4-2014

Comparative Analysis of Requirements Change Prediction Models: Manual, Linguistic, and Neural Network

Beshoy Markos
Clemson University

James L. Mathieson
Clemson University, jmathie@clemson.edu

Joshua D. Summers
Clemson University, jsummer@clemson.edu

Research in Engineering Design

Comparative Analysis of Requirements Change Prediction Models: Manual, Linguistic, and Neural Network

--Manuscript Draft--

Manuscript Number:	RIED-D-12-00093R2
Full Title:	Comparative Analysis of Requirements Change Prediction Models: Manual, Linguistic, and Neural Network
Article Type:	Original Paper
Corresponding Author:	Joshua Summers Clemson University Clemson, SC UNITED STATES
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Clemson University
Corresponding Author's Secondary Institution:	
First Author:	Beshoy Morkos, PHD
First Author Secondary Information:	
Order of Authors:	Beshoy Morkos, PHD
	James Mathieson, MS
	Joshua Summers
Order of Authors Secondary Information:	
Abstract:	Requirement change propagation, if not managed, may lead to monetary losses or project failure. The a posteri tracking of requirement dependencies is a well-established practice in project and change management. The identification of these dependencies often requires manual input by one or more individuals with intimate knowledge of the project. Moreover, the definition of these dependencies that help to predict requirement change is not currently found in the literature. This paper presents two industry case studies of predicting system requirement change propagation through three approaches: manually, linguistically, and bag-of-words. Dependencies are manually and automatically developed between requirements from textual data and computationally processed to develop surrogate models to predict change. Two types of relationship generation, manual keyword selection and part of speech tagging, are compared. Artificial neural networks are used to create surrogate models to predict change. These approaches are evaluated on three connectedness metrics: shortest path, path count, and maximum flow rate. The results are given in terms of search depth needed within a requirements document to identify the subsequent changes. The semi-automated approach yielded the most accurate results, requiring a search depth of 11%, but sacrifices on automation. The fully automated approach is able to predict requirement change within a search depth of 15% and offers the benefits of full minimal human input.
Response to Reviewers:	Response to Reviewer's 2nd Comments In order for publication, the following changes are required: For clarity, we have highlighted the changes and crossed out the eliminated sections. Minor edits were not captured in this manner, only the significant changes are captured. Reviewer #1: How were the questions on page 6/lines 15-17 addressed by your results? There should be a section answering these questions directly at the end of paper. Author Response: This is addressed through the inclusion of an additional section in the Discussion. A paragraph is added to discuss how each research question is address in the case study. The DSM figures are fine to include, but they really need more explanation in the text

of the paper about why they are relevant. What does it mean that one DSM is sparser than another?
Author Response: This is clarified in the “Populating Requirement Change Models” section of the paper. The use of DSMs is presented and their illustrative purposes. Moreover, the relevance of DSM density is discussed.
Many of the citations are still inconsistent. Some include a first initial with last name, some don't, and the bibliographies of a few of the papers include long hyperlinks in them.
Author Response:All initials have been reviewed for correctness and completeness. The hyperlinks have been removed as well.
Reviewer #3:
Would really help, if a single example was used to show the different approaches.
Author Response: A single example is presented here to show the differences between manual and linguistic. This example utilizes the same requirement. To present a similar example for the bag of words approach is difficult to perform here because the bag of words approach uses a pattern recognition ANN to analyze the entire requirements document relevant words. It would be difficult to show the pattern recognition results without showing the entire requirements document and the word patterns identified.
There is a still a need to explain the relationship between requirements and changes clearer, since not all changes to requirements lead to changes in products and also not all changes arise from changes in requirements, if they are emergent changes from other products as shown in the work of Oli de Weck and Claudia Eckert. Related to that later more reflection why shared words in the requirements actually lead to changes. This does not necessarily detract from the usefulness of the proposed method, but being explicit will put the paper in the context.
Author Response: You are correct. Not all requirement changes lead to subsequent change propagations. Some changes are stagnant and don’t propagate to other requirements. In some instances, requirement change propagation occurs yet the effected requirements (related requirements) are able to absorb the propagation without warranting a subsequent change. For example, a requirement controlling temperature may increase and related requirements do not change because the material used is capable of surviving sufficiently at the increased temperature. A new section is added to the motivation section of the paper to discuss the important points you mention.
Regarding why shared words in requirements lead to changes; we found in most instances the shared words are either nouns or verbs (and keywords in the manual approach). This was found true in the keyword utilized manual approach and bag of words approach. In other words, we are effectively stated that requirements are related at either the physical or functional domain. Though it is unknown which type of words (or domains) have a greater significance on the ability to predict change propagation, the scope of this study was to identify varying approaches. We know the physical domain is effective because it is currently used to predict change propagation in systems where a physical architecture is defined. However, we use requirements because of their availability earlier in the design process. Nonetheless, you present an interesting question that we have already started to investigate. A section is added in the future works section of the paper to address your very question.
Say how you familiarised yourselves (p 7.) with the requirements.
Author Response: This is explained in greater detail in the paper.

Response to Reviewer's 2nd Comments

In order for publication, the following changes are required:

For clarity, we have highlighted the changes and crossed out the eliminated sections. Minor edits were not captured in this manner, only the significant changes are captured.

Reviewer #1:

How were the questions on page 6/lines 15-17 addressed by your results? There should be a section answering these questions directly at the end of paper.

Author Response*: This is addressed through the inclusion of an additional section in the Discussion. A paragraph is added to discuss how each research question is address in the case study.*

The DSM figures are fine to include, but they really need more explanation in the text of the paper about why they are relevant. What does it mean that one DSM is sparser than another?

Author Response*: This is clarified in the "Populating Requirement Change Models" section of the paper. The use of DSMs is presented and their illustrative purposes. Moreover, the relevance of DSM density is discussed.*

Many of the citations are still inconsistent. Some include a first initial with last name, some don't, and the bibliographies of a few of the papers include long hyperlinks in them.

Author Response*:All initials have been reviewed for correctness and completeness. The hyperlinks have been removed as well.*

Reviewer #3:

Would really help, if a single example was used to show the different approaches.

Author Response*: A single example is presented here to show the differences between manual and linguistic. This example utilizes the same requirement. To present a similar example for the bag of words approach is difficult to perform here because the bag of words approach uses a pattern recognition ANN to analyze the entire requirements document relevant words. It would be difficult to show the pattern recognition results without showing the entire requirements document and the word patterns identified.*

There is a still a need to explain the relationship between requirements and changes clearer, since not all changes to requirements lead to changes in products and also not all changes arise from changes in requirements, if they are emergent changes from other products as shown in the work of Oli de Weck and Claudia Eckert. Related to that later more reflection why shared words in the requirements actually lead to changes. This does not necessarily detract from the usefulness of the proposed method, but being explicit will put the paper in the context.

Author Response*: You are correct. Not all requirement changes lead to subsequent change propagations. Some changes are stagnant and don't propagate to other requirements. In some instances, requirement change propagation occurs yet the effected requirements (related requirements) are able to absorb the propagation without warranting a subsequent change. For example, a requirement controlling temperature may increase and related requirements do not change because the material used is capable of surviving sufficiently at the increased temperature. A new section is added to the motivation section of the paper to discuss the important points you mention.*

Regarding why shared words in requirements lead to changes; *we found in most instances the shared words are either nouns or verbs (and keywords in the manual approach). This was found true in the keyword utilized manual approach and bag of words approach. In other words, we are effectively stated that requirements are related at either the physical or functional domain. Though it is unknown which type of words (or domains) have a greater significance on the ability to predict change propagation, the scope of this study was to identify varying approaches. We know the physical domain is effective because it is currently used to predict change propagation in systems where a physical architecture is defined. However, we use requirements because of their availability earlier in the design process. Nonetheless, you present an interesting question that we have already started to investigate. A section is added in the future works section of the paper to address your very question.*

Say how you familiarised yourselves (p 7.) with the requirements.

Author Response*: This is explained in greater detail in the paper.*

Comparative Analysis of Requirements Change Prediction Models: Manual, Linguistic, and Neural Network

Beshoy Morkos

Department of Mechanical and Aerospace Engineering

Florida Institute of Technology

245 F.W. Olin Engineering

Melbourne, FL 32901

(321) 674 – 8780

bmorkos@fit.edu

James Mathieson

Department of Mechanical Engineering

Clemson University

134 Fluor Daniel Building (EIB)

Clemson, SC 29634-0921

(864) 656-7889 (LAB)

(864) 656-4435 (FAX)

jmathie@clemson.edu

Joshua D. Summers (corresponding author)

Department of Mechanical Engineering

Clemson University

250 Fluor Daniel Building (EIB)

Clemson, SC 29634-0921

(864) 656-3295 (PHONE)

(864) 656-4435 (FAX)

jsummer@clemson.edu

Requirement change propagation, if not managed, may lead to monetary losses or project failure. The a posteri tracking of requirement dependencies is a well-established practice in project and change management. The identification of these dependencies often requires manual input by one or more individuals with intimate knowledge of the project. Moreover, the definition of these dependencies that help to predict requirement change is not currently found in the literature. This paper presents two industry case studies of predicting system requirement change propagation through three approaches: manually, linguistically, and bag-of-words. Dependencies are manually and automatically developed between requirements from textual data and computationally processed to develop surrogate models to predict change. Two types of relationship generation, manual keyword selection and part of speech tagging, are compared. Artificial neural networks are used to create surrogate models to predict change. These approaches are evaluated on three connectedness metrics: shortest path, path count, and maximum flow rate. The results are given in terms of search depth needed within a requirements document to identify the subsequent changes. The semi-automated approach yielded the most accurate results, requiring a search depth of 11%, but sacrifices on automation. The fully automated approach is able to predict requirement change within a search depth of 15% and offers the benefits of full minimal human input.

Keywords:

requirement change

change propagation

engineering change

dependency modeling

complex system design

Comparative Analysis of Requirements Change Prediction Models: Manual, Linguistic, and Neural Network

REVIEW OF REQUIREMENTS CHANGE AND PROPAGATION

Design is a complex and dynamic process (Dym & Little 1999; Andreou et al. 2003; Cohen et al. 2000; Kannapan & Marshek 1992; Ottosson & Björk 2004) and, as a result, system requirements are not fixed initially in the development process. Moreover, a requirement change may result in unanticipated propagating dynamic changes (Ottosson & Björk 2004). Requirements are defined as the purpose, goals, constraints, and criteria associated with a design project. These requirements may range from the initial functional requirements to the detailed specifications (Chen 2006; Chen et al. 2007). It has been shown that more than half of a system's requirements will change before project completion (Kobayashi & Maekawa 2001; Ramzan & Ikram 2005), thus having significant influence on the design process (Nurmuliani et al. 2006). Requirements may change internally or externally (Clarkson et al. 2004; Giffin et al. 2009). For example, changes may be initiated by a redesign effort to reduce costs or changes may derive from new customer needs or market competition (Shankar et al. 2010). In many instances, decisions on accepting the changes must be made considering the volatility of requirements due to technology, trends, perceptions, and regulations changes (Spitas 2011; Vajna et al. 2005).

Unanticipated, improperly managed requirement changes can introduce negative consequences such as increased complexity (Chen 2006), data loss (Morkos et al. 2010), and wasted time and money (Morkos & Summers 2010; Morkos et al. 2012). An engineer might avoid these consequences if it were possible to make a quick, yet accurate, assessment about the overall effects of a requirement change before implementing (Ollinger & Stahovich 2004). The earlier in a product development process one is able to identify potential impacts from proposed requirement changes, the more significant the savings would be. *Thus, a requirement change prediction tool is needed to predict what subsequent changes might be seen if a change to a current requirement is accepted* (Kotonya & Sommerville 1992).

Change propagation is when one element of a system results in additional changes in the system when otherwise these new changes would not have been required (Eckert et al. 2004;

Giffin et al. 2009). In most instances of change propagation, the change initiating engineer is not aware of the propagation consequence (Cohen et al. 2000). Change propagation has been studied in the context of engineering changes, concurrent engineering models, product development, complexity, graph theory, and design for flexibility (Almefelt et al. 2006; Shankar et al. 2012; Chen et al. 2002). Many product development processes are executed with the anticipation that the requirements will change at a rapid rate, and therefore some requirements are not fully enforced early in the process (Ottosson 1996). Despite these impacts, managing, modelling, and *predicting* new changes has not been thoroughly researched (Sugden & Strens 1996; Harker et al. 1993; Lee et al. 2006). *More significantly, none target the use of requirements as the domain in which to predict the propagation.*

Methods for predicting change propagation in design are available for software engineering (Schach & Tomer 2000; Rajlich 2000) by decomposing a program into elements linked in a propagation graph. In mechanical design, such pieces may be subsystems or components. Such methods presume that a system architecture has been defined, rendering the method useless in early conceptual stages. Nonetheless, the technique of decomposing a system into pieces highlights where subsequent, immediate changes might be necessary, based on the relationships between elements. Furthermore, predicting change in complex systems such as automobiles is difficult as the consequences of change are often hard to predict, especially when subsystems cross boundaries (Ollinger & Stahovich 2004).

While explicit requirement changes are difficult to capture in historical industrial cases where no requirement change documentation tools are employed, the engineering changes (ECs) can be traced. It is through these ECs that one can trace back to a necessary change in the requirements document to determine the source of change. In this paper ECs are defined as (Shankar et al. 2012):

> *"An engineering change is an alteration made to parts, from embodiment design stage to production stage of the product life cycle, in its form or fit or function, drawing or software that has already been released. The change can be of any size or type, can involve any number of people, and can take any length of time."*

It is important to note engineering change has many definitions through many authors. For instance, Jarratt et al. defines engineering change as "defined as changes to parts, drawings or software that have already been released during the product design process, regardless of the scale of the change" (Jarratt et al. 2011). Both definitions focus on component changes made during the latter stages of the design process.

