diego, CA*, pages 107–114, 2001.

[29] A. Marcus and J.I. Maletic. Recovering documentation-to-source-code traceability links using latent semantic indexing. *Proceedings of the Twenty-Fifth International Conference on Software Engineering*, pages 3–10, 2003.

[30] A. Marcus, J.I. Maletic, and A. Sergeyev. Recovery of Traceability Links between Software Documentation and Source Code. *International Journal of Software Engineering and Knowledge Engineering*, 15(5):811–836, 2005.

[31] Leon Moonen. *Exploring Software Systems*. PhD thesis, Faculty of Natural Sciences, Mathematics, and Computer Science, University of Amsterdam, December 2002.

[32] FAC Pinheiro and JA Goguen. An object-oriented tool for tracing requirements. *Software, IEEE*, 13(2):52–64, 1996.

[33] B. Ramesh and V. Dhar. Supporting systems development using knowledge captured during requirements engineering. *IEEE Transactions on Software Engineering*, 9(2):498–510, 1992.

[34] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

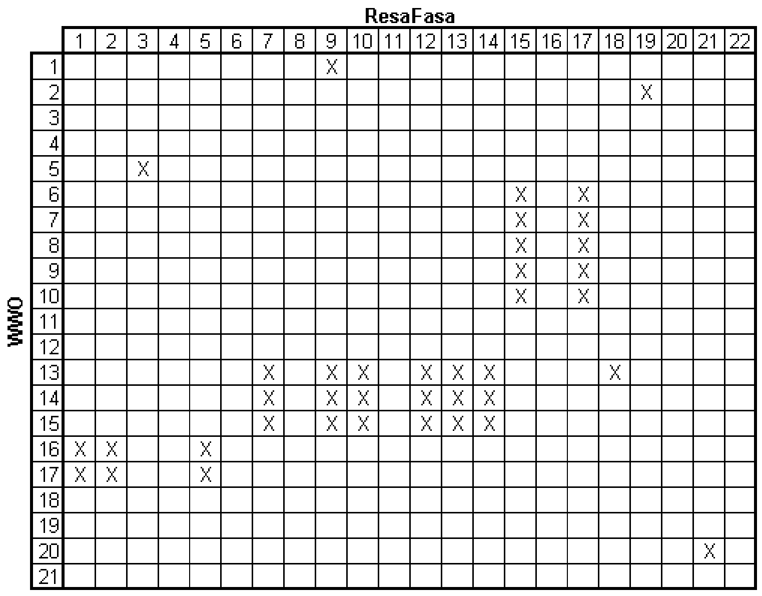
[35] G. Salton and MJ McGill. Introduction to Modern Information Retrieval. *McGrawHill Book Co., New York.*

[36] CJ Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann Newton, MA, USA, 1979.

[37] D Zeimpekis and E Gallopoulos. Text to Matrix Generator Users Guide. *Department of Computer Engineering and Informatics, University of Patras, Greece*.

