Spring 2019

A Hierarchical Core Reference Ontology for New Technology Insertion Design in Long Life Cycle, Complex Mission Critical Systems

Kevin J. Michael
Old Dominion University

Recommended Citation

Michael, Kevin J.. "A Hierarchical Core Reference Ontology for New Technology Insertion Design in Long Life Cycle, Complex Mission Critical Systems" (2019). Doctor of Engineering (D Eng), doctoral_project, Engineering Management, Old Dominion University, DOI: 10.25777/b3se-zt18
https://digitalcommons.odu.edu/emse_deng_projects/6

A HIERARCHICAL CORE REFERENCE ONTOLOGY FOR NEW TECHNOLOGY INSERTION DESIGN IN LONG LIFE CYCLE, COMPLEX MISSION CRITICAL SYSTEMS

by

Kevin J. Michael
B.S. Computer Science, University of Maryland
M.S. Management Information Systems, University of Maryland

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF ENGINEERING

ENGINEERING MANAGEMENT AND SYSTEMS ENGINEERING

OLD DOMINION UNIVERSITY
May 2019

Approved by:

T. Steven Cotter (Director)

Charles Daniels (Member)

Mamadou Seck (Member)

Daniel Haim (Member)

ABSTRACT

A HIERARCHICAL CORE REFERENCE ONTOLOGY FOR NEW TECHNOLOGY INSERTION DESIGN IN LONG LIFE CYCLE, COMPLEX MISSION CRITICAL SYSTEMS

Kevin J. Michael
Old Dominion University 2019
Director: Dr. T. Steven Cotter

Organizations, including government, commercial and others, face numerous challenges in maintaining and upgrading long life-cycle, complex, mission critical systems. Maintaining and upgrading these systems requires the insertion and integration of new technology to avoid obsolescence of hardware software, and human skills, to improve performance, to maintain and improve security, and to extend useful life. This is particularly true of information technology (IT) intensive systems. The lack of a coherent body of knowledge to organize new technology insertion theory and practice is a significant contributor to this difficulty. This research organized the existing design, technology road mapping, obsolescence, and sustainability literature into an ontology of theory and application as the foundation for a technology design and technology insertion design hierarchical core reference ontology and laid the foundation for body of knowledge that better integrates the new technology insertion problem into the technology design architecture.

ACKNOWLEDGEMENTS

I would like to recognize some individuals for steering and supporting this endeavor. My dissertation committee chair and supervisor Dr. T. Steven Cotter– The most devoted faculty I have ever seen to mentor a student. Likewise, I would like to recognize my doctoral committee members Dr. Charles B. Daniels and Dr. Daniel Hames for their support throughout the preparation of this dissertation document.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

CHAPTER 1.

INTRODUCTION

1.1 New Technology Insertion Theoretical Problem

Organizations, including government, commercial and others, face numerous challenges in maintaining and upgrading long life-cycle, complex, mission critical systems. Maintaining and upgrading these systems requires the insertion and integration of new technology to avoid obsolescence (both hardware and software), improve performance, maintain and improve security, and extend useful life. This is particularly true of information technology (IT) intensive systems.

Current research and experience shows that the process of inserting and integrating new technology into these systems is difficult, expensive, and slow (Webber, 2002) (Kerr, Phaal, & Probert, 2008) (Kubricky, 2008). Webber (2002) found that technology insertion into legacy systems "is often constrained by existing software architectures, proprietary interfaces, physical space, power provisions and existing acquisition processes." Both the United Kingdom Ministry of Defence and the United States Department of Defense (DoD) realize they need a better understanding of technology insertion and how to apply it more effectively (Kerr, Phaal, & Probert, 2008). According to Kubricky (2008) "moving technology is hard." He further states that corporations follow their technology into obscurity while defense experiences slow technology adoption. Being able to upgrade these systems must be cost-effective to preserve existing systems and investments and must be done more rapidly, so that the new technology being used is not obsolete before being deployed. The ability to rapidly integrate current or future technology into these systems has proven to be formidable.

New technology insertion is a direct result of "Technology Jumping", when new technologies are in their exponential growth stage. "Technology jumping sustains exponential growth as companies switch to new technologies when the current ones reach their points of diminishing return." (Denning & Lewis, 2017). Moore's Law, which predicts a doubling of computing power every 18 months has held true for over 50 years. While this exponential growth has begun reaching physical limits such as the number and size of individual transistors, other approaches such as multi-core processors have extended this growth. Several other approaches on the horizon promise to continue this trend further into the future. Indeed, Moore's Law is an economic theory rather than a physical law (Shalf & Leland, 2015). Based on current progress, miniaturization using current methods will reach an absolute limit around 2036 due to quantum mechanics and the uncertainty principle and potentially sooner due to the economics of producing such devices. Recent developments include three-dimensional integrated circuits (stacking of gates and transistors on a chip) rather than two-dimensional, using carbon nanotubes to further reduce size, and exploiting quantum computing to radically change the fundamental methods of computing (Wu, Shen, Reinhardt, Szu, & Dong, 2013). As technology advances rapidly, legacy systems must be able to incorporate technological advances in order to improve their efficiency and effectiveness while avoiding obsolescence.

The lack of a coherent body of knowledge to organize new technology insertion theory and practice is a significant contributor to this difficulty. This research organized the existing design and technology road mapping, obsolescence, and sustainability literature into an ontology of theory and application as the foundation for a new technology insertion body of knowledge.

1.3 Purpose of the Study

This research will examine the literature for the many challenges, approaches, and methods, of managing the insertion and integration of new or modern technology into long lifecycle, complex, mission critical systems. DoD Instruction 5000.02 (US Department of Defense, 2015) defines a Mission-Critical Information System as a “system that meets the definitions of “information system” and “National Security System” (NSS) in the Clinger-Cohen Act (CCA), the loss of which would cause the stoppage of warfighter operations or direct mission support of warfighter operations. The designation of mission critical should be made by a component head, a combatant commander (CCDR), or designee.” The Defense Acquisition University (2015) more generally defines a Mission Critical System as: “A system whose Operational Effectiveness (OE) and Operational Suitability (OS) are essential to successful completion or to aggregate residual combat capability. If this system fails, the mission likely will not be completed. Such a system can be an auxiliary or supporting system as well as a primary mission system.” In a business sense, “unexpected disruptions of mission-critical operations can lead to dramatic consequences. In some cases, such disruptions may cost firms millions of dollars, even if they last only a few hours or even minutes.” (Kim, Cohen, Netessine, & Veeraraghavan, 2010). “There are a number of fields including transportation, finance, telecommunications, medical devices, that are critical and require high assurance.” (Ponsard, et al., 2004). The working definition of mission critical systems for this research are those systems that must perform as designed and failure to do so may result in catastrophic loss of life, equipment or capability. Examples include military weapon and combat systems, commercial aircraft, spacecraft and satellites. Many of these systems are part of a system of systems or family of systems, which further increases the complexity from generation to generation.

The purpose of this research was to organize the literature into a new technology insertion core reference ontology that generalizes unique theory and application of existing approaches or combination of approaches into a new technology insertion design body of knowledge. Specifically, this project focused on those systems which have a life-cycle measured not in months or years but rather in decades, and the research focused further on those systems that have mission critical implications, although the approaches examined may also apply to shorter-lived or non-mission critical systems. There are many examples of such systems, both commercial and government. This project focused on new technology integration for long life-cycle, complex, mission critical systems.

Many factors were considered. As one example, the Department of Defense relies heavily on test and evaluation prior to deploying new technology (US Department of Defense, 2015). This is a costly and time-consuming process, which can be partially mitigated through the use of modeling and simulation (M&S) to reduce reliance on live test and evaluation. Another purpose of this research was to contribute to the streamlining of the design process to enable faster integration of new technology while maintaining the security, safety and performance of the systems.

1.3 The New Technology Insertion Problem

Over the past several decades it has become apparent that it is very difficult to integrate new technology into systems that are developed over a long period of time and are in service for decades, such as aircraft, ships, weapons systems and combat systems. Many of these are technology-intensive systems. They are considered mission critical and have a bearing on life and death of humans. Being able to integrate new technology into these systems is necessary to overcome obsolescence, reduce costs, maintain and improve security, and improve performance.

Many approaches have been tried, but none appear to provide a single best solution. Singh and Sandborn (2006) observed that the Mitigation of Obsolescence Cost Analysis

(MOCA) method enables early forecasting of refresh dates to allow optimum refreshes to be performed, however there are situations in which the present MOCA solution is incomplete such as the treatment of software. Another approach, Performance Based Logistics (PBL) has been used to address Diminishing Manufacturing Sources and Material Shortages (DMSMS), however it has been noted that "products with long development processes will likely become obsolete more quickly than anticipated." (Feldman & Sandborn, 2007). A more recent trend has been availability contracts where industry delivers a complete product-service system (PSS). The challenge is that when negotiating the contract, the solution provider and the customer must be confident of the whole life costs (WLC) for 20, 30 or even 40 years into the future (Romero-Rojo, Roy, Shehab, & Wardle, 2009). As a result, government and private industry spend a lot of money, time and resources on maintaining obsolete systems rather than upgrading them. In many cases this is compounded by laws and regulations that govern the procurement, test and evaluation of new technology.

New technology covers a broad spectrum. It may be new hardware such as larger capacity hard drives and storage that have a new interface (e.g., SATA), communications capability such as new higher speed or wireless, and new software standards such as JAVA. This makes it difficult to devise a consistent process for integrating new technology. Finding more effective ways of incorporating such technology into legacy systems, especially those that are mission critical, could result in significant performance improvements and cost reduction while reducing risk of failure.

1.4 Research Delimitation

In the knowledge domain, this research sought to develop only a new technology insertion and sustainability core reference ontology in support of future ontological and body of knowledge development. Roussey, Pinet, Kang, and Corcho (2011) present a taxonomy of ontologies by scope ranging from top-level foundational ontologies down to expert system local application ontologies (an expansion of this taxonomy will be set forth in the literature review). A core reference ontology is a second level ontology applied by a defined group of users to specify the central concepts and relations of a given knowledge domain. It depends on the top-level foundational ontology for its conceptual, taxonomical, and axiomatical foundation, and itself forms the foundation for the integration of domain ontologies to fully specify the operational knowledge scope of the domain. Lim, Ying, and Yong (2015) note that formal ontologies are comprised of concepts that are taxonomically and axiomatically based. Thus, a formal ontology is, at minimum comprised, of a set of concepts, the taxonomical hierarchy relationships within the concepts, and the axiomatic first order logical relationships between taxonomic concepts. The end product of this research is only the core reference ontology of physical new technology insertion and sustainability relationships necessary and sufficient to specify the different domains within the body of knowledge.

CHAPTER 2

BACKGROUND OF THE STUDY

2.1 Review of the Design Literature

Although highly fractionated and not organized as a general discipline, design theory can be roughly classified as visual and performing arts design, engineering design, social design, software/information design, and systems design. Much overlap exists among the categories of this general categorization. There is no top level foundational ontology organizing general design knowledge and practices.

The visual and performing arts body of knowledge exists in various books, journals, magazines, states' standards of learning for primary and secondary education, and college and university curricula. The most comprehensive approach to developing a visual and performing arts body of knowledge was initiated in 1994 by the National Art Education Association with the first release of its National Core Arts Standards. The stated goal of this initiative is the creation of "…voluntary national standards for visual arts, dance, music, theater and media arts." The NAEA (2014) released a new generation of NCAS standards in 2014. No work has been done toward creating a core reference ontology for this design sub-discipline body of knowledge.

Engineering design is a broad field that roughly covers aerospace, agricultural, architectural, biological, chemical, civil, electrical, military, and mechanical design. The engineering design body of knowledge is most completely expressed in the National Society of Professional Engineers *Professional Engineering Body of Knowledge* (2013). The NSPE-PEBoK defines the following engineering disciplines:

- Aerospace Engineering

- Agricultural Engineering
- Biochemical Engineering
- Bioengineering
- Biomedical Engineering
- Biomolecular Engineering
- Biological Engineering
- Ceramic Engineering
- Chemical Engineering
- Civil Engineering
- Computer Engineering
- Construction Engineering
- Electrical Engineering
- Engineering
- Engineering Management
- Engineering Mechanics
- Engineering Physics
- Engineering Science
- Environmental Engineering
- General Engineering
- Geological Engineering
- Industrial Engineering
- Manufacturing Engineering
- Marine Engineering

- Materials Engineering
- Metallurgical Engineering
- Mechanical Engineering
- Mining Engineering
- Naval Architecture
- Nuclear Engineering
- Radiological Engineering
- Ocean Engineering
- Petroleum Engineering
- Software Engineering
- Surveying
- Systems Engineering

Although the NSPE has worked toward a unified definition of the Professional Engineering Body of Knowledge, it has not specified a top-level foundational ontology for engineering design, and it has not specified a core reference ontology for each sub-discipline listed above.

Development of design ontologies is a comparatively new sub-discipline, and only isolated domain and application local engineering design ontologies have been developed. Literature search found the following seven core reference, nine domain, and eight application ontologies. A foundational design theory ontology was not identified. Gruber and Olsen (1996) were the first to develop a local application design ontology for elevator configuration. They demonstrated that a formal, machine readable ontology of input and output configuration task descriptions could be developed to characterize semantic constraints of possible design solutions. Lin, Fox, and Bilgic (1996) developed a domain ontology as support for the Knowledge Aided

Design (KAD) requirements configuration system to address ambiguity in design terminology, requirements traceability, detection of redundant or conflicting requirements, integration of parts with their features and parameters and constraints, document creation, reusability and extensibility, and the control of change management. Weilinga and Schreiber (1997) described the hierarchical taxonomy structure domain-specific, method-independent knowledge categories of the Sisyphus-VT developed ontology. Horvath, Vergerest, and Kuczogi (1998) specified that design concepts interactions and validity are governed by constraints that allow a design inference engine to select appropriate design concepts for incomplete user functional specifications. Soininen, Tiihonen, Mannisto, and Sulonen (1998) developed a general product configuration application ontology based on the main approaches to requirements configuration. López, Gómez-Pérez, Sierra, and Sierra (1999) used Methontology and Ontology Design Environment (ODE) in the development of the Chemicals application ontology to overcome the problems the absence of ontological development principles, criteria, and life cycle phases.

Richards and Simoff (2001) argued that ontology development is affected by human learning in the knowledge acquisition process and requires acquisition techniques that are able to identify and capture change. They demonstrated a knowledge acquisition process "…based on the combined use of cases, ripple-down rules (RDR), formal concept analysis (FCA), and the Activity/Space (A/S) ontology …" (p. 121) in the development of a Psycho-Geriatric applied ontology. Kitamura and Mizoguchi (2003, 2004) proposed, developed, and deployed a core reference ontology of meta-functional design concepts that specify a vocabulary of function behaviors and functional relationship between functions with a goal of systematizing functional design knowledge. Sim and Duffy (2003) proposed and core reference ontology of generic engineering design activities of definition, evaluation, and management. Liang and Paredis

(2004) developed a semantic structure for a core reference port ontology that formalized the conception of ports (points of interaction between design functions) and promoted reasoning about functional and component interconnections for design engineers and computer-aided design systems. Grosse, Milton-Benoit, and Wileden (2005) proposed a set of formal core reference ontologies classifying design engineering analysis models. Ahmed (2005) and Storga, Andreasen, and Marjanovié (2005) proposed and developed a product development (PD) domain ontology to provide a first organization of the body of data, information, and engineering knowledge for generic product design. Storga, Andreasen, and Marjanovic (2005) developed a core reference *Design Ontology* as a foundation for collaborative research and development of a general product development ontology. Grosse, Milton-Benoit, and Wileden (2005) developed a core reference ontology called ON-TEAM that provided the foundation for the exchange, adaptation, and interoperability of engineering analysis models (EAMs) within and across engineering design organizations. Kitamura, Sano, and Mizoguchi (2000) incorporated the automatic identifications of functional structures based on behavioral models with the objective of enabling machine understanding to limit and screen the functional search space into their core reference ontology of meta-functional design concepts (Kitamura & Mizoguchi, 2003) (Kitamura & Mizoguchi, 2004). Ahmed, Kim, and Wallace (2007) developed a domain ontology EDIT that indexed design knowledge captured within a design system, stored that knowledge and then provide a structured interface for navigating, browsing, and retrieving design knowledge through hierarchical product descriptions in the aerospace industry. Witherell, Krishnamurty, and Grosse (2007) developed a local application optimized design ontology ONTOP that incorporated standardized design optimization terminology, formal design optimization methods definitions, idealizations, and assumptions supporting optimized design models. Yang, Dong, and Miao

(2008) incorporated a configuration domain meta-ontology within a four layer product modeling architecture to define general and common terms and relations across product specific design configuration local application domains. Catalano, Camossi, Ferrandes, Cheutet, and Sevilmis (2009) developed a domain Product Design Ontology (PDO) to share shape data and shape processing methods across disparate product design domains.

Hsieh, Lin, Chi, Chou, and Lin (2011) proposed extraction of concepts, instances, and relationships from domain specific design handbooks to expedite development of domain and application level ontologies. Chen, Chen, Leong (2013) proposed an ontology-learning customer needs representation (OCNR) system that used natural language processing to identify and extract key concepts and relationships to establish application specific customer needs ontologies. Liu and Hu (2013) proposed an application design rational representation methodology to capture, rationalize, and represent key design concepts and relationships in Web Ontology Language. Liu, Lim, and Lee (2013) proposed revisions to application specific product family design methodologies to apply metrics of ontology-based commonalities to reveal conceptual similarities across designs, apply faceted concept rankings, and apply ranked results toward design architecture selection. Ming, Yan, Wang, Panchal, Goh, Allen, and Mistree (2016) proposed a domain ontology for capturing, representing, and documenting hierarchical design decisions in complex systems.

2.2 Review of the Technology Insertion Literature

The domains of this research were government defense, government non-defense, and commercial. While the specific applications differ greatly they share common difficulties in the integration of new technology. There are common problems with addressing technology obsolescence and maintaining current cybersecurity. All of these applications have requirements to maintain state-of-the-art performance, whether to defeat adversaries on the battlefield for the military, provide for public safety is in the case of the Federal Aviation Administration (FAA), maintain critical space assets as in the case of the National Aeronautics and Space Administration (NASA) and the National Oceanographic and Space Administration (NOAA), or remain commercially competitive.

Obsolescence is a key driver for new technology integration. Much research has been performed by the University of Maryland Center for Advanced Life Cycle Engineering. According to their web site “The Center for Advanced Life Cycle Engineering (CALCE) is recognized as a founder and driving force behind the development and implementation of physics-of-failure (PoF) approaches to reliability, as well as a world leader in accelerated testing, electronic parts selection and management, and supply-chain management. CALCE is at the forefront of international standards development for critical electronic systems, having chaired the development of several reliability and part selection standards. CALCE is staffed by over 100 faculty, staff, and students and in 1999 became the first academic research facility in the world to be ISO 9001 certified. Collectively, CALCE researchers have authored over 35 internationally acclaimed textbooks and well over 1000 research publications relevant to electronics reliability. Over the last 15 years, CALCE has invested over $75 million in developing methodologies, models, and tools that address the design, manufacture, analysis, and management of electronic systems.” (University of Maryland, 2016). CALCE staff have published numerous articles addressing various aspects of obsolescence management and technology insertion.

Technology obsolescence problems increase as the pace of technological progress increases, and affect sustainment-dominated industries to a greater degree. Reactive approaches to ensure enough parts to last through the platform's lifecycle include lifetime buys, aftermarket sources and other mitigation approaches. Strategically planned design refreshes can help reduce long-term costs over reactive mitigation alone. "Design refresh planning is performed by organizations who wish to avoid the high costs of purely reactive obsolescence solutions." (Myers & Sandborn, 2007).

In many cases the system lifecycle is longer than the lifecycle of its component technologies. These mismatches lead to high sustainment costs due to obsolescence in long lifecycle systems such as military and avionics applications. Singh and Sandborn (2006) propose a methodology for optimum design refresh planning. "The methodology minimizes the lifecycle cost by determining the optimum combination of design refresh schedule for the system (i.e., when to design refresh) and the design refresh content for each of the scheduled design refreshes."

Sandborn and Singh (2005) propose a methodology for forecasting technology insertion concurrent with obsolescence driven design refresh planning by optimizing the life cycle cost of the system. This analysis leads to a design refresh schedule for the system. Their approach considers both the date of the design refresh as well as what is changed at the design refresh.

Viability of systems should be a consideration for technology insertion. "Viability is a measure of the producibility, supportability, and evolvability of a system and can serve as a metric for assessing technology insertion opportunities." (Sandborn, Herald, Houston, & Singh, 2003). Sustainment is defined as "keeping an existing system operational and maintaining the ability to continue to manufacture and field versions of the system that satisfy the original requirements." This includes satisfying evolving requirements by manufacturing and deploying new versions of the system often requiring replacement of technologies with newer technologies.

Technology insertion includes determining which technologies to replace when that design refresh should take place. Technology replacement considerations include performance, reliability, environmental impact, cost, and logistics, and when or whether other design refreshes will take place. This approach considers the value of the technology refreshment and insertion to support both affordability and capability needs including hardware, software, information and intellectual property.

2.2.1 Design Theory of New Technology Insertion

Design Theory focuses on designing in methods of inserting new technology into a system during the system's lifecycle. This includes addressing expected obsolescence of components requiring integration of new components, as well as unexpected events such as disruptive advances in technology and resulting disadvantage to competitors or adversaries. Design theory includes the body of knowledge, models, decision making, controlling risk of failure, problem solving strategies, etc.

According to Singh and Sanborn (2006), technology mismatch occurs when "technologies have lifecycles that are shorter than the lifecycle of the product they are in" resulting in high sustainment costs. This can be addressed through design refresh planning. Long lifecycle, safety critical systems in particular present a barrier to new technology insertion and can result in a sustainment spiral, investing in existing technology rather than new technology (Sandborn & Myers, 2008).

Open Architecture and standardized interfaces provide the ability to upgrade a system and insert new technology over the system lifecycle (Bartels, Ermel, Sandborn, & Pecht, 2012). By using open architecture it is possible to "play and play" upgraded hardware or software, such as newer processors or memory, and accommodate new, unanticipated technology by providing a common hardware or software interface for new technology to be plugged in to an existing system.

Requirements management is the process of documenting, analyzing, tracing, prioritizing and agreeing on requirements and then controlling change and communicating to relevant stakeholders. Technology insertion and technology refresh are accomplished most effectively when requirements are written from the start to account for and require the ability to insert new technology during the system lifecycle. This may include requirements for Open Architecture and mandating system upgrades at periodic intervals.

2.2.2 Industry Applications in New Technology Insertion

These are systems or applications of technology in private industry such as manufacturing, health care and aerospace. There is significant overlap in technology uses in industry and government or military, however there are different constraints and requirements between them.

A study specific to commercial technology, although applicable to the military as well, discusses technology insertion in commercial avionics (Wilkinson, 2004). This study focused on obsolescence issues and problems in the technology insertion process, as well as previous solutions and their limitations.

As markets and requirements change, lifecycle management is a process for systematically incorporating new technology (Prasad, 1997). Herald (2000) proposes an evolutionary technology refreshment pan to leverage newer generation products. This provides a link between Systems Engineering and Supportability Engineering.

2.2.3 Lifecycle Sustainability within New Technology Insertion

A subset of design theory, lifecycle sustainability focuses on designing in sustainability over the system lifecycle. Sustainability is one of the "ilities" that are mandatory requirements for many major systems acquisitions, including DoD. Other "ilities" are reliability, maintainability and availability. These mandatory requirements help ensure the system remains viable over the expected system lifetime, and frequently well beyond the planned lifetime.

Sustainability can be achieved several ways. The "brute force" approach may include lifecycle buys of components (Feng, Singh, & Sandborn, 2007) up front resulting in high up-front costs and logistical costs and cannibalization of parts on the back end (Konoza & Sandborn, 2002) to keep fewer systems operating by taking working parts from other systems due to lack of spares.

Another approach is through technology road mapping to plan in advance the optimum refresh cycle, design in refresh planning and have a strategic vs. reactive approach (Sandborn, Herald, Houston, & Singh, 2003) which will contribute to system viability over the long term. It is important to address Diminishing Manufacturing Sources and Material Shortages (DMSMS), strategic management, and Mitigation of Obsolescence Cost Analysis (MOCA) (Sandborn P. , 2008) in the system lifecycle.

2.2.4 Obsolescence in New Technology Insertion

Technical obsolescence is not a design approach, but an inevitable result of normal technology advances and technology jumping rendering otherwise functional products, services, or systems no longer efficient or needed. It is a major consideration of every long lifecycle, mission-critical system and must be accounted for in the system requirements and system design. Obsolescence may be expected, such as improvements in hardware performance over time, or unexpected such as introduction of unanticipated advances.

Obsolescence of technology can occur when systems become unavailable before the demand for them ends (Sandborn P. , 2013). One way to address this is planning for design refresh to mitigate obsolescence (Feldman & Sandborn, 2007). Data mining and life cycle curve forecasting can also be useful (Sandborn P. , 2005). Environmental factors should be taken into consideration as well when disposing of systems (Pope, Elliott, & Turbini, 1998).

Human obsolescence must be addressed as well. Human obsolescence is a result of technical obsolescence. As technology advances, human skills can become scarce to support and maintain older technology. An example of this is the shortage of COBOL programmers to continue maintaining older COBOL systems while newer system take advantage of newer approaches such as web-based applications, SQL databases and client-server computing. One study found the average age of programmers to be 29, with a standard deviation of 7. Assuming a normal distribution, this means that 97.5% of developers are under the age of 44 (Johnson P. , 2013). A pressing problem on the horizon for many companies is a shortage of Cobol developers as the demand for Cobol has remained steady and the average age of COBOL programmers in 2014 being 55 years old (Florentine, 2014) – well above the average age of programmers in general. Human obsolescence also occurs when a person's ability to perform is degraded due to outmoded skills. It is often assumed that lost human resources can always be replenished. In many cases there is a lack of workers with the necessary skills and current workers sometimes cannot simply be retrained (Sandborn, Prabhakar, & Kusimo, 2012). There are impacts to system support due to the lack of workers with the required skill set as those skills become obsolete

(Sandborn & Prabhakar, 2015). Sandborn discusses a model for forecasting the loss of critical human skills and the impact of that loss on the future cost of system support; support, which can be substantial.

2.2.5 Government Defense Applications in New Technology Insertion

These are systems designed primarily for military applications, such as weapons and combat systems. These systems typically have a lifecycle measured in decades, a lifecycle cost in the billions of dollars, and mission-critical implications such as human life and national security missions.

One study, specific to the defense industry, describes how to take advantage of the latest technology while managing the technology insertion process (Kerr, Phaal, & Probert, 2008). The study discusses the rapid insertion of technology through a phased, or spiral approach and further discusses the process of technology management and enablers for technology insertion.

Several studies have been performed specific to the U.S. Navy. The Submarine Acoustic-Rapid Commercial-Off-the-Shelf Insertion (A-RCI) project was focused on leadership and management approaches to improve the acquisition process and resulted in substantial time and cost savings (Johnson, 2004). Another project specific to the U.S. Navy was an effort to achieve rapid technology insertion aboard U.S. Navy warships that employ the Aegis combat system (Sylvester, Konstanzer, & Rottier, 2001). Aegis is a very complex system of systems and integrating new technology typically takes several years, if not a decade.

Another aspect of rapid technology insertion is the use of science and technology roadmaps (Kostoff & Scaller, 2001). This paper explores the use of roadmaps to decisions coordinating resources and activities in environments that are complex and uncertain. It provides a taxonomy of roadmaps as well as a description of mapping techniques.

Research has been performed specific to software technology and the U.S. Navy. Service Oriented Architecture (SOA) has rapidly become a leading technology for implementing services in order to integrate software systems more effectively. One such paper examined the exchange of data between combat systems and command and control systems (Moreland, Sarkani, & Mazzuchi, 2014).

Much research has been presented in professional conferences as well. These include rapid technology insertion for DoD avionics systems (Siegel, Majernik, Davis, & Foster, 1999), rapid insertion of commercial off the shelf (COTS) hardware and use of open architecture to reduce cost and schedule (Davis, 1999), and how to accomplish rapid technology insertion for communications through the use of software defined radios (SDRs) (Cohlman & Osborn, 2005).

Additional sources of information come from public sources. The A-RCI contract modification was recently announced by the U.S. Navy Naval Sea Systems Command and is available through the Freedom of Information Act (FOIA): “Lockheed Martin Corp., Mission Systems and Training, Manassas, Virginia, is being awarded a $29,209,925 modification to previously awarded contract (N00024-11-C-6294) for the development and production of the Acoustic Rapid Commercial-Off-The-Shelf Insertion (A-RCI) and common acoustics processing for Technology Insertion 12 (TI12) through Technology Insertion 14 (TI14) for the U.S. submarine fleet and for foreign military sales. A-RCI is a sonar system that integrates and improves towed array, hull array, sphere array, and other ship sensor processing, through rapid insertion of commercial off-the-shelf-based hardware and software. This modification will purchase TI14 system upgrades for six ships including spares and pre-cable kits. The Naval Sea

Systems Command, Washington Navy Yard, Washington, District of Columbia, is the contracting activity."

The A-RCI project was further described in a report from the Naval Postgraduate School (Boudreau, 2006). It describes the use of open systems architecture and COTS to reduce cost and streamline the technology integration schedule.

Further information is available from public sources by examining the success of various DoD initiatives to streamline the acquisition process and reduce cost of acquiring and integrating new technology into existing systems. A plethora of official DoD instructions describe how new technology can be developed or procured. The overall process, the Joint Capabilities Integration Development System, is very detailed and time-consuming, frequently causing major defense acquisition programs to take upwards of a decade from program initiation to full operational capability with significant cost overruns along the way.

DoD has experimented with various ways to streamline this approach, especially for technology insertion for existing programs. One such approach is executing a rapid development capability (RDC) that eliminates much of the documentation, reporting and milestone reviews in a formal JCIDS program (US Navy, 2008). Another approach is a quick reaction assessment (QRA) that streamlines the testing process prior to deploying the new technology (US Navy, 2008). Other approaches are being tested as well, and these can provide a comparison of the formal JCIDS process to more streamlined processes.

2.2.6 Government Non-defense Applications in New Technology Insertion

These are systems designed primarily for US Government applications outside of the military. These can include systems such as the air traffic control system which has mission-critical implications, and is maintained and upgrades over decades at a cost of billions of dollars. Major government financial systems also fall in the category.

“Legacy electronic systems … and their effective system support lives may be governed by existing non-replenishable inventories of spare parts” (Konoza & Sandborn, 2013). These systems frequently depend on commercial off the shelf (COTS) components. As COTS components become obsolete, they become sustainment-dominated systems whose long-term sustainment- costs exceed their original procurement costs. An example of this is the Federal Aviation Administration (FAA) air traffic control system.

The National Oceanographic and Atmospheric Administration (NOAA) recently decided to perform a technology refresh on the Tropical Atmosphere Ocean (TAO) buoy array because “components are being discontinued or are no longer supported by the manufacturers due to the technology presently used being more than 10 years old.” (Teng, Bernard, & Lessing) TAO monitors the tropical Pacific to improve understanding of El Niño. This was an opportunity to perform a refresh while transitioning the project from the Pacific Marine Environmental Laboratory (PMEL) to the National Data Buoy Center (NDBC).

2.2 Limitations of Existing Theory and Application

Although there has been research into new technology insertion and many approaches have been tried, none appear to provide a single best solution. Singh and Sandborn (2006) observed that the Mitigation of Obsolescence Cost Analysis (MOCA) method enables early forecasting of refresh dates to allow optimum refreshes to be performed, however there are situations in which the present MOCA solution is incomplete such as the treatment of software. Another approach, Performance Based Logistics (PBL) has been used to address Diminishing

Manufacturing Sources and Material Shortages (DMSMS), however it has been noted that "products with long development processes will likely become obsolete more quickly than anticipated." (Feldman & Sandborn, 2007). A more recent trend has been availability contracts where industry delivers a complete product-service system (PSS). The challenge is that when negotiating the contract, the solution provider and the customer must be confident of the whole life costs (WLC) for 20, 30 or even 40 years into the future (Romero-Rojo, Roy, Shehab, & Wardle, 2009). As a result, government and private industry spend a lot of money, time and resources on maintaining obsolete systems rather than upgrading them. In many cases this is compounded by laws and regulations that govern the procurement, test and evaluation of new technology.

A second limitation is that new technology covers a broad spectrum. It may be new hardware such as larger capacity hard drives and storage that have a new interface (e.g., SATA), new communications capability such as new higher speed or wireless, or new software standards such as JAVA. This diversity makes it difficult to devise a consistent process for integrating new technology. Finding more effective ways of incorporating such technology into legacy systems, especially those that are mission critical, could result in significant performance improvements and cost reduction while reducing risk of failure.

CHAPTER 3

METHODOLOGY

3.1 Research Design – Ontology Types and Methodologies

In general, a set of concepts, the taxonomic hierarchical relationships among the concepts, and axiomatic first order logic to specify the logical relationships are the minimum components of a formal ontology. Other components necessary to operationalize the ontology include a glossary of terms, concept dictionary, and rules specific to the knowledge domain (Gómez-Pérez, Fernandez-Lopez, & Corcho, 2004) (Gómez-Pérez, et. al., 2004, pp. 130-142). Roussey, et. al., (2011) classify ontologies based on language expressivity and formality.

- A *formal* ontology requires clear semantics based on and strict rules defining the concepts and relationships and formal first order logic to define the distinctions between concepts. Examples of formal ontologies include OWL (Web Ontology Language) and CoBra (intelligent agent computing environments), and knowledge bases.
- *Software* ontologies specify data manipulation and storage schemas to achieve data consistency. Examples include the Unified Modeling Language (UML), Industry Foundation Clauses (IFC), and domain and local application knowledge ontologies.
- *Linguistic* or *terminological* ontologies such as dictionaries, glossaries, thesauri, and lexical databases. Examples include the Agrovoc, GEMET, HEREIN, and URBAMET thesauri, General Ontology for Linguistic Description (GOLD), Resource Description Framework (RDF), and Simple Knowledge Organization System (SKOS).

- *Information* ontologies are composed of relational diagrams organizing relationships among concepts and instances. Examples include the Information Artifact Ontology, Information Ontology of Architectural Design, Information Ontology of Construction Project, and Mind Map

Rousey, et. al., provide a second ontology classification based on scope and domain granularity. Proceeding from the broadest scope and least granularity to the narrowest scope and highest granularity yields the following classifications.

- A *top-level foundational ontology* is a generic ontology that provides taxonomic and axiomatic scope structure for a general body of knowledge. It provides the taxonomic and axiomatic basis for underlying core reference ontologies and domain ontologies. In this research, an example would be a design theory ontology. Foundational ontologies are designed and constructed using a top-down approach and general methodologies such as BFO, Cyc, DOLCE, GFO, PROTON, and SUMO (Mascardi & Paolo, 2007).
- A *core reference ontology* provides the generic taxonomical and axiomatic scope structure for a sub-discipline within a body of knowledge by integrating differing domain viewpoints. In this research, examples would include computer design theory, electrical design theory, mechanical design theory, new technology and sustainability design theory, social design theory, software design theory, visual arts design theory, etc. Core reference ontologies are designed and constructed using a top-down approach with reference to its foundational ontology using a general methodology such as SENSUS (Jones, Bench-Capon, & Visser, 1998).

- A *domain ontology* provides the specific taxonomical and axiomatic structure necessary to organize knowledge about a phenomenon or methodology within a sub-discipline. Examples include reciprocating engine design within mechanical design theory, memory design within computer design theory, and organizational design within social design theory. Domain ontologies are designed and constructed using a middle-out approach with reference to its core reference ontology using a general methodology such as SENSUS.
- An a*pplication* or *local* ontology provides the specific taxonomical and axiomatic structure necessary to organize specific competency knowledge about a particular phenomenon within a domain. Examples include design knowledge specific to a V-6 automobile engine or two-stroke boat engine or rotary aircraft engine within the reciprocating engine domain. Application ontologies are designed and constructed using a bottom-up approach with reference to its domain ontology using a specific methodology such as CommonKADS, DILIGENT, Enterprise Model Approach, KACTUS, KBSI IDEF5, METHONTOLOGY, or TOVE (Corcho, Fernandez-Lopez, & Gomez-Perez, 2003) (Cristani & Cuel, 2005).
- A *task* ontology provides the specific taxonomical and axiomatic structure necessary to organize specific or expert knowledge about a particular method or process necessary to produce a particular phenomenon within an application or local knowledge. An example would be the process steps necessary to build a V-6 automobile engine or two-stroke boat engine or rotary aircraft engine within the reciprocating engine domain.

Chandrasekaran and Josephson (1997) note that “Knowledge systems need to have two kinds of knowledge:

1. Knowledge about the objective realities in the domain of interest (Objects, relations, events, states, etc. that obtain in some domain)
2. Knowledge about problem solving.”

Rousey, et. al.’s hierarchical ontology classification above provides for the first case of specifying objective realities. On the other hand, Chandrasekaran and Josephson note that problem solving can entail the logical reasoning method (deductive, inductive, or abductive) and the specific reasoning process (deductive proof applied, inductive hypothesis test or model applied, or abductive variation and consequences method applied). Thus, a *methods* ontology specifies the problem-solving vocabulary and constructs necessary for the human or artificial intelligence problem solver to manipulate the concept’s state vector to describe the problem-solving goals and sub-goals and identify the problem-solving tasks to be applied toward attaining the stated goals or sub-goals. For systems mission accomplishment as a general design goal and new technology insertion as a specific sub-goal, the methods ontology would specify the systemic mission outcomes, candidate system designs, partial systems design solutions, the decision method(s) applicable to selecting the optimal partial solution among competing partial solutions, and the test method(s) and criteria to be applied. For the new technology insertion sub-goal, the methods ontology would specify the forecasted technology roadmap matrix, obsolescence matrix, partial interface solution candidates, the decision method(s) applicable to selecting the partial solution candidates that maintain minimum mission accomplishment, and the interface test method(s) and criteria to be applied.

In parallel to the above ontological hierarchical structure are *general* ontologies that provide taxonomical and axiomatic scope structure for a general knowledge such as language, written word, mathematics, or general science and *indexing ontologies* that guide knowledge selection. Figure 1 extends Rousey, et. al.'s hierarchical ontology classification to summarize the minimally sufficient ontological hierarchical structure necessary for providing a complete specification of a concept's body of knowledge.

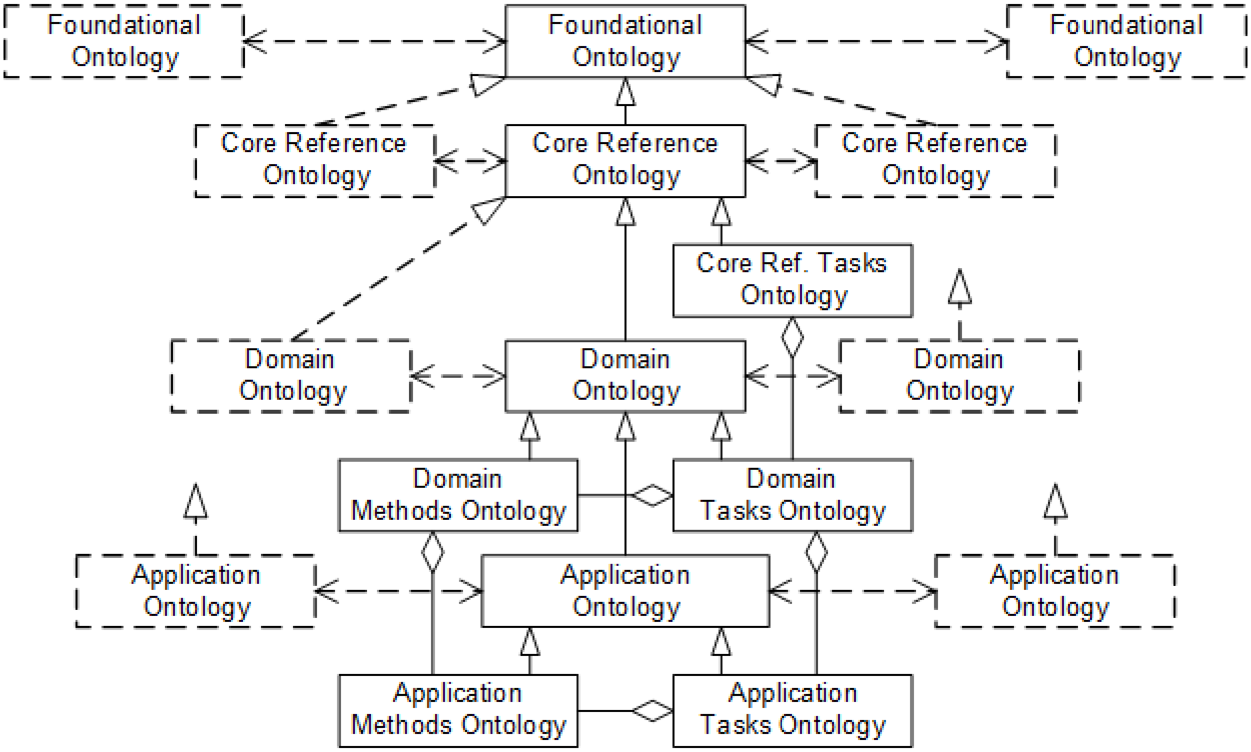


Figure 1. Concept Body of Knowledge Hierarchical Ontology System.

Obrst (2010) argues that an ontology architecture may need to be layered within levels in order to represent primitive ontology structures accurately. Rector (2003) specifies a primitive ontology as one that contains only primitive taxonomic concepts and their supporting primitive axioms as *necessary* conditions for existence. Rector argues that "If each primitive belongs explicitly to one specific module (taxonomy), then the (axiomatic) links between modules can be made explicit …" defined concepts are specified "… by *necessary and sufficient* conditions." The concept of a primitive ontology arises in quantum mechanics and was by proposed Durr, Goldstein, and Zanghì (1992) and Goldstein (1998). In quantum mechanics, a primitive ontology contains entities in three-dimensional space or four-dimensional space-time and are the fundamental building blocks of all other entities. The historic traces of primitive ontology entities through time provide a dynamic theory of the universe. Formalism of the dynamic theory contains the primitive entities and nonprimitive variables necessary to mathematically describe how the primitive entities dynamically evolve in time. It is the theoretical integration of primitive entities and nonprimitive variables that provides all the macroscopic properties to necessary and sufficient to explain the physical universe. Using the concept of layered primitive ontology architectures, Figure 2 expands on Figure 1 illustrating necessary and sufficient conditions to explain universal concepts.

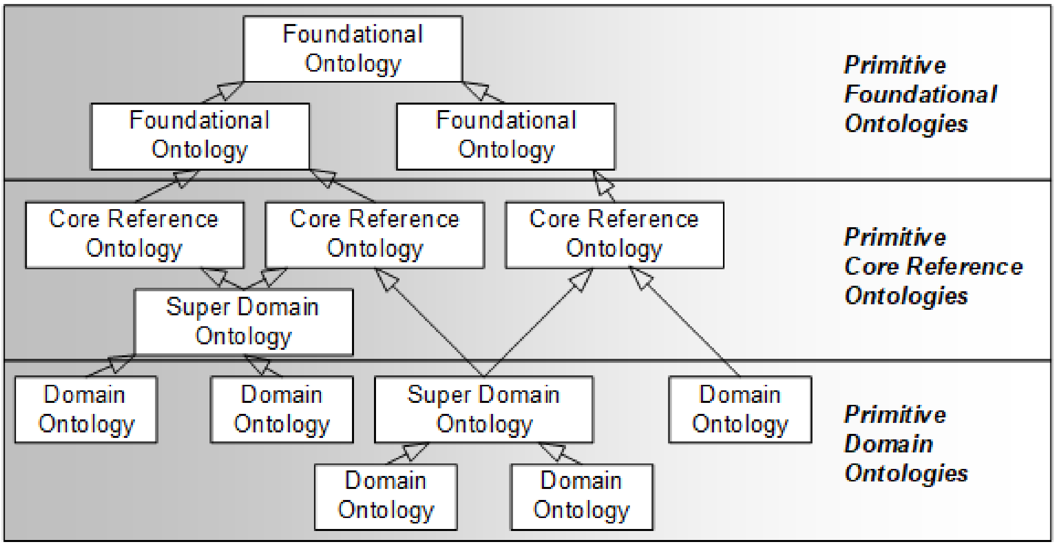


Figure 2. Layered Hierarchical Primitive Ontology Architecture.

3.2 Corpus Population and Criteria for Admittance

For this research, only peer reviewed articles obtained from professional design societies journals through searches on Google Scholar or the Old Dominion University Perry Library and United States Department of Defense (DoD) and Government procurement specifications were considered as being of sufficient quality for inclusion in the corpus. Based on the selected research domain, initially only articles directly related to design theory and new technology insertion were admitted. From this initial definition, this research applied Grounded Theory's (Glaser, 1965; Glaser and Strauss, 1967) constant comparative method to arrive at an initial classification of the articles by the primary emergent themes with secondary emergent themes noted for later development of potential axiomatic themes. Themes were identified by equally weighting the information each article's title, abstract, key words, introduction, problem statement or research question, and results and conclusions. Information in each article's background or literature review and the research method was not considered because of the potential for these to reflect the authors epistemological orientation. By filtering epistemological orientations, this research sought to maintain the etic perspective in the constant comparative method with the goal of arriving at epistemological free, overarching design and new technology insertion categories. Epistemological free categorical themes are necessary to develop a general theory of new technology insertion design applicable across all design domains rather that just restricted to the long life-cycle, complex, mission critical systems domain of focus. All admitted articles were in Adobe PDF and were converted to plain text documents for text mining.

Article searches for each category were terminated upon reaching thematic saturation using the Power Law as recommended by Guest and Johnson (2006) for Grounded Theory open coding. Guest and Johnson's Power Law approach formalizes Bowen's (2008) definition of saturation as occurring when the researcher gathers thematic information to the point of diminishing returns. At the point of diminishing returns, no new thematic categories emerge as new articles are reviewed and variability between categories are explained. Saturation analysis is presented in section 4.1.

3.3 New Technology Insertion Ontology Methodology

Since it was constructed as a core reference ontology, the general strategy for building the ontology for new technology insertion and sustainability design was to integrate text mining and content analysis within the Grounded Theory framework as the logical basis for identifying seed terms (primitive concepts) and path interrelationships within the SENSUS ontology method. The outcome objectives of this strategy were a human understandable theoretical basis for the ontology from Grounded Theory and a machine readable hierarchical taxonomic logic shareable across design domains. The general ontology creation approach was as follows.

SENSUS Process 1: Identify seed (primitive concept) terms.