These changes may occur when companies request changes to products, documents, components, manufactured or purchased parts, processes, or even supplies (Chen et al. 2002). However, it is assumed, in this research, that all engineering change may be related back to a requirement or set of requirements that are affected by this change. For instance, a change in suspension travel may affect all requirements relating to the suspension system and the requirements of the systems near the spatial boundaries. Further, because a requirement is affected does not mean a requirement must change, as a requirement may be able to absorb change. The intellectual challenge to this problem lies in trying to predict new changes to requirements given an initial perturbation or requirement change.

RESEARCH MOTIVATION

Lost money is a prime motivator for this work as it results from the lack of preparing for propagated changes earlier (Morkos et al. 2012). In addressing this change, the authors examine two heterogeneous industry projects where engineering changes are predicted through requirement changes. A previous historical case study of the data management system of the engineered-to-order company explored how requirements were managed and where information was lost in the process (Morkos et al. 2010). A subsequent study by the authors identified how requirements change could have been predicted (Morkos et al. 2012). To further understand requirement change propagation, this paper presents a study examining the relationships between requirements. More importantly, this paper identifies what forms of requirement relationships are pertinent to change propagation so they may be computationally modelled for prediction purposes. Based on the relationship models, the topological connectedness can be explored through three primary measures: (a) shortest path, (b) path count, and (c) maximum flow capacity. Connectedness metrics are reviewed below and detailed in (Mathieson & Summers 2010). Identifying the important and significant connectedness metrics enhances the capability of the model to predict change propagation. The connectedness metrics are used develop relationships between requirements which in turn are used to to calculate a "search depth" for each approach. The results are given in terms of search depth required to identify the subsequent changes. This depth is given in the form of a percentage, measuring the percentage of the requirements document that must be searched to address the propagated requirement. As it is possible that a propagated requirement change may have the same rating value as other requirements, the search depth reported is the depth of the last element of the same ranking value as the propagated requirement change in question. The purpose of the search depth measure is to determine the performance of each approach in predicting change propagation. In doing so, the following research questions are addressed:

(1) Can requirements connectedness be a predictor of change propagation?

(2) What metrics of connectedness are accurate predictors of change propagation?

This study presents methods for developing the change propagation model, introducing an artificial neural net (ANN) approach for identifying effective relaters between requirements. It is hypothesized that highly connected requirements are more prone to change than those of low connectedness.

This study uses two industry projects totalling six engineering changes where four are predicted through three varying approaches. As is typical in case study research, patterns are sought that might be suggestive which, in turn, can serve as foundations for subsequent experimental studies (Yin 2003; Teegavarapu et al. 2008).

The studied company is housed in a 60,000 sq. ft. manufacturing facility located in Greenville, South Carolina. The products are primarily one-off automation, manufacturing, and field testing systems. The company performs its own design, fabrication, assembly, and installation while employing over sixty associates including engineers, project managers, and business managers. The number of associates involved and their role varies depending on the size and scope of the project. Each requirement document is written by the customer as a contract.

To identify change propagation, all available data pertaining to engineering change was analysed from the industry partner. Engineering change notifications (ECNs) forms are used by the engineers to detail when and what items require change. To initiate a change, the engineer authors an ECN and negotiates with the client until final change terms are agreed, essentially modifying the initial contract and the requirements within. A final ECN (seen in Figure 1) is approved, recorded, and sent to the engineers. The changes in both industry studies were initiated by the customer as requirements changes. The ECN contains date regarding the change, the originator of the change, a tracking number, the condition or reason for change, and a status of approval or rejection. To predict change propagation, the details needed for analysis within each ECN are the cause, date, and requirements affected. Some of the information within Figure 1 has been removed for proprietary reasons.

Date:	January 16, 2008	**ECN#:**	Company Line 3 Creel-01 Rev. 1
Customer:	Company	**Customer PO #:**	P42730-00
Project:	Line 3 Conversion Creel Only	**Approved [ ]**	**Rejected [ ]**
Client Signature:			
Comments:			
Change Notice Originated by:			John Smith
Condition or Reason which Resulted in the Change: **Client Initiating Change:**			Change in customer requirements
Brief Description of Change or Deviation from Scope: Replacement of manual tool for opening and closing of core locks with automated air locks. Includes independent control of 5 lower spindles.			
Estimated Impact on Engineering Schedule Delay none		**Explanation and breakdown:**	
Additional Engineering Expense		**Explanation and breakdown:**	
Engineering			$
Programming			$
Clerical			$
Additional Equipment/Installation Expense		**Explanation and breakdown:**	
Fabrication			$
Materials			$
Total Cost of this Change:			**$**

Figure 1: Corporation's Engineering Change Notification (ECN) Form

This research focuses on predicting changes within the requirements domain. As these are completed projects, the physical domain can be mapped to the requirements domain, identifying which subsystems and components are related to each requirement. As a result, when a component or subsystem change is made, requirements changed during the engineering change are identified. The ECN documents reason for change and, in all instances studied, "change in customer requirements" was the reason given with no further details, thus, introducing the need to infer the connections between the ECN and requirements document. The authors of the paper individually familiarized themselves with the requirements and determined which ECNs related to which requirement or requirement set. The authors were able to familiarize themselves with the requirements by repeatedly reading and analysing the change propagations and discussions with the industry member regarding the change propagations identified to ensure propagations did in fact occur. In this manner, with parallel analysis of the ECN to requirement mapping, the correlation is objectively developed through inter-rater agreement. The changes that occur in these case studies occur during various times of the design process. However, most of them occur during the detailed design process when most subsystem and component design have been finalized.

Due to the number of ECNs and requirement changes presented in this paper, the requirement changes will be documented as follows: ECN<Number>.<Project>.<Requirement Change Number>. For instance, ECN1.T.RC refers to the requirement change in engineering change number one in the Toho project. A requirement change followed by a numerical (i.e. ECN1.P.RC2) refers to the specific requirement in a series of requirements which changed during the engineering change. Between the two requirements documents, a total of 9,221 words are counted with 1,923 unique words found. The details for each are illustrated below.

Toho

The first project of interest is the Toho project in which a yarn roller assembly was developed for a local textile firm. The project spanned fifteen months and included fifteen managers, engineers, and business associates. The client provided a contract detailing 160 requirements; written in 3,691 words composed of 973 unique words. Three requirement changes were initiated at different times in the project (Table 1). In previous studies, all requirements that could have been affected by the change were analyzed, whereas in this study only the requirements that do change are analyzed. This is a more aggressive approach, but is a better indicator of the model's ability to predict propagation changes.

Table 1: Toho Project ECNs and Requirements Changed.

Approved ECN	Date	Requirements Changed
ECN1	16-Jan-2008	ECN1.T.RC
ECN3	2-Oct-2008	ECN3.T.RC
ECN4	7-Nov-2008	ECN4.T.RC

Pierburg

The Pierburg project entailed a requirements document with 214 total requirements written with 5,595 total words from 1,370 unique words. The project lasted eleven months with twelve associates. The project goal was to develop, build, and install multiple manufacturing stations to be used on an assembly line. The ECNs documented during the project were collected and analyzed as with the Toho project. Likewise, only the three requirements which changed as a result of the ECN were noted for analysis (Table 2).

Table 2: Pierburg Project ECNs and Requirements Changed

Approved ECNs	Date	Requirements Affected
ECN1	10-Jun-08	ECN1.P.RC1 ECN1.P.RC2 ECN1.P.RC3 ECN1.P.RC4
ECN7	15-Aug-08	ECN7.P.RC1 ECN7.P.RC2
ECN11	2-Sep-08	ECN11.P.RC

Comparing Requirement Change Prediction Approaches

This paper presents three different approaches to develop requirement change models. The first approach is based on manual creation of relationships between requirements by finding keywords of one requirement that are found in the text of a second requirement. The second approach is based on syntactic and linguistic parsing of the requirements. This approach defines relations between requirements through grammatical constructs. The final approach is a fully automated machine-learning based approach that creates relationships through the use of artificial neural networks. Each approach is illustrated through the two case studies of Toho and Pierburg. These are projects that are selected because they provide the researchers an omniscient view of the changes that occurred in the projects. This backward looking view of the projects allows the researchers to identify changes that occurred in the project to the requirements and then to predict the next change to the requirements that occurred. In this way, the researchers are able to simulate change prediction from the initial change in the requirement to a subsequent change. The requirements documents are detailed as they form the contractual agreement between the consulting design firm and the customer. Therefore, the performance of these requirement change prediction models is tested only on detailed requirements documents. Further research is needed to explore the potential of these methods to address requirements documents as they are being synthesized in even earlier stages of design. This is deemed out of scope for this paper.

Study Assumptions and Limitations

While this study utilizes system requirements to predict change propagation, it is important to note not all changes can be predicting through change propagation. In the instance where change propagation does occur, requirements may be used to predict change propagation. This is performed by developing a requirement relationship network that is conducive to predicting change propagation through the realized relationships. Not all requirement changes lead to subsequent change propagations as some changes are stagnant and don't propagate to other requirements. In some instance, requirement change propagation occurs yet the effected requirements (related requirements) are able to absorb the propagation without warranting a subsequent change. For example, a requirement controlling temperature may increase and related requirements do not change because the material used is capable of surviving sufficiently at the increased temperature. The change propagation instances shown here are confirmed by the industry member to have propagated and hence a retrospective analysis is performed to identify if the presented approach could have predicted the occurrence of this change.

POPULATING REQUIREMENT CHANGE MODELS

Here, three methods to populate requirement change relation models are presented: manual, linguistic, and automated "bag-of-words". Manual creation represents a traditional method of populating relationships between individual requirements through human decision making and provides a baseline by which to compare other two new methods. This method was previously used in predicting change propagation (Morkos & Summers 2010; Morkos et al. 2012). The remaining methods introduce new approaches for generating the prediction models. The linguistic approach uses an automated part-of-speech (POS) tagger to identify nouns and verbs supplemented with manually selected keywords for each requirement. The model links each requirement with other requirements sharing subjects, verbs, or keywords. Next, a fully automated "bag-of-words" method is presented which uses a pattern recognition ANN to prune the text to important connecting words through which relationships are drawn symmetrically between requirement instances. Each method increases in automation and, thereby, in objectivity of use. It should be recognized that these models are not evaluated for "correctness" but rather in effectiveness. These relationship models are not checked to determine if they capture specific meaning. They are compared to see how well these models can be used to predict requirement changes. Therefore, relationships cannot be classified as "true" or "false".

The requirement relationship models are presented through Design Structure Matrices (DSMs). The DSMs illustrated here are developed through the three approaches presented. The

DSMs are relevant in this research because they illustrate the relationships between requirements and their directionality. For instance, the manual approach will generate a symmetric DSM while the Linguistic and Bag of Words approach may generate asymmetric DSMs. Moreover, the population densities of the DSMs illustrate the relationship quantity between requirements. The goal of the various approaches is generate a DSM which highlights the needed relationships to predict the change propagation, yet decrease the total number of relationships as to reduce the number of false positive relationships.

Manual Creation

In the first method, requirements are related to one another through manually identified requirement subjects and keywords, both used in two separate case studies. In the first study, the Toho project, the relationships between requirements were formed based on similar subjects. The selection of the subject within each requirement is manual and may be subjective based on those tagging the subject in the requirement.

The second study used manually developed keywords to form the requirement relationship models of the Pierburg project. In the previous study with the subjects, there were deficiencies that could not be addressed. For example, an important requirement needing constant consideration may be a federal regulation. However, such a requirement may not have a specific subject; rather it possesses regulation or standard numbers. As a result, a means for relating a requirement outside of its subject is needed which is robust enough to capture information such as subjects, standards, and functions. Keywords were selected by reviewing the semantics of the requirements rather than their syntactical subject. An example of this is shown in a later section of the paper. This was performed by studying the requirements document and understanding how each requirement specifically affected the system design or which keywords may be pertinent to the requirement. By studying the requirements document, each requirement was tagged with a maximum of five keywords relevant to the requirement and the overall system. A total of 1,070 keywords were selected for the 214 requirements in the Pierburg project. Many of the keywords were duplicates as there were 407 unique keywords. While the selection of keywords is subjective, a set of common words were identified that might be used as "seed keywords" in future projects to reduce some subjectivity. Moreover, a study suggests that there is not a significant variance based on keywords generated by different users (Morkos 2012).

Assumptions

Inherent within this method is the subjectivity of the relations created, particularly with the keyword method. In the subject based relationship, it is assumed all requirements are related through a single subject. Though this is found to be a limitation as requirements may pertain to multiple subjects. It is assumed the keywords selected are sufficient in representing the important artefacts of the requirement and three keywords are sufficed. Further, it is also assumed that each requirement can support the needed number of keywords to build relationships.

Protocol

The subject based approach requires the engineer to review the requirements document and manually select the first subject, be it subsystems or components. For example, "yarn comb" is the subject of Requirements 9.3.9 and ECN4.T.RC:

> *9.3.9: A **Yarn Comb** for (22) ends shall be provided for each layer of bobbins.*
> *ECN4.T.RC: A **Yarn Comb** for (220) ends shall be provided for each of the two (2) PAN sheets.*

Due to the shared subject, these requirements are related. Once this is identified for each requirement, a model is created for shared subjects. The matrix model is both binary (identifying whether the subjects match or not) and symmetric (the relationships cannot be unidirectional as both subjects have to match). An extract of the subject based study model is shown in Figure 2; when requirements of the columns are related to those in the rows the cell is green. The requirements here are listed in a hierarchical manner where requirements with similar subjects are written adjacently, thus creating the clustered blocks in the final subject-based model for the Toho project (Figure 3).

