1-1-2004

Reducing Misunderstanding of Software Requirements by Conceptualization of Mental Models using Pathfinder Networks

Udai Kumar Kudikyala

REDUCING MISUNDERSTANDING OF SOFTWARE REQUIREMENTS

BY CONCEPTUALIZATION OF MENTAL MODELS

USING PATHFINDER NETWORKS

By

Udai Kumar Kudikyala

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Science
in the Department of Computer Science and Engineering

Mississippi State, Mississippi

August 2004

REDUCING MISUNDERSTANDING OF SOFTWARE REQUIREMENTS

BY CONCEPTUALIZATION OF MENTAL MODELS

USING PATHFINDER NETWORKS

By

Udai Kumar Kudikyala

Approved:

Rayford B.Vaughn Jr.
Associate Professor of Computer
Science and Engineering
(Director of Dissertation)

Edward B. Allen
Assistant Professor of Computer
Science and Engineering
(Committee Member and Graduate
Coordinator)

Thomas Philip
Professor of Computer Science and
Engineering
(Committee Member)

Donna S. Reese
Professor of Computer Science and
Engineering
(Committee Member)

David A. Dampier
Assistant Professor of Computer
Science and Engineering
(Committee Member)

Robert P. Taylor
Interim Dean of the Bagley College of
Engineering

Name: Udai Kumar Kudikyala

Date of Dissertation: August 7, 2004

Institution: Mississippi State University

Major Field: Computer Science

Major Professor: Dr. Rayford B. Vaughn Jr.

Title of Study: REDUCING MISUNDERSTANDING OF SOFTWARE
REQUIREMENTS BY CONCEPTUALIZATION OF MENTAL
MODELS USING PATHFINDER NETWORKS

Pages in Study: 162

Candidate for Degree of Doctor of Philosophy

Understanding and communicating user requirements in a software requirement analysis effort is very important. Misunderstandings of user requirements between stakeholders will cause problems in terms of satisfying their needs, reduction of defects, cost and schedule during the software development process. This dissertation presents a new technique that has the ability to represent the mental models of the user, developers, project managers and sponsors (collectively referred to as "stakeholders") as network representations. The requirements are modeled as nodes and the perception of stakeholders is modeled as the interrelationships (links) among the requirements.

The requirements are first extracted from a requirements document. The requirements are then categorized into related groups as perceived by each stakeholder. The relatedness (proximity) data collected from the categories is then fed into the Pathfinder generation program that results in the generation of pathfinder networks

(PFNETs). The PFNETs of stakeholders are then compared for similarities/dissimilarities using a graph similarity metric referred to as a correlation coefficient.

During preliminary research work, this technique was applied to multiple student projects with real customers at Mississippi State University (MSU), and to a project at NORTEL, Dallas, Texas with encouraging results. This research was successful in identifying duplicate, ambiguous and misunderstood requirements. The next step was to validate this technique on small-scale and medium-scale projects in an industrial setting. During the summer of 2003, the National Science Foundation (NSF) and AmerInd Inc. jointly sponsored a collaborative industry-university research effort to validate the proposed technique. It was found that this technique is easy to apply and useful to gauge an overall understanding of requirements and identify potentially misunderstood requirements for small and medium scale projects. This technique scaled well from a small-scale project with two stakeholders to a medium-scale project with a little over one hundred requirements and six stakeholders. The correlations helped focus discussions on the requirements that were potentially misunderstood among stakeholders. Duplicate, misunderstood and ambiguous requirements were identified during the facilitation sessions. We also present a new technique that applies information theory-based software metrics to measure consensus about requirements among stakeholders.

DEDICATION

I would like to dedicate this research to all my teachers, my parents, my brother Raj Kumar, and Don Dearholt.

ACKNOWLEDGMENTS

The author wish to acknowledge Dr. Rayford B. Vaughn Jr. for his tremendous encouragement, guidance, support, perseverance and patience during the doctoral program. I wish to thank Dr. Donald W. Dearholt who imbibed the necessary confidence and faith to pursue the doctoral program at Mississippi State University.

The author wishes to acknowledge the support of the National Science Foundation (Grant #CCR-0303554), the James Worth Bagley College of Engineering at Mississippi State University for providing the necessary financial support to conduct the research. The author wishes to acknowledge NORTEL at Dallas Texas, and AmerInd Inc. of Alexandria Virginia for providing accesses to the projects for the research. I wish to specifically thank Ms. Patrica J. (Trisha) Parson and Jay Smith at AmerInd Inc. for providing the necessary support and encouragement during the internship. The author also wishes to thank some employees like Ms. Mary Miller, Ms. Amy Boyle and Jim Cowardin and others for participating and coordinating with their clients to conduct research.

The author expresses his sincere gratitude to Miguel Torres, Yi Huang and Xinquan Lu for their efforts to collect data, build tools and publish some of the results reported in this dissertation. The author wishes to express his sincere thanks to Dr. Donna Reese, Dr. Thomas Philip, Dr. David Dampier and Dr. Edward Allen for their constant support as committee members.

The author wishes to acknowledge Dr. Edward Allen and his students Sampath Gottipati and Rajeev Govindarajan for their guidance and help to conduct research on information theory-based metrics. This research was supported by grants CCR-0098024 and CCR-0085749 from the National Science Foundation. The author wishes to acknowledge the support and help provided by Dr. Donna Reese to build and implement the web-based tool at AmerInd.

I would like to thank my friend Nisreen Ammari for proofreading some of the chapters of this dissertation. I would like to thank Aravind, Neelu, Chandana, Diane, Bruce, Kumaran, Ecap, Jothi, Jyothi, Janna, Praveen and others for their help and encouragement.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER Page

LIST OF TABLES

TABLE Page

TABLE Page

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

1.1 Motivation

The process of building a software system according to needs of users, sponsors, project managers, and developers who are collectively termed as *stakeholders*, is a major goal in software engineering. The software engineering process involves following one of several development life cycle models. The software life cycle model gives developers a very abstract view of how to carry out various phases associated with software development, to include *analysis*, *design*, *coding*, *integration*, *testing* and *maintenance*.

Early in the life cycle, developers must complete a requirements engineering phase in order to understand and document their effort. The software requirements engineering process commonly has four phases associated with it:

- software requirements elicitation,
- software requirements analysis,
- software requirements specification and,
- software requirements verification and validation [61].

These phases of the requirements engineering process fall into the software analysis and specification phase of most software life cycle models. Information regarding stakeholders requirements are elicited and then analyzed to understand what the stakeholders expect from the software system. Once the requirements are understood, they are documented using formal or informal methods that can be understood by all stakeholders. Finally, the requirements are checked to see if they correctly implement the desired system functions (verification), and also, most importantly, that they satisfy the stakeholders' needs (validation). The final product of the requirements engineering process is generally the Software Requirement Specification (SRS) document, which specifies the external behavior of the system, i.e. what the system does, without prescribing how the system will implement that behavior [23].

During the initial phase of any software system development, software developers are challenged to uncover, understand, and specify the stakeholders' requirements [16]. It is important that there be a common understanding between stakeholders and developers with respect to requirements of a software system under development. This is one of the major risk factors in all software projects [35]. Brooks has aptly stated, "The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements… No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later." [8]. The sooner in the software life cycle that misunderstandings are cleared up, the better the ability of developers to build a product that meets stakeholders' requirements, and the lesser the expense associated with the corrections. A defect introduced into the system during analysis will generally be more costly to fix when it is detected later in the life cycle. Research has shown that this defect may cost a factor of ten times more if caught during the coding phase, twenty to fifty times more if discovered during testing, and up to two hundred times more if not discovered until the system maintenance phase [26].

Other research has shown that 45% - 56 % of errors in software can be traced back to errors in the Software Requirement Specification (SRS) [17]. Another survey revealed that 35% of 600 large companies had projects that either could not be completed on time or had critical errors even if delivered [55]. The hundreds of millions of dollars in cost overruns have been largely attributed to lack of user involvement while building the information systems. The need to understand the requirements at an early stage continues to be a major focus area in software engineering [6, 55]. Having a shared vision is important. Mann states, "… one of the keys to improving software is for all parties to reach an agreement in advance on what they're doing – 'a single, explicit, universally accepted focus.'" [45]. There is, therefore, a critical need to develop a consensus among stakeholders by reducing misunderstandings and identifying problematic requirements in the requirement specification stage in order to develop a product that satisfies the needs of the stakeholders and to reduce the overall system development cost.

1.2 Mental Models and Software Systems

Mental models represent what users actually think about a target system and how that system will be used in practice [49]. The mental models of stakeholders could be radically different from what the developer thinks about the same system. Conceptualization of mental models not only models the understanding of the system but also helps to identify any misconceptions that one may have about the system [10]. Mental models of a software system can be represented as networks - state transition diagrams are one example of representing a mental model [54]. Mental models of programmers have been conceptualized to represent software programs as a means of program comprehension [62]. They are also used as a means of understanding and for the purpose of communicating what a person actually thinks about a system [25]. Every stakeholder and system developer may have a different mental representation of the software system based on his or her different views, technical and cultural background [59].

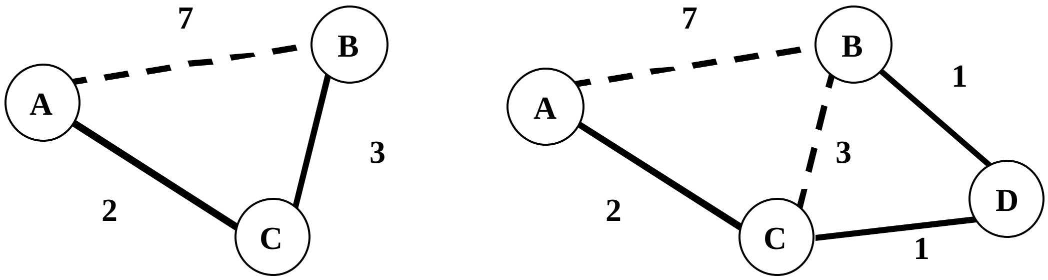
Uncovering the metal models of the various stakeholders involved in a project by eliciting and representing aspects of their *semantic memory*, e.g., how the concepts central to the system are organized, and their interrelationships, would seem to be appropriate. The *conceptual structure* of the software system under development that the stakeholders and the developers may envision, consists of the concepts central to the software system and how these concepts are interrelated. Revealing the conceptual structure of the system as perceived by the stakeholders and developers shows what these groups "think" about the system. The identification of the concepts central to the domain of the software system and the interrelationships among these concepts may be best represented as graphs (networks) where nodes represent concepts and links represent the relationships between the concepts. Once the mental models are generated as networks, and represented as graphs, quantitative and qualitative analysis may be possible in order to compare and evaluate the networks. Comparing the network structures reveals the differences in the structure of the mental models and uncovers misunderstandings about the software system between those involved. “*Requirements creep*” is a term we use to describe the problem of gradual change occurring in the needs and desires of stakeholders as the software system evolves. Every change requested by the stakeholder after the requirements have been initially agreed upon results in additional cost. Reducing misunderstandings by uncovering mental models during the initial phase of the software development may also contribute to a reduction in requirements drift associated with the lack of understanding about system requirements.

1.3 Pathfinder Networks as Mental Models

In order to uncover mental models, a *knowledge engineer* could use a paradigm that facilitates direct interaction with the stakeholders and system developers to identify the concepts that are central to the domain of the software system. This paradigm should also enable the knowledge engineer to measure how the various groups perceive relationships among concepts. It should then provide methods to generate a network structure(s) that is representative of what each group perceives or thinks about the software system. The *Pathfinder paradigm* described in this research provides such methods to uncover the mental models of the subjects involved.

The original objective of *Pathfinder networks* [20] was to generate network models from *psychological proximity data.* Psychological proximity data is the subjective estimate of the closeness or relatedness between concepts as perceived by a human subject. The primary goal is to arrive at network representations with nodes as concepts and links as relations between concepts with weights representing the relatedness between the concepts calculated from subjective estimates (symmetrical and asymmetrical) by human subjects. Scaling algorithms reveal hidden or latent structural information that cannot be seen from the initial raw data. Raw data can be thought of as a fully connected graph where every node in the network is connected to all other nodes. The Pathfinder procedure scales proximity data (relatedness) to reveal the latent structure in the raw proximity data (pair-wise estimates). Proximity (relatedness) distance is chosen because it represents the distance between concepts in human semantic memory. The latent structure only has the critical links between concepts and is revealed by eliminating spurious links from raw data. The spurious links are those links that violates *the triangle inequality.* In Figure 1.1 (a), the edge connecting nodes A and B violate the triangle inequality because there is a shorter alternate path from A to B through node C of length 5. In Figure 1.1 (b) edges connecting nodes A and B, and nodes B and C violates the triangle inequality. The edge connecting nodes B and C violates the triangle inequality because there is a shorter path of length 2 from B to C via node D. The essential idea in a Pathfinder network is that an edge exists between two nodes if and only if that weight is the minimum edge path between those two nodes (we refer to this as the triangle inequality).

The Pathfinder paradigm, by revealing latent structures, facilitates modeling aspects of human semantic memory like memory organization (conceptual structures) and a subject's ability to recall concepts.



(a) (b)

Figure 1.1 Generalized Triangle Inequality - (a) The Dotted Line Connecting Nodes A and B Violates the Triangle Inequality, (b) The Dotted Lines Connecting Nodes A and B, and Nodes B and C Violates the Triangle Inequality

Pathfinder networks are connected graphs where each node represents a concept and the edges represent the relation between the concepts. For the remainder of this dissertation, a Pathfinder network is always assumed to be an undirected graph unless otherwise explicitly stated. The weight associated with each edge represents the strength of the relation between the two concepts. There is a class of Pathfinder networks denoted as PFNET (r, q) (PFNETs) that can be derived based on two parameters, **r** and **q**:

r parameter : This is called the Minskowski distance measure to compute the distances of paths between any two nodes not directly connected in the network. The r parameter can assume values from 1 to 8. If r = 1, then the path length between a pair of nodes in the network is computed as the sum of all the edge weights along that path. If r = 8, then the path length between a pair of nodes is computed as the maximum of all edge weights in that path. Lets assume that the path (P) from two nodes n_1 and n_2 in a connected undirected graph has k number of edges with edge-weights say w_1, w_2 ……, w_k. The weight of the path denoted as W(P) between the two connected nodes n_1 and n_2 is computed as follows:

$$W(P) = \left(\sum_{i=1}^{k} w_i^r \right)^{1/r} \tag{1.1}$$

q parameter: This parameter imposes an upper limit on the number of links in the alternate paths between two nodes that do not viola te the triangle inequality, in order to establish a link between those two nodes. The q parameter can assume values from 1 to n -1, where n is the number of nodes in the network. If q =2, then a direct link between a pair of nodes remains in the resultant PFNET (r, q=2), unless there is an alternate shorter two edge path between that pair of nodes. If q = n-1, then a direct link between a pair of nodes remains in the resultant PFNET (r, q=n-1), unless there is an alternate shorter 1, 2, 3, 4, … n-1 edge paths between those pair of nodes. A class of Pathfinder networks denoted by PFNET (r = 8, q = n –1), where n is the number of nodes in that network has the following properties:

- This PFNET is the union of all minimum-cost spanning trees of the PFNET(r, q). Thus, this PFNET has the minimum number of links, provides all minimum cost paths, and has no violations of the triangle inequality. This PFNET is the least dense and only has the most critical edges i.e., reveals only the critical structural information in its simplest (discernable) form.
- Any monotonic transformations on the elements of the weight matrix still preserve link structures in undirected PFNETs(r = 8, q = n-1). Here we are assuming that the edge weights between nodes are represented as a matrix. Since monotonic

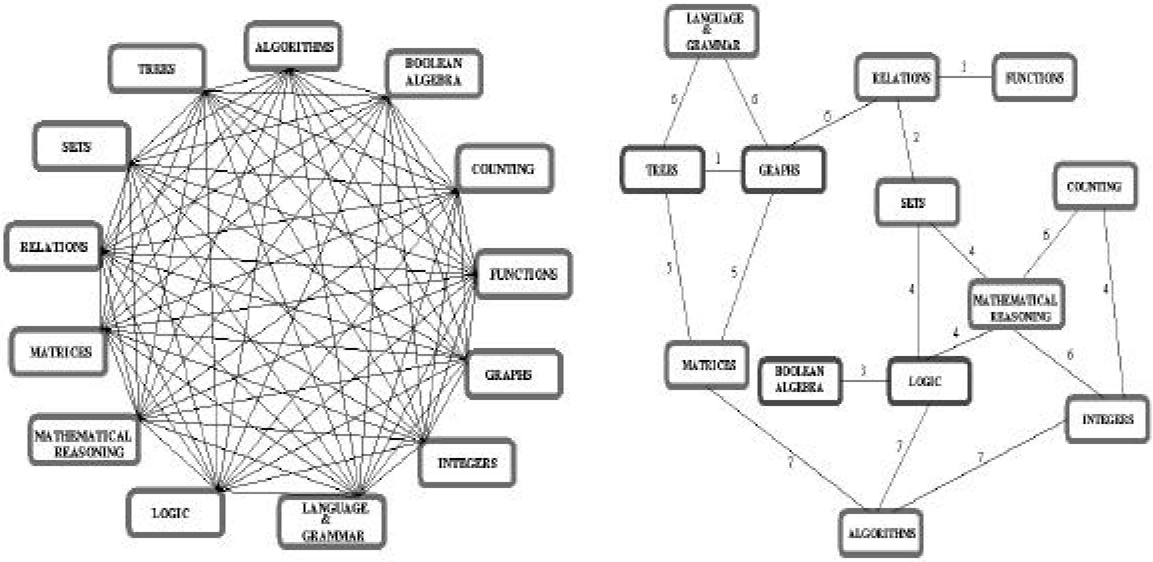
transformations preserve order in data, this class of networks is very suitable for modeling ordinal data where only the proper order of data is important. This particular property also applies to PFNETs(r = 8, q) in general. This means that the conceptual structure in networks is preserved even if there is some uncertainty associated with the meaning in the data values.

- PFNET(r = 8, q = n-1) is available in any PFNET(r, q). This means that the conceptual structure of the systems is preserved in all classes of PFNETs that could be generated for different values of r and q. Hierarchical clustering of concepts reveals important information about how subjects categorize the concepts (requirements) of the domain. Hierarchical Clustering Scheme (HCS) is a technique to reveal clustering (categorization) information. The information available in HCS is also available in PFNET(r = 8, q =n-1) but the converse is not true [14]. Figure 1.3 shows a consensus Pathfinder network for a group of developers with thirty-three nodes denoted by PFNET(r = 8, q = 32). Each node in Figure 1.3 represents a requirement in the Software Requirement Specification (SRS) document. The thick line between a node and a circle means that the node outside of the circle is connected to every node inside the circle with the same edge weights.

The following are useful properties that make Pathfinder networks ideal to represent and evaluate mental models of various groups involved in the system development process.

- The Pathfinder scaling algorithm can generate meaningful network models from psychological proximity data (ordinal) to uncover latent structures. Figure 1.2, illustrates how the Pathfinder algorithm scales raw data (fully connected graph) to a Pathfinder network that reveals the most salient links among the concepts. Figure 1.2 (b) reveals the conceptual structure that helps understand the organization of concepts among experts in a discrete structures domain. It identifies only critical links by eliminating spurious links that violate the triangle inequality [58]. This property may have software engineering applications in terms of revealing what the stakeholders think about a system and how that is different from that of the software developers. This may help predict how the final product would satisfy the stakeholders' needs. The conceptual structure of a software system may also model the expectations of the stakeholders [59].

- The algorithm provides explicit steps to elicit knowledge about the system under consideration and then generate Pathfinder networks that represent aspects of human semantic memory – the conceptual structure. This conceptual structure for different interest groups will reveal the understanding that the group may have and may also reveal any misconceptions between groups. Pathfinder networks can be generated for an individual or for a group of individuals. This may enable a software engineer to evaluate understanding about a system for an entire group or on an individual basis. Comparison of networks could also be made between groups or between individuals (whatever is deemed appropriate).



(a) (b)

Figure 1.2 Discrete Structure PFNETS - (a) Fully Connected Graph with Thirteen Concepts of Discrete Structures (b) The Pathfinder Network PFNET(r = ∞, q = 12) has the Same13 Nodes but Only 18 Links Revealing the Conceptual Structure.

- The algorithm can generate a class of Pathfinder networks based on the values of the **r** and **q** metric. A least dense network, with only the most critical links is generated when the value of r is infinity and the value of q is the total number of nodes (concepts) in the network minus one. Thus, different values of the r and q metrics may reveal more information in addition to the least dense network.

- Pathfinder networks reveal both hierarchical and non-hierarchical clustering among concepts. These clusters may reveal information about how different groups involved in building a system may categorize the requirements. This information, especially for developers, might be useful as a project management tool because, it may be possible to partition the software system by the developers based on the clustering information revealed by Pathfinder networks.

- Quantitative and qualitative analysis is possible on the networks generated. Qualitative analysis may involve visual inspection of the graphs under comparison. This may instantly and intuitively reveal any differences or misconceptions between networks of different groups. Cluster analysis may reveal related requirement categories as visualized by an individual or a group of individuals (consensus).

We believe that this technique has software engineering value in that experiments using the Pathfinder technique were conducted at Mississippi State University in the Department of Computer Science with interesting results. In our initial experiment, participants were students taking a Software Engineering class during the Fall semester of 1999. Four customers agreed to work with the students in order to obtain a needed software product that was intended to be placed into production. Each of the four user/developer teams were required to complete an entire development activity beginning with requirement elicitation and ending with product delivery. The resultant Pathfinder networks of the developers and users were then compared and analyzed to check for similarities and dissimilarities. This was achieved by computing an overall correlation coefficient (cc) based on path distances of the two networks and by also computing the correlation coefficient of individual requirements (node) in both the networks for a particular project.

The resultant Pathfinder networks of developers and users were then compared and analyzed to check for similarities and dissimilarities. This was achieved by computing the overall *correlation coefficient* (cc) based on path distances of the two networks and also by computing the correlation coefficient of individual requirements (nodes) in both networks for a particular project [29]. This topic will be dealt with in more detail in Chapter III.

The higher the value of the correlation coefficient, the more similar the mental models were. This may indicate a better understanding about the requirements between users and developers. A group of nodes (requirements) are considered a *clique* if all the nodes within a cluster form a fully connected graph with the same edge weight. A fully connected graph is a graph where every node in the graph is connected to every other node in the same graph. The bottom left group of nodes inside the ellipse in Figure 1.3 is a clique. The bottom right ellipse is not a clique because there were links within that ellipse that had different edge weights than the others (not shown). The clustering information from the cliques resulting from the Pathfinder networks, reveal patterns showing how the user and developer categorized the requirements. Generally, the Pathfinder network clustering information shows similar requirements that achieve a particular system function. Figure 1.4 is a network structure for the same software system as Figure 1.3, but represents the resulting mental model of the customer.

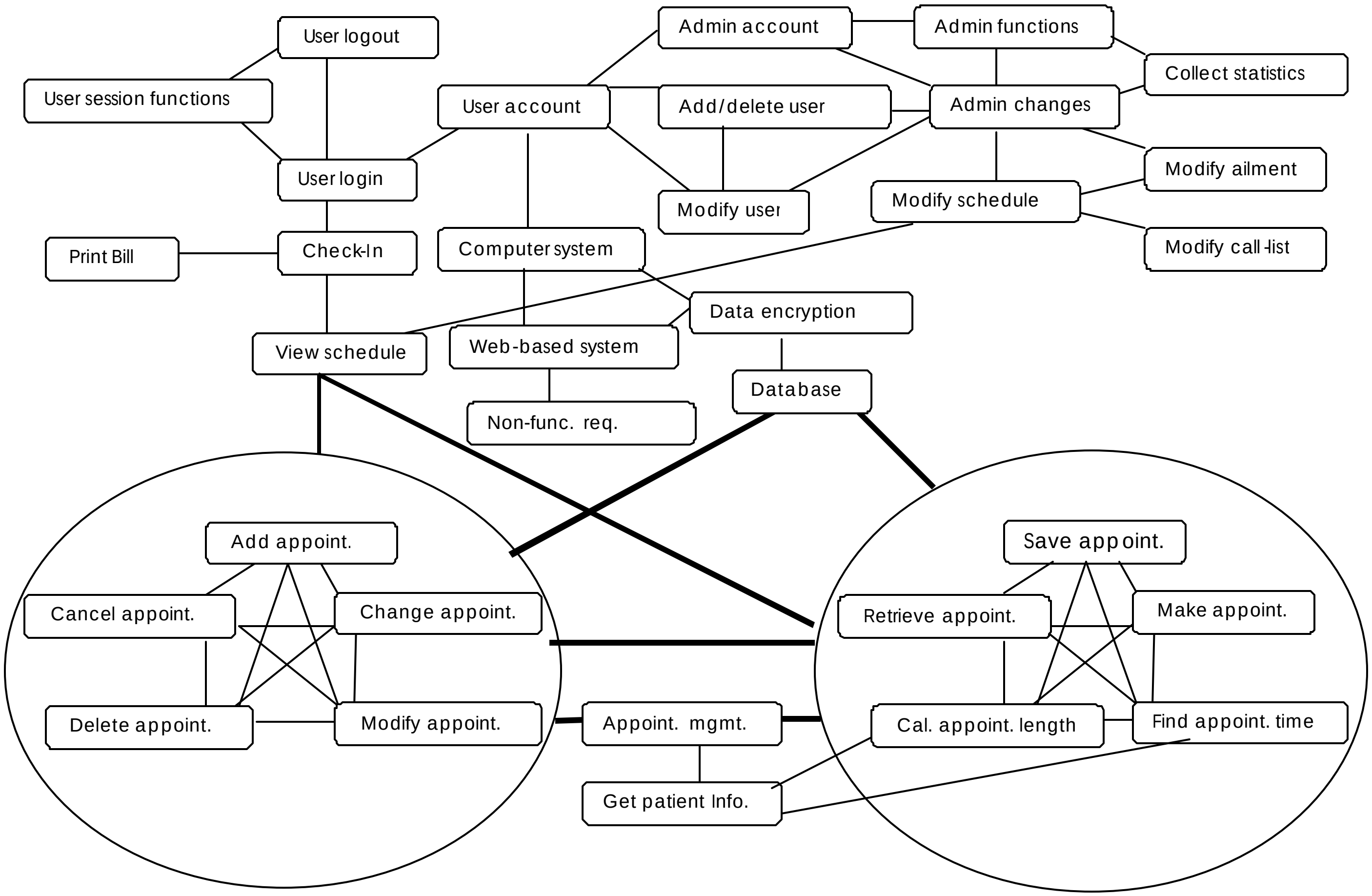


Figure 1.3 Pathfinder for Developer - Consensus Pathfinder Network with 33 Concepts for a Group of Developers – PFNET $_{\mathrm{DEV}}$ (r = ∞, q = 32), Without Edge Weights [44]

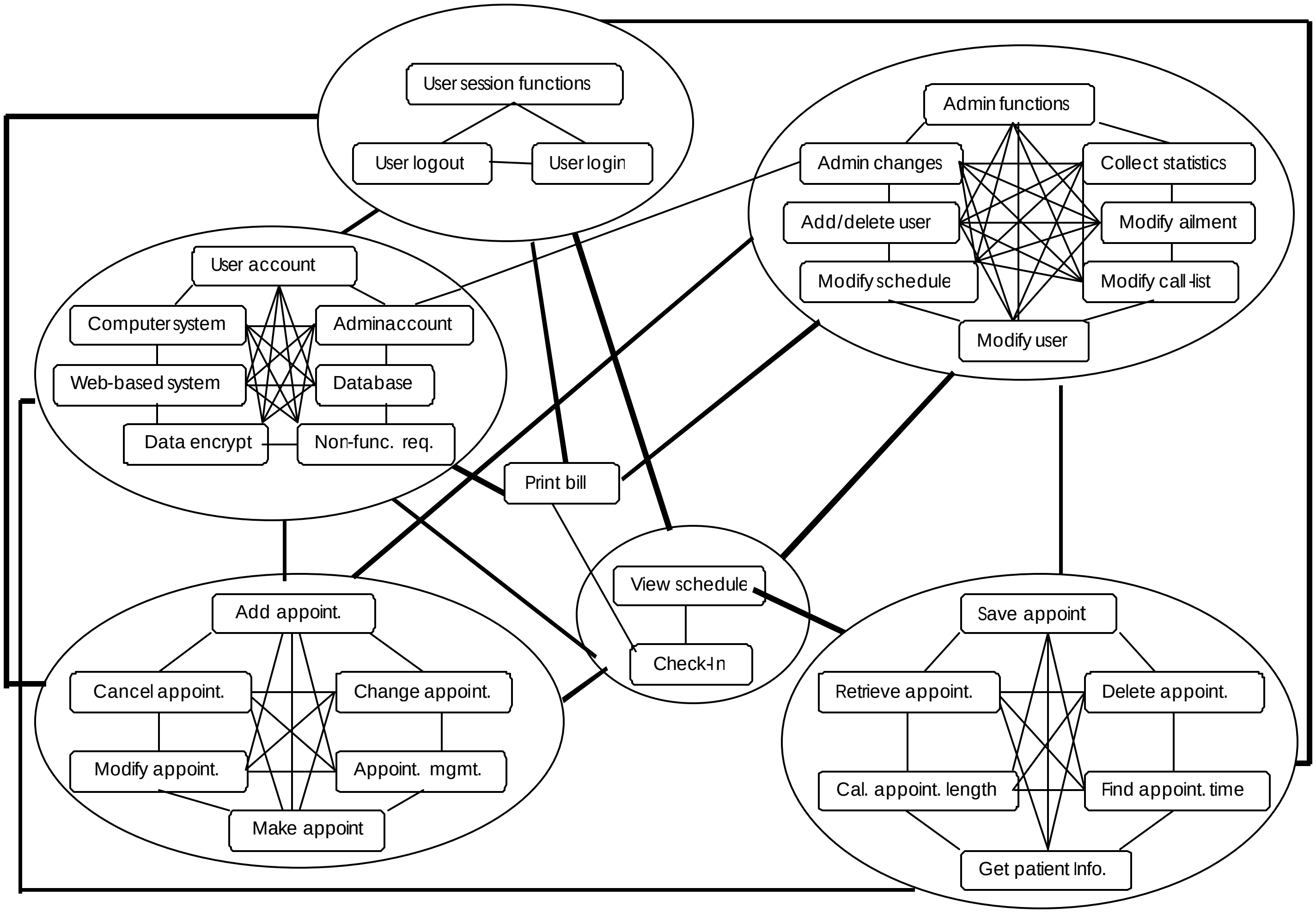


Figure 1.4 Pathfinder for Customer - Pathfinder Network with 33 Concepts for a Group of Customer– PFNET $_{CUS}$ (r = ∞, q = 32), Without Edge Weights for the Same Software System as Figure 1.3 [44]

Looking at the two networks in Figure 1.3 and Figure 1.4, it can be observed that they are very different in terms of number of links as well as the clustering of concepts. Some links connecting the various clusters in the $\mathrm{PFNET}_{\mathrm{CUS}}$ are missing from the $\mathrm{PFNET}_{\mathrm{DEV}}$. Thus, by visual inspection alone, one can conclude that the mental models are very different. This conclusion was later supported by the fact that the customer was dissatisfied with the delivered product.

Similar requirements tend to be directly linked since they are thought to be very closely related. This makes it easier to identify duplicate requirements since only neighborhood requirements (concepts) have to be compared in the Pathfinder network in order to check for duplicates instead of comparing every requirement with high correlation coefficients [44].

The procedures to collect similarity ratings for concepts are described in Chapter II. The generation of Pathfinder networks and the computation of the correlation coefficients were again done as separate steps of the procedure. We were able to demonstrate an ability to predict common understanding between customers and developers, determine where potential redundancy existed, and to communicate more effectively about the project using this technique. However, we did so using only small projects. Further research is warranted to study the feasibility of generating Pathfinder networks for medium-scale and large-scale software projects. The issue of scalability is an important one to address. The procedure used in our early experiments was time consuming and tedious for developers/users who had to go through the process of network generation for software projects with only few requirements. To this end, an automated web based tool was developed as part of a Master's project at Mississippi State University during the Fall 2001 semester. This tool demonstrates the feasibility of building an automated tool that uses an internet browser in order to automate the process of collecting similarity judgment ratings from the people involved in building a software system, automates the generation of Pathfinder networks (adjacency matrix) and automate the process of comparing the resultant networks. Comparison involves generating an overall correlation coefficient of path distances between concepts of the two networks. Results generated showed that quantitative analysis of the Pathfinder networks revealed duplicate, ambiguous and misunderstood requirements [68]. The issue of scalability, our experience to date, and the development of a useful tool set are further described in Chapter III.

1.4 Hypothesis

The hypothesis of this research is that by conceptualizing the mental models of stakeholders of a software system being developed using Pathfinder networks, it is possible to predict common understandings and misunderstandings among the stakeholders. This can be achieved by measuring the similarities/dissimilarities between the Pathfinder networks of the stakeholders using correlation coefficients between individual requirements. The Pathfinder networks thus generated using relatedness data will also provide insights needed to identify ambiguous, duplicate and misunderstood requirements very early in the requirements engineering phase of the software development life cycle. Resolution of the misunderstandings about requirements could be achieved by focusing discussions among stakeholders on requirements with low correlation coefficients.

We conjecture that the feedback in terms of misunderstanding and identifying ambiguous, duplicate and misunderstood requirements will effect the end product in terms of decreasing the number of defects in design and code, increase customer/user satisfaction, lower costs, and shorten development time. Research by Dr. Boehm at the University of Southern California [6] has shown that identification of errors early in the software engineering life cycle process can result in large savings versus finding the same errors during the maintenance phase. In some cases this can be a 100X savings.

1.5 Research Issues

The following research issues will be addressed in this work:

- **Research issue 1**: Identification of central requirements for a system being built and prioritization of those requirements. The focus here is on identifying the nodes of the Pathfinder network to be generated and also on identifying the relative importance of the requirements in terms of implementation for both the user and developer groups. The requirements will be displayed as a Pathfinder network based on ratings of similarity given by both the user and developer.
- **Research issue 2** Testing and validating different metrics used to compare the similarities/dissimilarities between the Pathfinder networks of users/customers and developers that measure the understanding of requirements during the requirement analysis phase. The same metrics can likely be used for measuring common understanding at an individual requirement level and we intend to validate this. We

will also analyze the information revealed by path distance and set-theoretic measures of similarity between Pathfinder networks. As a part of addressing this issue, studies to find an effective way to classify the correlation coefficients to identify low, moderate and high similarities in the networks will be undertaken.