Text mining.

1. Perform a structured search of the general design, new technology upgrade, and new technology insertion literature { on Google Scholar, professional design societies journals, and ODU Perry Library }.
2. Build a corpus of new technology upgrade and insertion journal articles. Apply Grounded Theory open coding as an initial organizing criterion for the corpus.
3. Perform text mining to identify common word associations and correlations to suggest initial seed (primitive concept) categories.

The key outputs from the text mining step were a linguistic taxonomy and a resultant new technology insertion and sustainability dictionary.

Grounded Theory – perform Grounded Theory open coding relative to identify seed (primitive concept) categories for the body of knowledge.

SENSUS Process 2: Link text mining and Grounded Theory open coding seed categories to root ontological (taxonomic) seed themes. Resolve differences between the text mining categories and Grounded Theory open coding categories to identify common seed theme (primitive concept) categories.

SENSUS Process 3: Identify and add logical paths from the seed (primitive concept) categories to the common seed theme.

SENSUS Process 4: For each seed (primitive concept) category, identify hierarchical branches and leaves (primitive taxonomic structure) and cross-paths among branches within each seed category (primitive hierarchical axiomatic relationships).

Content analysis – identify concept associations and correlations as the basis for axial coding for Grounded Theory analysis.

Grounded Theory – axial and selective coding to specify the taxonomical theoretical constructs as the knowledge basis for the design of new technology upgrade and insertion.

Descriptive Logic – test and organize theoretical new technology insertion and sustainability into conceptual categories with formal semantic rules defining relationships.

The output of process 4 is the technology design and technology insertion design core reference taxonomies to be used at the foundation for organizing the domain body of knowledge and initiate formal applied research into and development of optimum new technology insertion and sustainability methodologies.

3.4 New Technology Insertion Ontology Verification

The developed new technology insertion ontology was coded into Fluent Editor using controlled natural language. During encoding, concept classes and attributes definitions were verified using Fluent Editor's Validate RL+ for consistency with the World Wide Web Consortium (W3C®) Web Ontology Language OWL2 semantic profiles.

In the second verification step, Gomez-Perez's (1996, 1999, 2001, 2004) method for evaluating and verifying taxonomies and ontologies against Gruber's (1995) ontological design criteria of clarity, coherency, extendibility, minimal encoding bias, and minimal ontological commitment was applied. Gomez-Perez's method evaluates for:

- Inconsistency errors
    - *Circularity errors* result from a concept being defined as a semantic specialization or generalization of itself. Taxonomic circularity errors are tested by the distance criteria. No circularity exists at a distance 0, that is the concept is a unique concept. Circularity errors of distance 1 … n means that a concept has a semantically equivalent definition in subclass 1 … n.
    - *Partition errors* result from disjoint decompositions.
        - *Common classes in disjoint decompositions* occur when there is a partition of a concept class A $\{a_1, a_2, \ldots, a_n\}$ into class A $\{a_1, a_2, \ldots, a_i\}$ and class B $\{a_j, a_k, \ldots, a_n\}$.

        - *Common instances in disjoint decompositions* occur when several instances belong to more than one class of a disjoint decomposition.
        - *External instances in exhaustive decompositions* occur when there is an exhaustive decomposition of all concept classes and some instances of a class A $\{a_j, a_k, \ldots, a_n\}$ do not belong to any class.
    - *Semantic or instance errors* result from an incorrect semantic or instance classification.
- *Incomplete errors* result from the over-specification or imprecise specification of a concept class.
    - *Incomplete concept classification* results from an incomplete decomposition of the knowledge in a concept class.
    - *Partition errors* result when disjoint and exhaustive knowledge among classes is incompletely defined.
        - *Disjoint knowledge omission* occurs when a set of subclasses are omitted in the taxonomy.
        - Exhaustive knowledge omission occurs when a class is decomposed into two or more subclasses that carry the same knowledge.
- *Redundancy errors* occur in a taxonomy when there is more than one axiomatic hierarchical definition of a subclass relationship or there exists more than two classes or instances with the same formal definition.
    - Redundancies of Subclass-Of relations.
    - Redundancies of Instance-Of relations.
    - Identical formal definitions of two or more classes.

- Identical formal definitions of two or more instances.

The output of applying Gomez-Perez's criteria is verification that the resultant formal new technology insertion and sustainability taxonomy meets the design intents of maximally separated conceptual categories organized through logical relationships defined by axiomatic logical formulas.

The third verification step tested for a proper ontology structure by applying Guarino and Welty's (2000) and Welty and Guarino's (2001) subsumption criteria for concept "is-a" attributes and Rector's (2003) criteria for hierarchical "is-kind-of" attribute relationships. Welty and Guarino specify that for arbitrary properties (attributes), the statement "ψ subsumes ϕ, to mean that, necessarily:"

$$\forall x\ \phi(x) \rightarrow \psi(x) \qquad (1)$$

In their focus on concept attributes subsumption, they note, "Where for example description logics can determine whether one (complex) *description does* subsume another, this methodology can help determine whether or not a primitive property can subsume another" (Welty and Guarino, 2001; p. 53). Welty and Guarino develop "is-a" attribute proper subsumption on the philosophical concepts of *rigidity*, *identity*, *unity*, and *dependence*. Refer to Guarino and Welty (2000) and Welty and Guarino (2001) for the arguments linking these philosophical concepts to "is-a" attribute proper subsumption. Rather, for the purpose of being succinct, this work quotes Guarino and Welty's "is-a" attribute proper subsumption definitions in a list-like presentation.

Rigidity depends on the concept of essentiality. Welty and Guarino (2001, p. 57) define three levels of rigidity:

Definition 1: A *rigid property* is a property that is essential to *all* its (concept's) instances, i.e., a property ϕ: $\Box(\forall xt\ \phi(x, t) \rightarrow \Box\forall t'\ \phi(x, t'))$.

Definition 2: A *non-rigid property* is a property that is not essential to *some* of its (concept's) instances, i.e., a property ϕ: $\Diamond(\exists x, t\ \phi(x, t) \cap \Diamond\ (\exists t' \neg \phi(x, t'))$.

Definition 3: An *anti-rigid property* is a property that is not essential to *all* its (concept's) instances, i.e., a property ϕ: $\Box(\forall xt\ \phi(x, t) \rightarrow \Diamond(\exists t' \neg \phi(x, t'))$.

where $\Box\phi$ means necessarily true in all possible worlds and $\Diamond\phi$ means possibly true in at least one possible world. Rigid properties are designated with +R, non-rigid properties with -R, and anti-rigid properties with ~R.

Welty and Guarino (2011, pp. 58-59) refer the philosophical concept of *identity* as ability to distinguish a specific instance of a concept class from other instances of the same class by means of at least one of its characteristic properties. Welty and Guarino (2011, pp. 58-59) define "… an *identity condition* (*IC*) for an arbitrary attribute property ϕ …as a suitable relation ρ satisfying:

$$\phi(x) \cap \phi(y) \rightarrow (\rho(x, y) \leftrightarrow x = y) \qquad (2)$$

This definition admits the following definitions of identity:

Definition 4: An IC is a *sameness* formula Σ that satisfies either of the following conditions assuming the predicate E for actual existence.

$$\Box(E(x, t) \cap \phi(x, t) \cap E(y, t') \cap \phi(y, t') \cap x = y \rightarrow \Sigma(x, y, t, t') \qquad (3)$$

$$\Box(E(x, t) \cap \phi(x, t) \cap E(y, t') \cap \phi(y, t') \cap \Sigma(x, y, t, t') \rightarrow x = y) \qquad (4)$$

Definition 5: Any property *carries* an IC iff it is subsumed by a property supplying this IC, including the case where it supplies the IC itself. This property is marked as +I attribute.

Definition 6: A property ϕ *supplies* and IC iff (i) it is rigid, (ii) there is an IC for it, and (iii) the same IC is not carried by *all* the properties subsuming ϕ. Therefore, +O attribute.

Definition 7: Any property carrying and IC is called a *sortal*.

A property carrying an IC is designated as +I (–I otherwise), and any property supplying an IC is designated as +O (–O otherwise).

Conversely, Welty and Guarino (2011, p. 55) note that unity is "… the problem of distinguishing the *parts* of an instance from the rest of the world by means of a *unifying relation* that binds the parts, and only the parts together." Based on this concept, Welty and Guarino (2011, pp. 59-60) define unity as:

Definition 8: An object *x is a whole under* ω iff ω is a relation such that all the members of a certain division *x* are linked by ω, and nothing else is linked by ω.

Definition 9: A property ϕ *carries a unity condition* (UC) iff there exists a single relation ω such that each instance of ϕ is *necessarily* a whole under ω.

Definition 10: A property has *anti-unity* if every instance of the property is not necessarily a whole.

Welty and Guarino recognize three types of unity– (1) *Topological* based on a physical relationship; (2) *Morphological* based on some combination of topological unity and shape; and (3) *Functional* based on functional purpose. Any attribute property carrying an UC is designated as +U (–U otherwise). Any attribute property that has anti-unity is designated as ~U, but ~U implies –U.

Welty and Guarino (2011) distinguish between *intrinsic* and *extrinsic* properties based on whether they depend on the properties of their own concept entities and instances or the properties of other concept entities and instances. An intrinsic property is inherent to the concept entity or instance, whereas an extrinsic property is at least partially dependent on the properties of other concept entities or instances. Welty and Guarino (2011, p. 60) define dependence as:

> ***Definition 11:*** A property ϕ is externally dependent on a property ψ if, for all its instances x, necessarily some instances of ψ must exist, which is neither a part nor a constituent of x:

$$\forall x \, \Box(f(x) \rightarrow \exists y \, \psi(y) \cap \neg P(y, x) \cap \neg C(y, x)) \qquad (5)$$

An externally dependent attribute property is designated as +D (–D otherwise).

At the core reference ontology level, Welty and Guarino define a proper taxonomy as one that possess the following combinations of *rigidity*, *identity*, *unity*, and *dependence.*

Table 1. Core Reference Ontological Property Kinds.

	Property Combination			
Meta-Property	Rigidity	Identity	Unity	Dependence
Category	+R	+O, -I	+U	+D
				-D
Role	~R	+O, -I	+U	+D
Attribute	~R	+O, -I	+U	-D
	-R			+D
				-D

To assure a primitive taxonomy, Rector (2003) adds the criteria of *modularity* and *explicitness* to Guarino and Welty's criteria for a proper taxonomy. Rector proposes a two-step normalization. First, assure a proper ontology relative to Welty and Guarino's criteria. Second, normalize the ontology to assure a primitive architecture. Rector defines a primitive taxonomy as one that has *"... independent disjoint skeleton ... restricted by simple trees"* (p. 1). The essence of Rector's normalization proposal is that a primitive ontology "… should consist of disjoin homogeneous trees" (p. 2).

- Each concept can have one and only one primitive parent.
- Each categorical branch of a primitive ontology must be logical and homogeneous.
- Each primitive ontology must clearly distinguish self-standing concepts and explicit partitioning among self-standing concepts.
- Subsumption of each primitive concept by one and only one other primitive concept.

To normalize a proper ontological taxonomy, Rector proposes applying relational database normal forms.

- First Normal Form (1NF): Eliminate repeating duplicate groups of data [concepts] to guarantee Atomicity (data [concept attributes] that are self-contained and independent).
- Second Normal Form (2NF): Every row of data [instance] in a 1NF table [primitive ontology] must be unique and depend only on the table's whole key [the concept's attributes].
- Third Normal Form (3NF): A table [primitive ontology] must be in 2NF and no column data in any row [sub-concept] can have any dependency [equivalent attributes] on any other non-key column [sub-concept] (i.e., data in one column

cannot be derived from the data in any other column [sub-concept attributes in one hierarchical branch cannot be derived from another sub-concept hierarchical branch]).

- Boyce-Codd Normal Form (BC-NF):
  - All candidate keys are composite keys [all composite concepts are derivable only from independent parent concepts or other composite concepts themselves derived ultimately from independent parent concepts].
  - There is more than one candidate key [composite concept].
  - The candidate keys [composite concepts] each have at least one column [concept] that is in common with another candidate key [concept].
- Fourth Normal Form (4NF): No data column [sub-concept] may depend on another column [sub-concept] other than a primary key column and depends on the whole primary key [class concept or composite concept].
- Fifth Normal Form (5NF): A table [proper ontology] must be in 4NF and if a table is decomposed further to eliminate redundancy and anomaly, when the decomposed tables [primitive ontologies] re-joined by means of candidate keys [concepts], the original data [concept attributes] may not be lost or any new records [concept attributes] must not arise.

In assuring a primitive ontological architecture, Rector’s goals are ontology re-use, maintainability, and evolution; however, development of a hierarchical primitive ontological architecture (foundational, core reference, domain, and application) also assures meeting Gruber’s criteria of clarity, coherency, extendibility, minimal encoding bias, and minimal ontological commitment.

Rector noted a number of issues to be addressed in transforming a proper ontology to a primitive ontology.

- The notion of a "primitive concept" and "primitive sub-concepts" hierarchically dependent on only their respective primitive parent concept can be difficult to demonstrate.
- Whether or not a concept should be part of a primitive ontology might be better expressed by metaknowledge, however, not all ontology languages permit reasoning over metaknowledge. Rector advocates that the criterion for concept normalization include specifications of "self-standing" and "partitioning" concepts.
- The notions of ontology normalization and ontology views are not established in ontology theory. Rector advocates is provision for concept axes to demonstrate separation.
- Provide concept indexing pointers. If an ontology is modular, the same information will point to only one primitive branch. Under this approach, concept lattices inferred from normalized and well modularized ontologies will be complete and closed under Formal Concept Analysis.

Formal Concept Analysis has long been applied in knowledge discovery (Poelmans, Elzinga, & Dedene, 2010) and knowledge processing (Poelmans, Ignatov, Kuznetsov, & Dedene, 2013). The Complete Lattice definition, Closure Operator definition, and Basic Theorem of Concept Lattices (Ganter and Wille, 1999) are necessary and sufficient to demonstrate the formalism of modular tree graphs (primitive ontology branches) within concept lattices.

Complete Lattice Definition: An ordered set $V := (V, \leq)$ is a **lattice** if for any two elements x and y in V the supremum $x \vee y$ and the infimum $x \wedge y$ always exist. V is called a **complete lattice** if the supremum $\vee X$ and the infimum $\wedge X$ exist for any subset of X of V. Every complete lattice V has a largest element $\vee V$ called the **unit element** of the lattice, denoted by $\mathbf{1}\vee$. Dually, the smallest element $\mathbf{0}\vee$ is called the **zero element** (Ganter and Wille, 1999; p. 5).

***Closure Operator Definition*:** A closure operator φ on G is a map assigning a closure $\varphi X \subseteq G$ to each subset $X \subseteq G$ under the following conditions:

(1) $X \subseteq Y \Rightarrow \varphi X \subseteq \varphi Y$, monotony.

(2) $X \subseteq \varphi X$, extensity.

(3) $\varphi\varphi X = \varphi X$, idempotency.

Closure Theorem: If $\mathcal{U}$ is a closure system on G then

$$\varphi_{U} X := \cap \{A \in \mathcal{U} \mid X \subseteq A\} \tag{6}$$

defines a closure operator on G. Conversely, the set

$$\mathcal{U}_{\varphi} := \{ \varphi X \mid X \subseteq G\} \tag{7}$$

of all closures of a closure operator φ is always a closure system, and

$$\varphi\mathcal{U}_{\varphi} = \varphi \qquad \text{and} \qquad \mathcal{U}\varphi_{u} = \mathcal{U} \tag{8}$$

Proof provided by Ganter and Wille (1999, p. 8).

***Basic Theorem on Concept Lattice*s**: The concept lattice $\mathcal{B}(O$ objects, A attributes, I relations) is a complete concept lattice in which infimum and supremum are given by:

$$\wedge_{t \in T} (O_t, A_t) = (\cap O_t, (\cup A_t)'') \tag{9}$$

$$\wedge_{t \in T} (O_t, A_t) = ((\cup O_t)'', \cap A_t) \tag{10}$$

A complete lattice V is isomorphic to $\mathcal{B}(O, A, I)$ if and only if there are mappings $\gamma : O \rightarrow V$ and $\mu : A \rightarrow V$ such that $\gamma(O)$ is supremum-dense in V, $\mu(A)$ is infimum-dense in V, and oIa is equivalent to $\gamma o \leq \mu a$ for all $o \in O$ and all $a \in A$.

Proof provided by Ganter and Wille (1999, pp. 20-22).

Algebraic decomposition of closed and complete concept lattices provides the means for identifying modular tree graphs (primitive ontology branches) within concept lattices. This work adapts the definitions cohesion and coupling from software engineering (Lindig and Snelting, 1997) to define modular primitive concepts.

Modular Concept Object Definition: A modular concept object (*MCO*) consists of a set of set of objects $o \subseteq O$ and a set of attributes $a \subseteq A$ such that $\forall a \in A, o \in O: (o, a) \in V \Rightarrow a \in A$ and $\forall o \in O, a \in A: (o, a) \in V \Rightarrow o \in O$, where the MCO $\subseteq O \times A$.

Thus, in a modular concept object, all objects O have only attributes A, and all attributes A only describe objects O.

In order to map a modular concept object to Rector's proper ontology normal forms, we need to define concept of cohesion. Cohesion indicates the strength of relationship among modular objects O in an MCO via shared attributes A.

Cohesion Definition: A MCO (o, a) has *maximal cohesion* if $\forall o \in O, a \in A : (o, a) \in \mathrm{V}$.

A MCO $((o, \bar{o}), (\bar{a}, o))$ has *normal cohesion* if $\exists\, \bar{o} \in O\ \forall a \in A : (\bar{o}, a) \in V$ and $\exists\, \bar{a} \in A\ \forall o \in O : (o, \bar{a}) \in V$.

Maximal cohesion means that two or more concept objects within an MCO are described by the same attributes. Conversely, two sets of attributes maximally interfere if they describe the same concept objects. Normal cohesion means that concept objects in an MCO are not described by exactly the same attributes (each concept object is described by at least one attribute not used by the other objects in the MCO).

Coupling indicates the strength of relationship among modular concept objects via shared objects O and attributes A.

Coupling Definition 1: Let $O_1 \in MCO_1$ and $O_2 \in MCO_2$ be two modular concept objects and let $a \in A$ be an attribute. MCO_1 and MCO_2 be are coupled via a, iff $a \in O_1 \cap O_2$.

Coupling Definition 2: Let $A_1, A_2 \in A$ be two sets of disjoint attributes, and let $o \in O$ be an object. Then $A_{1,2}$ interfere via o, iff $o \in A_1 \cap A_2$.

Coupling definition 1 states that two conceptual objects are coupled if they require the same global attribute (or some intersection of global attributes) to define their respective existence. Similarly, two sets of attributes interfere if they are used to define the existence of the same conceptual object.

The Complete Lattice and Closure Operator definitions, Basic Theorem of Concept Lattices, cohesion and coupling definitions can be combined with tree structures from graph theory to specify the properties of a proper, normalized primitive ontology.

Basic Tree Theorem: Let T be a graph G with n vertices. Then, T has the following properties:

(i) T is a *tree*;

(ii) T contains no cycles and has $n - 1$ edges;

(iii) T is connected and has $n - 1$ edges;

(iv) T is connected and each edge is a bridge;

(v) Any two vertices of T are connected by exactly one path; and

(vi) *T* contains no cycles, but the addition of any new edge creates exactly one cycle (proofs provide by Wilson, 1996, p. 44).

A forest is a collection of connected trees that itself forms a tree with no cycles.

Forest Corollary: If *G* is a forest with n vertices and *k* components, then *G* has $n - k$ edges (Wilson, 1996, p.44).

Spanning Forest Theorem: If *T* is any spanning forest of a graph G, then

(i) Each cutset of G has an edge in common with T; and

(ii) Each cycle of G has an edge in common with the complement of T (proofs provide by Wilson, 1996, p. 45).

Now Rector's notion of a primitive ontology as being one that contains only primitive taxonomic concepts and their supporting primitive axioms as *necessary* conditions for existence can be formalized.

Primitive Ontology Definition: A primitive ontology is a complete and closed basic modular concept object lattice with regular cohesion among the attribute sets of concept object trees and maximal cohesion of the set of attributes defining the concept set.

Under the assumption of maximal cohesion within only concept object sets, each MCO(O, A) cross table corresponds to maximal primitive ontology rectangles in attributes as shown in Figure 3(a). Absence of couplings or interferences of attributes among concept leads to a pure, modular primitive ontological tree structure as shown in Figure 3(b). Primitive ontologies are represented graphically by lattice trees with a single root concept object for each tree and the concept object uniquely described a set of attributes of at least regular cohesion (illustrated in Figure 3(a)).

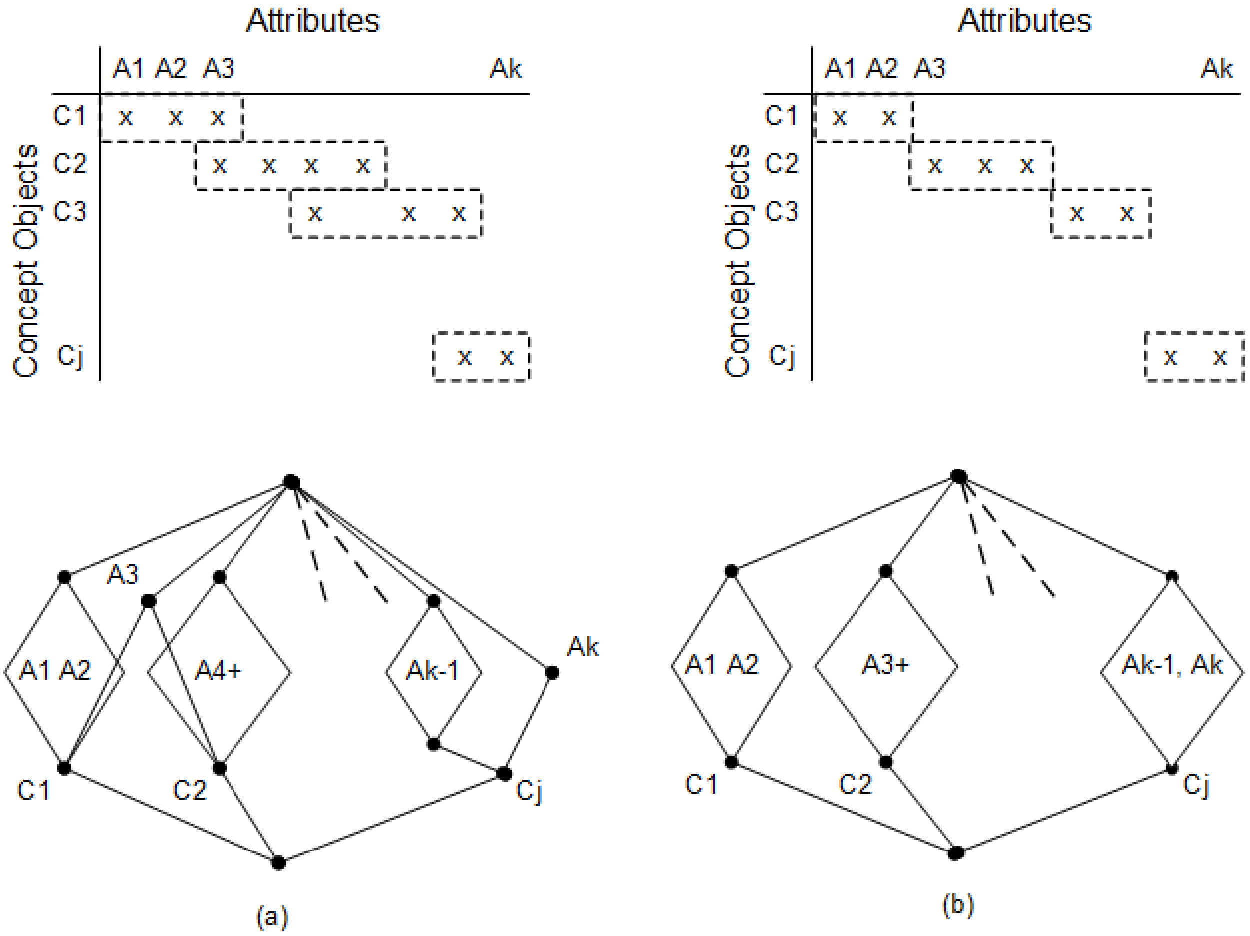


Figure 3. Primitive Ontology Cross Table and Lattice Structures.

Layered hierarchical primitive ontology subclasses must themselves form primitive ontologies with a single root concept object for each tree and the concept object uniquely described by a set of sub-attributes of at least regular cohesion of higher resolution semantics within the span of the parent attribute semantics.

Layered Hierarchical Primitive Ontology Definition: Each sub-class concept object $Oij \subseteq Oi\bullet$ in a layered hierarchical primitive ontology must itself be the root of a tree with attribute semantics $Oij(Aij) \subseteq Oi\bullet(Ai) \in$ MCOi• of at least inherited regular cohesion. In the cross table, *interference*($Oij(Aij)$) $\supseteq$ *interference* ($Oi\bullet(Ai)$).

CHAPTER 4

RESULTS

4.1 Corpus Open Coding Taxonomic Classes

Based on the selected research domain, initially only articles directly related to design theory and new technology insertion were admitted. Constant comparative analysis was performed using a modified approach to Boeije's (2002) method.

1. Comparison of each article's problem statement, research question, key words, and conceptual intent to identify primary and secondary themes.
2. Comparison of each article's primary and secondary themes to existing corpora primary and secondary themes.
3. Contrast each article's primary and secondary themes to existing corpora primary and secondary themes.
4. Assign the article to a candidate corpora category with which its primary and secondary themes most closely align.
5. As articles are assigned to corpora categories:
    a. Perform within corpora category cohesion checks of themes. If theme dissimilarity arises within a category, continue to steps b and c, otherwise continue with existing categories.
    b. Perform pairwise checks between category themes to identify emerging thematic overlaps or partitions.
    c. Reassign categories to maintain maximum theme cohesion within categories and dissimilarity among categories.

As the search progressed, 206 articles were admitted to the corpus with the open coding general technology design primary and secondary themes related to new technology insertion emerged as shown in Table 2. Author assigned key words were reduced to single key words to eliminate redundancy and instance-specific key words. As an example of redundancy reduction, one article's key words "design process, design knowledge, design research, engineering, design, knowledge reuse" were reduced to "design, engineering, knowledge, process, research, reuse." As an example of instance-specific reduction, one electronic assembly article's key words "lead-free electronics, repair simulation, RoHS, AHP, cost, availability" were reduced to non-specific key words "availability, cost, repair, simulation." Complete article assignment is presented in Appendix A.

Table 2. Corpus Primary and Secondary Themes.

Primary Theme	**Number Articles**	**Secondary Theme**	**Number Articles**	**Key Words**
Design Theory	16	Availability	1	availability, cost, simulation
		Complexity	4	adaptive, design, field, information, infrastructure, interaction, scaling, systems, technology, theory
		Combinatorial	1	combinatorics, design, theory
		Environmental	1	design, development, engineering, environment, product
		Knowledge	1	design, knowledge, process, research
		Models	2	availability, cost, management, model, repair, reuse, simulation
		Obsolescence	1	forecasting, obsolescence
		Participatory	1	critical, design, participatory, system, theory
		Product	1	design, environmental, product, realization, systems
		Quality	1	design, quality
		Reliability	2	analysis, cost, distribution, forecasting, optimize, reliability, simulation, warranty

Table 2. Corpus Primary and Secondary Themes (continued).

Primary Theme	**Number Articles**	**Secondary Theme**	**Number Articles**	**Key Words**
Design Ontology	29	Architecture	1	learning, ontology
		Cost	1	cost, enterprise, model
		Design	18	analysis, collaborative, computer-aided, concept, configuration, criteria, design, engineering, evaluation, functionality, genetic, informatics, knowledge, management, model, methodology, obsolescence, ontology, optimization, product, representation, requirement, support, system, taxonomy, theory, workflow
		Economics	1	buy, lifetime, obsolescence, taxonomy, warranty
		Models	4	analysis, artifacts, design, engineering, interoperability, knowledge, model, ontology, product
		Methodology	2	analysis, concept, computing, design, knowledge, model, ontology, product, semantic, web
		Obsolescence	1	design, forecast, lifecycle, obsolescence, ontology
		Requirements	1	development, learning, requirements, product, ontology, system
Lifecycle	9	Economics	7	cost, disruption, lifecycle, management, optimizing, ownership, part, product, reliability, supply, strategy, warranty
		Management	2	lifecycle, maintenance, management, product, strategy

Table 2. Corpus Primary and Secondary Themes (continued).

Primary Theme	Number Articles	Secondary Theme	Number Articles	Key Words
Obsolescence	31	Cost	1	components, cost, obsolescence
		Design	1	cost, design, forecast, obsolescence, optimization
		Management	24	acquisition, complexity, components, computer, cost, COTS, criteria, data, decision, design, diminishing, economic, forecasting, function, hardware, human, insertion, lifecycle, management, mining, mitigate, model, obsolescence, optimization, planning, process, refresh, risk, skills, software, strategy, sustainment, system, technical, technology, usage
		Planning	2	design, development, environment, lifecycle, obsolescence, planning, product, recycle, remanufacture, reuse
		Skills	1	economic, psychological, obsolescence, skills
		Sustainability	2	cost, lifecycle, management, obsolescence, product, sustainment, system
Open Architecture	3	Framework	1	architecture, meta-model
		Planning	2	architecture, cost, COTS, economics, insertion, network, open, technology
Requirements Management	8	Acquisition	1	acquisition, defense, system
		Automation	1	automation, concurrent, design, engineering, management, requirements, systems
		Contracting	1	disaster, maintenance, mission-critical, recovery, service, support, systems
		Engineering	1	engineering, goal, monitoring, requirements, validation, verification
		Metrics	1	cost, design, yield
		Ontology	1	analysis, iso-geometric, ontology, locally-refined-splines
		Optimize	1	algorithms, analysis, cost, embedded, genetic, integral, optimization, passives
		Planning	1	capability, requirements

Table 2. Corpus Primary and Secondary Themes (continued).

Primary Theme	Number Articles	Secondary Theme	Number Articles	Key Words
Sustainability	15	Design	13	analysis, artifacts, availability, COTS, demand, design, digital, economic, eco-interaction, evolving, e-waste, factors, forecast, human, information, legacy, lifecycle, material, obsolescence, optimization, product, reliability, requirements, smart, system, technology, user, warranty
		Mapping	1	sustainment
		System	1	enterprise, military, model, sustainment, systems
Technology Insertion	13	COTS	1	COTS, evaluation, insertion, technology
		Design	3	analysis, capability, cost, COTS, design, effectiveness, forecast, insertion, lifecycle, management, obsolescence, optimization, planning, risk, sustainment, system
		Development	2	COTS, development, insertion, modular, open, spiral, systems
		Economics	1	costs, insertion, strategy, technology
		Leadership	1	acquisition, leadership, risk, technology, insertion
		Management	2	defense, insertion, management, obsolescence, technology
		Planning	1	acquisition, insertion, obsolescence, strategy
		Process	1	insertion, process, technology
		Viability	1	cost, evolvability, producibility, supportability, system, viability

Table 2. Corpus Primary and Secondary Themes (continued).

Primary Theme	Number Articles	Secondary Theme	Number Articles	Key Words
Technology Planning	82	Availability	4	availability, design, maintainability, reliability, requirements
		Design	7	analysis, condition, cost, COTS, design, disruption, forecasting, lifecycle, management, model, obsolescence, performance, prognostic, reliability, resource, reuse, safety, specification, strategy, sustainment, system
		Economics	25	algorithm, analysis, architecture, assembly, condition-based, complex, cost, COTS, decision, deterministic, disassembly, economics, electronics, genetic, latency, lifecycle, maintenance, manufacturing, material, model, modernization, module, network, obsolescence, options, optimization, prototype, real, recycle, refresh, reliability, return-on-investment, routing, service, strategy, system, test, time, tradeoff
		Engineering	1	design, development, engineering, environment, product
		Forecasting	2	exponential, forecasting, growth, jumping, technology
		Knowledge	3	analysis, cost, design, economic, education, knowledge, model, packaging, technology
		Management	8	acquisition, aid, capabilities, cost, decision, exploitation, framework, identification, innovation, integration, lifecycle, management, model, process, prognostic, protection, refresh, reliability, selection, service, strategy, system, technology, upgrade
		Modularity	1	design, engineering, information, modularity, technology
		Open Arch.	1	architecture, insertion, open, roadmapping, strategy, technology
		Packaging	1	analysis, packaging, prototyping, optimizing, system

Table 2. Corpus Primary and Secondary Themes (continued).

Primary Theme	**Number Articles**	**Secondary Theme**	**Number Articles**	**Key Words**
Technology Planning (continued)		Replacement	1	COTS, replacement, strategy
		Roadmapping	23	align, analysis, business, case, communications, components, cost, credibility, design, development, disruptive, dynamic, emerging, envelop, evolution, forecasting, goals, heuristic, hierarchy, information, innovation, insertion, integration, management, mining, obsolescence, passive, patent, performance, planning, product, prognostics, QFD, renew, revolution, roadmapping, service, science, strategy, supply, sustainment, technology, text, theory
		Technology	3	forecasting, planning, roadmapping, technology
		Testing	1	reliability, testing
		Upgrade	1	COTS, legacy, upgrade, strategy

Table 3 examines the corpus open coding in Pareto order for the primary themes and Pareto order for secondary themes without regard to classification within primary theme. Only the Pareto significant top 84% frequency general technology design primary themes and top 61.7% frequency secondary themes are display in Table 3. It is observable in Table 2 that thematic saturation for general technology design was achieved in the primary themes. Conservatively, thematic saturation was achieved for only the secondary themes listed in Table 3. Note in Table 3 that new technology insertion falls just outside the Pareto significant themes of active research. This strongly suggests that new technology insertion is in the early conceptual stage of research rather than being a mature research theme. Also, note in Table 2 that the primary themes cannot be considered as primitive due to the significant overlap in the secondary themes and key words.

Table 3. Pareto Order of Primary and Secondary Themes.

Primary Theme	**Pareto Frequency**
Technology Planning	82 (39.8%)
Obsolescence	31 (15.0%)
Design Ontology	29 (14.1%)
Design Theory	16 (7.8%)
Sustainability	15 (7.3%)
Technology Insertion	13
Lifecycle	9
Requirements Management	8
Open Architecture	3

Secondary Theme	**Pareto Frequency**
Design	38 (18.4%)
Economics	34 (16.5%)
Management	24 (11.7%)
Roadmapping	23 (11.2%)
Availability	4 (1.9%)
Models	4 (1.9%)

4.2 Grounded Theory / Text Mining Taxonomic Classes

The process started with text cleaning. All 206 articles admitted to the corpus were in Adobe Portable Document Format (pdf). Each article was opened in MS Word converting each to the Word format and then saved as a plain text document. Each plain text article was manually edited to

- delete journal and author information,
- replace accented letters, and
- remove bullets, brackets, parentheses, and punctuation characters except commas and periods.

During initial text mining of the corpus, further text cleaning was performed to transform upper case letters to lower case, remove numbers, remove remaining punctuation, remove English stop words, and stem derived words to their root forms. Common words (can, will, may, etc.) that appeared in the 25 highest count word set were also removed. The base text mining code is set forth in Appendix B.

Since development of the new technology insertion core reference ontology did not have a design foundational ontology for reference, the first pass text mining and grounded theory open coding constant comparative analysis was performed on the entire corpus of 206 articles. This joint development was performed under the assumption that the 193 articles that were not directly focused on new technology insertion reflected general technology design theory and application. The objective was to examine the new technology ontological structure within the systemic context of general design and application knowledge. The 25 highest count word set is listed in Table 4.

Table 4. Highest Count Word Set.

Word	Frequency		Word	Frequency
system	11,166		manage	4,243
design	9,837		time	4,030
use	8,676		information	3,991
cost	8,168		support	3,854
product	8,013		data	3,741
technology	7,411		operation	3,739
requirements	7,402		obsolescence	3,666
model	6,567		provide	3,518
process	5,926		function	3,422
develop	5,716		roadmap	3,170
part	5,624		analysis	3,135
component	4,667		performance	3,124
capability	4,517			

Cluster dendograms at 5%, 10%, and 15% sparsity were created to explore potential taxonomic categories. The full sequence of cluster dendograms is presented in Appendix C. The 5% sparsity cluster dendogram did not include the words “new” or “technology” or “insertion.” The 10% sparsity dendogram included the word “new” in a grouping at a lower hierarchical level (Figure 4). The 15% sparsity included the word “technology” at the third level hierarchy as a subcategory of the word “design,” and it contained the word “new” still in the grouping at the lower hierarchical level (Figure 5). Thus, 10% and 15% sparsity were used for subsequent cluster and correlation analyses.

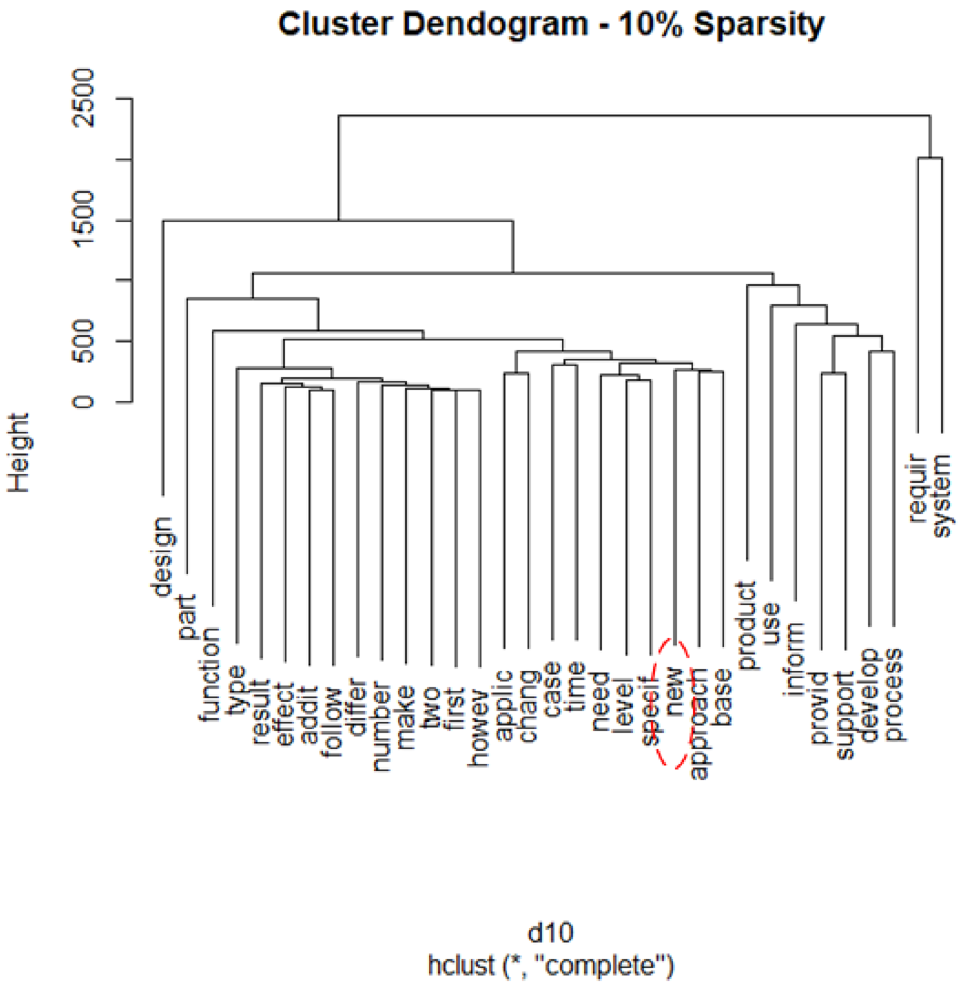


Figure 4. General Design Custer Dendogram at 10% Sparsity.

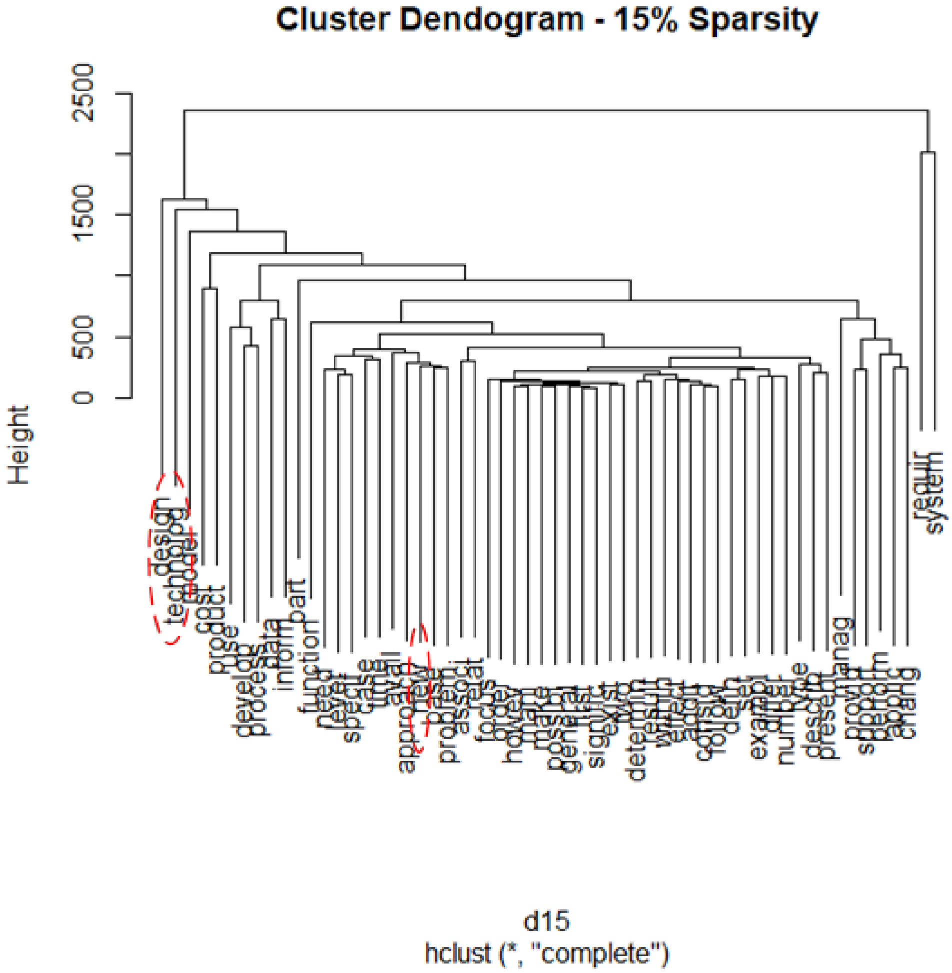


Figure 5. General Design Custer Dendogram at 15% Sparsity.

Cluster plots at 10% sparsity with 4 through 8 means were plotted to explore the potential number of independent taxonomic categories. The full sequence of cluster plots is presented in Appendix D. The 10% sparsity cluster plot with 4 means (Figure 6) partitions overall design from systems requirements; however, in cluster 3 product-use mission fulfillment concepts are classed with the concepts of information, provide, support, process, and develop, which are more closely related to the design process itself. These concepts were partitioned in the 10% cluster plot with 5 means in Figure 7; however, the concept of functionality is grouped with the design information concepts in cluster 5.

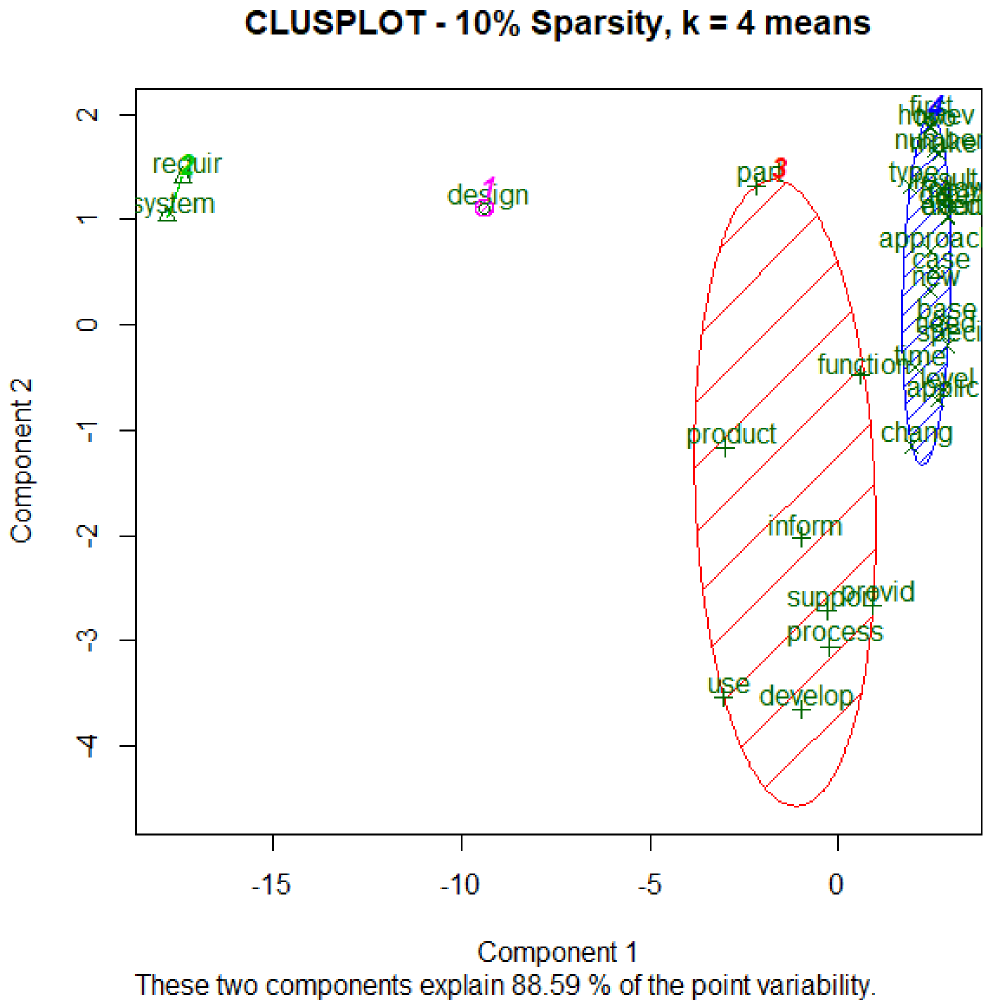


Figure 6. General Design Custer Plot at 10% Sparsity with 4 Means.