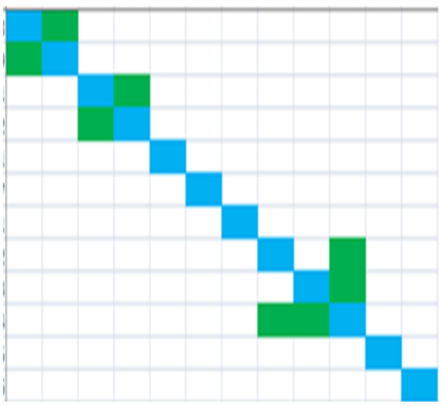


Figure 2: Small Segment of Model for Subject Based Study

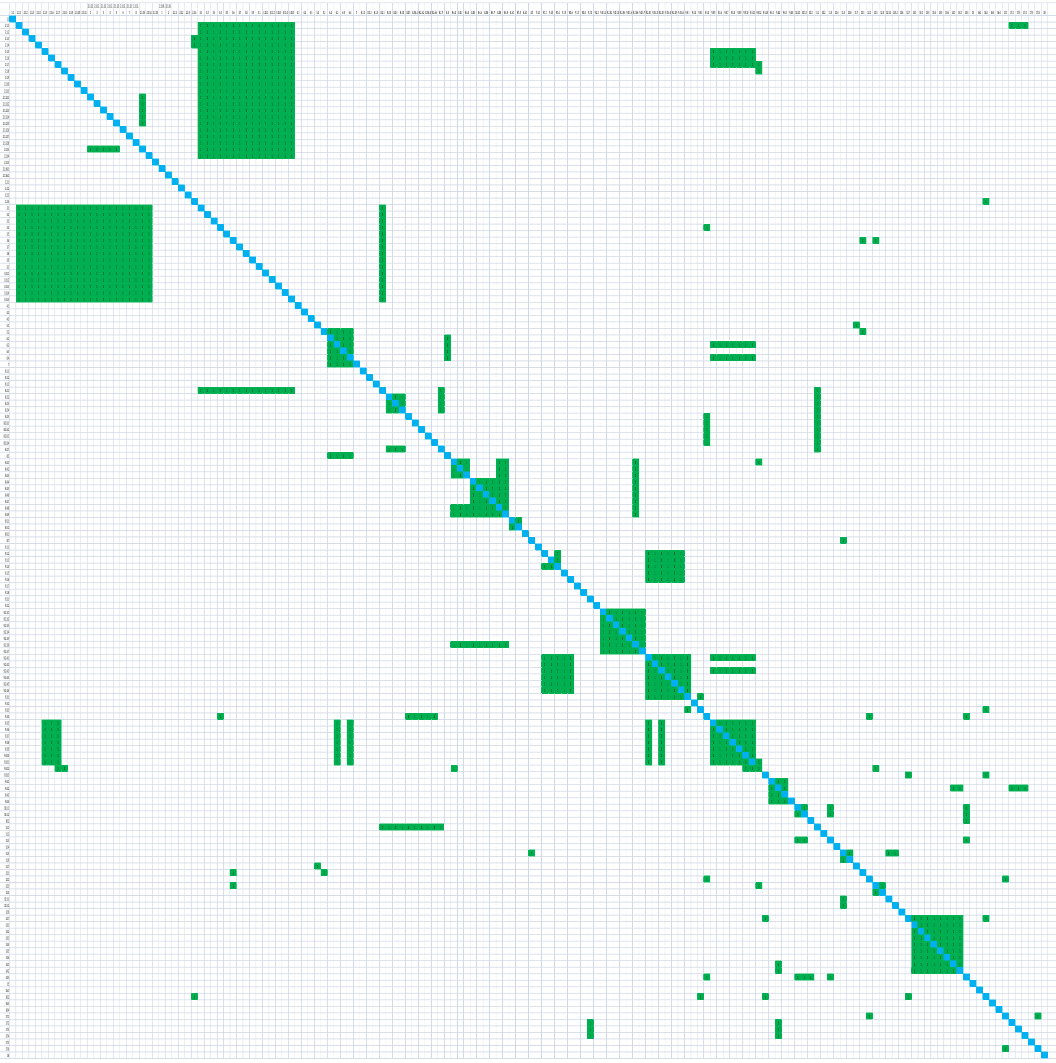


Figure 3: Toho Subject Based Manual Creation Study Model

Similarly, the keyword selection protocol is subjective, requiring the engineer to be familiar with the requirements document. Each requirement is reviewed at least once to understand the overall goal of the system and the major systems involved. After the initial review, the requirements are analyzed for potential keywords. To illustrate the keyword selection, consider the following example requirement from a project with each keyword underlined:

Requirement: Vibration^{KW1} *dampening*^{KW2} *level pads*^{KW3} *will be provided with a +/- 2-inch height*^{KW4} *adjustment*^{KW5} *capability.*

Vibration is selected as any system, subsystem, or component which experiences vibration could require dampening pads. Further, *level pads* were selected as this requirement affects this specific component of the subsystem and other components may also use level pads. *Dampening* was selected as a keyword addressing the working principle of the level pads as there may be other dampening mechanism which relate to this due to their shared objectives. *Height* is a keyword because of its overall dimensional effects on the system because of other spatial constraints that may be placed within other requirements. *Adjustment* is a keyword because it was important this system afford adjustability to satisfy the requirement. Though initially five keywords were selected, it was found only three keywords were needed as returns diminished significantly beyond three keywords. The three keywords selected were those which displayed the greatest frequency between other requirement keywords. Effectively, the keywords which appeared the most throughout the document were maintained. It is also important to recognize surplus keywords may result in superfluous relationships between requirements. Further justification and explanation for this can be found in the previous study performed (Morkos et al. 2012).

The requirement keywords are compared against the full text each of the other requirements to determine the relationship. A requirement may only be related to another requirement if at least one of its keywords exists within the text of the related requirement. Further, a requirement relationship does not vary in strength based on the number of keywords which match as it is a binary relationship. In contrast to the subject based relationship, the relationships are not all bidirectional resulting in an asymmetric model. For instance, the following requirements may have the keywords:

> *Example Requirement A: Tooling*^{KW1} *or fixtures*^{KW2} *switched during changeover*^{KW3} *shall attach to a sub-plate in accordance with "single minute exchange die" (SMED) design philosophies.*
>
> *Example Requirement B: Fragile*^{KW1} *Parts (Sensors, plastic parts, plastic gears etc..) or parts touching*^{KW2} *fragile parts (e.g. gear to gear assembly) must be assembled with tooling incorporating force control*^{KW3} *(and/or spring loaded mechanisms) to prevent part damage during the assembly.*

Tooling as the first keyword in Requirement Example Requirement A is found in the text of Requirement Example Requirement B. However, none of the keywords of Requirement Example Requirement B are found in the text of Requirement Example Requirement A. Thus, an asymmetric relationship between these requirements is established. Figure 4 illustrates the full relationship model for the Pierburg project based on the keyword protocol.

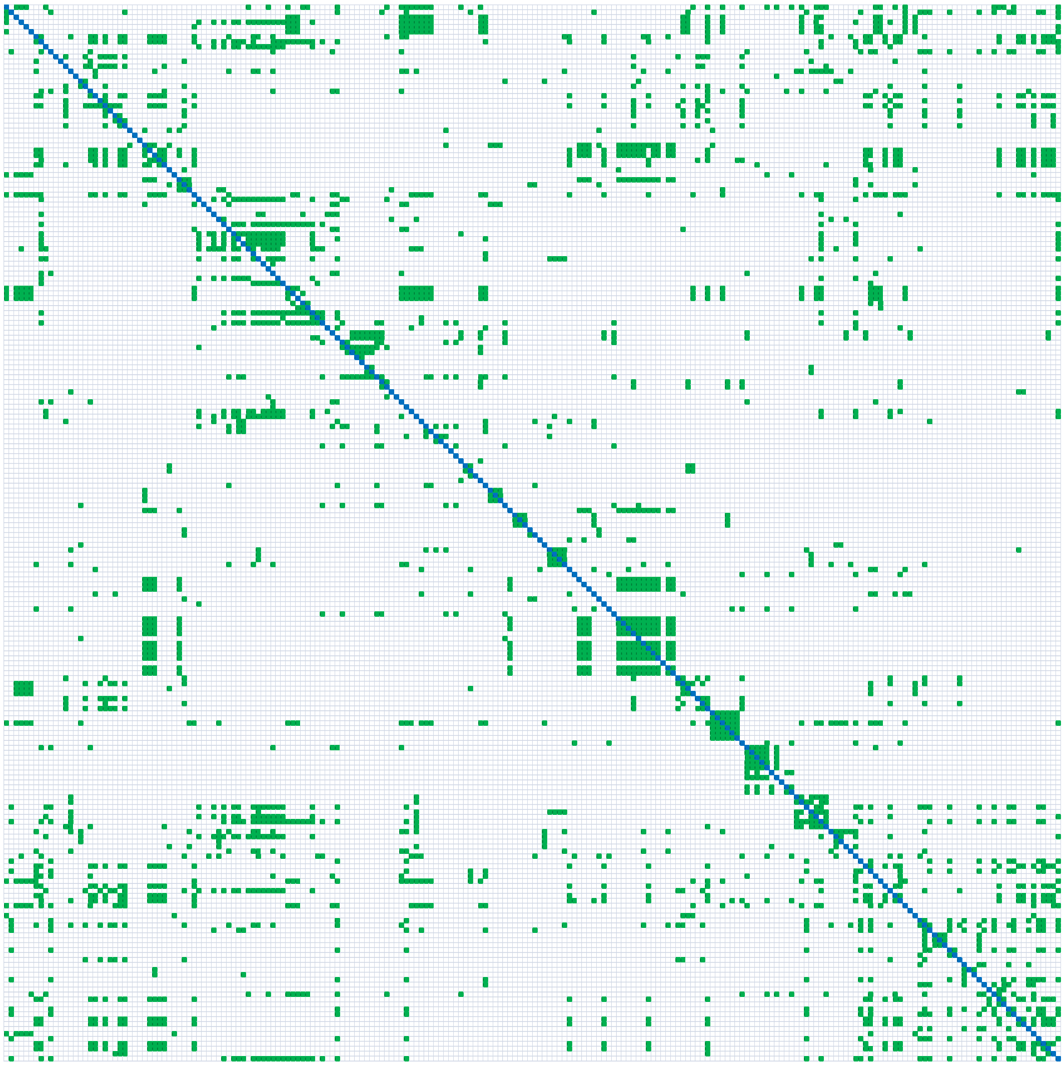


Figure 4: Pierburg Keyword Based Manual Creation Study Model

Linguistic Creation

The linguistic approach seeks to automate the promising process of the manual approach. The previous study recognized the potential for subjects and keywords to relate requirements in a manner conducive to predicting change propagation (Morkos et al. 2012). However, the selection is manual, tedious, and subjective. The linguistic approach uses a Part of Speech (POS) tagger to identify nouns and verbs within the requirement. Nouns were selected to maintain the subject based approach from the previous study and enhance it by implementing it in an automated manner. Verbs were introduced to identify if the function of the requirement could be used to relate requirements that shared similar functionality. To avoid overpopulation of the relationship models, only the first five nouns and verbs are selected for processing (Morkos 2012). The manual selection of keywords is maintained in this study to include words not tagged as nouns or verbs. A linguistic approach is possible because requirement sentences are not structurally different than other sentences (Lamar & Mocko 2010), allowing for a robust linguistic analysis.

Assumptions

The nouns and verbs selected are the first five to appear in a sentence. As a result, it is assumed the nouns pertinent to a requirement are those which are more likely to appear in the beginning of a sentence, not toward the end. Though the sequence of the set of nouns may not be important, the set of nouns used in the analysis are of significance. Though the use of manual keywords prevent the system from being completely automated, they assist in selecting those words, which may include additional nouns and verbs, that were not selected in the POS tagging. For instance, a pertinent noun may be of importance but resides as the eighth noun in the sentence. Since only five nouns are used, it is important this one is included through another avenue, such as keywords.

Protocol

The input is a requirements text document. It is important the model is robust enough to accept different types, both contextually and format, of requirements to ensure it is appropriate in all design applications. The requirements are parsed to extract all nouns and verbs to develop requirement relationships. Each requirement is tagged with a POS (Figure 5) and keywords are manually elicited. Parsing is performed through the Stanford POS tagger (Toutanova & Manning 2000) and passed into a MATLAB code to use the nouns and verbs for analysis, resulting in five nous, five verbs, and five keywords. Using the example shown in the previous approach, the following requirement and its POS output are examined:

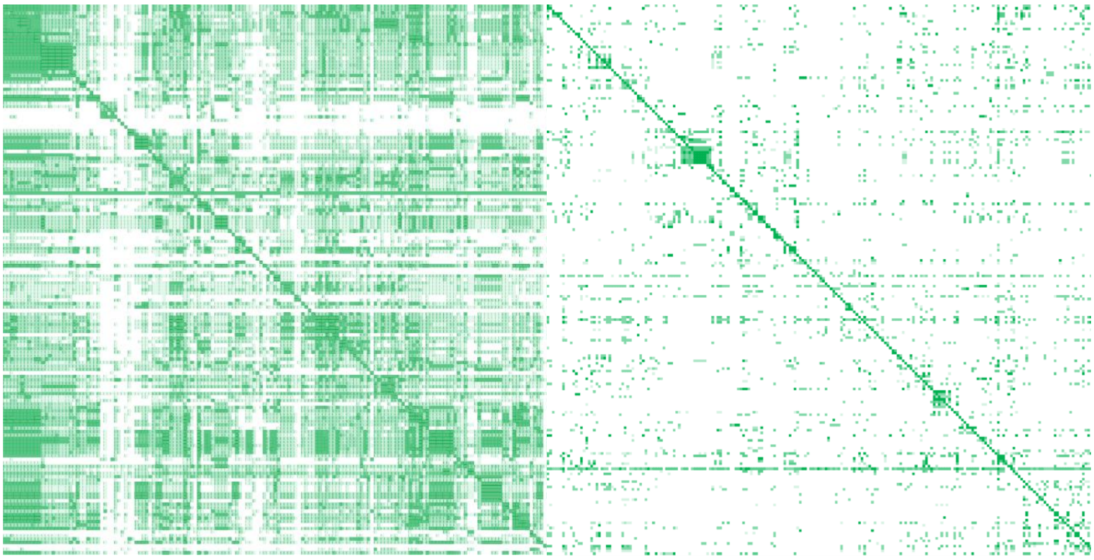
> *Example Requirement A: Tooling*[KW1] *or fixtures*[KW2] *switched during changeover*[KW3] *shall attach to a sub-plate in accordance with "single minute exchange die" (SMED) design philosophies.*
>
> ***Tooling/NN*** *or/CC* ***fixtures/NNS*** *switched/VBN during/IN* ***changeover/NN*** *shall/MD attach/VB to/TO a/DT sub-plate/JJ in/IN* ***accordance/NN*** *with/IN ``/`` single/JJ* ***minute/NN*** *exchange/NN die/VB "/" -LRB-/-LRB- SMED/NNP -RRB-/-RRB- design/NN philosophies/NNS*