Appendix A

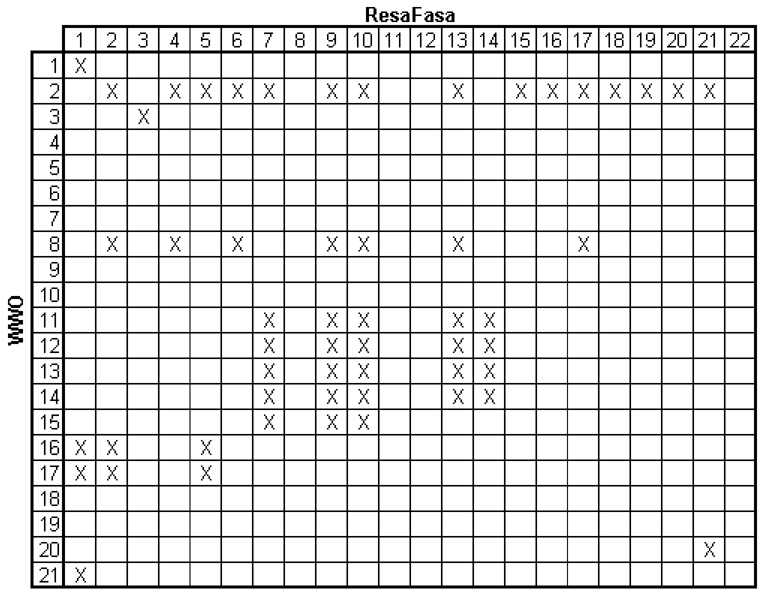
Diagrams



ResaFasa

WWO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1									X													
2																			X			
3																						
4																						
5			X																			
6															X		X					
7															X		X					
8															X		X					
9															X		X					
10															X		X					
11																						
12																						
13							X		X	X		X	X	X				X				
14							X		X	X		X	X	X								
15							X		X	X		X	X	X								
16	X	X			X																	
17	X	X			X																	
18																						
19																						
20																					X	
21																						

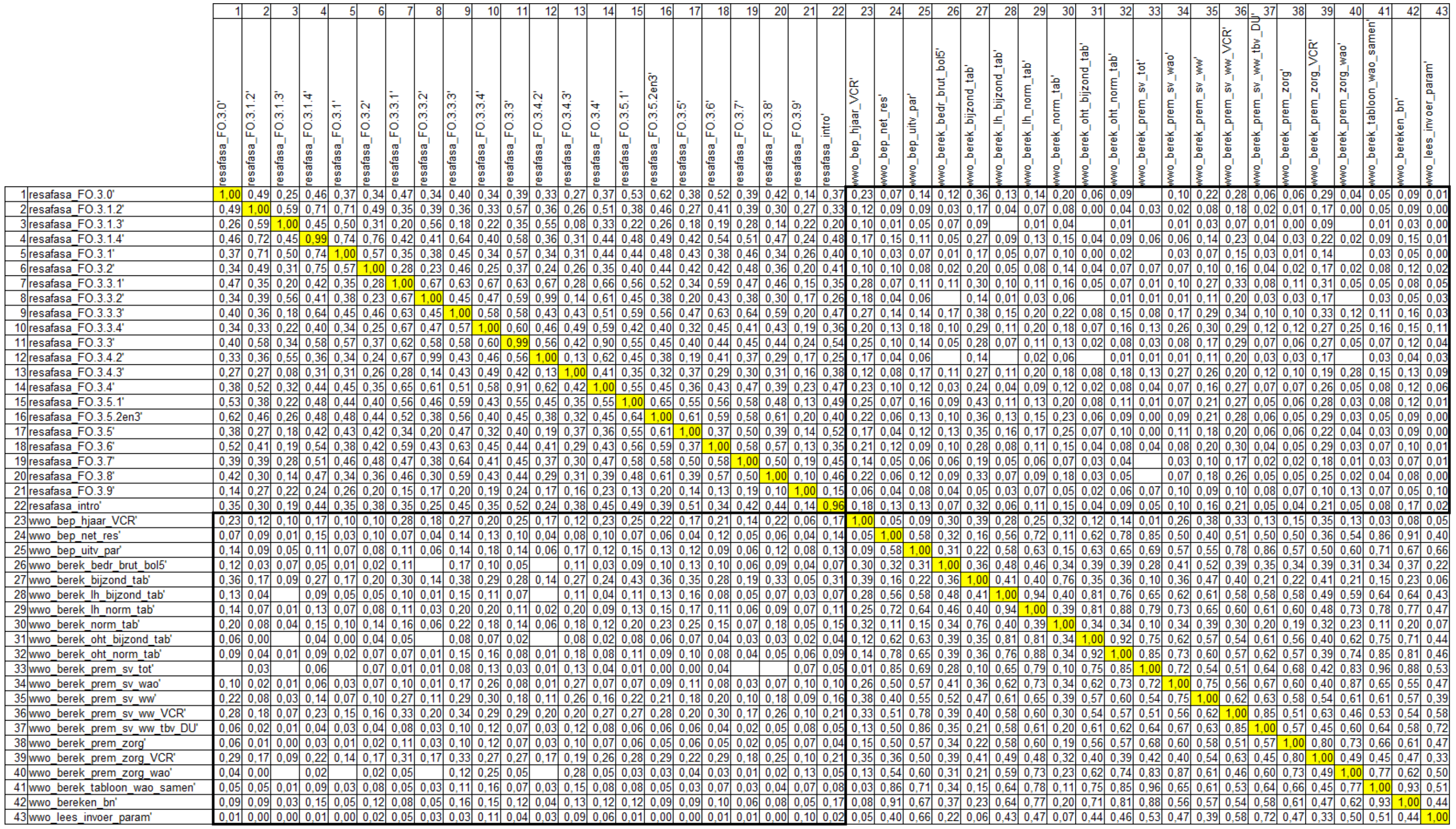
Figure A.1: The expert traceability matrix of WWO