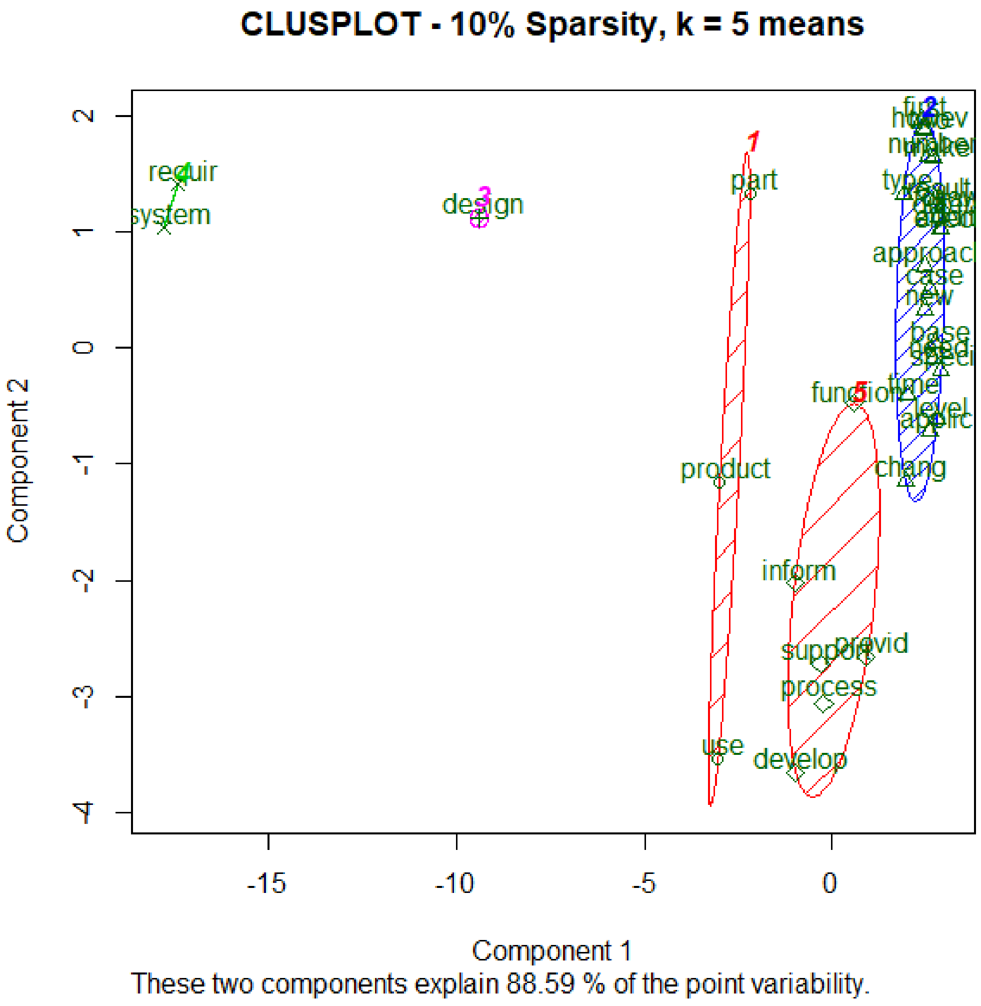


Figure 7. General Design Custer Plot at 10% Sparsity with 5 Means.

The 10% sparsity cluster plot with 6 means (Figure 8) was selected as the most representative candidate for the independent taxonomic categories, because it was the first to partition design process concept from the function concept and logically group function as concept with the similar concepts of new, base, specification, time, level, and change. Additionally, the 10% cluster plot with 6 means was the first to link the concepts of new and function. The 10% sparsity cluster plot with 7 means (Appendix D) only further partitioned noise terms in cluster 3. In the 10% sparsity cluster plot with 8 means (Appendix D), the part-product-use concept was decomposed into part and product-use clusters.

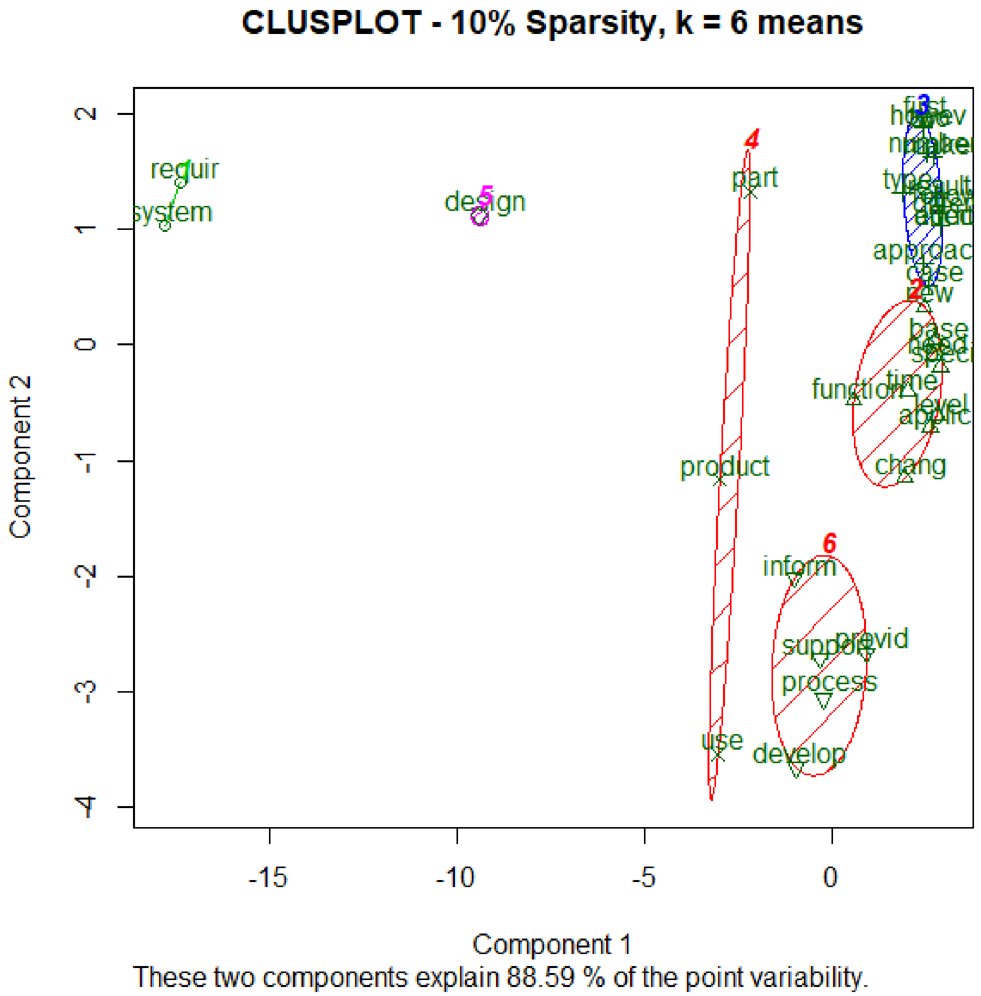


Figure 8. General Design Custer Plot at 10% Sparsity with 6 Means.

Joint examination of Figure 4 and Figure 8 with respect to development of a new technology insertion taxonomy within general design theory and application reveals a conceptual problem in relation to new technology insertion. At 10% sparsity, technology is not present as a concept. The concepts of new and function are in the same cluster in Figure 8 but at hierarchical levels 9 and 17 respectively. This strongly suggests that new functionality receives low consideration in technology design. From Figure 4, technology design focus is hierarchically ordered system requirements, design, product, part, information, use, process development, and function with the remaining concepts including new grouped in lower clusters.

Cluster plots at 15% sparsity with 4 through 8 means were plotted to explore this conceptual inconsistency further. The 15% sparsity with 4 means cluster plot in Figure 9 was the only plot that grouped systems-requirement-design into cluster 4; however, it grouped differing primitive concepts and in clusters 1. The concepts of technology, model, cost, product and use are conceptually more closely rated to product mission performance, whereas the concepts of data, information, support, process, and develop are more closely related to the design process itself.

The 15% sparsity with 5 means cluster plot in Figure 10 partitions the concept of system requirements into a cluster; the concepts of design, technology, model, cost, product, and use most closely related to product mission performance into a second cluster; and the concepts data, information, process, provide, support, and develop most closely related the design process in a third cluster. The concepts of part and function are grouped into the third cluster but spaced from the design process concepts. The concept of new is grouped into a separate cluster (4).

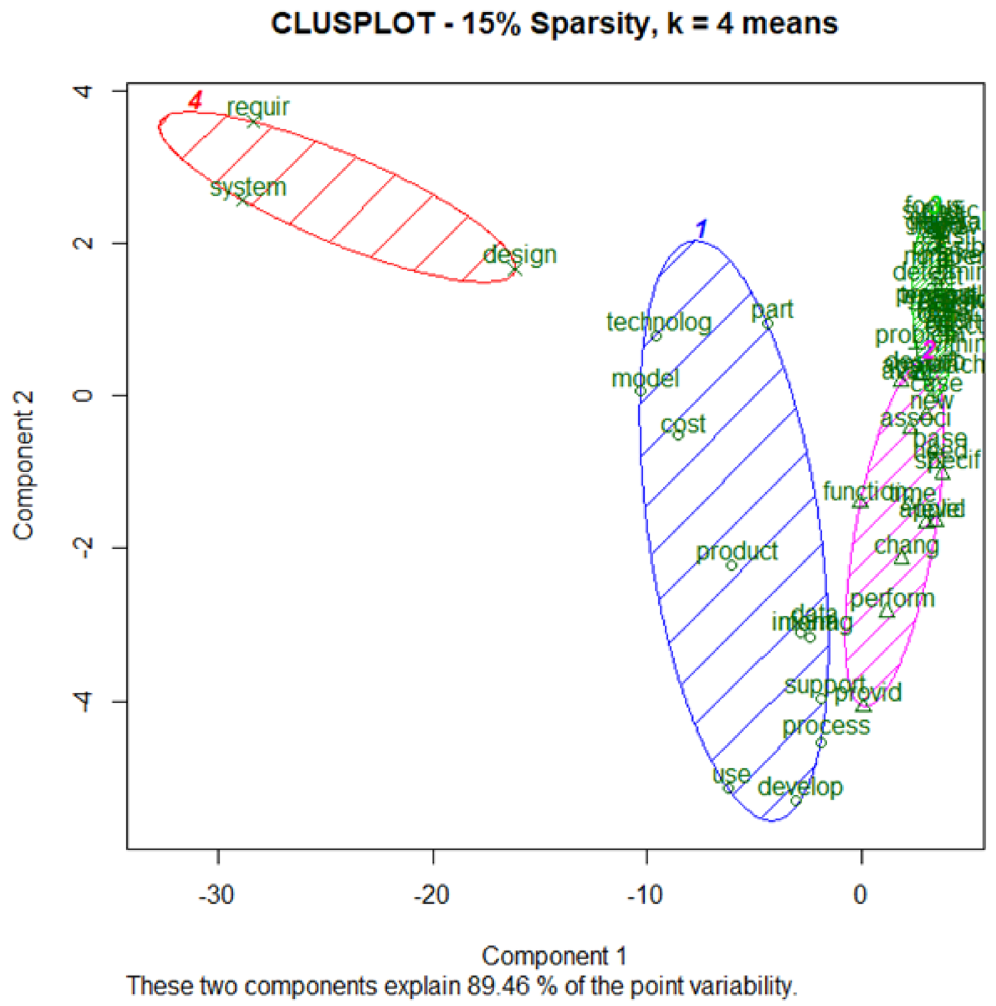


Figure 9. General Design Custer Plot at 15% Sparsity with 4 Means.

The 15% sparsity with 6 means cluster plot in Figure 11 leaves the concept of system requirements in a cluster. The concepts of design, technology, model, and cost are clustered. Now, part and product are clustered with the design process concepts in a third cluster. The concepts function is now grouped with the concepts of change and performance. Again, the concept of new is grouped into a separate cluster (6).

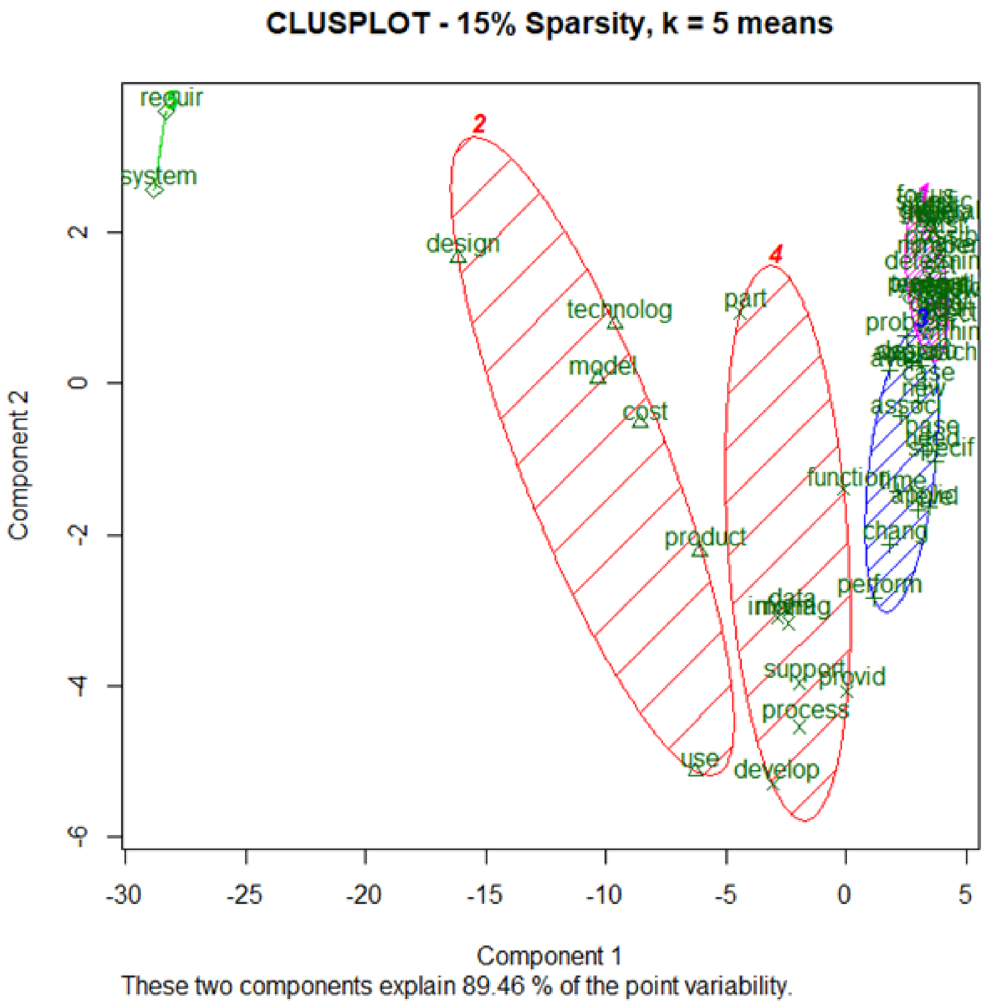


Figure 10. General Design Custer Plot at 15% Sparsity with 5 Means.

The 15% sparsity with 7 means cluster plot maintained the major clusters of the 6 means cluster plot and partitioned one lower level cluster (Appendix D). The 15% sparsity with 8 means cluster plot in Figure 12 partitioned the concept of design into its own cluster (agreeing with the 10% sparsity cluster plots). The concepts of technology, model, cost, product, and group reformed into a cluster. The concept of part remained in the cluster with design process concepts. The concepts of function, change, performance, and provide remained clustered. Finally, the concept of new was grouped into a separate cluster (3).

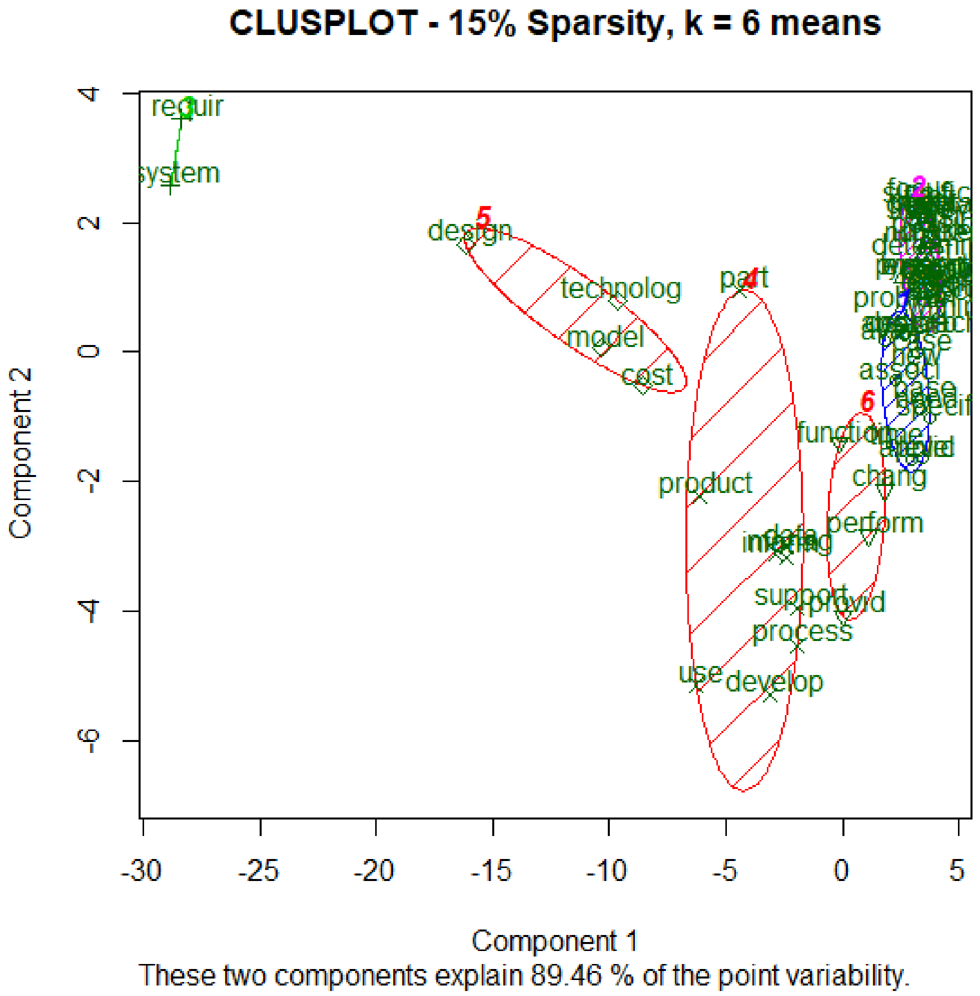


Figure 11. General Design Custer Plot at 15% Sparsity with 6 Means.

Examination of the 10% sparsity and 15% sparsity cluster plots strongly suggests that in general design theory and application the concept of new functionality receives low consideration in technology design and the concept of new technology insertion is, at best, disjoint. Thus, this research turned to text mining the 13 articles assigned to the primary theme of new technology insertion in open coding of the corpus.

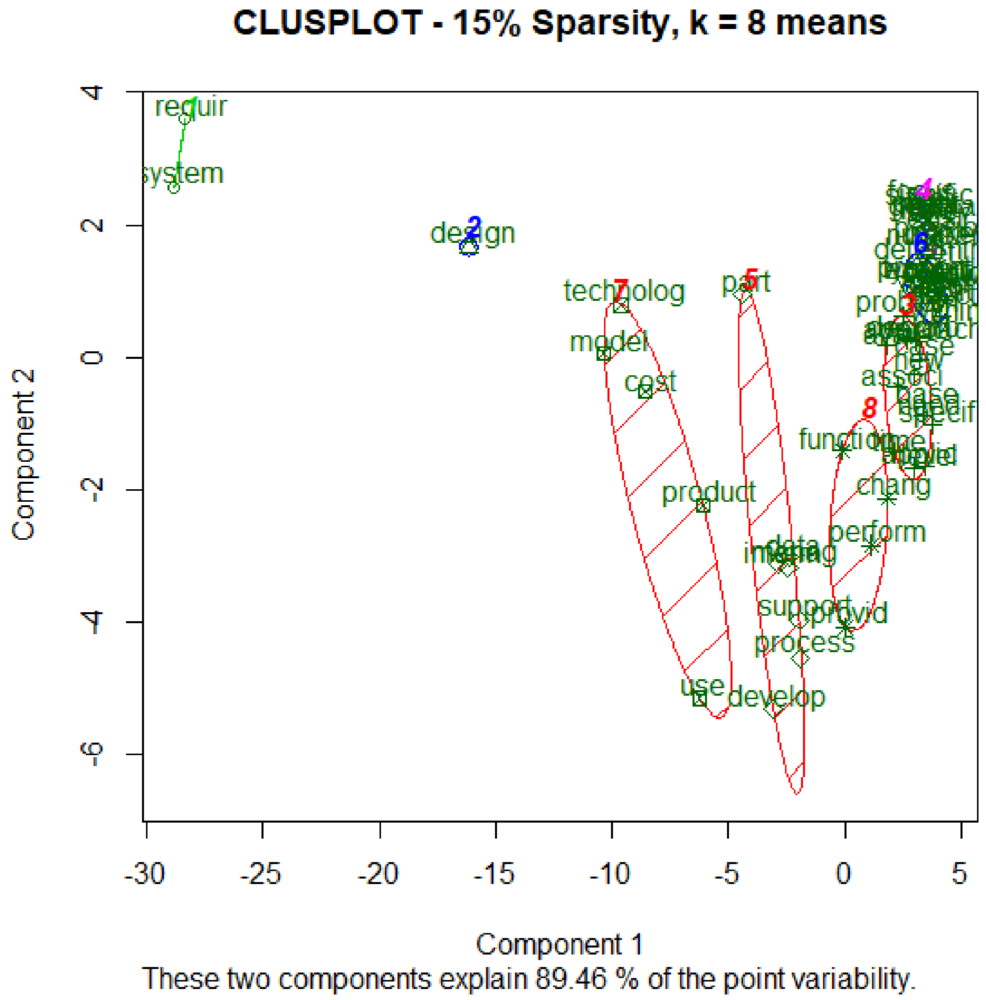


Figure 12. General Design Custer Plot at 15% Sparsity with 8 Means.

Cluster dendograms of the new technology insertion corpus at 10% and 15% sparsity were created to explore potential taxonomic categories. The 25 highest count word set is listed in Table 5.

Table 5. Highest Count New Technology Insertion Word Set.

Word	Frequency		Word	Frequency
system	1,444		contractor	372
technology	1,115		program	324
design	782		data	321
cost	781		test	321
refresh	604		performance	313
part	596		capability	312
use	589		provide	305
product	528		obsolescence	301
develop	519		manage	300
requirements	477		time	294
insert	427		change	282
process	425		support	281
plan	381			

Table 6. Order by Count General Design versus New Technology Insertion Word Sets.

Order	Gen. Design	New Tech. Ins.	Order	Gen. Design	New Tech. Ins.
1	system	system	14	manage	contractor
2	design	technology	15	time	program
3	use	design	16	information	data
4	cost	cost	17	support	test
5	product	refresh	18	data	performance
6	technology	part	19	operation	capability
7	requirements	use	20	obsolescence	provide
8	model	product	21	provide	obsolescence
9	process	develop	22	function	manage
10	develop	requirements	23	roadmap	time
11	part	insert	24	analysis	change
12	component	process	25	performance	support
13	capability	plan			

Continuing with constant comparison analysis, Table 6 maps the frequency order agreement and disagreement between the general design application and the new technology insertion top 25 words. Both levels are focused on the system. General design application places design, use cost, and product above technology considerations. Conversely, new technology insertion design places technology considerations second ahead of design, cost, refresh, and part. Interestingly, cost was the fourth level consideration for both. The concept of use was the third level consideration for general design practice, whereas use was the seventh level consideration for new technology insertion design. The concepts of product, requirements, and process receive higher consideration in general design application than in new technology insertion design. Again, interestingly, the concept of requirements ranked moderately high for both general design application (rank 7) and new technology insertion design (rank 10) even though only eight articles on requirements management were admitted to the corpus. The concept of model was not considered in new technology insertion design. Surprisingly, although obsolescence was the second highest in the number of articles admitted to the corpus, both general design application and new technology insertion design gave low consideration to obsolescence at ranks of 20 and 21 respectively. Similarly, technology roadmapping received low consideration in general design application (rank 21) and was not considered in new technology insertion design. Neither ranked sustainability in the top 25 words. It appears that general design application is focused on the general problem of delivering product designs that meets initial performance requirements, whereas new technology insertion design is focused on the micro problem of technology refresh or upgrade within the constraints of an existing system.

Again, cluster dendograms at 10% and 15% sparsity were created to explore potential taxonomic categories. The new technology insertion cluster dendograms are presented as part of the full sequence of cluster dendograms is in Appendix C. The 10% sparsity dendogram included the word “insertion” in a grouping at the seventh hierarchical level (Figure 13), and the 15% sparsity dendogram (Figure 14) included “insertion” at the eleventh hierarchical level.

Neither included the word “new.” It appears that the general concept of technology insertion is the main consideration rather than the more specific case of new technology insertion.

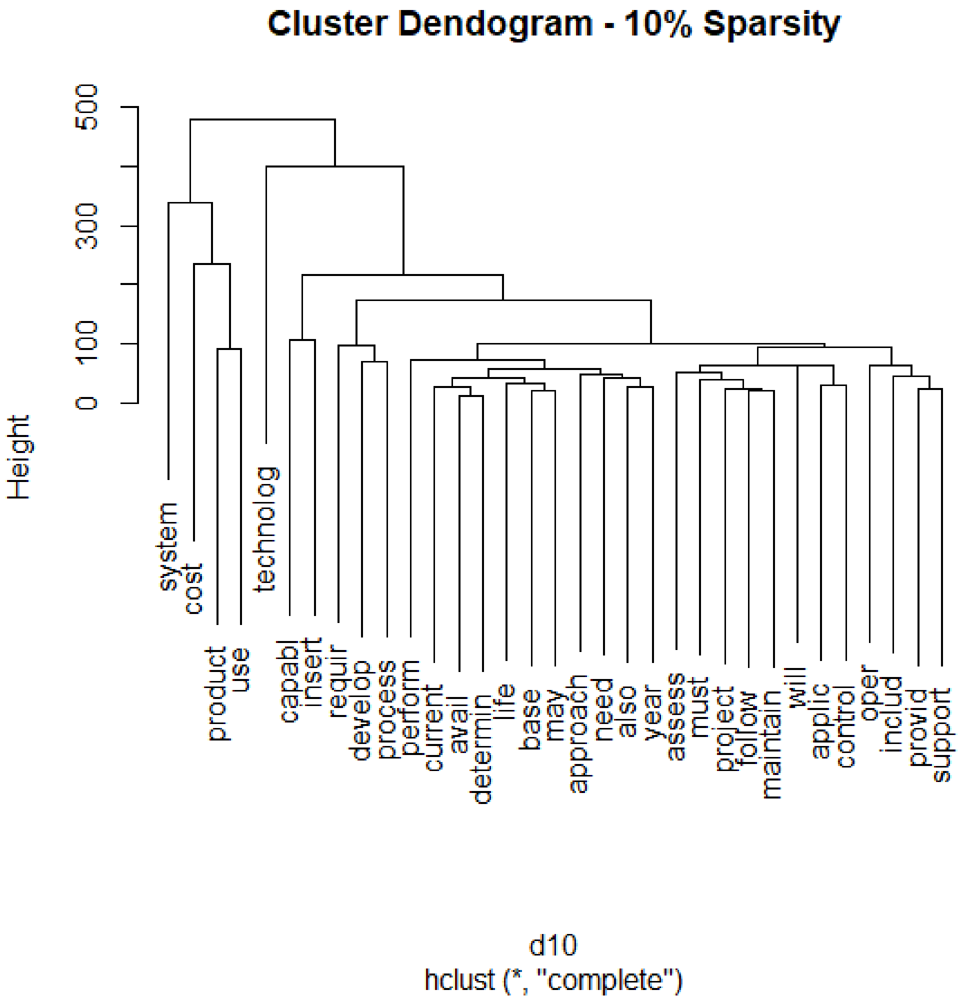


Figure 13. New Technology Insertion Custer Dendogram at 10% Sparsity.

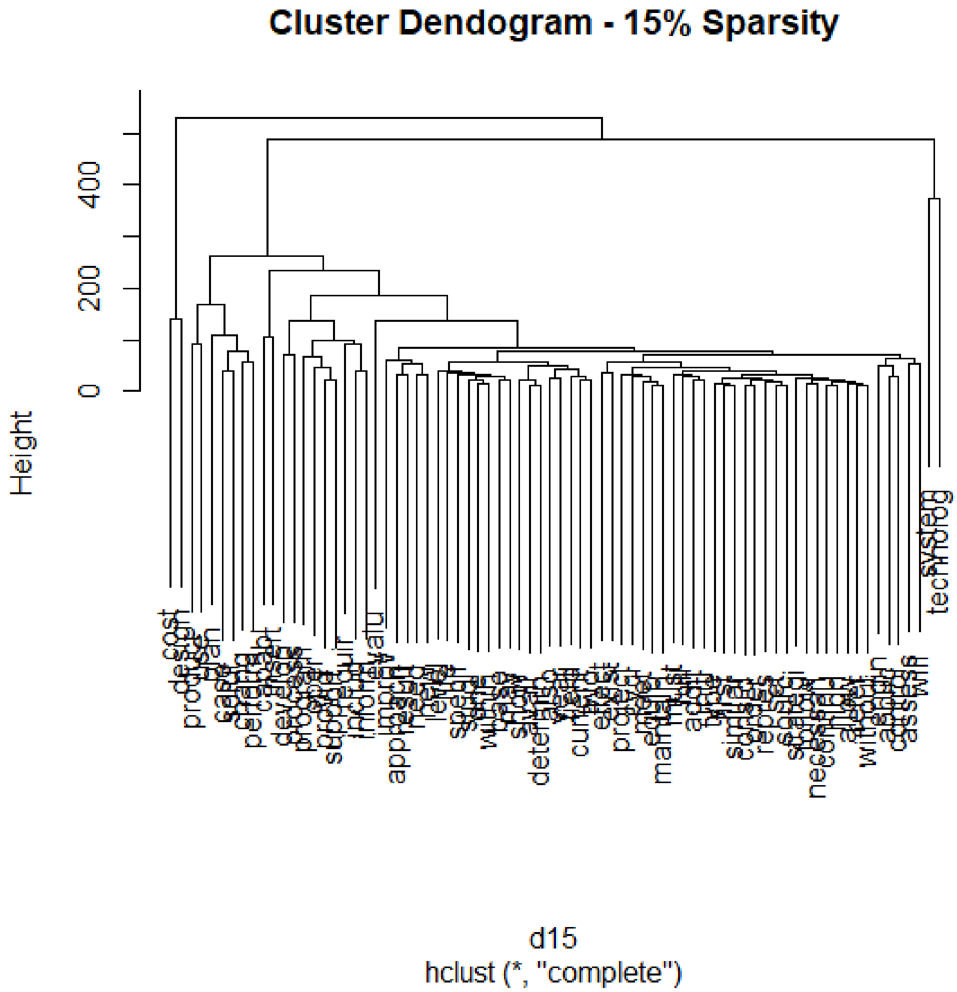


Figure 14. New Technology Insertion Custer Dendogram at 15% Sparsity.

Cluster plots at 10% and 15% sparsity with 4 through 6 means were plotted to explore the potential number of independent new technology insertion taxonomic categories. Both sets were almost identical in the identification of clusters. The cluster plots with 6 means (Figures 15 and 16) were selected as the stable clustering, because at both sparsity percentages only additional lower hierarchical clusters were partitioned. The full sequence of new technology insertion cluster plots is also presented in Appendix D.

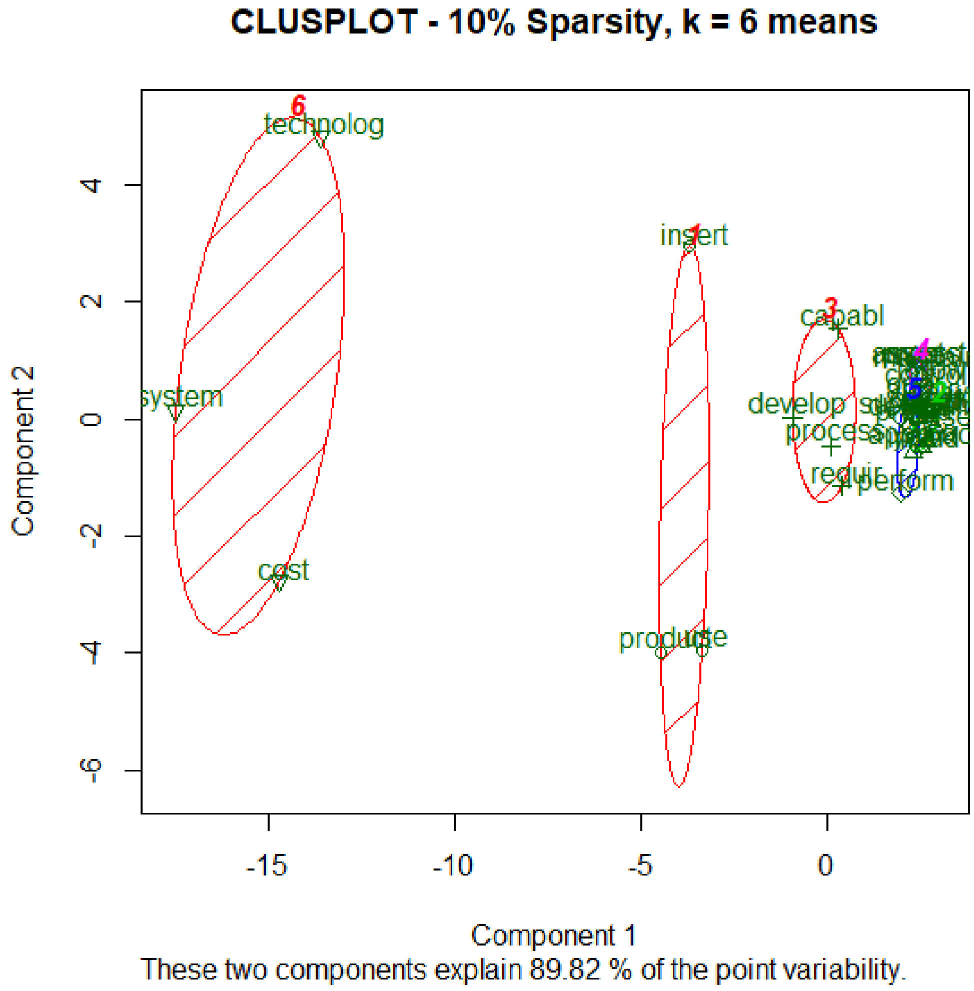


Figure 14. New Technology Insertion Custer Plot at 10% Sparsity with 6 Means.

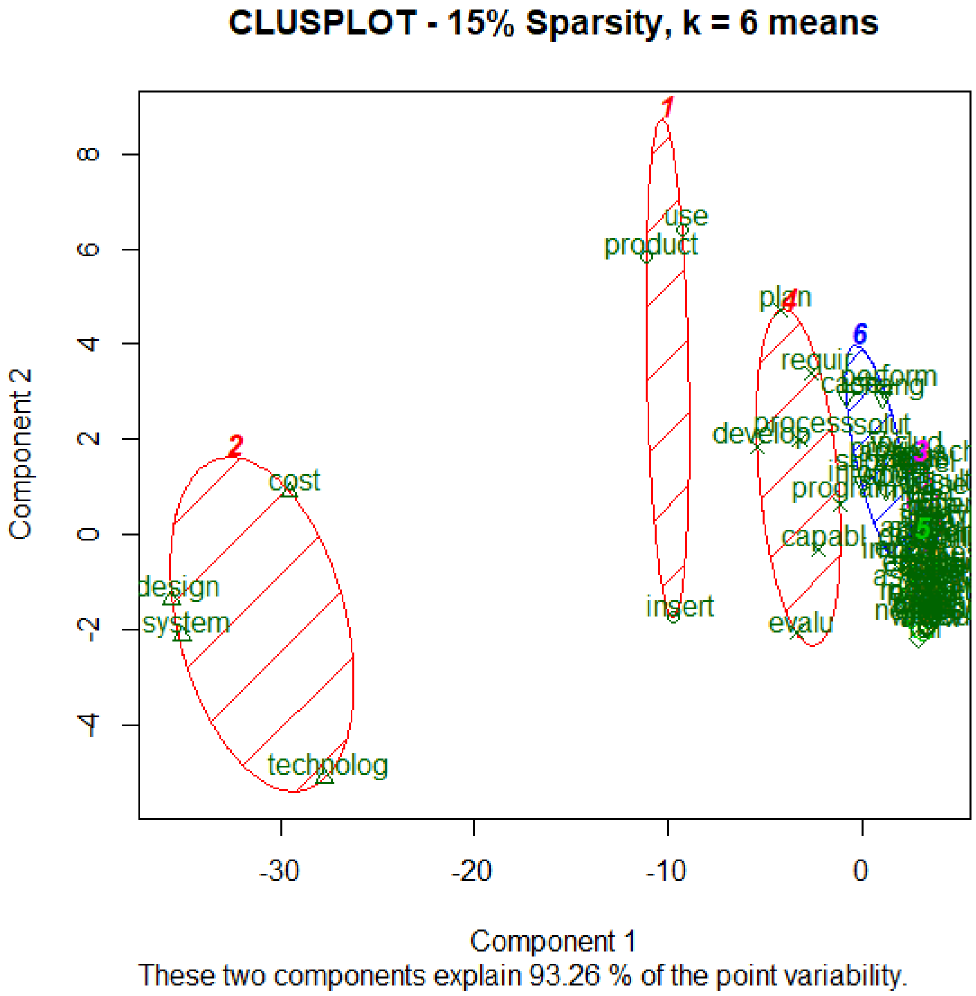


Figure 16. New Technology Insertion Custer Plot at 15% Sparsity with 6 Means.

Examination of the general design dendogram and cluster plot at 15% sparsity with 6 means in Figures 5 and 11 respectively and the new technology insertion dendogram and cluster plot at 15% sparsity with 6 means in Figures 14 and 16 respectively infer a layered hierarchical technology design and technology insertion design core reference taxonomy (Obsrt, 2010) with the primitive and composite concepts as illustrated in Figure 17. The technology design core reference taxonomy is a primitive semantic subordination at a higher level of granularity of the missing design foundational taxonomy. Likewise, the technology insertion design taxonomy is a

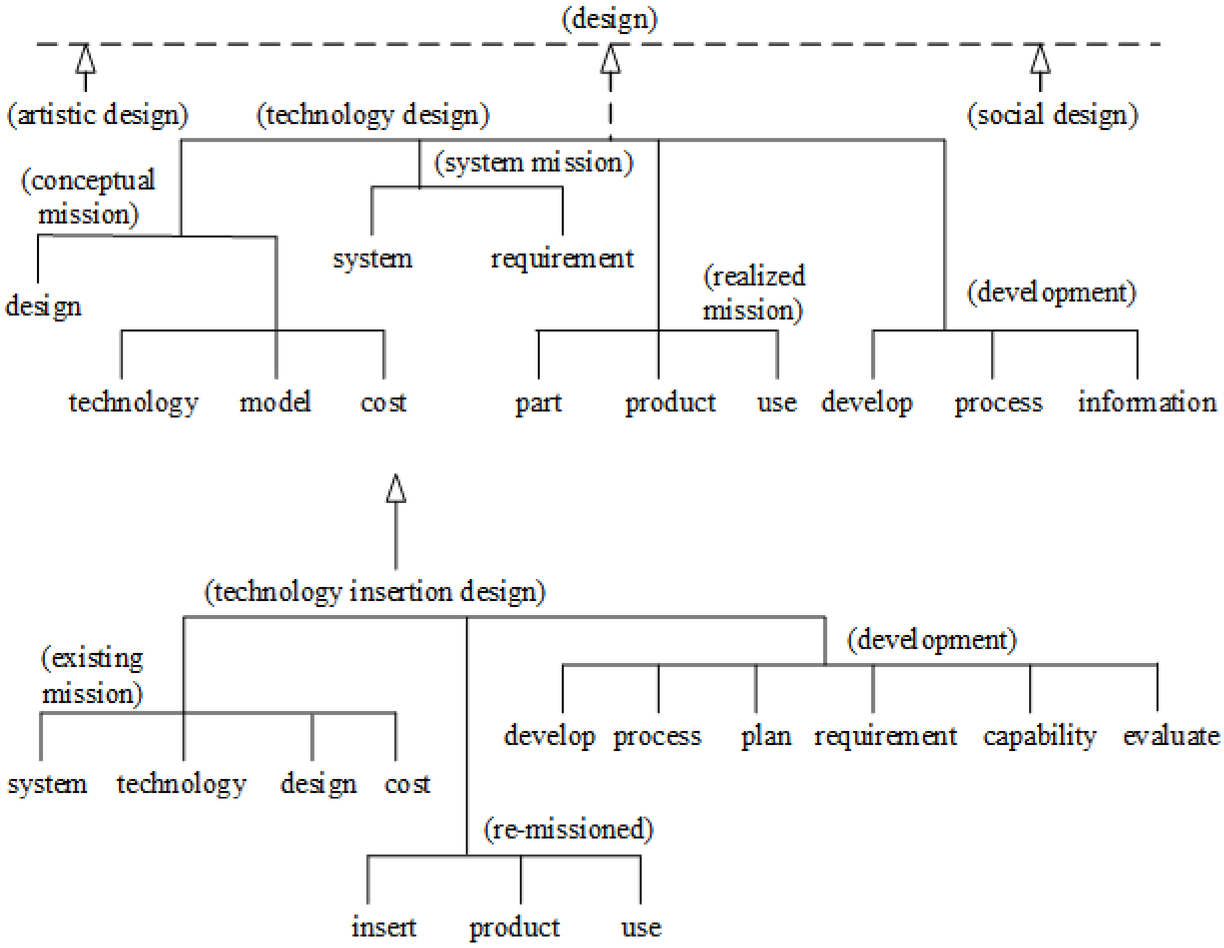


Figure 17. Hierarchical Technology Design and Technology Insertion Taxonomies.

primitive semantic subordination at a higher level of granularity of the technology design core reference taxonomy. Interestingly, not all overt primary and secondary themes identified in the corpus comparative analysis mapped to the design technology or technology insertion design taxonomic categories. Table 7 summarizes the mapping of corpus primary and secondary themes to taxonomic primitive concepts. This observation strongly suggests that primitive concepts are communicated as latent themes in underlying semantic meaning within text. Support for this supposition can be found in Zipf's Law (1936, 1949) of word frequency

Table 7. Corpus Primary and Secondary Themes Mapping to Taxonomic Concepts.

Corpus Primary Theme	Taxonomic Concept
Technology Planning	mission
Obsolescence	NA
Design Ontology	information
Design Theory	NA
Sustainability	NA
Technology Insertion	technology insertion design
Lifecycle	NA
Requirements Management	requirements
Open Architecture	NA

Corpus Secondary Theme	Taxonomic Concept
Design	mission
Economics	cost
Management	development
Roadmapping	NA
Availability	NA
Models	mission

distribution. Mathematically, word frequency in a constrained language approximately follows the power law. That is, the r^{th} most frequent word can be assigned a frequency rank $f(r)$ that scales according to

$$f(r) = C / r^{a} \qquad (11)$$

where f = frequency of occurrence, C = constant to be determined, r = numerical rank order, and a is a constant to be determined. Words are not just a collection of symbols; rather, words are a specific ordering of symbols that are assigned one or more units of semantic meaning in any given language. Calude and Pagel (2011) plotted word frequencies from Swadesh lists of 17 languages and found that the rank order frequency of words that were assigned the same semantic meaning conformed to Zipf's Law in ranking and frequency. Manin (2008) argued that Zipf's Law arises from constrained semantic hierarchies of specializations within a general language evolved to minimize synonymy overlap of the lexicons within the semantic space. Manin developed numerical models that demonstrated constrained hierarchies result in Zipf's Law.

To test for conformance of the corpus primary theme, technology design, and technology insertion design primitive concepts to Zipf's Law, the corpus primary theme words from Table 2 and the top 25 words from Tables 4 and 5 were each plotted on a log-log graph with the axes being Ln(rank) and Ln(frequency) as is standard practice in statistical linguistics. Performing algebraic simplification and taking the natural logarithm of both sides, equation 11 becomes,

$$\mathrm{Ln}(C) = \mathrm{Ln}(f(r)) + a\,\mathrm{Ln}(r) \tag{12}$$

Figure 18 shows that the corpus primary theme words fit to Zipf's Law with Ln(C) = 4.5037 at rank 1, $a = -1.2423$, and $\mathrm{Ln}(r) = x$ with $R^2 = 0.8983$. Figure 19 shows that the design technology primitives fit to Zipf's Law with Ln(C) = 9.2978 at rank 1 and $a\mathrm{Ln}(r) = -0.1168x^2 - 0.0245x$ with $R^2 = 0.9861$. Figure 20 shows that the technology insertion design primitives fit to Zipf's Law with Ln(C) = 7.3343 at rank 1, $a = -0.5345$, and $\mathrm{Ln}(r) = x$ with $R^2 = 0.9852$. Based on fit, the primitive concepts mapped to the design technology and technology insertion design taxonomies in Figure 17 can be considered as best representing the latent semantic meaning within the corpus text.

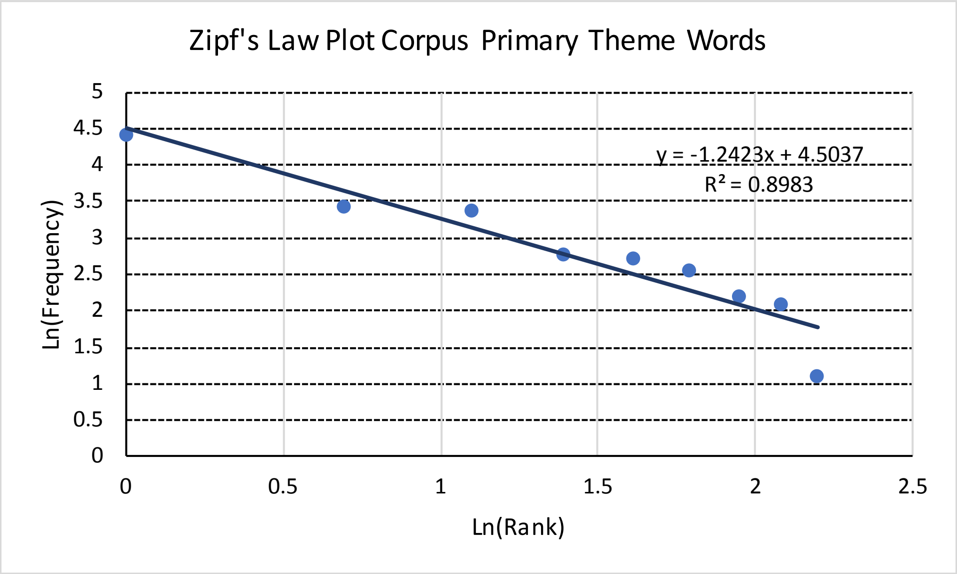


Figure 18. Corpus Primary Theme Words Fit to Zipf's Law.