As seen from the example, the first five nouns selected by the POS tagger are tooling, fixtures, changeover, accordance, and minute. Tooling, changeover, accordance, and minute were denoted as NN (singular noun) while fixtures was denoted as NNS (plural noun). More significantly, it is seen that the keywords selected might also be identified in the POS tagger as nouns and verbs.

```
(ROOT
  (S
    (NP
      (NP (NN Tooling)
        (CC or)
        (NNS fixtures))
      (VP (VBN switched)
        (PP (IN during)
          (NP (NN changeover)))))
    (VP (MD shall)
      (VP (VB attach)
        (PP (TO to)
          (NP (DT a)
            (ADJP (JJ sub-plate)
              (PP (IN in)
                (NP
                  (NP (NN accordance))
                  (PP (IN with)
                    (NP (`` ``)
                      (S
                        (NP (JJ single) (NN minute) (NN exchange))
                        (VP (NN die)))
                      ('' '')
                      (PRN (-LRB- -LRB-)
                        (NP (NNP SMED))
                        (-RRB- -RRB-)))))))
            (NN design) (NNS philosophies)))))
    (. .)))
```

Figure 5: Example POS Requirement Tagging

The syntactically extracted POS, alongside the selected keywords are used to develop relationship models for both Toho and Pierburg (Figure 6). It was identified that not all nouns, verbs, and keywords are needed to accurately predict requirement propagation (Morkos 2012). Further, excessive relaters may cause the model to overpopulate, making it difficult to accurately predict change propagation. Overpopulate here refers to the saturation of the DSM due to several false positive relationships. As a result, a part of this study, though not presented in this paper, was to identify which sets of nouns, verbs, and keywords are the most efficient in predicting change propagation. Such relationships could be based on nearly thirty three thousand possible noun, verb, and keyword relationship combinations ($2^{(5 \text{ nouns} + 5 \text{ verbs} + 5 \text{ keywords})} = 2^{15} = 32,768$). A separate study (Morkos 2012) developed a quality metric algorithm to determine the effectiveness of each relationship based on the project type and propagation characteristic of the requirements. An optimized combination is selected to provide the greatest requirement change propagation prediction accuracy, which is measured through specific performance metrics. The results of the analysis identified three combinations which consistently and accurately could be used for predicting engineering change propagation on multiple, varying design projects. The combinations made use of the first two nouns to appear in a requirement and a keyword (not similar to the nouns). All three combinations were used by overlaying the models developed for each type of combination, hence creating a single, combined relation model for this approach. The relationships are not developed on nouns-to-keyword or keywords-to-nouns, but rather they based on whether nouns or keyword are found in the full requirement text. This is similar to the manner in which relationships were developed in the keyword manual creation approach. As a result, the matrix model developed is asymmetric.



(a) Toho (b) Pierburg

Figure 6: Linguistic Creation Model for Pierburg and Toho

Automated Bag of Word Creation

In this approach, the text of each requirement is distilled down into a set of connecting words with each word representing a means by which two or more requirements can be related. This distillation process is conducted irrespective of the actual content of each word string (requirement), using instead the properties of each word in the context of two graph constructions of the requirement document text to identify words which should be removed from consideration. The graph constructions create a representation of the sequence of words and the word content of each requirement respectively. After the distillation process, the latter of these two approaches, a "bag-of-words" model, is reversed to change from words related through requirements to requirements related through words. This roughly approximates the human process of identifying keywords for drawing connections.

Assumptions

This method makes similar probabilistic assumptions as made in latent semantic indexing (LSI) and probabilistic LSI is terms of dimensionality reduction (Deerwester et al. 1990; Hofmann 2001). This method applies the "bag-of-words" assumption that neglects the order of words to relate requirements. This is an assumption of exchangeability of the words within each requirement (Aldous 1985). Further, the order of the requirements within the requirements document is likewise neglected. However, the exchangeability assumption is only applied to words when connecting requirements, not in the process of identifying words for removal.

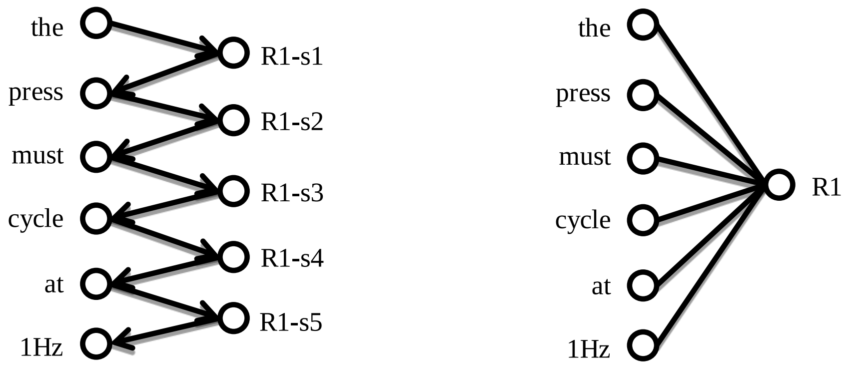
The removal process underlies the second major assumption of this method; there are extraneous words in the text which do not create meaningful connections. Prime examples of this would be articles, such as "a", "the", and "this", as well as forms of "to be", such as "is", "are", and "were". These words might be removed with pre-processing, but are pruned from the corpus through the single automated "bag-of-word" system to ensure objectivity and complete automation. It is desirable that user interaction by designating certain custom "stopwords" in creating the relations does bias the system. For this reason, pre-processing to clean the documents of extraneous words was not done here. Words such as these occur with high frequency, but it would be naive to relate two requirements on the basis of both containing the word "must", especially in a requirement document. In identifying these words in the document, it is assumed that the position of words is not exchangeable and also that the words within a given requirement share that requirement as an additional relating factor. It is further assumed that the nature of the relationships, analogous to context, vary from one document to the next, precluding the use of a pre-trained approach.

Protocol

Per the given assumptions, after basic parsing to remove extraneous characters and suffixes, the sequential and "bag-of-words" graph constructions of the text are analysed with to established graph theoretic approaches. The results of this analysis establish a characteristic vector describing each word for use in artificial neural network (ANN) pattern recognition of words to remove. The remaining words in each requirement are subsequently used to draw connections between requirements, forming the model.

The first step in this process, parsing of raw requirement document text, is based on a series of deterministic rules which progressively alter the text. The majority of these rules exist to remove extraneous characters such as punctuations, hyphens, tabs, or carriage returns that might cause later errors. Another key function of this step is to remove suffixes, such as tense and plurality, reducing to a root word. For example, a document may contain references to “hammer”, “hammers”, and “hammering”. In this case, all instances of these words are converted into “hammer”. Further, words representing units following a value are conjoined with the value to form a single string. This prevents two requirements from being marked as related by containing the same units (kg) but allowing relations between repeated references to the same unit-value pair (25kg).

Once the text has been parsed into consistent words, these are arranged into the two b-partite graph constructions of sequential and “bag-of-words” (Figure 7). The element set in both cases is the list of unique words in the document. In the sequential construction, the relationship set is representative of the spaces between words, while in the “bag of words” construction the relationship set represents each of the requirements. A further difference in the constructions is that edges in the sequential construction are directed with the flow of the text and the “bag-of-words” construction is undirected. For computational efficiency, words that do not exist in more than one requirement are excluded from the graphs.



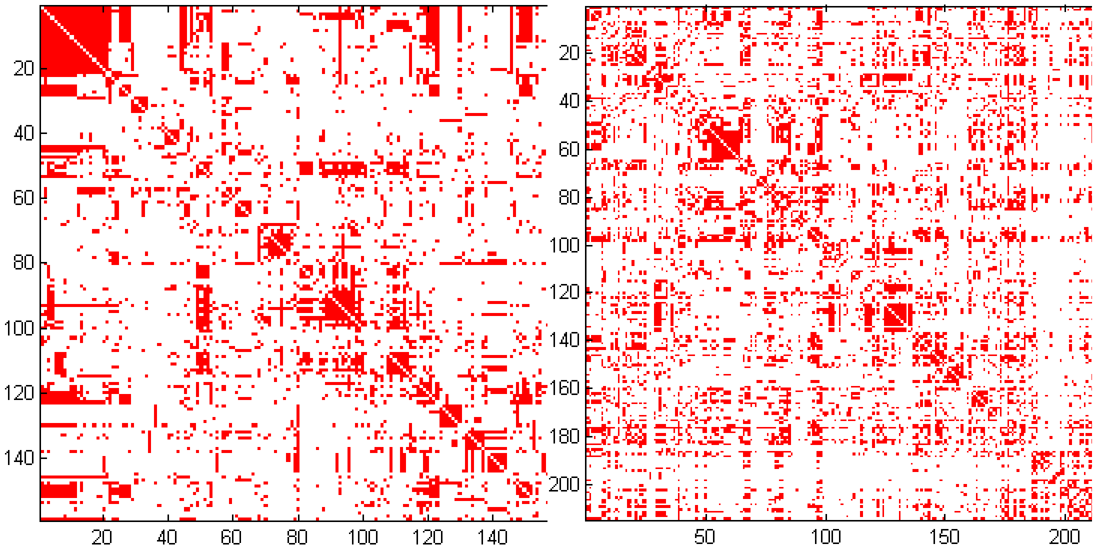
(a) sequential graph (b) “bag-of-words” graph

Figure 7: Examples of graph constructions

For both of the graphs, the relationship set is treated as a definition of hyperedges connecting the words in the element set for the purpose of extracting graph properties which characterize each word’s role in the structure of the graph. The properties analyzed for each word are degree, betweenness, clustering coefficient, in-core number, out-core number, minimum cycle size, and maximum cyclic flow rate (Mathieson & Summers 2010). It is not known *a priori* which of these properties are critical to the model construction, so a vector of 29 metrics is used. Betweenness is a measure of the number of shortest paths passing through a given node (Freeman 1977). The clustering coefficient is the proportion of a node's neighbours connected to the other neighbors (Watts & Strogatz 1998). In/out core numbers represent the threshold at which the given node becomes disconnected when all nodes of that degree are removed (Bader & Hogue 2003). The shortest unique path that connects a node to itself defines the minimum cycle size (Pramanick & Ali 1994). Maximum cyclic flow rate is the network capacity for looping back to the given node (Goldberg & Tarjan 1986; Goldberg & Tarjan 1988).

These properties, taken from both graphs, are combined with the length of the given word string to form a characteristic vector describing each word. This vector, with the normalized values for each metric is the input to a pattern recognition ANN of 30 tan-sigmoid neurons. This ANN is trained by comparing the word list against a reference list of example words that have been marked as either a term to keep or a term to eliminate. The intersection of the current word list and the reference list forms the training set that is presented to the ANN. Training is conducted using Levenberg-Marquardt training in conjunction with Bayesian regularization. Bayesian regularization is used to ensure repeatable generalization of the patterns underlying unwanted words. This results in lower variation in results than with early stopping generalization techniques and precludes the need to divide the training set into training and validation sets. Training is limited to 150 iterations and a gradient of 0.1 on the sum square error performance metric used in Bayesian regularization. The method is implemented in MATLAB with the ANN toolbox.

After training, the input vectors for the entire word list are applied to the ANN model. The output of the ANN is a value between zero and one, with a value of one representing absolute certainty that the given word should be removed. An aggressive removal approach is used by removing words assigned a value over 0.4. Next, the process of constructing the graphs and training the ANN are repeated until no words are indicated for removal, the training set becomes monotonic, or there is no longer an intersection between the current and reference word lists. Any remaining intersection between the current word list and words marked for removal in the reference list are accordingly removed from the current list. Finally, the "bag-of-words" graph is reversed such that words are treated as the edges connecting the requirements (elements) and represented in an adjacency matrix model (Figure 8).



(a) Toho (b) Pierburg

Figure 8: Automated “bag-of-words” output Models

EXPERIMENT ON CONNECTEDNESS FOR REQUIREMENT CHANGE PROPAGATION

The three requirement relationship modeling approaches are evaluated against two industry projects where change propagation occurred, with no a priori knowledge. It is hypothesized that connectedness can be used as an indicator of change propagation, where propagated requirements could have been predicting through previous, highly connected changed requirements. Finally, multiple metrics, rooting from graph theory, exist to measure requirement to requirement connectedness.

Discussion on Measures of Connectedness

Each requirement is a node within an adjacency matrix model. Graph theory offers several means by which the level of connection between two nodes can be evaluated. The majority of these measures are derived from the paths that exist between each pair of nodes. A path is a sequence of connected nodes and edges between two nodes such that no node or edge is repeated. Three measurements on the path properties between nodes are addressed using Figure 9 as a reference: shortest path length, path count, and maximum flow capacity.

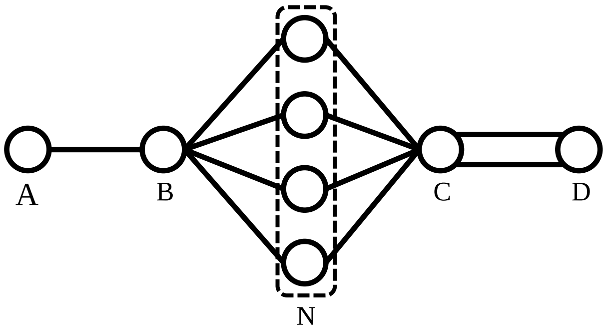


Figure 9: Example Graph

The shortest path length between two nodes is a measure of the minimum number of edges that must be traversed to travel from one node to another. For example, nodes A and B have a shortest path length of one as they are directly connected. However, the shortest path length between nodes A and D is four as edges AB, BN, NC, and CD must be traversed between them.