ResaFasa

WWO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	X																					
2		X		X	X	X	X		X	X			X		X	X	X	X	X	X	X	
3			X																			
4																						
5																						
6																						
7																						
8		X		X		X			X	X			X				X					
9																						
10																						
11							X		X	X			X	X								
12							X		X	X			X	X								
13							X		X	X			X	X								
14							X		X	X			X	X								
15							X		X	X												
16	X	X			X																	
17	X	X			X																	
18																						
19																						
20																					X	
21	X																					

Figure A.2: The expert traceability matrix of ResaFasa



		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
		resafasa_FO.3.0'	resafasa_FO.3.1.2'	resafasa_FO.3.1.3'	resafasa_FO.3.1.4'	resafasa_FO.3.1'	resafasa_FO.3.2'	resafasa_FO.3.3.1'	resafasa_FO.3.3.2'	resafasa_FO.3.3.3'	resafasa_FO.3.3.4'	resafasa_FO.3.3'	resafasa_FO.3.4.2'	resafasa_FO.3.4.3'	resafasa_FO.3.4'	resafasa_FO.3.5.1'	resafasa_FO.3.5.2en3'	resafasa_FO.3.5'	resafasa_FO.3.6'	resafasa_FO.3.7'	resafasa_FO.3.8'	resafasa_FO.3.9'	resafasa_intro'	wwo_bep_hjaar_VCR'	wwo_bep_net_res'	wwo_bep_uitv_par'	wwo_berek_bedr_brut_bol5'	wwo_berek_bijzond_tab'	wwo_berek_lh_bijzond_tab'	wwo_berek_lh_norm_tab'	wwo_berek_norm_tab'	wwo_berek_oht_bijzond_tab'	wwo_berek_oht_norm_tab'	wwo_berek_prem_sv_tot'	wwo_berek_prem_sv_wao'	wwo_berek_prem_sv_ww'	wwo_berek_prem_sv_ww_VCR'	wwo_berek_prem_sv_ww_tbv_DU'	wwo_berek_prem_zorg'	wwo_berek_prem_zorg_VCR'	wwo_berek_prem_zorg_wao'	wwo_berek_tabloon_wao_samen'	wwo_bereken_bn'	wwo_lees_invoer_param'
1	resafasa_FO.3.0'	1,00	0,49	0,25	0,46	0,37	0,34	0,47	0,34	0,40	0,34	0,39	0,33	0,27	0,37	0,53	0,62	0,38	0,52	0,39	0,42	0,14	0,37	0,23	0,07	0,14	0,12	0,36	0,13	0,14	0,20	0,06	0,09		0,10	0,22	0,28	0,06	0,06	0,29	0,04	0,05	0,09	0,01