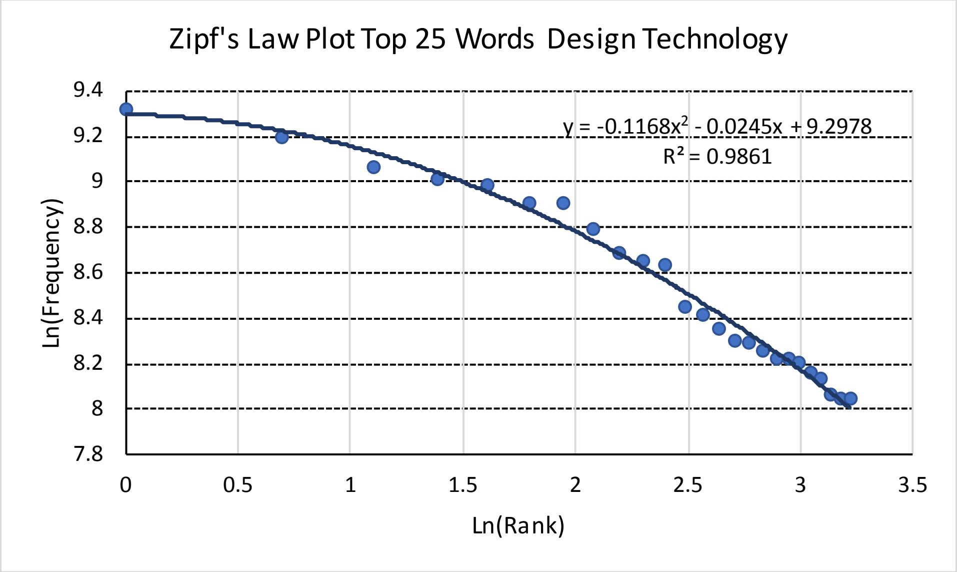


Figure 19. Design Technology Primitives Fit to Zipf’s Law.

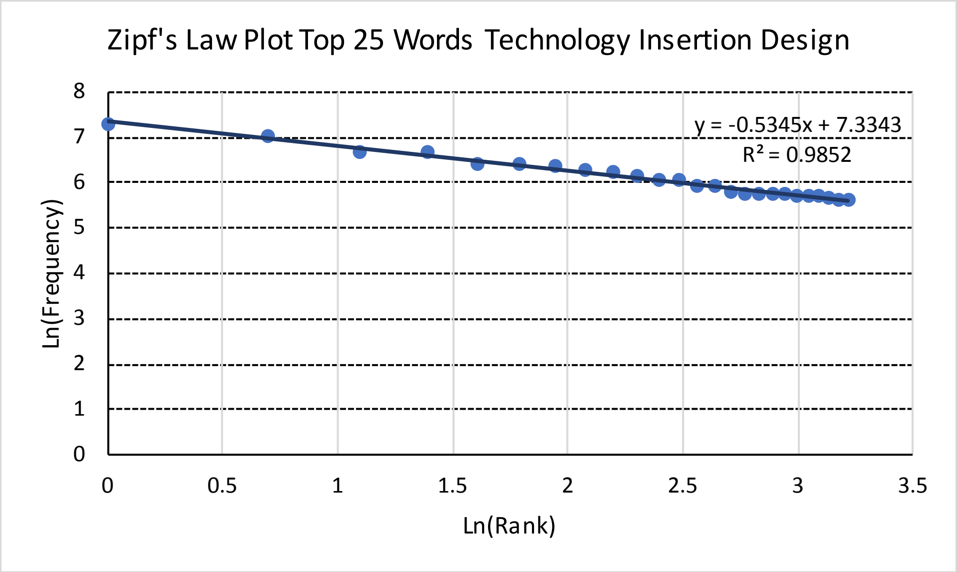


Figure 20. Technology Insertion Design Primitives Fit to Zipf’s Law.

Definitions, roles, and attributes of the technology design taxonomy primitive concept categories are specified in Table 8. Definitions, roles, and attributes of the technology insertion design taxonomy primitive concept categories are specified in Table 9. Definitions, roles, and attributes were specified to be minimum primitives in themselves based on Synsets (noun, adjective, verb, adverb, etc. expressing a distinct concept) specified in WordNet 3.1, WordNet®, developed by the Cognitive Science Laboratory at Princeton University (http://wordnetweb.princeton.edu/perl/webwn/). Concept Synsets downloaded from WordNet 3.1 are presented in Appendix E. The definition of a composite primitive concept category is derived from Boyce-Codd Normal Form (BC-NF) and Fourth Normal Form (4NF) applied to modular concepts (MCOs) is set forth as:

> ***Composite Primitive Concept Definition***: Let $\{O_1 \in MCO_1, \ldots, O_2 \in MCO_k\}$be a set of modular primitive concept objects that form a cluster, and let $\{a_{1m} \in A , \ldots, a_{kn} \in A\}$be the attributes of $O_1, \ldots, O_k$ respectively. Then, $\cup\{a_{1m} \in A , \ldots, a_{kn} \in A)$ specifies the composite primitive concept object $Comp(O_1, \ldots, O_k)$ that has normal cohesion. A composite primitive concept object $Comp(O_1, \ldots, O_k)$ that has normal cohesion is in Boyce-Codd Normal Form and Fourth Normal Form.

Table 8. Specification of Technology Design Primitive Concepts.

Primitive Concept Category	**Definition**	**Role**	**Existential Attributes (is-a relation)**	**State-Modification Attributes (has-a relation)**
system	Organized interacting entities working together under a unifying governance to achieve a common purpose.	Unify	▪ Governance ▪ Interactions ▪ Purpose ▪ Transformation	▪ Boundary ▪ Coordination ▪ Complexity ▪ Coupling ▪ Dynamic ▪ Environment ▪ Homeostasis ▪ Inputs ▪ Interdependency ▪ Niche ▪ Outputs ▪ Pluralism ▪ Policy ▪ Wholeness
requirement	A necessary attribute or function of an entity.	Necessity	▪ Condition ▪ Necessary	▪ Attribute ▪ Constraint ▪ Function ▪ Level ▪ Value
design	Purposeful creation or division of an entity's pattern or function.	Purposeful creation	▪ Creation ▪ Devise ▪ Purpose	▪ Action ▪ Function ▪ Pattern ▪ Representation
technology	The realization of an entity's application from scientific and engineering knowledge.	Functional performance	▪ Application ▪ Engineering ▪ Knowledge ▪ Realization ▪ Scientific	▪ Capacity ▪ Performance ▪ Robustness ▪ Stability
model	A theoretical or physical representation of an entity architecture.	Representation	▪ Architecture	▪ Accuracy ▪ Effectiveness ▪ Efficiency ▪ Precision ▪ Robustness

Table 8. Specification of Technology Design Primitive Concepts (continued).

Primitive Concept Category	**Definition**	**Role**	**Existential Attributes (is-a relation)**	**State-Modification Attributes (has-a relation)**
cost	Economic value of an expenditure.	Value	▪ Expenditure	▪ Amount ▪ Denomination ▪ Time
part	Fundamental type or unit of an entity.	Entity	• Type • Unit	• Composition • Form • Substance
product	Assemblage of entities to achieve a function level of performance.	Assemblage	• Assemblage • Function • Performance	• Entities • Interactions • Interconnections
use	Application of an entity to accomplish a purpose.	Application	• Accomplish • Purpose	• Method • Objectives
develop	Innovate or evolve new functionality or performance.	Originate	• Evolution • Innovation	• Change • Create • New • Purpose
process	Particular course of action intended to achieve a result.	Task	• Actions • Course • Intention	• Activity • Event • Mode • Path • Purpose
information	Facts received and understood.	Knowledge	• Facts • Understanding	• Assertion • Interpretation • Meaning • Proposition • Realization

Table 9. Specification of Technology Insertion Design Primitive Concepts.

Primitive Concept Category	**Definition**	**Role**	**Existential Attributes (is-a relation)**	**State-Modification Attributes (has-a relation)**
system	Table 8			
technology	Table 8			
design	Table 8			
cost	Table 8			
insert	A part, product, or system placed between or within other parts, products, or systems.	Upgrade	• Between • Placement • Within	• Interaction • Interface • Location
product	Table 8			
use	Table 8			
develop	Table 8			
process	Table 8			
plan	A sequence of steps to be carried out.	Actions	• Sequence • Steps	• Arrangement • Series • Location
requirement	Table 8			
capability	Limit or boundary of functionality or performance.	Limits	• Boundary • Limit	• Degree • Demarcation • Extent • Termination
evaluate	Assess or measure the ability, extent, nature, or significance.	Assessment	• Assess • Measure	• Estimate • Classification • Determination • Amount

The technology design taxonomic structure of Figure 17 and its corresponding attributes of Table 8 strongly imply the following structure, theorems, and questions for a technology design body of knowledge.

System Requirements – composite concept

Theorem: System mission performance can be specified such that the system can be designed, implemented, and managed to maximize its fit to its viable environmental niche.

- System – questions.
    1. What governance defines system purpose (fit to viable niche)?
    2. What interactions determine system performance necessary to achieve its purpose?
    3. What systemic transformation creates the system's purpose?
- Requirements – question: How can systemic mission performance be identified or created or devised, planned, and translated into technical model requirements that specify the necessary conditions of its purpose?

Conceptual Mission – composite concept

Theorem: Systemic mission performance purpose can be conceptualized and designed with respect to technological constraints and each system's viable environmental niche?

- Design – question.
    1. How can systemic mission performance purpose be created or devised to achieve systemic purpose?
- Technology – questions.
    1. Given knowledge and intellectual property constraints, what combination of existing and new technology capabilities are necessary to realize required systemic mission performance purpose?
    2. Lacking existing technological capability, how can innovative engineering and scientific knowledge be developed to realize required new technology functionalities yielding new systemic mission performance purpose?

- Model – questions.
  1. What modeling methods are needed to overlay systemic mission performance purpose mathematical models onto qualitative architectural views.
  2. How can systemic mission performance purpose relative to its viable environmental niche be quantified, measured, modeled, verified, and validated?
- Cost – questions.
  1. How can the expenditure for differing levels of systemic mission performance purpose be quantified and measured in terms of cost?
  2. How can the performance/cost ratio be maximized for existing technological functional capabilities for each level of achieved systemic mission performance purpose relative to required systemic mission performance purpose?
  3. How can the performance/cost ratio be maximized in the development in innovative new technologies necessary to achieve new levels of systemic mission performance purpose needed to maintain or expand the system's viable environmental niche?

Realized Mission – composite concept

Theorem: Part and product functionalities can be synthesized into systemic mission performance purpose that maintains or expands its viable environmental niche?

- Part – question: Given knowledge and intellectual property constraints, what fundamental type or unit forms required parts functionality?
- Product – questions.
  1. Given knowledge and intellectual property constraints, how can parts be assembled into products of required functionality and performance?

2. Given knowledge and intellectual property constraints, how can joint products functionality produce systemic mission performance purpose that maintains or expands the viable environmental niche?

- Use – question: How can joint product functionality be designed to achieve customer use and systemic mission performance purpose?

Development – composite concept

Theorem: The development process must continually innovate or evolve to most efficiently and effectively integrate information, parts, and products that produce new functionality or performance.

- Development – questions:
  1. How can designed products functionality be scaled up to deliver or exceed expected systemic mission performance purpose in customer use and stakeholder expectations?
  2. How can designed systemic performance purpose be scaled up to deliver or exceed necessary systemic viable mission performance purpose?
- Process – question: How can the optimum development process be designed or evolved to deliver product family functionality or systemic viable performance?
- Information – questions:
  1. How can systemic mission performance purpose facts be identified and understood?
  2. What necessary and sufficient information is required to develop products' functionality and resultant systemic mission performance purpose?

The technology insertion taxonomic structure of Figure 17 and its corresponding additional attributes of Table 9 strongly imply the following structure, theorems, and questions for a technology design body of knowledge.

Existing Mission – composite concept

Theorem: Realized mission performance can be re-conceptualized to admit the insertion of parts, components, or products that extend or upgrade systemic mission performance.

- System questions
    1. How can viable environmental niches requiring new technology be identified, quantified, and mathematically modeled?
    2. What robust system identification methodology can be developed for building mathematical models of dynamic systems using measurements of the environmental constraints and the system's input, new technology transformation, and output factors and variables?
- Technology questions
    1. Given knowledge and intellectual property constraints, what optimum combination of new technology insertion functionalities are necessary and sufficient to achieve identified systemic mission performance purpose in the new environmental niche?
    2. Lacking existing technological capability, how can innovative methodologies and methods be developed to create needed new technology insertion functionalities?
- Design questions
    1. What is the optimum methodology and methods for conceptualizing new systemic mission performance purpose with respect to its viable environmental niche?

   2. How can new systemic mission performance be synthesized into internal subsystems, components, and parts functional requirements with reference to environmental and internal dependencies and correlations?

- Cost questions
   1. How can the performance/cost ratio be maximized for new technological functional capabilities for each level of achieved systemic mission performance purpose relative to required systemic mission performance?
   2. How can the performance/cost ratio be maximized in the development of innovative new technologies necessary to achieve new levels of systemic mission performance purpose needed to expand the system's viable environmental niche?

Re-missioned – composite concept

Theorem: Realized mission performance can be re-designed to admit the insertion of parts, components, or products that extend or upgrade systemic mission performance purpose that exceeds required performance in customer use and stakeholders' expectations.

- Product – question: Given knowledge and intellectual property constraints, how can joint products functionality produce new systemic performance purpose that expands the viable environmental niche?
- Use – question: How can joint product functionality be designed to be robust to customer use and stakeholders' expectations in new application?
- Insert question: How can new parts, components, products, or systems be interfaced and integrated into existing systems such that the insertion achieves required or expected extended or upgraded systemic mission performance purpose and minimizes the risk for non-designed failures or degradation of performance?

Development – composite concept

Theorem: The development process must continually innovate or evolve to most efficiently and effectively insert parts, components, products, and systems that produce extended or upgraded systemic performance.

- Development - questions:
    1. How can designed new products functionality be scaled up and inserted to deliver or exceed expected performance in customer use and stakeholders, expectations?
    2. How can designed systemic performance be scaled up and inserted to deliver or exceed necessary systemic mission performance purpose?
- Process - question: How can the optimum development process be designed or evolved to deliver new product family functionality or new systemic performance purpose?
- Requirements – question: How can new systemic mission performance requirements be identified, quantified, and translated into mathematical models?
- Plan question – How must the conceptualization, design, and development sequence be restructured to efficiently and effectively insert new parts, components, products, or systems' technology into existing systems?
- Capability questions
    1. How can the boundaries or limits of existing knowledge and new technology functionalities be identified?
    2. Given that existing knowledge and new technology functionality do not provide for the required extended or upgraded technological capability, how can innovative methodologies and methods be developed to create needed new technology functionalities?

- Evaluate question – How can extended or upgraded systemic mission performance purpose relative to its viable environmental niche be quantified, measured, modeled, verified, and validated?

4.3 Grounded Theory / Text Mining Axiomatic Relationships

From SENSUS Process 4, for each taxonomic primitive category, axiomatic relationships were specified within and among the primitive concepts. Based on fits to Zipf's Law, the correlational relationships within the design technology and technology insertion design taxonomies in Figure 17 can be considered as best representing the latent axiomatic relationships within the corpus text. Coefficients of determination among primitive concepts were identified using the R package tm text mining findAssocs() function using $R^2 = 0.50$ as the lower limit for the 15% sparsity document term matrix. Table 10 presents the coefficient of determination relationships among primitive concepts within the technology design taxonomy. Figure 21 shows that the coefficient of determination values formed a bimodal distribution. Thus, $0.50 \leq R^2 \leq 0.74$ was considered as moderate axiomatic dependency among concepts and $0.75 < R^2$ was considered as high axiomatic dependency.

The technology design axiomatic dependencies are plotted in Figure 22. Examination of Table 10 and Figure 22 show that the strongest dependencies exist within the development composite concept, moderate dependencies within system mission and conceptual mission

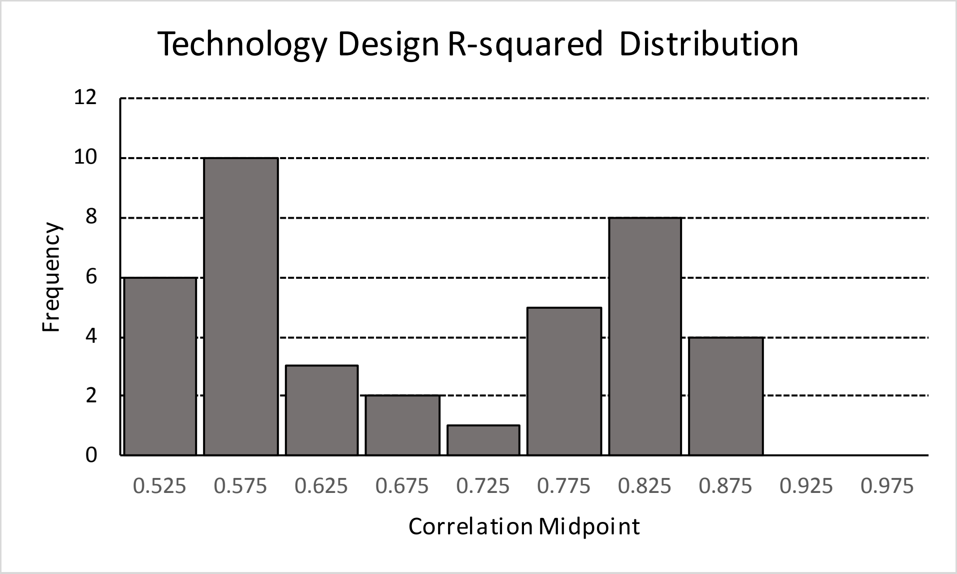


Figure 21. Technology Design R-squared Distribution.

composite concepts, and the weakest dependencies within the realized mission composite concept. This strongly suggests: (1) technology design and development processes are well defined and executed, (2) the identification of system mission requirements and translation into conceptual mission requirements are not as well defined and achieved, and (3) realized mission performance among parts, products, and use are not well linked. Again, Examination of Table 10 and Figure 22 show that the strongest set of dependencies exist between the development, systems requirements, and conceptual mission composite concepts, and the weakest set of dependencies exist between these three composite concepts and the realized mission composite concept. The strongest primitive concept dependencies between the realized mission composite concepts and the other three composite concept primitives are: (1) requirements to product, (2) model to product, (3) cost to use, and (4) development process to mission use. This strongly suggests that the focus of technology design is on product cost and functionality and less on realized systemic mission performance.

Table 10: Correlational Relationships Among Technology Design Primitive Concepts

	system	requirements	design	technology	model	cost	part	product	use	develop	process	information
system	1.00	0.59		0.84	0.84			0.55	0.90	0.82	0.76	0.54
requirements	0.59	1.00	0.78			0.88		0.88	0.61	0.80	0.71	0.66
design		0.78	1.00		0.56	0.56	0.56		0.53	0.56	0.50	0.56
technology	0.84			1.00						0.51		
model	0.84		0.56		1.00	0.84		0.84	0.56		0.84	0.59
cost		0.88	0.56		0.84	1.00		0.58	0.51			
part			0.56				1.00					
product	0.55	0.88			0.84	0.58		1.00	0.65	0.62	0.57	
use	0.90	0.61	0.53		0.56	0.51		0.65	1.00	0.85	0.85	0.70
develop	0.82	0.80	0.56	0.51				0.62	0.85	1.00	0.88	0.77
process	0.76	0.71	0.50		0.84			0.57	0.85	0.88	1.00	0.77
information	0.54	0.66	0.56		0.59				0.70	0.77	0.77	1.00

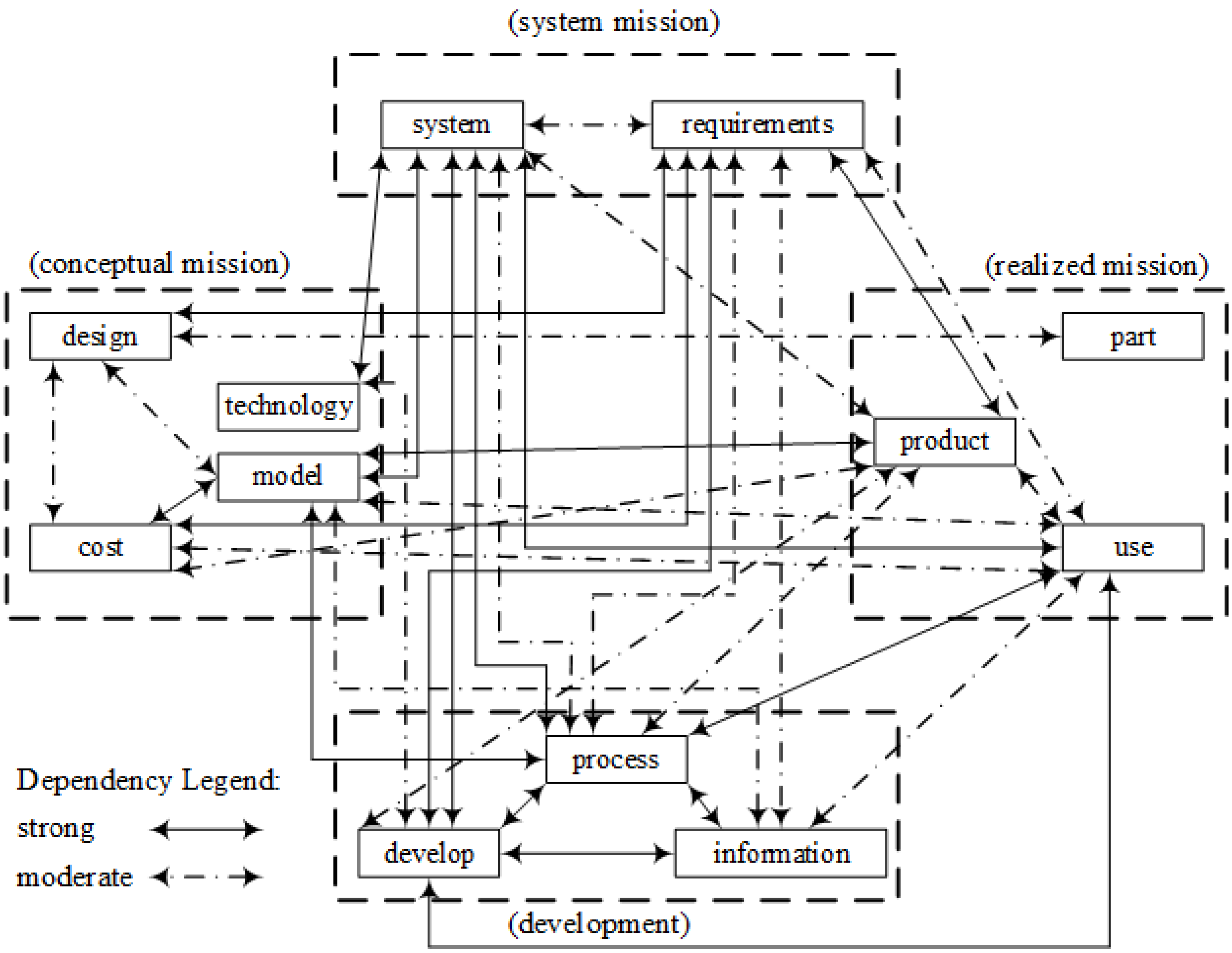


Figure 22. Technology Design Axiomatic Dependencies.

Table 11 sets forth logical axiomatic relationships within and between technology design composite and primitive concepts.

Table 11. Axiomatic Relationships Between Technology Design Primitive Concepts.

Composite	**Primitive**	**Dependency**	**Axiom**
system mission	system	within	System is strongly correlated with mission.
		between	System is strongly correlated with technology.
		between	System is strongly correlated with model.
		between	System is moderately correlated with product.
		between	System is strongly correlated with use.
		between	System is strongly correlated with develop.
		between	System is strongly correlated with process.
		between	System is moderately correlated with information.
	requirement	within	Requirement is moderately correlated with system.
		between	Requirement is strongly correlated with design.
		between	Requirement is strongly correlated with cost.
		between	Requirement is strongly correlated with product.
		between	Requirement is moderately correlated with use.
		between	Requirement is strongly correlated with develop.
		between	Requirement is moderately correlated with process.
		between	Requirement is moderately correlated with information.
conceptual mission	design	within	Design is moderately correlated with cost.
		within	Design is moderately correlated with model.
		between	Design is strongly correlated with requirement.
		between	Design is moderately correlated with part.
		between	Design is moderately correlated with use.
		between	Design is moderately correlated with develop.
		between	Design is moderately correlated with process.
		between	Design is moderately correlated with information.
	technology	between	Technology is strongly correlated with system.
		between	Technology is moderately correlated with develop.
	model	within	Model is moderately correlated with design
		within	Model is strongly correlated with cost
		between	Model is strongly correlated with system.
		between	Model is strongly correlated with product.
		between	Model is moderately correlated with use.
		between	Model is strongly correlated with process

Table 11. Axiomatic Relationships Between Technology Design Primitive Concepts (continued).

Composite	**Primitive**	**Dependency**	**Axiom**
conceptual mission	model	between	Model is moderately correlated with information.
	cost	within	Cost is moderately correlated with design.
		within	Cost is strongly correlated with model.
		between	Cost is strongly correlated with requirement.
		between	Cost is moderately correlated with product.
		between	Cost is moderately correlated with use.
development	develop	within	Develop is strongly correlated with process.
		within	Develop is strongly correlated with information.
		between	Develop is strongly correlated with system.
		between	Develop is strongly correlated with requirement.
		between	Develop is moderately correlated with design.
		between	Develop is moderately correlated with technology.
		between	Develop is moderately correlated with product.
		between	Develop is strongly correlated with use.
	information	within	Information is strongly correlated with develop.
		within	Information is strongly correlated with process.
		between	Information is moderately correlated with system.
		between	Information is moderately correlated with requirements.
		between	Information is moderately correlated with design.
		between	Information is moderately correlated with model.
		between	Information is moderately correlated with use.
	process	within	Process is strongly correlated with develop.
		within	Process is strongly correlated with information.
		between	Process is strongly correlated with system.
		between	Process is moderately correlated with requirement.
		between	Process is moderately correlated with design.
		between	Process is strongly correlated with model.
		between	Process is moderately correlated with product.
		between	Process is strongly correlated with use.
realized mission	part	between	Part is moderately correlated with design.

Table 11. Axiomatic Relationships Between Technology Design Primitive Concepts (continued).

Composite	**Primitive**	**Dependency**	**Axiom**
realized mission	product	within	Product is moderately correlated with use.
		between	Product is moderately correlated with system.
		between	Product is strongly correlated with requirement.
		between	Product is strongly correlated with model.
		between	Product is moderately correlated with cost.
		between	Product is moderately correlated with develop.
		between	Product is moderately correlated with process.
	use	within	Use is moderately correlated with product.
		between	Use is strongly correlated with system.
		between	Use is moderately correlated with requirement.
		between	Use is moderately correlated with design.
		between	Use is moderately correlated with model.
		between	Use is moderately correlated with cost.
		between	Use is strongly correlated with develop.
		between	Use is strongly correlated with process.
		between	Use is moderately correlated with information.

Table 12 presents the coefficient of determination relationships among primitive concepts within the technology insertion design taxonomy. Figure 23 shows that the coefficient of determination values formed a bimodal distribution. Thus, $0.50 \le R^2 \le 0.84$ was considered as moderate axiomatic dependency among concepts and $0.85 < R^2$ was considered as high axiomatic dependency.

The technology insertion design axiomatic dependencies are plotted in Figure 24. Examination of Table 12 and Figure 24 show that the strongest dependencies exist between the existing mission and re-missioned composite concepts. Mainly moderate dependencies exist between the existing mission and re-missioned composite concepts to the development composite concept. The highest dependencies between the existing mission and re-missioned composite concepts are: system to use 0.87 high, technology to insert 0.94 high, design to insert

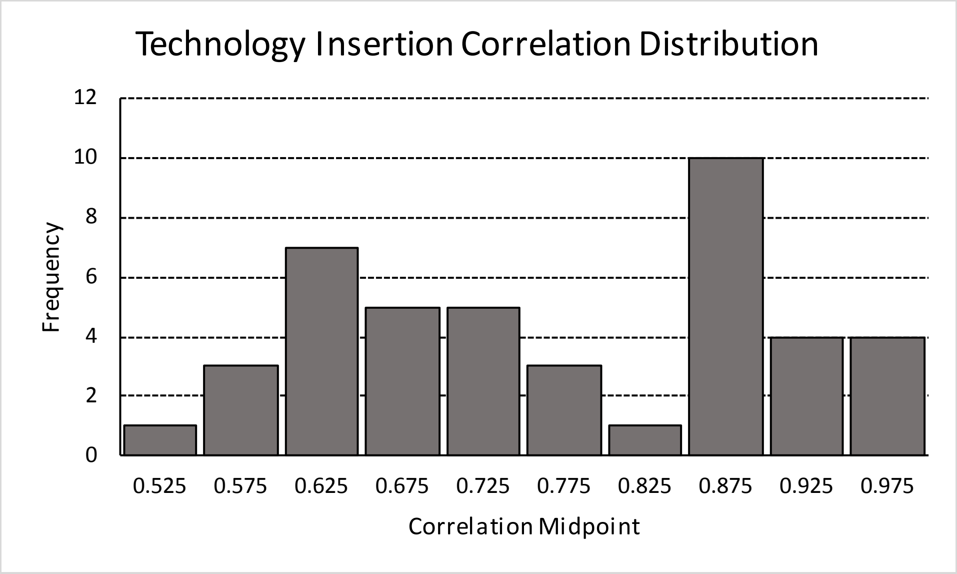


Figure 23. Technology Insertion Design R-squared Distribution.

0.98 high, design to product 0.92 high, design to use 0.90 high, cost to insert 0.96 high, cost to product 0.87 high, and cost to use 0.92 high. Overall, the technology insertion design dependency relationships were higher between the composite concepts of existing mission and re-missioned than they were between the composite concepts of conceptual mission and realized mission for technology design. For technology design, the corresponding high dependency relationship was only model to product 0.84. This strongly suggests that planning for technology insertion is not a priority in initial technology design. Rather, from Table 10 and Figure 22, initial technology design appears to prioritize identification and translation of system mission requirements into conceptual mission requirements through the design and development process. Conversely, technology insertion design focuses on the design of technology insertion to provide re-missioned usefulness within the constraints of existing technology and costs.

Table 13 sets forth logical axiomatic relationships within and between technology insertion design composite and primitive concepts.

Table 12: Correlational Relationships Among Technology Insertion Design Primitive Concepts

	system	technology	design	cost	insert	product	use	develop	process	plan	requirement	capability	evaluate
system	1.00	0.58	0.64	0.71	0.58	0.70	0.87	0.73	0.68	0.75	0.77		0.58
technology	0.58	1.00		0.62	0.94	0.55		0.62		0.70		0.87	
design	0.64		1.00	0.95	0.98	0.92	0.90			0.80			0.98
cost	0.71	0.62	0.95	1.00	0.96	0.87	0.92			0.77			0.96
insert	0.58	0.94	0.98	0.96	1.00			0.66		0.75		0.89	
product	0.70	0.55	0.92	0.87		1.00	0.90			0.84	0.64		0.87
use	0.87		0.90	0.92		0.90	1.00			0.90	0.62		0.88
develop	0.73	0.62			0.66			1.00	0.87		0.70		
process	0.68							0.87	1.00		0.63		
plan	0.75	0.70	0.80	0.77	0.75	0.84	0.90			1.00	0.63		0.75
requirement	0.77					0.64	0.62	0.7	0.63	0.63	1.00		
capability		0.87			0.89							1.00	
evaluate	0.58		0.98	0.96		0.87	0.88				0.75		1.00

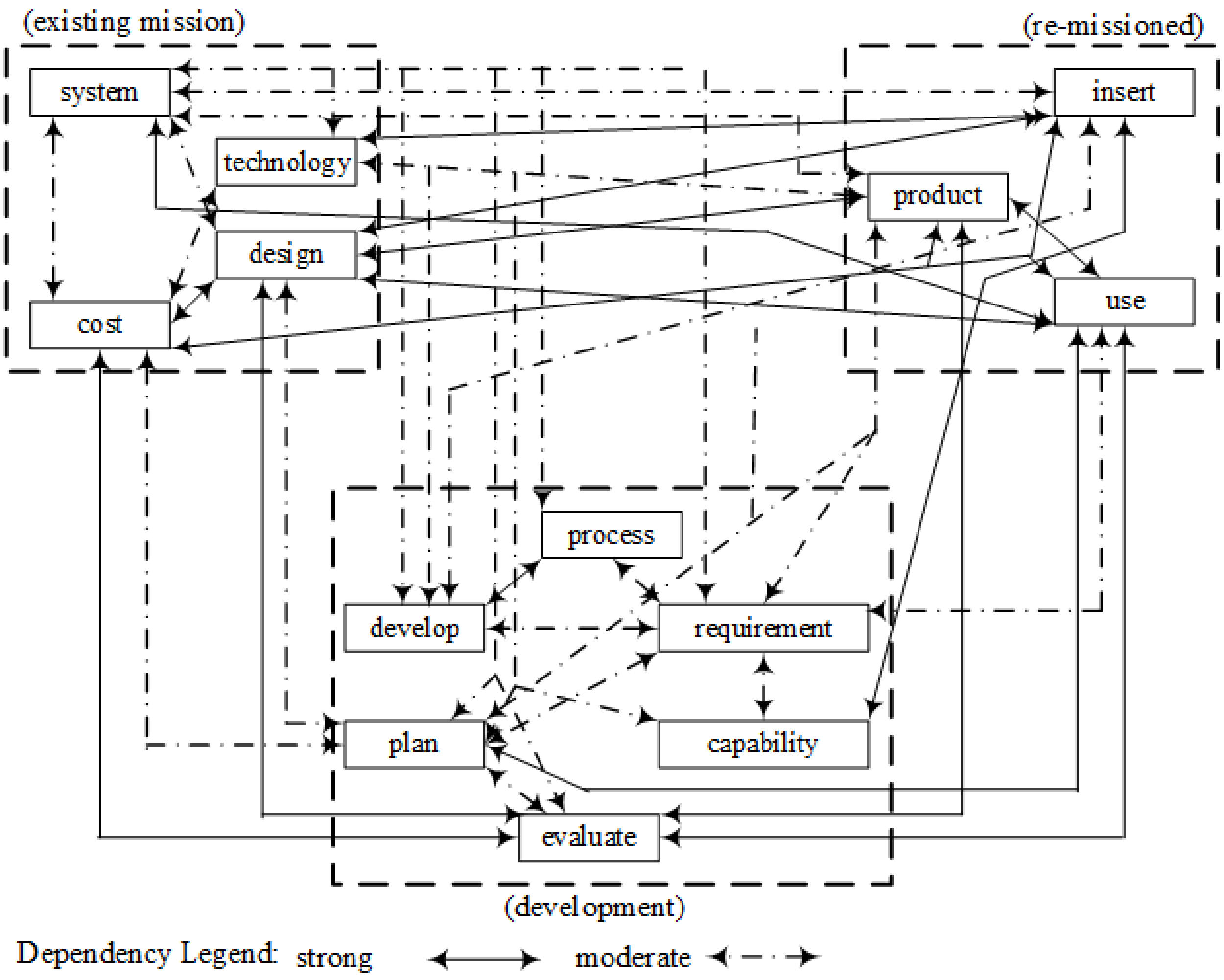


Figure 24. Technology Insertion Design Axiomatic Dependencies.

Table 13. Axiomatic Relationships Between Technology Insertion Design Primitive Concepts.

Composite	Primitive	Dependency	Axiom
existing mission	system	within	System is moderately correlated with technology
		within	System is moderately correlated with design.
		within	System is moderately correlated with cost.
		between	System is moderately correlated with insertion.
		between	System is moderately correlated with product.
		between	System is strongly correlated with use.
		between	System is moderately correlated with develop.
		between	System is moderately correlated with process.
		between	System is moderately correlated with plan.
		between	System is moderately correlated with requirement.
		between	System is moderately correlated with evaluate.
	technology	within	Technology is moderately correlated with system.
		within	Technology is moderately correlated with cost.
		between	Technology is strongly correlated with insertion.
		between	Technology is moderately correlated with product.
		between	Technology is moderately correlated with develop.
		between	Technology is moderately correlated with plan.
		between	Technology is strongly correlated with capability.
	design	within	Design is moderately correlated with system.
		within	Design is strongly correlated with cost.
		between	Design is strongly correlated with insertion.
		between	Design is strongly correlated with product.
		between	Design is strongly correlated with use.
		between	Design is moderately correlated with plan.
		between	Design is strongly correlated with evaluate.
	cost	within	Cost is moderately correlated with system.
		within	Cost is moderately correlated with technology.
		within	Cost is strongly correlated with design.
		between	Cost is strongly correlated with insertion.
		between	Cost is strongly correlated with product.
		between	Cost is strongly correlated with use.
		between	Cost is moderately correlated with plan.
		between	Cost is strongly correlated with evaluate.

Table 13. Axiomatic Relationships Between Technology Insertion Design Primitive Concepts. (continued)

Composite	**Primitive**	**Dependency**	**Axiom**
re-missioned	insert	between	Insertion is moderately correlated with system.
		between	Insertion is strongly correlated with technology.
		between	Insertion is strongly correlated with design.
		between	Insertion is strongly correlated with cost.
		between	Insertion is moderately correlated with develop.
		between	Insertion is moderately correlated with plan.
		between	Insertion is strongly correlated with capability.
	product	within	Product is strongly correlated with use.
		between	Product is moderately correlated with system.
		between	Product is moderately correlated with technology.
		between	Product is strongly correlated with design.
		between	Product is strongly correlated with cost.
		between	Product is moderately correlated with plan.
		between	Product is moderately correlated with requirement.
		between	Product is strongly correlated with evaluate.
	use	within	Use is strongly correlated with product.
		between	Use is strongly correlated with design.
		between	Use is strongly correlated with cost.
		between	Use is strongly correlated with product.
		between	Use is strongly correlated with plan.
		between	Use is moderately correlated with requirement.
		between	Use is strongly correlated with evaluate.
development	develop	within	Develop is strongly correlated with process.
		within	Develop is moderately correlated with requirement.
		between	Develop is moderately correlated with system.
		between	Develop is moderately correlated with technology.
		between	Develop is moderately correlated with insertion.
	process	within	Process is strongly correlated with develop.
		between	Process is moderately correlated with system.
		between	Process is moderately correlated with requirement.

Table 13. Axiomatic Relationships Between Technology Insertion Design Primitive Concepts. (continued)

Composite	Primitive	Dependency	Axiom
development	plan	within	Plan is moderately correlated with requirement.
		between	Plan is moderately correlated with system.
		between	Plan is moderately correlated with technology.
		between	Plan is moderately correlated with design.
		between	Plan is moderately correlated with cost.
		between	Plan is moderately correlated with insertion.
		between	Plan is moderately correlated with product.
		between	Plan is strongly correlated with use.
	requirement	within	Requirement is moderately correlated with develop.
		within	Requirement is moderately correlated with process.
		within	Requirement is moderately correlated with plan.
		within	Requirement is moderately correlated with evaluate.
		between	Requirement is moderately correlated with system.
		between	Requirement is moderately correlated with product.
		between	Requirement is moderately correlated with use.
	capability	between	Capability is strongly correlated with technology.
		between	Capability is strongly correlated with insertion.
	evaluate	within	Evaluate is moderately correlated with plan.
		between	Evaluate is moderately correlated with system.
		between	Evaluate is strongly correlated with design.
		between	Evaluate is strongly correlated with cost.
		between	Evaluate is strongly correlated with product.
		between	Evaluate is strongly correlated with use.

4.4 Core Reference New Technology Insertion Ontology Design

The taxonomic classes of Figure 17 and Table 8 with their corresponding axiomatic relationships of Table 11 were encoded into a technology design ontology in Fluent Editor using its controlled natural language (CNL). Similarly, the taxonomic classes of Figure 17 and Table 9 with their corresponding axiomatic relationships of Table 13 were encoded into a technology insertion design ontology hierarchically referencing the technology design ontology in Fluent Editor. Fluent Editor's controlled natural language (CLN) is a restricted English (simple noun-verb phrases without adjectives) for human communication that encodes ontology semantics consistent with and translatable into description logic, SWRL rules, and OWL standards. Thus, ontologies encoded in Fluent Editor's CLN meet Gruber's criteria of clarity, coherency, extendibility, minimal encoding bias, and minimal ontological commitment. To conform strictly with minimal ontological commitment, only the following hierarchical and axiomatic relationships.

Hierarchical: "is-a" existential.
"has-a" state modification.
Axiomatic: "be moderately correlated with" in accordance with definitions derived from Figures 21 and 23.
"be strongly correlated with" in accordance with definitions derived from Figures 21 and 23.

The ontologies were materialized in OWL2-RL+ and validated with the OWL2-RL+ reasoner. The Fluent Editor CLN technology design ontology encoding is presented in Appendix E, and the CLN technology insertion design ontology encoding is presented in Appendix F. The referenced technology design and technology insertion design taxonomies as represented by Fluent Editor's hierarchical layout tool are presented in Figure 25.

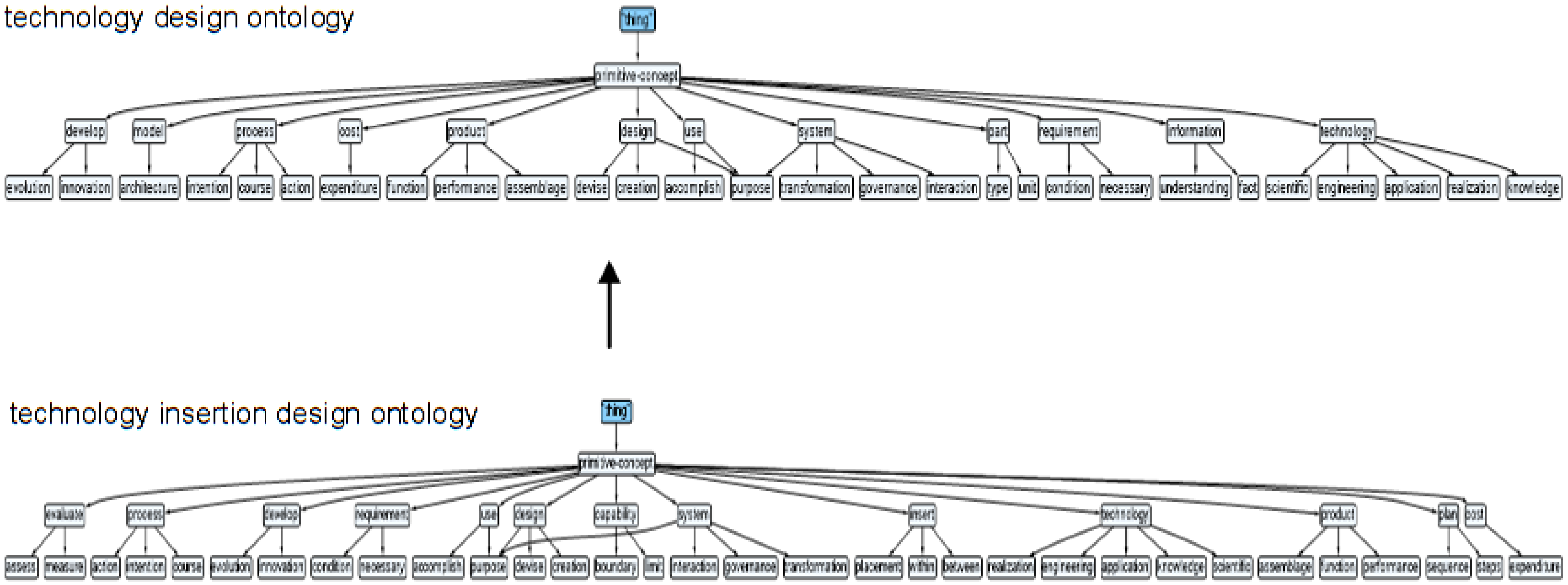


Figure 25. Fluent Editor Hierarchical Layout Tool Taxonomy.

4.5 Proofs of Ontological Concept-Attribute Relationships

Assessment of the technology design ontology against Welty and Guarino's (2001) subsumption criteria for concept "is-a" attributes is set forth in Table 14. The properties of each 'is-a" attribute meets the category criteria specified in Table 1. Table 14 also demonstrates that each primitive concept acts as a primary key for its "is-a" attributes meeting Rector's (2003) normalization criteria necessary and sufficient for *modularity* and *explicitness*.

Table 14. Technology Design "is-a" Attribute Properties.

Primitive Concept	**Existential "is-a" Attribute**	**Attribute Property**	**Property Combination**			
system	governance	Development and application of policies.	+R	+O,–I	+U	-D
	interactions	Reciprocal causality or influence.	+R	+O,–I	+U	-D
	purpose	Reason for existence.	+R	+O,–I	+U	-D
	transformation	Induced change in functionality.	+R	+O,–I	+U	-D
requirement	condition	State of an entity.	+R	+O,–I	+U	-D
	necessary	Essential for existence.	+R	+O,–I	+U	-D
design	creation	Bringing something into existence.	+R	+O,–I	+U	-D
	devise	Invent.	+R	+O,–I	+U	-D
	purpose	Intent.	+R	+O,–I	+U	-D
technology	application	Operational implementation.	+R	+O,–I	+U	-D
	engineering	Application of mathematical and natural laws to create an entity.	+R	+O,–I	+U	+D
	knowledge	Acquisition and application of facts, information, and skills.	+R	+O,–I	+U	+D
	realization	Bringing into being.	+R	+O,–I	+U	-D
	scientific	Natural laws.	+R	+O,–I	+U	-D

Table 14. Technology Design “is-a” Attribute Properties (continued).