Path count is the combinatorial evaluation of the number of possible paths between any two nodes (Berge 1976). Considering Figure 9, there is only one possible path between A and B, while between A and C there are as many possible paths as there are nodes in layer N that act as intermediaries. Further, as there are two edges between C and D, the path count between A and D will be 2N.

The last measure, maximum flow capacity, is a network property on the amount of information that can flow between two nodes at any one time. Assuming the capacity of all nodes and edges to be equal, this becomes a measure of the maximum number of paths that may be simultaneously active between the two nodes. For example, while there are N possible paths between A and C, only one of these can be used at a time due to the singular connection between A and B. Likewise, the maximum flow capacity between B and D is limited to two by the double connection between C and D.

Metrics to be evaluated

The goal of this research is to be able to determining the likelihood that a change to a given requirement or set will result in a change to other requirements. This therefore implies a ranking of requirements. These rankings are evaluated against observed instances of propagated change by the depth of search required to identify those requirements that were observed to have changed. As there may be more than one propagated change, this requires that each ranking method be evaluated on the profile of % search depth versus % of propagated changes identified. This section discusses the ranking method applied to each model population approach.

Manual

The initial manual approach used a simple evaluation of relationship order to identify if requirements related. As a preliminary study, it was important to identify if change propagation could be predicted using a relationship between the initial requirement and the subsequently changed requirement. If a relationship exists, the order of this relationship is noted. The output is a simple relationship order. Further, to differentiate between potential requirements for propagation, path count rankings are developed and such requirements are sorted. For added granularity, requirements with high path counts have a higher chance of propagation than those of low path counts.

Linguistic

Recognizing the importance of requirement relationship order from the manual creation study, the linguistic approach results extends requirement order analysis by incorporating weightings. A 9-3-0 scale is used for weighting relationships where a first order relationship had a score of 9 and a second order a score of 3. Third order and beyond are not score because of their insignificance and oversaturation of the model in previous studies (Morkos et al. 2012).

The linguistic approach propagates requirement change through all previous requirement changes, cumulatively. Whereas one change may cause another change, it is important to propagate change that occurs from a multiple changes. In doing this, propagated changes are analyzed by predicting change resulting from all previous changes. Consider the Pierburg study: ECN1 (Requirements ECN1.P.RC1, ECN1.P.RC2, ECN1.P.RC3, ECN1.P.RC4) and ECN7 (Requirements ECN7.P.RC1and ECN7.P.RC2) could have cumulatively propagated to ECN11 (Requirement ECN11.P.RC). Further, a single requirement from ECN1 and ECN7 could have propagated to ECN11. To ensure the significance of the requirement which caused the change is not minimized or treated equally with other requirements in the analysis, a root mean square (RMS) approach is taken. RMS is used here to maximize the significance of first order relationships between previously changed requirements while incorporating the other requirements which may not be of great significance but must be maintained. The scoring for this propagation is calculated as followed in Equation 1. Each previous requirement changed is RMS summed to find a propagation score.

$$S_{Propagation} = \sqrt{\frac{{W_{R1}}^2 + {W_{R2}}^2 + {W_{R3}}^2 + \dots + {W_{Rn}}^2}{n}} \qquad \textit{Equation 1}$$

Where:

S = Propagation Score for potentially propagated requirement

W_{Ri} = Weighting of previous requirements i based on relationship order

Using the example from Pierburg, Equation 1 is simplified to Equation 2.

$$S_{\text{ECN11.P.RC}} = \sqrt{\frac{(W_{\text{ECN1.P.RC1}})^2 + (W_{\text{ECN1.P.RC2}})^2 + (W_{\text{ECN1.P.RC3}})^2 + (W_{\text{ECN1.P.RC4}})^2 + (W_{\text{ECN7.P.RC2}})^2}{5}} \qquad \textbf{Equation 2}$$

Each requirement is scored, based on RMS, to determine the likelihood of propagating forward. The results are used to analyze the score ranking of the requirements and the depth of review an engineer would have had to do before reaching the requirement which next changed. For instance, if a requirements document has 200 requirements and the propagated requirement scored as the 18 highest, this means the designer or engineer has to review a depth of 9% of the requirements before finding the changed requirement.

Automated Bag of Words

The automated "bag-of-words" method generates a symmetric matrix model. This results in less gradation of shortest path lengths than in the asymmetric model generated in the linguistic method. As such, an additional measure is used with shortest path length to add granularity; maximum flow capacity. A greater capacity between two requirements should imply a higher level of connection. A longer shortest path length between two requirements is believed to reduce the level of connection. Thus, the connectedness between requirements for this case is evaluated as the quotient of maximum flow capacity over shortest path length. A given change can be evaluated for the vector of the changed requirement against all requirements. When a given instance of change involves more than one requirement, the mean value is taken.

RESULTS OF CONNECTEDNESS ANALYSES ON CREATION APPROACHES

Three approaches are analysed based on the protocol given for each. The result of each approach is given and a comparison for each approach is discussed to compare the means for developing requirement relationships conducive to change propagation. Four change propagations, stemming from the two projects studied are evaluated in each approach.

Manual

The manual creation metrics measure the order at which requirements are related, such as first and second order relationships, and the number of such relationships (path count). This study is thoroughly reported in (Morkos et al. 2012) and is used as a baseline comparison with the other two proposed methods.

Toho

The results of the Toho manual creation study indicated through the 160 requirements, all engineering changes could be predicted through a maximum relationship order of two (second order) where 43 requirements must be evaluated for possible change propagation. Though it was found that second order was the greatest relationship order needed to predict propagation, an engineer would have to review 27% of the requirements document. There was no further granularity evaluated for this approach on the Toho project. The Pierburg project added this granularity by incorporating relationship order path count ranking.

Pierburg

To further distinguish between requirements, a relationship ranking is developed. The Pierburg requirements document contained 214 requirements and the ranking was developed to compare the results of change propagation between the multiple requirements highlighted. The ranking gives insight as to the strength of relationship, based on the path count, compared to other requirements. A requirement may be related to another requirement in the second order through multiple intermediate requirements, increasing the number of second order relationship paths.

Four requirements changed during the first engineering change, ECN1. Two requirements change during ECN7 (Requirements ECN7.P.RC1and ECN7.P.RC2). A limitation to this model is it cannot predict self-propagation, which is an instance when a requirement continues to change at a later time. The propagated requirement, ECN7.P.RC2, was ranked as the 13^{th} highest relationship path count. With this propagation approach a designer would review at least 13 of the 214 requirements to find the propagated requirements, a depth of 6%.

To propagate to the final engineering change, ECN11, all previous requirement changes are evaluated to identify if change could have been predicted. The final requirement which changes is Requirement ECN11.P.RC. Due to previous changes, Requirement ECN11.P.RC has the 16th highest path count from previous changes. With this approach, to identify the propagated change in Requirement ECN11.P.RC, an engineer would review at least 16 requirements or 7.5% depth.

Linguistic

The linguistic approach made use of the scoring metric used for the linguistic approach. Each propagated requirement was scored with respect to the all the previous requirement changes. The score was developed as an RMS approach to add significance to those requirements related in the first order and add separation to those relationships relating in second and third order. This score was ranked amongst the rest of the requirements document similarly scored. A search depth, based on the score ranking, is given to evaluate the capability of the approach to predict change propagation.

Toho

The first propagation analyzed was from ECN1 to ECN3; Requirement ECN1.T.RC to Requirement ECN3.T.RC. After a change in Requirement ECN1.T.RC, the scoring system ranked requirement ECN3.T.RC as the 16th ranked requirement to change. This meant the engineer would review a depth of 10% of the requirements document. The second change occurred to ECN4, affecting requirement ECN4.T.RC. Using the information gathered from previous changes (both ECN1 and ECN3) and scoring them results in a propagation scoring ranked 5th. At such a ranking an engineer would need to review a depth of only 3.1% of the requirements document. Thus, the linguistic method and scoring system used for the Toho project would require an engineer to review, at most, 10% of the requirements document to identify change propagation.

Pierburg

The first Pierburg ECN caused a change in four requirements. A subsequent change occurred two month later related to two requirements. Unlike previous examples, two requirements are propagated here. The propagated requirements, ECN7.P.RC1and ECN7.P.RC2, are scored based on the previous changes. Requirement ECN7.P.RC1is found to score 7^{th} highest amongst all requirements while requirement ECN7.P.RC2 is the highest, indicating it was the most likely to change. For this specific change, an engineer would review at least 7 requirements, 3.3% of the requirements document. The next engineering change affected a single requirement. Using the six previous requirements changed (from both ECNs), this requirement was found to score the 23^{rd} highest score amongst other requirements or 11% depth of the document. Overall, for the entirety of a project, the linguistic approach and scoring method used were sufficient as long as the engineer or designer addressed the top 11% requirement scores.

Bag-of-Words

In this section, the performance results for the automated "bag-of-words" methods are presented. Also addressed is the position of the ranking value with respect to the overall range of ranking values by normalizing the range to between zero and one, with zero representing the minimum value and one representing the maximum value.

Toho

The results for the Toho project (Table 3) indicate a capacity to predict the propagated changes. The largest search depth observed is between ECN3 and ECN4 at 14.5% of requirements. An interesting effect is seen in that there is a lower search depth required to identify ECN4 from ECN1 than from ECN3. This is suggestive of a propagation chain which was not directly sequential.

Table 3: Toho Search Depth Results

		Propagated To	
		ECN3	**ECN4**
		ECN3.T.RC	ECN4.T.RC
Initiated By	**ECN1**	13.2%	8.2%
	ECN3		14.5%

However, the suggestion of a difference in propagation between ECN1 and ECN3 to ECN4 is called into question by the ranking value assigned to the requirement changed in ECN4 (Table 4). For the ranking from both ECN1 and ECN3, the requirement changed in ECN4 is assigned the maximum value in the ranking. Therefore, the variation in search depth between ECN1 and ECN3 is caused by the number of other requirements assigned the maximum value.

The connection between ECN1 and ECN3 is less strong by the virtue that the ranking value assigned to the requirement changed in ECN3 is only given a normalized score of 0.857.

Table 4: Toho Normalized Ranking Value Results

		Propagated To	
		ECN3	**ECN4**
		ECN3.T.RC	ECN4.T.RC
Initiated By	**ECN1**	0.857	1.000
	ECN3		1.000

Pierburg

The results of the Pierburg study are shown in Table 5. ECN7 contains more than one changed requirement. As such, both of these requirements, and results pertaining to them, are shown under ECN7. The multiple requirements are not divided in the initiation rows as the ranking method specifies that multiple requirements be combined by taking the mean of each possible propagation. The results show an ability to indicate the potential change propagations as a result of a given change, requiring no more than 11.2% to be reviewed. The low search depths required to identify ECN11 is indicative of both a highly connected requirement that is sensitive to change as well as a fine granularity of ranking values without large equivalent zones.

Table 5: Pierburg Search Depth Results

		Propagated To		
		ECN7		**ECN11**
		ECN7.P.RC1	ECN7.P.RC2	ECN11.P.RC
Initiated By	**ECN1**	11.2%	9.3%	3.3%
	ECN7			1.9%

This is supported when looking at the normalized ranking values presented (Table 6). Only a single pairing is assigned the maximum value in the range, with all others coming in the top quarter. This is another example of a variation in the level of connectedness between requirement change instances. However, in this case the variation is captured though a fine gradation of ranking values. The fact that the upper quarter of the value range exists within the top 11% of rankings suggests that predictive lists of possible change propagations could be filtered by normalized ranking value as opposed to a set depth of the requirement list.

Table 6: Pierburg Normalized Ranking Value Results

Propagated To		
ECN7		**ECN11**
ECN7.P.RC1	ECN7.P.RC2	ECN11.P.RC

Initiated By	**ECN1**	0.781	0.852	0.890
	ECN7			1.000

Comparison of results

Three approaches are used to predict requirement change propagation, each varying in level of automation. The manual approach yielded results that indicated an engineer would have to search a depth of 27% if no ranking system is used and all first and second order relationships are addressed. When a scoring system is introduced, ranking the path count for each requirement, the search depth is decreased to 7.5%. The linguistic uses an intricate scoring system to amplify first order relationships without sacrificing second order relationships. Further, a semi-automatic approach is presented for developing requirement relationships. Using this approach, favourable results are achieved as the maximum search depth for the Toho and Pierburg projects are 10% and 11% respectively. Finally, the fully automated "bag-of-words" approach yields a search depth of 14.5% and 11.2% for the Toho and Pierburg projects, indicating an efficient means for depicting requirement change propagation at a lower human capital cost.

Table 7: Comparison of Results (amount of complete sorted requirement list required to process to find subsequent change)

	Project	
Relationship Approach	**Toho**	**Pierburg**
Manual	27%	7.5%
Linguistic	10%	11%
Bag of Words	14.5%	11.2%

Of the three approaches, the linguistic and the "bag-of-words" both provide sorted lists of requirements that would necessitate the engineers to only examine between 15-10% of the requirements in the documents. These results are better than the manual approach, on average. More importantly, the level of automation for these two approaches is significantly higher than the manual approach. The "bag-of-words", having performance results similar to the linguistic, but being fully automated appears to have the greatest potential for further study and refinement.

DISCUSSION

Two research questions, presented in the Research Motivation section, are addressed in this case study. The first research question pertains to the ability of requirement relationships to predict change propagation. Three approaches are presented here to predict four instances of change propagation. The results indicate all approaches are capable of using requirement relationships to predict change propagation, though with varying levels of accuracy. The second research question addresses the various connectedness metrics which are most accurate at predicting change propagation. As each approach resulted in a varying level of accuracy, it is seen in this study that the linguistic approach, where requirements are related based on their noun and verb POS, are most accurate at predicting change propagation. Hence, the use of automated POS taggers are currently the most accurate means for developing requirement relationships most conducive to predicting change propagation.