2	resafasa_FO.3.1.2'	0,49	1,00	0,59	0,71	0,71	0,49	0,35	0,39	0,36	0,33	0,57	0,36	0,26	0,51	0,38	0,46	0,27	0,41	0,39	0,30	0,27	0,33	0,12	0,09	0,09	0,03	0,17	0,04	0,07	0,08	0,00	0,04	0,03	0,02	0,08	0,18	0,02	0,01	0,17	0,00	0,05	0,09	0,00
3	resafasa_FO.3.1.3'	0,26	0,59	1,00	0,45	0,50	0,31	0,20	0,56	0,18	0,22	0,35	0,55	0,08	0,33	0,22	0,26	0,18	0,19	0,28	0,14	0,22	0,20	0,10	0,01	0,05	0,07	0,09		0,01	0,04		0,01		0,01	0,03	0,07	0,01	0,00	0,09		0,01	0,03	0,00
4	resafasa_FO.3.1.4'	0,46	0,72	0,45	0,99	0,74	0,76	0,42	0,41	0,64	0,40	0,58	0,36	0,31	0,44	0,48	0,49	0,42	0,54	0,51	0,47	0,24	0,48	0,17	0,15	0,11	0,05	0,27	0,09	0,13	0,15	0,04	0,09	0,06	0,06	0,14	0,23	0,04	0,03	0,22	0,02	0,09	0,15	0,01
5	resafasa_FO.3.1'	0,37	0,71	0,50	0,74	1,00	0,57	0,35	0,38	0,45	0,34	0,57	0,34	0,31	0,44	0,44	0,48	0,43	0,38	0,46	0,34	0,26	0,40	0,10	0,03	0,07	0,01	0,17	0,05	0,07	0,10	0,00	0,02		0,03	0,07	0,15	0,03	0,01	0,14		0,03	0,05	0,00
6	resafasa_FO.3.2'	0,34	0,49	0,31	0,75	0,57	1,00	0,28	0,23	0,46	0,25	0,37	0,24	0,26	0,35	0,40	0,44	0,42	0,42	0,48	0,36	0,20	0,41	0,10	0,10	0,08	0,02	0,20	0,05	0,08	0,14	0,04	0,07	0,07	0,07	0,10	0,16	0,04	0,02	0,17	0,02	0,08	0,12	0,02
7	resafasa_FO.3.3.1'	0,47	0,35	0,20	0,42	0,35	0,28	1,00	0,67	0,63	0,67	0,63	0,67	0,28	0,66	0,56	0,52	0,34	0,59	0,47	0,46	0,15	0,35	0,28	0,07	0,11	0,11	0,30	0,10	0,11	0,16	0,05	0,07	0,01	0,10	0,27	0,33	0,08	0,11	0,31	0,05	0,05	0,08	0,05
8	resafasa_FO.3.3.2'	0,34	0,39	0,56	0,41	0,38	0,23	0,67	1,00	0,45	0,47	0,59	0,99	0,14	0,61	0,45	0,38	0,20	0,43	0,38	0,30	0,17	0,26	0,18	0,04	0,06		0,14	0,01	0,03	0,06		0,01	0,01	0,01	0,11	0,20	0,03	0,03	0,17		0,03	0,05	0,03
9	resafasa_FO.3.3.3'	0,40	0,36	0,18	0,64	0,45	0,46	0,63	0,45	1,00	0,58	0,58	0,43	0,43	0,51	0,59	0,56	0,47	0,63	0,64	0,59	0,20	0,47	0,27	0,14	0,14	0,17	0,38	0,15	0,20	0,22	0,08	0,15	0,08	0,17	0,29	0,34	0,10	0,10	0,33	0,12	0,11	0,16	0,03
10	resafasa_FO.3.3.4'	0,34	0,33	0,22	0,40	0,34	0,25	0,67	0,47	0,57	1,00	0,60	0,46	0,49	0,59	0,42	0,40	0,32	0,45	0,41	0,43	0,19	0,36	0,20	0,13	0,18	0,10	0,29	0,11	0,20	0,18	0,07	0,16	0,13	0,26	0,30	0,29	0,12	0,12	0,27	0,25	0,16	0,15	0,11