Primitive Concept	**Existential “is-a” Attribute**	**Attribute Property**	**Property Combination**			
model	architecture	Designed structure.	+R	+O,–I	+U	+D
cost	expenditure	Value exchanged.	+R	+O,–I	+U	-D
part	type	Defining characteristics.	+R	+O,–I	+U	-D
	unit	Single thing.	+R	+O,–I	+U	-D
product	assemblage	Fitting together parts.	+R	+O,–I	+U	+D
	function	Operational transformation.	+R	+O,–I	+U	+D
	performance	Purpose accomplishment.	+R	+O,–I	+U	+D
use	accomplish	Complete successfully.	+R	+O,–I	+U	-D
	purpose	Reason for existence.	+R	+O,–I	+U	-D
develop	evolution	Self-organizing change.	+R	+O,–I	+U	-D
	innovation	Revolutionary creation.	+R	+O,–I	+U	+D
process	actions	Doing.	+R	+O,–I	+U	-D
	course	Direction or route.	+R	+O,–I	+U	-D
	intention	Bring about.	+R	+O,–I	+U	-D
information	facts	Entity proven to be true.	+R	+O,–I	+U	-D
	understanding	Comprehend.	+R	+O,–I	+U	-D

Assessment of the differential terms in the technology insertion design ontology against Welty and Guarino’s (2001) subsumption criteria for concept “is-a” attributes is set forth in Table 15. The properties of each ‘is-a” attribute meets the category criteria specified in Table 1. Table 15 also demonstrates that each differential primitive concept acts as a primary key for its “is-a” attributes meeting Rector’s (2003) normalization criteria necessary and sufficient for *modularity* and *explicitness*.

Table 15. Technology Insertion Design “is-a” Attribute Properties.

Primitive Concept	**Existential “is-a” Attribute**	**Attribute Property**	**Property Combination**			
insert	between	Into separating two entities.	+R	+O,–I	+U	-D
	placement	Putting into a location.	+R	+O,–I	+U	-D
	within	Inside.	+R	+O,–I	+U	-D
plan	sequence	Order of entities.	+R	+O,–I	+U	-D
	step	Unit difference between adjacent ordered entities.	+R	+O,–I	+U	-D
capability	boundary	Extent of an entity.	+R	+O,–I	+U	+D
	limit	Extent of.	+R	+O,–I	+U	-D
evaluate	assess	Determination of a quality or quantity.	+R	+O,–I	+U	-D
	measure	Assertation amount of a quantity.	+R	+O,–I	+U	+D

Conformance to Formal Concept Analysis’s Complete Lattice Definition, Closure Operator Definition, Basic Theorem on Concept Lattices, and the Spanning Forest Theorem is demonstrated graphically in the technology design ontology concept lattice in Figure 26. Technology design ontology concept lattices in Figures 27 through 38 graphically demonstrate conformance to the Modular Concept Object Definition, Cohesion Definition, Coupling Definitions, and the Primitive Ontology Definition.

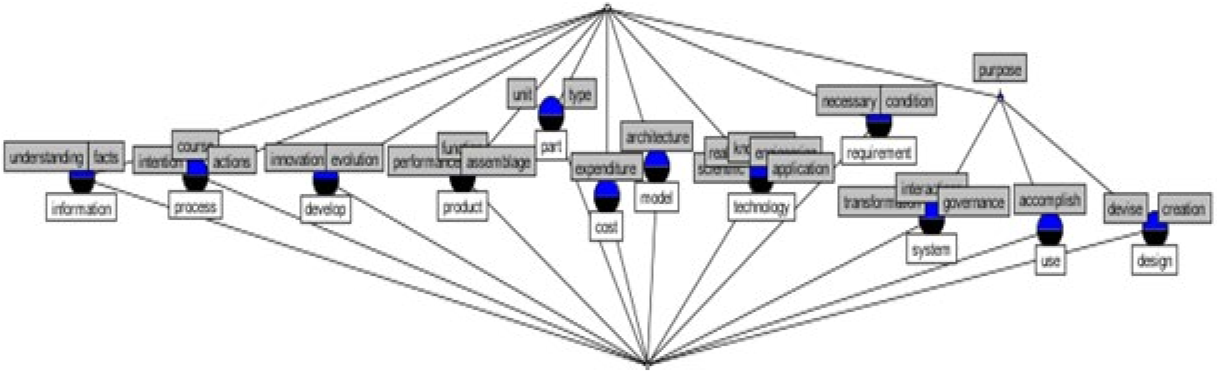


Figure 26. Technology Design Ontology Concept Lattice.

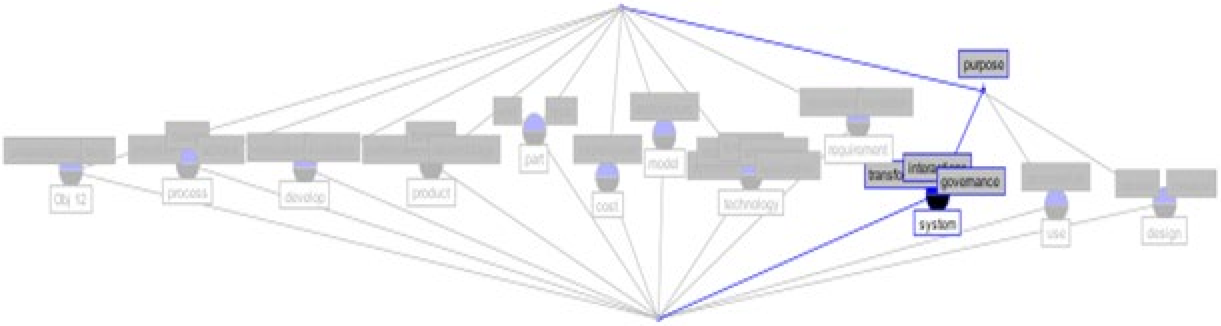


Figure 27. System Concept Primitive Tree.

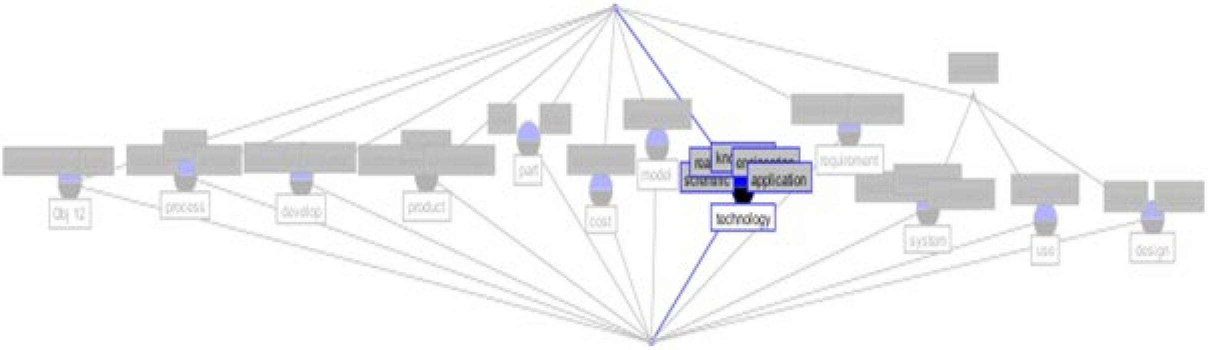


Figure 28. Requirement Concept Primitive Tree.

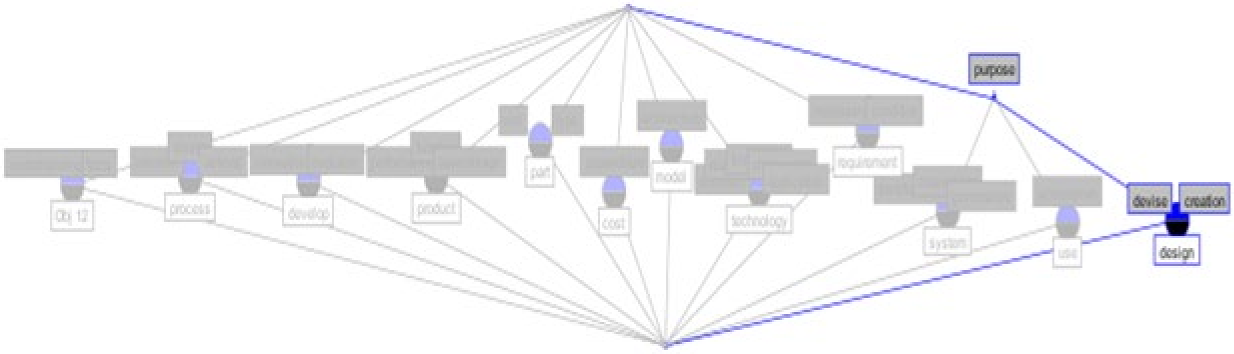


Figure 29. Design Concept Primitive Tree.

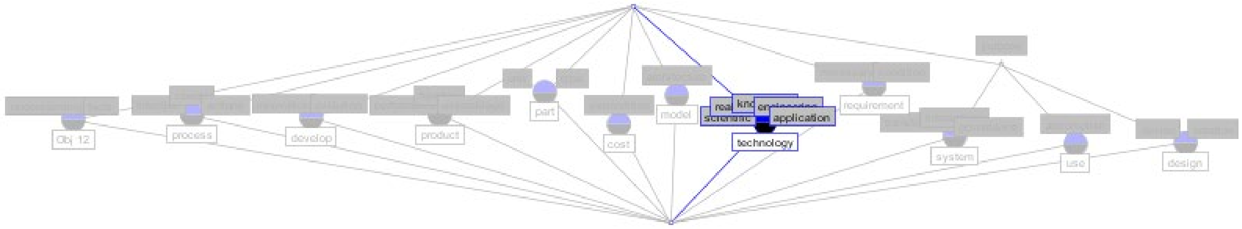


Figure 30. Technology Concept Primitive Tree.

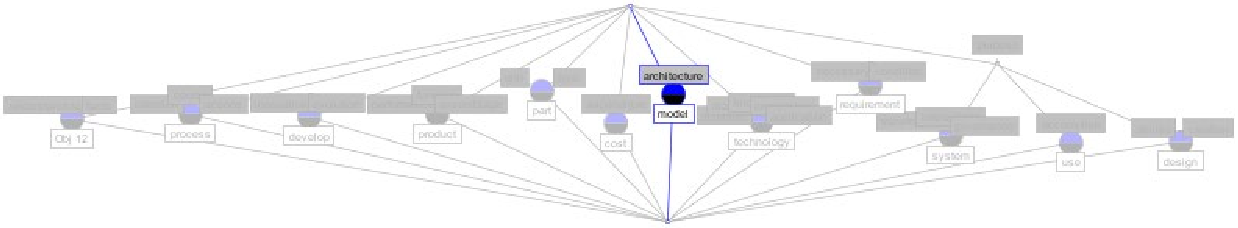


Figure 31. Model Concept Primitive Tree.

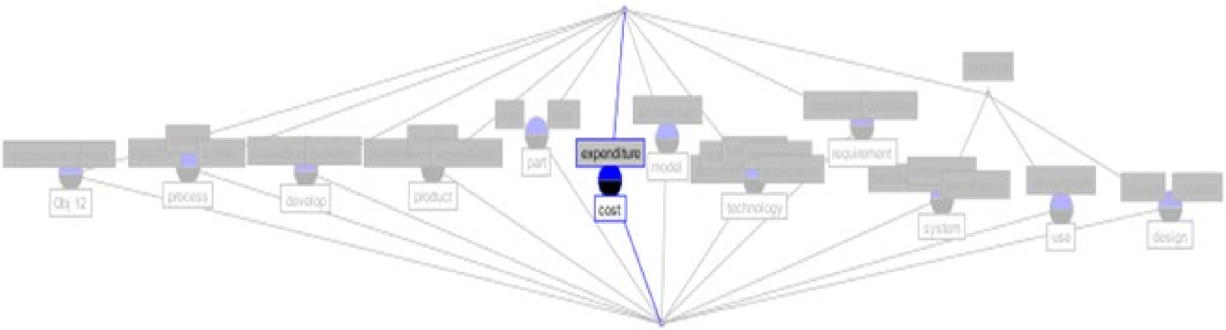


Figure 32. Cost Concept Primitive Tree.

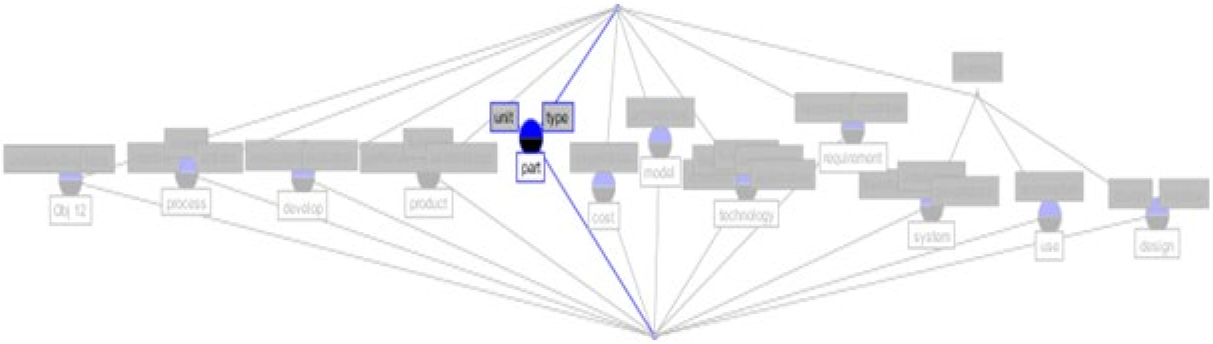


Figure 33. Part Concept Primitive Tree.

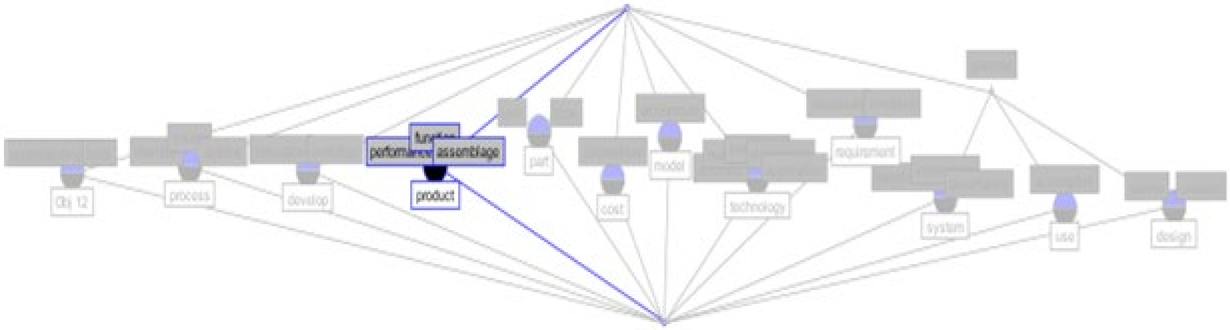


Figure 34. Product Concept Primitive Tree.

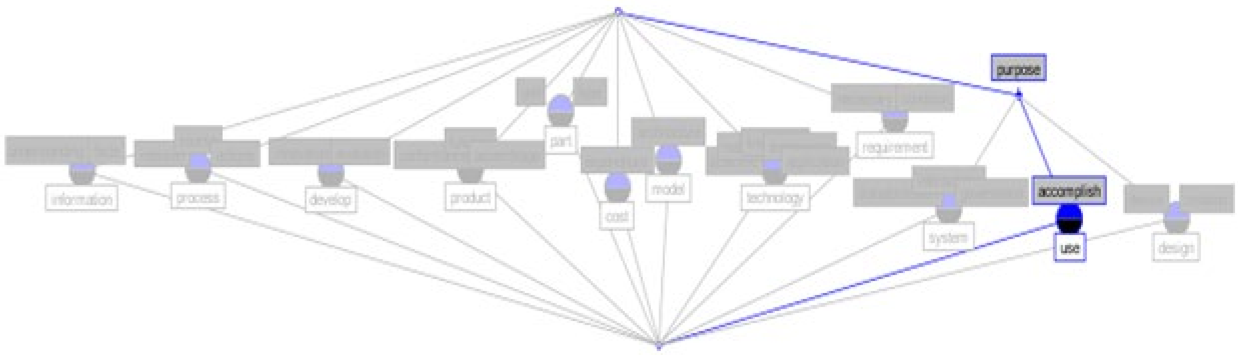


Figure 35. Use Concept Primitive Tree.

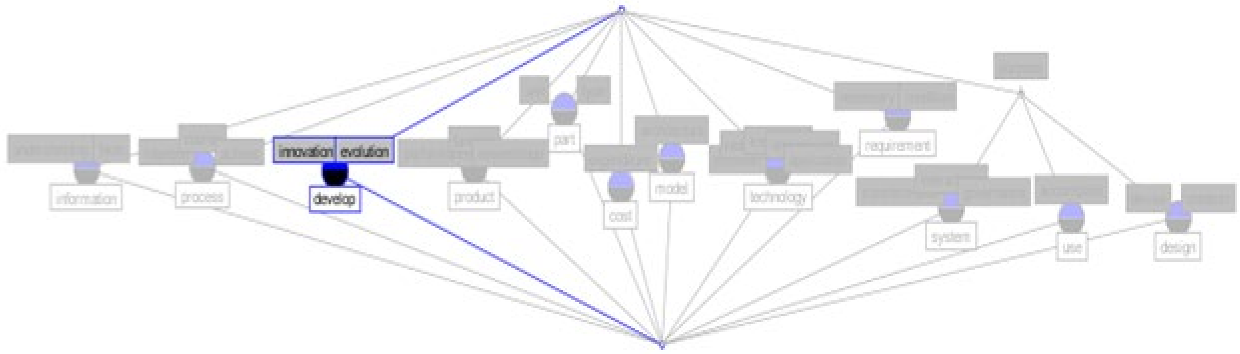


Figure 36. Develop Concept Primitive Tree.

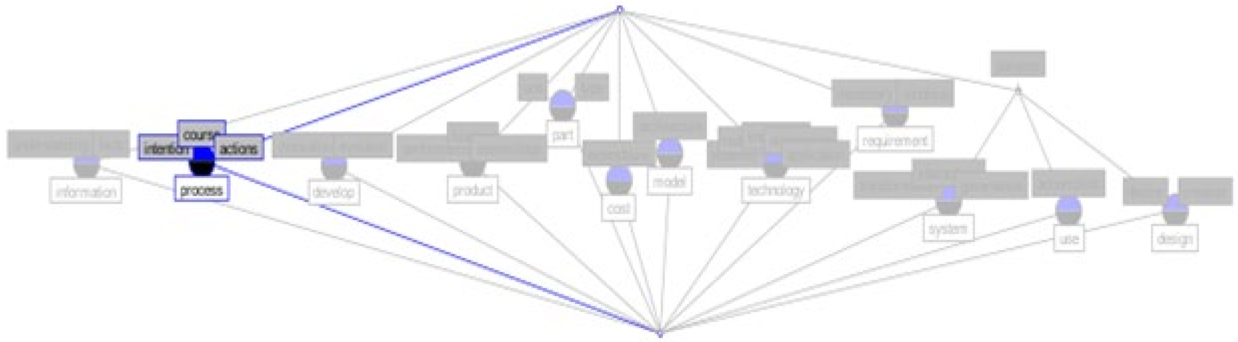


Figure 37. Process Concept Primitive Tree.

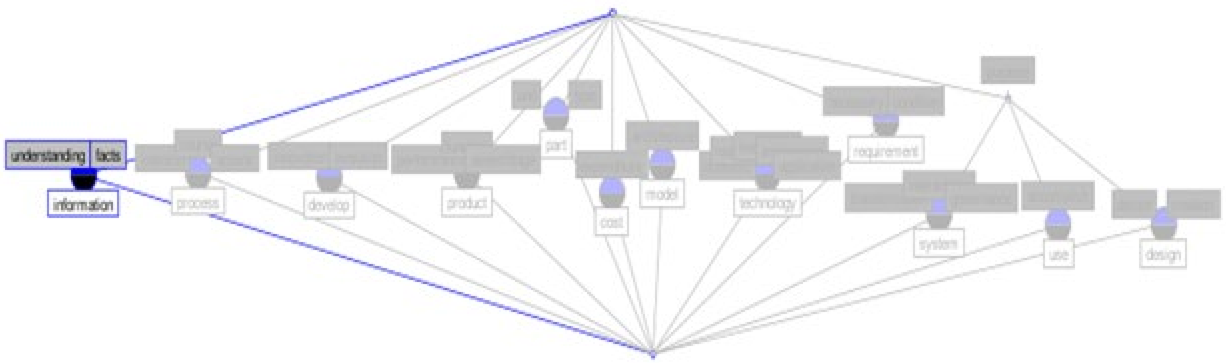


Figure 38. Information Concept Primitive Tree.

Conformance to Formal Concept Analysis's Complete Lattice Definition, Closure Operator Definition, Basic Theorem on Concept Lattices, and the Spanning Forest Theorem is demonstrated graphically in the technology insertion design ontology concept lattice in Figure 39. Technology insertion design ontology concept lattices in Figures 40 through 43 graphically demonstrate conformance to the Modular Concept Object Definition, Cohesion Definition, Coupling Definitions, and the Primitive Ontology Definition for the four differential concepts of insert, plan, capability, and evaluate. Conformance of the remaining concepts were inherited by reference to the corresponding technology design concepts.

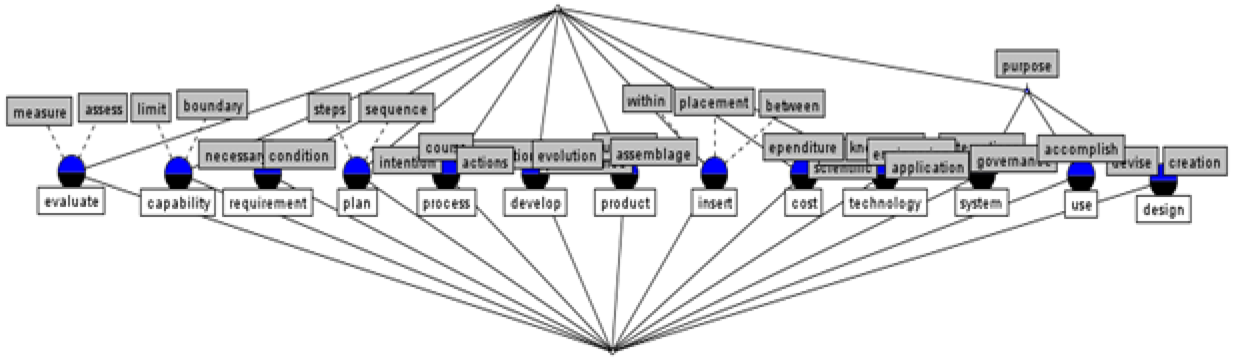


Figure 39. Technology Insertion Design Ontology Concept Lattice.

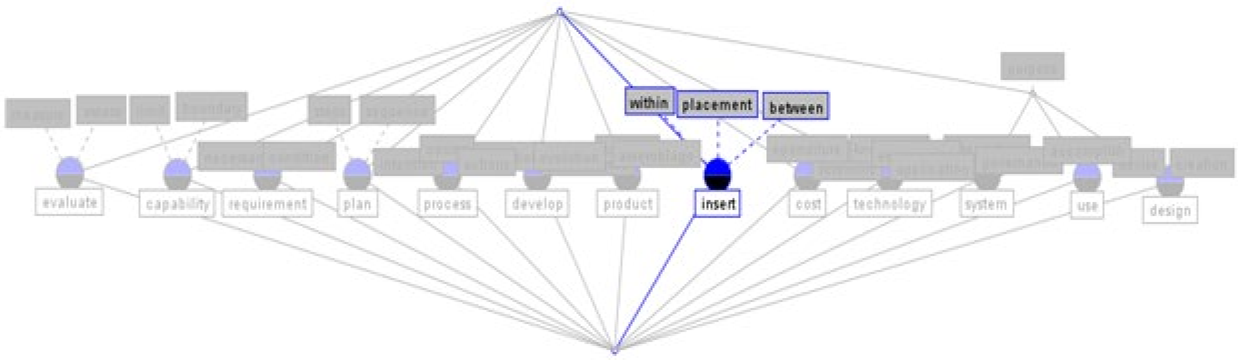


Figure 40. Insert Concept Primitive Tree.

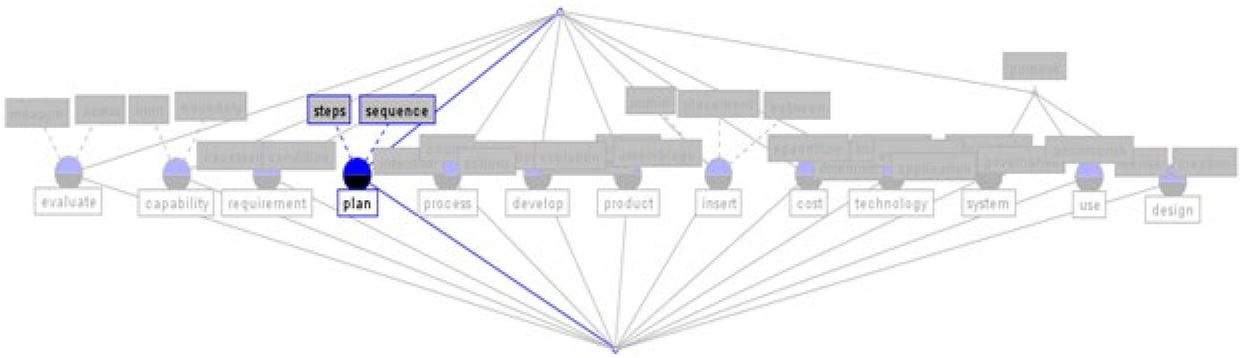


Figure 41. Plan Concept Primitive Tree.

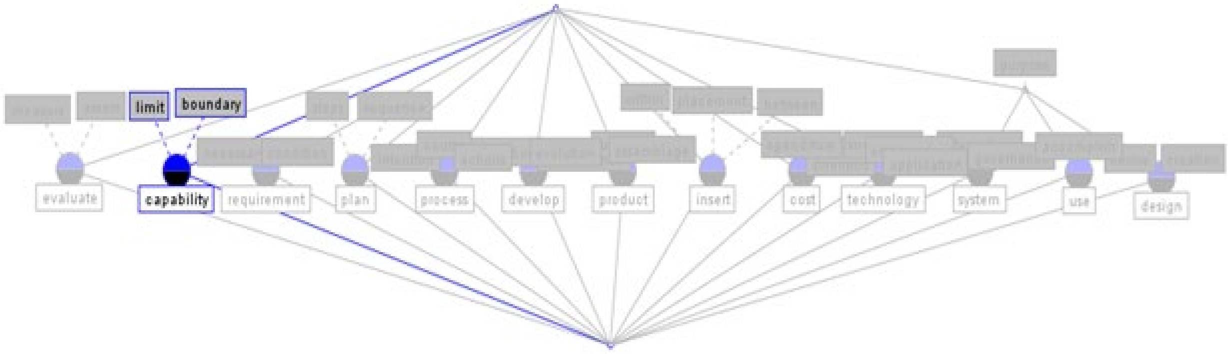


Figure 42. Capability Concept Primitive Tree.

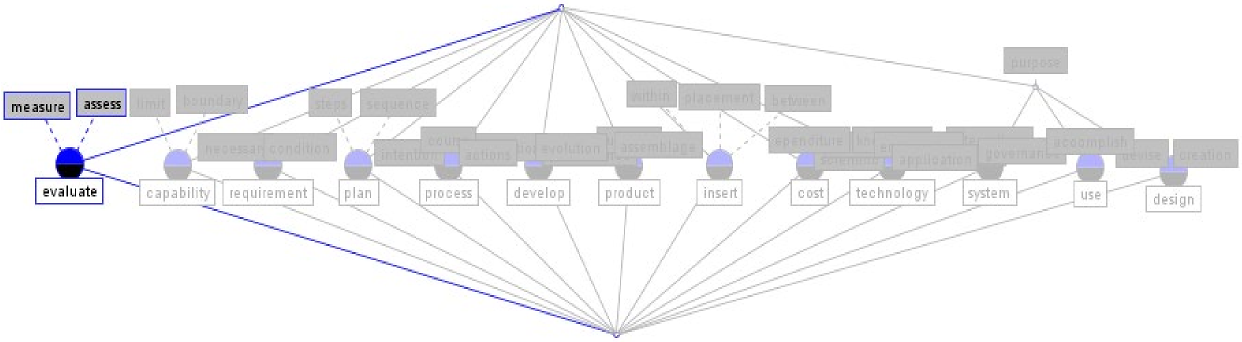


Figure 43. Evaluate Concept Primitive Tree.

CHAPTER 5

DISCUSSION

5.1 Core Reference New Technology Insertion Ontology

The failure of technology insertion design to materialize as a primitive category of the more general technology design taxonomy suggest two conclusions. First, human knowledge, including published bodies of knowledge, is comprised of lexical and semantic inconsistencies and discontinuities. The ontology engineering methodology needs to be generalized to map taxonomic hierarchies as a mix of primitive categories and referenced subsumed primitive ontologies to reflect existing knowledge inconsistencies and discontinuities. Second, the deficiency in the existing systems technology design methodology with respect to technology insertion design is its exclusive focus on developing initial mission capability within cost constraints. Conversely, technology insertion design focuses on re-missioning existing mission capability within cost constraints. Since it is driven by existing mission capability and cost constraints, systems technology design does not adequately consider future technology insertion (and possibly technology lifecycle, obsolescence, roadmapping, and sustainability). Subsequent technology insertion design is constrained within the original technological design limitations.

5.2 Research Implications

Three implications arise directly from the two conclusions of this research. First, the ontology engineering methodology needs to be generalized to map taxonomic hierarchies as a mix of primitive categories and referenced subsumed primitive ontologies to reflect existing knowledge inconsistencies and discontinuities. Once existing knowledge inconsistencies and discontinuities are mapped, the second iterative phase of the ontology engineering methodology must address resolution of identified inconsistencies and discontinuities toward the limiting primitive hierarchical categorical subsumptions.

Second, the design body of knowledge needs to be reorganized around an ontology hierarchy that extends from the foundational ontology level to the expert systems applications level like that illustrated in Figure 1 with intermediate, within-level hierarchies as illustrated in Figure 2. Currently, there is no general design foundational ontology that provides an architecture around which to organize and integrate the disparate design disciplines (architecture, arts, biological, communications and information, computer and software, decision and game, fashion, industrial, instruction pedagogy, interior, landscape, organization, political, process, service, social, strategy, systems, urban, and web). The design ontologies that exist were developed primarily as expert systems within design domain applications, and, as such, contribute to the propagation of design knowledge inconsistencies and discontinuities. Development of a hierarchical design theory and practice ontology will admit mapping of existing knowledge inconsistencies and discontinuities like that identified in this research between technology design knowledge and practice and technology insertion design knowledge and practice. Once existing knowledge inconsistencies and discontinuities are mapped, the second iterative phase of the design ontology engineering methodology must address resolution of identified inconsistencies and discontinuities toward the limiting primitive hierarchical categorical subsumptions. However, development of a hierarchical design theory and practice ontology will require an interdisciplinary effort among the design disciplines.

5.3 Applied New Technology Insertion Implications

The third implication relates directly to resolution of the new technology insertion design problem. Specifically, until future technology insertion (lifecycle, obsolescence, roadmapping, and sustainability) decision and cost drivers are built into the original technology design process, subsequent technology insertion design will continue to be constrained within the original technological design limitations. Original technological design constraints will continue to force limited new technology insertion solutions such as parts purchase to last through the platform's lifecycle including lifetime buys, development of aftermarket sources, and backward compatible technology patches.

5.4 Research Limitations

There were two primary limitations in this original research into engineering a new technology insertion ontology as the organizing architecture for the body of knowledge. First, as noted in section 2.2, the domains for assembling the text corpora for this research were government defense, government non-defense, and commercial. While the specific applications differ greatly, they share common difficulties in the integration of new technology. However, there may also be structural differences between these domains respective solutions to the new technology insertion problem that may have resulted in unmapped bias discontinuities in the resultant ontology.

Second, in the identification of primitive concepts through text mining and content analysis, this research relied exclusively on Zipf's law, which associates a word's semantic meaning importance with its frequency rank within a language corpus. Although there is extensive research supporting the relationship between a word's frequency rank and its semantic meaning across multiple disciplines (Aitchison, Corradi, and Latham, 2016; Furusawa, 2003; Griffin and Bock, 1998; Kanter and Kessler, 1995; Li, 1991; Marinellie and Chan, 2006; Piantadosi, 2014; Powers, 1998), Ferrer-i-Cancho (2014) showed that there is also a linear dependency between a word and the meanings assigned to it in a population's general language. Ferrer-i-Cancho termed this dependency as "… a weak version of the meaning-frequency law…." (p. 28). Lestrade (2017) proposed that Zipf's law "… follows from the interaction of syntax (word classes differing in class size) and semantics (words having to be sufficiently specific to be distinctive and sufficiently general to be reusable). Using a computational model, it is shown that neither of these ingredients suffices to produce a Zipfian distribution on its own and that the results deviate from the Zipfian ideal only in the same way as natural language itself does …." (p. 1). Wyllys (1981) argued, "Practically all the work on developing a rationale for Zipf's law has involved probabilistic models related to the Poisson… it is clear that the process cannot be a pure Poisson process, since the choices of words are not independent …." (p. 62).

CHAPTER 6

CONSLUSIONS

6.1 Primary Contributions of This Research

The primary contribution of this research was the development of the technology insertion design ontology subsumed within the technology design ontology of Figure 17 and its supporting axiomatic dependences of Figure 24. The differences in the technology design and technology insertion design taxonomies of Figure 17 and the axiomatic dependencies in Figure 22 Technology Design Axiomatic Dependencies and Figure 24 Technology Insertion Design Axiomatic Dependencies contributed to understanding the difficulty of the new technology insertion design problem. Specifically, original technology design is driven by existing mission capability and cost constraints and does not consider adequately future technology insertion. Subsequent technology insertion design is constrained within the original technological design limitations forcing adoption of patches such as lifetime buys, development of aftermarket sources, and backward compatible technology fixes.

The second contribution of this research was in the failure of technology insertion design taxonomy to materialize as a primitive category of the more general technology design taxonomy. The ontology engineering methodology needs to be generalized to map taxonomic hierarchies as a mix of primitive categories and referenced subsumed primitive ontologies to reflect inconsistencies and discontinuities in existing bodies of knowledge. Once existing knowledge inconsistencies and discontinuities are mapped, the second iterative phase of the ontology engineering methodology must address resolution of identified inconsistencies and discontinuities toward the limiting primitive hierarchical categorical subsumptions.

The third contribution of this research was the observation that the discontinuity between technology design and technology insertion design reflects the lack of a coherent design body of knowledge. Currently, there is no general design foundational ontology that provides an architecture around which to organize and integrate the disparate design disciplines. The result is inconsistencies and discontinuities among design disciplines bodies of knowledge.

6.2 Widening The Scope

As a consequence of the failure of technology insertion design taxonomy to materialize as a primitive category of the more general technology design taxonomy, technology lifecycle, obsolescence, roadmapping, and sustainability ontologies should be developed to determine whether reference subsumption taxonomic relationships exist between their respective taxonomies and the technology design taxonomy and whether similar differences exist between their axiomatic architectures and that of technology design.

6.3 Future Research

This research will be extended to the development of technology lifecycle, obsolescence, roadmapping, and sustainability ontologies to confirm the existence of reference subsumption taxonomic relationships and differences in axiomatic relationships with respect to the more general technology design ontology. In the development of the above ontologies, this research will also contribute to the generalization of the ontology engineering methodology to mapping taxonomic hierarchies as a mix of primitive categories and referenced subsumed primitive ontologies.

REFERENCES

Ahmed, S. (2005, November 6). Encouraging reuse of design knowledge: a method to index knowledge. *Design Studies*, (26), 565-592.

Ahmed, S., Kim, S., & Wallace, K. (2007). A Methodology for Creating Ontologies for Engineering Design. *Journal of Computing and Information Science in Engineering*, 132-140.

Aitchison, L., Corradi, N., and Latham, P. (2016) Zipf's Law Arises Naturally When There Are3 Underlying, Unobserved Variables. *PLOS Computational Biology*, December. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005110.

Bartels, B., Ermel, U., Sandborn, P., & Pecht, M. (2012). *Strategies for the Prediction, Mitigation and Management of Product Obsolescence*. New York: Wiley.

Boeije, H. (2002). A purposeful approach to the constant comparative method in the analysis of qualitative interviews. Quality & Quantity, 36, 391-409.

Boudreau, M. (2006). *Acoustic Rapid COTS Insertion: A Case Study in Spiral Development*. Monterey, CA: Naval Postgraduate School.

Bowen, G. (2008). Naturalistic Inquiry and the Saturation Concept: A research Note. *Qualitative Research*, 8(1), 137-152

Calude, A. and Pagel, M. (2011) How do we use language? shared patterns in the frequency of word use across 17 world languages. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1567):1101–1107.

Catalano, C., Camossi, E., Ferrandes, R., Cheutet, V., & Sevilmis, N. (2009). A Product Design Ontology for Enhancing Shape Processing in Design Workflows. *Journal of Intelligent Manufacturing*, 20(5), 553-567.

Chandrasekaran, B. and Josephson, J. (1997) Ontology of Tasks and Methods. *Proceedings of the AAAI 1997 Spring Symposium on Ontological Engineering*, March 24-26, Stanford University, CA, USA.

Chen, X., Chen, C., Leong, K., & Jiang, X. (2013). An ontology learning system for customer needs representation in product development. *International Journal of Advanced Manufacturing Technology*, 67, 441-453.

Cohlman, D., & Osborn, G. (2005). Feasibility and roadmap for SCA, wideband, and networking technology insertion into a fielded SDR. *MILCOM 2005 - 2005 IEEE Military Communications Conference*, 835-838.

Corcho, O., Fernandez-Lopez, M., & Gomez-Perez, A. (2003). Methodologies, tools, and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering*, 46, 41-64.

Cristani, M., & Cuel, R. (2005). A Survey on Ontology Creation Methodologies. *International Journal on Semantic Web 7 Information Systems*, 1(2), 48-68.

Davis, D. (1999). Support of COTS and open system architecture modules [in military avionics systems]. *AUTOTESTCON '99. IEEE Systems Readiness Technology Conference*, IEEE, 719-722.

Defense Acquisition University. (2015, September). *Glossary of Defense Acquisition Acronyms & Terms*. 16. Fort Belvoir, VA, US. Retrieved from Glossary of Defense Acquisition Acronyms and Terms.

Denning, J., & Lewis, T. (2017, January). Exponential Laws of Computing Growth. *Communications of the ACM*, 60(1), 54-65.

Durr, D., Goldstein, S., and Zanghì, N. (1992) Quantum Equilibrium and the Origin of Absolute Uncertainty. *Journal of Statistical Physics*, 67: 843–907.

Feldman, K., & Sandborn, P. (2007). Integrating Technology Obsolescence into Product Design. *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference,* Las Vegas, Nevada, USA: ASME, 1-8.

Feng, D., Singh, P., & Sandborn, P. (2007) Lifetime Buy Optimization to Minimize Lifecycle Cost. *Proceedings of the 2007 Aging Aircraft Conference*.

Ferrer-i-Cancho, R. (2016) The meaning-frequency law in Zipfian optimization models of communication. *Glottometrics*, 35, 28-37.

Florentine, S. (2014, October 2). CIOs Should Prepare for Lack of Cobol (Yes, Cobol) Developers. *CIO Magazine*, (IDG) Retrieved January 2, 2017: http://www.cio.com/article/2690555/careers-staffing/cios-should-prepare-for-lack-of-cobol-yes-cobol-developers.html

Furusawa, C. (2003) Zipf's law in Gene Expression. *Physical Review Letters*, 90(8), 88-102.

Ganter, B. and Wille, R. (1999) *Formal Concept Analysis: Mathematical Foundations*. Berlin: Springer-Verlag.

Glaser, B. (1965) The constant comparative method of qualitative analysis. *Social Problems*, 12(4), 436-445.

Glaser, B, and Strauss, L. (1967) *The discovery of grounded theory: Strategies for qualitative research*. New York: Aldine De Gruyter.

Goldstein, S. (1998) Quantum Theories without Observers. *Physics Today*, 51(3), 42– 46 and 51(4), 38–42.

Gomez-Perez, A. (1996). Towards a Framework to Verify Knowledge Sharing Technology. *Expert Systems with Applications*, 519-529.

Gomez-Perez, A. (1999). Evaluation of Taxonomic Knowledge in Ontologies and Knowledge Bases. *Proceedings of the Banff Knowledge Acquisition for Knowledge-Based Systems Workshop KAW'99*. Banff, Alberta, Canada: Ontology Engineering Group, 6.1.1 - 6.1.18.

Gomez-Perez, A. (2001). Evaluation of Ontologies. *International Journal of Intelligent Systems*, 16, 391-409.

Gómez-Pérez, A., Fernandez-Lopez, M., & Corcho, O. (2004). *Ontological Engineering*. London: Springer.

Griffin, Z. and Bock, K. (1998) Constraint, Word Frequency, and the Relationship between Lexical Processing Levels in Spoken Word Production. *Journal of Memory and Language*, 38, 313-338.

Grosse, I., Milton-Benoit, J., & Wileden, J. (2005). Ontologies for supporting engineering analysis models. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 19, 1-18.

Gruber, T. R. (1995). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*, 43 (5-6), 907-928.

Gruber, T., & Olsen, G. (1996). The configuration design ontologies and the VT elevator domain theory. *International Journal of Human-Computer Studies*, 44, 569-598.

Guarino, N. and Welty, C. (2000). A Formal Ontology of Properties. *Proceedings of the ECAI-2000 Workshop on Applications of Ontologies and Problem-Solving Methods*, 97-112.

Guest, G., Bunce, A., and Johnson, L. (2006). How Many Interviews Are Enough? An Experiment with Data Saturation and Variability. *Field Methods*, 18 (1), 59-82

Herald, T. (2000). Technology refreshment strategy and plan for application in military systems a “How-to systems development process” and linkage with CAIV. *National Aerospace and Electronics Conference*, IEEE., 729-736.

Horvath, I., Vergerest, J., & Kuczogi, G. (1998). Development and Application of Design Concept Ontologies for Contextual Conceptualization. *Proceedings of ASME design engineering technical conferences dETC,* New York: ASME, 13-16.

Hsieh, S., Lin, H., Chi, N., Chou, K., & Lin, K. (2011). Enabling the development of base domain ontology through extraction of knowledge from engineering domain handbooks. *Advanced Engineering Informatics*, 25, 288-296.

Johnson, P. (2013) Older programmers are more knowledgeable, but harder to find. *IT World*, May. Retrieved January 3, 2017, from http://www.itworld.com/article/2710015/it-management/older-programmers-are-more-knowledgeable--but-harder-to-find.html

Johnson, W. (2004, Fall). The A-RCI process - Leadership and management principles. *Naval Engineers Journal*, 116(4), 99-105.

Jones, D., Bench-Capon, T., & Visser, P. (1998). Methodologies for Ontology Development. *Proceedings IT&KNOWS Conference of the 15th IFIP World Computer Congress*. Budapest: Chapman-Hall, 62-75.

Kanter, I. and Kessler, A. Markov Processes: Linguistics and Zipf's Law. *Physical Review Letters*, 74(22), 4550-4562.

Kerr, C., Phaal, R., & Probert, D. R. (2008, August). Technology insertion in the defence industry: a primer. *Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture*, 222(8), 1009-1023.

Kim, S., Cohen, M., Netessine, S., & Veeraraghavan, S. (2010, September). Contracting for Infrequent Restoration and Recovery of Mission-Critical Systems. *Management Science*, 56(9), 1551-1567.

Kitamura, & Mizoguchi, R. (2003). Ontology-based description of functional design knowledge and its use in a functional way server. Expert Systems with Applications, 24, 153-166.

Kitamura, & Mizoguchi, R. (2004). Deployment of a functional framework of functional design knowledge. *Advanced Engineering Informatics*, 18, 115-127.

Kitamura, Y., Sano, T., & Mizoguchi, R. (2000). Functional Understanding based on an Ontology of Functional Concepts. *Proceedings of The Sixth Pacific Rim International Conference on Artificial Intelligence (PRJCAI 2000)*, (pp. 723-733).

Konoza, A., & Sandborn, P. (2002). An Evaluation of End of Maintenance Dates for Electronic Assemblies. *Proceedings of the Aircraft Airworthiness & Sustainment Conference*. Baltimore, MD USA.

Konoza, A., & Sandborn, P. (2013). Evaluating the End of Maintenance Dates for Electronic Assemblies Composed of Obsolete Parts. *Journal of Mechanical Design*, 136(3),

Kostoff, R., & Scaller, R. (2001, May). Science and technology roadmaps. *IEEE Transactions on Engineering Management*, 48(2), 132-143.

Kubricky, J. (2008, July-August). *The Rapid Insertion of Technology in Defense*. Defense AT&L, pp. 12-17.

Lestrade, S. (2017) Unzipping Zipf's law. *PLOS Computational Biology*, August. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0181987

Li, W. (1991) Random Texts Exhibit Zipf's Law-Like Word Frequency Distribution. *Santa Fe Institute Working Paper 1991-03-016*.

Liang, V., & Paredis, C. (2004). A Port Ontology for Conceptual Design of Systems. Journal of *Computing and Information Science in Engineering*, 4, 206-217.

Lim, S., Ying, L., & Yong. (2015). Ontology in design engineering: status and challenges. *International Conference on Engineering Design, ICED15*. Politecnico Di Milano, Italy.

Lin, J., Fox, M., & Bilgic, T. (1996). A Requirement Ontology for Engineering Design. *Concurrent Engineering: Research and Application*, 279-291.

Lindig, C. and Snelting, G. (1997) Assessing Modular Structure of Legacy Code Based on Mathematical Concept Analysis. *ICSE '97 Proceedings of the 19th international conference on Software engineering*, Boston, 349-359.

Liu, J., & Hu, X. (2013). A reuse representation model for capturing and formalizing the evolving design rationale. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 27, 401-413.

Liu, Y., Lim, S., & Lee, W. (2013). Product Family Design Through Ontology-Based Faceted Component Analysis, Selection, and Optimization. *Journal of Mechanical Design*, 135(8), 081007-1-17.

López, M., Gómez-Pérez, A., Sierra, J., & Sierra, A. (1999, January/February). Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems*, 37-46.

Manin D. (2008) Zipf's law and avoidance of excessive synonymy. *Cognitive Science*, 32(7):1075– 1098. [PubMed: 21585444]

Marinellie, S. and Chan, Y. (2006) The Effect of Word Frequency on Noun and Verb Definitions: A Developmental Study. *Journal of Speech, Language, and Hearing Research*, 49, 1001-1021.

Mascardi, V., & Paolo, R. (2007). A Comparison of Upper Ontologies. *Proceedings of the Workshop on Objects and Agents* (WOA 2007), (pp. 55-64). Genova, Italy.