- **Research issue 3**: The effectiveness of this technique to identify duplicate, misunderstood and ambiguous requirements will be investigated.
- **Research issue 4**: Feasibility of generating Pathfinder networks for medium-scale software projects and to validate the technique in an industrial setting. This research will form the basis for more experimentations and conclusions relative to large-scale projects.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction to Pathfinder Networks

Pathfinder networks were the result of trying to develop network models from proximity data during the year 1981. Other structural models that existed at that time were multidimensional scaling and cluster analysis. None of these other models were represented as networks even though they used proximity data as a starting point for structure analysis. Network models to represent human semantic memory have played an important role in cognitive psychology and artificial intelligence. Since their inception, Pathfinder networks, have been widely used to represent knowledge structures in categories, scripts [24], room schemata [56], and problem solving schemata [18]. They have been used to model the knowledge of experts and novices in the domain of computer programming [14] - using clustering information to derive object definitions in software development [9] and the assessment of the knowledge of students when compared to that of experts [1, 30, 32]. Pathfinder networks have been used in the field of Human Computer Interaction (HCI). Some of the applications of Pathfinder networks have been in designing interfaces and providing navigation for on-line help for UNIX operating system commands [46]; the design of a document retrieval interface [27, 67]; content-based image visualization [12]; image database visualization [22]; and, information visualization on the world wide web [11]. Pathfinder objectives, to model aspects of human semantic memory, were compared to other scaling techniques existing at that time like Multidimensional Scaling (MDS), Hierarchical Clustering Scheme (HCS), and Alternate Least Square Scaling (ALSCAL). The comparisons revealed that the Pathfinder paradigm did fulfill its original objectives.

Pathfinder scaling procedures could model both asymmetrical (directed PFNETs) and symmetrical data (undirected PFNETs). They were also complementary to MDS in that the models local structure (pair-wise) was enforced by the triangle inequality more than global structure as in MDS [57]. It proved better than HCS in terms of not only revealing hierarchical clusters but also retaining the information used to form clusters and revealing structural properties that cannot be modeled in HCS [20].

The two psychological experiments that exemplified the objectives of the Pathfinder paradigm in modeling aspects of human semantic memory are:

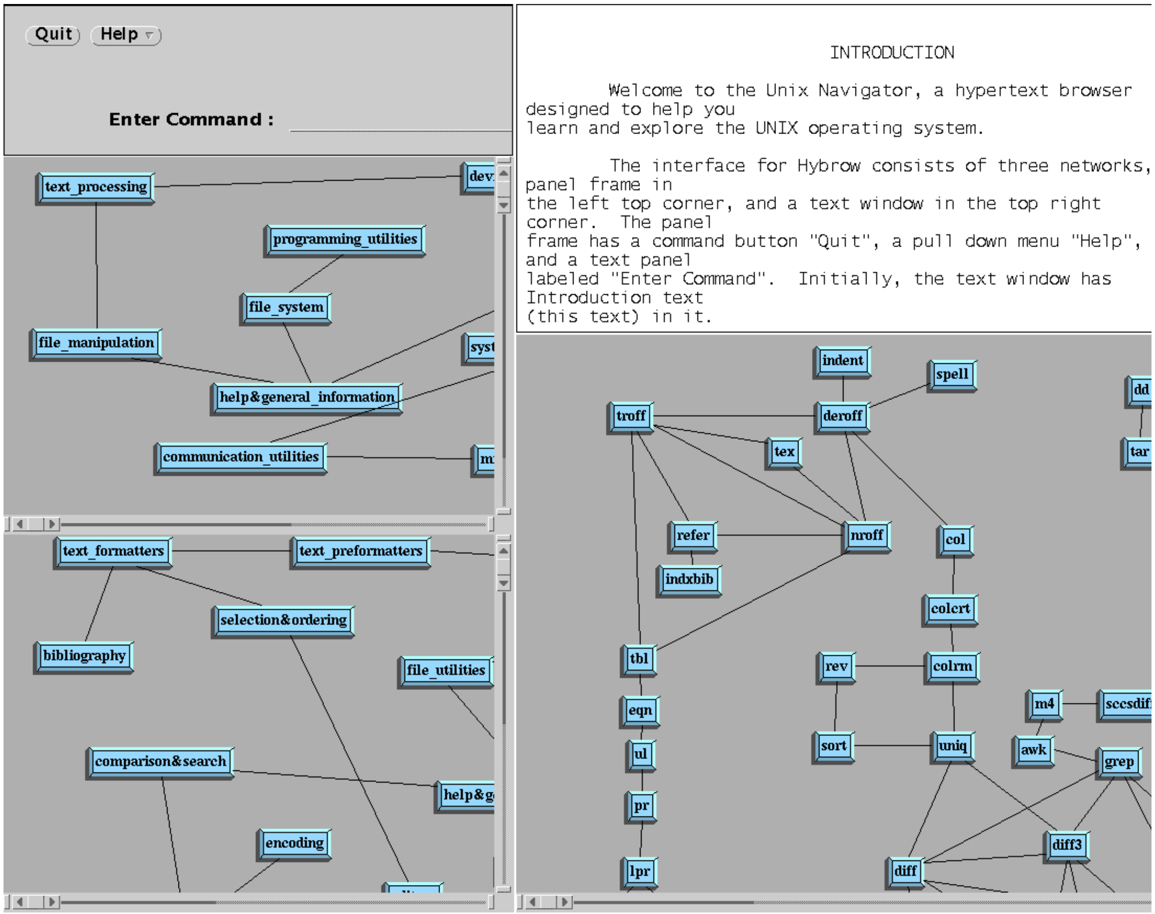
- Classification of expert and novice fighter pilots [58], and
- Study of recall performance with respect to memory organization [14].

In the first experiment, successful classification of expert and novice fighter pilots was made based on analyses of the Pathfinder paradigm’s conceptual structure (networks) elicited from experts. Thus, it was shown that Pathfinder networks could represent the organization of information in the memory of experts. It also demonstrated that quantitative analyses performed on Pathfinder networks revealed useful information that could be used to classify novice and expert pilots.

The second experiment showed that subjects could recall a list of concepts (serial recall) organized by Pathfinder better than a list organized by MDS. Additionally, Pathfinder networks were shown to better predict a subject's organization of free recall than MDS. Thus, with this experiment, it was shown that the Pathfinder paradigm could not only scale data to reveal network representations but also reveal conceptual structures that were predictive of human recall performance.

Other important areas for the application of Pathfinder networks are as follows:

- User interface design to communicate system structure and organization: This is based on the hypothesis that effective human-computer interaction is achieved by designing user interfaces based on system structures and organization derived from cognitive models [47]. The closer the mental model of the user to that of the system, the better the ability of the user to learn and understand how to use the system. The goal is to communicate the system structure as imagined by experts to the users. This is achieved through mapping the conceptual structure and organization of the system represented by Pathfinder networks to the user interface of that system. In a sense, the user interfaces are designed based on the mental models of the experts captured by the Pathfinder networks. The Pathfinder paradigm is used to elicit and represent mental representations of the designers/experts using proximity (relatedness) data rather than subjective impression of designers. The ability of the paradigm to preserve minimum cost paths (strongly connected links) between concepts were used as paths for navigation between the concepts of the system. The Hypertext Browser (HyBrow) for the UNIX online documentation system is one example [47] as shown in Figure 2.1. The concepts were the *man* entries of the UNIX system commands. The links between the concepts from the Pathfinder networks were directly mapped into the user interface to provide navigation for the user between different *man* commands.



```
                         INTRODUCTION

        Welcome to the Unix Navigator, a hypertext browser
designed to help you
learn and explore the UNIX operating system.

        The interface for Hybrow consists of three networks,
panel frame in
the left top corner, and a text window in the top right
corner.  The panel
frame has a command button "Quit", a pull down menu "Help",
and a text panel
labeled "Enter Command".  Initially, the text window has
Introduction text
(this text) in it.
```

Figure 2.1 Snapshot of the User Interface to HyBrow System

- Information retrieval: Pathfinder networks have been applied in document retrieval from a database using document indexing [27]. This is done by matching the queries of the user (conceptual representations) with indexes that represent the conceptual domain of the document. Both queries and the indexes of documents could be represented as Pathfinder networks. Also for a naïve user, the presentation of indexing terms for a conceptual domain will provide more information regarding the association among the indexes as perceived by the experts and also guides in exploration in that domain. This is done by the Pathfinder network's ability to communicate the system structure through the user interface as explained previously.

Also, the ability of Pathfinder networks to provide alternate paths is very useful in providing multiple path access to documents. The ability of Pathfinder networks to cluster similar concepts and model the most salient links among the concepts as graphs makes them suitable for visualization as opposed to linear textual representations. The Pathtrieve System [27] is one such example. This is a system where the query typed by a user in natural language is displayed as a Pathfinder network. This network could be modified with information obtained from the Pathfinder network representing the domain of the database and networks representing the thesaurus terms. The system then matches the query representation with that of the database and retrieves the documents and references. Another example where the Pathfinder networks are proposed to be used as document retrieval interfaces is the author co-citation analysis (ACA) technique [67]. This technique provides visualization using data from Arts and Humanities Citation Index for 1998-1997 and as potential user interface to digital libraries. Here the name of the author is the input to the system. Similarity measures are generated between the input author and the authors most often co-cited with that author. This is expected to capture the intellectual structure in a particular domain. The similarity matrix thus generated is used to generate a PFNET to visualize the information and potentially use it as an interface to digital libraries to retrieve the documents of the author and the co-cited authors. The graphical displays also have advantages in conveying more information than just plain text.

- Image database retrieval and visualization: Here the retrieval is concerned with matching two images according to some visual features that are extracted from the images. The visualization part is concerned with generating graphs as a means to display information. One example is CBIR (Content Based Image Retrieval). The first step typically involves feature extraction where images are matched using visual features like color, texture, shape and spatial constraints. Subsequently image similarity data is generated based on these visual features. The Pathfinder networks are used to evaluate the similarity data, to reveal the most salient features in the similarity data and display them as graphs in a two-dimensional space. This is accomplished by using the ability of the Pathfinder paradigm to model relatedness proximity data as networks to cluster similar concepts. These clusters potentially have applications in data mining and image retrieval from film and video archives.

The other feature that is used is the ability of PFNETs to generate only the strongest links PFNET with r = 8 and q = n – 1 (number of nodes less one). This feature generates the least dense network having a minimum number of links but revealing the most salient links. The Pathfinder graphs thus generated are proposed to be used as a user interface for image retrieval and for browsing images in digital libraries [12].

- Associative networks for database organization and search: Pathfinder networks have been used in the design of associative network databases for computer vision. The goal is to design a database and provide a mechanism to match a visual image with that in a database. Each entity is represented as a feature vector derived from features like color and shape of the entities. This has application in robotic vision systems [19]. Pathfinder networks, and subsequently monotonic search networks (MSNET) based on feature vectors of objects, are used to design a database to support search without backtracking and identification and classification of unknown objects. An MSNET derived from a Pathfinder network has the unique property of supporting search without backtracking. The properties of a Pathfinder network that support these kind of applications are their ability to preserve the link structure, even under monotonic transformation of similarity data, thus making them able to withstand some noise in the data, the clustering of entities using similarity features that support different levels of abstraction, and the ability to generate different classes of Pathfinder networks based on different values of *r* and *q*.
- Shortest-path routing in dynamic networks: This is another application where the Pathfinder paradigm was proposed for effective search and to enforce fault-tolerance in case of failure in network links and nodes. The feature of the Pathfinder network that supports this capability is the preservation of the geodetic distances[1]. Also when

[1] The minimum length path between any pair of nodes in a network

links fail in a dynamic network, the ability of the Pathfinder network to store the intermediate information while eliminating links (the triangle inequality) may be used to search alternate routes. Shortest-path algorithms using K-Local Image Graphs [43] based on Pathfinder networks have been proposed for the Web Operating System (WOS) to locate resources for applications submitted over the Internet [22].

- Modeling dynamic phenomenon using Pathfinder networks [21]: This has been shown to be useful in modeling aspects of human learning. The features of Pathfinder networks that support this approach are consistency with respect to enforcing generalized the triangle inequality, representation of sparser networks that preserve the minimum cost paths, the ability to represent hierarchical clustering, and the ability to withstand noise in data.

2.2 Information System Modeling – Ontology Versus Cognition

The applications of Pathfinder networks discussed in the previous section have, to a certain extent, motivated the current research work. The Pathfinder technique discussed in this proposal is a way of modeling requirements of a software system; information system modeling is more general. Our literature review conducted with respect to information system modeling revealed some interesting theories. More specifically, two tracks of research are being pursued in the area of information system modeling. These theories are based on the premise that information system modeling should have some strong theoretical foundation. The literature review revealed that *ontology* and *cognitive science* are being pursued as the theoretical foundations to model information systems.

In that sense, modeling requirements using Pathfinder networks is based on a similar belief that the Pathfinder paradigm has strong foundations in cognitive science and artificial intelligence.

Information modeling usually refers to all activities that are performed to model the domain for which the information system is being built [50]. Information modeling could involve building conceptual models, data models, semantic models, and executable models [50, 51]. Information modeling in this research primarily addresses conceptual modeling and semantic modeling since Pathfinder networks are not only conceptual models, but can also represent certain aspects of human semantic memory. Modeling of information systems needs to be based on a strong theoretical foundation for the research to be "…enduring, substantive, coherent and transcendental"[66]. Thus, modeling of information systems could be based on a strong theoretical foundation like ontology or cognition. The question of which theoretical foundation to use for information modeling depends on the nature of the assumptions made about what it means to model an information system.

Ontology is used as a theoretical foundation for conceptual modeling of information systems under the assumption that information systems directly represent models of the real world [66]. Ontology refers to the nature of things in the real world. In a sense an ontology provides the theory and structures to represent a real-world phenomenon. On the other hand, if information systems are assumed to be representations of what humans perceive, or based on the knowledge of humans and the organization of the knowledge about the domain, then cognitive science forms the theoretical foundation for modeling information systems. This has led to using *classification theory* as the basis for information modeling [50]. Classification theory basically deals with how humans classify perceptions. In a sense, information systems based on cognitive foundations assume that conceptual models are based on representations that are filtered and constructed through human perceptions [52, 53]. We believe that conceptual models using Pathfinder networks are based on a cognitive foundation. This is because Pathfinder networks are models of how a subject perceives a system being constructed and Pathfinder networks also model the structure and organization of knowledge in the domain.

Although Pathfinder networks have shown great utility in a wide variety of applications over the years, we can find no evidence of their use in the software analysis and specification phase of software engineering – particularly in the context of requirement engineering. Pathfinder networks have been used to model software programs to understand how novice, intermediate and expert programmers organize knowledge about programming structures and also, clusters generated by Pathfinder networks have been used to derive object definitions in software development. Early experimentation (presented in the next chapter) provides evidence that Pathfinder network, properly employed and used can substantially contribute to this area.

2.3 The PFNET Technique

In order to use the PFNETs as cognitive models of requirements understanding, co-occurrence data is first collected directly for the stakeholders of the software system. Then, PFNETS that could potentially represent the perception about the requirements will be generated for all the stakeholders. Correlation coefficients for each requirement between the two groups of stakeholders like the customers and developers will be generated. The correlation coefficients would then predict the potential misunderstanding of each requirement between the two groups. Overall correlation coefficient to predict the potential overall misunderstandings between the two groups can also be computed.

Thus, the modeling of the perception of stakeholders about requirements using PFNETs, and then predicting the potentially misunderstood requirements using correlation coefficients to improve the overall understanding about the requirements between those stakeholders will be referred to as the PFNET technique. In a related research effort, we applied *information theory-based* metrics to measure consensus about requirements among stakeholders.

2.4 Information Theory-Based Metrics

Information theory-based metrics have been proposed for ordinary graphs at the system level and at the module level [2, 5]. These metrics have been extended to hypergraphs by Allen and Gottipati [3]. Since the categorization tables can be viewed as hypergraphs, we applied these complexity measures to the two projects at AmerInd. The "complexity" measure applied to the hypergraphs conforms to the properties of the family of software metrics proposed by Briand, Morasca, and Basili [2, 7].

Information theory-based metrics are measures of attributes such as size, length, complexity, coupling, and cohesion resulting from using graphs that represent software systems and modules that are represented as sub-graphs. They are an alternative to counting-based metrics, which measure the attribute size as number of nodes, length as number of nodes in a path, complexity as number of edges, coupling as inter-module edges, and cohesion as number of intra-module edges divided by the maximum possible. In contrast to counting-based metrics, information theory-based metrics consider the design decisions of software embodied by a graph abstraction as information [2].

Formal Notation to Define Complexity Metric for Measuring Consensus

The following defines some of the concepts and notation that underlie our definitions of complexity, which was applied in our experiments. A system, **S**, is an abstraction of categorization of requirements represented by a graph with n nodes and n_e hyperedges connecting some of the nodes [2]. The system graph, S, of a system **S**, with n nodes, is all nodes in **S** and all its hyperedges, plus a disconnected node modeling the lack of relationship to the system's environment. Without loss of generality, we index the environment node as $i = 0$, and the nodes in **S** as $i = 1, \ldots, n$. A system graph can be depicted by a nodes x hyperedges table, where rows represent nodes and columns represent hyperedges, and the patterns of ones and zeros in the table represent connections [2].

Given a system, **S**, its hyperedges-only graph, $\mathbf{S}^{\#}$, consists of all nodes in **S** that are end points of hyperedges and all its hyperedges [2]. $S^{\#}$ denotes the corresponding system graph. Removing all the requirements that are not categorized yields desirable properties for the metrics [2, 7].

Given a system, **S**, with n nodes, we model its system graph S as a set of statistically independent samples from a probability distribution on the possible row patterns of its nodes x hyperedges table, p_l, l = 1, …, n_S, where n_S is the number of possible distinct row patterns. This probability distribution represents the model of stakeholder's preferences when categorizing requirements. The entropy of the distribution row patterns [60] is calculated by the following.

$$H(S) = \sum_{l=1}^{n_s} p_l(-\log\ p_l) \tag{2.1}$$

Entropy in this case is the average information per node. Base-two logarithms are used in information-theoretic calculations. The unit of measure is a bit. A bit is a commonly used measure of information in the communications field. When we estimate the distribution by the proportions, $\hat{p}_1$, of distinct row patterns that exist in the nodes x hyperedges table of S [64], entropy can be estimated by the following.

$$\hat{H}(S) = \sum_{i=0}^{n} \frac{1}{n+1}(-\log \hat{p}_{L(i)}) \tag{2.2}$$

where L(i) is a function that gives row pattern, l, of node i. Note that the summation is over the set of nodes (i), rather than the row patterns (l), as in Equation (2.1). Explicit modeling of the system boundary by the disconnected environment node in S assures that the possibility of the uncategorized requirement has a nonzero probability, even if it is not observed in the data. This, in turn, results in the desirable properties of Allen's metrics [2].

Given the hyperedges-only graph, $\mathbf{S}^{\#}$, the node subsystem graph, S_i, consists of all the nodes in $\mathbf{S}^{\#}$ and all hyperedges connected to the i^{th} node [2, 3]. Its system graph is denoted by S_i. This implies that the nodes X hyperedges table of S_i is the subset of columns of S that have a 1 in row i. We also model S_i as the probability distribution, estimated by the proportions of distinct row patterns, $\hat{p}_{l}$. Similar to Equation (2.2), we estimate the entropy of the distribution of row patterns by the following [2, 3].

$$\hat{H}(S_i) = \sum_{j=0}^{n} \frac{1}{n+1}(-\log \hat{p}_{L_i(j)}) \tag{2.3}$$

where $L_i(j)$ is function that gives the pattern index, l, of the jth row of S_i. Because each row of each S_i is a subset of the corresponding row of $S^{\#}$, $S^{\#}$ represents the joint distribution of all the S_i. Information theory states that the entropy of the joint distribution is less than or equal to the sum of the entropy of the components.

$$\sum_{i=0}^{n} H(S_i) \geq H(S^{\#}) \tag{2.4}$$

Watanabe shows that the difference is a measure of the relationships among the components [65]. *Excess entropy* [64] of $S^{\#}$ is defined as

$$C(S^{\#}) = \sum_{i=0}^{n} H(S_i) - H(S^{\#}) \tag{2.5}$$

Excess entropy is the average information in relationships. Connected nodes are related to each other by the presence of a hyperedge and the other nodes are related by the absence of that hyperedge. If the S_i are highly related to each other by common hyperedge connections and common disconnected nodes, then the excess entropy is high. Mohanty's approach [48] measures the complexity of a hypergraph [2, 4] based on excess entropy. Multiplying excess entropy by the number of nodes yields the amount of information (bits) in all the relationships among the nodes, (n+1) $C(S^{\#})$.

The complexity of a system, **S**, is given by the amount of information in relationships in its hyperedges only graph, less the contribution of the environment node [2, 3].

$$Complexity(\mathbf{S}) = (n+1)C(S^{\#}) - \left(\sum_{i=1}^{n} \left(-\log \hat{p}_{L_i(0)} \right) - \left(-\log \hat{p}_{L(0)} \right) \right) \qquad (2.6)$$

By Equations (2.2), (2.3), (2.5), and (2.6),

$$Complexity(\mathbf{S}) = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \left(-\log \hat{p}_{L_i(j)} \right) \right) - \sum_{i=1}^{n} \left(-\log \hat{p}_{L(i)} \right) \qquad (2.7)$$

Note that each summation begins with i = 1.

The complexity of row i in a system, **S**, is its contribution to the complexity of the system, given by [2, 3]

$$Complexity(i \mid \mathbf{S}) = \sum_{j=1}^{n} \left(-\log \hat{p}_{L_i(j)} \right) - \left(-\log \hat{p}_{L(i)} \right) \qquad (2.8)$$

In our application the complexity of a row, Complexity(i | **S**) is the contribution of one requirement to the overall complexity, Complexity(**S**) [2, 3]. By the definition,

$$Complexity(\mathbf{S}) = \sum_{i=1}^{n} Complexity(i \mid \mathbf{S}) \qquad (2.9)$$

If we add a column to a nodes x hyperedges table that is same as another existing column, then the $\hat{p}_1$ for all row patterns in not changed. This implies that a redundant column does not affect Complexity(**S**), nor its components, Complexity(i | **S**).

These metrics were extended to hypergraphs by Allen and Gottipati [3]. Ordinary graphs have edges with two ends. A hypergraph allows hyperedge to have multiple end points (connections). Hypergraphs can represent knowledge structures. Nodes in a hypergraph model the requirements and the hyperedges model the relationship among the requirements as perceived by the stakeholder.

The research presented in this dissertation falls between software requirements elicitation and high-level design in the software engineering life cycle. This work addresses how to represent and evaluate the understanding about software requirements among stakeholders. During the literature survey, elicitation techniques were found that try to solicit and specify requirements as perceived by stakeholders. No evidence was found of previous research work regarding representing software requirements as mental models and evaluating/comparing them to measure the degree of understanding/misunderstandings among stakeholders.

The next chapter outlines the experimental designs that were used to predict and reduce misunderstanding about requirements using the PFNET technique. It outlines the experimental set up for applying information theory-based metrics to measure consensus about requirements among stakeholders. Dr. Edward Allen [2] proposed these metrics.

CHAPTER III

EXPERIMENTAL DESIGN

3.1 Introduction

The experimental setup that was applied to predict misunderstandings between customer and developer groups in our early experimentation involved multiple student projects and one industry project. This chapter will also describe the experimental setup that applied a modified version of the PFNET technique in the second industrial setting at AmerInd Inc. Our early experiments were conducted at Mississippi State University (MSU) and at NORTEL based in Dallas, TX [68]. These experiments showed very encouraging results. The objective of the experimental design was to validate the ability of the PFNET technique to predict misunderstandings about requirements. In the early experiments we did not inform the stakeholders of the predictions made by this technique in terms of misunderstood, duplicate, and ambiguous requirements. The predictions made by the analysis of the PFNETs were only revealed after the software product had been delivered so that we could analyze the prediction against actual results. In addition, the correlation coefficients for the four systems built by MSU students of a Software Engineering class were not revealed, even though they predicted that misunderstandings existed at a very early stage.

Motivated by these results, additional experimentation was then conducted in industrial settings. One of the more important experiments was conducted during thesummer of 2003, at AmerInd Inc., based in Alexandria, VA. A modified version of the PFNET technique was used for this experimentation. This modified experimental design went a step further in terms of providing feedback to the stakeholders concerning the overall and individual correlation coefficients for the system and individual requirements respectively. Our purpose was also to validate the usefulness of this technique, which required us to conduct a facilitated session between stakeholders. Stakeholders needed to be made aware of what others thought about the requirements to resolve the differences that allowed them to leave the session with a better understanding. The experimental design also had to account for validation of the correlation coefficients as intuitively perceived by the stakeholders. This meant that if the stakeholders categorized the requirements in a similar manner, then this should be indicated by the high values of correlation coefficients (greater than 0.7). On the other hand, if the stakeholders did not agree on how the requirements were categorized then this should be indicated by the low values of correlation coefficients (less than 0.7). The changes in the initial designs, which led to modifications of the original technique, are described in the following section. The value of 0.7 was informally validated with the stakeholders at AmerInd for small-scale projects. During analysis of correlation coefficients this value seemed reasonable to identify potentially misunderstood requirements for all the student projects at MSU and the project at NORTEL.

Also described in this chapter is the experimentation accomplished in incorporating information theory-based software metrics to measure consensus among the stakeholders. These metrics were applied only to the data collected at AmerInd Inc. while applying the PFNET technique. The application of the PFNET technique results in hypergraphs and hyperedges. In other research by Allen [2, 3, 4, 5], information theory-based metrics have been proposed to measure various characteristics of such graphs and other related structures. The concept of hypergraphs with hyperedges fits very well with our observation in the way stakeholders categorized low-level requirements into high-level categories. The low-level requirements represent the nodes in the hypergraph and the high-level categories represent the hyperedges. This is explained in greater detail in section 3.5.

3.2 Experimental Design for Early Classroom Experimentation

In 1999, an investigation was initiated at Mississippi State University (MSU) to gather evidence supporting our belief that the Pathfinder technique could be a viable software engineering tool – particularly early in the software engineering life cycle [44]. Experiments conducted in the Department of Computer Science and Engineering at MSU to explore the utility of using PFNETs in a software engineering context is reported here. We were particularly interested in applying this technique to software requirement analysis and specification to learn what Pathfinder networks might reveal about requirement understanding at an early phase in the development process. Four development activities used in the initial experimentation were designed and implemented by a group of 4 to 6 students in a software engineering class (CS

4213/6213). An SRS document was generated during the requirement analysis and specification phase in conjunction with a customer located on campus. The SRS developed for each system was the basis for generating all subsequent PFNETs.

A graduate student was allowed to work with each team to create the necessary PFNETs - one for each team and customer. For these experiments, the graphs were undirected. Path distance was measured as the sum of edge weights associated with the edges traversed between a pair of nodes. For example, if three edges with edge-weights as one are to be traversed from a source node to the destination node, then the path distance between those two nodes is 3. A shortest path is defined as the minimum of all possible distances between any pair of nodes. The *structural property* of graphs reveals information about higher-order relations among the nodes. Comparing the structural properties can reveal structural similarities between graphs [29]. *Path distance* is a structural property of a graph since it is based on edge relations. Path distance was then used to compute similarity between given graphs. The *correlation coefficient* referred to in this work is the ratio of shared attributes to total attributes. In this particular experiment, the shared attributes are the paths with the same path distances between a pair of corresponding nodes in the two graphs, whereas, the total attributes would be the total number of paths between those corresponding nodes. The degree of similarity was measured by computing a correlation coefficient between the path distances of the graphs under consideration. Calculating the mean across all possible pairs in the two graphs gives us the *overall correlation coefficient*.

The resultant Pathfinder networks of developers and users were then compared and analyzed to check for similarities and dissimilarities. This was achieved by computing the overall correlation coefficient (cc) based on path distances of the two networks and also, by computing the correlation coefficient of the individual requirements in both networks for a particular project.

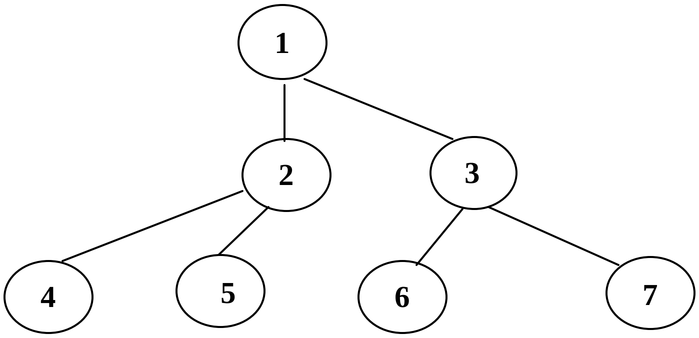


Figure 3.1 Undirected Graph A [44]

The graph shown in Figure 3.1 can be represented as the distance matrix shown in Table 3.1. Only the upper triangular half is shown in Table 3.1 since the graph is symmetric. All the edge weights are assumed to be one and the distance between two nodes is the sum of the edge weights along that path.

Table 3.1 Upper Triangular Distance Matrix for an Undirected Graph for Figure 3.1 [44]

Nodes	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	-	1	1	2	2	2	2
2		-	2	1	1	3	3
3			-	3	3	1	1
4				-	2	4	4
5					-	4	4
6						-	2
7							-

The distance vector for an undirected graph A represented by Table 3.1 is (1 1 2 2 2 2 2 1 1 3 3 3 3 1 1 2 4 4 4 4 2).

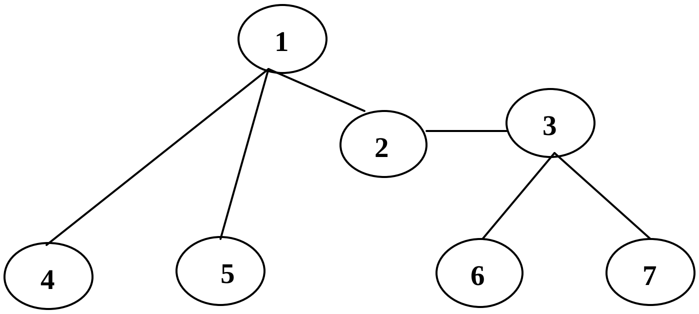


Figure 3.2 Undirected Graph B

The graph shown in Figure 3.2 can be represented as a distance matrix shown in Table 3.2. Again, only the upper triangular half is shown in Table 3.2 since the graph is symmetric. Here too, the edge weights are assumed to be one and the distance between two nodes is the sum of the edge weights along that path.

Table 3.2 Upper Triangular Distance Matrix for Graph B for Figure 3.2

Nodes	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	-	1	2	1	1	3	3
2		-	1	2	2	2	2
3			-	3	3	1	1
4				-	2	4	4
5					-	4	4
6						-	2
7							-

If we let **A** be the distance vector of the first graph A and let **B** be the distance vector of the second graph B, then the overall correlation coefficient cc_{pAB} is computed as:

$$cc_{pAB} = \frac{\sum (a-\bar{a})(b-\bar{b})}{\sqrt{\sum (a-\bar{a})^2 \sum (b-\bar{b})^2}} \qquad (3.1)$$

where a is the value of an element in the distance vector **A**, $\bar{a}$ is the mean of all elements in the distance vector of graph A, b is the value of a corresponding element in the distance vector **B,** and $\bar{b}$ is the mean of all elements in the distance vector of graph B. The possible values of the correlation coefficient range from –1 to 1. A value of zero is assigned to the correlation coefficient if either $\sum(a-\bar{a})^2$ or $\sum(b-\bar{b})^2$ is zero to prevent division by zero.

A value of –1 means the two graphs are very dissimilar and a value of 1 means the graphs are identical. The lower the value of the correlation coefficient the less similarity exists between the two graphs [23]. Similarly, the correlation coefficient of the path distance for two nodes can be calculated by using the same technique as used for the overall correlation coefficient. The only difference is that the path distance from a particular node to all other nodes in that graph is represented in the vectors **A** and **B**. For example the distance vector for Node 1 for the graph in Figure 3.1 from Table 3.1 is (1 1 2 2 2 2).

A Pathfinder graph can be represented as a distance matrix where each entry into the matrix is the distance between those corresponding nodes. Recall that in this work we are only concerned with undirected graphs. The distance matrix is represented as a vector of length $(n^2 – n) /2$ that covers all pair node distances. Once the distance vectors are determined for each graph to be compared, the correlation coefficient can be computed to determine the similarity/dissimilarity between the two networks. First, the path distances between all pairs of concepts is computed for both networks. The overall cc measures correlation between the corresponding path distances for all nodes between the two networks. Similarly, the cc for a single requirement can be computed. The overall correlation coefficient between graphs A and B is 0.79 and individual correlations are shown in Table 3.3.

Table 3.3 Path Distance Correlations Between Graphs A and B

Node	**cc_{pAB}**
1	0.26
2	0.26
3	0.79
4	0.86
5	0.86
6	0.86
7	0.86

For all experiments, the following heuristics were applied:

- A correlation coefficient of a network/node below 0.4 was assumed to indicate little or no similarity
- A correlation coefficient from 0.4 through 0.7 was assumed to indicate a moderate degree of similarity
- A correlation coefficient more than 0.7 was assumed to show very good to strong similarity.

We then assumed that the higher the value of the correlation coefficient, the more similar the mental models were. This may help software engineers measure the understanding of requirements between users and developers.

The Methodology for Early Experimentation (Classroom)

In generating PFNETs for each of the classroom experiments, the following procedures were used as reported in [44].

- **Step 1**: Identify the participants, namely the customers and developers who will participate in this process. Appropriate requirements document from which we extract the requirements also have to be identified.