11	resafasa_FO.3.3'	0,40	0,58	0,34	0,58	0,57	0,37	0,62	0,58	0,58	0,60	0,99	0,56	0,42	0,90	0,55	0,45	0,40	0,44	0,45	0,44	0,24	0,54	0,25	0,10	0,14	0,05	0,28	0,07	0,11	0,13	0,02	0,08	0,03	0,08	0,17	0,29	0,07	0,06	0,27	0,05	0,07	0,12	0,04
12	resafasa_FO.3.4.2'	0,33	0,36	0,55	0,36	0,34	0,24	0,67	0,99	0,43	0,46	0,56	1,00	0,13	0,62	0,45	0,38	0,19	0,41	0,37	0,29	0,17	0,25	0,17	0,04	0,06		0,14		0,02	0,06		0,01	0,01	0,01	0,11	0,20	0,03	0,03	0,17		0,03	0,04	0,03
13	resafasa_FO.3.4.3'	0,27	0,27	0,08	0,31	0,31	0,26	0,28	0,14	0,43	0,49	0,42	0,13	1,00	0,41	0,35	0,32	0,37	0,29	0,30	0,31	0,16	0,38	0,12	0,08	0,17	0,11	0,27	0,11	0,20	0,18	0,08	0,18	0,13	0,27	0,26	0,20	0,12	0,10	0,19	0,28	0,15	0,13	0,09
14	resafasa_FO.3.4'	0,38	0,52	0,32	0,44	0,45	0,35	0,65	0,61	0,51	0,58	0,91	0,62	0,42	1,00	0,55	0,45	0,36	0,43	0,47	0,39	0,23	0,47	0,23	0,10	0,12	0,03	0,24	0,04	0,09	0,12	0,02	0,08	0,04	0,07	0,16	0,27	0,07	0,07	0,26	0,05	0,08	0,12	0,06
15	resafasa_FO.3.5.1'	0,53	0,38	0,22	0,48	0,44	0,40	0,56	0,46	0,59	0,43	0,55	0,45	0,35	0,55	1,00	0,65	0,55	0,56	0,58	0,48	0,13	0,49	0,25	0,07	0,16	0,09	0,43	0,11	0,13	0,20	0,08	0,11	0,01	0,07	0,21	0,27	0,05	0,06	0,28	0,03	0,08	0,12	0,01
16	resafasa_FO.3.5.2en3'	0,62	0,46	0,26	0,48	0,48	0,44	0,52	0,38	0,56	0,40	0,45	0,38	0,32	0,45	0,64	1,00	0,61	0,59	0,58	0,61	0,20	0,40	0,22	0,06	0,13	0,10	0,36	0,13	0,15	0,23	0,06	0,09	0,00	0,09	0,21	0,28	0,06	0,05	0,29	0,03	0,05	0,09	0,00
17	resafasa_FO.3.5'	0,38	0,27	0,18	0,42	0,43	0,42	0,34	0,20	0,47	0,32	0,40	0,19	0,37	0,36	0,55	0,61	1,00	0,37	0,50	0,39	0,14	0,52	0,17	0,04	0,12	0,13	0,35	0,16	0,17	0,25	0,07	0,10	0,00	0,11	0,18	0,20	0,06	0,06	0,22	0,04	0,03	0,09	0,00
18	resafasa_FO.3.6'	0,52	0,41	0,19	0,54	0,38	0,42	0,59	0,43	0,63	0,45	0,44	0,41	0,29	0,43	0,56	0,59	0,37	1,00	0,58	0,57	0,13	0,35	0,21	0,12	0,09	0,10	0,28	0,08	0,11	0,15	0,04	0,08	0,04	0,08	0,20	0,30	0,04	0,05	0,29	0,03	0,07	0,10	0,01