Ming, Z., Yan, Y., Wang, G., Panchal, J., Goh, C., Allen, J., & Mistree, F. (2016). Ontology-Based Representation of Design Decision Hierarchies. *Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Charlotte, NC.

Moreland, J., Sarkani, S., & Mazzuchi, T. (2014). Service-Oriented Architecture (SOA) Instantiation within a Hard Real-Time, Deterministic Combat System Environment. *Systems Engineering*, Fall, 17(3), pp. 264-277.

Myers, J., & Sandborn, P. (2007). Integration of Technology Roadmapping Information and Business Case Development into DMSMS-Driven Design Refresh Planning of the V-22 Advanced Mission Computer. *Proceedings of the 2007 Aging Aircraft Conference*.

National Art Education Association. (2014). Retrieved November 3, 2016, from *National Core Arts Standards*: https://www.arteducators.org/ learn-tools/national-visual-arts-standards

National Society of Professional Engineers. (2013). Retrieved November 3, 2016, from *Professional Engineering Body of Knowledge*: https://www.nspe.org/sites/default/files/resources/nspe-body-of-knowledge.pdf

Obrst, L. Chapter 2 Ontological Architectures. In Poli, R., Healy, M., and Kameas, A. (2010) *Theory and Applications of Ontology: Computer Applications*. New York: Springer.

Piantodosi, S. (2014) Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5), 1112-1130.

Poelmans J., Elzinga P., Viaene S., Dedene G. (2010) Formal Concept Analysis in Knowledge Discovery: A Survey. In: Croitoru M., Ferré S., Lukose D. (eds) *Conceptual Structures: From Information to Intelligence. ICCS 2010. Lecture Notes in Computer Science*, vol 6208. Springer, Berlin, Heidelberg, 130-153.

Poelmans, J., Ignatov, D., Kuznetsov, S., and Dedene, G. (2013) Formal Concept Analysis in Knowledge Processing: A Survey on Applications. *Expert Systems with Applications*, 40(16), 6538-6560.

Ponsard, C., Massonet, P., Rifaut, A., Molderez, J., van Lamsweerde, A., & Tran Van, H. (2004). *Early Verification and Validation of Mission Critical Systems*. Universit´e catholique de Louvain, Department of Computer Science. Charleroi, Belgium: CETIC Research Center.

Pope, S., Elliott, J., & Turbini, L. (1998). *Designing for technological obsolescence and discontinuous change: an evaluation of three successional electronic products*. Atlanta, GA, USA: Georgia Institute of Technology.

David M.W. Powers (1998) Applications and Explanations of Zipf's Law. In D.M.W. Powers (ed.) NeMLaP3/CoNLL98: *New Methods in Language Processing and Computational Natural Language Learning*, ACL, 151-160.

Prasad, B. (1997). Re-engineering life cycle management of products to achieve global success in the changing global marketplace. *Industrial Management & Data Systems*, 90-98.

Princeton University (2010) "About WordNet." *WordNet*. Princeton, NJ: Princeton University.

Rector, A. (2003) Modularization of Domain Ontologies Implemented in Description Logics and related formalisms including OWL. In *Proceedings of K-CAP'03*, October 23–25, 2003, Sanibel Island, Florida, USA., 121-128.

Richards, D., & Simoff, S. (2001). Design ontology in context - a situated cognition approach to conceptual modelling. *Artificial Intelligence in Engineering*, 15, 121-136.

Romero-Rojo, F., Roy, R., Shehab, E., & Wardle, P. (2009). Obsolescence Challenges for Product-Service Systems in Aerospace and Defence Industry. *Proceedings of the 1st CIRP Industrial Product-Service Systems (IPS2) Conference* (pp. 255-260). Cranfield University.

Roussey, C., Pinet, F., Kang, P., & Corcho, O. (2011). An Introduction to Ontologies and *Ontology Engineering*. London: Springer.

Sandborn, P. (2005). A Data Mining Based Approach to Electronic Part Obsolescence Forecasting. *Proceedings of the 2005 DMSMS Conference*.

Sandborn, P. (2008). Strategic Management of DMSMS in Systems. DSP Journal, 24-30.

Sandborn, P. (2013). Design for obsolescence risk management. *2nd International Through-life Engineering Services Conference*. Elsevier B.V.

Sandborn, P., & Myers, J. (2008). Designing Engineering Systems for Sustainability. In *Handbook of performability engineering* (pp. 81-104). London: Springer.

Sandborn, P., & Prabhakar, V. (2015). The Forecasting and Impact of the Loss of Critical Human Skills Necessary for Supporting Legacy Systems. *IEEE Transactions on Engineering Management*, 361-371.

Sandborn, P., & Singh, P. (2005). Forecasting Technology Insertion Concurrent with Design Refresh Planning for COTS-Based Electronic Systems. *Proceedings of the Reliability and Maintainability Symposium.* Arlington, VA.

Sandborn, P., Herald, T., Houston, J., & Singh, P. (2003, December). Optimum Technology Insertion into Systems Based on the Assessment of Viability. *IEEE Transactions on Components and Packaging Technologies*, 26(4), 734-738.

Sandborn, P., Prabhakar, V., & Kusimo, A. (2012). Modeling the Obsolescence of Critical Human Skills Necessary for Supporting Legacy Systems. *Proceedings ASME International Design Engineering Conferences & Computers and Information in Engineering Conference* (pp. 1-7). Chicago, IL, USA: ASME.

Shalf, J., & Leland, R. (2015, December). Computing beyond Moore's Law. *Computer*, 14-23.

Siegel, C., Majernik, D., Davis, D., & Foster, C. (1999). Using behavioral mixed-signal simulation to implement COTS rapid technology insertion in avionics systems. AUTOTESTCON '99. IEEE Systems Readiness Technology Conference (pp. 769-773). IEEE.

Sim, S., & Duffy, A. (2003). Towards an ontology for generic engineering design activities. *Research in Engineering Design*, 14, 200-223.

Singh, P., & Sandborn, P. (2006, April-June). Obsolescence Driven Design Refresh Planning for Sustainment-Dominated Systems. *The Engineering Economist*, 51(2), pp. 115-139.

Soininen, T., Tiihonen, J., Mannisto, T., & R., R. (1998). Towards a general ontology of configuration. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 12, 357-372.

Storga, M., Andreasen, M., & Marjanovic, D. (2005). Towards a formal design model based on a genetic design model system. *DS 35: Proceedings ICED 05, the 15th International Conference on Engineering Design*. Melbourne, Australia.

Storga, M., Andreasen, M., & Marjanovic, D. (2005). Towards a Formal Design Model Based on a Genetic Design Model System. DS 35: *Proceedings ICED 05, the 15th International Conference on Engineering Design*, (pp. 15-18). Melbourne, Australia.

Sylvester, J., Konstanzer, G., & Rottier, J. (2001, October-December). Aegis anti-air warfare tactical decision aids. *Johns Hopkins APL Technical Digest*, 22(4), pp. 473-487.

Teng, C., Bernard, L., & Lessing, P. (n.d.). *Technology Refresh of NOAA's Tropical Atmosphere Ocean (TAO) Buoy System*. Stennis Space Center, MS, USA: NOAA National Data Buoy Center.

University of Maryland. (2016). *Center for Advanced Life Cycle Engineering*. Retrieved January 3, 2017, from About CALCE: http://www.calce.umd.edu/general/general.htm

US Department of Defense. (2015, January 7). *Operation of the Defense Acquisition System*. DoDI 5000.02. Washington, DC, US: US Department of Defense.

US Navy. (2008, October 16). SECNAV INSTRUCTION 5000.2D. *Implementation and Operation of the Defense Acquisition System and the Joint Capabilities Integration and Development System*. DASN(RD&A)ALM.

Webber, G. (2002). *Life Cycle Technology Insertion*. Naval Research Advisory Committee. Washington, DC: Assistant Secretary of the Navy (RD&A).

Welty, C, and Guarino, N. (2001) Supporting ontological analysis of taxonomic relationships. *Data and Knowledge Engineering*, 39(1), 51-74.

Weilinga, B., & Schrieber, A. (1997). Configuration design problem solving. *IEEE Expert*, 12(2), 49-56.

Wilkinson, C. (2004, October). Commercial technology & avionics architecture. *IEEE Aerospace and Electronic Systems Magazine*, 19(10), pp. 7-10.

Wilson, R. (1996) *Introduction to Graph Theory*. London: Prentice-Hall.

Witherell, P., Krishnamurty, S., & Grosse, I. (2007, June). Ontologies for Supporting Engineering Design Optimization. *Journal of Computing and Information Science in Engineering*, 7, 141-150.

Wu, J., Shen, Y., Reinhardt, K., Szu, H., & Dong, B. (2013). A Nanotechnology Enhancement to Moore's Law. *Applied Computational Intelligence and Soft Computing*, January 2013.

Wyllys, R. (1981) Empirical and Theoretical Bases of Zipf's Law. *Library Trends*, 30(1), 53-64.

Yang, D., Dong, M., & Miao, R. (2008). Development of a product configuration system with an ontology-based approach. *Computer-Aided Design*, 40, 863-878.

Zipf, G. (1936) *The Psychobiology of Language*. London: Routledge.

Zipf, G. (1949) *Human Behavior and the Principle of Least Effort*. New York: Addison-Wesley.

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Design	Knowledge	Ahmed S	2005	design, knowledge, process, research	Encouraging reuse of design knowledge
Design	Quality	Alben L	2005	design, quality	Defining the criteria for effective interacton design
Design	Participatory	Asaro P	2000	critical, design, participatory, system, theory	The science and politics of participatory design
Design	Product	Bras B	1997	design, environmental, product, realization, systems	Incorporating Environmental Issues in Product Design Realizaiton
Design	Model	Chaloupka A	2011	availability, cost, repair, simulation	Thermal Cycling Ramifications of Lead-free Solder in the Electronic Assembly Repair Process
Design	Obsolescence	Feldman K	2007	forecasting, obsolescence	Integrating Technology Obsolescence Considerations into Product Design Planning
Design	Complexity	Hanseth O	2010	adaptive, design, information, infrastructure, systems, theory	Design theory for dynamic complexity in informatin infrastructures
Design	Combinatorial	Dukes P	2008	combinatorics, design, theory	Combinatorial Design Theory
Design	Environmental	Fitzgerald D	2005	design, development, engineering. environment, product	Beyond Tools: A Design for Environment Process
Design	Reliability	Kleyner A	2005	analysis, cost, optimize, reliability, warranty	Determining Optimal Reliability Targets Through Analysis of Product Validation Cost and Field Warranty Data
Design	Reliability	Kleyner A	2005	analysis, distribution, forecasting, simulation, warranty	A warranty forecasting model based on piecewise statistical distributions and stochastic simulation
Design	Avaiability	Konoza A	2012	avaiability, cost, simuation	An Analysis of the Electronic Assembly Repair Process for Lead-Free Parts Under Combined Loading Conditions

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Design	Model	Liu J	2013	design, knowledge, management, model, reuse	A reuse-oriented representation model for capturing and formalizing the evolving design rationale
Design	Complexity	Schalf J	2015	scaling, technology	Computing Beyond the End of Moore's Law: Is it really the end, and what are the alternatives?
Design	Complexity	Shedroff N	1994	design, field, information, interaction, theory	Information Interaction Design: A Unified Field Theory of Design
Design	Complexity	Wu J	2013	scaling, technology	A Nanotechnology Enhancement to Moore's Law
Ontology	Design	Ahmed S	2007	methodology, ontology, taxonomy	A Methodology for Creating Ontologies for Engineering Design
Ontology	Design	Ahmed S	2007	design, knowledge, ontology, theory	Engineering Design Ontologies
Ontology	Design	Ahmed S	2007	engineering, ontology, optimization	Ontologies for Supporting Engineering Design Optimization
Ontology	Architecture	Berri J	2006	learning, ontology	Ontology-based Framework for Context-aware Mobile Learning
Ontology	Design	Catalano C	2009	design, ontology, product, workflow	A product design ontology for enhancing shape processing in design workflows
Ontology	Requirements	Chen X	2013	development, learning, requirements, product, ontology, system	An ontology learning system for customer needs representation in product development
Ontology	Design	Catalano C	2009	design, ontology, product, workflow	A product design ontology for enhancing shape processing in design workflows
Ontology	Requirements	Chen X	2013	development, learning, requirements, product, ontology, system	An ontology learning system for customer needs representation in product development

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	**Secondary Theme**	**Lead Author**	**Year**	**Key Words**	**Title**
Ontology	Models	Grosse I	2005	analysis, engineering, interoperability, knowledge, models, ontology	Ontologies for Supporting Engineering Analysis Models
Ontology	Design	Gruber T	1996	configuration, design, ontology	The configuration design ontologies and the VT elevator domain theory
Ontology	Design	Horvath I	1998	computer-aided, design, knowledge, ontology	Development and Application of Design Concept Ontologies for Contextual Conceptualization
Ontology	Economics	Jennings C	2015	buy, lifetime, obsolescence, taxonomy, warranty	Taxonomy of Factors for Lifetime Buy
Ontology	Design	Kitamura Y	2003	design, functionality, knoweldge, ontology, support	Ontology-based description of functional design knowledge and its use in functional way server
Ontology	Models	Kitamura Y	2006	artifacts, design, engineering, knowledge, model, ontology	Roles of Ontologies of Engineering Artifacts for Design Knowledge Modeling
Ontology	Models	Kumar P	2008	design, engineering, model, ontology	Design Process Modeling: Towards an Ontology of Engineering Design Activities
Ontology	Models	Lim S	2011	models, ontology, product	A methodology for building a semantically annotated multi-faceted ontology for product family modeling
Ontology	Design	Johnson L	2015	design, engineering, informatics, knowledge, management, ontology	Ontology in Design Engineering: Status and Challenges
Ontology	Design	Lin J	1996	design, ontology, requirement	A Requirement Ontology for Engineering Design
Ontology	Design	Lin J	1997	product, ontology	A Product Ontology

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Ontology	Design	Lin Y	2013	design, product, ontology, optimization	Product Family Design Through Ontology-Based Faceted Component Analysis, Selection,and Optimization
Ontology	Methodology	Nanda J	2006	analysis, concept, ontology, product, semantic, web	A Methodology for Product Family Ontology Development Using Formal Concept Analysis and Web Ontology Language
Ontology	Methodology	Richards D	2001	analysis, concept, computing, design, knowledge, model, ontology	Design ontology in context - a situated cognition approach to conceptual modelling
Ontology	Design	Sandborn P	2007	criteria, evaluation, information, obsolescence	A Taxonomy and Evaluation Criteria for DMSMS Tools, Databases and Services
Ontology	Design	Soinnen T	1998	analysis, concept, configuration, design, knowledge, ontology, product	Towards a general ontology of configuration
Ontology	Design	Storga M	2005	design, genetic, model, ontology, system	Towards a Formal Design Model Based on Genetic Design Model System
Ontology	Cost	Than D	1994	cost, enterprise, model	A Cost Ontology for Enterprise Modelling
Ontology	Design	Witherell P	2007	design, engineering, ontology, optimization	Ontologies for Supporting Engineering Design Optimization
Ontology	Design	Witherell P	2010	design, engineering, knowledge, management, ontology, optimization	Improved knowledge management through first-order logic in engineering design ontologies
Ontology	Design	Yang D	2008	configuration, design, product, ontology	Development of a product configuration system with an ontology-based approach

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Ontology	Design	Zhang W	2008	collaborative, design, engineering, knowledge, representation	Exploring Semantic Web technologies for ontology-based modeling in collaborative engineering design
Ontology	Obsolescence	Zheng L	2013	design, forecast, lifecycle, obsolescence, ontology	Ontology-based Knowledge Representation for Obsolescence Forecasting
Lifecycle	Economics	Feng D	2007	cost, lifecycle, optimizing	Lifetime Buy Optimization to Minimize Lifecycle Cost
Lifecycle	Economics	Kleyner A	2004	cost, lifecycle, optimizing, reliability, warranty	Minimization of Life Cycle Costs Through Optimization of the Validation Program - A Test Sample Size and Warranty Cost Approach
Lifecycle	Economics	Kleyner A	2008	cost, lifecycle, optimizing, reliability, warranty	Minimizing life cycle cost by managing product reliability via validation plan and warranty return cost
Lifecycle	Economics	Prabhakar V	2010	cost, lifecycle, management, product	A Part Total Cost of Ownership Model for Long Life Cycle Electronic Systems
Lifecycle	Economics	Prabhakar V	2011	cost, lifecycle, management, product	A Model for Making Part Sourcing Decision for Long Life Cycle Products
Lifecycle	Economics	Prabhakar V	2013	cost, disruptions, lifecycle, management, product, supply, strategy	A model for comparing sourcing strategies for parts in long life cycle products subject to long-term supply chain disruptions
Lifecycle	Economics	Prabhakar V	2013	cost, lifecycle, management, ownership, part, supply	Optimizing Part Sourcing Strategies for Low-Volume, Long Life Cycle Products, Using Second Sources and Part Hoarding

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Lifecycle	Management	Prasad B	1997	lifecycle, management, product, strategy	Re-engineering life-cycle management of products to achieve global success in the changing marketplace
Lifecycle	Management	Scanff E	2007	lifecycle, maintenance, management, strategy	Life cycle cost impact using prognostic health management (PHM) for helicopter avionics
Obsolescence	Management	Condra L	1999	components, forecasting, process, obsolescence	Combating Electronic Component Obsolescence by Using Common Processes
Obsolescence	Costs	Condra L	2016	components, cost, obsolescence	Electronic Components Obsolescence
Obsolescence	Planning	Feldman	2007	design, lifecycle, obsolescence, planning	Integrating Technology Obsolescence Considerations into Product Design Planning
Obsolescence	Skills	Fossum J	1986	economic, psychological, obsolescence, skills	Modeling the Skills Obsolescence Process: A Psychological/Economic Integration
Obsolescence	Management	Meyer A	2003	management, obsolescence	A Management Approach to Component Obsolescence in the Military Electronic Support Environment
Obsolescence	Management	Munoz R	2015	complexity, lifecycle, obsolescence, system	Key Challenges in Software Application Complexity and Obsolescence Management within Aerospace Industry
Obsolescence	Management	Nelson R	2011	design, lifecycle, management, planning, obsolescence	Modeling Constraints in Design Refresh Planning
Obsolescence	Management	Nelson R	2012	cost, design, lifecycle, management, obsolescence, sustainment	Strategic management of component obsolescence using constraint-driven design refresh planning

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Obsolescence	Management	Pingle P	2015	criteria, decision, model, management, obsolescence, strategy	Selection of obsolescence resolution strategy based on a multi criteria decision model
Obsolescence	Planning	Pope S	1998	development, environment, lifecycle, product, recycle, remanufacture, re-use	Designing for Technological Obsolescence and Discontinuous Change: An Evaluation of Three Successional Electronic Products
Obsolescence	Sustainability	Rojo F	2009	lifecycle, management, obsolescence, sustainment, systems	Obsolescence Management for Long-life Cycle Contracts: State of the Art and Future Trends
Obsolescence	Sustainability	Rojo F	2009	cost, product, management, obsolescence, system	Obsolescence Challenges for Product-Service Systems in Aerospace and Defence Industry
Obsolescence	Design	Sandborn P	2002	cost, design, forecast, obsolescence, optimization	Electronic Part Obsolescence Driven Product Redesign Optimization
Obsolescence	Management	Sandborn P	2004	design, obsolescence, optimization	Beyond Reactive Thinking - We Should be Developing Pro-Active Approaches to Obsolescence Management Too!
Obsolescence	Management	Sandborn P	2006	COTS, function, hardware, obsolescence, technical, software	The Other Half of the DMSMS Problem Software Obsolescence
Obsolescence	Management	Sandborn P	2007	data, design, forecasting, mining, obsolescence	A Data Mining Based Approach to Electronic Part Obsolescence Forecasting
Obsolescence	Management	Sandborn P	2008	economic, management, obsolescence	Trapped on the Technology's Trailing Edge: We're Paying Too Much to Deal with Obsolete Parts

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Obsolescence	Management	Sandborn P	2011	design, economic, management, strategic, system	Forecasting electronic part procurement to enable the management of DMSMD obsolescence
Obsolescence	Management	Sandborn P	2012	cost, design, economic, management, strategic, system	Making Business Cases to Support Obsolescence Management
Obsolescence	Management	Sandborn P	2012	cost, design, human, economic, management, skills, system	Modeling the Obsolescence of Critical Human Skills Necessary for Supporting Legacy Systems
Obsolescence	Management	Sandborn P	2012	cost, design, human, economic, management, obsolescence, skills, system	The Forecasting and Impact of the Loss of Critical Human Skills Necessary for Supporting Legacy Systems
Obsolescence	Management	Sandborn P	2013	design, obsolescence, management, risk	Design for Obsolescence Risk Management
Obsolescence	Management	Shearer R	1975	design, human, obsolescence, management	Manpower Obsolescence: A New Definition and Empirical Investigation of Personal Variables
Obsolescence	Management	Singh P	2002	cost, design, forecast, optimum, refresh, system	Determining Optimum Redesign Plans for Avionics Based on Electronic Part Obsolescence Forecasts
Obsolescence	Management	Singh P		cost, design, forecast, optimum, sustainment	Electronic Part Obsolescence Driven Product Redesign Planning
Obsolescence	Management	Soloman R	2000	forecasting, lifecycle, management, obsolescence, parts	Electronic Part Life Cycle Concepts and Obsolescence Forecasting
Obsolescence	Management	Stear E	2001	acquisition, COTS, diminishing, insertion, obsolescence, parts, refresh, technology	Strategies to Mitigate Obsolescence in Defense Systems Using Commercial Components

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Obsolescence	Management	Tomczykowski W	2003	design, mitigate, obsolescence, strategies	A Study of Component Mitigation Strategies and Their Impact on R&M
Obsolescence	Management	Torresen J	2007	part, obsolescence, risk, system	Parts Obsolescence Challenges for the Electronics Industry
Obsolescence	Management	Whelan K	2000	computer, obsolescence, productivity, usage	Computers, Obsolescence,and Productivity
Obsolescence	Management	Zheng L	2012	design, lifecycle, planning, model, obsolescence, refresh	Design Refresh Planning Models for Managing Obsolescence
Architecture	Planning	Bond G	2004	architecture, open, network	An Open Architecture for Next-Generation Telecommunication Services
Architecture	Framework	DoD	2010	architecture, meta-model	Department of Defense Architectural Framework Version 2.02
Architecture	Planning	Wilkinson C	2004	architecture, cost, COTS, economics, insertion, technology	Commercia1 Technology & Avionics Architecture
Requirements	Metrics	Becker D	2001	cost, design, yield	Integrating Technology Obsolescence Considerations into Producte Design Planning
Requirements	Planning	DoD	2012	capability, requirements	Joint Capabilities Integration and Development System
Requirements	Ontology	Dokken T	2010	analysis, isogeometric, ontology, locally-refined-splines	Requirements from Isogeometric Analysis for Changes in Product Design Ontologies
Requirements	Acquisition	DoD	2015	acquisition, defense, system	Operation of the Defense Acquisition System
Requirements	Optimize	Etienne B	2007	algorithms, analysis, cost, embedded, genetic, integral, optimization, passives	Optimizing Embedded Passive Content in Printed Circuit Boards

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Requirements	Automation	Fiksel J	1993	automation, concurrent, design, engineering, management, requirements, systems	Computer-aided Requirements Management
Requirements	Contracting	Kim S	2010	disaster, maintenance, mission-critical, recovery, service, support, systems	Contracting for Infrequent Restoration and Recovery of Mission-Critical Systems
Requirements	Engineering	Ponsard C	2004	engineering, goal, monitoring, requirements, validation, verification	Early Verification and Validation of Mission Critical Systems
Sustainability	System	Agripino M	2002	enterprise, military, model, sustainment, systems	A Lean Sustainment Enterprise Model for Military Systems
Sustainability	Design	Blevis E	2006	design, sustainment	Advancing Sustainable Interaction Design
Sustainability	Design	Blevis E	2007	design, sustainment	Sustainable Interaction Design
Sustainability	Design	Bonanni L	2011	design, human factors	Sustainable Interaction Designing Professional Domains
Sustainability	Design	Diegel O	2010	design, product, quality, sustainment	Tools for Sustainable product Design: Additive Manufacturing
Sustainability	Mapping	DiSalvo C	2010	sustainment	Mapping the Landscape of Sustainable HCI
Sustainability	Design	Huang E	2008	design, e-waste, sustainability	Breaking the Disposable Technology Paradigm: Opportunities for Sustainable Interaction Design for Mobile Phones
Sustainability	Design	Jung H	2010	artifacts, design, digital, material, sustainability	Conceptualizations of the Materiality of Digital Artifacts and the Implication for Sustainable Interaction Design
Sustainability	Design	Konoza A	2012	COTS, demand, forecasting, legacy, sustainment, systems	An Evaluation of End of Maintenance Dates for Electronic Assemblies

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Sustainability	Design	Konoza A	2013	COTS, demand, forecasting, legacy, obsolescence, sustainment, systems	Evaluating the End of Maintenance Dates for Electronic Assemblies Composed of Obsolete Parts
Sustainability	Design	Sandborn P	2008	availability, evolving, obsolescence, reliability, requirements, system, warranty	Designing Engineering Systems for Sustainability
Sustainability	Design	Sandborn P	2012	sustainability	Sustainability/Sustainment Definition
Sustainability	Design	Sandborn P	2014	analysis, lifecycle, product, optimize	Development of a Maintenance Option Model to Optimize Offshore Wind Farm Sustainment
Sustainability	Design	Wever R	2008	design, economic, product, sustainment, user	User-centered Design for Sustainable Behavior
Sustainability	Design	Yang R	2014	eco-interaction, information, smart, technology	Making Sustainability Sustainable: Challenges in the Design of Eco-Interaction Technologies
Insertion	Development	Boudreau M	2006	COTS, development, insertion, open, spiral, systems	Acoustic Rapid COTS Insertion Spiral Development
Insertion	Development	Boudreau M	2007	COTS, development, insertion, modular, open, systems	Acoustic Rapid COTS Insertion Modular Development
Insertion	Leadership	Coombs C		acquisition, leadership, risk, technology, insertion	Acquisition Leaders for Rapid Technology Insertion Programs
Insertion	Process	DoD	2010	insertion, process, technology	SOW - Rapid Technology Insertion Process Description Introduction
Insertion	Planning	GAO	2015	acquisition, insertion, obsolescence, strategy	Space Based Infrared System Could Benefit from Technology Insertion Planning
Insertion	COTS	Julian C	2011	COTS, evaluation, insertion, technology	Commercial-Off-The-Shelf Selection Process
Insertion	Management	Kerr C	2008	defense, insertion, management, obsolescence, technology	Technology insertion in the defence industry: a primer

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Insertion	Management	Kubricky J	2008	defense, insertion, management, technology	The Rapid Insertion of Technology in Defense
Insertion	Economics	Marrow R	1998	cost, insertion, strategy, technology	High Precision IFOG Insertion Into the Strategic Submarine Navigation System
Insertion	Viability	Sandborn P	2003	cost, evolvability, producibility, supportability, system, viability	Optimum Technology Insertion into Systems Based on the Assessment of Viability
Insertion	Design	Sandborn P	2004	design, cost, COTS, insertion, lifecycle, optimize, planning, system	Forecasting Technology Insertion Concurrent with Design Refresh Planning for COTS-Based Electronic Systems
Insertion	Design	Singh P	2004	analysis, cost, COTS, forecast, obsolescence, sustainment, systems	Forecasting Technology Insertion Concurrent with Design Refresh Planning for COTS-Based Obsolescence Sensitive Sustainment-dominated Systems
Insertion	Design	Stocker M	2010	capability, cost, effectiveness, management, obsolescence, optimize, risk	Technology Insertion and Management Options for Canadian Forces
Planning	Technology	Garcia M	1997	planning, roadmapping, technology	Integrating Technology Obsolescence Considerations into Producte Design Planning
Planning	Roadmapping	Bray O.	1997	planning, roadmapping, technology	Technology Roadmapping; Integration of Strategic & Technology Planning
Planning	Roadmapping	Bray O.	1997	planning, roadmapping, technology	Fundamentals of Technology Roadmapping
Planning	Technology	Carvalho M	2013	roadmapping, technology	Overview of the literature on technology roadmapping
Planning	Technology	Cho Y	2016	roadmapping, forecasting, planning	An industrial technology roadmap for supporting public R&D planning

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Planning	Forecasting	Coates V	2001	forecasting, technology	On the Future of Technological Forecasting
Planning	Open Architecture	Cohlman D	2005	architecture, insertion, open, roadmapping, strategy, technology	Feasibility and Roadmap for SCA, Wideband, and Networking Technology Insertion in to a Field SDR
Planning	Forecasting	Denning P		exponential growth, forecasting, technology jumping	Exponential Laws of Computing Growth
Planning	Management	DoD	2015	capabilities, integration, system	Joint Capabilities Integration and Development System
Planning	Management	DoD	2015	acquisition, management	Operation of the Defense Acquisition
Planning	Roadmapping	Dougherty J	2003	components, cost, passive , performance, roadmapping	The NEMI Roadmap: Integrated Passives Technology and Economics
Planning	Economics	Feldman K	2008	cost, electronics, model, return-on-investment	The Analysis of Return on Investment for PHM Applied to Electronic Systems
Planning	Economics	Feldman K	2008	cost, electronics, model, return-on-investment	A Methodology for Determining the Return on Investment Associated with Prognostics and Health Management
Planning	Engineering	Fitzgerald D	2005	design, development, engineering, environment, product	A Design for Environmental Process
Planning	Roadmapping	Galvin R	2004	roadmapping, science, technology	Roadmapping - A practitioner's update
Planning	Roadmapping	Garcia M	1997	planning, roadmapping, technology	Fundamentals of Technology Roadmapping
Planning	Roadmapping	Gerdsri N	2007	development, envelope, roadmapping, technology	An Analytical Approach to Building a Technology Development Envelope (TDE) for Roadmapping of Emerging Technologies

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Planning	Modularity	Goure D	2006	design, engineering, information, modularity, technology	Modularity, The Littoral Combat Ship and the Future of the United States Navy
Planning	Roadmapping	Gerdsri N	2009	dynamic, planning, roadmapping, strategy, technology	Dealing with the dynamics of technology roadmapping implementation
Planning	Availability	Haddad G	2011	availability, design, maintainability, reliability	Guaranteeing high availability of wind turbines
Planning	Economics	Haddad G	2011	availability, components, decision, economic, maintenance, optimization	Using Real Options to Manage Condition-Based Maintenance Enabled PHM
Planning	Economics	Haddad G	2014	condition-based, cost, decision, lifecycle, maintenance, options, real	Using maintenance options to maximize the benefits of prognostics for wind farms
Planning	Management	Herald T	2000	refreshment, strategy, technology	Technology Refreshment Strategy and Plan for Application in Military Systems
Planning	Availability	Jazouli T	2010	availability, design	A Design for Availability Approach for Use with PHM
Planning	Availability	Jazouli T	2011	availability, design, requirements	Using PHM to Meet Availability-Based Contracting Requirements
Planning	Availability	Jazouli T	2014	availability, design, requirements	A Direct Method for Determining Design and Support Parameters to Meet Availability Requirement
Planning	Roadmapping	Jin G	2015	analysis, patent, quality-function-deployment, technology, text mining	Technology-driven roadmaps for identifying new product/market opportunities: Use of text mining and quality function deployment
Planning	Economics	Johnson W	2004	cost, economics, modernization, refresh	The A-RCI Process - Leadership and Management Principles

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Planning	Roadmapping	Kajikawa Y	2008	emerging, forecasting, renewable, sustainable	Tracking emerging technologies in energy research: Toward a roadmap for sustainable energy
Planning	Roadmapping	Kostoff R	2001	analyses, insertion, roadmaps, science, technology	Science and Technology Roadmaps
Planning	Roadmapping	Lee J	2012	credibility, communications, roadmaps, theory	An analysis of factors improving technology roadmap credibility: A communications theory assessment of roadmapping processes
Planning	Roadmapping	Lee S	2008	development, products, roadmaps	Using patent information for designing new product and technology: keyword-based technology roadmapping
Planning	Economics	Lillie, E	2015	cost, modeling, reliability	Assessing the value of a lead-free solder control plan using cost-based FMEA
Planning	Roadmapping	Linton J	2004	disruptive, roadmapping, sustainability, technology	Roadmapping: from sustaining to disruptive technologies
Planning	Replacement	Luke J	1999	COTS, replacement, strategy	Replacement Strategy for Aging Avionics Computers
Planning	Upgrade	Madisetti V	2000	COTS, legacy, upgrade, strategy	On Upgrading Legacy Electronic Systems: Methodology, Enabling Technologies & Tools
Planning	Roadmapping	Martin H	2012	analysis, hierarchy, roadmap, service, technology	Technology roadmap development process (TRDP) for the service sector: A conceptual framework
Planning	Economics	Moreland J	2009	adaptive, complex, cost, COTS, strategy, system	Structuring a Flexible Affordable Naval Force to Meet Strategic Demand in the 21st Century

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Planning	Economics	Moreland J	2014	architecture, deterministic, latency, network, quality, real-time, service	Service-Oriented Architecture (SOA) Instantiation within a Hard, Real-Time Deterministic Combat Environment
Planning	Roadmapping	Myers J	2007	analysis, business, case, design, obsolescence, roadmapping, technology	Integration of Technology Roadmapping Information and Business Case Development into DMSMS-Driven Design Refresh Planning of the V-22 Advanced Mission Computer
Planning	Roadmapping	Pecht M	2010	management, prognostics, roadmap	A prognostics and health management roadmap for information and electronics-rich systems
Planning	Roadmapping	Petrick I	2004	heuristic, information, management, roadmap, supply, sustainable	Technology roadmapping in review: A tool for making sustainable new product development decisions
Planning	Roadmapping	Phaal R	2001	business, integration, planning, product, roadmap, technology	Characterization of Technology Roadmaps: Purpose and Format
Planning	Management	Phaal R	2001	acquisition, exploitation, framework, identification, innovation, management, process, protection, selection, technology	A framework for supporting the management of technologically innovation
Planning	Roadmapping	Phaal R	2001	business, planning , product, roadmap, strategy, technology	Technology Roadmapping: Linking technology resources to business strategy
Planning	Roadmapping	Phaal R	2004	evolution, revolution, roadmap, technology	Technology roadmapping - A planning framework for evolution and revolution

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Planning	Roadmapping	Phaal R	2005	align, goals, roadmap, technology	Developing a Technology Roadmapping System
Planning	Economics	Ragan	2002	cost, performance, trade-off	A Detailed Cost Model for Concurrent Use With Hardware/Software Co-Design
Planning	Economics	Rajagopal S	2014	cost, maintenance, obsolescence	Software obsolescence in defence
Planning	Management	Redling T	2004	lifecycle, service, upgrade	Considerations for Upgrading Aging Military Avionics Systems with State-of-the-Art Technology
Planning	Roadmapping	Rinne M	2004	innovation, roadmap, technology	Technology roadmaps: Infrastructure for innovation
Planning	Testing	Salzano L	2005	reliability, testing	Environmental Qualification Testing and Failure Analysis of Embedded Resistors
Planning	Packaging	Sandborn P	1998	analysis, packaging, prototyping, optimizing, system	Analyzing Packaging Trade-offs During System Design
Planning	Economics	Sandborn P	1998	analysis, cost, design, material, model, tradeoff	Material-Centric Modeling of PWB Fabrication: An Economic and Environmental Comparison of Conventional and Photovia Board Fabrication Processes
Planning	Economics	Sandborn P	1999	assembly, cost, design, disassembly, model, prototype, recycle	A Model for Optimizing the Assembly and Disassembly of Electronic Systems
Planning	Economics	Sandborn P	2000	analysis, cost, design, modules, routing	A comparison of routing estimation methods for microelectronics modules
Planning	Economics	Sandborn P	2001	analysis, cost, design, model	Analysis of the Cost of Embedded Passives in Printed Circuit Boards

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Planning	Economics	Sandborn P	2001	analysis, cost, design, model, tradeoff	Application-Specific Economic Analysis of Integral Passives in Printed Circuit Boards
Planning	Knowledge	Sandborn P	2001	design, economic, education, packaging, technology	Progress on Internet-Based Educational Material Development for Electronic Products and Systems Cost Analysis
Planning	Economics	Sandborn P	2002	analysis, cost, design, model	An Assessment of the Applicability of Embedded Resistor Trimming and Rework
Planning	Economics	Sandborn P	2003	analysis, cost, design, model	Cost and production analysis for substrates with embedded passives
Planning	Economics	Sandborn P	2003	analysis, cost, design, model	A Review of the Economics of Embedded Passives
Planning	Management	Sandborn P	2005	cost, decision, management, model, prognostic, system	A Decision Support Model for Determining the Applicability of Prognostic Health Management (PHM) Approaches to Electronic Systems
Planning	Knowledge	Sandborn P	2006	analysis, cost, design, knowledge, model	Using Teardown Analysis as a Vehicle to Teach Electronic Systems Manufacturing Modeling
Planning	Design	Sandborn P	2007	analysis, cost, design, model	Cost Model for Assessing the Transition to Lead-Free Electronics
Planning	Economics	Sandborn P	2007	analysis, cost, design, model	DMSMS Lifetime Buy Characterization Via Data Mining of Historical Buys
Planning	Design	Sandborn P	2007	condition, lifecycle, prognostic, reliability, safety, systems	Introduction to Special Section on Electronic Systems Prognostics and Health Management

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Planning	Design	Sandborn P	2008	design, disruption, economic, finite, model, resource, reuse	The Application of Product Platform Design to the Reuse of Electronic Components Subject to Long-Term Supply Chain Disruptions
Planning	Design	Sandborn P	2008	analysis, design, strategy	A Random Trimming Approach for Obtaining High-Precision Embedded Resistors
Planning	Design	Sandborn P	2008	design, economic, management, strategic, system	Strategic Management of DMSMS in Systems
Planning	Economics	Sandborn P	2008	cost, economics, management, ROI	The Economics of Prognostics and Health Management
Planning	Knowledge	Sandborn P	2006	analysis, cost, design, knowledge, model	Using Teardown Analysis as a Vehicle to Teach Electronic Systems Manufacturing Cost Modeling
Planning	Economics	Sandborn P	2010	cost, economics, management, ROI	Calculating the Return on Investment (ROI) for DMSMS Management
Planning	Economics	Shi Z	2003	cost, economics, manufacturing, model, process, test	Modeling Test Diagnosis, and Rework Operations and Optimizing Their Location in General Manufacturing Processes
Planning	Economics	Shi Z	2003	algorithms, cost, economics, genetic, manufacturing, model, process, test	Optimization of Test/Diagnosis/Rework Location(s) and Characteristics in Electronic Systems Assembly Using Real-Coded Genetic Algorithms
Planning	Economics	Shi Z	2006	algorithms, cost, economics, genetic, manufacturing, model, process, test	Optimization of Test/Diagnosis/Rework Location(s) and Characteristics in Electronic Systems Assembly

APPENDIX A

Article Assignment to Corpus Primary and Secondary Themes (continued)

Primary Theme	Secondary Theme	Lead Author	Year	Key Words	Title
Planning	Design	Singh P	2006	analysis, cost, COTS, forecast, obsolescence, sustainment, systems	Obsolescence Driven Design Refresh Planning for Sustainment-Dominated Systems
Planning	Management	Swarminathan R	2003	reliability	Reliability Assessment of Delamination in Chip-to-Chip Bonded MEMS Packaging
Planning	Management	Sylvester J	2001	aid, decision	Aegix Anti-Air Warfare Tactical Decision Aids
Planning	Economics	Trichy T	2001	cost, economics, model, optimize	A New Test/Diagnosis/Rework Model for Use in Technical Cost Modeling of Electronic Systems Assembly
Planning	Roadmapping	Vishnevskiy K	2016	innovation, planning, roadmap, strategy, technology	Integrated roadmaps for strategy management and planning
Planning	Design	Wright M	1997	design, performance, reliability, specification	Uprating Electronic Components for Use Outside Their Temperature Specification Limits

APPENDIX B

R Text Mining Code

```
> install.packages("tm")
> library(tm)
> install.packages("SnowballC")
> library(SnowballC)
> install.packages("ggplot2")
> library(ggplot2)
> install.package(“cluster”)
> library(cluster)
> install.packages(“fpc”)
> library(fpc)
> cname <- file.path("H:", "MichaelK_LitCorpus")
> cname
[1] "D:/MichaelK_LitCorpus"
> docs <- VCorpus(DirSource(cname))
> docs <- tm_map(docs, content_transformer(tolower))
> docs <- tm_map(docs, removeNumbers)
> docs <- tm_map(docs, removePunctuation)
> docs <- tm_map(docs, removeWords, stopwords("english"))
> docs <- tm_map(docs, stemDocument)
> dtm <- DocumentTermMatrix(docs)
> tdm <- TermDocumentMatrix(docs
> #
> freq <- colSums(as.matrix(dtm))
> ord <- order(freq)
> freq <- sort(colSums(as.matrix(dtm)), decreasing=TRUE)
> docs <- tm_map(docs, removeWords, "common_word")
> #
> dtm <- DocumentTermMatrix(docs)
> tdm <- TermDocumentMatrix(docs)
> dtm
<<DocumentTermMatrix (documents: 205, terms: 19605)>>
Non-/sparse entries: 163044/3855981
Sparsity           : 96%
Maximal term length: 110
Weighting          : term frequency (tf)
> freq <- colSums(as.matrix(dtm))
> ord <- order(freq)
> freq <- sort(colSums(as.matrix(dtm)), decreasing=TRUE)
```

APPENDIX B

R Text Mining Code (continued)

```
> #
> p <- ggplot(subset(wf, freq>3000), aes(x = reorder(word, -freq), y = freq)) +
+ geom_bar(stat = "identity") +
+ theme(axis.text.x=element_text(angle=45, hjust=1))
> p
> #
> dtmss05 <- removeSparseTerms(dtm, 0.05)
> d05 <- dist(t(dtmss05), method="euclidian")
> fit <- hclust(d=d05, method="complete")
> plot(fit, hang=1, main = "title")
> groups <- cutree(fit, k = 4)
> rect.hclust(fit, k = 4, border = "red")
> #
> dtmss10 <- removeSparseTerms(dtm, 0.10)
> d05 <- dist(t(dtmss10), method="euclidian")
> fit <- hclust(d=d10, method="complete")
> plot(fit, hang=1, main = "title")
> groups <- cutree(fit, k = 14)
> rect.hclust(fit, k = 14, border = "red")
> #
> dtmss15 <- removeSparseTerms(dtm, 0.15)
> d05 <- dist(t(dtmss15), method="euclidian")
> fit <- hclust(d=d15, method="complete")
> plot(fit, hang=1, main = "title")
> groups <- cutree(fit, k = 14)
> rect.hclust(fit, k = 14, border = "red")
> #
> d5 <- dist(t(dtmss05), method="euclidian")
> kfit <- kmeans(d5,4)
> clusplot(as.matrix(d5), kfit$cluster, color=T, shade=T, labels=2, lines=0)
> clusplot(as.matrix(d5), kfit$cluster, color=T, shade=T, labels=2, lines=0, main = "CLUSPLOT
- 5% Sparsity, k = 4 means")
> #
> d10 <- dist(t(dtmss10), method="euclidian")
> kfit <- kmeans(d10,4)
> clusplot(as.matrix(d10), kfit$cluster, color=T, shade=T, labels=2, lines=0, main =
"CLUSPLOT - 10% Sparsity, k = 4 means")
```

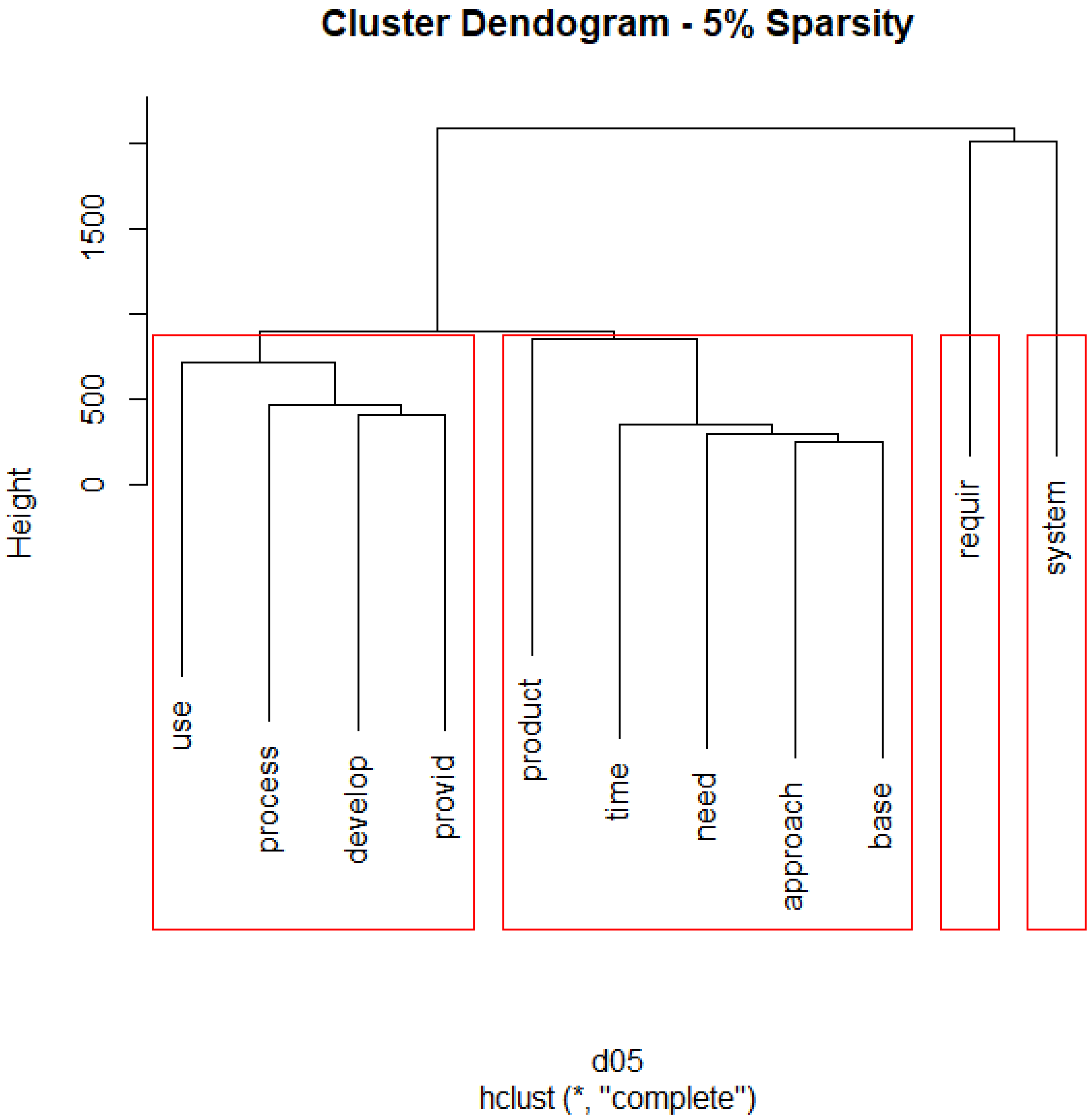
APPENDIX B

R Text Mining Code (continued)

```
> #
> kfit <- kmeans(d10,5)
> clusplot(as.matrix(d10), kfit$cluster, color=T, shade=T, labels=2, lines=0, main =
"CLUSPLOT - 10% Sparsity, k = 5 means")
> #
> kfit <- kmeans(d10,6)
> clusplot(as.matrix(d10), kfit$cluster, color=T, shade=T, labels=2, lines=0, main =
"CLUSPLOT - 10% Sparsity, k = 6 means")
> #
> kfit <- kmeans(d10,7)
> clusplot(as.matrix(d10), kfit$cluster, color=T, shade=T, labels=2, lines=0, main =
"CLUSPLOT - 10% Sparsity, k = 7 means")
> #
> kfit <- kmeans(d10,8)
> clusplot(as.matrix(d10), kfit$cluster, color=T, shade=T, labels=2, lines=0, main =
"CLUSPLOT - 10% Sparsity, k = 8 means")
> #
> kfit <- kmeans(d15,4)
> clusplot(as.matrix(d15), kfit$cluster, color=T, shade=T, labels=2, lines=0, main =
"CLUSPLOT - 15% Sparsity, k = 4 means")
> #
> kfit <- kmeans(d15,5)
> clusplot(as.matrix(d15), kfit$cluster, color=T, shade=T, labels=2, lines=0, main =
"CLUSPLOT - 15% Sparsity, k = 5 means")
> #
> kfit <- kmeans(d15,6)
> clusplot(as.matrix(d15), kfit$cluster, color=T, shade=T, labels=2, lines=0, main =
"CLUSPLOT - 15% Sparsity, k = 6 means")
> #
> kfit <- kmeans(d15,7)
> clusplot(as.matrix(d15), kfit$cluster, color=T, shade=T, labels=2, lines=0, main =
"CLUSPLOT - 15% Sparsity, k = 7 means")
> #
> kfit <- kmeans(d15,8)
> clusplot(as.matrix(d15), kfit$cluster, color=T, shade=T, labels=2, lines=0, main =
"CLUSPLOT - 15% Sparsity, k = 8 means")
> #
> findAssocs(dtm, c("word_list"), corlimit = 0.50)
```