- **Step 2**: Requirements were extracted from each SRS and recorded on index cards. After explaining the purpose of the experiment to participants, a set of index cards representing the requirements and the SRS document were given to each student and to the customer for each system under development.
- **Step 3**: Participants were asked to read the SRS in order to become familiar with the requirements on the index cards (a process not unlike that expected in a traditional software walkthrough/inspection process). Each student and the customer were then required to group the index cards into different categories based on perceived similarities and dissimilarities among the requirements. The customers and developers were not allowed to consult with each other. They were allowed to duplicate a requirement if they thought it belonged in more than one category.
- **Step 4**: The categories for each participant were collected and a similarity matrix (N by N, where N is the number of requirements for a particular project) was generated. A "1" was recorded in the cell of the similarity matrix, corresponding to a pair of requirements, if that pair of requirements co-occurred in a stack. We refer to this as the *co-occurrence count* for a pair of requirements. For example, if a pair of requirements appeared in three different categories, then a co-occurrence count of three was recorded in the corresponding cell of the similarity matrix. A zero was recorded if a pair of requirements did not co-occur in any of the stacks (categories). Thus, two similarity matrices, one for the group of developers (students) and one for the customer were generated. For example, consider four requirements R1, R2, R3, and R4 from an SRS written on index cards and given to a member of the developer

group. If the first developer (D1) placed the requirements into three categories say (R1, R2, R3), (R2, R3), (R2, R3, R4), where the parenthesis denote cards grouped in a particular stack, then the resulting similarity matrix is as shown in Table 3.4. The co-occurrence count for R2 and R3 is 3 because they appear together in three different categories. The co-occurrence count for R1 and R4 is 0 because they do not co-occur in any of the categories. Assume that a second developer (D2) placed the requirements into four categories say, (R1, R2, R3, R4), (R2, R3), (R2, R3, R4), and (R2, R4). The resulting similarity matrix is shown in Table 3.5. To generate a single PFNET (consensus) for both the developers D1 and D2, first the corresponding elements of the similarity matrices represented by Table 3.4 and Table 3.5 respectively, are added. The resulting matrix called the *consensus similarity matrix* is shown in Table 3.6.

Table 3.4 Similarity (Symmetric) Matrix for D1

	R1	**R2**	**R3**	**R4**
R1	-	1	1	0
R2	1	-	3	1
R3	1	3	-	1
R4	0	1	1	-

Table 3.5 A Similarity (Symmetric) Matrix for D2

	R1	**R2**	**R3**	**R4**
R1	-	1	1	1
R2	1	-	3	3
R3	1	3	-	2
R4	1	3	2	-

Table 3.6 A Consensus Similarity (Symmetric) Matrix for D1 and D2

	R1	**R2**	**R3**	**R4**
R1	-	2	2	1
R2	2	-	6	4
R3	2	6	-	3
R4	1	4	3	-

- **Step 5:** Dissimilarity matrices were generated for the group of developers and for the customer based on their similarity matrices. These are generated by subtracting each co-occurrence count in the similarity matrix from the maximum co-occurrence count in the same similarity matrix plus one (to avoid a zero dissimilarity count). The dissimilarity matrix for Table 3.7 is obtained by subtracting each co-occurrence count from the maximum co-occurrence count plus one (i.e., 4). The dissimilarity matrix for Table 3.4 is shown in Table 3.7. As a result, the pair of requirements that co-occurs the maximum number of times will be considered least dissimilar. Pairs of requirements that do not co-occur will have the maximum dissimilarity count. Similarly, the consensus dissimilarity matrix is next generated from the consensus similarity matrix. The dissimilarity matrix for the consensus similarity matrix represented by Table 3.6 is shown in Table 3.9.

Table 3.7 A Dissimilarity Matrix Derived from Table 3.4

	R1	**R2**	**R3**	**R4**
R1	-	3	3	4
R2	3	-	1	3
R3	3	1	-	3
R4	4	3	3	-

Table 3.8 A Dissimilarity Matrix Derived from Table 3.5

	R1	**R2**	**R3**	**R4**
R1	-	3	3	3
R2	3	-	1	1
R3	3	1	-	2
R4	3	1	2	-

Table 3.9 A Dissimilarity Matrix Derived from Table 3.6

	R1	**R2**	**R3**	**R4**
R1	-	5	5	6
R2	5	-	1	3
R3	5	1	-	4
R4	6	3	4	-

- **Step 6**: The dissimilarity matrices for the group of developers and that of the customer were used as input to the Pathfinder generation program. Thus, two Pathfinder networks, one for the developers (a consensus response), and one for the user were generated for each project. The consensus PFNET for graphs D1 and D2 represented as PFNET_{D1D2} ($r = \infty$, $q = 3$) that keeps the common edges from PFNET_{D1} and PFNET_{D2} is shown in Figure 3.5. The PFNET_{D1} ($r = \infty$, $q = 3$) for the dissimilarity matrix of Table 3.7 is shown in Figure 3.3. A web-based tool was implemented as part of a Master's project during the Fall 2001 semester [68], which automates this procedure.

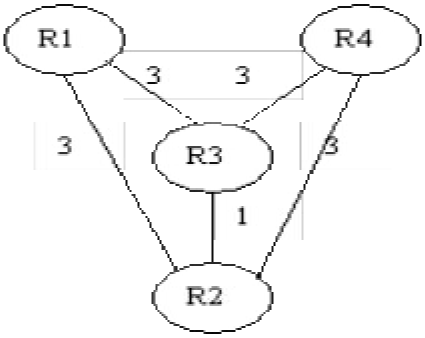


Figure 3.3 The PFNET $_{D1}$ (r = ∞, q = 3) from Table 3.7

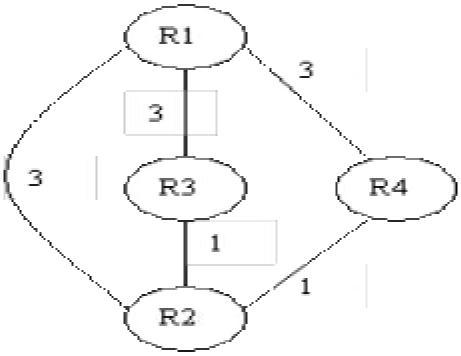


Figure 3.4 The PFNET $_{D2}$ (r = ∞, q = 3) from Table 3.8

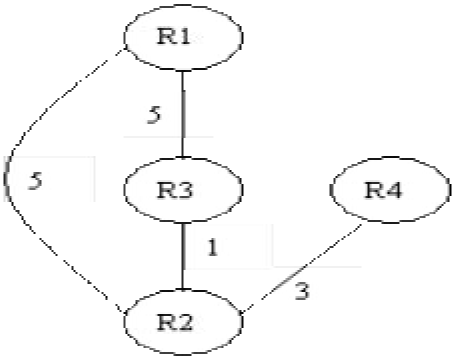


Figure 3.5 Consensus PFNET $_{D1D2}$ (r = ∞, q = 3) from Table 3.9

- **Step 7:** The resulting PFNETs are then correlated with each other producing a correlation coefficient (cc), which is used to measure similarity between the mental models. This is achieved by first generating minimum distance matrices for each of the graphs. These minimum distance matrices are shown in Table 3.10, Table 3.11, and Table 3.12 for the PFNETs of the developers D1, D2 and for consensus PFNET of both the developers respectively. They form the basis for generating the distance vectors and for computing the overall and individual correlation coefficients.

Table 3.10 Minimum Distance Matrix for PFNET_{D1}

	R1	**R2**	**R3**	**R4**
R1	-	3	3	6
R2	3	-	1	3
R3	3	1	-	3
R4	6	3	3	-

Table 3.11 Minimum Distance Matrix for PFNET_{D2}

	R1	**R2**	**R3**	**R4**
R1	-	3	3	3
R2	3	-	1	1
R3	3	1	-	2
R4	3	1	2	-

Table 3.12 Minimum Distance Matrix for PFNET_{D1D2}

	R1	**R2**	**R3**	**R4**
R1	-	5	5	8
R2	5	-	1	3
R3	5	1	-	4
R4	8	3	4	-

These distance vectors are used to compute the overall correlation coefficient needed to compare the two PFNETs and to measure their similarities and dissimilarities. The distance vector for an undirected graph is computed from the minimum distance matrix. For example, from Table 3.10, the distance vector for PFNET_{D1} is (3 3 6 1 3 3) and from Table 3.11, the distance vector for PFNET_{D2} is (3 3 3 1 1 2). This vector consists of all the elements above the diagonal elements of that matrix. Thus, the distance vector in this case has six elements. Similarly, we compute the correlation coefficient for individual requirements by first computing the distance vector for each node and then applying the cc formula. The distance vector for a single node, say R4, in Table 3.10 would be represented as (6 3 3).

The overall correlation coefficient (cc) between PFNET_{D1} and PFNET_{D2} is approximately 0.61. Table 3.13 shows the individual correlation coefficients generated. Requirements **R1** and **R2** have low values (less than 0.7), hence requirements **R1** and **R2** may be misunderstood. On the other hand, requirements **R3** and **R4** show higher values of correlation coefficients compared to other requirements. These requirements are considered well understood.

Table 3.13 Correlation Coefficients Between PFNET$_{D1}$ and PFNET$_{D2}$

ID.	Requirement	Correlation (cc)
1	R1	0.00
2	R2	0.49
3	R3	0.87
4	R4	0.87

If it is deemed necessary, to see how each individual's requirements compare with that of the consensus PFNET$_{D1D2}$, correlations may then be computed.

The next section describes how the technique described was applied at NORTEL and how the procedure described in this section was automated using a web-based tool.

3.3 Experimental Design for the NORTEL Experiment

For the purpose of the experiment at NORTEL, a web-based tool to group related requirements as perceived by each stakeholder was developed to eliminate the use of index cards [68]. At NORTEL, the two groups participating in the PFNET experiment were both development groups since there was not a "true" customer available. The results of the PFNET exercise between the two groups were then compared in the same manner as if one was a user and the other a developer.

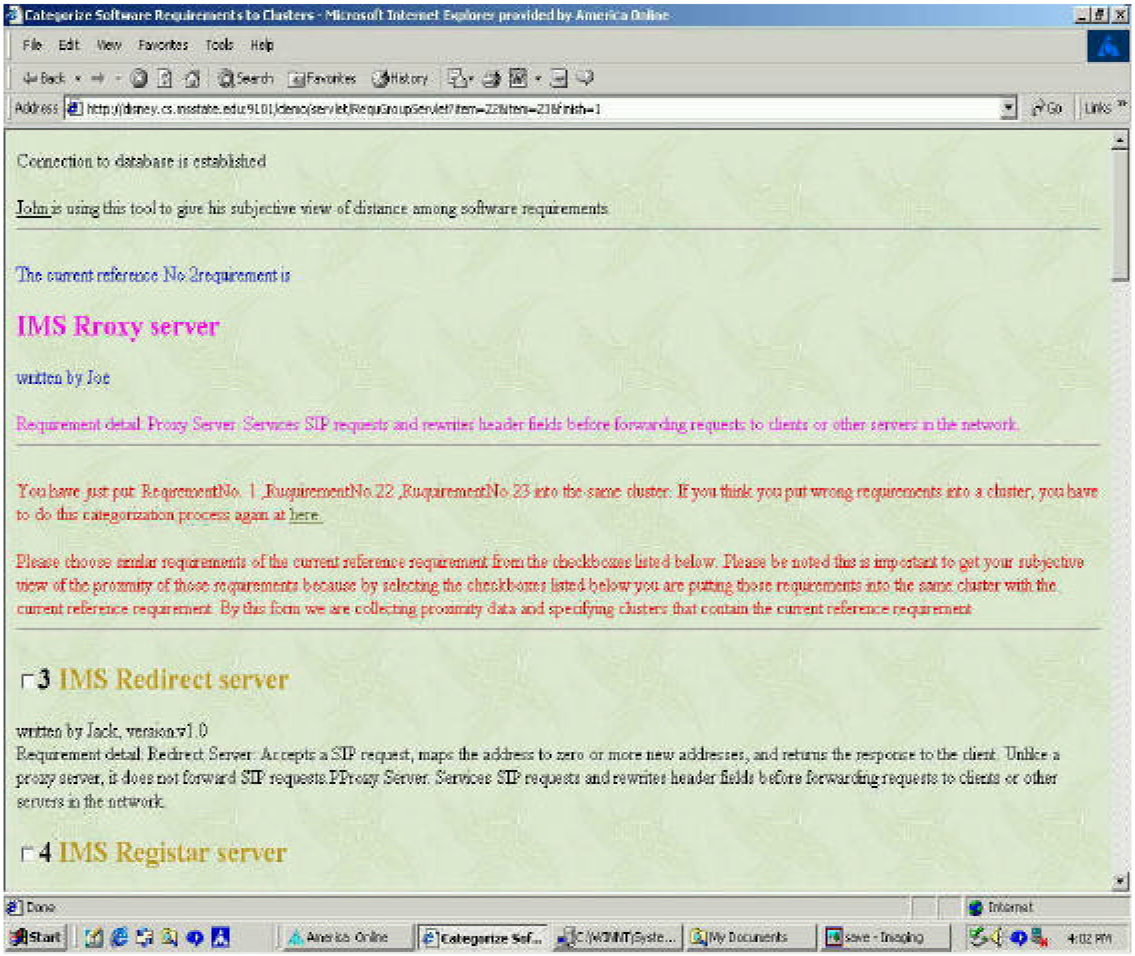
The Methodology Used at NORTEL

The procedure involved the following steps:

- **Step 1:** Identify the participants and the appropriate requirements document.
- **Step 2**: Each participant was given an account so that they could login using a web browser. A requirement and its description were displayed at top of each page. The rest of the requirements and their descriptions with a check box for each requirement followed as shown in Figure 3.6.

- **Step 3**: Each user was asked to check the box (es) of the other requirements if that requirement was perceived as related to the reference requirement at the top of the page. The end of the page had a submit button and also an option to change before submitting the information.

The rest of the steps i.e., **Step 4**, **Step 5**, **Step 6, and Step 7** are the same as described in section 3.2.


Categorize Software Requirements to Clusters - Microsoft Internet Explorer provided by America Online
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://disney.cs.msstate.edu:9101/demo/servlet/RequGroupServlet?item=22&item=23&finish=1
Go Links
Done
Internet
Start
America Online
Categorize Sof...
C:\WINNT\Syste...
My Documents
save - Imaging
4:02 PM

Connection to database is established

John is using this tool to give his subjective view of distance among software requirements

The current reference No 2requirement is

IMS Rroxy server

written by Joe

Requirement detail: Proxy Server: Services SIP requests and rewrites header fields before forwarding requests to clients or other servers in the network.

You have just put RequirementNo. 1 ,RuquirementNo.22 ,RuquirementNo.23 into the same cluster. If you think you put wrong requirements into a cluster, you have to do this categorization process again at here.

Please choose similar requirements of the current reference requirement from the checkboxes listed below. Please be noted this is important to get your subjective view of the proximity of those requirements because by selecting the checkboxes listed below you are putting those requirements into the same cluster with the current reference requirement. By this form we are collecting proximity data and specifying clusters that contain the current reference requirement.

☐ **3** IMS Redirect server

written by Jack, version:v1.0
Requirement detail: Redirect Server: Accepts a SIP request, maps the address to zero or more new addresses, and returns the response to the client. Unlike a proxy server, it does not forward SIP requests.PProxy Server: Services SIP requests and rewrites header fields before forwarding requests to clients or other servers in the network.

☐ **4** IMS Registar server

Figure 3.6 Proximity Data Collection Using Web Based Tool

3.4 Design for Experimentation at AmerInd Inc.

The groundwork for this research effort first involved finding a company willing to provide access to their projects. We then designed and implemented appropriate data collection and analysis tools. These tools automated the entire process of data collection, analysis [63], generation of graphs [63] and the generation of appropriate reports. During the summer of 2003, Mississippi State University (MSU), the National Science Foundation (NSF) and AmerInd Inc. jointly sponsored this research effort at AmerInd Inc. based in Alexandria, VA. Experiments were conducted on two software development projects. The first project will be referred to as the Professional Services Council (PSC) project and the second project will be referred to simply as the New Material Acquisition (NMA) project. A non-disclosure agreement prevents us from discussing details about the customer or project requirements.

The experiments at AmerInd required some modification to the technique presented in sections 3.2 and 3.3. While our original process worked well for small numbers of requirements, it did not work as well for large numbers. The changes were also useful improvements we found necessary to adjust to the commercial customers. The rationale for changes in the original technique is explained in detail at the end of this section. As part of these changes, a new metric was used to measure similarities/dissimilarites between the PFNETs of stakeholders. Consider two undirected graphs with the same set of nodes. The *neighborhood* property measures the number of nodes that are directly linked to a particular node say, ?. Let V be the set of all nodes in a graph. The neighborhood property is a structural property of a graph since it is based on edge relations [29]. This new metric's measure of similarity between graphs is based on the neighborhood property. Given a graph A with a particular node, say ?, A_v represents all nodes in the neighborhood of the node ?. Similarly B_v represents all nodes in the neighborhood of the same node ? in graph B. The neighborhood correlation [29] of the node, cc_{nAB}, between the two undirected graphs A and B is computed as:

$$cc_{nAB} = \frac{1}{n}\sum_{veV}\frac{|A_v \cap B_v|}{|A_v \cup B_v|} \tag{3.2}$$

If the denominator is zero then we employ the convention of assigning the correlation coefficient a value of zero. The denominator is zero when there are no other requirements in the neighborhood. This is the scenario where a requirement is not categorized into any of the pre-determined high-level categories. The correlation coefficients based on neighborhood property have values between 0 and 1. A value of 0 means that the node/graph is most dissimilar. A value of 1 means the node/graph is least dissimilar, i.e. very similar.

Suppose that we had N number of stakeholders. A PFNET was generated for each. Let i and j take values from 1 through N. Let $cc_{?ij}$ represent the correlation for a particular node across a pair of PFNETs represented by a value for i and j. For example $cc_{?12}$ is the neighborhood correlation for a given node ? between the PFNETs for stakeholder 1 and stakeholder 2. Similarly, for the same node ?, $cc_{?13}$ would represent the neighborhood correlation between PFNETs for stakeholder 1 and stakeholder 3. The average correlation coefficient for the same node across all pairs of PFNETs is computed as follows:

$$cc_{vavg} = \frac{1}{P}\sum_{i=1}^{N}\sum_{j=i+1}^{N} cc_{vij} \tag{3.3}$$

where ? ε V, P = $\begin{pmatrix} N \\ 2 \end{pmatrix} = N!/(N-2)!*2!$ where N is the number of stakeholders, N > 1 and, j ≤ N.

The two projects at AmerInd Inc. required us to identify requirements at two levels of abstraction. The requirements that were identified as having a higher level of abstraction will be referred as *high-level categories*. The requirements that were identified as being at a lower level of abstraction will be referred to as *low-level requirements*. As a result, the stakeholders were required to categorize the low-level requirements into one or more of the high-level categories if they perceived any relation or similarity. This exercise will be referred to as the *categorization activity*. Our assumption is that if stakeholders categorize low-level requirements into high-level categories in a similar manner, then they have common understanding about the requirements. This is because stakeholders who have a common understanding about the low-level requirements, would perceive similar relations to the high-level categories. Stakeholders are said to have reached a *consensus* about requirements when they categorize the low-level requirements in a similar manner. The requirements that are not categorized in a similar manner could be considered as potentially misunderstood.

The information submitted by each stakeholder during the categorization activity can be represented as *categorization tables*. A categorization table has low-level requirements as rows and high-level categories as columns. Each cell in a categorization table has an integer number that represents the number of stakeholders who categorized that low-level requirement (row) into a high-level category (column). The categorization tables of individual stakeholders are then simply added to generate the categorization table for the group of stakeholders (consensus). The categorization table for the group of stakeholders can then be used to validate the average correlation coefficients generated for each requirement for the entire group (see Equation (3.3)).

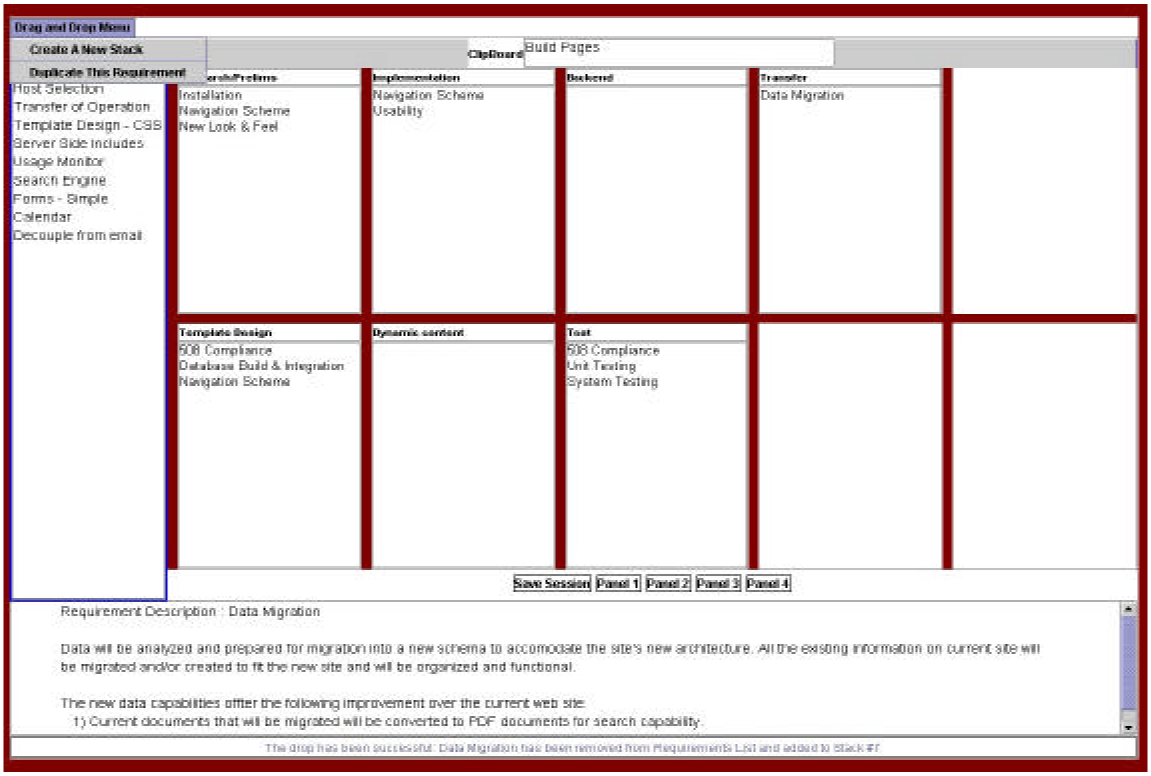
Section 0 describes each step of the modified technique in detail. Following that, the issues that contributed to the changes in the original technique are explained in detail.

The Methodology Applied at AmerInd Inc.

The application of the modified technique involved the following steps:

- **Step 1**: Identify the stakeholders for the project under study. The appropriate requirements document to be used is also identified in this step.
- **Step 2**: Identify the low-level requirements and high-level categories from a proposal, or a Concept of Operations document, or from a Software Requirement Specification (collectively referred to as requirements documents). This step is best achieved with help of an expert or a project manager who is closely involved with the project. The identification of high-level categories and low-level requirements provided a framework that established a minimum basic structure in terms of limited number of high-level categories and fixed number of low-level requirements for the stakeholders to categorize. It has to be noted that the stakeholders were allowed to create their own high-level categories if necessary. It is important that all the stakeholders have same definitions for both low-level requirements and high-level before the implementation of the next step.

- **Step 3**: In this step, each stakeholder is required to perform the categorization activity on an individual basis. A web-based tool was implemented for this purpose. Figure 3.7 shows the user-interface that was designed and implemented to aide each stakeholder in performing the categorization activity. The user-interface shows the low-level requirements on the left hand side and the high-level categories (in bold) as labels to the rectangular boxes on the right-hand side. The seven, labeled rectangular boxes represent the seven high-level categories that were identified for the PSC project. Each stakeholder was asked to drag and drop the low-level requirements into the labeled rectangular boxes. Stakeholders were allowed to create their own categories and edit the labels of those categories if necessary. They were also allowed to duplicate a low-level requirement if they perceived that it could be related to more than one high-level category. The rational for categorizing a low-level requirement into a high-level category will be referred to as the *perception* of that stakeholder. They were not allowed to delete any requirements. The description of a low-level requirement could be displayed by the interface by just clicking on that requirement. The description is displayed as scrollable text box at the bottom of the interface as shown in Figure 3.7. Each stakeholder was provided with a unique username and password. The session information was saved when they submitted their information.



Requirement Description : Data Migration

Data will be analyzed and prepared for migration into a new schema to accomodate the site's new architecture. All the existing information on current site will be migrated and/or created to fit the new site and will be organized and functional.

The new data capabilities offter the following improvement over the current web site:
1) Current documents that will be migrated will be converted to PDF documents for search capability.

The drop has been successful: Data Migration has been removed from Requirements List and added to Stack #7

Figure 3.7 Web Based Tool to Aide Users in Categorizing Requirements

- **Step 4**: This step generates N X N similarity matrix, where N is the number of low-level and high-level categories. For example, consider four low-level requirements R1, R2, R3, and R4 identified from a document. Also, consider four high-level categories, as S1, S2, S3, and S4 identified from the same document. Assume that a stakeholder (M1) had placed the requirements into four groups in **Step 2**. A group of requirements that belong to a high-level category are enclosed in parenthesis and separated by commas. Let (R1, R2, R3), (R2, R3), (R2, R3, R4), and (R2) be the four groups, each belonging to a high-level category represented by labels S1, S2, S3 and S4 respectively. The similarity matrix for stakeholder M1 is shown in Table 3.14.

Table 3.14 Similarity (Symmetric) Matrix for M1

	R1	**R2**	**R3**	**R4**	**S1**	**S2**	**S3**	**S4**
R1	-	0	0	0	1	0	0	0
R2	0	-	0	0	1	1	1	1
R3	0	0	-	0	1	1	1	0
R4	0	0	0	-	0	0	1	0
S1	1	1	1	0	-	0	0	0
S2	0	1	1	0	0	-	0	0
S3	0	1	1	1	0	0	-	0
S4	0	1	0	0	0	0	0	-

Assume that a second stakeholder (M2) had placed the requirements into four groups, say (R1, R2, R3), (R1, R3), (R2, R3, R4) and (R2, R3) in **Step 2**. Each group of low-level requirements belongs to a high-level category represented by labels S1, S2, S3 and S4 respectively. The similarity matrix for stakeholder M2 is shown in Table 3.15. The similarity matrix is populated by incrementing a cell by '1' when a high-level requirement co-occurs with a low-level requirement. A cell value of '0' is assigned otherwise. The resultant matrix is known as the similarity matrix. Assume that a third stakeholder (M3) had placed the requirements into four groups say (R1, R2, R3, R4), (R2, R3), (R2, R3, R4), and (R2) in categories S1, S2, S3 and S4 respectively.

Table 3.15 Similarity (Symmetric) Matrix for M2

	R1	**R2**	**R3**	**R4**	**S1**	**S2**	**S3**	**S4**
R1	-	0	0	0	1	1	0	0
R2	0	-	0	0	1	0	1	1
R3	0	0	-	0	1	1	1	1
R4	0	0	0	-	0	0	1	0
S1	1	1	1	0	-	0	0	0
S2	1	0	1	0	0	-	0	0
S3	0	1	1	1	0	0	-	0
S4	0	1	1	0	0	0	0	-

- **Step 5**: A dissimilarity matrix is generated by subtracting each co-occurrence count in the similarity matrix from the maximum co-occurrence count in the same similarity matrix plus one (to avoid a zero dissimilarity count) The dissimilarity matrix for Table 3.14 is obtained by subtracting each co-occurrence count from maximum co-occurrence count plus one (i.e., 2). The dissimilarity matrix for Table 3.14 is shown in Table 3.16. Pairs of requirements that do not co-occur will have the maximum dissimilarity count. The similarity and dissimilarity matrices (not shown) are also computed for stakeholder M3.

Table 3.16 A Dissimilarity Matrix (M1) Derived from Table 3.14

	R1	**R2**	**R3**	**R4**	**S1**	**S2**	**S3**	**S4**
R1	-	2	2	2	1	2	2	2
R2	2	-	2	2	1	1	1	1
R3	2	2	-	2	1	1	1	2
R4	2	2	2	-	2	2	1	2
S1	1	1	1	2	-	2	2	2
S2	2	1	1	2	2	-	2	2
S3	2	1	1	1	2	2	-	2
S4	2	1	2	2	2	2	2	-

Table 3.17 A Dissimilarity Matrix (M2) Derived from Table 3.15

	R1	**R2**	**R3**	**R4**	**S1**	**S2**	**S3**	**S4**
R1	-	2	2	2	1	1	2	2
R2	2	-	2	2	1	2	1	1
R3	2	2	-	2	1	1	1	1
R4	2	2	2	-	2	2	1	2
S1	1	1	1	2	-	2	2	2
S2	1	2	1	2	2	-	2	2
S3	2	1	1	1	2	2	-	2
S4	2	1	1	2	2	2	2	-

- **Step 6**: The dissimilarity matrix generated for each stakeholder becomes input to the Pathfinder generation program. This results in the generation of text files each containing the Pathfinder network represented as nodes and edge weights for each stakeholder. Table 3.18 shows a typical text file generated by the Pathfinder generation program. The first column represents the number of edges in the PFNET. Columns two and three show the node numbers that represent requirements. The fourth column shows the weight associated with each edge. It has to be noted that node numbers 1, 2, 3, and 4 correspond to labels R1, R2, R3, and R4. Node numbers 5, 6, 7, and 8 correspond to labels S1, S2, S3, and S4 respectively. The graphical representation of this table is shown in Figure 3.8.

Table 3.18 Text File Representing PFNET$_{M1}$(r = ∞, q =7)

	Edges		Weights
1	1	5	1
2	2	5	1
3	2	6	1
4	2	7	1
5	2	8	1
6	3	5	1
7	3	6	1
8	3	7	1
9	4	7	1

- **Step 7:** The text file is then input to a tool that automates the generation of correlation coefficients and visualization (graph generation) of the PFNETs [63]. The interface that was designed and implemented for analysis and generation of PFNETs is shown in Figure 3.11. For the purpose of clarity in illustration, the original PFNETs generated by the tool here were redrawn with node labels rather than with just node numbers. The PFNET for M1 represented by PFNET$_{M1}$ (r = ∞, q =7) is shown in Figure 3.8. Similarly, PFNET$_{M2}$ (r = ∞, q =7) for M2 shown in Figure 3.9, and PFNET$_{M3}$ (r = ∞, q =7) for M3 shown in Figure 3.10 are generated. The categorization tables for stakeholders M1, M2 and M3 are shown in Table 3.19, Table 3.20, and Table 3.21respectively.

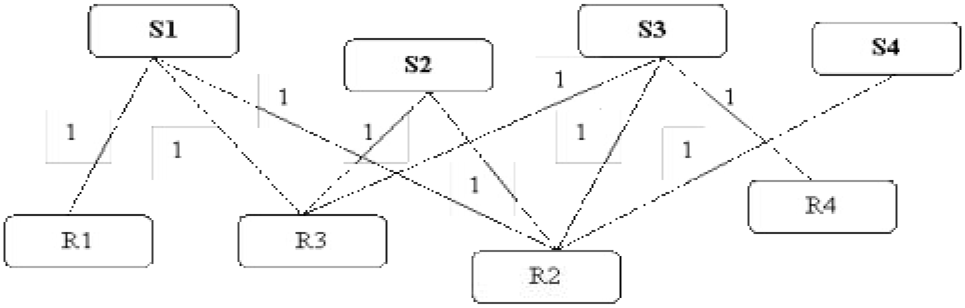


Figure 3.8 PFNET$_{M1}$ (r = ∞, q = 7)

Table 3.19 Categorization Table for Stakeholder M1

Req ID	Requirement	S1	S2	S3	S4
1	R1	1			
2	R2	1	1	1	1
3	R3	1	1	1	
4	R4			1	

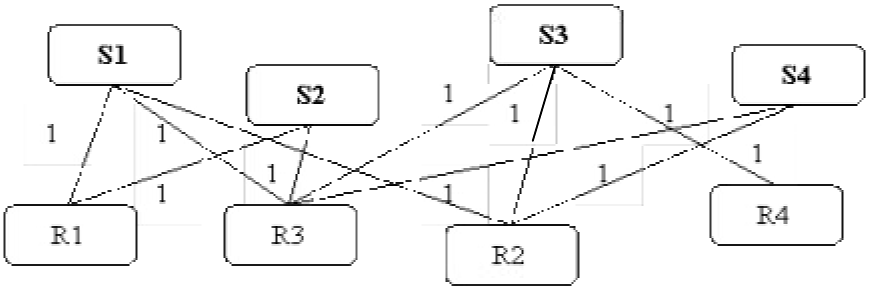


Figure 3.9 PFNET$_{M2}$ (r = ∞, q = 7)

Table 3.20 Categorization Table for Stakeholder M2

Req ID	Requirement	S1	S2	S3	S4
1	R1	1	1		
2	R2	1		1	1
3	R3	1	1	1	1
4	R4			1	

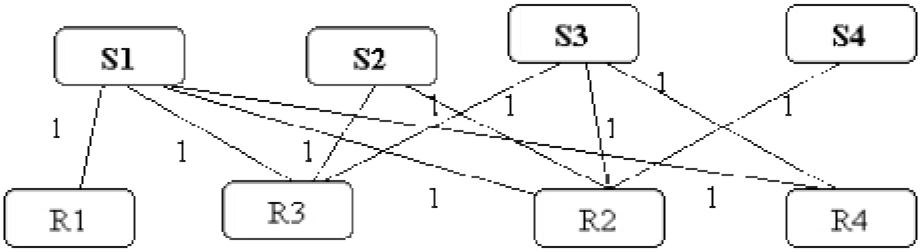


Figure 3.10 PFNET$_{M3}$ ($r = \infty$, $q = 7$)

Table 3.21 Categorization Table for Stakeholder M3

Req ID	Requirement	S1	S2	S3	S4
1	R1	1			
2	R2	1	1	1	1
3	R3	1	1	1	
4	R4	1		1	

In Table 3.21, stakeholder M3 categorized requirement R2 (second row) into all the four categories since there is a one under the four columns. An empty cell in the categorization table means that, a low-level requirement represented by a row was not categorized into that high-level category.