19	resafasa_FO.3.7'	0,39	0,39	0,28	0,51	0,46	0,48	0,47	0,38	0,64	0,41	0,45	0,37	0,30	0,47	0,58	0,58	0,50	0,58	1,00	0,50	0,19	0,45	0,14	0,05	0,06	0,06	0,19	0,05	0,06	0,07	0,03	0,04		0,03	0,10	0,17	0,02	0,02	0,18	0,01	0,03	0,07	0,01
20	resafasa_FO.3.8'	0,42	0,30	0,14	0,47	0,34	0,36	0,46	0,30	0,59	0,43	0,44	0,29	0,31	0,39	0,48	0,61	0,39	0,57	0,50	1,00	0,10	0,46	0,22	0,06	0,12	0,09	0,33	0,07	0,09	0,18	0,03	0,05		0,07	0,18	0,26	0,05	0,05	0,25	0,02	0,04	0,08	0,00
21	resafasa_FO.3.9'	0,14	0,27	0,22	0,24	0,26	0,20	0,15	0,17	0,20	0,19	0,24	0,17	0,16	0,23	0,13	0,20	0,14	0,13	0,19	0,10	1,00	0,15	0,06	0,04	0,08	0,04	0,05	0,03	0,07	0,05	0,02	0,06	0,07	0,10	0,09	0,10	0,08	0,07	0,10	0,13	0,07	0,05	0,10
22	resafasa_intro'	0,35	0,30	0,19	0,44	0,35	0,38	0,35	0,25	0,45	0,35	0,52	0,24	0,38	0,45	0,49	0,39	0,51	0,34	0,42	0,44	0,14	0,96	0,18	0,13	0,13	0,07	0,32	0,06	0,11	0,15	0,04	0,09	0,05	0,10	0,16	0,21	0,05	0,04	0,21	0,05	0,08	0,17	0,02
23	wwo_bep_hjaar_VCR'	0,23	0,12	0,10	0,17	0,10	0,10	0,28	0,18	0,27	0,20	0,25	0,17	0,12	0,23	0,25	0,22	0,17	0,21	0,14	0,22	0,06	0,17	1,00	0,05	0,09	0,30	0,39	0,28	0,25	0,32	0,12	0,14	0,01	0,26	0,38	0,33	0,13	0,15	0,35	0,13	0,03	0,08	0,05
24	wwo_bep_net_res'	0,07	0,09	0,01	0,15	0,03	0,10	0,07	0,04	0,14	0,13	0,10	0,04	0,08	0,10	0,07	0,06	0,04	0,12	0,05	0,06	0,04	0,14	0,05	1,00	0,58	0,32	0,16	0,56	0,72	0,11	0,62	0,78	0,85	0,50	0,40	0,51	0,50	0,50	0,36	0,54	0,86	0,91	0,40
25	wwo_bep_uitv_par'	0,14	0,09	0,05	0,11	0,07	0,08	0,11	0,06	0,14	0,18	0,14	0,06	0,17	0,12	0,15	0,13	0,12	0,09	0,06	0,12	0,08	0,13	0,09	0,58	1,00	0,31	0,22	0,58	0,63	0,15	0,63	0,65	0,69	0,57	0,55	0,78	0,86	0,57	0,50	0,60	0,71	0,67	0,66
26	wwo_berek_bedr_brut_bol5'	0,12	0,03	0,07	0,05	0,01	0,02	0,11		0,17	0,10	0,05		0,11	0,03	0,09	0,10	0,13	0,10	0,06	0,09	0,04	0,07	0,30	0,32	0,31	1,00	0,36	0,48	0,46	0,34	0,39	0,39	0,28	0,41	0,52	0,39	0,35	0,34	0,39	0,31	0,34	0,37	0,22
27	wwo_berek_bijzond_tab'	0,36	0,17	0,09	0,27	0,17	0,20	0,30	0,14	0,38	0,29	0,28	0,14	0,27	0,24	0,43	0,36	0,35	0,28	0,19	0,33	0,05	0,31	0,39	0,16	0,22	0,36	1,00	0,41	0,40	0,76	0,35	0,36	0,10	0,36	0,47	0,40	0,21	0,22	0,41	0,21	0,15	0,23	0,06