Four engineering change propagation situations are predicted in two separate industry projects using three different requirement relationship approaches. The results of the approach find that a semi-automated or automated means for predicting requirement change propagation is possible. The linguistic and bag-of-words approach are both modelled using MATLAB and can be extended to a full, stand-alone computational reasoning program. Observations are made from this study that contributes to change propagation practice and research. A primarily contribution of this work is the realization of (semi)automated approaches which perform more efficiently than that of the baseline, human approach of building relationships. While the manual approach can yield varying results, the linguistic and bag of words approach results in consistent results.

An important observation is that path counts can discriminate results from the second order relation sets. First order path counts are limited to one as there is only one way a requirement may be directly related to another requirement. However, at the second order, path counts increase significantly as many requirements may serve as intermediate requirements. The greater number of path counts indicates a greater number of intermediate requirements between the related requirements. These intermediate requirements serve as possible avenues for change propagation. While first order relationships are of chief importance, second order must be incorporated into the scoring system. Further, third order relationships were of little importance and distort the rankings from the model. This requires further analysis to determine if this relationship order phenomenon is consistent with other types of projects. Though the linguistic approach was found to be the most accurate and lowest search depth, the disadvantage of such a technique is the manual selection of keywords. This keyword selection may be costly in terms of time analysis. Therefore, a trade-off between accuracy and cost presents two options: the accurate linguistic approach and the fully automated bag-of-words approach.

Regardless, such findings indicate the importance of requirements in design as they can be used as a tool to predict change propagation. From a research perspective, the possibility to use requirements for change prediction and management is demonstrated. Such research further strengthens the foundation of requirements in the design process.

Implications on Practice

This study showed that it is possible for an automated, or semi-automated, model to predict engineering change through requirement change propagation. Such a tool could have an immediate impact on how engineering change is handled, modelled, and analyzed. Many monetary and time losses are involved with change propagation, as was identified when performing this analysis for the corporation. Currently, no tool exists to aid in mitigating some of the issues involved with change propagation. The methods presented here could have an immediate impact on change practice and how system requirements are handled throughout the design process.

Future Work

There are several avenues for extending this research. At the fundamental level, we aim to understand why shared words between requirements allow for the prediction of change propagation. Utilizing all three approaches, most requirement relationships are developed based on nouns and verbs. In other words, we are effectively stating that requirements are related at either the physical or functional domain. Though it is unknown which type of words (or domains) have a greater significance on the ability to predict change propagation, the scope of subsequent studies may be to explore this. We know the physical domain is effective because it is currently used to predict change propagation in systems where a physical architecture is defined. However, we use requirements because of their availability earlier in the design process.

Of particular interest is to extend the use of pattern recognition ANN into the identification of connections between requirements to build on data generated in the "bag-of-words" method. This approach requires a more robust set of known change instances than is currently available. Another approach of interest is the possibility that the keyword segment of the linguistic approach may be omitted. Results presented in this paper suggest that this may be possible with little loss in performance.

In each of these methods, the words found within the requirements are essentially treated as symbols and are not part of a more comprehensive semantic ontology. Therefore, there are opportunities for extending this research to include semantics that would allow researchers to add weights to relationships between requirements based on word similarity. For instance, synonyms, hyponyms, and other semantic relations between words would possible add context and meaning beyond the simple binary relations of the current approaches.

Beyond the realm of requirement change prediction, the methods presented here may also present an opportunity to expand into the analysis of connections within the text of other documents. This may include other structured documents, such as process descriptions, as well as unstructured documents, such as email communications. Such extensions offer the possibility of new methods of knowledge capture and predictive capabilities.

REFERENCES

Aldous, D., 1985. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII Lecture Notes in Mathematics*, 1117, pp.1–198.

Almefelt, L. et al., 2006. Requirements management in practice: findings from an empirical study in the automotive industry. *Research in Engineering Design*, 17(3), pp.113–134.

Andreou, A.S., Zographos, A.C. & Papadopoulos, G.A., 2003. A three-dimensional requirements elicitation and management decision-making scheme for the development of new software components. In *Proc. of the Fifth International Conference On Enterprise Information Systems (ICEIS)*. Angers, France, pp. 3–13.

Bader, G.D. & Hogue, C.W. V., 2003. An Automated Method for Finding Molecular Complexes in Large Interaction Network. *BMC Bioinformatics*, 4(2).

Berge, C., 1976. *Graphs and hypergraphs*, New York, NY: North-Holland Publishing Company.

Chen, Y.M., Shir, W.S. & Shen, C.Y., 2002. Distributed engineering change management for allied concurrent engineering. *International Journal of Computer Integrated Manufacturing*, 15(2), pp.127–151.

Chen, Z.Y., 2006. Classification of Product Requirements Based on Product Environment. *Concurrent Engineering*, 14(3), pp.219–230.

Chen, Z.Y. et al., 2007. Formalisation of product requirements: from natural language descriptions to formal specifications. *International Journal of Manufacturing Research*, 2(3), pp.362–387.

Clarkson, P.J., Simons, C. & Eckert, C., 2004. Predicting Change Propagation in Complex Design. *Journal of Mechanical Design*, 126(5), pp.788–797.

Cohen, T., Navathe, S.B. & Fulton, R.E., 2000. C-FAR, change favorable representation. *Computer Aided Design*, 32(5), pp.321–338.

Deerwester, S. et al., 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), pp.391–407.

Dym, C.L. & Little, P., 1999. *Engineering Design: A Project-Based Introduction*, New York, NY: John Wiley and Sons.

Eckert, C., Clarkson, P.J. & Zanker, W., 2004. Change and customisation in complex engineering domains. *Research in Engineering Design*, 15(1), pp.1–21.

Freeman, L., 1977. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1), pp.35–41.

Giffin, M. et al., 2009. Change Propagation Analysis in Complex Technical Systems. *Journal of Mechanical Design*, 131(8), p.081001.

Goldberg, A. V. & Tarjan, R.E., 1986. A New Approach to the Maximum Flow Problem. In *Annual ACM Symposium on Theory of Computing*. New York, NY: ACM, pp. 136–146.

Goldberg, A. V. & Tarjan, R.E., 1988. A new approach to the maximum flow problem. *Journal of the ACM*, 35(4), pp.921–940.

Harker, S.D.P., Eason, K.D. & Dobson, J.E., 1993. The change and evolution of requirements as a challenge to the practice of software engineering. In *Proceedings of IEEE International Symposium on Requirements Engineering*. San Diego, CA: IEEE, pp. 266–272.

Hofmann, T., 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1), pp.177–196.

Jarratt, T.A.W. et al., 2011. Engineering change: an overview and perspective on the literature. *Research in Engineering Design*, 22(2), pp.103–124.

Kannapan, S.M. & Marshek, K.M., 1992. A schema for negotiation between intelligent design agents in concurrent engineering. In *Intelligent Computer Aided Design*. North Holland: Elsevier Science Publishers, pp. 1–25.

Kobayashi, A. & Maekawa, M., 2001. Need-based requirements change management. In *Proceedings. Eighth Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems*. Washington, DC: IEEE, pp. 171–178.

Kotonya, G. & Sommerville, I., 1992. Viewpoints for requirements definition. *Software Engineering Journal*, 7(6), pp.375–387.

Lamar, C. & Mocko, G., 2010. Linguistic Analysis of Natural Language Engineering Requirement Statements. In *Tools and Methods for Competitive Engineering*. Ancona, Italy.

Lee, H.J. et al., 2006. Capturing and reusing knowledge in engineering change management: A case of automobile development. *Information Systems Frontiers*, 8(5), pp.375–394.

Mathieson, J.L. & Summers, J.D., 2010. Complexity Metrics for Directional Node-Link System Representations: Theory and Applications. In *Proceedings of the ASME International Design Engineering Technical Conferences*. Montreal, Quebec, Canada, pp. 13–24.

Morkos, B., 2012. Computational representation and reasoning support for requirements change management in complex system design. *Ph.D. Dissertation, Clemson University*.

Morkos, B. et al., 2010. Requirements and Data Content Evaluation of Industry In-House Data Management System. In *Proceedings of the ASME International Design Engineering Technical Conferences*. Montreal, Quebec, Canada, pp. 493–503.

Morkos, B., Shankar, P. & Summers, J.D., 2012. Predicting requirement change propagation, using higher order design structure matrices: an industry case study. *Journal of Engineering Design*, 23(12), pp.905–926.

Morkos, B. & Summers, J.D., 2010. Requierment Change Propagation Prediction Approach Results from an Industry Case Study. In *ASME International Design Engineering Technical Conferences*. Montreal, Quebec, Canada, pp. 111–121.

Nurmuliani, N., Zowghi, D. & Williams, S.P., 2006. Requirements volatility and its impact on change effort: Evidence-based research in software development projects. In *Australian Workshop on Requirements Engineering*. Adelaide, Australia.

Ollinger, G.A. & Stahovich, T.F., 2004. RedesignIT—A Model-Based Tool for Managing Design Changes. *Journal of Mechanical Design*, 126(2), pp.208–216.

Ottosson, S., 1996. Dynamic product development: findings from participating action research in a fast new product development process. *Journal of Engeering Design*, 7(2), pp.151–169.

Ottosson, S. & Björk, E., 2004. Research on dynamic systems—some considerations. *Technovation*, 24(11), pp.863–869.

Pramanick, I. & Ali, H., 1994. Analysis and Experiments for a Parallel Solution to the All Pairs Shortest Path Problem. In *IEEE International Symposium on Circuits and Systems*. New York, NY, pp. 479 – 482.

Rajlich, V., 2000. Modeling software evolution by evolving interoperation graphs. *Annals of Software Engineering*, 9(1), pp.235–248.

Ramzan, S. & Ikram, N., 2005. Making decision in requirement change management. In *International Conference on Information and Communication Technologies*. Karachi, Pakistan: IEEE, pp. 309–312.

Schach, S.R. & Tomer, A., 2000. A maintenance-oriented approach to software construction. *Journal of Software Maintenance: Research and Practice*, 12(1), pp.25–45.

Shankar, P., Morkos, B. & Summers, J.D., 2010. A Hierarchical Modeling Scheme With Non Functional Requirements. In *ASME Design Engineering Technical Conferences*. Montreal, Quebec, Canada: ASME, pp. 283–296.

Shankar, P., Morkos, B. & Summers, J.D., 2012. Reasons for change propagation: a case study in an automotive OEM. *Research in Engineering Design*, 23(4), pp.291–303.

Spitas, C., 2011. Analysis of systematic engineering design paradigms in industrial practice: A survey. *Journal of Engineering Design*, 22(6), pp.427–445.

Sugden, R.C. & Strens, M.R., 1996. Strategies, tactics and methods for handling change. In *IEEE Symposium and Workshop on Engineering of Computer-Based Systems*. Friedrichshafen, Germany, pp. 457–463.

Teegavarapu, S., Summers, J.D. & Mocko, G., 2008. Case study method for design research: A justification. In *ASME International Design Engineering Technical Conferences*. Brooklyn, NY, USA, pp. 495–503.

Toutanova, K. & Manning, C.D., 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *SIGDAT conference on Empirical methods in natural language processing and very large corpora*. Hong Kong, China: Association for Computational Linguistics, pp. 63–70.

Vajna, S. et al., 2005. The autogenetic design theory: an evolutionary view of the design process. *Journal of Engineering Design*, 16(4), pp.423–440.

Watts, D.J. & Strogatz, S., 1998. Collective Dynamics of "Small-World" Networks. *Nature*, 393(6), pp.440–442.

Yin, R., 2003. *Case Study Research: Design and Methods*, Thousand Oaks, CA: Sage.

Figure

Click here to download high resolution image

Date:	January 16, 2008	**ECN#:**	Company Line 3 Creel-01 Rev. 1
Customer:	Company	**Customer PO #:**	P42730-00
Project:	Line 3 Conversion Creel Only	**Approved [ ]**	**Rejected [ ]**
Client Signature:			
Comments:			
Change Notice Originated by:			John Smith
Condition or Reason which Resulted in the Change:			Change in customer requirements
Client Initiating Change:			
Brief Description of Change or Deviation from Scope: Replacement of manual tool for opening and closing of core locks with automated air locks. Includes independent control of 5 lower spindles.			
Estimated Impact on Engineering			
Schedule Delay		**Explanation and breakdown:**	
none			
Additional Engineering Expense		**Explanation and breakdown:**	
Engineering			$
Programming			$
Clerical			$
Additional Equipment/Installation Expense		**Explanation and breakdown:**	
Fabrication			$
Materials			$
Total Cost of this Change:			$