APPENDIX C

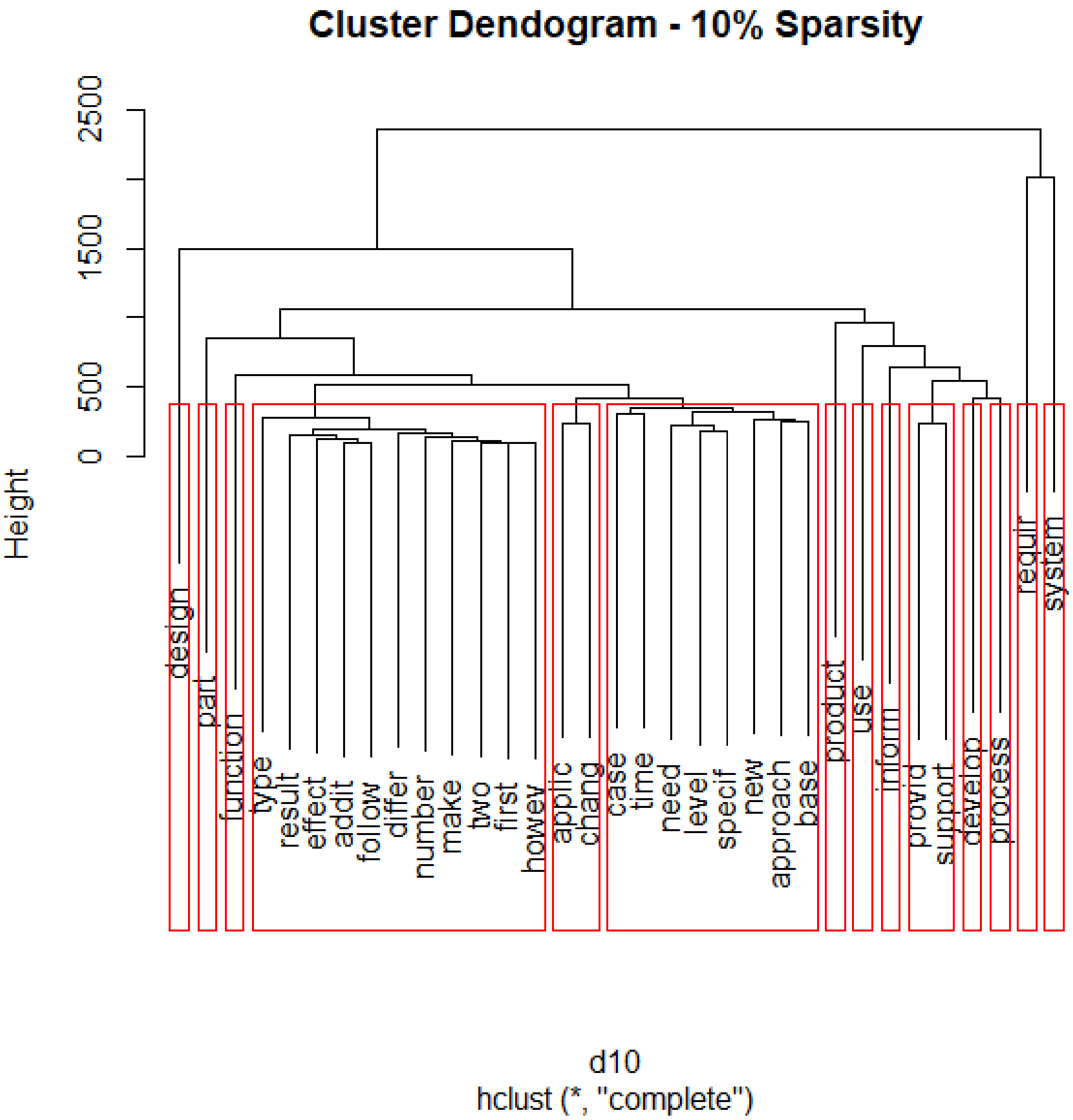
Complete Sequence 5%, 10%, and 15% Cluster Dendograms – Technology Design



APPENDIX C

Complete Sequence 5%, 10%, and 15% Cluster Dendograms – Technology Design

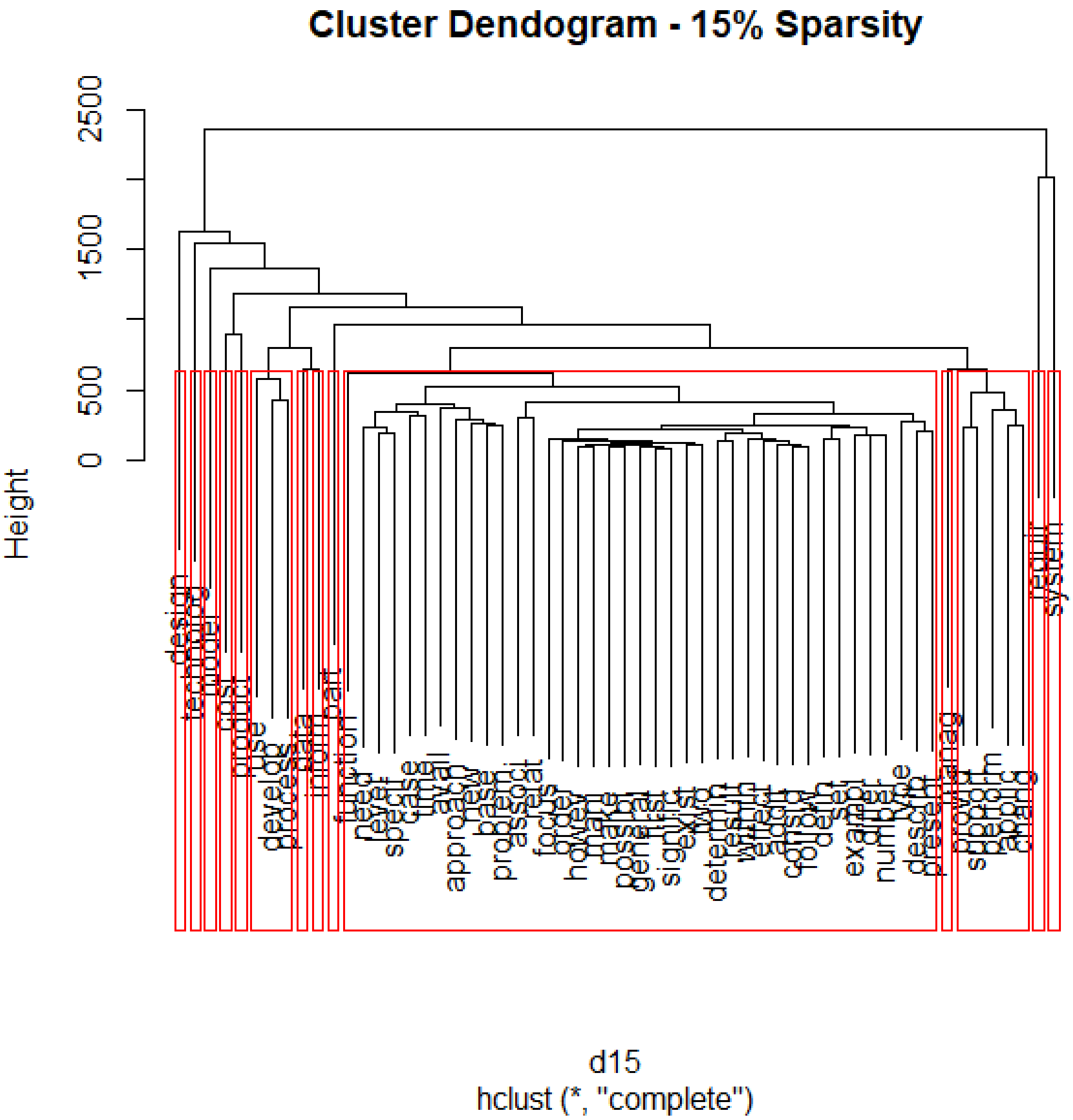
(continued)



APPENDIX C

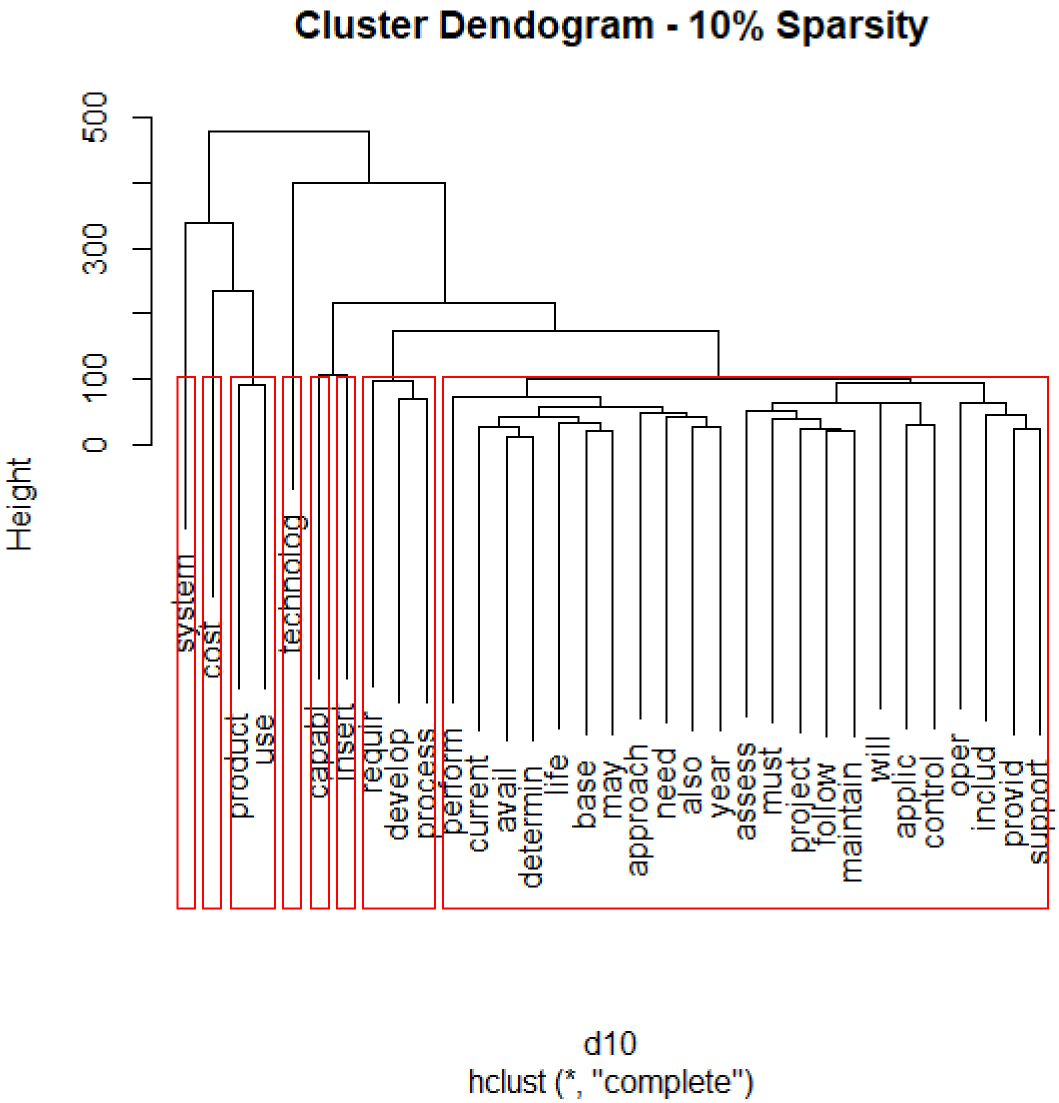
Complete Sequence 5%, 10%, and 15% Cluster Dendograms – Technology Design

(continued)



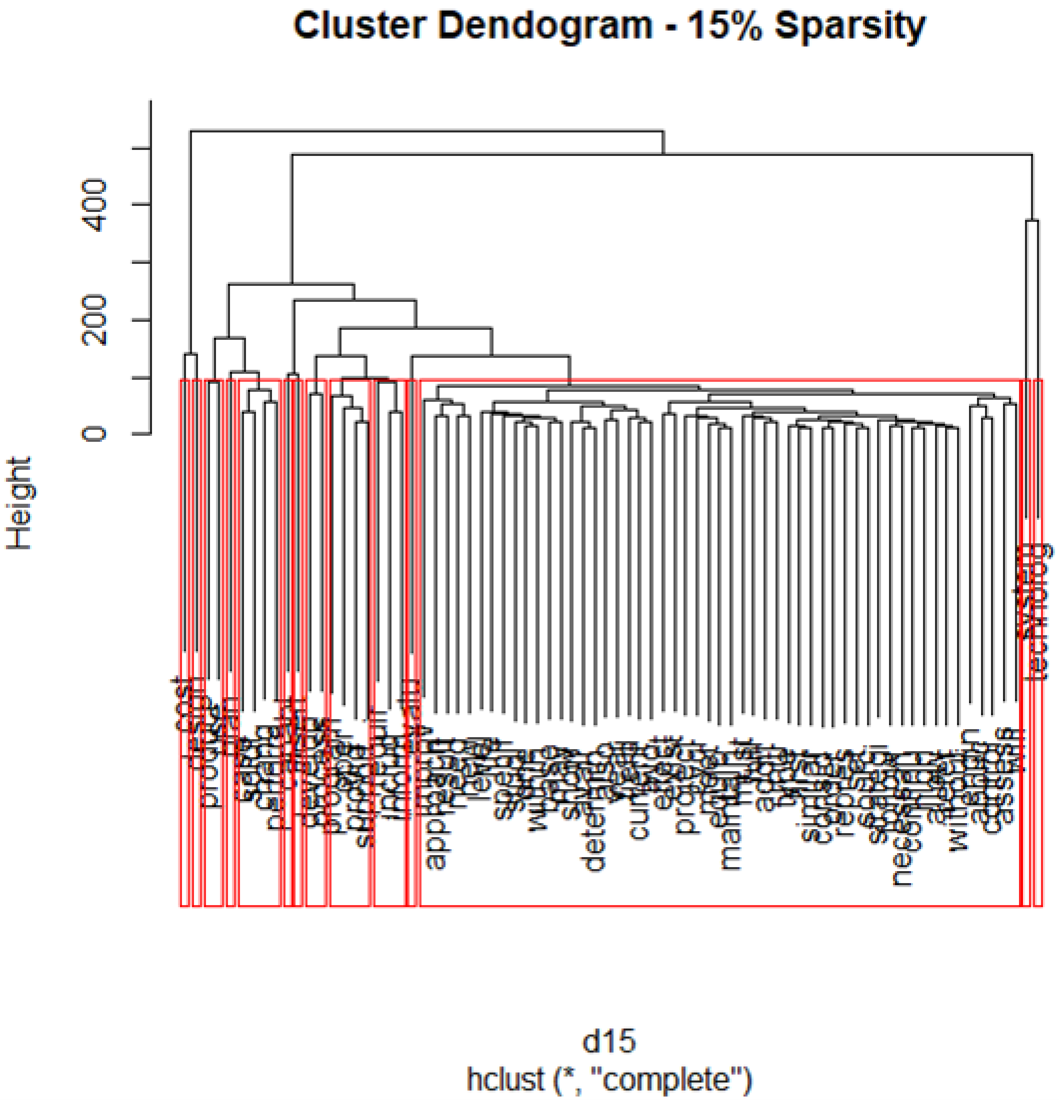
APPENDIX C

Complete Sequence 5%, 10%, and 15% Cluster Dendograms – Technology Insertion Design



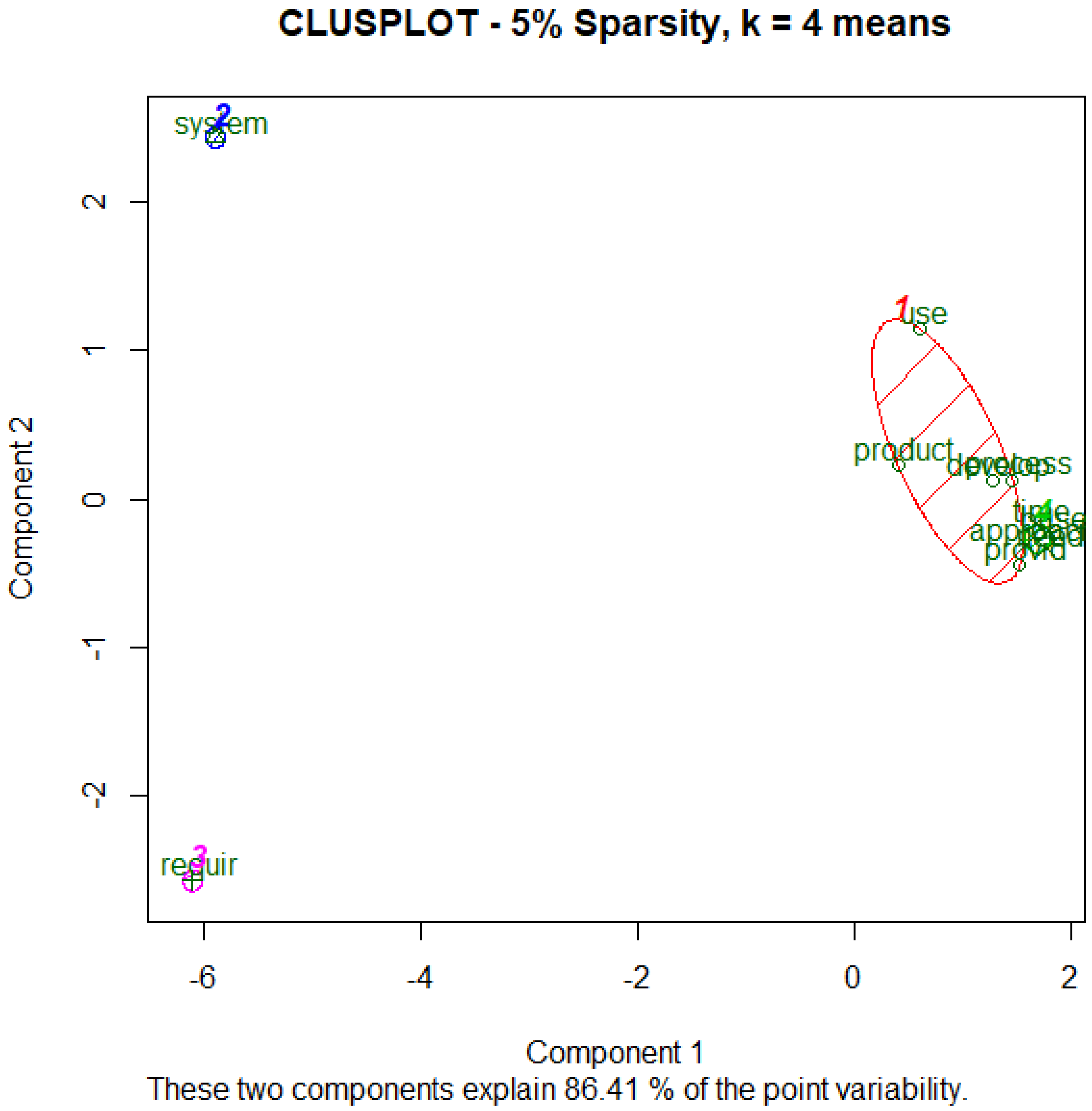
APPENDIX C

Complete Sequence 5%, 10%, and 15% Cluster Dendograms – Technology Insertion Design (continued)



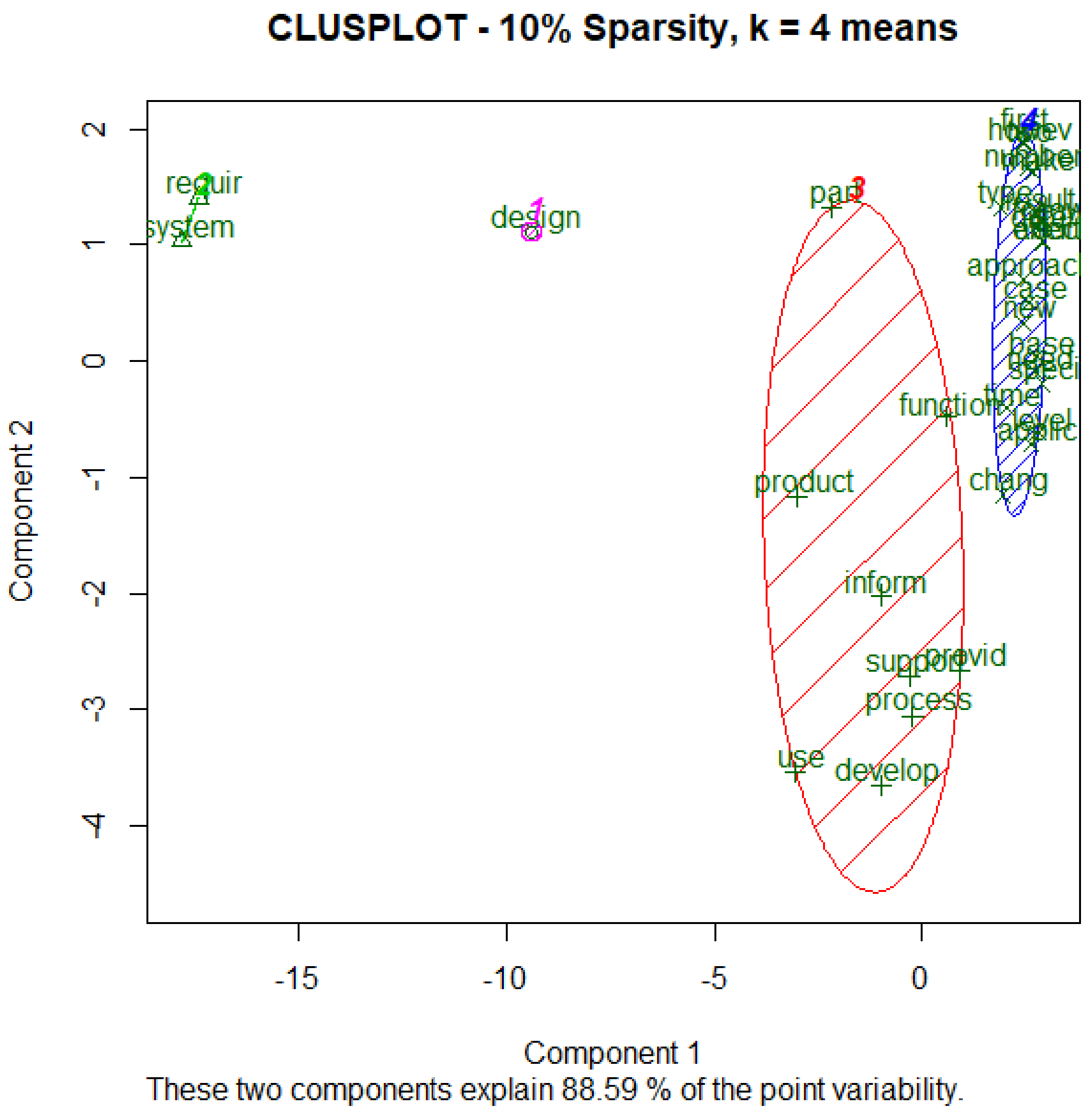
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design



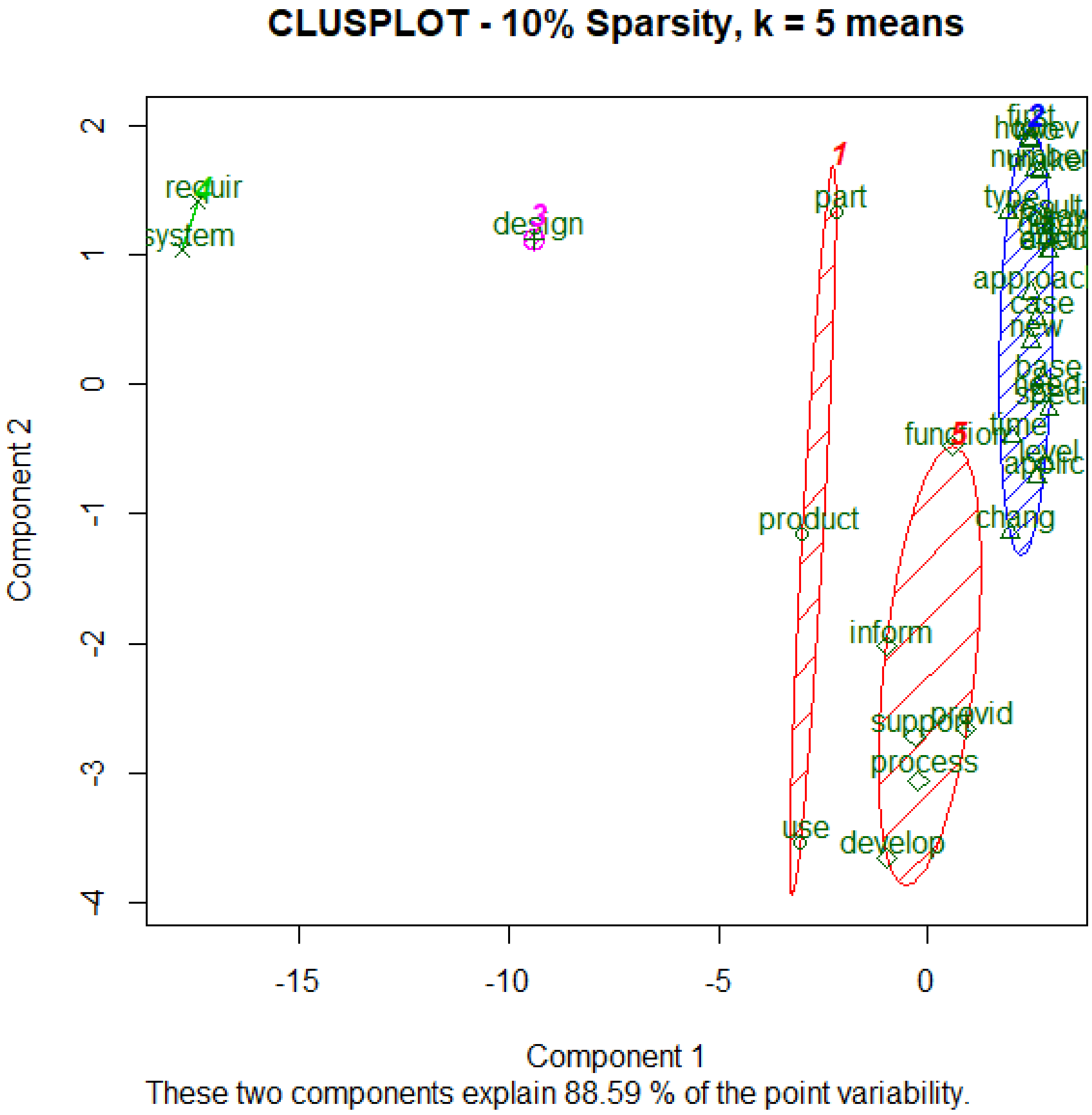
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design (continued)



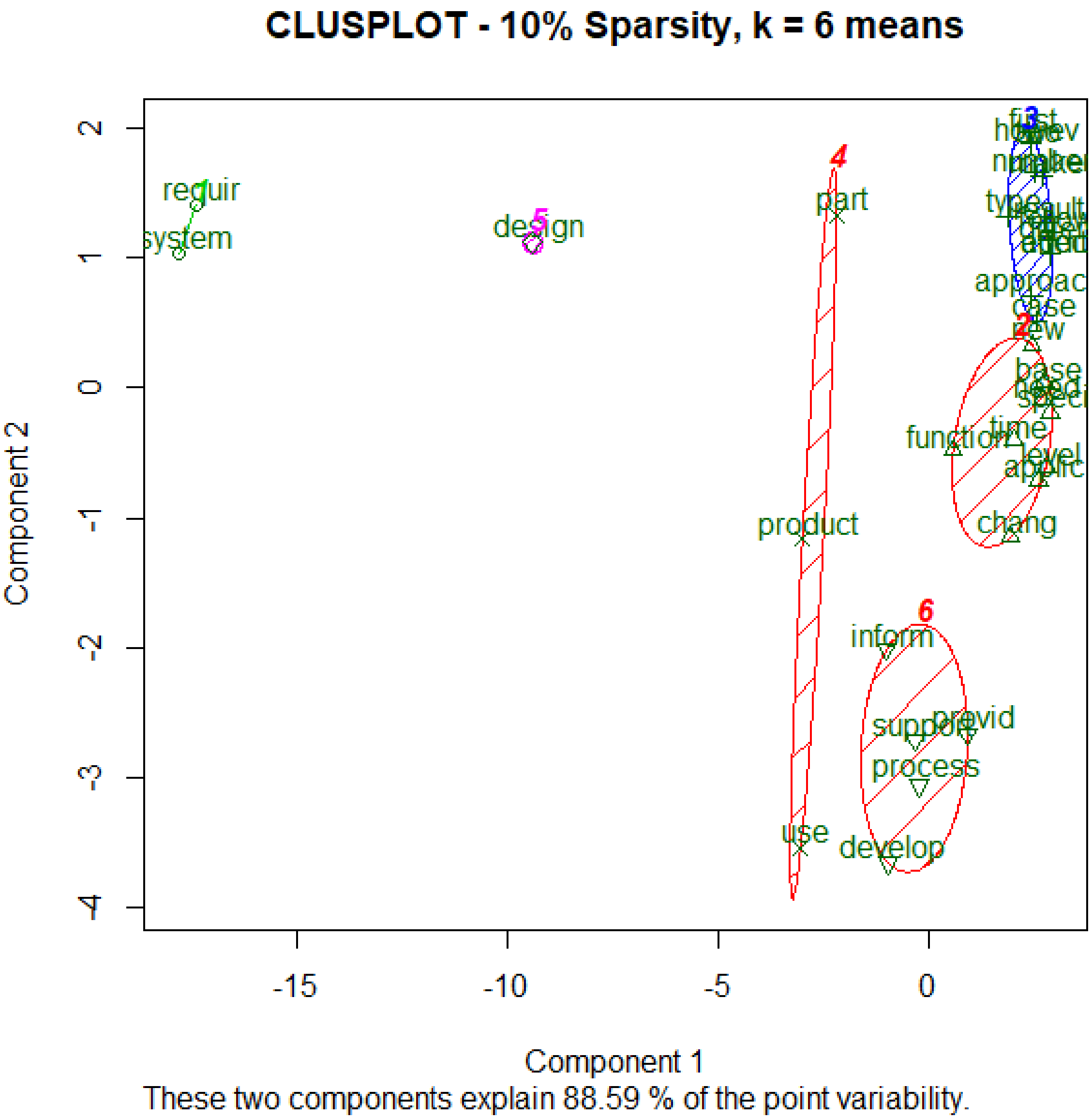
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design (continued)



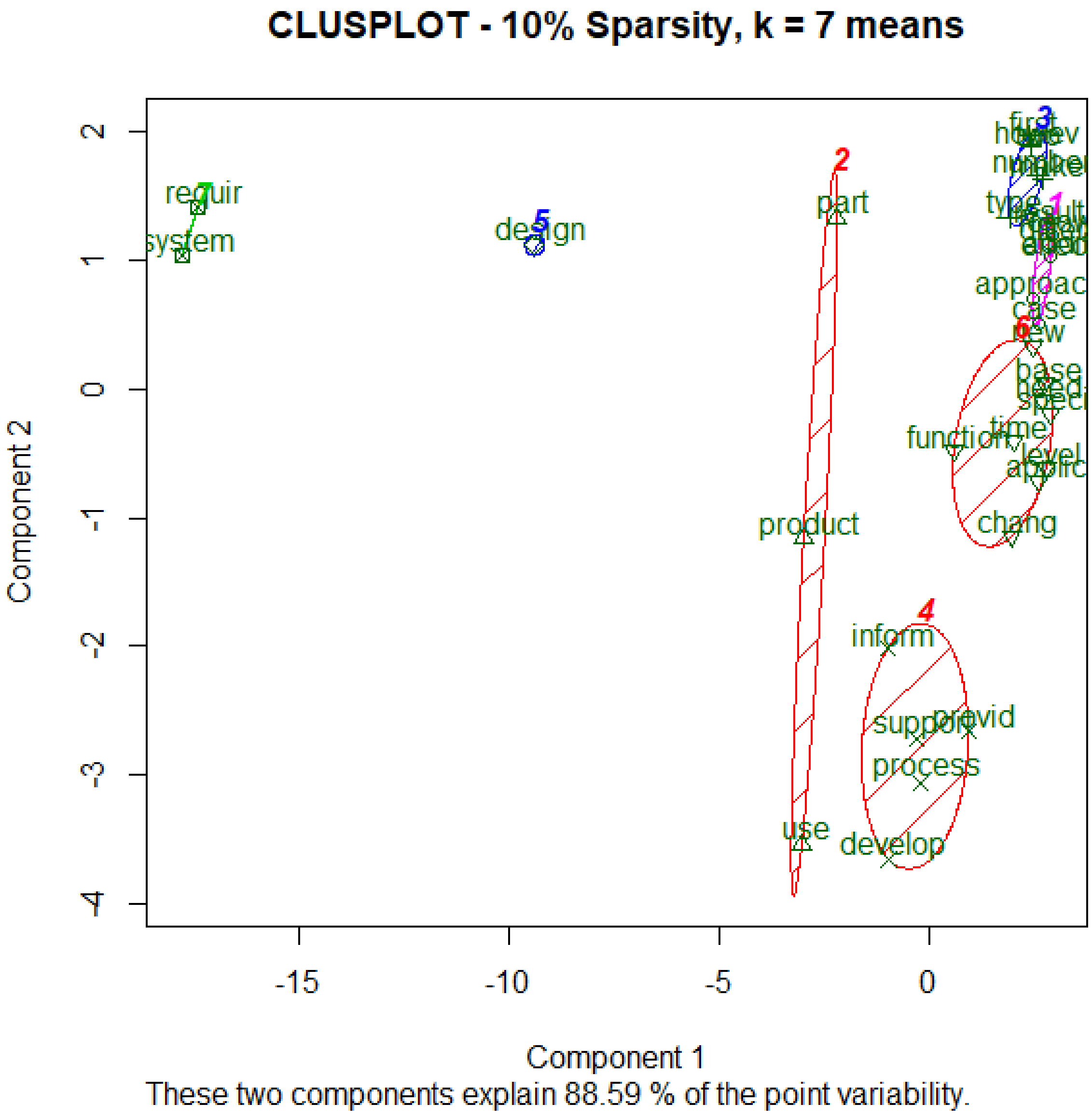
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design (continued)



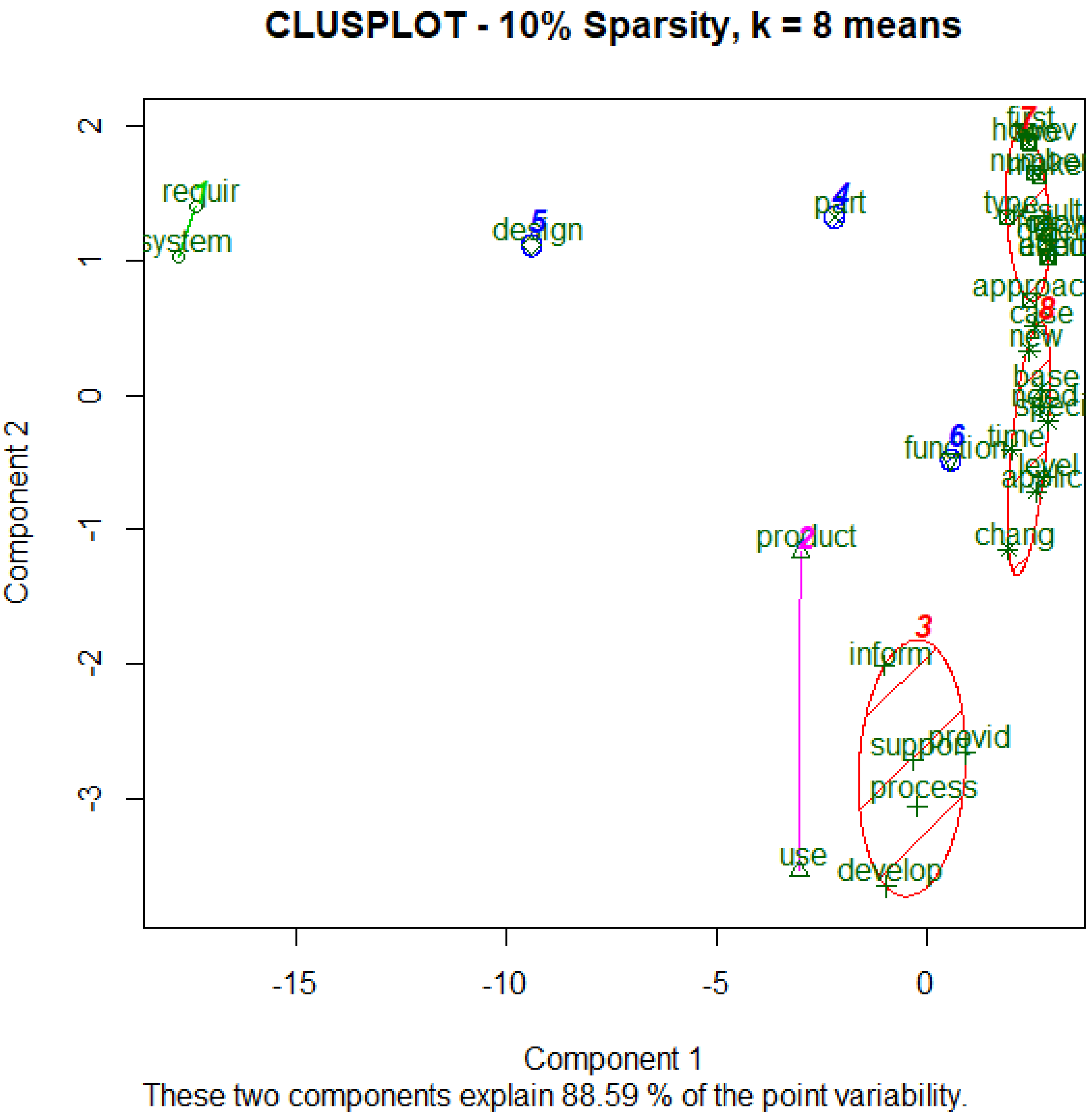
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design (continued)



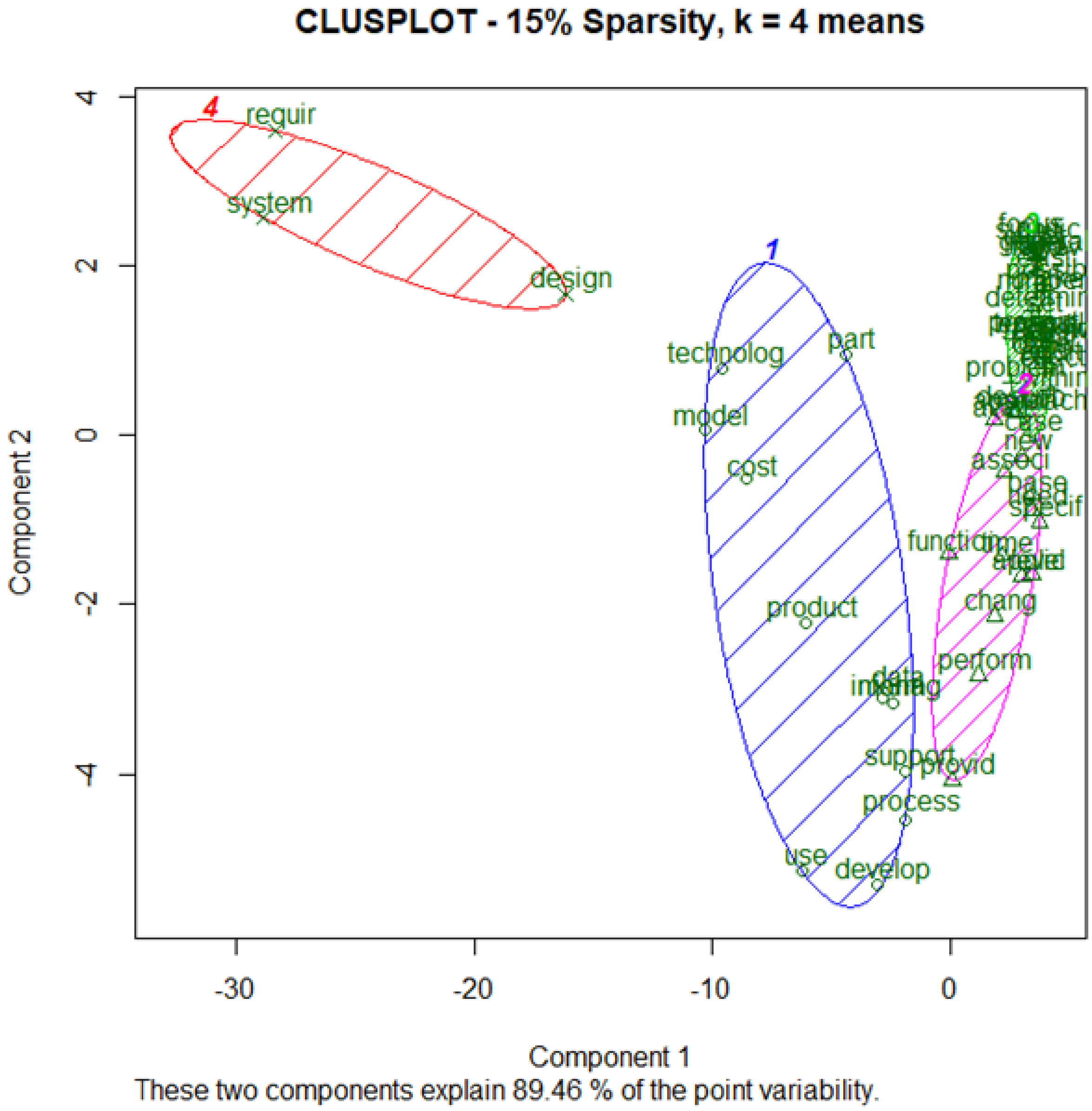
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design (continued)