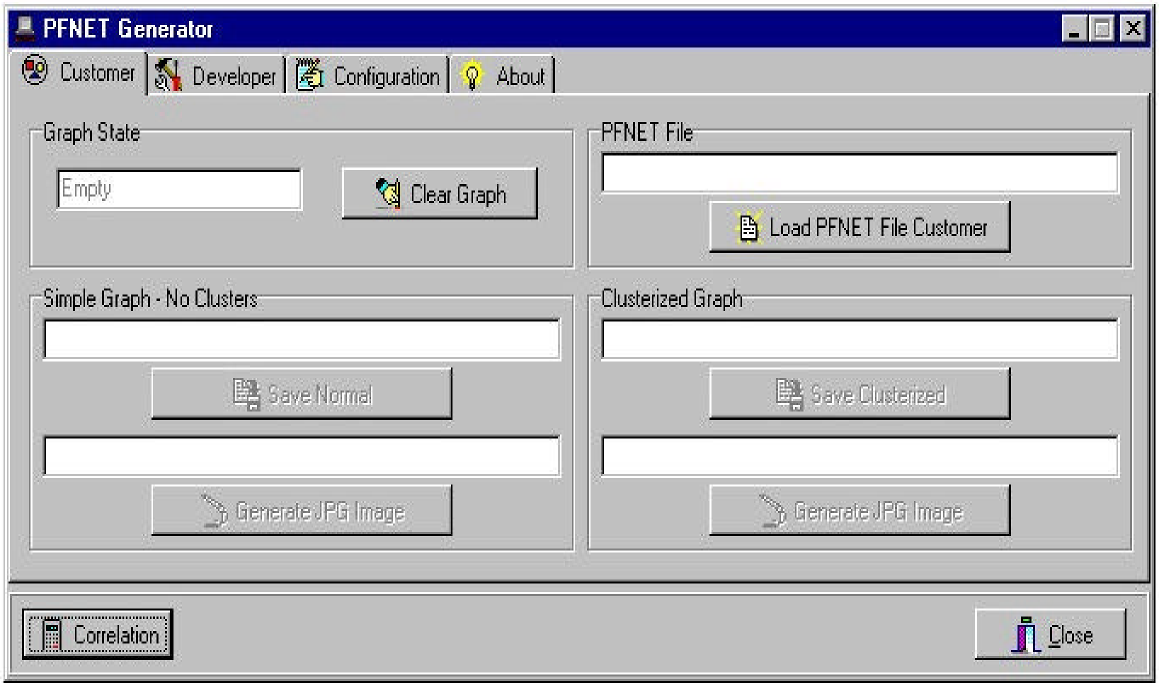


Figure 3.11 Tool Interface for Automatic Analysis and Generation of PFNETs

Thus, the neighborhood correlations computed from the PFNETS shown in Figure 3.8, Figure 3.9, and Figure 3.10 are shown in Table 3.22. The last column of this table represents the average correlation coefficient for each requirement computed according to Equation (3.3). Requirements R1 and R4 seem to have low values of correlations compared to the remaining low-level requirements in that table. The consensus categorization table for the three stakeholders is shown in Table 3.23. The individual categorization tables for the three stakeholders M1, M2, and M3 were added to generate the consensus categorization table. In Table 3.23, requirement R1 is categorized into the high-level category S1 by three stakeholders and one stakeholder categorized it into S2. This indicates disagreements among the stakeholders. The consensus categorization tables can be used to validate the correlation coefficients computed from the PFNETs of individual stakeholders.

Table 3.22 Neighborhood Correlations for the PFNETS

Requirement ID.	cc_{n12}	cc_{n13}	cc_{n23}	cc_{navg}
R1	0.50	1.00	0.50	0.67
R2	0.75	1.00	0.75	0.83
R3	0.75	1.00	0.75	0.83
R4	1.00	0.50	0.50	0.67
		Average		**0.75**
S1	1.00	0.75	0.75	0.83
S2	0.33	1.00	0.33	0.56
S3	1.00	1.00	1.00	1.00
S4	0.50	1.00	0.50	0.67
		Overall Average =		**0.76**

Table 3.23 Categorization Table for M1, M2 and M3 (Consensus)

Req ID	**Requirement**	**S1**	**S2**	**S3**	**S4**
1	R1	3	1		
2	R2	3	2	3	3
3	R3	3	3	3	1
4	R4	1		3	

- **Step 8:** The goal in this step is to facilitate a session to discuss the requirements having low values of correlation coefficients (less than 0.7). The discussion during this session is essential to identify ambiguous, duplicate and misunderstood requirements. This session could also be used to resolve differences and establish consensus among stakeholders about potentially misunderstood requirements. During the session, the requirements with the lowest correlation were discussed one at a time. All sessions for the two projects at AmerInd Inc. lasted for not more than an hour.

The facilitated sessions that were conducted between stakeholders involved the following steps:

a. Each stakeholder was given the categorization information initially submitted during **Step 3**. They were also given a rating sheet consisting of the requirement and instructions to rate each requirement on a scale of 1 to 5. The rating indicates the opinion of the stakeholders about that requirement being potentially misunderstood. Each stakeholder was also given the requirements document that contained a description of the requirements.

b. Every stakeholder was required to verbally state their perception about the low-level requirement being discussed. This step helped the other stakeholders to understand and evaluate their own perception about the requirement. It usually helped the stakeholders in starting a dialogue when perceptions differed.

c. The stakeholders were then required to rate the low-level requirement on a scale of 1 through 5. The rating for a requirement were taken for the purpose of validating the correlation coefficients with the stakeholder’s assessment about misunderstanding. The ratings could potentially help with determining a *threshold correlation coefficient* value for that project. The threshold correlation coefficient is described as the minimum value of a correlation coefficient for which a requirement is considered well understood by the stakeholders. A subjective estimate of 0.7 was used as the threshold correlation coefficient for our experiments. Any correlation coefficient that has a value less than the threshold value was considered potentially misunderstood. We believe determining a threshold value for the correlation coefficient will enable the stakeholders to focus only on the potentially misunderstood requirements.

It has to be noted that, compared to the original technique described in sections 3.2 and 3.3, the modified technique has an additional step (**Step 8**) intended to facilitate a session between stakeholders. The correlations that were generated in the original technique were never actually used to reduce the misunderstandings among stakeholders. However, in the modified technique, there is an explicit effort to resolve the misunderstandings and the categorization activity is repeated if possible. The similarities in the categorization tables validate the consensus among the stakeholders. This is attributed to the previously held facilitated session. The consensus among stakeholders was also reflected in the high values of correlation coefficients computed from PFNETs.

The following general reasons contributed to the changes made in the original technique:

- **Scaling issue**: In the modified technique, low-level requirements and high-level categories were identified up front. In the original technique, no high-level categories are initially identified. Each stakeholder created these categories during the categorization activity. The identification of these categories was essential to scale the original technique to work with a medium-scale project with a little over one hundred requirements. All the projects in our initial experimentation had only tens of requirements. As the number of low-level requirements increases, so do the high-level categories. Early identification of the high-level categories decreases the mental workload of the stakeholders and saves time during the categorization activity. Stakeholders were given the option to create additional high-level categories.

- **Facilitation issue**: The identification of high-level categories was helpful to the stakeholders in articulating their perception about requirements. The stakeholders did not have to remember the high-level categories since they were already identified and agreed upon. On the other hand, the stakeholders found it difficult to remember what category they had in mind when the original technique was put to test for the PSC project. The stakeholders are very likely to forget the high-level categories since the facilitation session is usually held a few days after the categorization activity. Thus, early identification of high-level categories upfront saved time during facilitation sessions.
- **Consensus issue**: The identification of high-level categories helped the stakeholders to resolve their misunderstandings about requirements. In order to build consensus about low-level requirements, it was first necessary for the stakeholders to resolve the differences in categorization during the facilitated session. Resolving their differences in categorization for a requirement entails that there be consistent agreement about the low-level requirement and high-level categories. In the original technique, the stakeholders could consistently agree on only the low-level requirements. The high-level categories would not be consistent across the stakeholders since they were all created during the categorization activity. Thus, during a facilitated session, to resolve the differences, the stakeholders would have to first agree on the high-level categories. Only then, the rationale given by a

stakeholder for categorizing a requirement into a high-level category made sense to other stakeholders. It then became easier to comprehend the differences when compared to their rationale. During a facilitated session, it was found difficult to resolve differences unless the stakeholders had prior agreement on definitions of high-level categories. Because the modified technique identified the high-level categories early, there was prior agreement about them among stakeholders. The earlier the stakeholders resolved the differences in categorization about individual requirements, the easier it was to build needed consensus.

Reaching a consensus without identifying the high-level categories seemed to take more iterations and effort, even for a small-scale project. Table 3.24 shows the overall correlation coefficients after two iterations for the PSC project without pre-determining the high-level categories. There was very little improvement in the overall correlation coefficient even after the first facilitated session.

Table 3.24 Overall Correlation Coefficients Without Step 2 Activity

	Overall Correlations
Iteration 1	0.26
Iteration 2	0.36

- **Categorization issue**: The stakeholders' thoughts about how low-level requirements were related to high-level categories changed during the experimentation at AmerInd Inc. We initially assumed that when requirements were grouped together based on relatedness or similarities they were all related to one another. This worked well during our initial experimentation because high-level categories were not identified early. We had to reassess this assumption at AmerInd based on the analysis of recorded facilitated sessions and interviews with stakeholders. It was found that when low-level requirements were grouped under a high-level category, the stakeholders did not necessarily see any relationship between low-level requirements. They seemed to only see a relation or similarity between the low-level requirement and its high-level category. Based on this discovery, the following changes were adopted:

a. We changed the way co-occurrence counts were assigned while computing similarity matrices. The co-occurrence count of '1' was assigned when a low-level requirement co-occurs with a high-level category. A co-occurrence value of '0' was assigned otherwise. This modification gave more weight to the lower-level requirement and its high-level category by incrementing the co-occurrence count by 1 whenever they co-occur. In order to model the relationship between the low-level requirement and the high-level category, we had to include the high-level category as a node in the PFNET. This was not the case with the PFNETs generated by the original technique.

b. We changed the property of the graph used to assess similarities between PFNETs from path distance in original technique to neighborhood in the modified technique. This is because the stakeholders saw a relation between low-level requirements and their immediate neighbor, i.e. high-level categories. The correlation coefficient based on the neighborhood property intuitively measures the similarities between nodes based on their links to immediate neighbors (directly linked). On the other hand, the correlation coefficient based on path distance measures similarities between nodes based on their links to all other nodes even if they are not directly linked. Two nodes are not directly linked when the path between them must include an intermediate node. Thus, the correlation coefficients for nodes based on the neighborhood property were more intuitive than the correlations coefficients for nodes based on path distance in our experiments at AmerInd Inc.

The changes adapted by the original technique at AmerInd helped us to better model the perception, scale and validate the PFNET technique.

The next section describes our effort to apply information theory-based metrics to measure consensus among stakeholders based on the categorization information collected at AmerInd Inc. This research effort is related to our previous work because measuring consensus among stakeholders was, in a sense, intended towards establishing a formal framework to gauge understanding about requirements. Moreover, both techniques are intended to focus the discussions among stakeholders on requirements that have divergent perceptions or that lack consensus.

3.5 Experimental Design to Apply Information Theory-Based Metrics

During the experimentation at AmerInd each stakeholder grouped low-level requirements into high-level categories. The raw data used in this research for applying the information theory-based metrics are the categorization tables developed from the stakeholder data. It should be noted that the categorization information here has a different representation when compared to that of the PFNET technique. Here all the information was represented as bits. We use the same terms, like the *categorization table* and *consensus categorization tables*, for the purpose of being consistent with our earlier terminology.

The categorization information can be transformed into a categorization table for each stakeholder or for a group of stakeholders (consensus). For example, assume that there were ten low-level requirements labeled R1 through R10 which were extracted from the requirements document. Assume five high-level categories labeled S1 through S5 have also been identified from the requirements document. Further, assume that there are two stakeholders, M1 and M2. Let (R1, R2, R3), (R3, R4, R5, R6), (R4, R9, R10), (R6, R7, R8), and (R3) be the five groups submitted by stakeholder M1, each group belonging to a high-level category represented by labels S1, S2, S3, S4, and S5 respectively. Let (R1, R2, R3, R7, R10), (R3, R4, R5, R6), (R2, R4, R9, R10), (R1, R6, R7, R8), and (R3, R8) be the five groups submitted by the stakeholder M2, again each group belonging to a high-level category represented by labels S1, S2, S3, S4 and S5 respectively. Table 3.25 is an example of a categorization table for the stakeholder (M1). A "1" is assigned if the stakeholder has categorized a low-level requirement (row) into a particular high-level requirement (column). Otherwise, a "0" is assigned. For example, the binary edge pattern for requirement R4 i.e., 01100, suggests that the stakeholder had categorized this requirement into category S2 and category S3. Each row in the table is referred to a *binary row pattern*.

Table 3.25 Categorization Table for Stakeholder (M1)

Requirement	S1	S2	S3	S4	S5
R1	1	0	0	0	0
R2	1	0	0	0	0
R3	1	1	0	0	0
R4	0	1	1	0	0
R5	0	1	0	0	0
R6	0	1	0	1	0
R7	0	0	0	1	0
R8	0	0	0	1	1
R9	0	0	1	0	0
R10	0	0	1	0	0

Table 3.25 is equivalent to an undirected hypergraph. Each row corresponds to a node and each column corresponds to a hyperedge. This is shown in Figure 3.12. Hypergraphs are knowledge structures with nodes as concepts and relationships among the concepts are hyperedges. Ordinary graphs have edges with two end points but hypergraphs have multiple end points (connections).

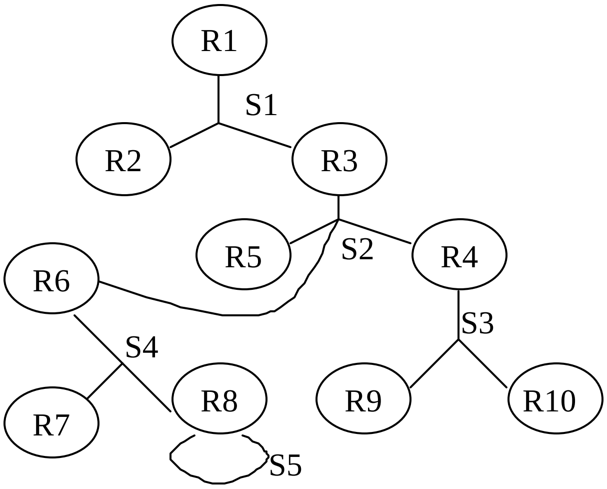


Figure 3.12 Hypergraph for the Categorization Table

A *consensus table* for a group of stakeholders can be constructed by concatenating the binary row patterns of every requirement (row). For example, suppose one stakeholder has a binary row pattern 10000 for R1, but another stakeholder has a binary row pattern of 11000 for R1. The consensus table will have a binary row pattern of 1000011000 for R1. This concatenation of binary row patterns can be extended to multiple stakeholders. We use the concatenation symbol '|', to denote concatenation of multiple categorization tables. For example, the consensus table that results by concatenating the categorization tables of stakeholders M1 and M2 is represented as M1| M2. Assume that for the same set of requirements, Table 3.26 represents the categorization table for a second stakeholder. The consensus table of both stakeholders is shown in Table 3.27. It should be noted at this point that all stakeholders work with the same set of low-level requirements and pre-determined high-level categories.

Table 3.26 Categorization Table (M2)

Requirement	S1	S2	S3	S4	S5
R1	1	0	0	1	0
R2	1	0	1	0	0
R3	1	1	0	0	1
R4	0	1	1	0	0
R5	0	1	0	0	0
R6	0	1	0	1	0
R7	1	0	0	1	0
R8	0	0	0	1	1
R9	0	0	1	0	0
R10	1	0	1	0	0

Table 3.27 Consensus Categorization Table (M1|M2)

Requirement	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
R1	1	0	0	0	0	1	0	0	1	0
R2	1	0	0	0	0	1	0	1	0	0
R3	1	1	0	0	0	1	1	0	0	1
R4	0	1	1	0	0	0	1	1	0	0
R5	0	1	0	0	0	0	1	0	0	0
R6	0	1	0	1	0	0	1	0	1	0
R7	0	0	0	1	0	1	0	0	1	0
R8	0	0	0	1	1	0	0	0	1	1
R9	0	0	1	0	0	0	0	1	0	0
R10	0	0	1	0	0	1	0	1	0	0

3.5.1 Methodology to Use Information-theory Based Metrics

This research was accomplished using data gathered at AmerInd Inc. The information-theory based metrics for hypergraphs fits well with the categorization data that was collected for the two projects there.

Lack of consensus is determined based on the amount of information in the patterns of categorization. We adopt information theory as the basis for measurement because each categorization decision is an element of information. The following method for assessing the consensus and resolving differences in perception among stakeholders has been proposed and was the basis for our experimental effort.

Step 1: Extract low-level and high-level categories from the requirements documents.

Step 2: Use the high-level categories as an initial set of categories.

Step 3: Identify the stakeholders who will participate in the categorization activity.

Step 4: Provide each stakeholder with brief descriptions of both low-level and high-level categories before and during the categorization activity to remind participants about the meaning of each requirement.

Step 5: Each stakeholder is required to stack all the low-level requirements into predetermined categories or new categories. The stakeholders are allowed to create these new categories if they deem it necessary.

Step 6: Calculate the information theory-based complexity metric [2, 3] at a system level, namely the total complexity and complexity of individual requirements i.e., the contributions of each requirement to the total.

Step 7: Form a consensus data set by concatenating all stakeholders' data.

Step 8: Calculate the information complexity of the consensus data, both for the total system and for each requirement.

Step 9: Assess the degree of consensus overall and for each requirement. An additional goal of this step is to see what information is revealed by the metrics and also pick the metric(s) that seems to be most intuitive and predict misunderstandings due to divergent views of stakeholders.

Step 10: Guided by the assessment of the metrics, facilitate a meeting of stakeholders to discuss divergent understandings of requirements. This may result in the resolution of many points of ambiguity or disagreement.

We also applied *system size* [2, 3] and *system coupling* [2, 3] measures to the data obtained from AmerInd Inc. The system size and system coupling measures did not contribute any more useful information to the analysis of data compared to that of the complexity measure. The analysis of the data basically consisted of measuring the consensus among stakeholders at a system and requirement level. For each stakeholder, we also performed analysis to identify requirements put in a single category and requirements that were not categorized into any of the predetermined categories. Identifying such requirements gives information about divergent understanding among the stakeholders. The following section will only outline the measurement of consensus using the complexity metric at system and requirement level.

3.5.2 Measuring Consensus Using Information Theory-Based Metric

Suppose a set of stakeholders perfectly agree. A consensus table formed by concatenating the row patterns for each requirement, consists of repetitions of same patterns for a given requirement. If the corresponding columns for each category have the same patterns of ones and zeros, then the repeated categories are redundant. The consensus table's complexity measurement will be same as for one categorization table, because each $\hat{p}_l$ for the consensus table will be same as the corresponding $\hat{p}_l$ of a categorization table. This means that there is no extra information in row patterns of the consensus table compared to the individual tables of stakeholders because they all perfectly agree on the categorization information.

On the other hand, suppose the stakeholders do not agree in the way they categorize low-level requirements. The consensus table's complexity will be higher than any of the individual's categorization table. This means that there is more information in the consensus categorization than in any of the individual categorizations because of the differences in perception. Thus, lack of consensus can be measured in bits by taking the difference between consensus measurement and the maximum of individual category measurements. Assume that there are *s* number of stakeholders represented as M1, M2 …, Ms. The lack of consensus on a system level may be computed as

$$\begin{aligned} LackofConsensus(M1,...,Ms) &= Complexity(\mathbf{S}_{M1|...|Ms}) - \\ &\max(\,Complexity(\mathbf{S}_{M1}),...,Complexity(S_{Ms})\ ) \end{aligned} \qquad (3.4)$$

The lack of consensus for individual requirements may be computed as

$$\begin{aligned} LackofConsensus_{i}\,(M1,...,Ms) &= Complexity(i\,|\,\mathbf{S}_{M1|...|Ms}) - \\ &\max(\,Complexity(i\,|\,\mathbf{S}_{M1}),...,Complexity(i\,|\,\mathbf{S}_{Ms})\ ) \end{aligned} \qquad (3.5)$$

where i is the index of rows, and M1|…|Ms refers to the consensus table.

Thus applying the complexity measurement for the entire system and individual requirements for M1, M2 and the consensus data (M1|M2) in bits are shown in Table 3.28. The values are computed using an automated tool developed at MSU [33].

Table 3.28 Complexity Measurements for the Two Stakeholders

	Scope of data		
	M1	M2	M1|M2
Entire System	90.60	144.19	169.21
Requirement.	M1	M2	M1|M2
R1	15.09	17.38	21.38
R2	6.29	6.29	6.29
R3	8.13	13.46	16.21
R4	5.38	17.38	18.38
R5	6.38	17.38	21.13
R6	15.09	15.63	18.38
R7	6.38	17.38	18.38
R8	6.38	6.29	9.54
R9	6.38	17.38	21.13
R10	15.09	15.63	18.38

The lack of consensus at the system level is computed by Equation (3.4) as 169.21 – max (90.60, 144.19) = 25.02 bits. This indicates that there is a small disagreement between the two stakeholders.

The next chapter presents the analysis of the results obtained by applying the PFNET technique on the student projects at MSU, a project at NORTEL and two projects at AmerInd Inc. The results obtained from our experimentation with information theory-based software metrics that were applied to the two projects at AmerInd Inc. are also presented.

CHAPTER IV

RESULTS AND ANALYSIS

This chapter describes the results obtained from the early experimentation that involved five student projects and one project at NORTEL. The results from the experimentation at AmerInd Inc. are also discussed. The AmerInd results are discussed separately because, as noted in Chapter III, we modified the procedures used as we moved to a larger project. The results that were obtained using information theory-based software metrics are also discussed separately since they represent a very different approach.

4.1 Results of Early Experimentation for Student Projects

- The four development activities used in the initial experimentation were designed and implemented by a group of 4 to 6 students in a software engineering class (CS 4213/6213) [44]. An SRS document was generated during the requirements analysis and specification phase in conjunction with a customer located on campus. The SRS developed for each system was the basis for generating all subsequent Pathfinder networks. For the purpose of completeness and understanding, a brief description of each development activity follows:

- **System 1**: A system called the COOP Recommendation Process System (CORPS) was designed and implemented for the purpose of managing and distributing recommendations for students to companies participating in the MSU COOPprogram. The PFNETs for the developers and customers are shown in Figure 4.3 and Figure 4.4 respectively.
- **System 2**: A system to manage patient appointments with doctors (referred to as DocTIME) at the Longest Student Health Center of MSU was designed and implemented. The PFNETs for the customer and developers are shown in Figure 4.5 and Figure 4.6 respectively.
- **System 3**: A web-based system to manage the Career Day event called the Career Day Registration System (CDRS) was designed and implemented for the MSU Career Services Center. The PFNETs for the customer and developers are shown in Figure 4.7 and Figure 4.8 respectively.
- **System 4**: A system called Transportation Support System (TSS) was designed and implemented to record and maintain the rental and maintenance information of the MSU motor pool for the transportation department of MSU. The PFNETs for the customer and developers are shown in Figure 4.9 and Figure 4.10 respectively.

The following notation and symbols can be used to interpret the PFNETs for customers and developers reported in Figures 4.3 through 4.11:

- Each rectangle is a requirement with the requirement identification labeled
- Ordinary links between any pair of requirements are represented as thin lines.

- Nodes enclosed in a circle represent requirements that are connected by two or more links (see Figure 4.1).

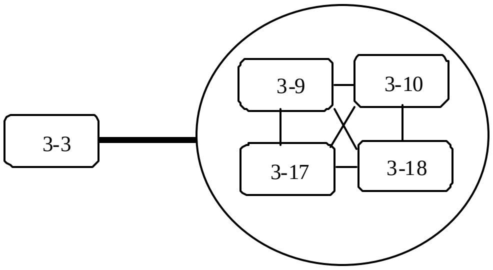


Figure 4.1 Graph Notation for Circle and Thick Lines

- A thick line connecting a node and a circle means that every node inside the circle is connected to the node outside the circle (see Figure 4.1).
- Requirements that are enclosed in a circle and connected by a star symbol means that those requirements form a fully connected graph and their link weight is the same (see Figure 4.2). Such a group of requirements are referred to as a *clique*.
- When two cliques are connected by a thick link, then every node inside one clique is linked to every other node inside the second clique by equal edge weights (see Figure 4.2).

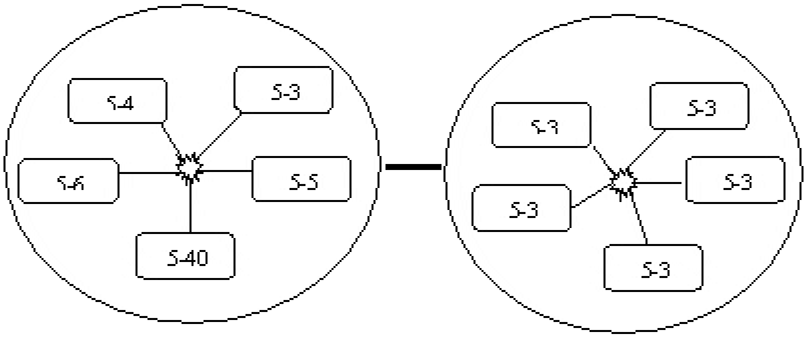


Figure 4.2 Graph Notation for Cliques

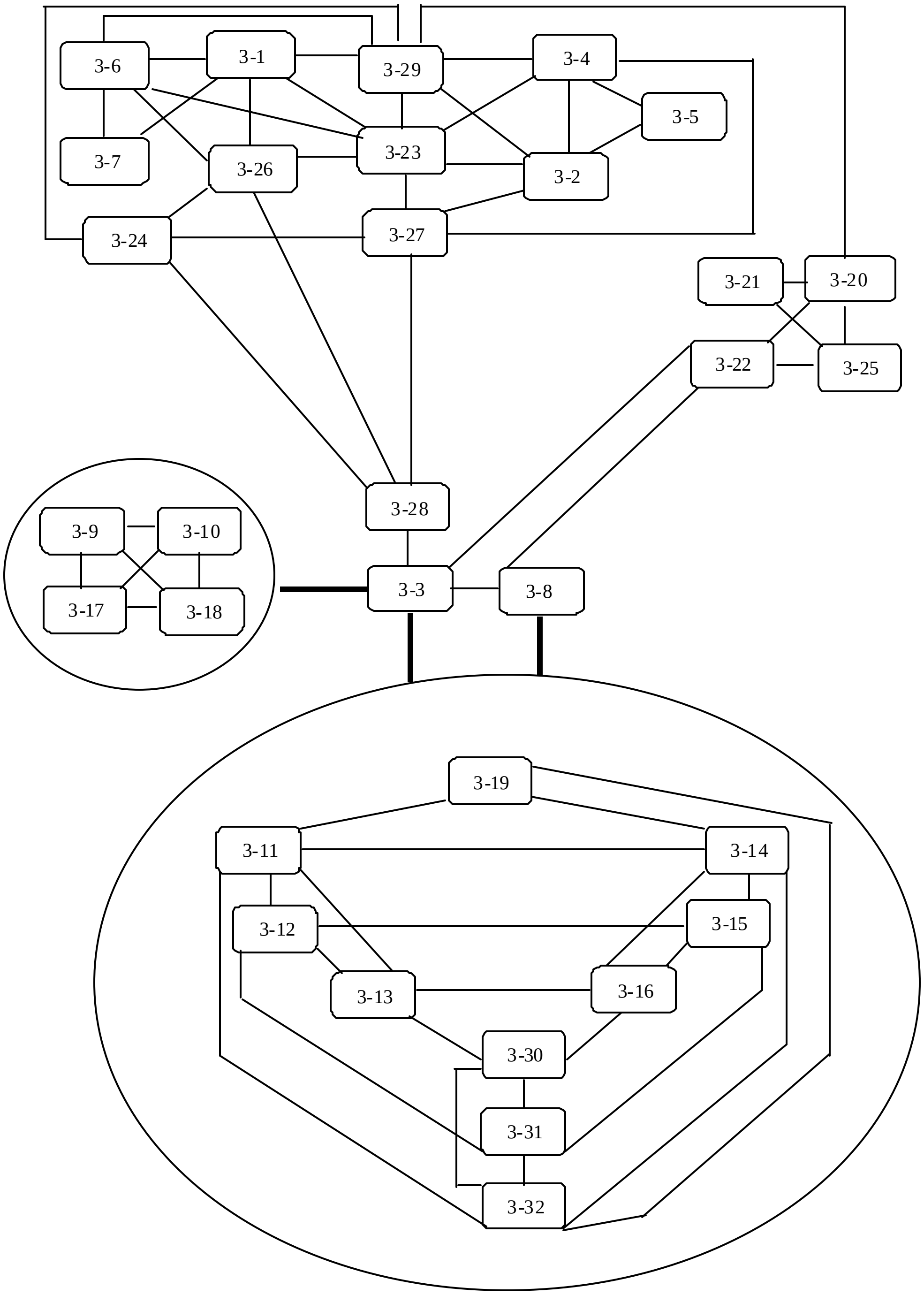


Figure 4.3 PFNET for Developers of CORPS $PFNET_{DEV}(r = \infty, 31)$

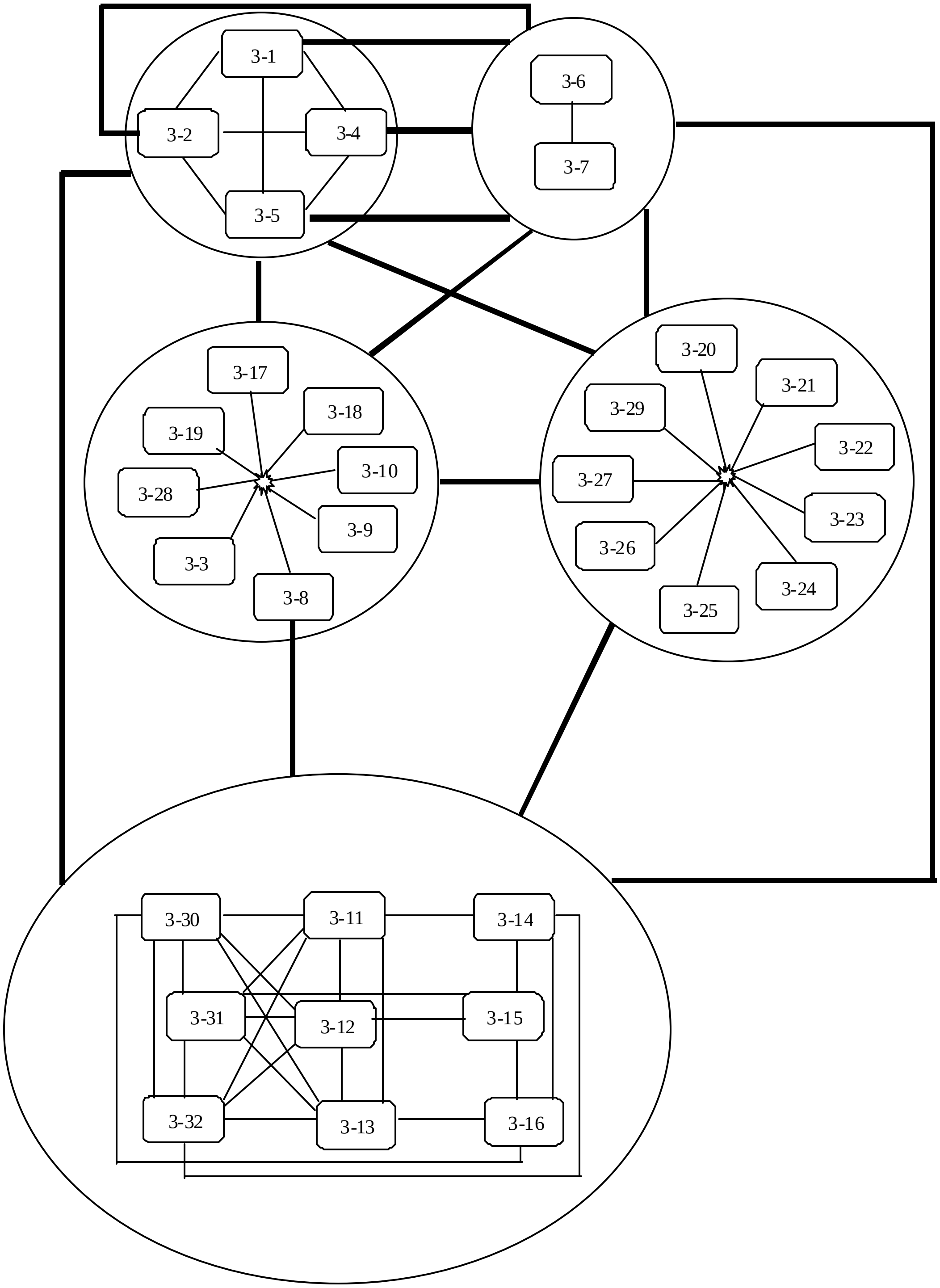


Figure 4.4 PFNET for the Customer of CORPS, $\text{PFNET}_{\text{CUS}}(r = \infty, q = 31)$

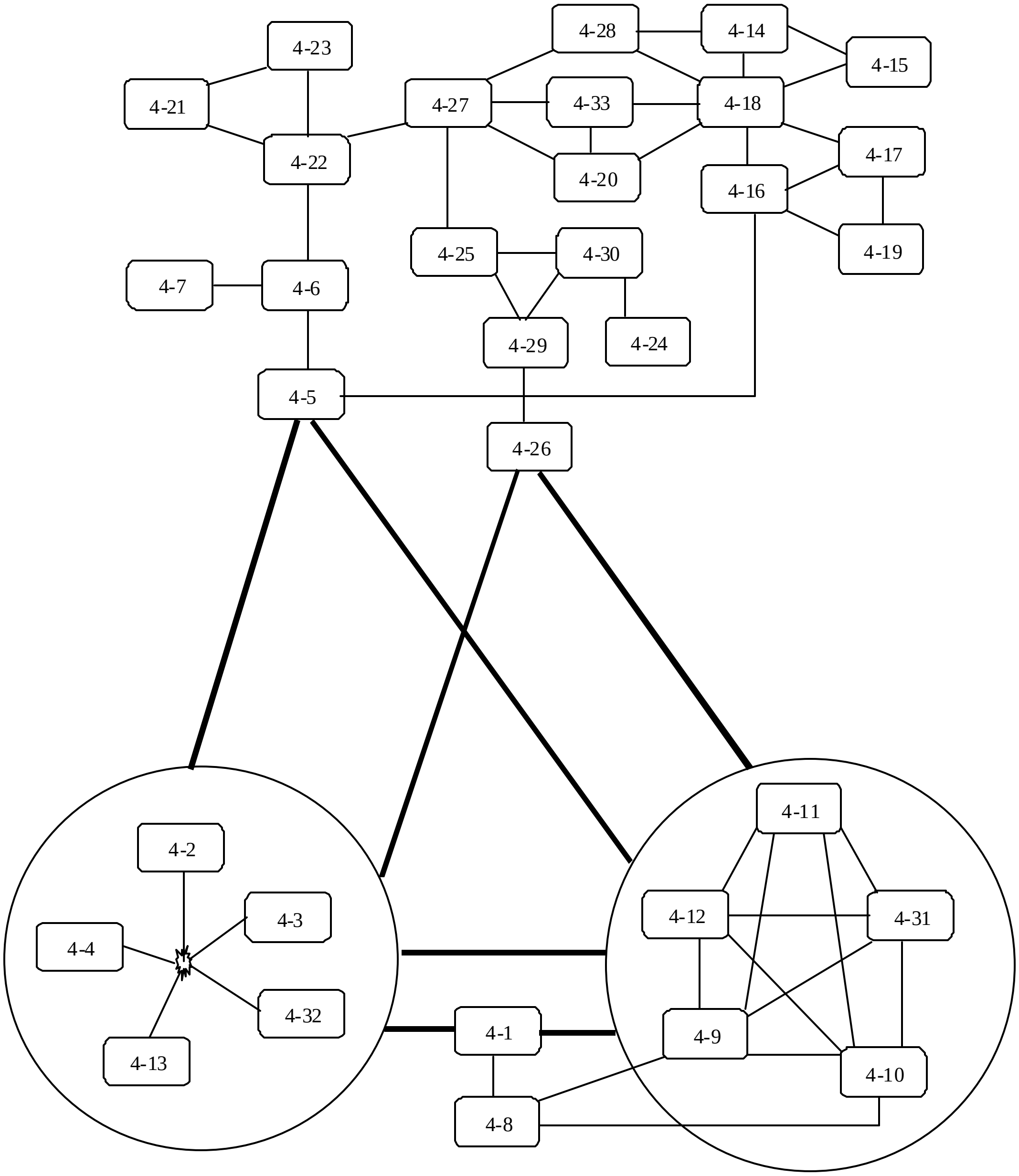


Figure 4.5 PFNET for DocTime Developers PFNET $_{DEV}$(r = ∞, q = 32)