28	wwo_berek_lh_bijzond_tab'	0,13	0,04		0,09	0,05	0,05	0,10	0,01	0,15	0,11	0,07		0,11	0,04	0,11	0,13	0,16	0,08	0,05	0,07	0,03	0,07	0,28	0,56	0,58	0,48	0,41	1,00	0,94	0,40	0,81	0,76	0,65	0,62	0,61	0,58	0,58	0,58	0,49	0,59	0,64	0,64	0,43
29	wwo_berek_lh_norm_tab'	0,14	0,07	0,01	0,13	0,07	0,08	0,11	0,03	0,20	0,20	0,11	0,02	0,20	0,09	0,13	0,15	0,17	0,11	0,06	0,09	0,07	0,11	0,25	0,72	0,64	0,46	0,40	0,94	1,00	0,39	0,81	0,88	0,79	0,73	0,65	0,60	0,61	0,60	0,48	0,73	0,78	0,77	0,47
30	wwo_berek_norm_tab'	0,20	0,08	0,04	0,15	0,10	0,14	0,16	0,06	0,22	0,18	0,14	0,06	0,18	0,12	0,20	0,23	0,25	0,15	0,07	0,18	0,05	0,15	0,32	0,11	0,15	0,34	0,76	0,40	0,39	1,00	0,34	0,34	0,10	0,34	0,39	0,30	0,20	0,19	0,32	0,23	0,11	0,20	0,07
31	wwo_berek_oht_bijzond_tab'	0,06	0,00		0,04	0,00	0,04	0,05		0,08	0,07	0,02		0,08	0,02	0,08	0,06	0,07	0,04	0,03	0,03	0,02	0,04	0,12	0,62	0,63	0,39	0,35	0,81	0,81	0,34	1,00	0,92	0,75	0,62	0,57	0,54	0,61	0,56	0,40	0,62	0,75	0,71	0,44
32	wwo_berek_oht_norm_tab'	0,09	0,04	0,01	0,09	0,02	0,07	0,07	0,01	0,15	0,16	0,08	0,01	0,18	0,08	0,11	0,09	0,10	0,08	0,04	0,05	0,06	0,09	0,14	0,78	0,65	0,39	0,36	0,76	0,88	0,34	0,92	1,00	0,85	0,73	0,60	0,57	0,62	0,57	0,39	0,74	0,85	0,81	0,46
33	wwo_berek_prem_sv_tot'		0,03		0,06		0,07	0,01	0,01	0,08	0,13	0,03	0,01	0,13	0,04	0,01	0,00	0,00	0,04			0,07	0,05	0,01	0,85	0,69	0,28	0,10	0,65	0,79	0,10	0,75	0,85	1,00	0,72	0,54	0,51	0,64	0,68	0,42	0,83	0,96	0,88	0,53
34	wwo_berek_prem_sv_wao'	0,10	0,02	0,01	0,06	0,03	0,07	0,10	0,01	0,17	0,26	0,08	0,01	0,27	0,07	0,07	0,09	0,11	0,08	0,03	0,07	0,10	0,10	0,26	0,50	0,57	0,41	0,36	0,62	0,73	0,34	0,62	0,73	0,72	1,00	0,75	0,56	0,67	0,60	0,40	0,87	0,65	0,55	0,47
35	wwo_berek_prem_sv_ww'	0,22	0,08	0,03	0,14	0,07	0,10	0,27	0,11	0,29	0,30	0,18	0,11	0,26	0,16	0,22	0,21	0,18	0,20	0,10	0,18	0,09	0,16	0,38	0,40	0,55	0,52	0,47	0,61	0,65	0,39	0,57	0,60	0,54	0,75	1,00	0,62	0,63	0,58	0,54	0,61	0,61	0,57	0,39