Figure

Click here to download high resolution image

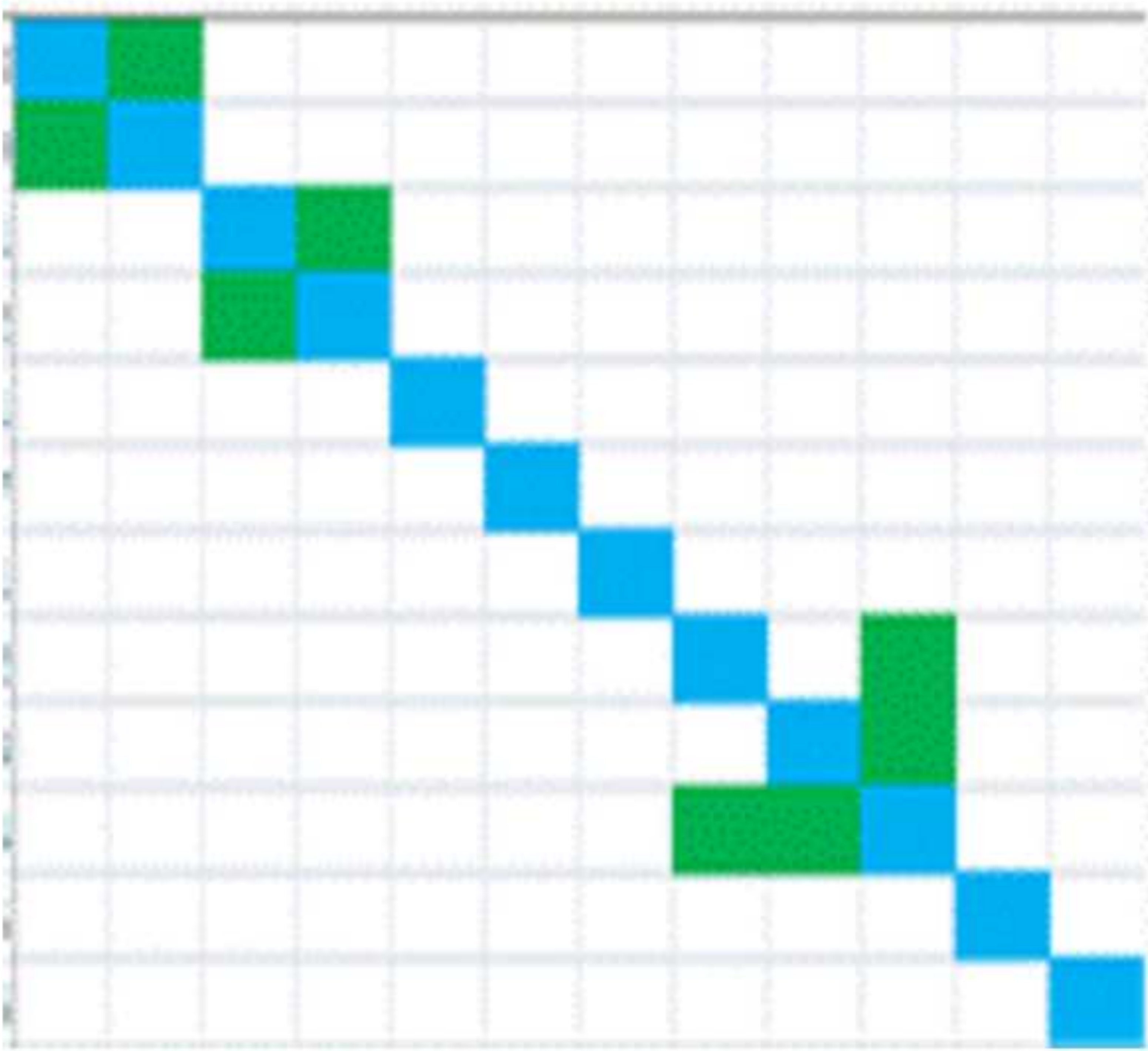


Figure
Click here to download high resolution image

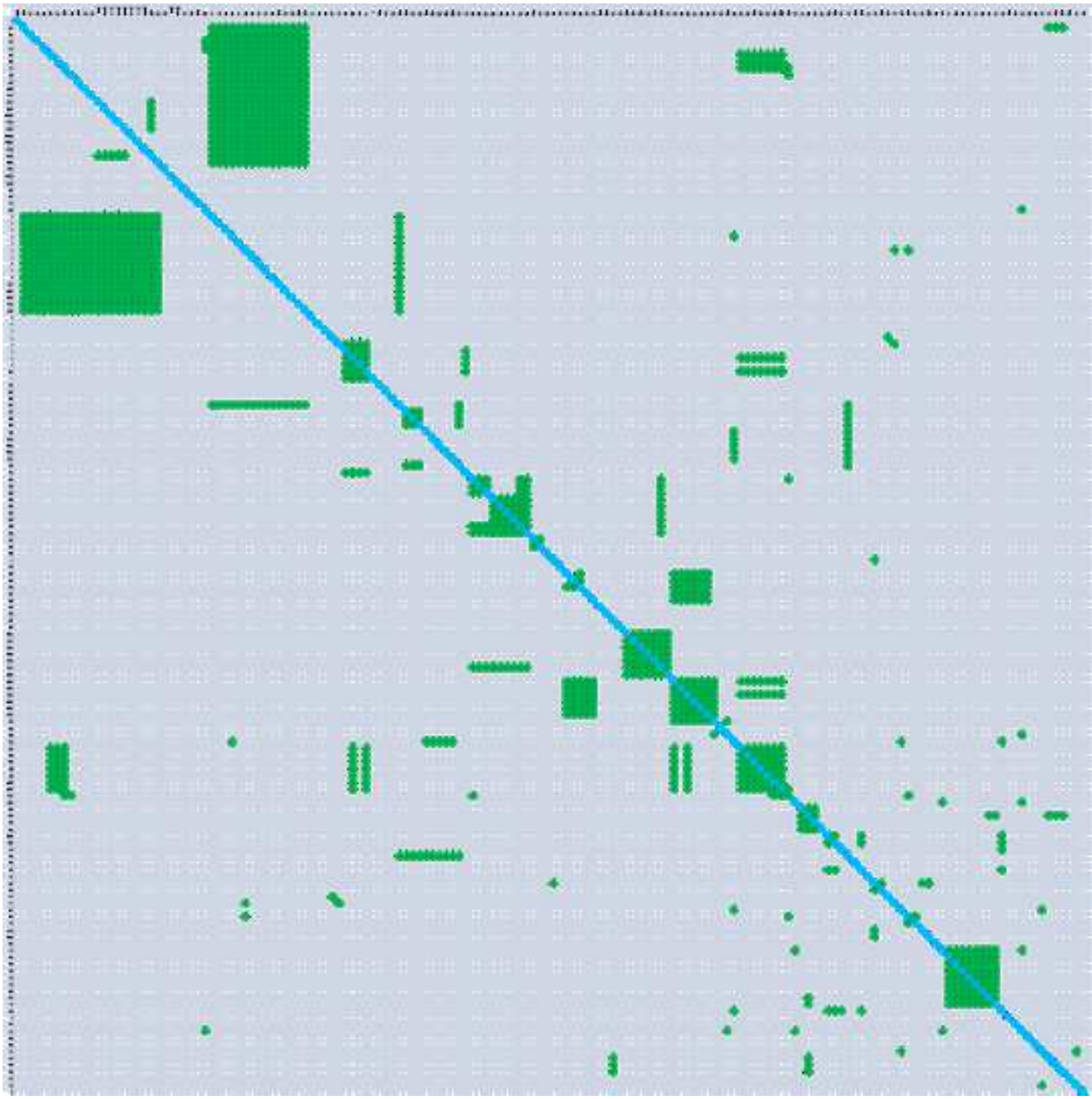


Figure
Click here to download high resolution image

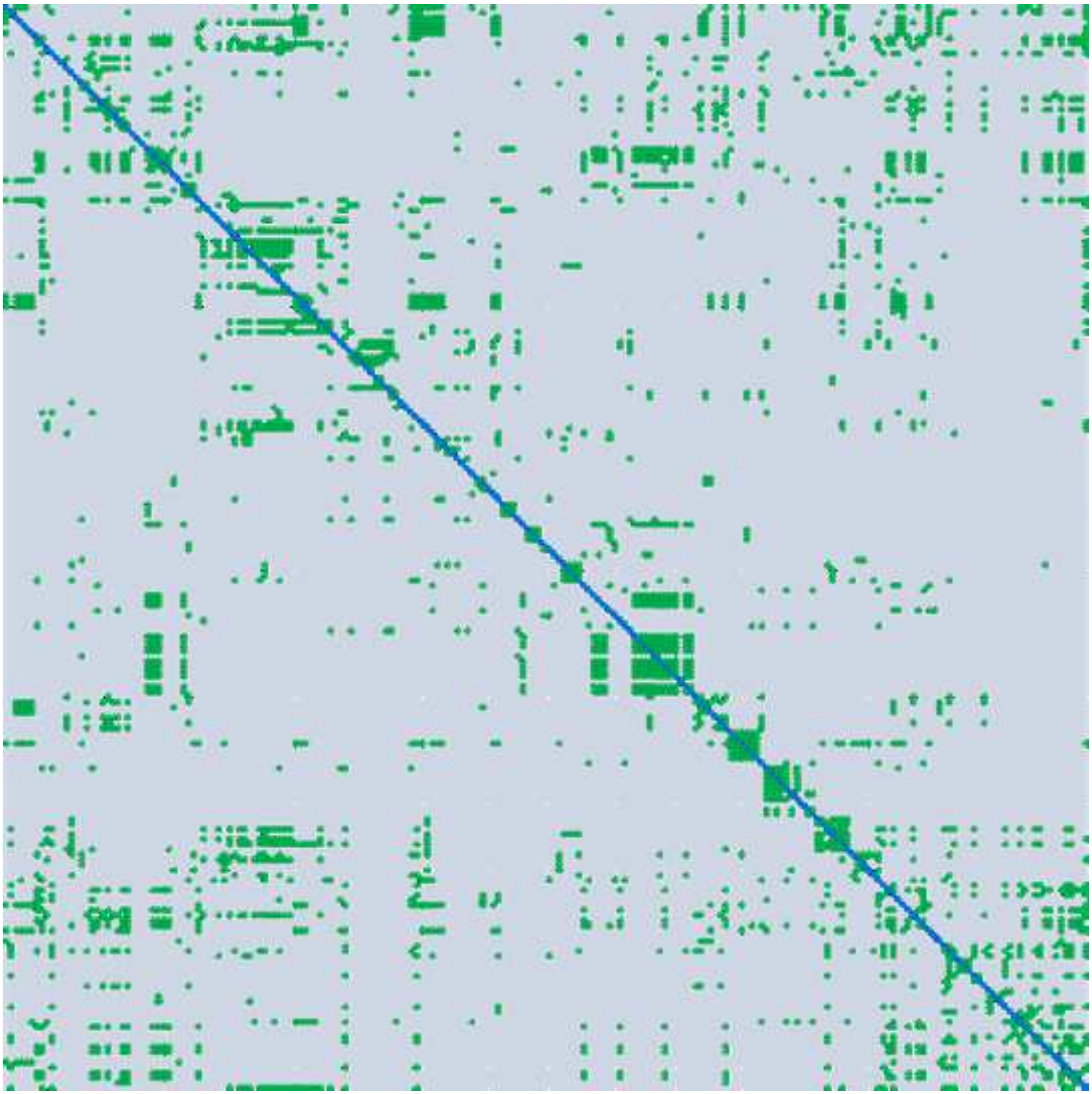


Figure
Click here to download high resolution image

```
(ROOT
  (S
    (NP
      (NP (NN Tooling)
        (CC or)
        (NNS fixtures))
      (VP (VBN switched)
        (PP (IN during)
          (NP (NN changeover)))))
    (VP (MD shall)
      (VP (VB attach)
        (PP (TO to)
          (NP (DT a)
            (ADJP (JJ sub-plate)
              (PP (IN in)
                (NP
                  (NP (NN accordance))
                  (PP (IN with)
                    (NP (`` ``)
                      (S
                        (NP (JJ single) (NN minute) (NN exchange))
                        (VP (NN die)))
                      ('' '')
                      (PRN (-LRB- -LRB-)
                        (NP (NNP SMED))
                        (-RRB- -RRB-)))))))
            (NN design) (NNS philosophies)))))
    (. .)))
```