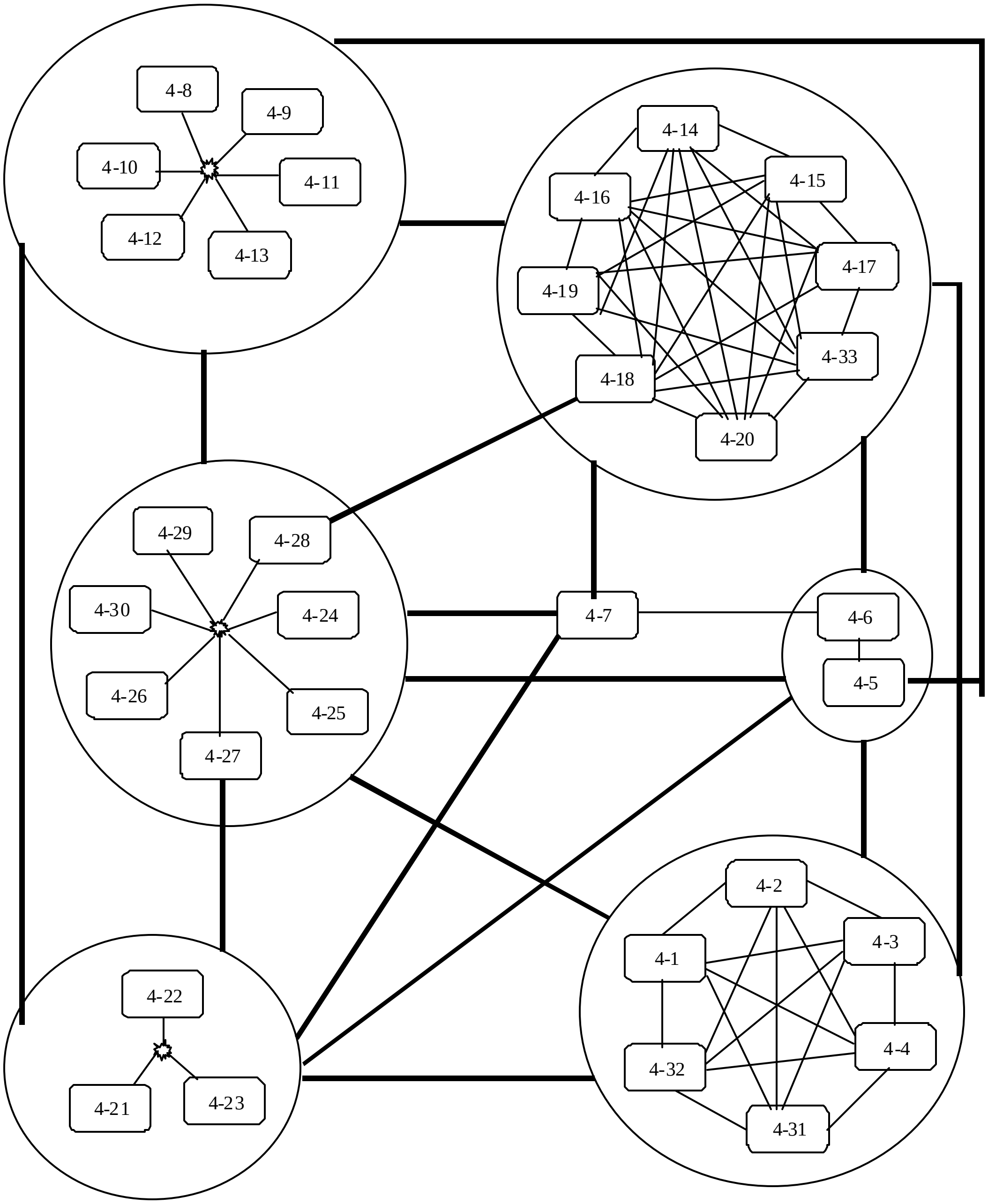


Figure 4.6 PFNET for DocTime Customer PFNET$_{CUS}$(r = ∞, q = 32)

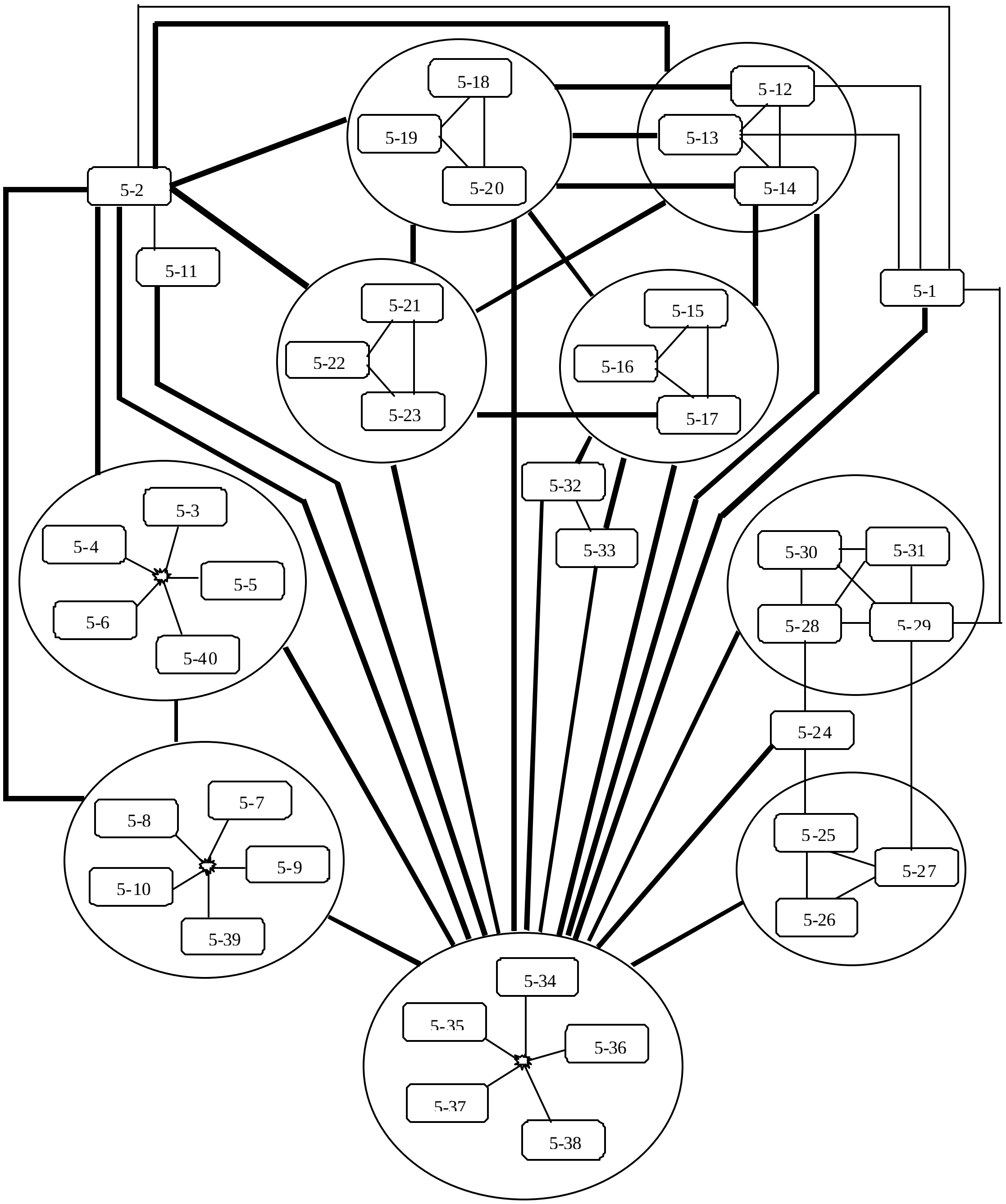


Figure 4.7   PFNET for CDRS Developers PFNET$_{DEV}$($r = \infty$, $q = 39$)

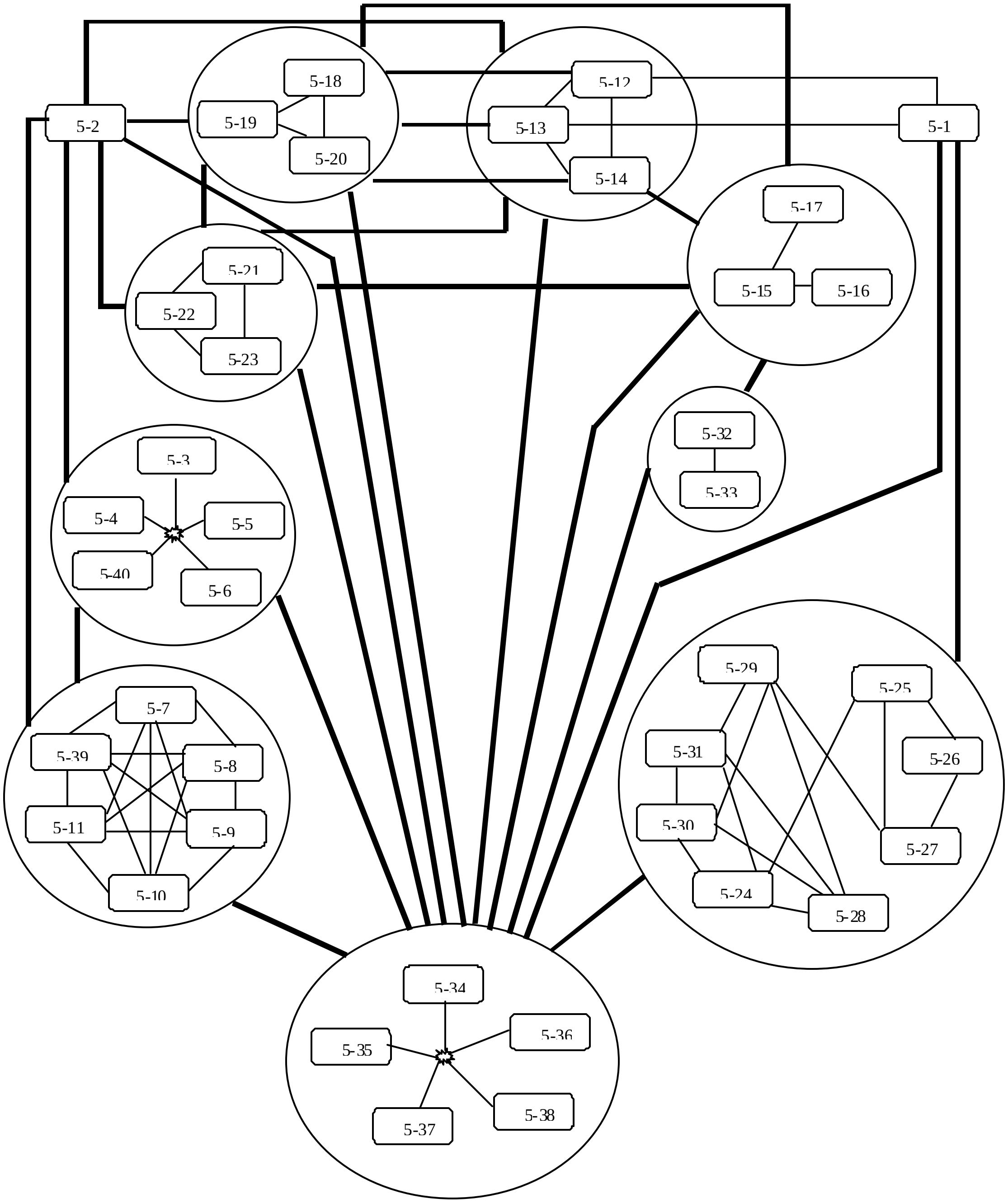


Figure 4.8 PFNET for CDRS Customer PFNET$_{CUS}$(r = ∞, q = 39)

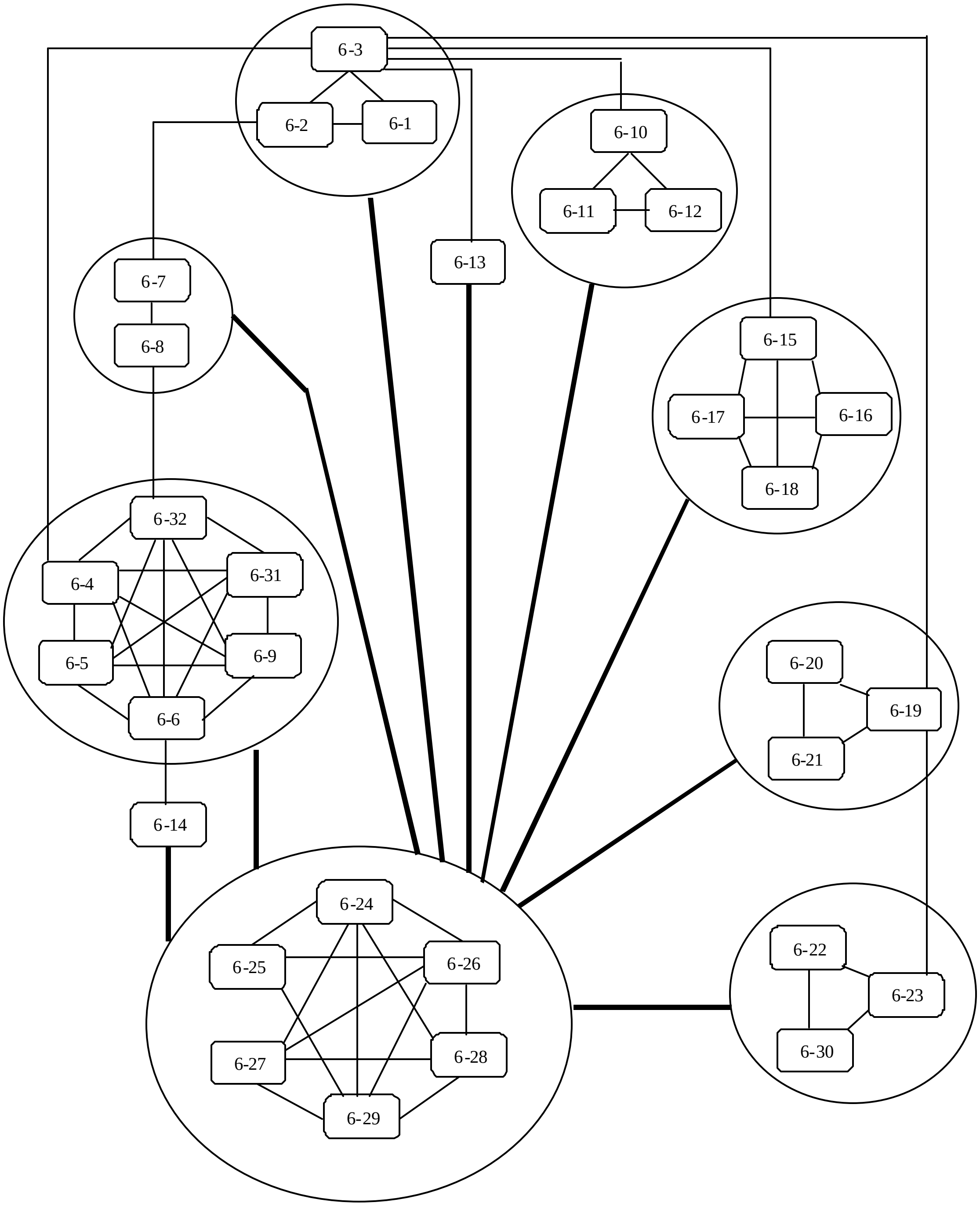


Figure 4.9 PFNET for the Developers of TSS $\mathrm{PFNET}_{\mathrm{DEV}}$ (r = ∞, q = 31)

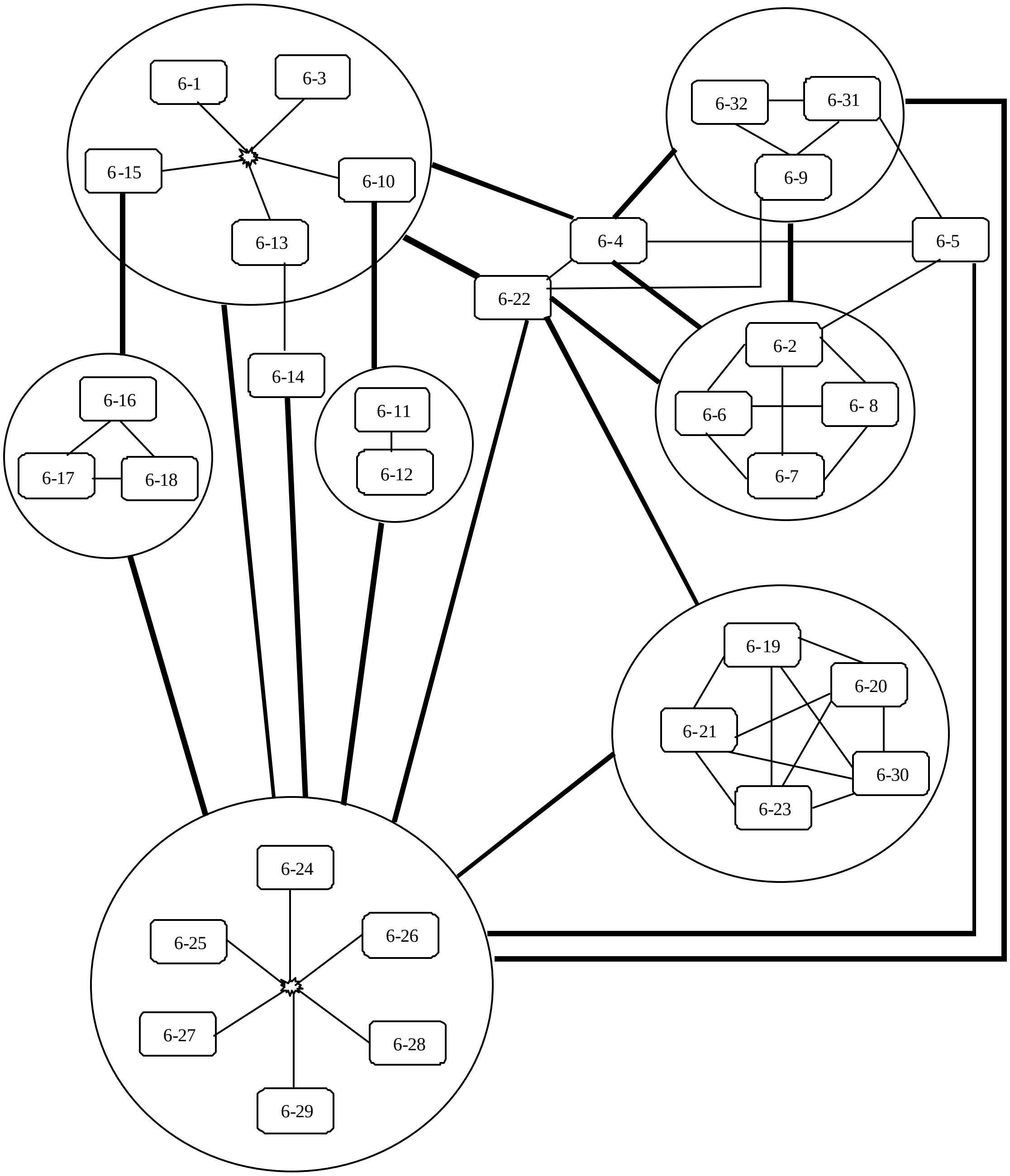


Figure 4.10 PFNET for the Customer of TSS PFNET$_{CUS}$ (r = ∞, q = 31)

The overall cc measures correlation between the corresponding path distances for all nodes between the two networks. Similarly, the cc for a single requirement can be computed. For all experiments, the following rules were applied (repeated here for ease of reading):

- A correlation coefficient of a network/node below 0.4 was assumed to indicate little or no similarity
- A correlation coefficient from 0.4 through 0.7 was assumed to indicate a moderate degree of similarity
- A correlation coefficient more than 0.7 was assumed to show very good to strong similarity.

The boundary values we assigned above to the correlation coefficients are subjective. They were selected based on evidence gathered in the classroom experiment and through additional empirical research. Table 4.1. shows the overall correlation coefficients between the developer and user Pathfinder networks for the four projects in our experiment. In Table 4.2, each row shows the percentage of requirements with different correlation coefficients between developer and user Pathfinder networks for each project.

Table 4.1 Correlation Coefficients Between Developer and User Pathfinder Networks

Software Systems	**Overall Correlation Coefficient (cc)**
CORPS	0.77
DocTime	0.46
CDRS	0.91
TSS	0.87

Table 4.2 Percentage of Requirements Within Specific Ranges of Correlation Coefficients

Correlation Coefficient (cc) **Percentage of Requirements (%)** **Systems**	**cc >=0.9**	**cc < 0.9 and cc > = 0.7**	**cc < 0.7**
CORPS	43.75	21.88	34.38
DocTime	0.00	0.00	100.00
CDRS	50.00	50.00	0.00
TSS	50.00	50.00	0.00

We then assume that the higher the value of the correlation coefficient, the more similar the mental models were. This may help software engineers measure the understanding of requirements between users and developers. In Table 4.2, the shaded row indicates that severe misunderstanding existed between the two groups in System 2. In fact, after delivery of this system, the user of System 2 was not at all satisfied with the final product, which seems to actually validate our thought that the tool can be a predictor. Analyzing the correlation coefficient for individual requirements may indicate how well a particular desired function is understood between the user and developer communities. A very low correlation coefficient may also indicate ambiguous requirements.

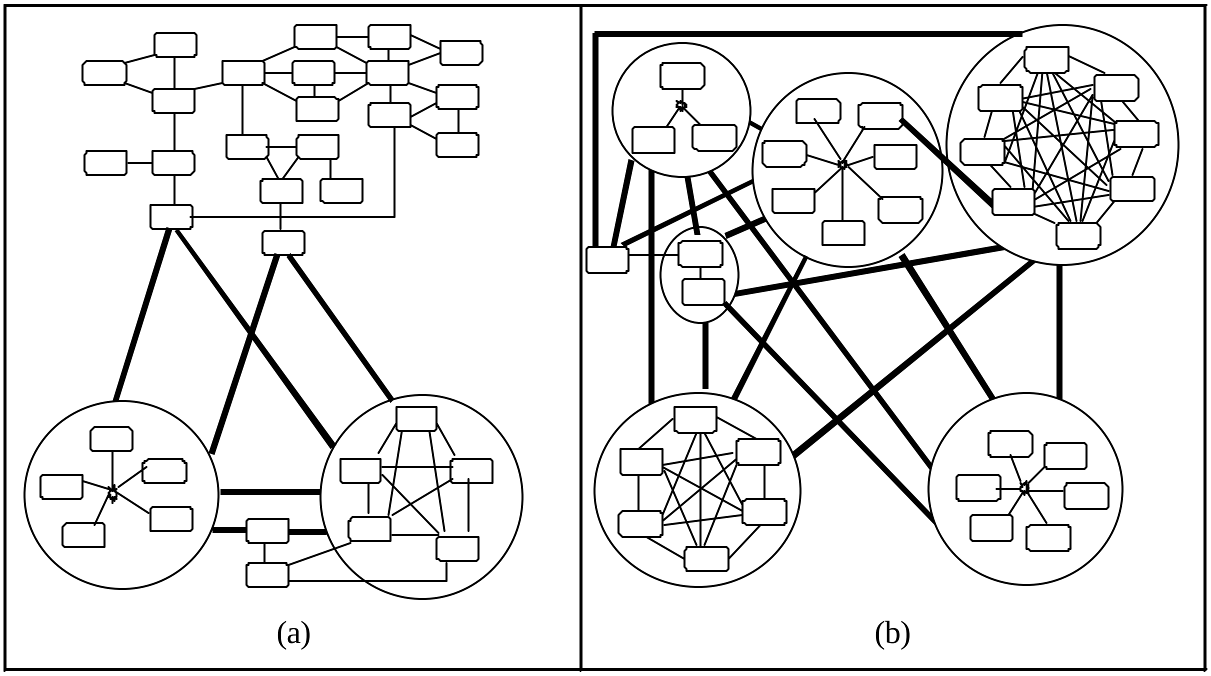


Figure 4.11 Graphical Pathfinder Networks - (a) Developers (b) User on System 2

The clustering information (also referred to as cliques) resulting from the Pathfinder networks, reveal patterns showing how the user and developer categorized the requirements. The bottom left circle in Figure 4.11 (a) and the bottom right circle in Figure 4.11 (b) are cliques. Generally, Pathfinder network clustering information shows similar requirements that achieve a particular system function. In our experiment, the SRS in each project was also seeded with duplicate requirements to determine if duplicate requirements could be identified through the use of this technique. Table 4.3 shows the correlation coefficient based on path distances for each of the original and seeded duplicate requirements in the developer Pathfinder networks.

Table 4.3 Correlation Coefficients for Original and Seeded Duplicate Requirements

Software Systems	**Original Requirement**	**Seeded Requireme nt**	**Correlation Coefficient**
System 1	1-13: Emailing a Student Resume to a Company	1-30: Sending a Student Resume to a Company by Electronic Mail	1.00
	1-16: Faxing a Student Resume to a Company	1-30: Sending a Student Resume to a Company by Electronic Mail	0.96
	1-12: Emailing a Student Transcript to a Company	1-31: Sending a Student Transcript to a Company by Electronic Mail	1.00
	1-15: Faxing a Student Transcript to a Company	1-31: Sending a Student Transcript to a Company by Electronic Mail	0.96
	1-11: Emailing a Letter Of Recommendation to a Company	1-32: Sending a Letter Of Recommendation to a Company by Electronic Mail	1.00
	1-14: Faxing A Letter Of Recommendation to a Company	1-32: Sending A Letter Of Recommendation to a Company by Electronic Mail	0.96
System 2	2-2: Add Appointment	2-31: Make Appointment	0.93
	2-3: Change Appointment	2-32: Modify Appointment	1.00
	2-13: Delete Appointment	2-32: Modify Appointment	1.00
	2-20: Modify User	2-33: Add And Delete User	1.00
System 3	3-8: Add a Major	3-39: Modify a Major	0.98
	3-9: Edit a Major	3-39: Modify a Major	1.00
	3-10: Delete a Major	3-39: Modify a Major	0.98
	3-4: Add a College	3-40: Modify a College	1.00
	3-5: Edit a College	3-40: Modify a College	1.00
	3-6: Delete a College	3-40: Modify a College	1.00
System 4	4-23: Display Available Vehicle	4-30: Check Available Vehicle	1.00
	4-5: Delete Reservation	4-31: Cancel Reservation	1.00
	4-9: Make Reservation	4-32: Add Reservation	0.99

The correlation coefficients of the original and seeded requirements were very high providing some evidence that the technique may be useful in finding such duplicates or at least in identifying suspect duplicates for further investigation. Similar requirements tend to be directly linked since they represent the shortest path distance. This makes it easier to identify duplicate requirements since only the neighborhood requirements (concepts) have to be compared in the Pathfinder network in order to check for duplicates instead of comparing every requirement with a high correlation coefficient. In Figure 4.12, the thick lines show how the duplicate requirements were connected in developers' Pathfinder network.

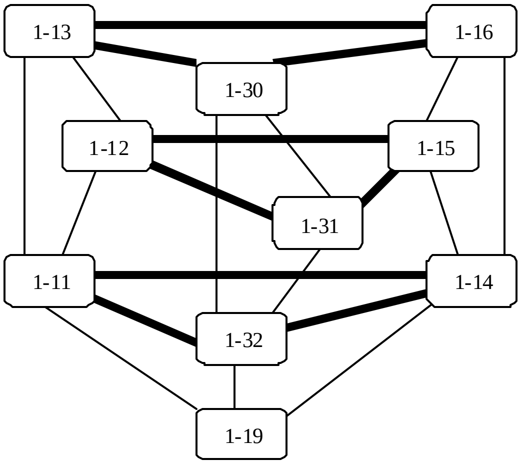


Figure 4.12 Part of the Pathfinder Network for System 1 Developers

It has to be noted that high values of correlation coefficients could potentially indicate the requirements are well understood. It is also possible that these requirements might indicate potential duplication. Duplicate requirements are only revealed after further analysis of the PFNETs.

A more recent study of a student development project conducted at Mississippi State University provided the following results regarding the predictive ability of Pathfinder networks on overall understanding of requirements and understanding of individual requirements between customers and developers of a software system. The system was developed as a class exercise and was called the Mississippi Science and Engineering Fair (MSEF) system. It had a total of twenty-one high-level categories at the end of the analysis phase and seventeen requirements after the implementation phase. Only sixteen of the original twenty-one requirements were unchanged from design through product delivery. The MSEF online system was built for a customer who intended to use this system to manage local high school science fair results.

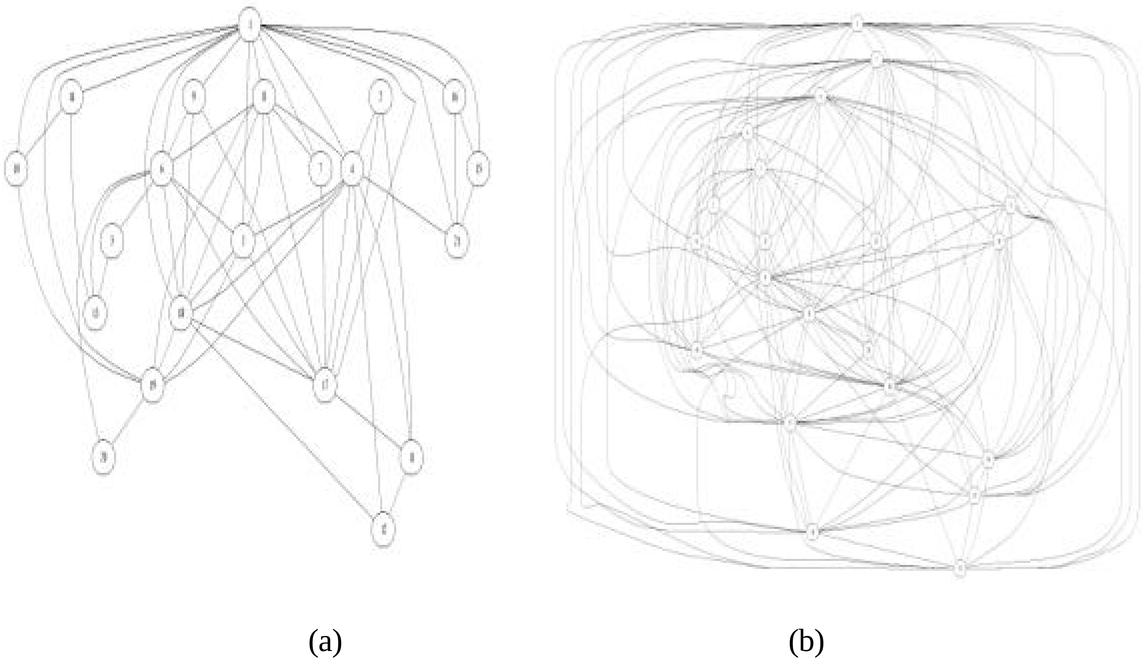
The mental models of the system (represented as graphs) for both the customer and developers were captured soon after the requirements phase. Developers and customers were interviewed after the delivery of the product to determine what they thought about the requirements that were initially listed in the SRS document. The developers never met again with the customer between the design phase and product delivery. Table 4.4 provides compiled results. Column one of Table 4.4 indicates the requirement number, column two of that table indicates the correlation coefficient between the customer and a consensus developer Pathfinder network. The last column of Table 4.4 indicates the reasons the developers thought were responsible for not fully/partially implementing the requirements.

Table 4.4 The Correlations and Feedback for MSEF

Req.	**Correlation (analysis phase)**	**Comments of developers (after implementation)**
1	0.87	
2	0.64	
3	0.73	
4	0.41	
5	0.60	
6	0.58	
7	0.51	Customer and time
8	0.74	
9	0.68	
10	0.58	Customer and time
11	0.62	Customer, technical and time
12	0.72	
13	0.72	
14	0.51	Customer and time
15	0.62	Customer and time
16	0.48	

The path correlations were computed based on the comparison of Pathfinder networks of customer and the consensus Pathfinder network of developers captured soon after the requirements analysis phase. The overall correlation between the customer's and developers' Pathfinder network was 0.48. As shown in Figure 4.13, this could have been noticed just by visual inspection of the resulting Pathfinder networks for the customer and developers. The low correlation might have contributed to the fact that three requirements that were not in the original SRS document submitted at the end of the requirement analysis phase, were added during design and implementation phase and that three of the original requirements were never implemented. During interviews after product delivery, the developers cited that they did not fully understand what the customer wanted. The last column of Table 4.4 indicates what the developers thought the reasons were for not being able to implement or partially implement the requirement. "Customer" means that the developers were not able to obtain a needed understanding from their customer who was seen as disengaged. "Time" means schedule pressures did not allow the developers to pursue additional understanding. "Technical" means that the developers had technical difficulties implementing the requirement.