36	wwo_berek_prem_sv_ww_VCR'	0,28	0,18	0,07	0,23	0,15	0,16	0,33	0,20	0,34	0,29	0,29	0,20	0,20	0,27	0,27	0,28	0,20	0,30	0,17	0,26	0,10	0,21	0,33	0,51	0,78	0,39	0,40	0,58	0,60	0,30	0,54	0,57	0,51	0,56	0,62	1,00	0,85	0,51	0,63	0,46	0,53	0,54	0,58
37	wwo_berek_prem_sv_ww_tbv_DU'	0,06	0,02	0,01	0,04	0,03	0,04	0,08	0,03	0,10	0,12	0,07	0,03	0,12	0,08	0,06	0,06	0,06	0,04	0,02	0,05	0,08	0,05	0,13	0,50	0,86	0,35	0,21	0,58	0,61	0,20	0,61	0,62	0,64	0,67	0,63	0,85	1,00	0,57	0,45	0,60	0,64	0,58	0,72
38	wwo_berek_prem_zorg'	0,06	0,01	0,00	0,03	0,01	0,02	0,11	0,03	0,10	0,12	0,07	0,03	0,10	0,07	0,06	0,05	0,06	0,05	0,02	0,05	0,07	0,04	0,15	0,50	0,57	0,34	0,22	0,58	0,60	0,19	0,56	0,57	0,68	0,60	0,58	0,51	0,57	1,00	0,80	0,73	0,66	0,61	0,47
39	wwo_berek_prem_zorg_VCR'	0,29	0,17	0,09	0,22	0,14	0,17	0,31	0,17	0,33	0,27	0,27	0,17	0,19	0,26	0,28	0,29	0,22	0,29	0,18	0,25	0,10	0,21	0,35	0,36	0,50	0,39	0,41	0,49	0,48	0,32	0,40	0,39	0,42	0,40	0,54	0,63	0,45	0,80	1,00	0,49	0,45	0,47	0,33
40	wwo_berek_prem_zorg_wao'	0,04	0,00		0,02		0,02	0,05		0,12	0,25	0,05		0,28	0,05	0,03	0,03	0,04	0,03	0,01	0,02	0,13	0,05	0,13	0,54	0,60	0,31	0,21	0,59	0,73	0,23	0,62	0,74	0,83	0,87	0,61	0,46	0,60	0,73	0,49	1,00	0,77	0,62	0,50
41	wwo_berek_tabloon_wao_samen'	0,05	0,05	0,01	0,09	0,03	0,08	0,05	0,03	0,11	0,16	0,07	0,03	0,15	0,08	0,08	0,05	0,03	0,07	0,03	0,04	0,07	0,08	0,03	0,86	0,71	0,34	0,15	0,64	0,78	0,11	0,75	0,85	0,96	0,65	0,61	0,53	0,64	0,66	0,45	0,77	1,00	0,93	0,51
42	wwo_bereken_bn'	0,09	0,09	0,03	0,15	0,05	0,12	0,08	0,05	0,16	0,15	0,12	0,04	0,13	0,12	0,12	0,09	0,09	0,10	0,06	0,08	0,05	0,17	0,08	0,91	0,67	0,37	0,23	0,64	0,77	0,20	0,71	0,81	0,88	0,56	0,57	0,54	0,58	0,61	0,47	0,62	0,93	1,00	0,44
43	wwo_lees_invoer_param'	0,01	0,00	0,00	0,01	0,00	0,02	0,05	0,03	0,03	0,11	0,04	0,03	0,09	0,06	0,01	0,00	0,00	0,01	0,01	0,00	0,10	0,02	0,05	0,40	0,66	0,22	0,06	0,43	0,47	0,07	0,44	0,46	0,53	0,47	0,39	0,58	0,72	0,47	0,33	0,50	0,51	0,44	1,00

Figure A.3: Mapping by indexing both systems