Figure
Click here to download high resolution image

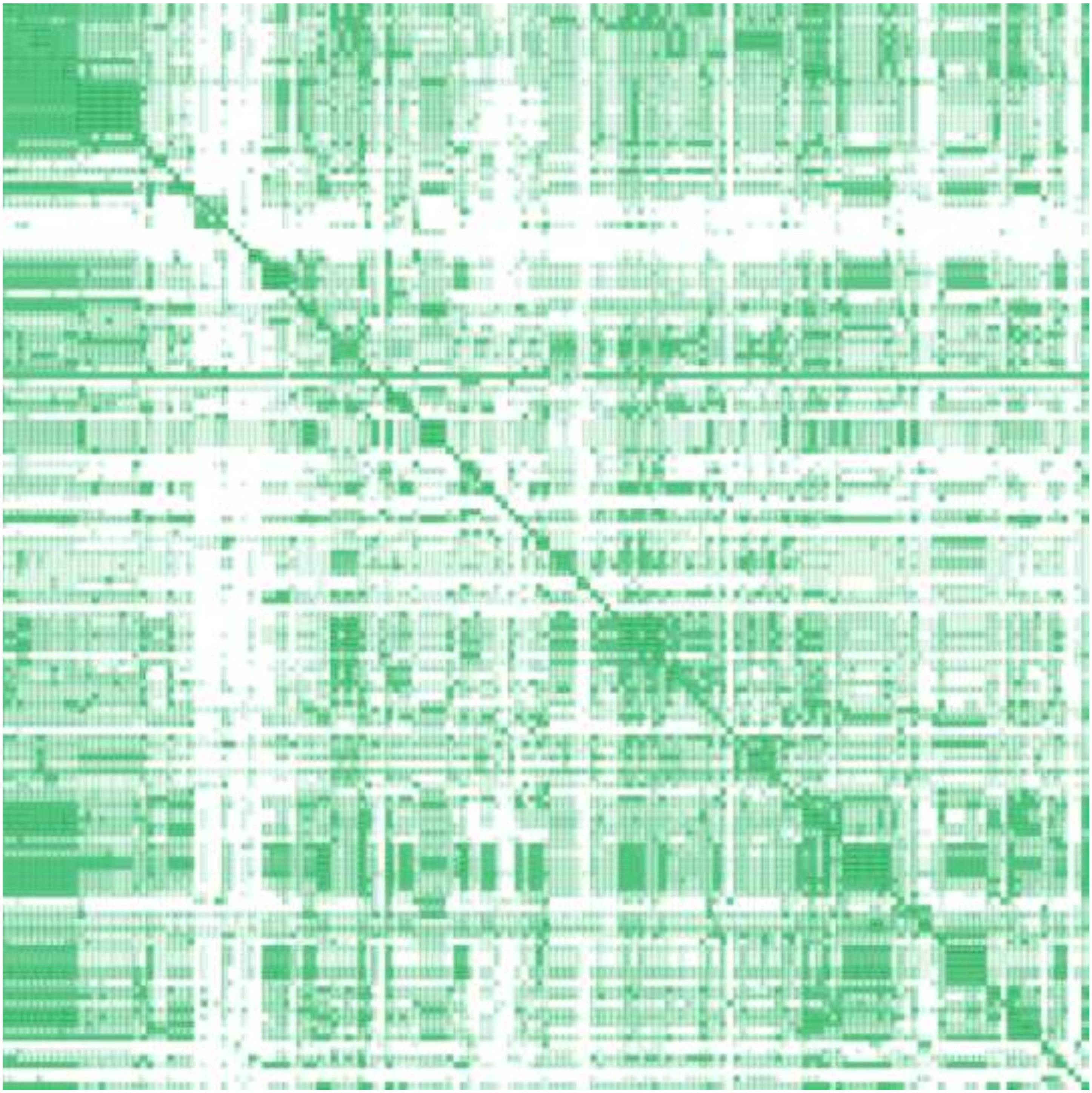


Figure
Click here to download high resolution image

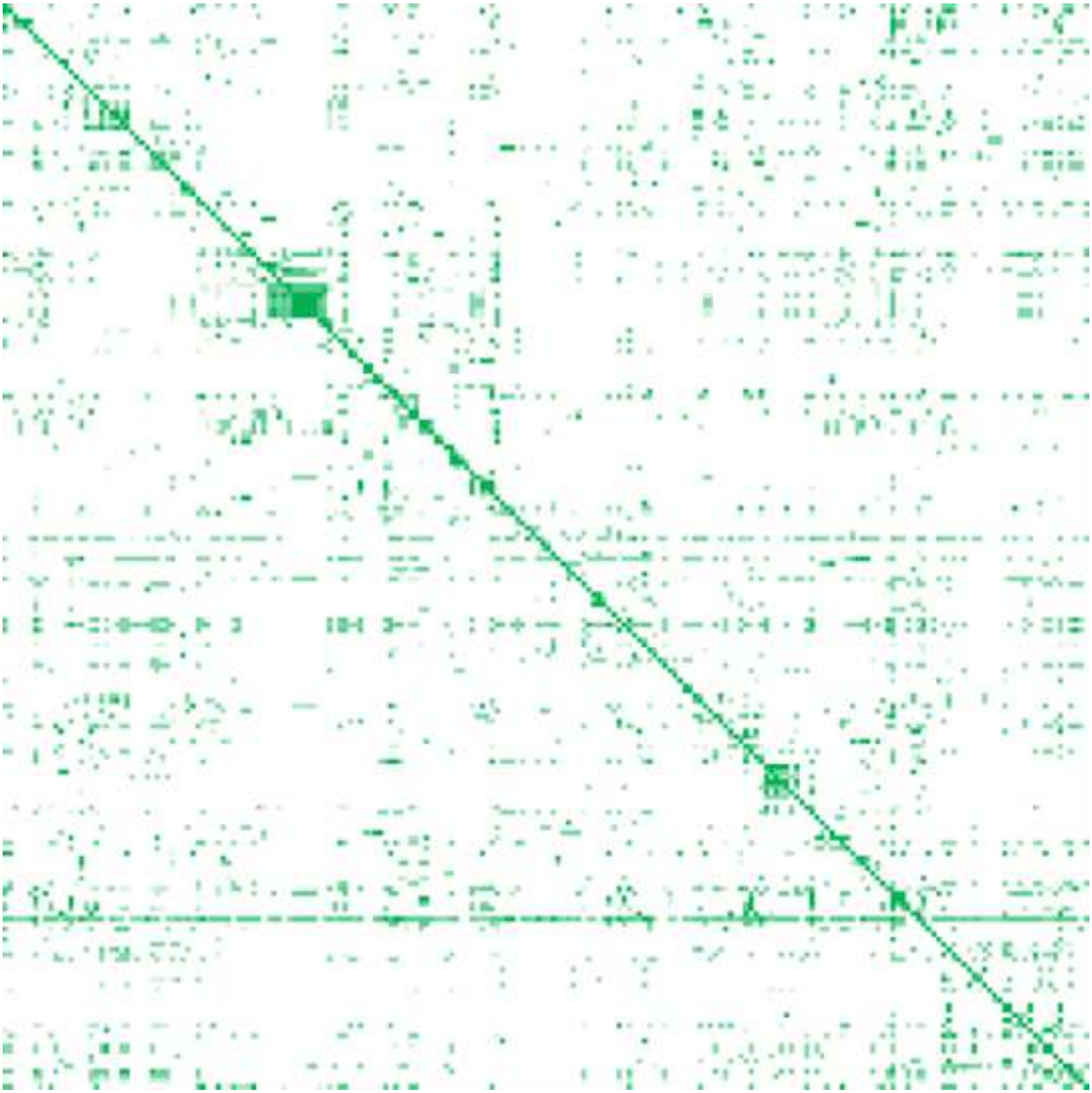
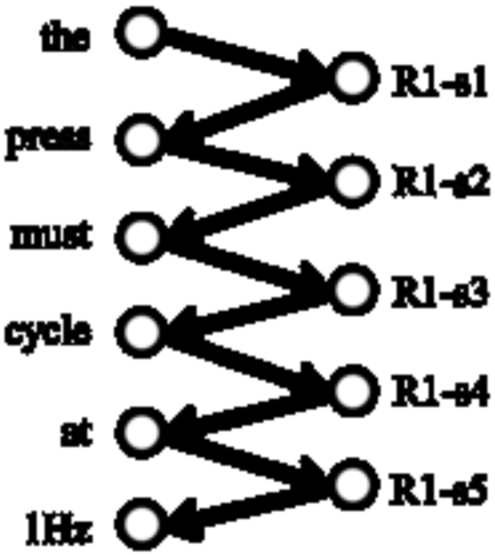


Figure
Click here to download high resolution image



Figure

Click here to download high resolution image

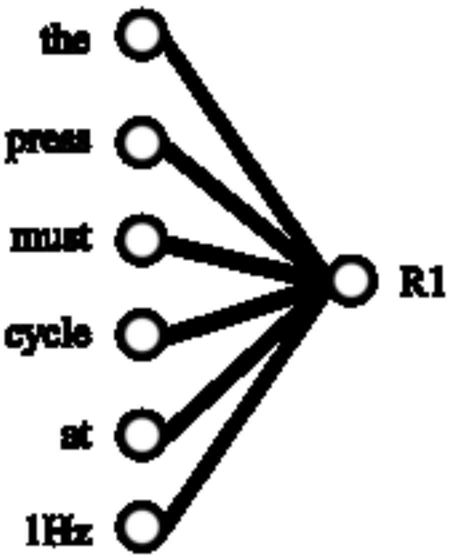
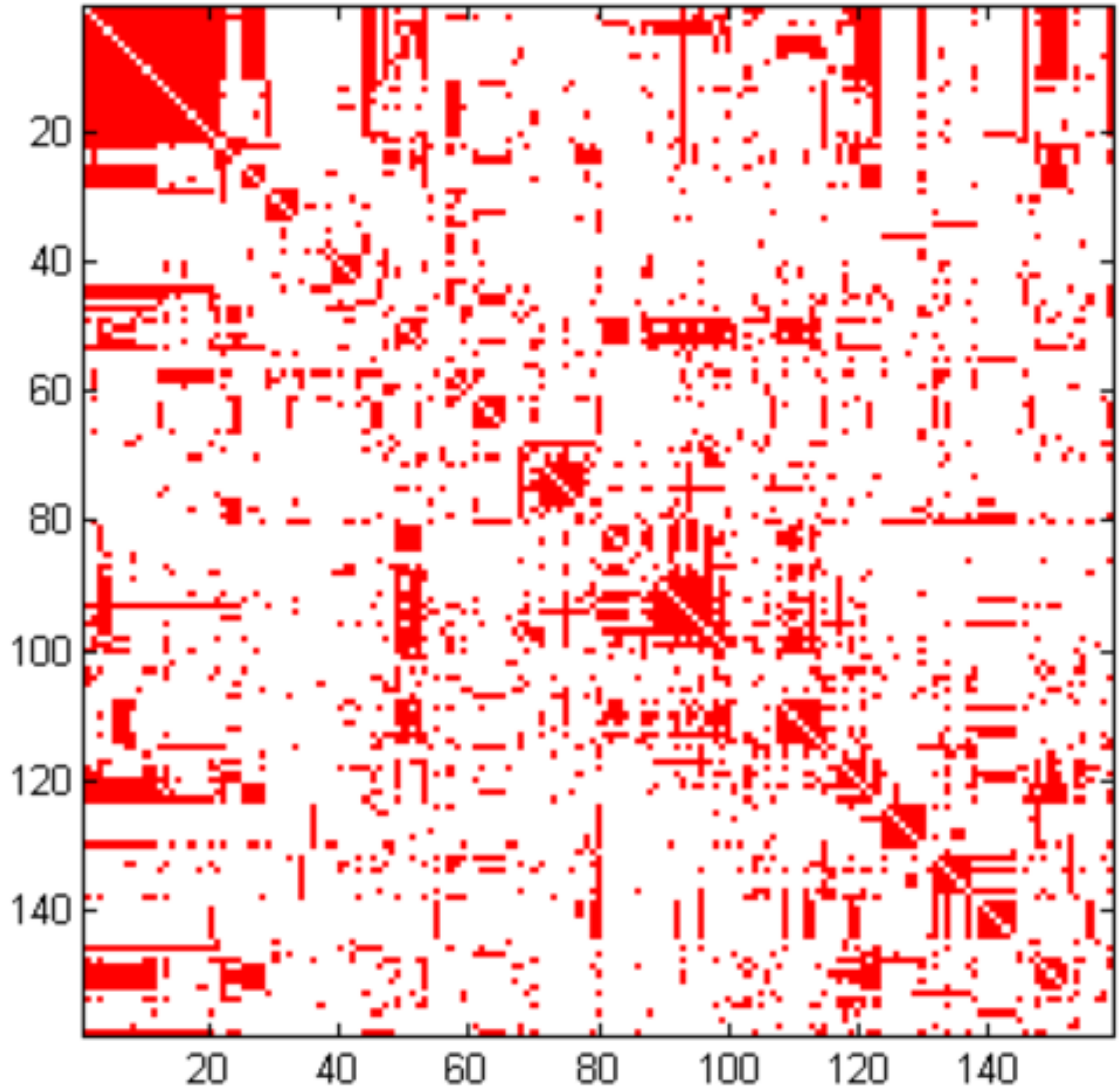


Figure
Click here to download high resolution image



Figure

Click here to download high resolution image

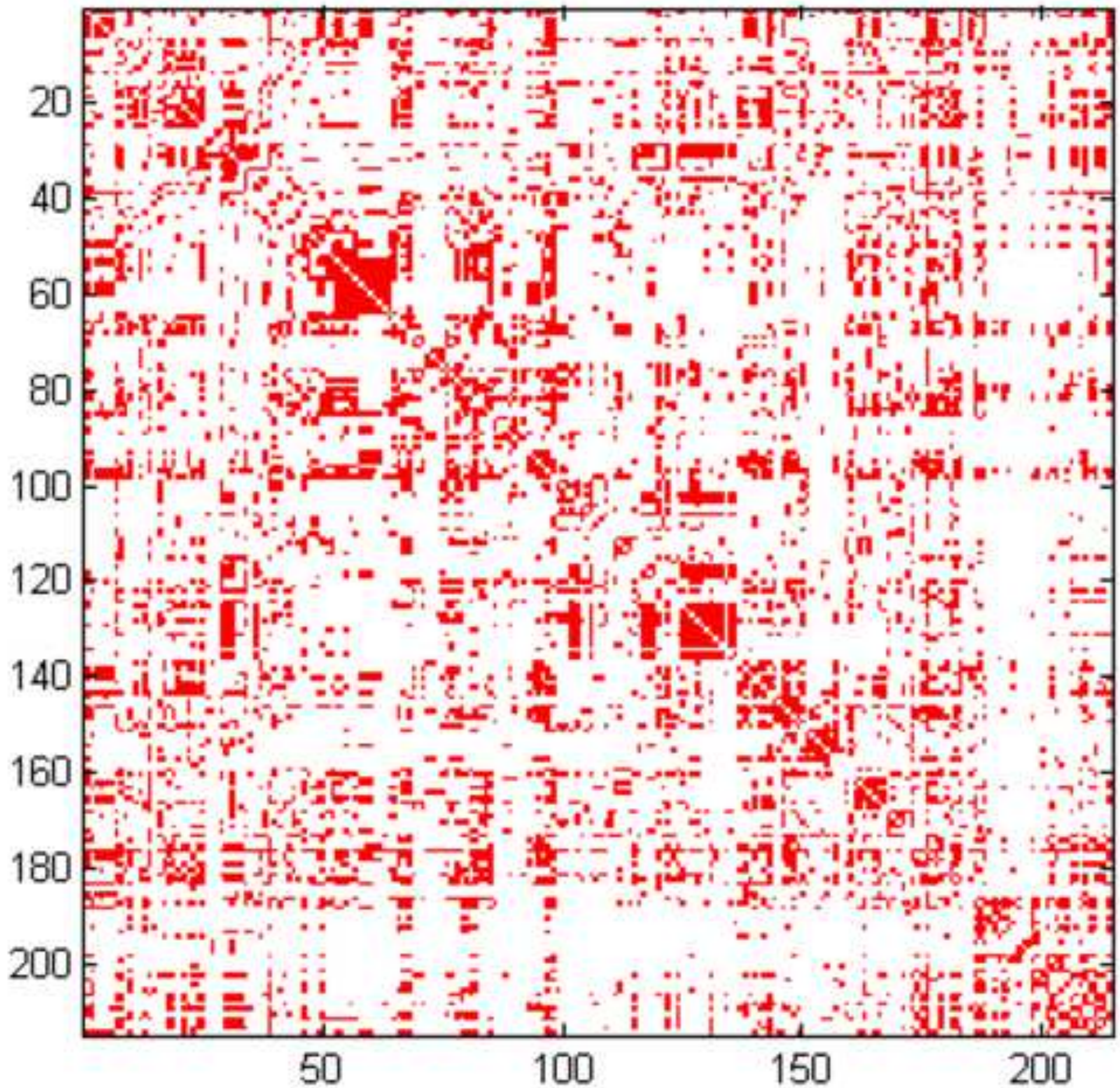


Figure
Click here to download high resolution image