The dark gray shade for a requirement in Table 4.4 indicates that the requirement was not implemented, and a light gray shade for a requirement indicates that the requirement was partially implemented. All of the requirements that were partially implemented or not implemented had correlation coefficients of less than 0.7. This number seems to be consistent with previous experiments reported in this paper. The correlation coefficients seem to point to the requirements that need to be re-examined by developers and customers because of possible misunderstanding early in the development process.



(a) (b)

Figure 4.13 (a) Consensus PFNET of 21 Nodes for Developers (b) PFNET of 21 Nodes for the Customer for the MSEF System

4.2 Results of Experimentation at NORTEL

A web-based tool was developed and applied to a small telecom project in Dallas, Texas at NORTEL by a student [68] working on his MS degree at MSU. The Pathfinder networks were generated for the customer and the developers for the Interactive Multimedia Server (IMS) software project – an internal NORTEL development. Figure 4.14 shows the PFNET that was generated using co-occurrence data from the customers for this project and Figure 4.15 shows the developers’ PFNET for the requirements of the same system. The analysis of the PFNETs and subsequent discussion between the various groups about the resulting links revealed that the networks were intuitive and represented what they understood about the requirements.

Table 4.5 The Correlation Coefficients between NORTEL Pathfinder Networks for the Customer and Developer for Individual Requirements [68]

Requirements.	Correlation coefficients
1. IMS application server	0.95
2. IMS proxy server	0.87
3. IMS redirect server	0.99
4. IMS registrar server	0.94
5. IMS location server	1.00
6. IMS external interface	0.05
7. IMS hardware platform	1.00
8. IMS database interface	0.79
9. IMS SIP interface	0.67
10. IMS H.323 client interface	0.73
11. IMS PSTN gateway interface	0.67
12. IMS media gateway	0.79
13. IMS media server	1.00
14. IMS multi-domain support	0.05
15. IMS performance and capacity	1.00
16. IMS BBUA component	0.75
17. IMS application server security	1.00
18. IMS discriminator service	0.98
19. IMS arbitrator service	1.00
20. IMS network handling	0.69
21. IMS call transfer service	0.99
22. IMS call conference service	1.00
23. IMS accounting	0.77

Comparison of customer and developer Pathfinder networks revealed that requirements “IMS external interface” and “IMS multi-domain support” both had very low correlation coefficients of 0.05 even though the overall correlation coefficient between the customer and developer networks was 0.88. The shaded rows in Table 4.5 show the correlation coefficients for the two requirements just mentioned. Note that the overall correlation coefficient was considered as very good to strong similarity. The Pathfinder networks were also able to pinpoint misunderstood requirements. Figure 4.14 and Figure 4.15 show what the customers and developers thought were ambiguous (shaded in gray).

The Pathfinder networks were helpful in identifying redundant requirements. The redundant requirements were directly connected which reduced the effort of searching for such requirements. In this case, the requirements “IMS arbitrator services” and “IMS call transfer services” were found to have similar descriptions and one of them was eventually declared redundant. These requirements are shown by dotted rectangles in Figure 4.14 and Figure 4.15. It can be observed that these two requirements are directly linked in both the customer and developer networks.

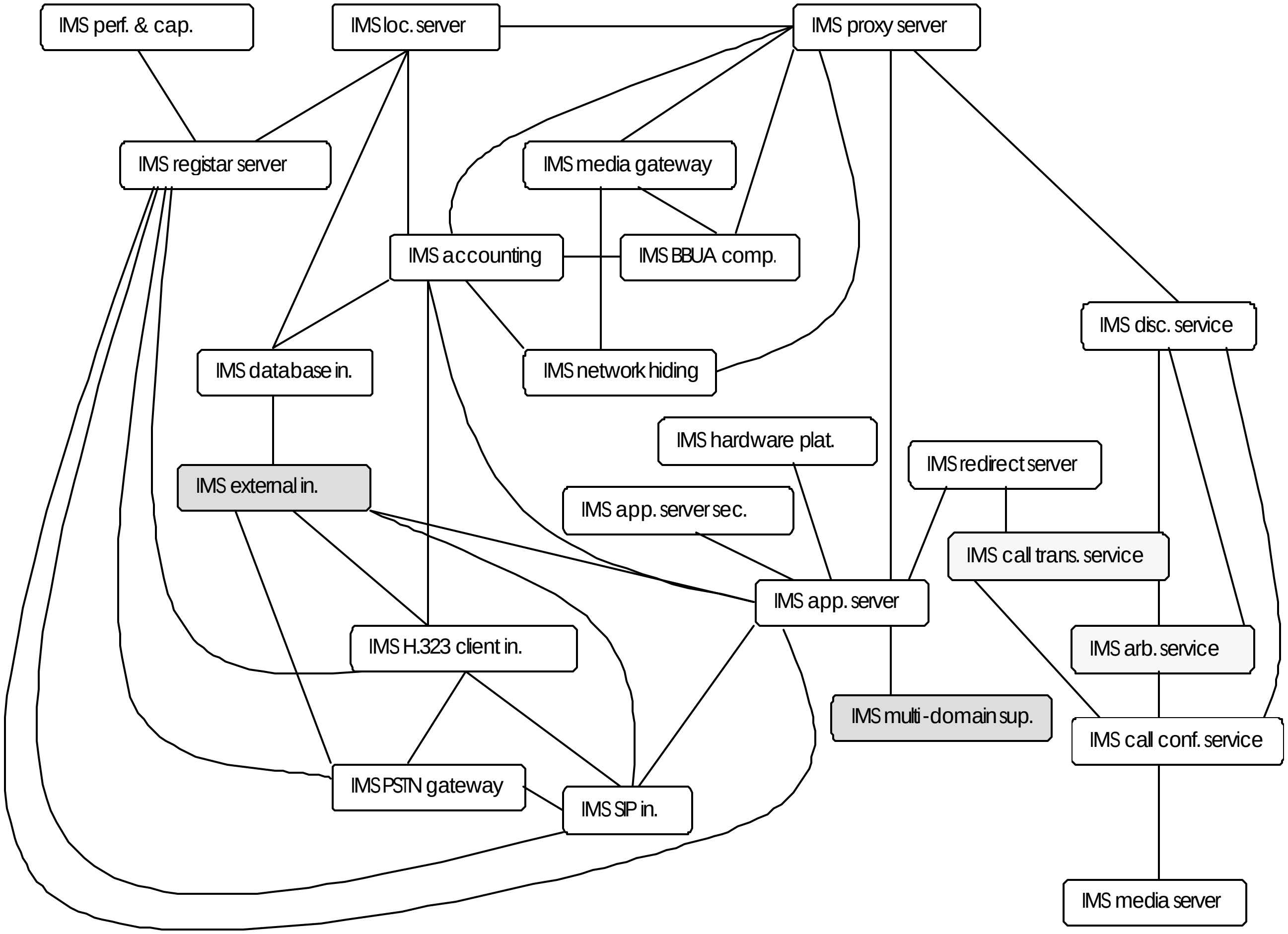


Figure 4.14 Pathfinder Network for Customer, PFNETcus (r = ∞, q = 22) for a 23-Node Network

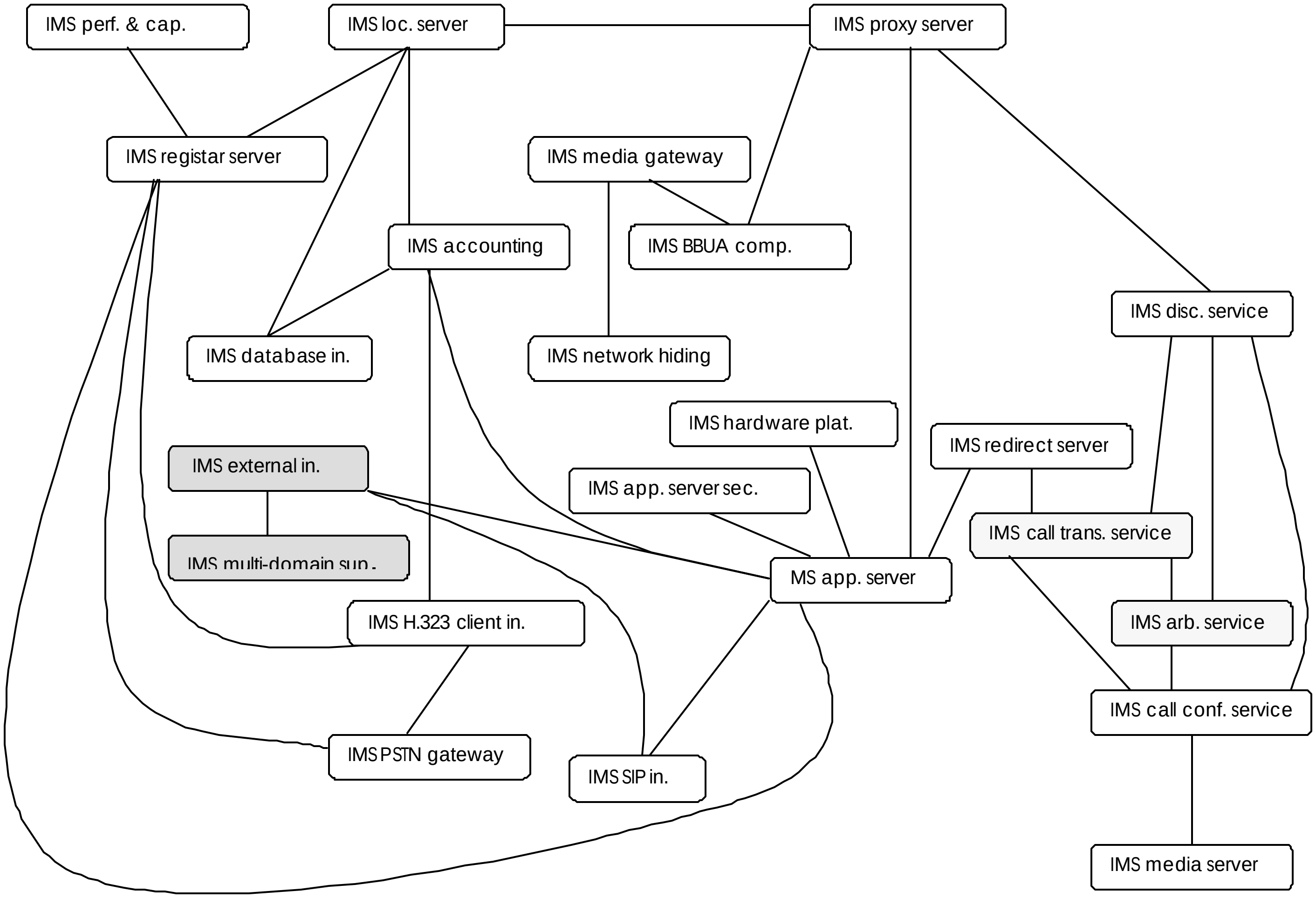


Figure 4.15 Pathfinder Network for Developer, PFNETdev (r = ∞, q = 22) for a 23-Node Network.

4.3 Results of Experimentation at AmerInd Inc.

4.3.1 Results and Analysis for PSC Project

The goal of the PSC project was to build a new website to organize information, provide intuitive and consistent navigation, and to provide speed and efficiency in accessing information for its members. The proposal document became the basis used to investigate misunderstandings between the stakeholders. It involved the project manager and a software developer. The first goal of the study was to improve the overall understanding about the proposal between the two stakeholders. The second goal was to check if the technique actually helped with convergence of the mental models after each facilitated session. The third goal was to learn how to scale the technique for the next project (i.e., NMA).

There were a total of twenty-seven requirements including the high level requirements agreed upon by both the members of the group. There were a total of two iterations of steps 1 through 6 of the methodology described in Section 0. Both the facilitation sessions were recorded on tape for further analysis.

Iteration 1: There were twenty low-level requirements and seven high-level categories (categories). All the low-level requirements were categorized into these high-level categories. Column 3 of Table 4.7 shows the correlations generated from the Pathfinder networks generated for each member of the group.

Table 4.6 shows the categorization information of requirements by the two stakeholders. The values in Table 4.6 indicate the number of members who categorized a requirement under a specific high-level category (column). A number "1" indicates that only one stakeholder categorized the requirement under that high-level category. A number "2" indicates that two members have categorized the same requirement under that high-level category. The shaded rows show potentially misunderstood requirements since they lack consensus among the stakeholders regarding how the requirement were categorized.

Table 4.6 Categorization of Requirements for PSC Project – First Iteration

Req ID	Requirement	S1	S2	S3	S4	S5	S6	S7
1	New Look & Feel	2				2		
2	Usability	1	1					
3	Navigation Scheme	1	2			2		
4	Data Migration	1	2		1			
5	Security	1			1		2	
6	Build Pages		2					
7	508 Compliance	1	2			1		2
8	Host Selection	2			1			
9	Installation			1	1			
10	Unit Testing		1					2
11	System Testing						1	2
12	Transfer of Operation				2			
13	Template Design - CSS		2			1		
14	Server Side Includes		2			1		
15	Usage Monitor			2				
16	Search Engine						2	
17	Forms - Simple		1				2	
18	Calendar		1				1	
19	Decouple from email			2				
20	Database Build & Integration			1			1	

For example, **Usability** was categorized into **S1** by one stakeholder and categorized into **S2** by another stakeholder in Table 4.6. This could potentially represent a misunderstanding among the stakeholders about the requirement **Usability**.

Figure 4.16 and Figure 4.17 show the graphs derived from the PFNETs of the customer and the developer by discarding the spurious links. These graphs are derived by discarding all the links with maximum edge weights in the PFNETs. This is because the PFNETs preserve all the salient relationships as lower edge weights when compared to the spurious links. The derived graphs reveal the most salient links and show clustering among the low-level requirements with respect to high-level categories.

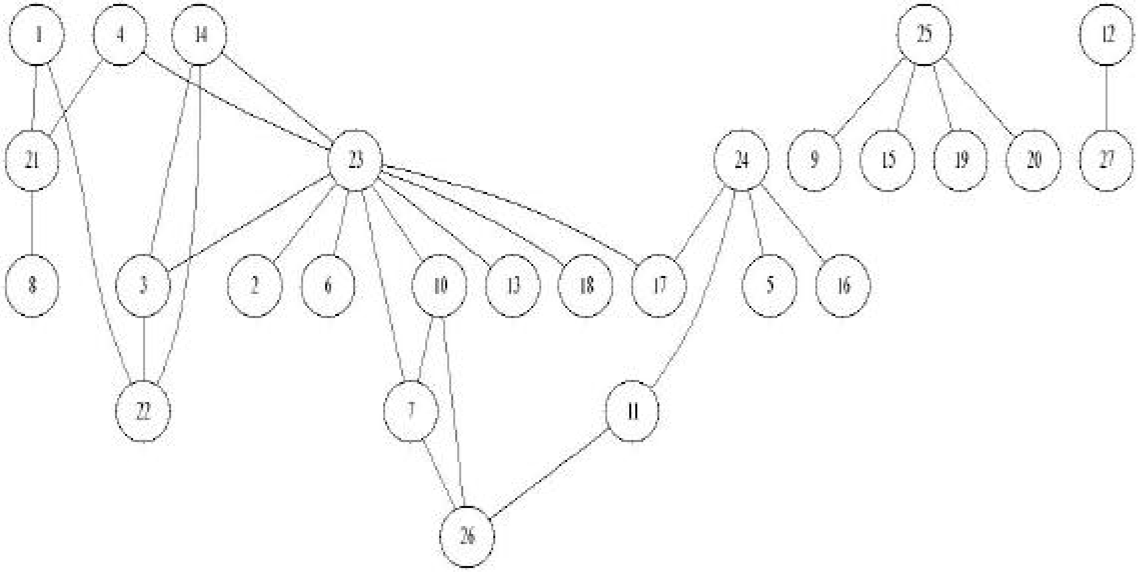


Figure 4.16 PSC – Graphs Derived from PFNET $_{\text{customer}}$ that Includes High-Level Requirements

Table 4.7 shows the correlations generated from the Pathfinder networks for each member of the group after the first iteration. During the facilitation sessions the requirements with the lower correlation coefficients were discussed first. The correlations helped focus the discussions on potentially misunderstood requirements. Table 4.8 shows the ratings given by the two stakeholders based on what they perceived about the requirements being discussed. Each stakeholder assigned the rating independently as soon as a requirement was discussed. For example a “1” was assigned by each stakeholder to “New Look & Feel” which indicated that, they understood the requirement very well (VWU). The correlation coefficient and the categorization seemed to reinforce that observation by the stakeholders.

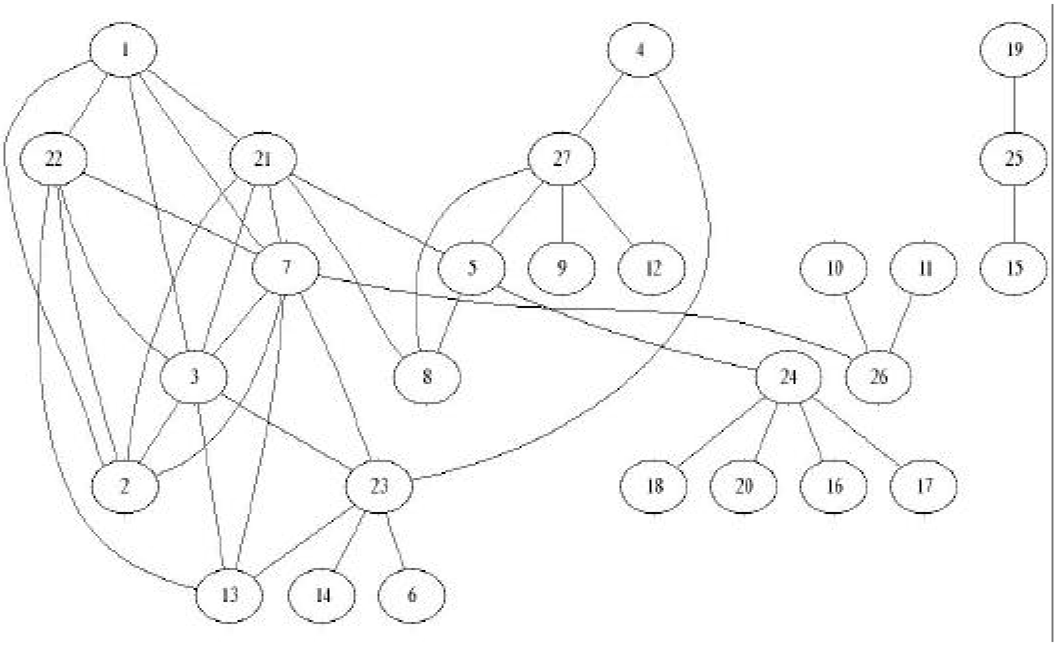
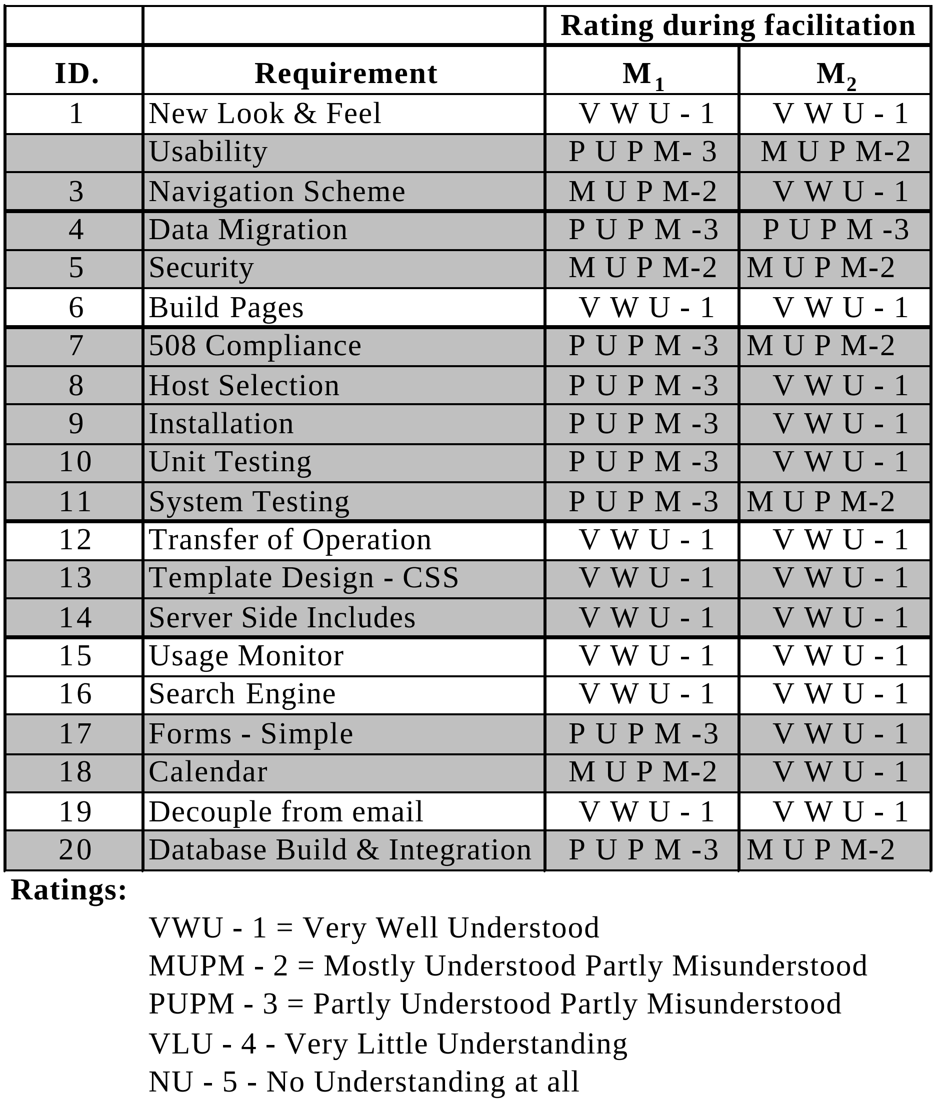


Figure 4.17 PSC – Graphs Derived from PFNET $_{\text{developer}}$ that Includes High-Level Requirements

Table 4.7 PSC – Correlation Coefficients for First Iteration

ID.	Requirement	Correlations
1	New Look & Feel	1.00
	Usability	0.00
3	Navigation Scheme	0.67
4	Data Migration	0.33
5	Security	0.33
6	Build Pages	1.00
7	508 Compliance	0.50
8	Host Selection	0.50
9	Installation	0.00
10	Unit Testing	0.50
11	System Testing	0.50
12	Transfer of Operation	1.00
13	Template Design - CSS	0.50
14	Server Side Includes	0.50
15	Usage Monitor	1.00
16	Search Engine	1.00
17	Forms - Simple	0.50
18	Calendar	0.00
19	Decouple from email	1.00
20	Database Build & Integratio	0.00
	Average	**0.54**
21	Research/ Prelims (S1)	0.29
22	Template Design (S5)	0.33
23	Implementation (S2)	0.60
24	Dynamic content (S6)	0.50
25	Backend (S3)	0.50
26	Test (S7)	1.00
27	Transfer (S4)	0.20
	Over all	**0.53**

Table 4.8 Rating of Stakeholders for PSC – First Iteration



		Rating during facilitation	
ID.	**Requirement**	**M_1**	**M_2**
1	New Look & Feel	V W U - 1	V W U - 1
	Usability	P U P M- 3	M U P M-2
3	Navigation Scheme	M U P M-2	V W U - 1
4	Data Migration	P U P M -3	P U P M -3
5	Security	M U P M-2	M U P M-2
6	Build Pages	V W U - 1	V W U - 1
7	508 Compliance	P U P M -3	M U P M-2
8	Host Selection	P U P M -3	V W U - 1
9	Installation	P U P M -3	V W U - 1
10	Unit Testing	P U P M -3	V W U - 1
11	System Testing	P U P M -3	M U P M-2
12	Transfer of Operation	V W U - 1	V W U - 1
13	Template Design - CSS	V W U - 1	V W U - 1
14	Server Side Includes	V W U - 1	V W U - 1
15	Usage Monitor	V W U - 1	V W U - 1
16	Search Engine	V W U - 1	V W U - 1
17	Forms - Simple	P U P M -3	V W U - 1
18	Calendar	M U P M-2	V W U - 1
19	Decouple from email	V W U - 1	V W U - 1
20	Database Build & Integration	P U P M -3	M U P M-2

Ratings:

VWU - 1 = Very Well Understood
MUPM - 2 = Mostly Understood Partly Misunderstood
PUPM - 3 = Partly Understood Partly Misunderstood
VLU - 4 - Very Little Understanding
NU - 5 - No Understanding at all

Iteration 2: Following iteration 1, the stakeholders were encouraged to categorize the same requirements within a few days following the first facilitated session. Table 4.9 and Table 4.10 summarizes the results from the Pathfinder networks generated for each member. The shaded rows are the potentially misunderstood requirements since they have low correlation coefficients. The shaded rows in Table 4.9 show the potentially misunderstood requirements. Rows represent low-level requirements and columns represent high-level categories. The values in each cell under the categories indicate how many members categorized that low-level requirement (row) into the high-level category (column). Table 4.9 shows that only three requirements (shaded) showed dissimilar categorization by the two stakeholders. This represented potential misunderstandings among the stakeholders.

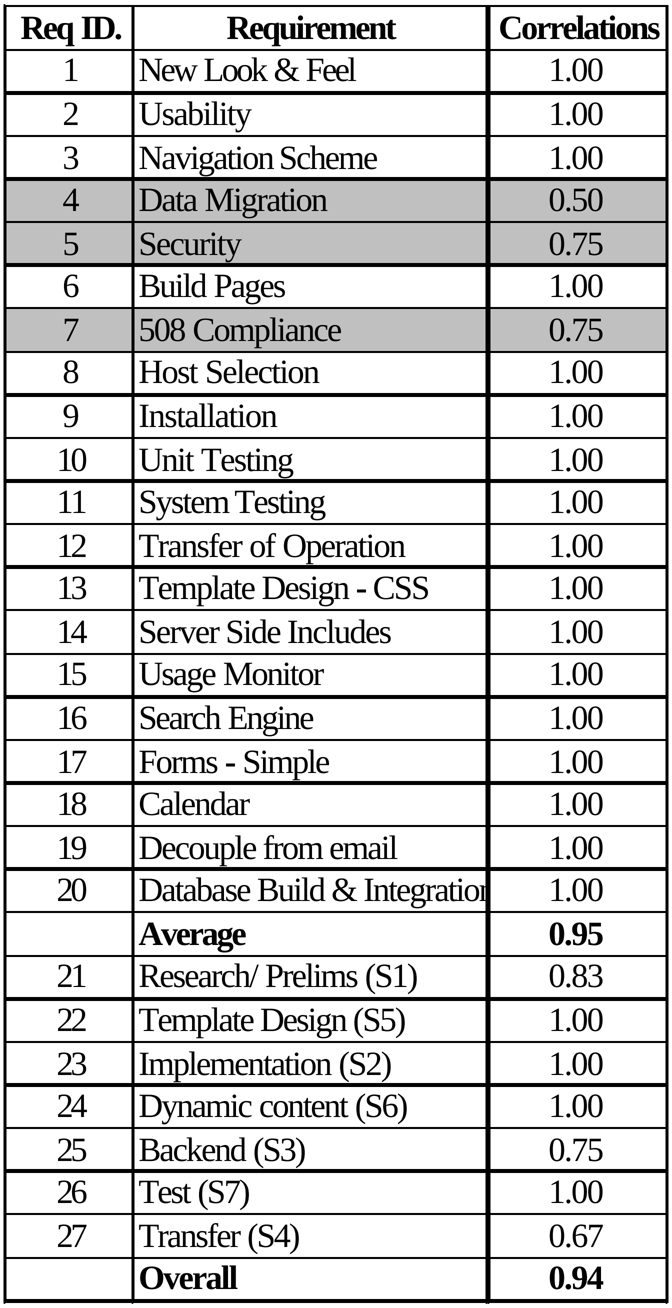
Table 4.9 Categorization of Requirements for PSC Project – Second Iteration

Req ID	Requirement	S1	S2	S3	S4	S5	S6	S7
1	New Look & Feel	2				2		
2	Usability	2				2		
3	Navigation Scheme	2	2			2		
4	Data Migration		2		1			
5	Security	2		1			2	2
6	Build Pages		2					
7	508 Compliance	1	2			2		2
8	Host Selection	2						
9	Installation			2	2			
10	Unit Testing		2				2	2
11	System Testing						2	2
12	Transfer of Operation				2			
13	Template Design - CSS		2			2		
14	Server Side Includes		2			2		
15	Usage Monitor			2				
16	Search Engine						2	
17	Forms - Simple						2	
18	Calendar		2				2	
19	Decouple from email			2				
20	Database Build & Integratio						2	

Table 4.10 shows the correlations generated from the Pathfinder networks for each member of the group after the first iteration. The shaded rows showed requirements that were thought to be misunderstood since they had low correlation coefficients.

During the second facilitation session, only three requirements were discussed based on correlation values.

Table 4.10 PSC - Correlation Coefficients for Second Iteration



Req ID.	Requirement	Correlations
1	New Look & Feel	1.00
2	Usability	1.00
3	Navigation Scheme	1.00
4	Data Migration	0.50
5	Security	0.75
6	Build Pages	1.00
7	508 Compliance	0.75
8	Host Selection	1.00
9	Installation	1.00
10	Unit Testing	1.00
11	System Testing	1.00
12	Transfer of Operation	1.00
13	Template Design - CSS	1.00
14	Server Side Includes	1.00
15	Usage Monitor	1.00
16	Search Engine	1.00
17	Forms - Simple	1.00
18	Calendar	1.00
19	Decouple from email	1.00
20	Database Build & Integratior	1.00
	Average	**0.95**
21	Research/ Prelims (S1)	0.83
22	Template Design (S5)	1.00
23	Implementation (S2)	1.00
24	Dynamic content (S6)	1.00
25	Backend (S3)	0.75
26	Test (S7)	1.00
27	Transfer (S4)	0.67
	Overall	**0.94**

The stakeholders' ratings during the facilitated session for the second iteration of the PSC project are shown in Table 4.11. The ratings assigned by the two stakeholders indicate how much they thought that requirement was misunderstood. Each stakeholder assigned the rating independently as soon as a requirement was discussed. For example a rating of "3" was assigned by stakeholder (M_1) to "Data Migration", and a rating of "4" was assigned by stakeholder (M_2). They had differing opinions about how well they thought that they understood the requirement after expressing the rationale for categorization.

Table 4.11 Rating of Stakeholders for PSC – Second Iteration

		Rating during facilitation	
Req ID.	**Requirement**	**M $_1$**	**M$_2$**
1	New Look & Feel	VWU - 1	VWU - 1
2	Usability	VWU - 1	VWU - 1
3	Navigation Scheme	VWU - 1	VWU - 1
4	Data Migration	PUPM-3	VLU - 4
5	Security	MUPM-2	MUPM-2
6	Build Pages	VWU - 1	VWU - 1
7	508 Compliance	MUPM-2	MUPM-2
8	Host Selection	VWU - 1	VWU - 1
9	Installation	VWU - 1	VWU - 1
10	Unit Testing	VWU - 1	VWU - 1
11	System Testing	VWU - 1	VWU - 1
12	Transfer of Operation	VWU - 1	VWU - 1
13	Template Design - CSS	VWU - 1	VWU - 1
14	Server Side Includes	VWU - 1	VWU - 1
15	Usage Monitor	VWU - 1	VWU - 1
16	Search Engine	VWU - 1	VWU - 1
17	Forms - Simple	VWU - 1	VWU - 1
18	Calendar	VWU - 1	VWU - 1
19	Decouple from email	VWU - 1	VWU - 1
20	Database Build & Integratio	VWU - 1	VWU - 1

Ratings:

VWU - 1 = Very Well Understood
MUPM - 2 = Mostly Understood Partly Misunderstood
PUPM - 3 = Partly Understood Partly Misunderstood
VLU - 4 - Very Little Understanding
NU - 5 - No Understanding at all

Figure 4.18 and Figure 4.19 illustrate the graphs from the second iteration. They are very similar except for a few links. Note that PFNET of the developer did not have any spurious links. The graph derived from the customer's PFNET was obtained after removing the spurious links. The correlations in Table 4.10 were derived from comparing PFNETs shown in Figure 4.18 and Figure 4.19.

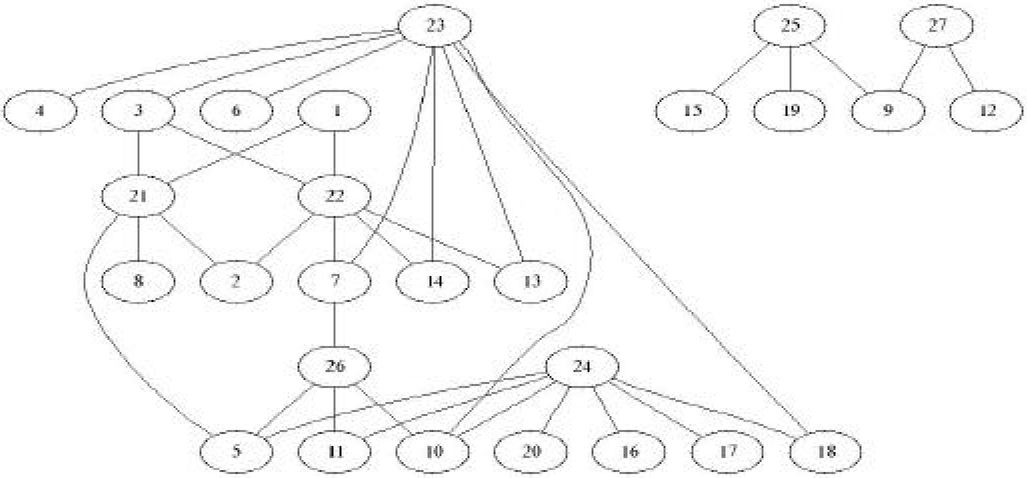


Figure 4.18 PSC – Graphs Derived from the Customer PFNET

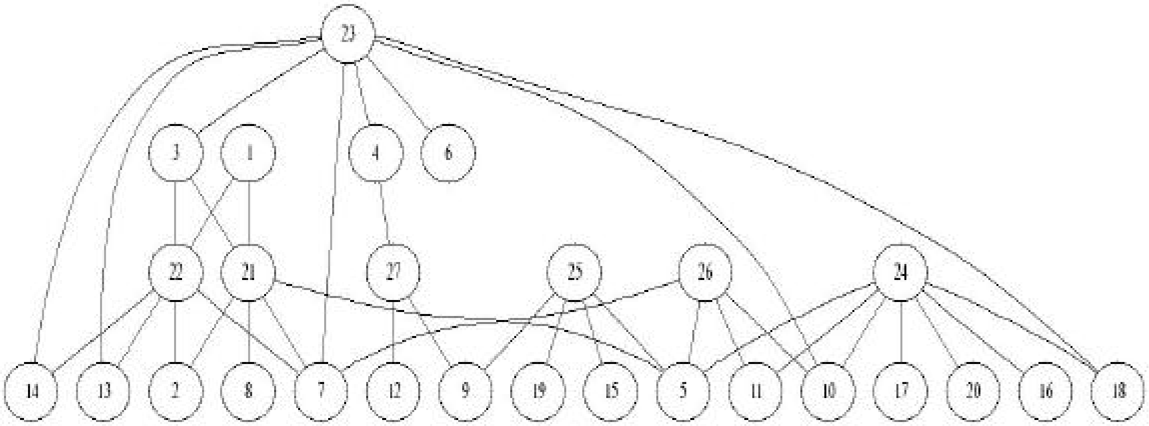


Figure 4.19 PSC- Graph Derived from the Developer PFNET

The Table 4.7 from the first iteration and the Table 4.10 from the second iteration show that the overall correlation coefficients improved from 0.54 to 0.95. This indicates that the discussions about potentially misunderstood requirements based on analysis of PFNETs during first iteration, contributed to the convergence of categorization, and hence assisted in improving the common understanding of requirements between the two stakeholders. There was common agreement between the members that the requirement *Data Migration* was an ambiguous requirement since there was not a proper description provided by the proposal. This was clearly predicted by low correlation coefficients for *Data Migration* in Table 4.7 and Table 4.10. The correlations for the low-level requirements predicted the understanding of those requirements. It is interesting to note that requirement *Data Migration* belongs to the high-level requirement *Transfer,* which had the lowest correlation coefficient among the seven high-level categories.

4.3.2 Results and Analysis for the NMA project

NMA was a medium-scale project with a total of one hundred and eight requirements listed in the requirements document. Each requirement had a requirement identification number, a brief description of the requirement and a detailed description of the requirement. Most requirements were already implemented and the remaining were implemented as the study progressed. Due to a Non-Disclosure agreement signed on this project, we cannot reveal the exact nature, descriptions of requirements or the customers for this project. NMA is a software system that allows customers to keep track of product information and their commercial supplier(s). This is particularly useful when a product is to be acquired with short notice. This system provides a quick cross-reference function to gather information in order to quickly acquire a product from commercial producer(s).

The implementation of this project was carried out at the customer's site in New Jersey. This project was coordinated between the customers, the developers and the headquarters located in Alexandria, VA. The project-manager (also a business case analyst), reports the progress on this project to the Vice President, Information Solutions Group at AmerInd Inc. A total of nineteen high-level (categories) requirements and eighty-six low-level requirements were identified in consultation with the project manager. The project manager oversees the implementation of the project and has an overall view of how the system should function. Any requirement which included "Parent" in its description was usually considered a high-level requirement. A total of six stakeholders were involved in this study. The stakeholders consisted of three system analysts, a coder, a project manager, and a Quality Control and Testing analyst.

Table 4.12 Categorization of Requirements for NMA Project

Req. ID	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
REQ4153	1				1			1	1				1	
REQ3307	1	1				1			1	1	1			
ARC3509		1				1		1	2	1				
REQ5027	2			1	1			1	1					
REQ4425						1			1				1	
REQ3966	1					1			2	1	1			
REQ4415	1							1	1	1	1			1
REQ4379							1	1	1				2	
REQ4287	1	4							1					
ARC3903	1						1		1	1				2
ARC3943	2								3					
REQ3881	1		1		2			2						
REQ4505				6										
ARC3488	1						2	2				1		
REQ4013				4	1		1							

Table 4.12 shows the categorization of the requirements by the stakeholders. The rows are requirements and columns are high-level categories. A value within each cell indicates the total number of stakeholders who categorized the low-level requirement (row) into a high-level category (column). For sake of simplicity, the categories are denoted by S1 through S14.

Table 4.13 Correlations Between All Possible Pairs of PFNETs

Req. ID.	cc_{12}	cc_{13}	cc_{14}	cc_{15}	cc_{16}	cc_{23}	cc_{24}	cc_{25}	cc_{26}	cc_{34}	cc_{35}	cc_{36}	cc_{45}	cc_{46}	cc_{56}	cc_{vavg}
REQ4153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
REQ3307	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
ARC3509	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.07
REQ5027	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.07
REQ4425	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
REQ3966	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0.07
REQ4415	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
REQ4379	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0.07
REQ4287	0	0	0	0	0	1	0	1	1	0	1	1	0	0	1	0.40
ARC3903	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0.07
ARC3943	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0.27
REQ3881	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0.13
ARC3488	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0.13
REQ4013	0	1	0	1	1	0	0	0	0	0	1	1	0	0	1	0.40
REQ4505	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.00

Table 4.14 shows just the average correlations for all the stakeholders for the NMA project derived from Table 4.13. The correlation values in Table 4.14 reinforce the categorization seen in Table 4.12. The lower the value of correlation coefficient the more the stakeholders seem to disagree on how they categorized that requirement.

For example the REQ4153 has the lowest correlation of 0.00 since there seems to be no agreement about the categorization among the stakeholders. REQ4505 has the highest value of correlation coefficient since all the stakeholders seem to agree on its categorization as seen in Table 4.12.

Table 4.14 Correlation Coefficients for NMA

Req ID.	Correlations
REQ4153	0.00
REQ3307	0.00
ARC3509	0.07
REQ5027	0.07
REQ4425	0.00
REQ3966	0.07
REQ4415	0.00
REQ4379	0.07
REQ4287	0.40
ARC3903	0.07
ARC3943	0.27
REQ3881	0.13
ARC3488	0.13
REQ4013	0.40
REQ4505	1.00

It should be noted that while discussing REQ3881, it was noticed that some members had categorized that requirement into S5 and some into S8. This is because S5 and S8 had similar meaning and hence were duplicate requirements, which was acknowledged during the facilitation session. There were some requirements that the stakeholders did not categorize into any of the high-level categories. They were allowed to create their own high-level categories if necessary. The low-level requirements that were not categorized into any of the predetermined categories are considered not related to any of the high-level categories. Requirements REQ4505, REQ4013, REQ4287 were put in the list even though they have higher correlation coefficients during the facilitation session as part of an effort to determine a threshold value of correlation coefficient and as an attempt to validate the correlations derived from PFNETs. Threshold correlation coefficients help to identify the requirements that are perceived to be well understood by the stakeholders. It would be possible to identify a threshold value for the correlations via ratings on what stakeholders thought about the requirement discussed. Comparing the correlation coefficients with the rating given by stakeholder, a threshold value of correlations may be identified. This is possible by analyzing the correlation coefficients that were assigned high ratings. All correlations coefficients that are greater than the 0.7 threshold value for a project need not be discussed during the facilitated session.

Table 4.15 shows the ratings collected from the stakeholders who participated in the facilitated session. M_1, M_2, M_3 and M_4 represent the four stakeholders who were present. An ‘x’ indicates that no rating was given. The purpose of collecting the ratings was to determine what the stakeholders actually thought after discussing a potentially misunderstood requirement and to see if a threshold value of correlation coefficient could be determined. The selected requirements were mostly based on low values of correlation coefficients.

The ratings for each requirement were based on the stakeholder's perception about how well they agreed with the categorization rational given by the rest of the stakeholders. A rating of '1' indicates that the stakeholder thought the rationale offered by the rest of the stakeholders was completely different from their own. A rating of '5' indicated that the stakeholder thought the rationale for categorization of that requirement was the same as their own.

Table 4.15 Stakeholders Ratings From the NMA Facilitated Session

Req ID.	**Ratings during Facilitation**			
	M1	**M2**	**M3**	**M4**
REQ4153	1	x	2	1
REQ3307	4	4	4	4
ARC3509	1	1	2	1
REQ5027	1	3	3	4
REQ4425	1	1	1	1
REQ3966	4	3	2	3
REQ4415	4	4	4	4
REQ4379	1	1	3	5
REQ4287	5	5	5	5
ARC3903	1	1	2	1
ARC3943	1	1	4	2
REQ3881	1	2	4	2
ARC3488	2	2	4	4
REQ4013	1	1	2	1
REQ4505	5	5	5	5

Rating scale

1- Req. has totally different interpretations compared to others
2- Req. has partially same interpretation but mostly differs compared to others
3- Req. has interpretations that seems to be equally split compared to others
4- Req. has mostly same interpretation but differs only a little compared to other
5- Req seems to have exact same interpretation as yours compared to others

Table 4.16 shows the sorted correlation coefficients in ascending order for all the requirements. Nodes in column one represent the node numbers in the PFNETs. The correlations were computed between every possible pair (fifteen) of members and then the average of those correlations was computed as shown in Table 4.13. If there were n members in both the groups then, there were a total of $n! / (n-2)! * 2!$ possible pairs of PFNETs.

Table 4.16 Average Correlation Coefficients (Sorted) for the NMA

Number	Code Number	CCavg.
15	REQ4153	0.00
19	REQ3307	0.00
41	REQ4425	0.00
45	REQ4415	0.00
4	REQ4602	0.07
20	ARC3509	0.07
23	REQ5027	0.07
30	REQ3998	0.07
42	REQ3966	0.07
44	REQ3645	0.07
46	REQ4379	0.07
55	ARC3903	0.07
78	ARC3829	0.07
79	REQ3395	0.07
16	REQ3881	0.13
22	ARC3820	0.13
35	ARC3488	0.13
47	ARC 3490	0.13
48	REQ4039	0.13
51	ARC2945	0.13
54	ARC3495	0.13
60	REQ4058	0.13
75	ARC3967	0.13
77	ARC2848	0.13
80	ARC4122	0.13
81	REQ4912	0.13
1	REQ5148	0.17
3	ARC3869	0.20
11	REQ4059	0.20
21	REQ4640	0.20
32	REQ5183	0.20
33	REQ3455	0.20
34	REQ4641	0.20
36	REQ4644	0.20
39	REQ3834	0.20
50	REQ4356	0.20
53	ARC3494	0.20
56	ARC3901	0.20
58	ARC3892	0.20
66	REQ4866	0.20
74	ARC3994	0.20
82	ARC3819	0.20
26	REQ3915	0.22
2	REQ4138	0.27
13	ARC3895	0.27
57	ARC3904	0.27
72	ARC3943	0.27
71	REQ4416	0.30
5	REQ4287	0.40
7	REQ4422	0.40
8	REQ4014	0.40
9	ARC3891	0.40
10	REQ4029	0.40
14	REQ3672	0.40
17	REQ4013	0.40
25	ARC3825	0.40
31	ARC4788	0.40
38	REQ4274	0.40
43	REQ4237	0.40
52	REQ4285	0.40
62	ARC3890	0.40
64	REQ4193	0.40
65	ARC3289	0.40
67	ARC3493	0.40
69	ARC3510	0.40
70	ARC3900	0.40
73	ARC3535	0.40
83	REQ5186	0.40
84	ARC3991	0.40
85	ARC4865	0.40
18	ARC3482	0.47
24	ARC3913	0.47
28	REQ4484	0.47
29	REQ4814	0.47
49	ARC3264	0.47
59	ARC3870	0.47
63	ARC3894	0.47
6	REQ4643	0.67
27	ARC4121	0.67
37	REQ3997	0.67
40	ARC3965	0.67
61	ARC3902	0.67
68	ARC3811	0.67
76	REQ3992	0.67
12	REQ4505	1.00

For the purpose of illustration, the consensus PFNETs for the two groups are shown in Figure 4.20 and Figure 4.21. The node numbers in consensus PFNETs are consistent with the node numbers in Table 4.16.

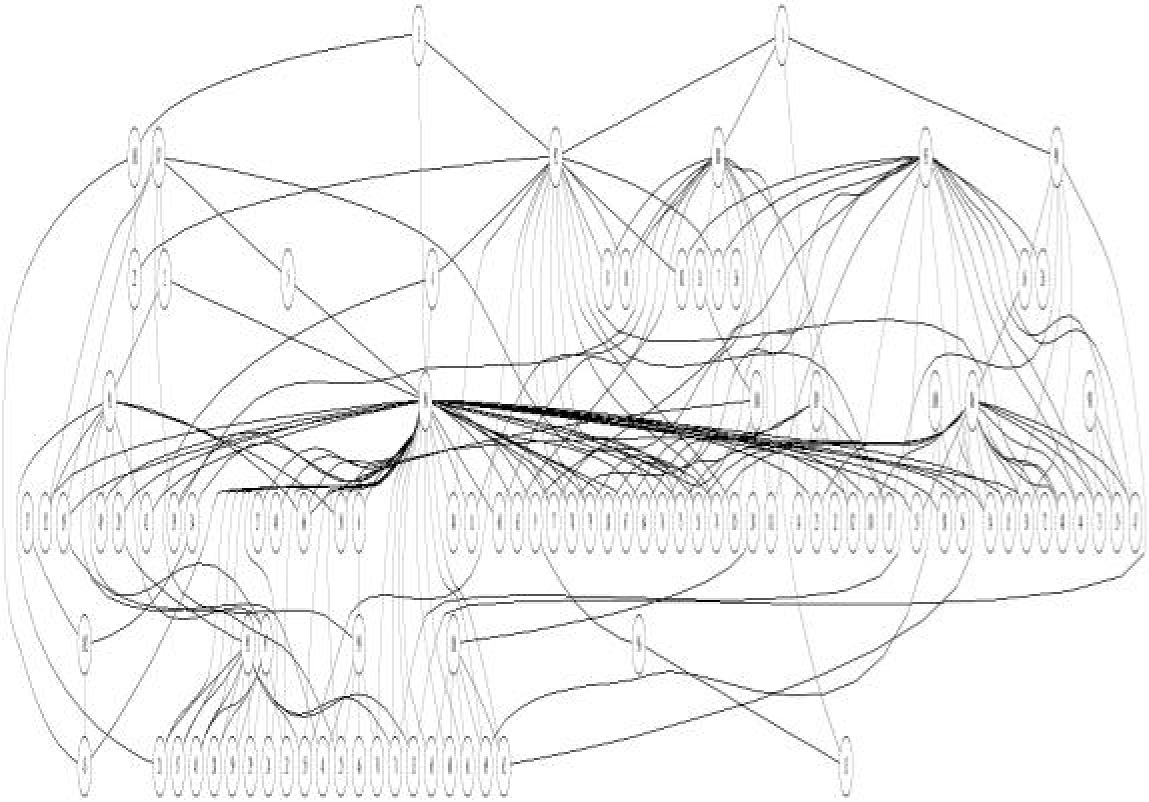


Figure 4.20 NMA: Graph Derived from Consensus PFNET$_{groupI}$

The two groups were divided on a subjective basis based on their experience and nature of their work on the project. This was done in consultation with the project manager. The first group consisted of designers and programmers of the system and second group consisted of users who would verify the system functionality. It is possible to compare the PFNETs of groups to determine their misunderstandings about requirements.

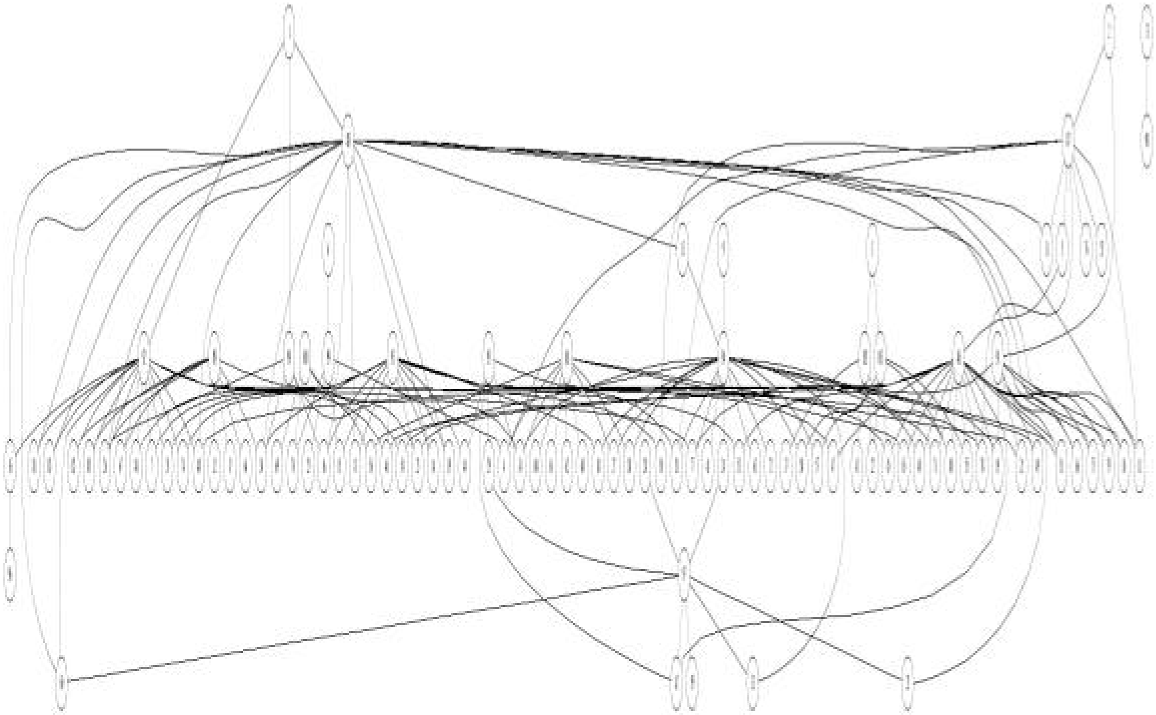


Figure 4.21 NMA: Graph Derived from Consensus PFNET $_{groupII}$

The next section presents the results obtained by applying information theory-based software metrics on two projects. This effort was aimed at experimenting with other metrics to measure consensus about requirements among stakeholders in order to reduce misunderstandings.

4.4 Results and Discussion for Information Theory-Based Metrics

This section describes the application of the complexity metric to measure consensus among the stakeholders for the data collected at AmerInd Inc. It should be noted that the analysis was done at MSU. These metrics were not used to analyze or facilitate sessions at AmerInd Inc.

4.4.1 Results and Analysis for the PSC Project

The raw data for applying the information theory-based complexity metric was based on the two iterations of the PSC project. Each iteration of the PSC project generated the categorization information submitted by each stakeholder. There were a total of two stakeholders in this project, referred to as M_1 and M_2. The seven pre-determined high-level categories are represented as S1 through S7. Categorization tables and consensus categorization tables (see Section 3.5) were generated for both the iterations of the PSC project. Table 4.17 and Table 4.18 show the individual categorization information collected from each stakeholder during the first iteration of the PSC project.

Table 4.17 Categorization Table for Stakeholder M_2 – First Iteration

		Category						
	Requirements	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
1	New Look & Feel	1	0	0	0	1	0	0
2	Usability	0	1	0	0	0	0	0
3	Navigation Scheme	0	1	0	0	1	0	0
4	Data Migration	1	1	0	0	0	0	0
5	Security	0	0	0	0	0	1	0
6	Build Pages	0	1	0	0	0	0	0
7	508 Compliance	0	1	0	0	0	0	1
8	Host Selection	1	0	0	0	0	0	0
9	Installation	0	0	1	0	0	0	0
10	Unit Testing	0	1	0	0	0	0	1
11	System Testing	0	0	0	0	0	1	1
12	Transfer of Operation	0	0	0	1	0	0	0
13	Template Design - CSS	0	1	0	0	0	0	0
14	Server Side Includes	0	1	0	0	1	0	0
15	Usage Monitor	0	0	1	0	0	0	0
16	Search Engine	0	0	0	0	0	1	0
17	Forms - Simple	0	1	0	0	0	1	0
18	Calendar	0	1	0	0	0	0	0
19	Decouple from email	0	0	1	0	0	0	0
20	Database Build & Integration	0	0	1	0	0	0	0

Table 4.18 Categorization Table for Stakeholder M_1 – First Iteration

		Category						
	Requirements	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
1	New Look & Feel	1	0	0	0	1	0	0
2	Usability	1	0	0	0	1	0	0
3	Navigation Scheme	1	1	0	0	1	0	0
4	Data Migration	0	1	0	1	0	0	0
5	Security	1	0	0	1	0	1	0
6	Build Pages	0	1	0	0	0	0	0
7	508 Compliance	1	1	0	0	1	0	1
8	Host Selection	1	0	0	1	0	0	0
9	Installation	0	0	0	1	0	0	0
10	Unit Testing	0	0	0	0	0	0	1
11	System Testing	0	0	0	0	0	0	1
12	Transfer of Operation	0	0	0	1	0	0	0
13	Template Design - CSS	0	1	0	0	1	0	0
14	Server Side Includes	0	1	0	0	0	0	0
15	Usage Monitor	0	0	1	0	0	0	0
16	Search Engine	0	0	0	0	0	1	0
17	Forms - Simple	0	0	0	0	0	1	0
18	Calendar	0	0	0	0	0	1	0
19	Decouple from email	0	0	1	0	0	0	0
20	Database Build & Integration	0	0	0	0	0	1	0

Table 4.19 shows the consensus categorization table. The consensus categorization table shown in Table 4.19 is constructed by concatenating the row patterns for each requirement from Table 4.17 and Table 4.18. The concatenation of categorization of row patterns of stakeholder M_1 and M_2 is denoted as $M_1|M_2$. All of the above tables are for the first iteration of the PSC project.

Table 4.19 Categorization Consensus Table for PSC ($M_1|M_2$) – First Iteration

	Category													
				M_1							**M_2**			
Requirements	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
1 New Look & Feel	1	0	0	0	1	0	0	1	0	0	0	1	0	0
2 Usability	1	0	0	0	1	0	0	0	1	0	0	0	0	0
3 Navigation Scheme	1	1	0	0	1	0	0	0	1	0	0	1	0	0
4 Data Migration	0	1	0	1	0	0	0	1	1	0	0	0	0	0
5 Security	1	0	0	1	0	1	0	0	0	0	0	0	1	0
6 Build Pages	0	1	0	0	0	0	0	0	1	0	0	0	0	0
7 508 Compliance	1	1	0	0	1	0	1	0	1	0	0	0	0	1
8 Host Selection	1	0	0	1	0	0	0	1	0	0	0	0	0	0
9 Installation	0	0	0	1	0	0	0	0	0	1	0	0	0	0
10 Unit Testing	0	0	0	0	0	0	1	0	1	0	0	0	0	1
11 System Testing	0	0	0	0	0	0	1	0	0	0	0	0	1	1
12 Transfer of Operation	0	0	0	1	0	0	0	0	0	0	1	0	0	0
13 Template Design - CSS	0	1	0	0	1	0	0	0	1	0	0	0	0	0
14 Server Side Includes	0	1	0	0	0	0	0	0	1	0	0	1	0	0
15 Usage Monitor	0	0	1	0	0	0	0	0	0	1	0	0	0	0
16 Search Engine	0	0	0	0	0	1	0	0	0	0	0	0	1	0
17 Forms - Simple	0	0	0	0	0	1	0	0	1	0	0	0	1	0
18 Calendar	0	0	0	0	0	1	0	0	1	0	0	0	0	0
19 Decouple from email	0	0	1	0	0	0	0	0	0	1	0	0	0	0
20 Database Build & Integration	0	0	0	0	0	1	0	0	0	1	0	0	0	0

The information complexity metric is applied to individual categorization information and to the consensus categorization information to generate values for the entire system and for individual requirements. Table 4.20 shows the complexity measurement values generated for the entire system and for each requirement during the first iteration.

Table 4.20 Complexity Measurements for PSC First Iteration (Bits)

		M_1	**M_2**	**$M_1 \| M_2$**
Entire System		402.28	362.46	669.27
Requirements				
1	New Look & Feel	24.96	19.45	42.02
2	Usability	24.96	17.64	40.19
3	Navigation Scheme	38.10	28.56	53.70
4	Data Migration	29.25	27.57	46.16
5	Security	41.70	11.05	49.86
6	Build Pages	14.25	17.64	25.35
7	508 Compliance	47.34	28.56	57.86
8	Host Selection	29.26	7.81	37.26
9	Installation	12.84	12.05	26.29
10	Unit Testing	8.81	28.56	27.56
11	System Testing	8.81	22.00	22.00
12	Transfer of Operation	12.84	1.34	15.45
13	Template Design - CSS	27.54	17.64	39.29
14	Server Side Includes	14.25	28.56	35.56
15	Usage Monitor	5.99	12.05	15.05
16	Search Engine	13.84	11.05	22.00
17	Forms - Simple	13.84	29.17	40.61
18	Calendar	13.84	17.64	31.70
19	Decouple from email	5.99	12.05	15.05
20	Database Build & Integration	13.84	12.05	26.29

The consensus between the two stakeholders as computed by Equation (3.4) for the entire system is 669.27 – max (402.28, 362.46) = 266.99 bits. The positive value of the information complexity measurement metric indicates that there is lack of consensus among the stakeholders. Stakeholder M_2 has the lowest value of complexity measurement for requirement *12. Transfer of Operation* (1.34 bits). This particular requirement has the lowest information complexity because it is the only requirement belonging to its category (see Table 4.17).

Table 4.21 Categorization Table for Stakeholder M_1 – Second Iteration

	Category						
Requirements	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
1 New Look & Feel	1	0	0	0	1	0	0
2 Usability	1	0	0	0	1	0	0
3 Navigation Scheme	1	1	0	0	1	0	0
4 Data Migration	0	1	0	0	0	0	0
5 Security	1	0	0	0	0	1	1
6 Build Pages	0	1	0	0	0	0	0
7 508 Compliance	0	1	0	0	1	0	1
8 Host Selection	1	0	0	0	0	0	0
9 Installation	0	0	1	1	0	0	0
10 Unit Testing	0	1	0	0	0	1	1
11 System Testing	0	0	0	0	0	1	1
12 Transfer of Operation	0	0	0	1	0	0	0
13 Template Design - CSS	0	1	0	0	1	0	0
14 Server Side Includes	0	1	0	0	1	0	0
15 Usage Monitor	0	0	1	0	0	0	0
16 Search Engine	0	0	0	0	0	1	0
17 Forms - Simple	0	0	0	0	0	1	0
18 Calendar	0	1	0	0	0	1	0
19 Decouple from email	0	0	1	0	0	0	0
20 Database Build & Integration	0	0	0	0	0	1	0

Table 4.21 and Table 4.22 show the individual categorization information collected from each stakeholder during the second iteration of the PSC project. Table 4.23 shows the consensus categorization table for the second iteration of the PSC project.

Table 4.22 Categorization Table for Stakeholder M_2 – Second Iteration

		Category						
	Requirements	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
1	New Look & Feel	1	0	0	0	1	0	0
2	Usability	1	0	0	0	1	0	0
3	Navigation Scheme	1	1	0	0	1	0	0
4	Data Migration	0	1	0	1	0	0	0
5	Security	1	0	1	0	0	1	1
6	Build Pages	0	1	0	0	0	0	0
7	508 Compliance	1	1	0	0	1	0	1
8	Host Selection	1	0	0	0	0	0	0
9	Installation	0	0	1	1	0	0	0
10	Unit Testing	0	1	0	0	0	1	1
11	System Testing	0	0	0	0	0	1	1
12	Transfer of Operation	0	0	0	1	0	0	0
13	Template Design - CSS	0	1	0	0	1	0	0
14	Server Side Includes	0	1	0	0	1	0	0
15	Usage Monitor	0	0	1	0	0	0	0
16	Search Engine	0	0	0	0	0	1	0
17	Forms - Simple	0	0	0	0	0	1	0
18	Calendar	0	1	0	0	0	1	0
19	Decouple from email	0	0	1	0	0	0	0
20	Database Build & Integration	0	0	0	0	0	1	0

Table 4.23 Categorization Consensus Table for PSC ($M_1|M_2$) – Second Iteration

	Category													
	M_1							M_2						
Requirements	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
1 New Look & Feel	1	0	0	0	1	0	0	1	0	0	0	1	0	0
2 Usability	1	0	0	0	1	0	0	1	0	0	0	1	0	0
3 Navigation Scheme	1	1	0	0	1	0	0	1	1	0	0	1	0	0
4 Data Migration	0	1	0	0	0	0	0	0	1	0	1	0	0	0
5 Security	1	0	0	0	0	1	1	1	0	1	0	0	1	1
6 Build Pages	0	1	0	0	0	0	0	0	1	0	0	0	0	0
7 508 Compliance	0	1	0	0	1	0	1	1	1	0	0	1	0	1
8 Host Selection	1	0	0	0	0	0	0	1	0	0	0	0	0	0
9 Installation	0	0	1	1	0	0	0	0	0	1	1	0	0	0
10 Unit Testing	0	1	0	0	0	1	1	0	1	0	0	0	1	1
11 System Testing	0	0	0	0	0	1	1	0	0	0	0	0	1	1
12 Transfer of Operation	0	0	0	1	0	0	0	0	0	0	1	0	0	0
13 Template Design - CSS	0	1	0	0	1	0	0	0	1	0	0	1	0	0
14 Server Side Includes	0	1	0	0	1	0	0	0	1	0	0	1	0	0
15 Usage Monitor	0	0	1	0	0	0	0	0	0	1	0	0	0	0
16 Search Engine	0	0	0	0	0	1	0	0	0	0	0	0	1	0
17 Forms - Simple	0	0	0	0	0	1	0	0	0	0	0	0	1	0
18 Calendar	0	1	0	0	0	1	0	0	1	0	0	0	1	0
19 Decouple from email	0	0	1	0	0	0	0	0	0	1	0	0	0	0
20 Database Build & Integration	0	0	0	0	0	1	0	0	0	0	0	0	1	0

Table 4.24 shows the complexity measurements values generated for the entire system and for each requirement during the second iteration. It should be noted that a facilitation session was conducted to discuss potentially misunderstood requirements based on the correlation coefficients generated by the PFNET technique. The lack of consensus between the two stakeholders as computed by Equation (3.4) for the entire system was 548.60 – max (522.21, 480.82) = 26.39 bits. The positive value of the

information complexity measurement metric indicated that there was still a residual lack of consensus between the stakeholders.

Table 4.24 Complexity Measurement for PSC Second Iteration (Bits)

		M_1	**M_2**	**$M_1 \mid M_2$**
Entire System		522.21	480.82	548.60
Requirements				
1	New Look & Feel	28.05	28.54	31.29
2	Usability	28.05	28.54	31.29
3	Navigation Scheme	42.09	41.34	44.09
4	Data Migration	27.21	16.05	27.21
5	Security	47.77	40.09	47.77
6	Build Pages	15.05	16.05	15.05
7	508 Compliance	53.70	44.58	53.70
8	Host Selection	13.25	11.84	17.15
9	Installation	22.00	16.05	26.76
10	Unit Testing	44.09	44.09	44.09
11	System Testing	26.29	26.29	26.29
12	Transfer of Operation	7.81	4.99	10.57
13	Template Design - CSS	31.86	31.86	31.86
14	Server Side Includes	31.86	31.86	31.86
15	Usage Monitor	11.05	8.81	14.30
16	Search Engine	15.89	15.89	15.89
17	Forms - Simple	15.89	15.89	15.89
18	Calendar	33.33	33.33	33.33
19	Decouple from email	11.05	8.81	14.30
20	Database Build & Integration	15.89	15.89	15.89

The level of disagreement or lack of consensus as observed from the values of the information complexity measurement suggests that the level of disagreement was much lower (26.39 bits) during the second iteration. This is because the *LackOfConsensus* *(M_1,*

M_2) value for the entire system obtained from the first iteration (266.99 bits) is much lower than the value of information complexity metric (26.39 bits) obtained from the second iteration. The lower value for the lack of consensus also suggests that the facilitation session during the first iteration was successful in building a better consensus about requirements among the stakeholders.

4.4.2 Results and Aanalysis for the NMA Project

The six stakeholders of the NMA project are represented as M_1 through M_6. Each stakeholder submitted their categorization information. The consensus categorization of the stakeholders is represented as $M_1|\ldots|M_6$. Table 4.25 shows the information complexity metric values applied to the entire system and values of the metric only for some requirements. These requirements were selected because they reveal some interesting information regarding their categorization. The dark gray shaded values indicate requirements that had a value of 0.00 as the value of the complexity measurement. The requirements that have light gray shade shows the minimum value of complexity metric greater than 0.00 for a particular stakeholder (column). Table 4.26 shows the information complexity measurement values for all the requirements of the NMA project.

Table 4.25 Complexity Measurement for NMA (Bits)

	M_1	M_2	M_3	M_4	M_5	M_6	$M_1\|\ldots\|M_6$
Entire System	3107.40	3182.21	2881.65	4311.12	2119.37	2298.91	15704.65
Requirement							
REQ3645	0.00	49.73	23.33	0.00	38.17	30.83	139.46
REQ3997	13.81	8.24	8.24	0.00	8.24	23.04	38.37
REQ4644	43.77	47.08	1.42	0.00	34.83	48.82	136.58
ARC3535	1.42	1.42	1.42	39.42	27.45	1.42	69.18
ARC3943	46.67	49.73	52.25	0.00	27.45	37.58	186.13
ARC3510	13.23	27.45	1.42	72.07	13.88	13.74	137.98
REQ3307	27.27	44.28	23.33	1.41	18.84	13.74	119.31
REQ4422	1.42	52.25	44.28	39.42	23.33	13.74	157.51
REQ4029	40.66	47.08	1.42	18.40	34.83	48.82	143.81
REQ4814	1.42	31.27	23.33	18.40	31.27	30.83	108.92
REQ4193	34.16	23.33	13.88	72.07	18.84	0.00	129.70
ARC3494	54.09	49.73	44.28	0.00	27.45	13.74	172.82
ARC3895	46.67	47.08	8.24	7.10	8.24	37.58	137.75
REQ4379	43.77	18.84	52.25	0.00	23.33	30.83	156.80
ARC3490	54.09	49.73	44.28	39.42	8.24	0.00	175.16
REQ4425	0.00	18.84	52.25	0.00	23.33	0.00	94.64
REQ5183	54.09	44.28	18.84	0.00	18.84	30.83	147.78
ARC3869	1.42	52.25	27.45	72.07	13.88	23.04	174.53
ARC3509	46.67	31.27	41.31	1.41	27.45	30.83	170.19
REQ4153	18.74	49.73	44.28	72.07	23.33	0.00	195.20

Table 4.26 Complexity Measurements for All NMA Requirements (Bits)

	Requirement	**M_1**	**M_2**	**M_3**	**M_4**	**M_5**	**M_6**	**M_1|…|M_6**
1	ARC3493	34.16	23.33	47.08	72.07	18.84	27.08	174.91
2	ARC3819	34.16	23.33	47.08	107.14	34.83	27.08	235.86
3	ARC3870	8.21	31.27	18.84	18.40	31.27	30.83	99.57
4	REQ4416	46.67	44.28	13.88	107.14	31.27	37.58	254.46
5	REQ3645	0.00	49.73	23.33	0.00	38.17	30.83	139.46
6	ARC4122	54.09	44.28	27.45	72.07	27.45	23.04	240.72
7	REQ3881	40.66	49.73	44.28	39.42	23.33	23.04	167.27
8	REQ4237	13.81	8.24	8.24	72.07	13.88	23.04	111.33
9	REQ4415	54.09	31.27	27.45	72.07	18.84	30.83	235.61
10	ARC3289	54.09	27.45	27.45	72.07	18.84	23.04	180.58
11	ARC3891	46.67	47.08	52.25	72.07	8.24	37.58	229.33
12	REQ3992	8.21	13.88	41.31	26.70	18.84	13.74	89.71
13	REQ3997	13.81	8.24	8.24	0.00	8.24	23.04	38.37
14	REQ4644	43.77	47.08	1.42	0.00	34.83	48.82	136.58
15	REQ3998	74.80	49.73	44.28	72.07	23.33	23.04	236.68
16	ARC3967	54.09	52.25	8.24	72.07	27.45	48.82	259.51
17	REQ5027	54.09	49.73	44.28	17.99	27.45	23.04	190.75
18	REQ4059	27.27	47.08	41.31	72.07	34.83	48.82	236.28
19	ARC3535	1.42	1.42	1.42	39.42	27.45	1.42	69.18
20	REQ4285	8.21	18.84	41.31	72.07	18.84	13.74	156.65
21	ARC3943	46.67	49.73	52.25	0.00	27.45	37.58	186.13
22	ARC3820	13.81	52.25	27.45	18.40	13.88	23.04	117.00
23	REQ3672	40.66	47.08	44.28	17.99	34.83	48.82	179.54
24	REQ5148	40.66	13.88	47.08	27.27	23.33	23.04	145.95
25	ARC3510	13.23	27.45	1.42	72.07	13.88	13.74	137.98
26	ARC3495	43.77	49.73	47.08	72.07	38.17	13.74	244.23
27	REQ4643	18.74	18.84	13.88	72.07	23.33	13.74	125.20
28	ARC2848	46.67	52.25	47.08	107.14	38.17	27.08	300.63
29	REQ3307	27.27	44.28	23.33	1.41	18.84	13.74	119.31
30	REQ4422	1.42	52.25	44.28	39.42	23.33	13.74	157.51
31	REQ4029	40.66	47.08	1.42	18.40	34.83	48.82	143.81
32	ARC3811	30.83	27.45	27.45	72.07	18.84	23.04	144.97
33	REQ4013	34.16	13.88	13.88	18.40	18.84	27.08	87.38
34	REQ4058	43.77	47.08	18.84	72.07	27.45	48.82	235.90
35	ARC3913	46.67	44.28	52.25	72.07	18.84	30.83	244.09
36	ARC3902	30.83	27.45	27.45	72.07	18.84	23.04	144.97
37	REQ3966	54.09	44.28	23.33	72.07	27.45	30.83	237.01
38	REQ4640	27.27	18.84	27.45	72.07	31.27	27.08	195.09
39	REQ4274	30.83	27.45	52.25	26.70	18.84	23.04	132.67
40	ARC3482	46.67	44.28	52.25	72.07	38.17	30.83	267.88
41	ARC3825	18.74	18.84	13.88	72.07	27.45	13.74	144.53

continued

	Requirement	M_1	M_2	M_3	M_4	M_5	M_6	$M_1\|\ldots\|M_6$
42	REQ3455	46.67	31.27	52.25	72.07	23.33	30.83	237.33
43	ARC3994	43.77	52.25	47.08	72.07	27.45	30.83	254.55
44	ARC3903	54.09	52.25	27.45	72.07	31.27	8.17	237.99
45	REQ4641	45.73	44.28	23.33	72.07	8.24	37.58	222.50
46	REQ4138	27.27	13.88	41.31	72.07	27.45	27.08	187.93
47	REQ4602	30.83	52.25	41.31	72.07	23.33	13.74	231.99
48	ARC4788	40.66	47.08	41.31	26.70	34.83	48.82	190.88
49	REQ3834	40.66	47.08	44.28	72.07	18.84	48.82	236.31
50	REQ4014	40.66	47.08	44.28	18.40	34.83	48.82	178.39
51	ARC3904	46.67	31.27	52.25	72.07	31.27	8.17	216.03
52	ARC4121	30.83	44.28	52.25	72.07	27.45	37.58	247.90
53	REQ3395	40.66	52.25	13.88	72.07	38.17	48.82	254.61
54	ARC3488	43.77	49.73	44.28	39.42	8.24	13.74	173.60
55	ARC3991	43.77	52.25	47.08	72.07	23.33	30.83	236.24
56	ARC4865	43.77	52.25	47.08	26.70	23.33	27.08	184.04
57	ARC3901	13.23	49.73	27.45	72.07	13.88	13.74	177.30
58	REQ4039	54.09	49.73	47.08	18.40	38.17	48.82	240.15
59	ARC2945	54.09	52.25	52.25	72.07	27.45	27.08	278.10
60	REQ4484	8.21	31.27	18.84	18.40	31.27	30.83	99.57
61	ARC3829	54.09	52.25	41.31	72.07	38.17	13.74	268.56
62	ARC3892	46.67	49.73	27.45	72.07	13.88	37.58	234.16
63	REQ4814	1.42	31.27	23.33	18.40	31.27	30.83	108.92
64	ARC3965	54.09	44.28	52.25	72.07	27.45	37.58	266.95
65	ARC3900	18.74	18.84	13.88	72.07	23.33	13.74	136.93
66	REQ5186	43.77	52.25	47.08	39.42	23.33	48.82	211.17
67	REQ4866	27.27	44.28	13.88	26.70	27.45	27.08	146.70
68	REQ4193	34.16	23.33	13.88	72.07	18.84	0.00	129.70
69	REQ4356	34.16	23.33	41.31	72.07	27.45	27.08	201.14
70	ARC3494	54.09	49.73	44.28	0.00	27.45	13.74	172.82
71	ARC3895	46.67	47.08	8.24	7.10	8.24	37.58	137.75
72	REQ4379	43.77	18.84	52.25	0.00	23.33	30.83	156.80
73	ARC3890	30.83	27.45	27.45	72.07	38.17	23.04	180.29
74	ARC3894	46.67	47.08	52.25	72.07	34.83	37.58	242.41
75	ARC3490	54.09	49.73	44.28	39.42	8.24	0.00	175.16
76	REQ4912	43.77	47.08	47.08	26.70	38.17	48.82	206.45
77	ARC3264	27.27	18.84	52.25	72.07	27.45	27.08	204.09
78	REQ3915	117.54	13.88	47.08	39.42	23.33	23.04	209.56
79	REQ4425	0.00	18.84	52.25	0.00	23.33	0.00	94.64
80	REQ5183	54.09	44.28	18.84	0.00	18.84	30.83	147.78
81	ARC3869	1.42	52.25	27.45	72.07	13.88	23.04	174.53
82	REQ4505	34.16	23.33	13.88	17.99	18.84	27.08	77.76
83	ARC3509	46.67	31.27	41.31	1.41	27.45	30.83	170.19
84	REQ4153	18.74	49.73	44.28	72.07	23.33	0.00	195.20
85	REQ4287	54.09	13.88	41.31	72.07	18.84	13.74	194.95

The lack of consensus among the six stakeholders for the entire system (see Table 4.25) was $LackOfConsensus(M_1,\ldots, M_6)$ = 11393.53 bits. This large value of the complexity metric compared to that of the PSC are due to larger number of requirements in NMA (85) compared to that of PSC (20) and major disagreements among the stakeholders.

Categories that had single requirements (shaded light gray) were identified based on the analysis of requirements with lowest information complexity metric value greater than 0.00 (see Table 4.25) for a given stakeholder. The analysis of the requirements (shaded dark gray) with information complexity metric values of 0.00 led to the identification of requirements that were not categorized into any of the predetermined high-level categories. The identification of categories with single requirements and requirements that were not categorized were discovered from the categorization tables. It is important to note that the information complexity metric values of these requirements helped focus the search.

The next chapter gives a summary of the research work that was carried out to validate the hypothesis of the dissertation. Conclusions and recommendations based on the analysis of the results we obtained during the period of study and some future work will also be presented.

CHAPTER V

SUMMARY, CONCLUSIONS, AND FUTURE WORK

Our research to date has generated encouraging results that validate the applicability and usefulness of the PFNET technique for requirement understanding between stakeholders both in a university and industrial setting. We also experimented with some information theory-based software metrics in order to explore the usefulness of other metrics in requirements understanding. Early experiments applied the PFNET technique to five student projects all at MSU working for real customers and one project at NORTEL in Dallas, Texas. Later, a modified version of the PFNET technique was applied to two industrial projects at AmerInd Inc., Alexandria, Virginia. These projects varied from small-scale with ten's of requirements to a medium scale project with over one hundred requirements. We applied information theory-based software metrics on the data collected for the two projects at AmerInd Inc.

5.1 Summary of Research Work

The initial results achieved in the classroom encouraged us to extend the PFNET technique to an industrial setting. A brief summary of each experiment is provided below.

5.1.1 Research on Student Projects Research at MSU

During experimentation with student projects, the correlation coefficients predicted misunderstandings between the customer and developers. The DocTime system, which showed very low overall correlation coefficient at the analysis phase, was rejected by the customer after product delivery. In the MSEF-online project, the requirements that were not completely implemented or requirements that were not implemented at all showed low values of correlations coefficients at the requirements analysis phase. The analysis of correlation coefficients revealed all the seeded duplicate requirements in the CORPS project.

5.1.2 Research on Industrial Project at NORTEL

To collect more evidence on the ability of the PFNET technique to reveal misunderstandings about requirements, we conducted another experiment at NORTEL Inc., Dallas Texas. This study revealed misunderstood and duplicate requirements based on the analysis of correlation coefficients for individual requirements. There were no external customers on this project but the technique revealed misunderstood and duplicate requirements for an internal development team.

5.1.3 Research on Iindustrial Projects at AmerInd Inc.

Based on the preliminary evidence from five student projects and one industrial project, we implemented the PFNET technique on a small-scale and medium-scale project at AmerInd Inc. In these experiments, we used the ability of the PFNET technique to predict misunderstanding among stakeholders and used it for their benefit by resolving differences and building consensus.

We modified the original technique to deal with the scalability, consensus, and perception issues among stakeholders (see section 0). The technique was found to be scalable from small-scale projects to a medium-scale project with over one hundred requirements. The correlation coefficients for individual requirements were used to help focus the discussions on potentially misunderstood requirements. The stakeholders found the values of correlations to be intuitive with the way they categorized the requirements.

Ratings on a scale of 1 through 5 were assigned to each requirement discussed during the facilitated sessions. This was an attempt to collect empirical evidence to determine if stakeholders ratings conformed to our initial subjective estimate for a threshold value of 0.7 for the correlation coefficient. It should be noted that we proposed that any requirement that has a correlation coefficient value of less than 0.7 would be considered potentially misunderstood. During the PSC project, the stakeholders did not want to discuss requirements with values of correlation coefficients over 0.7 during the facilitated sessions. They thought that these requirements were well understood.

5.1.4 Research on Information Theory-Based Metrics

Information theory-based metrics have been proposed to measure consensus about requirements among stakeholders for the entire system and for individual requirements.

The results showed that the information theory-based complexity metric was the more useful than the information theory-based size, coupling and cohesion metrics. The complexity metric appeared to be able to measure the overall consensus among the stakeholders for both small-scale and medium-scale projects.

5.2 Conclusions

The experimental results achieved validated the original hypothesis of the dissertation that conceptualization of mental models using PFNETs predict the misunderstandings about the requirements at a very early phase of the software development life cycle. The correlation coefficients helped focus the discussions to resolve the misunderstandings about requirements at AmerInd Inc as well as in other projects.

The research issues (see Section 1.5) addressed during the experimentation led to the following conclusions and recommendations:

- **Research issue 1**: This research issue was to find a way to identify central requirements to represent the mental model of stakeholders. This will be referred to as the requirements identification process. The requirements identification process was achieved by using requirements documents like the SRS for all student projects and for the project at NORTEL. For the PSC project at AmerInd, the requirements identification process was accomplished by using the proposal document. The project manager and developer were also consulted during the requirement identification process. In the NMA project at AmerInd, the SRS document and the project manager were consulted during the requirements identification process. In a typical industrial setting, we recommend that the software engineer consult the requirements document and project managers for requirements identification.
- **Research issue 2**: This research issue was to find suitable and intuitive measures for measuring similarities about the overall system and individual requirements among

the stakeholders. In the PFNET technique correlation coefficients based on the path distance and correlation coefficient based on neighborhood property were examined. The correlation based on the path distance measure seemed intuitive with student projects and with the project at NORTEL. Neighborhood correlations worked well with the projects at AmerInd, since we identified low-level requirements and high-level categories. Based on these observations we recommend using correlation coefficients based on path distances when no high-level categories are identified. For projects where stakeholders are more comfortable with low-level requirements and high-level categories, we recommend correlation coefficients based on the neighborhood measure.

We tested information theory-based metrics on the data collected at AmerInd Inc. The complexity metric was found to be the most useful in measuring the consensus for the entire system among stakeholders. In hindsight, the facilitation sessions could have used this information to decide if more facilitated sessions were required to build consensus. The complexity metric values for the individual requirements were not very intuitive compared to the correlation coefficients from the PFNET technique, but the analysis of the complexity metric for the individual requirements did reveal requirements that were not categorized into any high-level categories and requirements that belonged to just one high-level category. This could be potentially useful information during facilitated sessions.

- **Research issue 3**: This research issue was to validate the effectiveness of the PFNET technique to identify duplicate, ambiguous, and misunderstood requirements by analyzing the correlation coefficients of the requirements. It should be noted that correlation coefficients only point to potentially misunderstood requirements. Further analysis of PFNETs and discussions between stakeholders was required to fully identify the nature of the problem with the requirements. For example, two requirements with very high correlations might be duplicate if they are directly linked in the PFNETs. Ambiguous requirements seem to have low correlations even after facilitation sessions as evidenced in the PSC project. Duplicate requirements in the NMA project were discovered because the stakeholders were evenly split about the categorization of the requirement. Misunderstood requirements were revealed when the stakeholders did not agree with the rationale for categorization by the rest of the stakeholders during a facilitated session.

Our initial subjective estimate of a threshold value for the correlation coefficient as 0.7 was found to be applicable to all small-scale projects at MSU, NORTEL and AmerInd. We could not come to any reasonable conclusions regarding the threshold value of correlation coefficients for a medium-scale project like the NMA.

- **Research issue 4**: This research issue was to validate the feasibility of this technique to scale from a small-scale to medium-scale projects. The PFNET technique was successfully applied to the NMA project with more than one hundred requirements. The technique worked well for both the PSC and the NMA projects at AmerInd Inc., demonstrating the feasibility of applying this technique to projects in a typical industrial setting.

Overall, the experimentation at AmerInd and NORTEL showed that we could identify potentially misunderstood requirements based on the values of their correlation coefficients. Furthermore, the discussion of potentially misunderstood requirements revealed misunderstood, ambiguous and duplicate requirements. The facilitated sessions helped resolve the misunderstandings and build consensus about the requirements. This research also showed that this technique scaled well from small-scale projects to medium-scale projects.

5.3 Contributions

The research reported in this dissertation proposes an inter-disciplinary approach consisting of applying artificial intelligence (AI) techniques in the area of software engineering (SE), specifically to requirements engineering. The proposed technique uses an AI technique to conceptualize the mental models of stakeholders as graphs based on how they categorize requirements. It is then possible to objectively assess the requirements understanding among stakeholders by measuring similarities between the graphs. The technique encourages more interaction among the stakeholders of software projects via facilitated sessions.

We believe that empirical evidence presented in this dissertation may encourage management in industry to spend more time and effort in getting the requirements right at an early stage rather than discovering and fixing the requirement problems later. Our experience has been that management is generally very skeptical about investing resources into understanding requirements unless they see empirical evidence and value in a proposed technique. The research work presented here convinced the management at

AmerInd, of the value of this technique in fostering better understanding about requirements.

This research has been published as journal articles and conference proceedings [36, 37, 38, 39, 40, 41, 42]. We hope that the information disseminated via these publications would encourage more research in the area of requirements engineering. Applying information theory-based software metrics to measure consensus among stakeholders was one step towards this direction and merits additional investigation.

5.4 Future Work

As future work, more evidence may be collected on medium-scale projects and extend this technique to large-scale projects. Further experimentation may be conducted as future work to validate the use of information theory-based software metric to measure consensus among individual requirements. This should perhaps be undertaken as master’s thesis.

We made a subjective estimate on the value of a threshold correlation coefficient (0.7) based on path distance as 0.7. This correlation coefficient has values ranging from –1 to 1. During the experimentation at AmerInd, the correlation coefficients were based on neighborhood property, which has values ranging from 0 to 1. Two requirements during the second iteration of PSC project had a correlation coefficient of 0.75 and were dismissed as well understood by the stakeholders. As far as the evidence suggests, the current value of 0.7 seem to hold for all small-scale projects studied in this research. We were not able to verify this for the medium-scale NMA project. More experimentation is needed in the future to validate the current threshold value of correlation coefficients for medium and large-scale projects.

Our attempt to determine the threshold value of correlation coefficient by collecting the ratings of stakeholders during facilitation did not lead to any definite conclusions. Self-assessment by stakeholders to evaluate misunderstandings about requirements does not seem to work well. There were instances where, during facilitation sessions, some stakeholders seemed pressured to change their ratings based on other's assessment. Stakeholders seemed to be very reluctant to agree that there were misunderstandings even though their categorization of requirements suggested otherwise. They were quick to acknowledge better understandings when the categorization of requirements was similar. As future work, new methods to collect ratings about disagreements among stakeholders may be explored. Further experimentation to include the size of the project, available resources and project deadlines as factors that might contribute in determining the threshold value of correlation coefficient may be explored.

The stakeholders found the graphical display of PFNETs very difficult to understand. The graphs were too large to be displayed on the monitor for a medium-scale project like the NMA. As future work, better notation for displaying the PFNETs may be explored. Interactive ways for presenting PFNETs for medium and large-scale projects like a zooming in and zooming out facility may be explored. Incorporating techniques and notation to display PFNETs that can be easily understood by stakeholders will add value to this technique.

The return on investment (ROI) for adopting this technique in a software process is very important for project managers. More studies to incorporate the amount saved in dollars in identifying duplicate, ambiguous and misunderstood requirements may be carried out. It is also important to determine the cost in dollars involved for adopting this technique for a particular project. There was minimal training required by the stakeholders to use the tool that helps with categorization of requirements. The facilitated sessions took about an hour to discuss twenty requirements when two stakeholders were involved. The NMA project with six stakeholders took about an hour to discuss fifteen requirements. As future work, better techniques to conduct facilitated sessions may be explored to save time. This might require additional training.

REFERENCES

[1] W.H. Acton, P.J. Johnson, and T.E. Goldsmith, “Structure Knowledge Assessment: Comparison of Referent Structures,” *Journal of Educational Psychology*, vol. 86, no.2, 1994, pp. 303-331.

[2] E.B. Allen, “Measuring Graph Abstractions of Software: An Information-Theory Approach,” *Proceedings: 8th Symposium on Software Metrics (METRICS’02),* Ottawa, Canada, June 2002, IEEE Computer Society, pp. 182-193.

[3] E.B. Allen and S. Gottipati, *Measuring Size, Complexity, and Coupling of Hypergraph Abstractions of Software: An Information Theory Approach*, technical report MSU-0212219, Mississippi State University, Mississippi State Mississippi, December 2002, Under review by Software Quality Journal.

[4] E.B. Allen and T.M. Khoshgoftaar, “Measuring Coupling and Cohesion: An Information-Theory Approach,” *Proceedings*: *6th Sixth International Software Metrics Symposium*, Boca Raton, Florida, November 1999, IEEE Computer Society, pp. 119-127.

[5] E.B. Allen and T.M. Khoshgoftaar, and Y. Chen, “Measuring Coupling and Cohesion of Software Modules: An Information-Theory Approach,” *Proceedings: 7th International Software Metrics Symposium*, London, England, April 2001, IEEE Computer Society, pp. 124-134.

[6] B. Boehm and Victor R. Basili, “Software Defect Reduction Top 10 List,” *Computer*, vol. 34, no. 1, January 2001, pp. 135-137.

[7] L.C. Briand, S. Morasca, and V.R. Basili, “ Property-Based Software Engineering Measurement,” *IEEE Transactions on Software Engineering*, vol. 22, no. 1, January 1996, pp. 68-85.

[8] F.P. Brooks, “No Silver Bullet: Essence and Accidents of Software Engineering,” *Computer*, vol. 20, no. 4, April 1987, pp. 10-19.

[9] K.A. Butler, C. Esposito, and R. Hebron,“Connecting the Design of Software to the Design of Work,” *Communications of the ACM*, vol. 42, no. 1, January 1999, pp. 38-46.

[10] M.J. Caroll and J.R. Olson, “Mental models in Human-Computer Interaction,” *Handbook of Human-Computer Interaction*, M. Helander, ed., Elsevier Sceince Publishers B. V., North-Holland, 1989, pp. 45-60.

[11] C. Chen, “Structuring and Visualizing the WWW by Generalized Similarity Analysis,” *Proceedings: 8th ACM conference on Hypertext*, Southamton, United Kingdom, 1997, pp. 177-186.

[12] C. Chen, G. Gagaudakis, and P. Rosin, “Similarity-based Image Browsing,” *Proceedings: 16th IFIP World Computer Congress, International Conference on Intelligent Information Processing*, Beijing, China, 2000, Publishing House of Electronics Industry, pp. 206-213.

[13] N.M. Cooke and R.W. Schvaneveldt, “Effects of Computer Programming Experience on Network Representations of Abstract Programming Concepts,” *International Journal of Man-Machine Studies*, vol. 29, 1988, pp. 407-427.

[14] N.M. Cooke, F.T. Durso, and R.W. Schvaneveldt, “Recall and Measures of Memory Organization,” *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 12, no. 4, 1986, pp. 538-549.

[15] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, 1991.

[16] A.M. Davis, *Software Requirements: Analysis and Specification*, Prentice Hall, New Jersey, 1990.

[17] A. Davis et al., “Identifying and Measuring Quality in a Software Requirement Specification,” *Software Requirements Engineering*, R.H. Thayer, and M. Dorfman, ed., IEEE Computer Society Press, Los Alamitos, California , 1997, pp.164-175.

[18] T. Dayton, F.T. Durso, and J.D. Shepard, “A measure of the Knowledge Reorganization Underlying Insight,” *Pathfinder Associative Networks: Studies in Knowledge Organization*,” R. Schvaneveldt, ed., Ablex Publishing Corp., Norwood, New Jersey, pp. 267-277, 1990.

[19] D.W. Dearholt, N. Gonzales, and G. Kurup,“Monotonic search for computer vision databases,” *Proceedings: 22nd Asilomar Conference on Signals, Systems and Computers*, Pasedena, California, October 31 – November 2, 1988, vol. 2, pp. 548 –553.

[20] D.W. Dearholt and R.W. Schvaneveldt, “Properties of Pathfinder Networks,” *Pathfinder Associative Networks: Studies in Knowledge Organization*,” R. Schvaneveldt, ed., Ablex Publishing Corp., Norwood, New Jersey, pp. 267-277, 1990.

[21] D.W. Dearholt, *Modeling Dynamic Systems Using Co-occurrence and Associative Networks*, Technical Report MSU-980313, Mississippi State University, Mississippi State, Mississippi, January 1998.

[22] D.W. Dearholt, U. K. Kudikyala, I. Banicescu and M.L. Bilderback, “Providing the Web Operating System with Effective Search Method for Sharing Available Resources,” *Proceedings: Workshop Distributing Computing on the Web (DCW ’99)*, Germany, 1999, http://www.cse.msstate.edu/~ioana/publications.html, (current June, 2004).

[23] M. Dorfman, “Requirement Engineering,” *Software Requirements Engineering*, R.H. Thayer and M. Dorfman, ed., IEEE Computer Society Press, Los Alamitos, California, pp. 7-22, 1997.

[24] F.T. Durso and K. A. Coggins, “Graphs in the Social and Psychological Sciences: Empirical Contributions of Pathfinder,” *Pathfinder Associative Networks: Studies in Knowledge Organization*,” R. Schvaneveldt, ed., Ablex Publishing Corp., Norwood, New Jersey, pp. 267-277, 1990.

[25] A. Faro, and D. Giordano, “From User’s Mental Models to Information System’s Specification and Vice Versa by Extended Visual Notation,” *Proceedings: International Professional Communication Conference (IPCC’ 95)*, Smooth Sailing to the Future, Savannah, Georgia, September 27-29, 1995, IEEE International, pp. 44-48.

[26] S.R. Faulk, “Software Requirements: A tutorial in Software Requirements Engineering,” *Software Requirement Engineering*, R.H. Thayer and M. Dorfman, ed., IEEE Computer Society Press, Los Alamitos, California, pp. 128-149, 1997.

[27] R.H. Fowler and D.W. Dearholt, “Information Retrieval Using Pathfinder Networks,” *Pathfinder Associative Networks: Studies in Knowledge Organization*,” R. Schvaneveldt, ed., Ablex Publishing Corp., Norwood, New Jersey, pp. 267-277, 1990.

[28] G. Gagaudakis, P.L. Rosin, and C. Chen, “Using CBIR and Pathfinder Networks for Image Database Visualization,” *Proceedings: 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 3-7, 2000, vol. 1, pp. 1052-1055.

[29] T.E. Goldsmith and D. M. Davenport, "Assessing Structural Similarity of Graphs," *Pathfinder Associative Networks: Studies in Knowledge Organization*," R.Schvaneveldt, ed., Ablex Publishing Corp., Norwood, New Jersey, pp. 267-277, 1990.

[30] T.E. Goldsmith and P. J. Johnson, "A Structural Assessment of Classroom Learning," *Pathfinder Associative Networks: Studies in Knowledge Organization*," R. Schvaneveldt, ed., Ablex Publishing Corp., Norwood, New Jersey, pp. 267-277, 1990.

[31] R.L. Gomez, O.D. Hadfield, and L.D. Housner, "Conceptual Maps and Simulated Teaching Episodes Indicators of Competence in Teaching Elementary Mathematics," *Journal of Educational Psychology*, vol. 88, no. 3, 1996, pp. 572-585.

[32] P. Gonzalvo, J. Canas, and M.T. Bajo, "Structural Representations in Knowledge Acquisition," *Journal of Educational Psychology*, vol. 86, no. 4, 1994, pp. 601-616.

[33] S. Gottipati, *Empirical Validation of the Usefulness of Information Theory-Based Metrics*, master's thesis, Department of Computer Science, Mississippi State University, Mississippi State, Mississippi, May 2003.

[34] C. Jones, "Strategies for managing requirements creep," *Computer*, vol. 29, no. 6, June 1996, pp. 92-94.

[35] M. Keil, P.E. Cule, K. Lyytinen, and R.C. Schmidt, "A Framework for Identifying Software Project Risks," *Communications of the ACM*, vol. 41, no. 11, November 1998, pp 76-83.

[36] U. K. Kudikyala, E.B. Allen and R. B. Vaughn, "Measuring Consensus during Verification and Validation of Requirements", Proceedings of the tenth IEEE International Software Metrics Symposium (METRICS 2004), Chicago, Illinois, September 2004. (under review as Late Breaking Paper)

[37] U. K. Kudikyala and R. B. Vaughn, "Software Requirements Understanding using Pathfinder Networks: Discovering and Evaluating Mental Models " *Journal of Systems and Software*, 2004. (in print)

[38] U. K. Kudikyala and R. B. Vaughn, "Software Requirements Understanding using Pathfinder Networks, " *CrossTalk: The Journal of Defense Software Engineering*, vol. 17, no. 5, May 2004, pp. 16-25.

[39] U. K. Kudikyala and R. B. Vaughn, “Software Requirements Understanding using Pathfinder Networks”, *Systems and Software Technology Conference (STC),* Salt Lake City, Utah, April 19-22, 2004.

[40] U.K. Kudikyala and R.B.Vaughn, “Software Requirements Understanding Using Pathfinder Networks As Mental Models, *”Proceedings of the 2004 ASEE Southeastern Section Conference*, Auburn, Alabama, April 4-6 2004.

[41] U. K. Kudikyala and R.B. Vaughn, “Reducing Misunderstanding of Software Requirements by Conceptualization of Mental Models using Pathfinder Networks,” *Proceedings of the Second Southeastern Software Engineering Conference (SE)*^{2}, Huntsville, Alabama, March 29- April 1, 2004.

[42] U. K. Kudikyala, “Conceptualization of mental models using Pathfinder networks to understand software requirements, ” *Proceedings of the First Southeastern Software Engineering Conference (SE)*^{2}, Huntsville, Alabama, April 9-11, 2002, pp. 316-333.

[43] U. Kudikyala, *Comparison of Shortest-path Algorithms for K-Local Image Graphs*, master’s project, Department of Computer Science, Mississippi State University, Mississippi State, Mississippi, June 1999.

[44] X. Lu, *Using Pathfinder Networks to Analyze and Categorize Software Requirements*, master’s thesis, Department of Computer Science, Mississippi State University, Mississippi State, Mississippi, August 2000.

[45] C.C. Mann, “Why Software is so Bad,” *Technology Review*, vol. 105, no. 6, August 2002, pp. 33-38.

[46] J.E. McDonald and R. W. Schvaneveldt, “The Application of User Knowledge to Interface Design,” *Cognitive Science and its Applications for Human-computer Interface*, R. Guindon, ed., Erlbaum, Hillsdale, New Jersey, 1988, pp. 289-338.

[47] J.E. McDonald, K.R. Paap, and D.R. McDonald, “Hypertext Perspectives: Using Pathfinder to Build Hypertext Systems,” *Pathfinder Associative Networks: Studies in Knowledge Organization,*” R. Schvaneveldt, ed., Ablex Publishing Corp., Norwood, New Jersey, pp. 197-212, 1990.

[48] S.N. Mohanty, “Entropy Metrics for Software Design Evaluation,” *Journal of Systems and Software*, vol. 2, 1981, pp. 39-46.

[49] A.J. Norman, “Some Observations on Mental Models,” *Mental models,* D. Gentner and A. L. Stevens, ed., Lawrence Erlbaum, Hillsdale, New Jersey, pp. , 1983.

[50] J. Parsons, “An Information Model Based on Classification Theory,” *Management Science*, 1996, vol. 42, no. 10, pp.1437-1453.

[51] J. Parsons and Y. Wand, “Emancipating Instances from the Tyranny of Classes in Information Modeling,” *ACM Transactions on Database Systems*, vol. 25, no. 2, June 2000, pp. 228-268.

[52] J. Parsons and Y. Wand, “Using Objects for System Analysis,” *Communications of the ACM*, vol. 40, no. 12, December 1997, pp. 104-110.

[53] J. Parsons and Y. Wand, “Choosing Classes in Conceptual Modeling,” *Communications of the ACM*, June, 1997, vol. 40, no. 6, pp. 63-69.

[54] K. Potosnak, “Mental Models: Helping Users Understand Software,” *IEEE Software*, September 1989, vol. 6, no. 5, pp. 85-88.

[55] J. Rothfeder, “It’s Late, Costly, Incompetent-but Try Firing a Computer System.,” *Business Week*, 7 November 1988, pp. 164-165.

[56] R.W. Schvaneveldt, “Proximities, Networks, and Schemata,” *Pathfinder Associative Networks: Studies in Knowledge Organization*,” R.Schvaneveldt, ed., Ablex Publishing Corp., Norwood, New Jersey, pp. 135-148, 1990.

[57] R.W. Schvaneveldt et al., “Measuring the Structure of Expertise,” *International Journal of Man-Machine Studies*, 1985, vol. 23, no.3, pp. 699-728.

[58] R.W. Schveneveldt, F.T. Durso, and D.W. Dearholt, “Network Structures in Proximity Data,” *The Psychology of Learning and Motivation: Advances in Research and Theory*, G. H. Bower, ed., Academic Press, New York, 1989, pp. 249-284.

[59] M. R. Shand, “User Manuals as Project Management Tools: Part I-Theoretical Background,” *IEEE Transactions on Professional Communication*, 1994, vol. 37, no. 2, pp. 75-80.

[60] C.E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, Illinois, 1949.

[61] I. Sommerville and P. Sawyer, *Requirements Engineering: A good practice guide*, John Wiley and Sons, New York, 1999.

[62] M-A.D. Storey, F.D. Fracchia, and H.A. Müller, “Cognitive Design Elements to Support the Construction of a Mental Model during Software Visualization,” *Proceedings: 5th International Workshop on Program Comprehension (IWPC’97)*, Dearborn, Michigan, May 1997, pp. 17-28.

[63] M.E. Torres and R.B. Vaughn, *An Automated Tool for Software Requirement Refinement and Pathfinder Network Generation*, Technical Report MSU-030922, Mississippi State University, Mississippi State, Mississippi, September 2003.

[64] M.H. van Emden, “Hierarchical Decomposition of Complexity,” *Machine Intelligence*, vol. 5, 1970, pp. 361-380.

[65] S. Watanabe, “Information Theoretical Analysis of Multivariate Correlation,” *IBM Journal of Research and Development*, vol. 4, no. 1, January 1960, pp. 66-82.

[66] R. Weber, “The Information Systems Discipline: The Need for the Nature of the Foundational Core,” *Proceedings: Information Systems Foundations Workshop on Ontology, Semiotics and Practice (ISF’99)*, September 1999, Sydney, Australia, http://www.comp.mq.edu.au/isf99/Weber.htm, (current February 28, 2002).

[67] H.D. White, X. Lin, and J. Buzydlowski, “Co-cited Author Maps as Interfaces to Digital Libraries: Designing Pathfinder Networks in Humanities,” *Proceedings: International Conference on Information Visualization (IV2000)*, July 2000, London, England, pp. 25-30.

[68] H. Yi, *Automated Web Based Tool for Software Requirement Refinement Using Pathfinder Networks*, master’s project, Department of Computer Science, Mississippi State, Mississippi State University, December 2001.