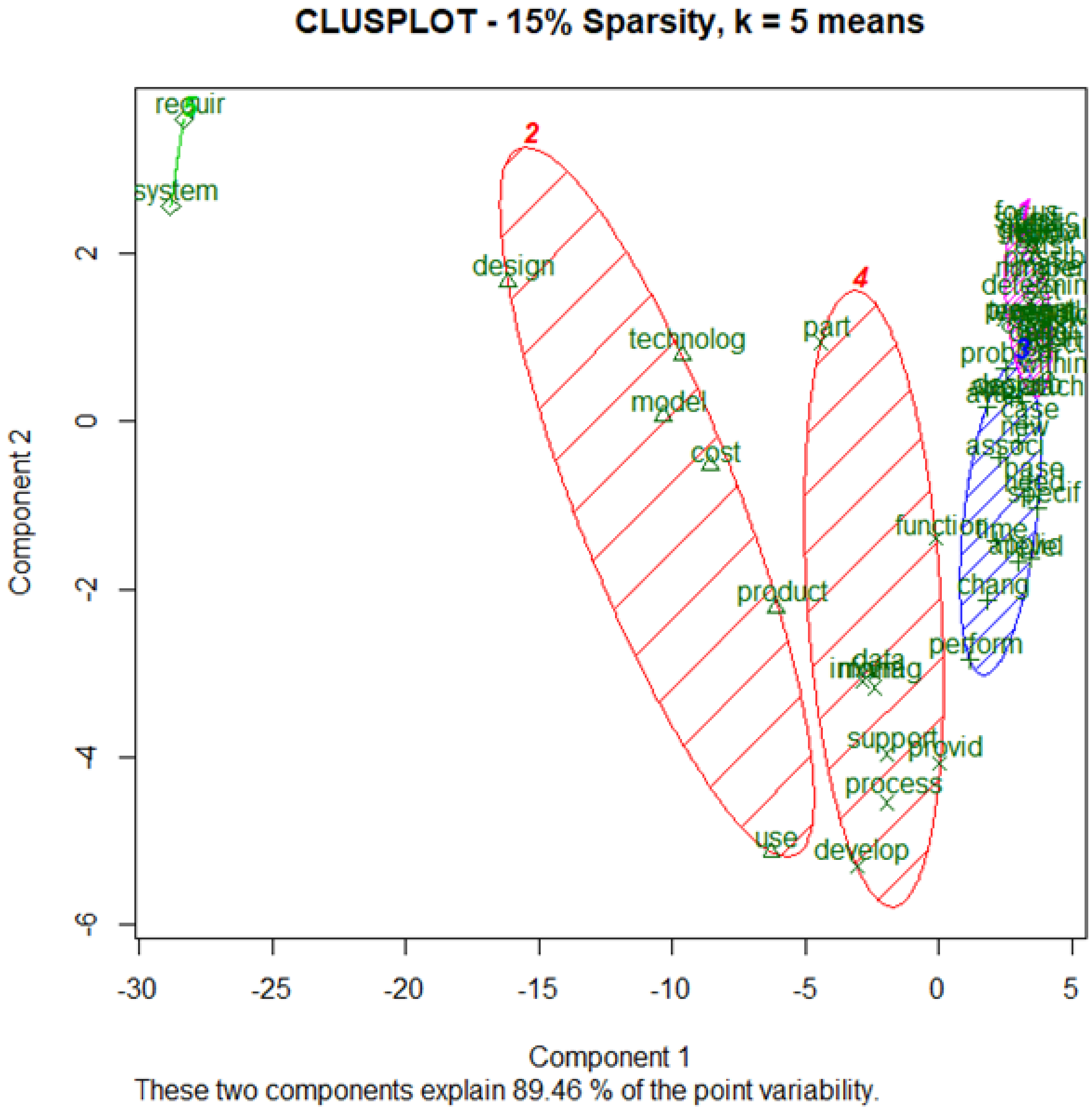
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design (continued)



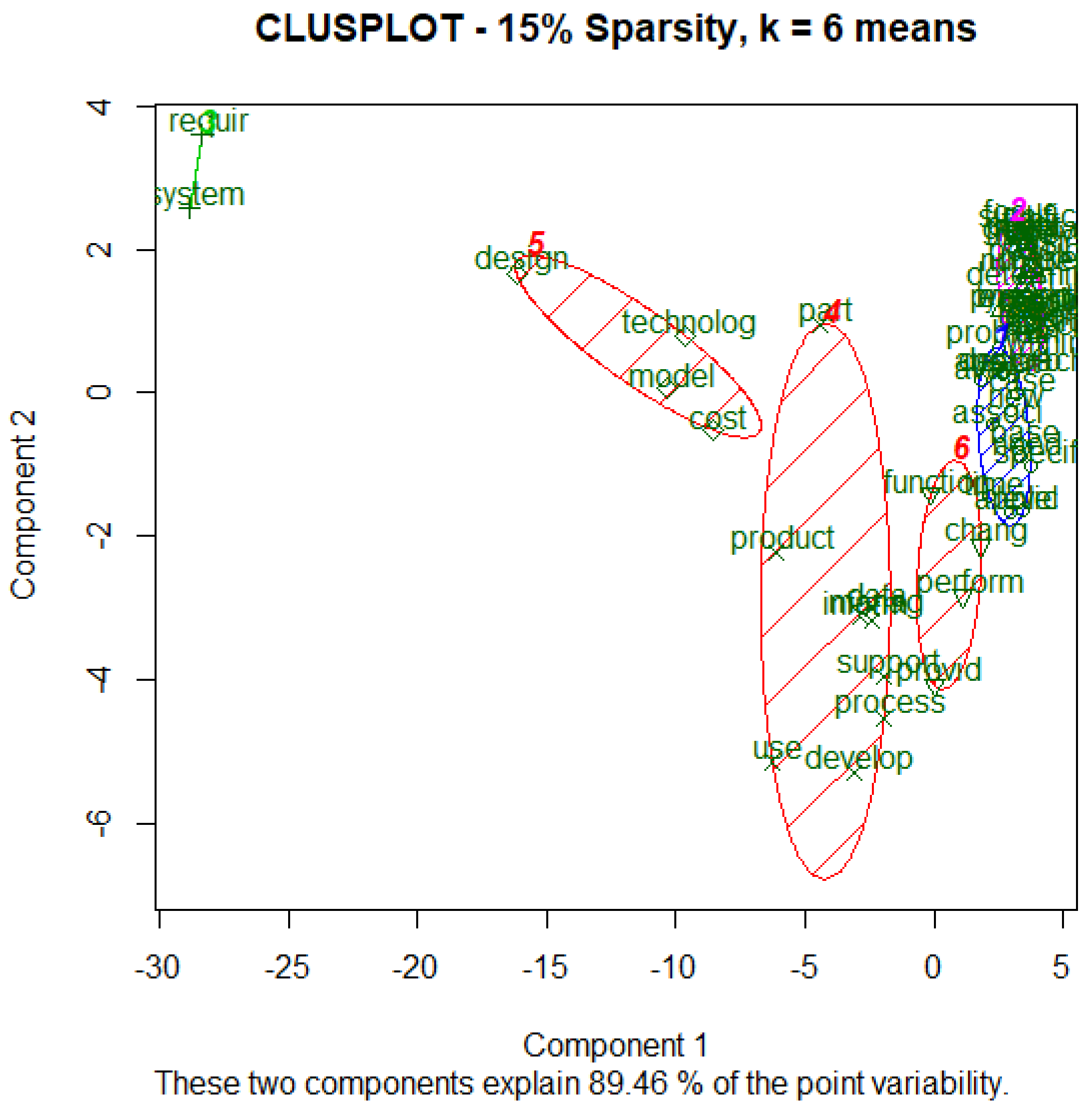
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design (continued)



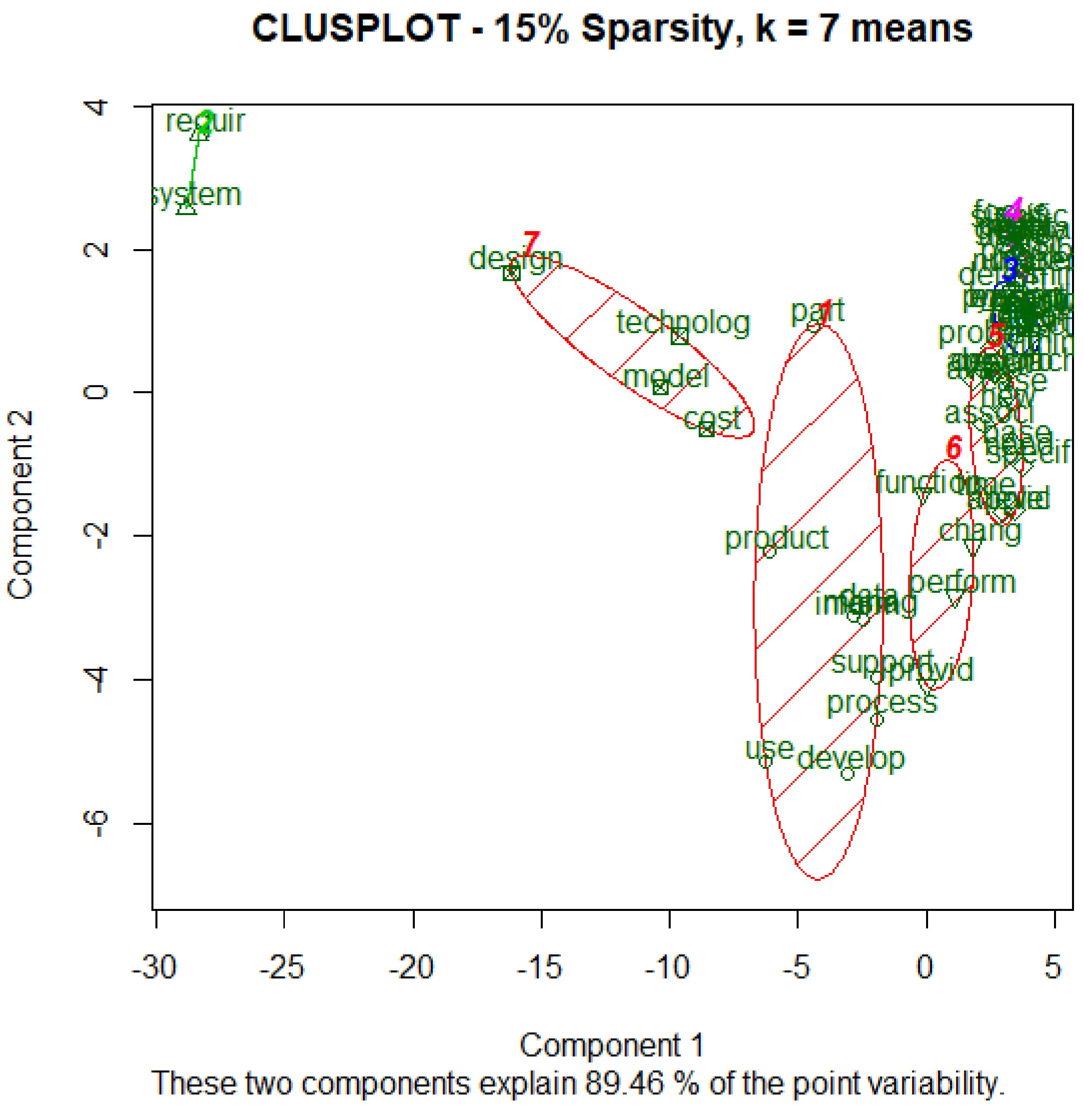
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design (continued)



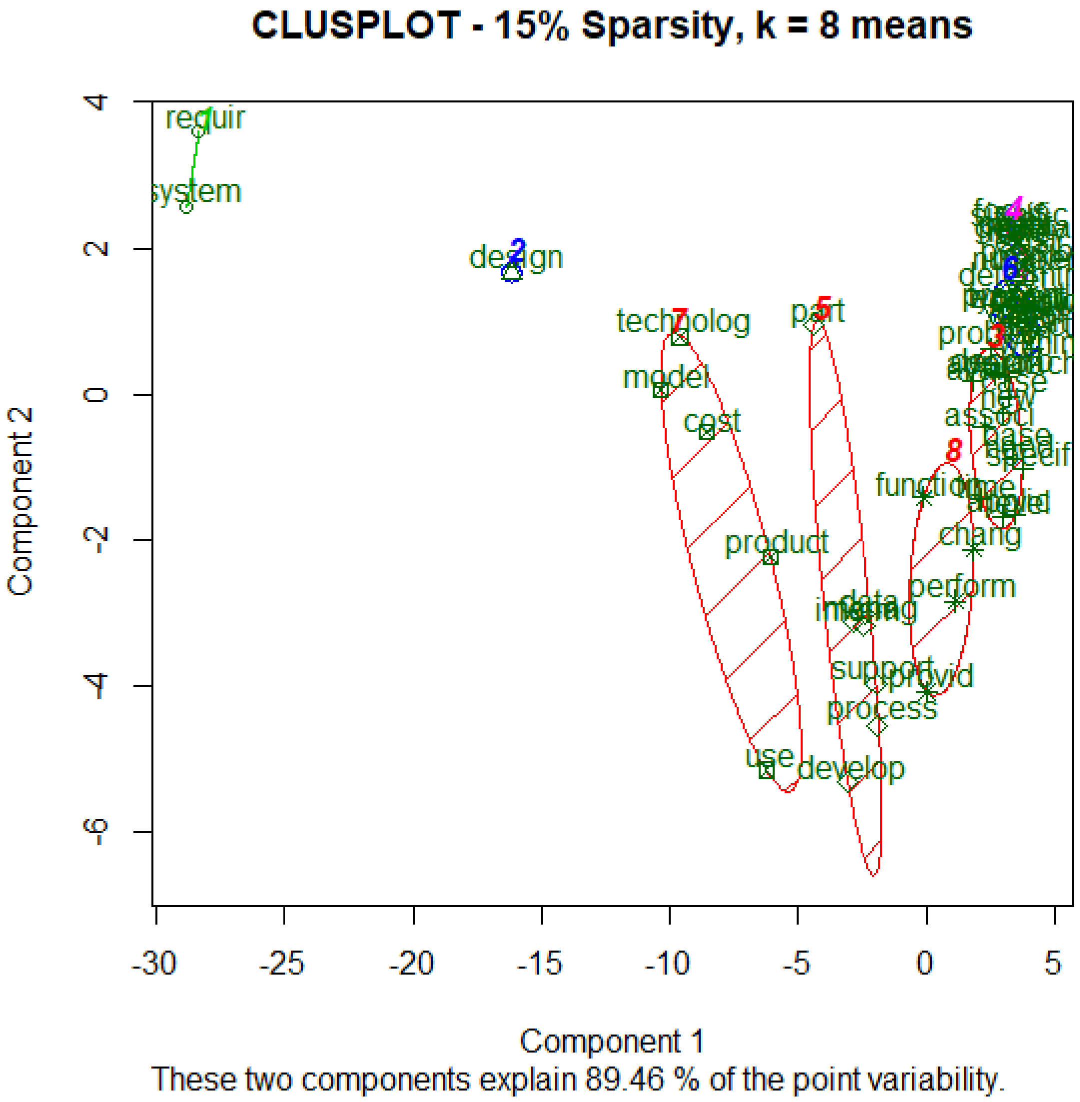
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design (continued)



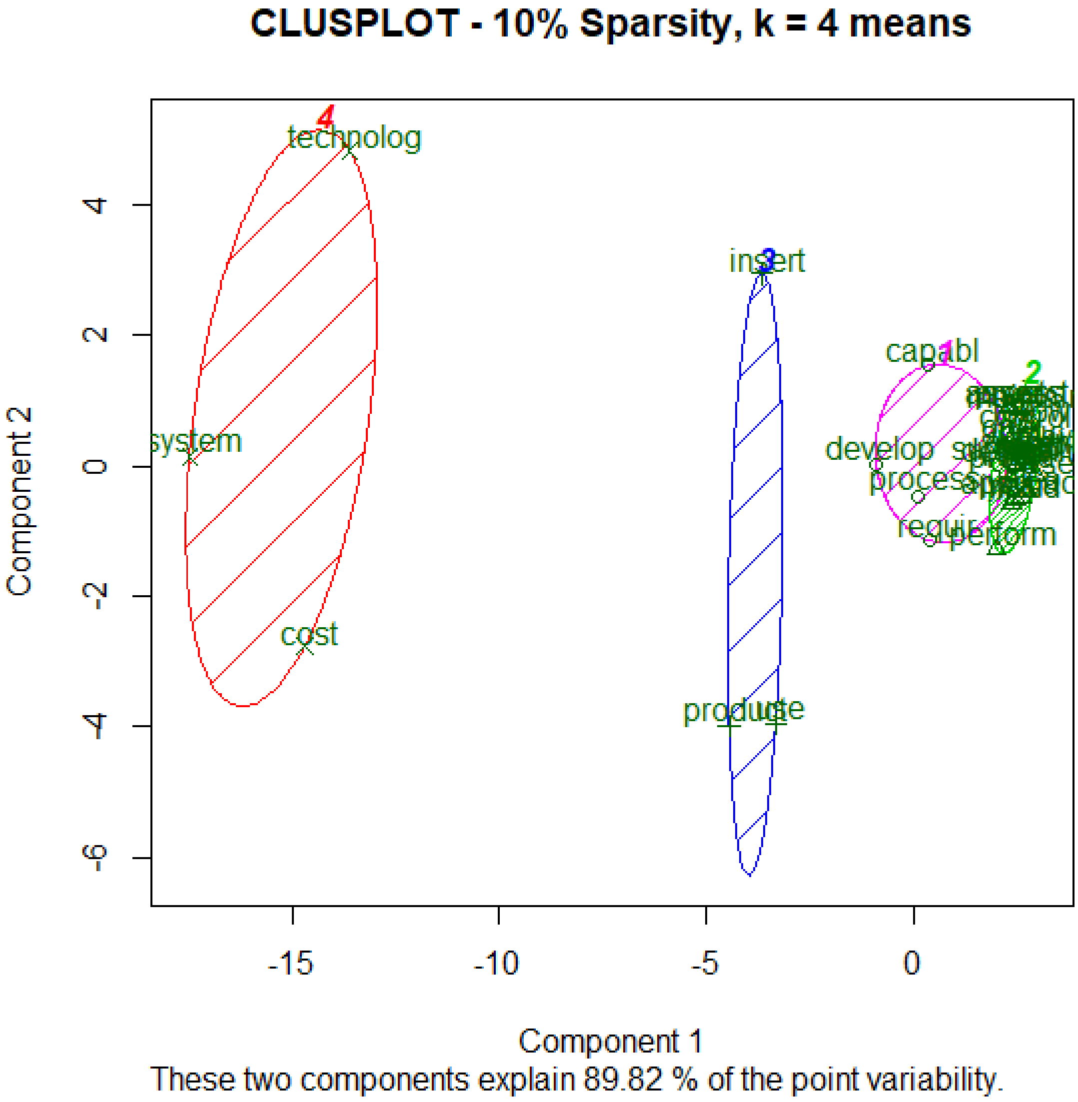
APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Design (continued)



APPENDIX D

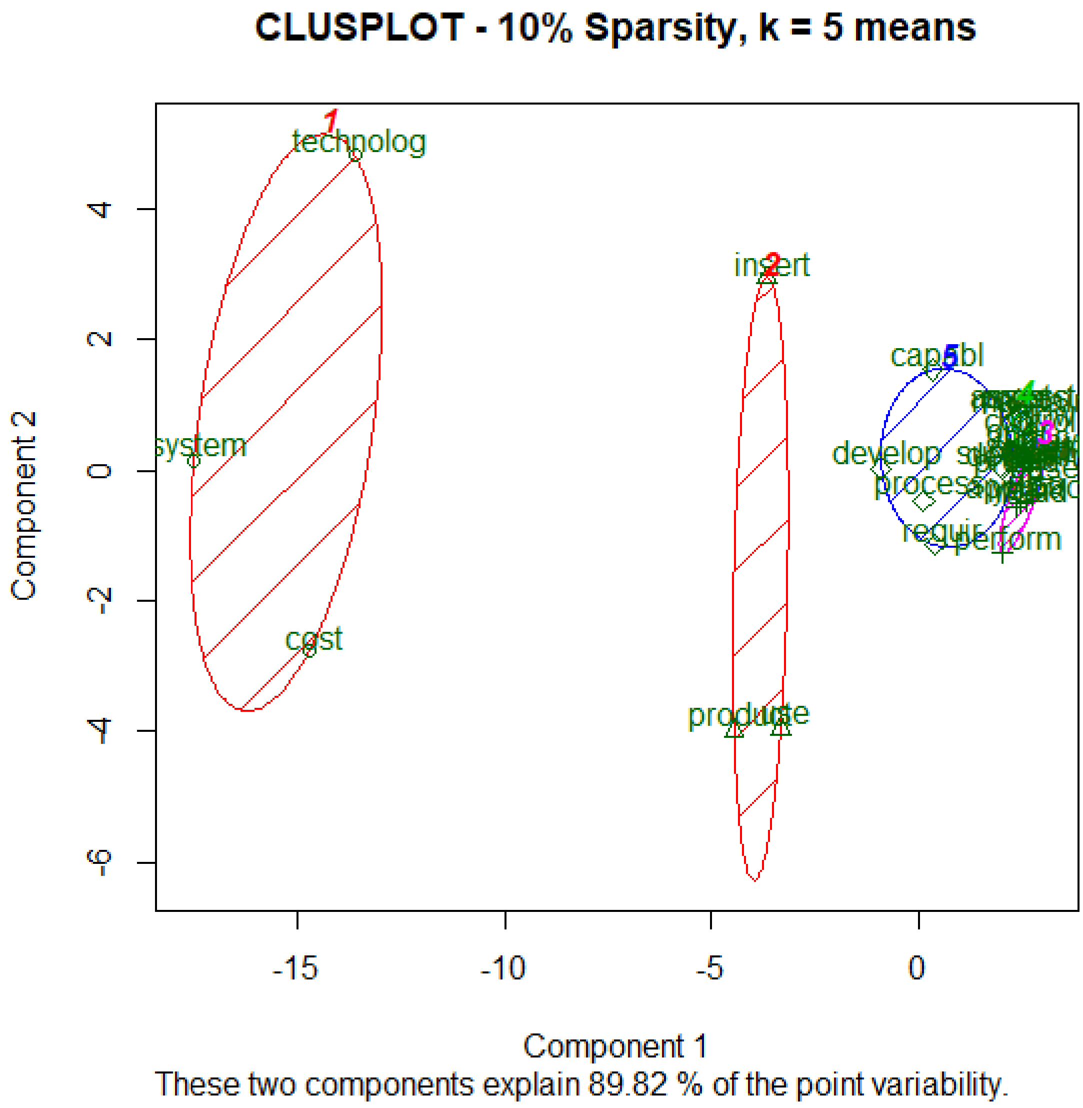
Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Insertion Design



APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Insertion Design

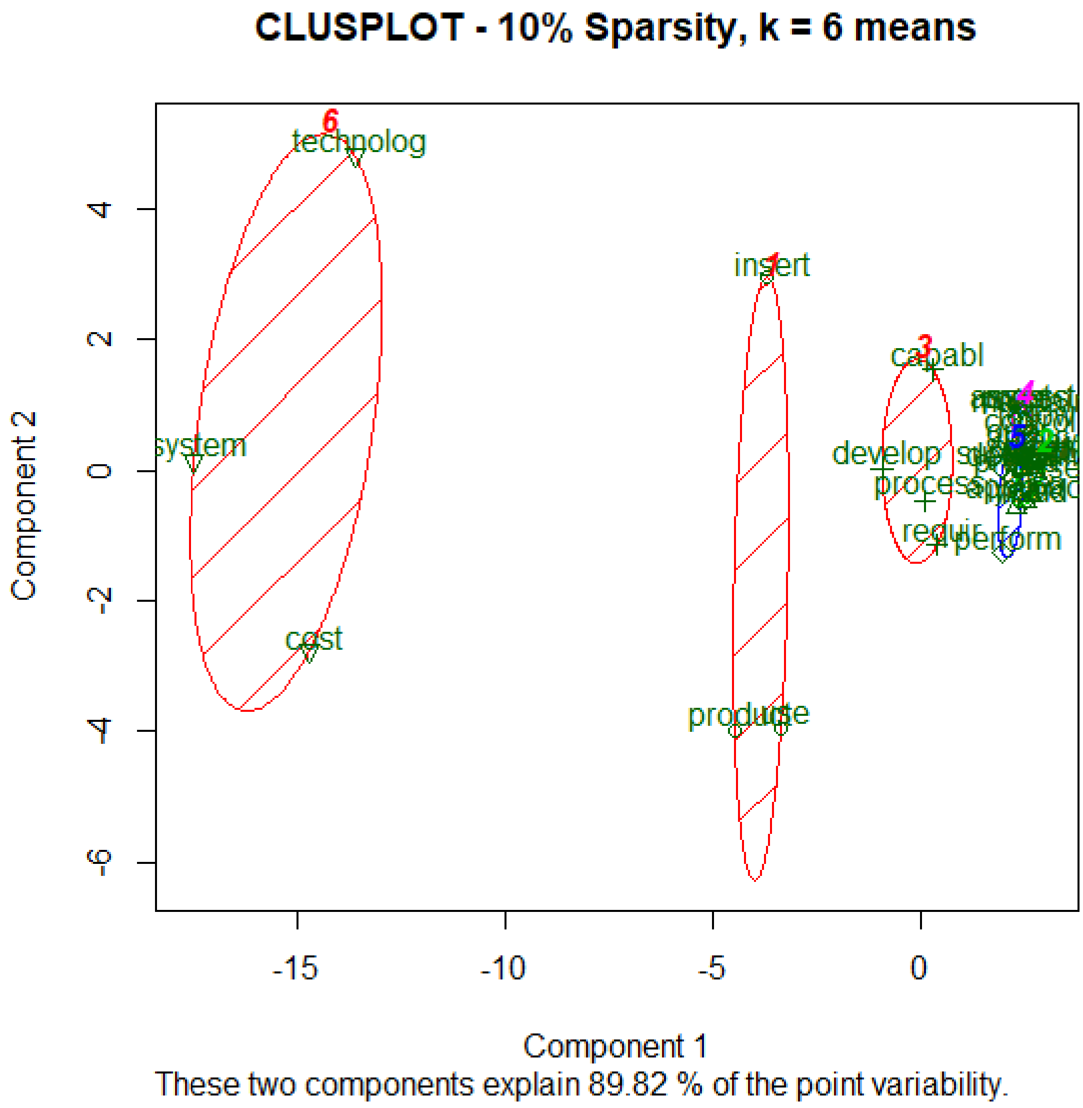
(continued)



APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Insertion Design

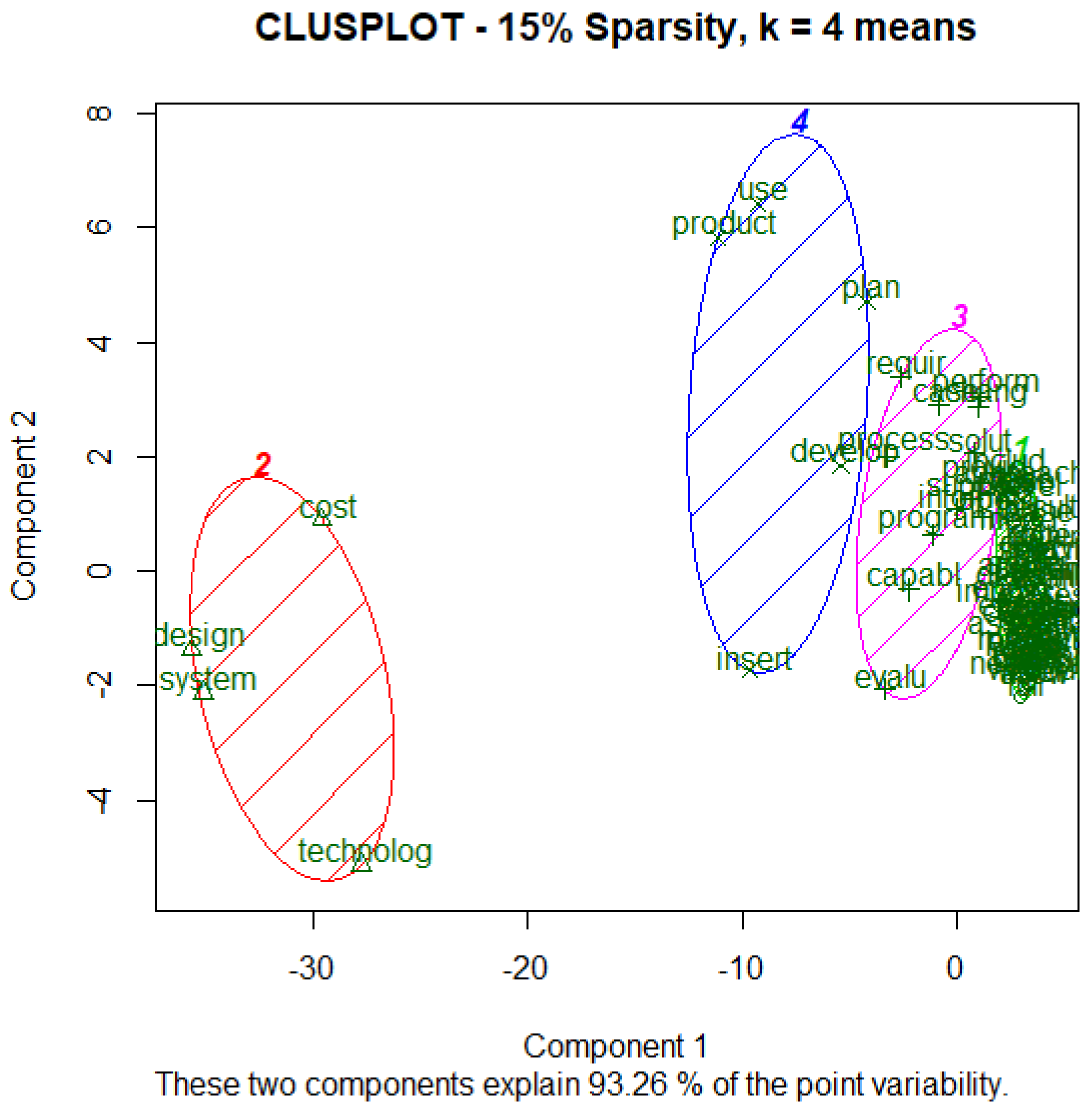
(continued)



APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Insertion Design

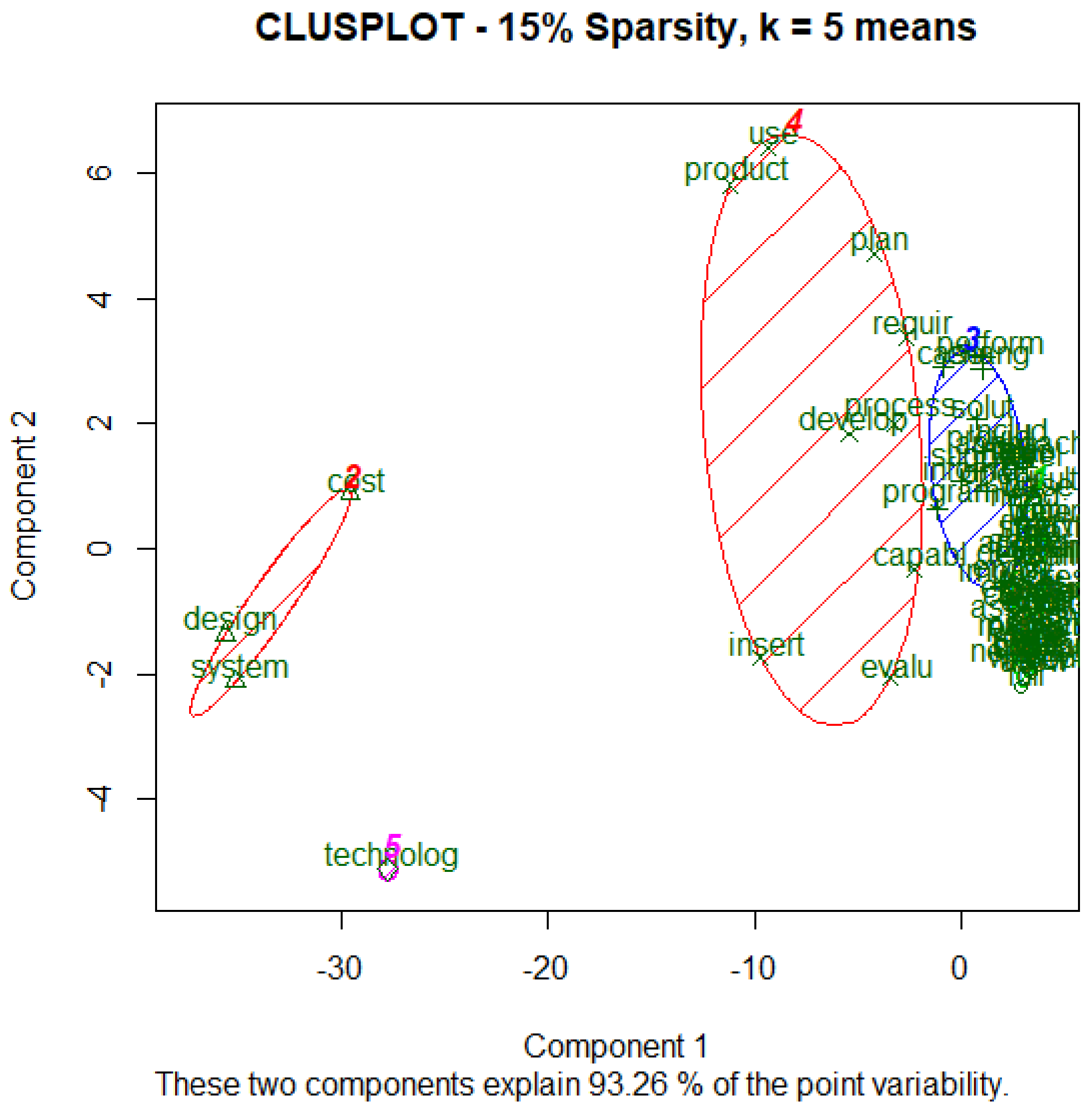
(continued)



APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Insertion Design

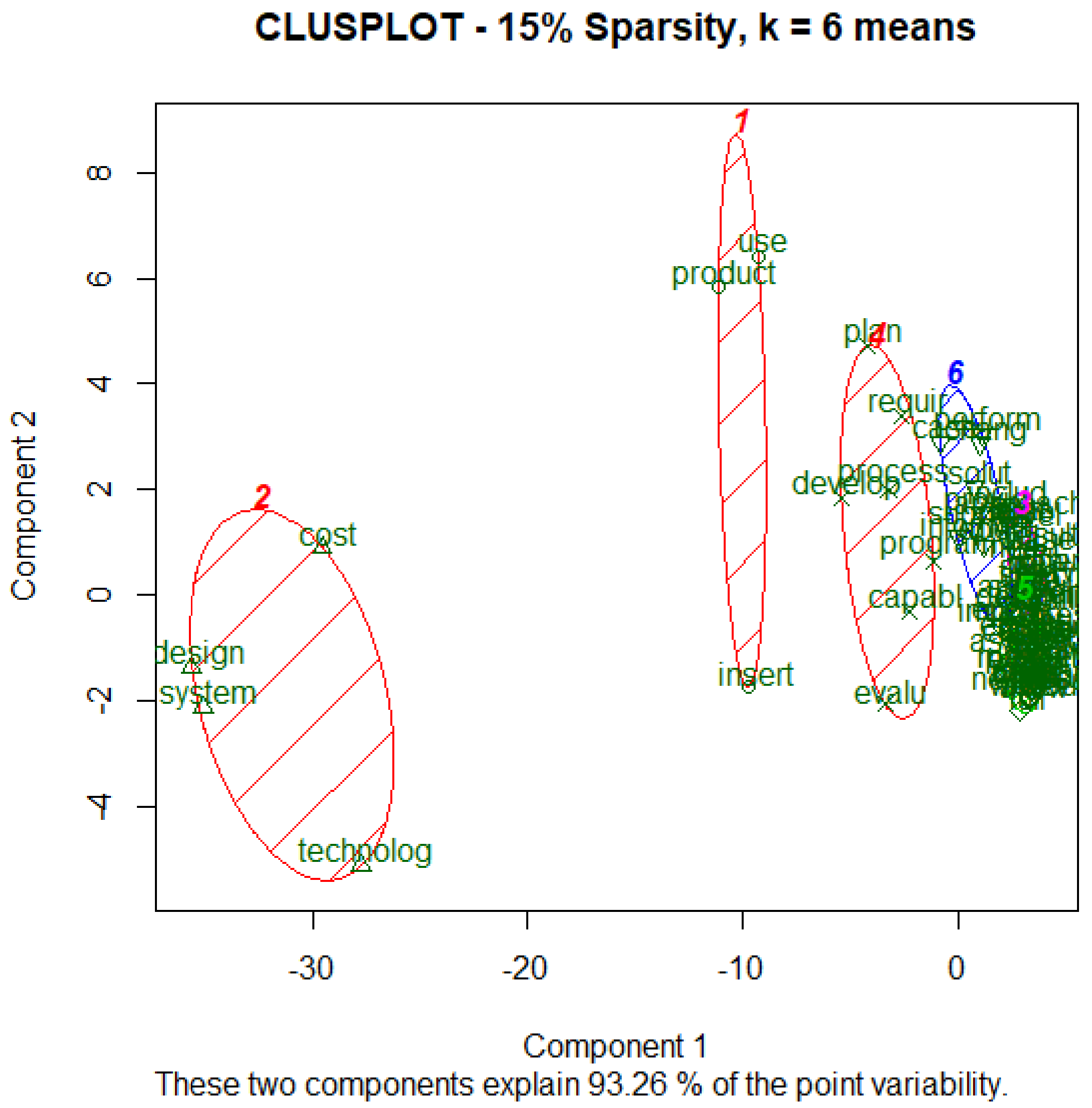
(continued)



APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Insertion Design

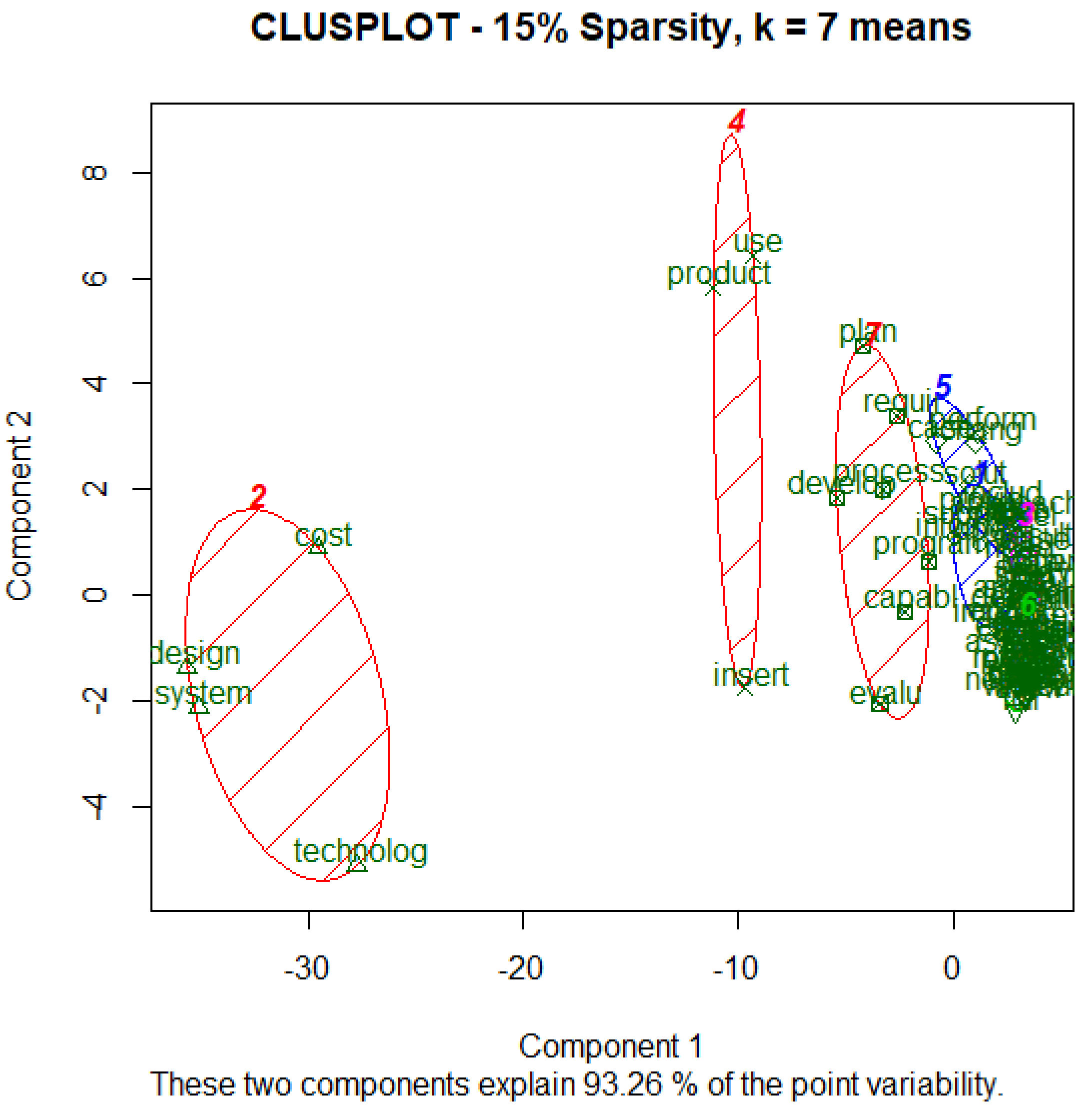
(continued)



APPENDIX D

Complete Sequence 5%, 10%, and 15% Cluster Plots – Technology Insertion Design

(continued)



APPENDIX E

technology design ONTOLOGY ENCODING

Title: 'technology design'.
Author: 'Kevin J Michael'.
Namespace: 'http://ontorion.com/namespace'.

Comment: 'Primitive concept definitions'.

Every design is a primitive-concept.
Every technology is a primitive-concept.
Every model is a primitive-concept.
Every cost is a primitive-concept.
Every system is a primitive-concept.
Every requirement is a primitive-concept.
Every part is a primitive-concept.
Every product is a primitive-concept.
Every use is a primitive-concept.
Every develop is a primitive-concept.
Every process is a primitive-concept.
Every information is a primitive-concept.

Comment: 'Primitive concepts existential attribute specifications'.

Every creation is a design.
Every devise is a design.
Every purpose is a design.
Every application is a technology.
Every engineering is a technology.
Every knowledge is a technology.
Every realization is a technology.
Every scientific is a technology.
Every architecture is a model.
Every expenditure is a cost.
Every governance is a system.
Every interaction is a system.
Every purpose is a system.
Every transformation is a system.
Every condition is a requirement.
Every necessary is a requirement.
Every type is a part.
Every unit is a part.
Every assemblage is a product.
Every function is a product.

APPENDIX E

technology design ONTOLOGY ENCODING (continued)

Every performance is a product.
Every accomplish is a use.
Every purpose is a use.
Every evolution is a develop.
Every innovation is a develop.
Every action is a process.
Every course is a process.
Every intention is a process.
Every fact is a information.
Every understanding is a information.

Comment: 'Primitive concepts state modification attribute specifications'.

Every design has-action equal-to 'creation'.
Every design has-action equal-to 'devise'.
Every design has-action that-matches-pattern 'purpose'.
Every design has-action different-from 'system'.
Every design has-action different-from 'requirement'.
Every design has-function equal-to 'creation'.
Every design has-function equal-to 'devise'.
Every design has-function that-matches-pattern 'purpose'.
Every design has-function different-from 'system'.
Every design has-function different-from 'requirement'.
Every design has-pattern equal-to 'creation'.
Every design has-pattern equal-to 'devise'.
Every design has-pattern that-matches-pattern 'purpose'.
Every design has-pattern different-from 'system'.
Every design has-pattern different-from 'requirement'.
Every design has-representation equal-to 'devise'.
Every design has-representation equal-to 'creation'.
Every design has-representation that-matches-pattern 'purpose'.
Every design has-representation equal-to 'model'.
Every design has-representation different-from 'system'.
Every design has-representation different-from 'requirement'.
Every technology has-capacity equal-to 'application'.
Every technology has-capacity equal-to 'knowledge'.
Every technology has-capacity equal-to 'realization'.
Every technology has-capacity equal-to 'function'.
Every technology has-capacity different-from 'requirement'.
Every technology has-performance equal-to 'application'.
Every technology has-performance equal-to 'knowledge'.

APPENDIX E

technology design ONTOLOGY ENCODING (continued)

Every technology has-performance equal-to 'realization'.
Every technology has-performance equal-to 'function'.
Every technology has-performance different-from 'requirement'.
Every technology has-robustness equal-to 'application'.
Every technology has-robustness equal-to 'knowledge'.
Every technology has-robustness equal-to 'realization'.
Every technology has-robustness equal-to 'function'.
Every technology has-robustness different-from 'requirement'.
Every technology has-stability equal-to 'application'.
Every technology has-stability equal-to 'knowledge'.
Every technology has-stability equal-to 'realization'.
Every technology has-stability equal-to 'function'.
Every technology has-stability different-from 'requirement'.
Every model has-accuracy equal-to 'architecture'.
Every model has-accuracy equal-to 'design'.
Every model has-accuracy different-from 'requirement'.
Every model has-effectiveness equal-to 'architecture'.
Every model has-effectiveness equal-to 'design'.
Every model has-effectiveness different-from 'requirement'.
Every model has-efficiency equal-to 'architecture'.
Every model has-efficiency equal-to 'design'.
Every model has-efficiency different-from 'requirement'.
Every model has-robustness equal-to 'architecture'.
Every model has-robustness equal-to 'design'.
Every model has-robustness different-from 'requirement'.
Every cost has-amount equal-to 'expenditure'.
Every cost has-amount different-from 'requirement'.
Every cost has-denomination equal-to 'expenditure'.
Every cost has-denomination different-from 'requirement'.
Every cost has-time equal-to 'expenditure'.
Every cost has-time different-from 'requirement'.
Every system has-boundary greater-or-equal-to 'governance'.
Every system has-boundary different-from 'interactions'.
Every system has-boundary that-matches-pattern 'purpose'.
Every system has-boundary greater-or-equal-to 'transformation'.
Every system has-coordination lower-or-equal-to 'governance'.
Every system has-coordination equal-to 'interactions'.
Every system has-coordination that-matches-pattern 'purpose'.
Every system has-coordination equal-to 'transformation'.
Every system has-complexity different-from 'governance'.
Every system has-complexity equal-to 'interactions'.

APPENDIX E

technology design ONTOLOGY ENCODING (continued)

Every system has-complexity different-from 'purpose'.
Every system has-complexity different-from 'transformation'.
Every system has-coupling different-from 'governance'.
Every system has-coupling equal-to 'interactions'.
Every system has-coupling different-from 'purpose'.
Every system has-coupling equal-to 'transformation'.
Every system has-dynamic different-from 'governance'.
Every system has-dynamic equal-to 'interactions'.
Every system has-dynamic different-from 'purpose'.
Every system has-dynamic equal-to 'transformation'.
Every system has-environment greater-than 'governance'.
Every system has-environment greater-than 'interactions'.
Every system has-environment greater-than 'purpose'.
Every system has-environment greater-than 'transformation'.
Every system has-homeostasis different-from 'governance'.
Every system has-homeostasis equal-to 'interactions'.
Every system has-homeostasis different-from 'purpose'.
Every system has-homeostasis equal-to 'transformation'.
Every system has-inputs different-from 'governance'.
Every system has-inputs different-from 'interactions'.
Every system has-inputs equal-to 'purpose'.
Every system has-inputs equal-to 'transformation'.
Every system has-interdependency different-from 'governance'.
Every system has-interdependency different-from 'interactions'.
Every system has-interdependency equal-to 'purpose'.
Every system has-interdependency different-from 'transformation'.
Every system has-niche different-from 'governance'.
Every system has-niche different-from 'interactions'.
Every system has-niche equal-to 'purpose'.
Every system has-niche different-from 'transformation'.
Every system has-outputs different-from 'governance'.
Every system has-outputs different-from 'interactions'.
Every system has-outputs different-from 'purpose'.
Every system has-outputs equal-to 'transformation'.
Every system has-pluralism equal-to 'governance'.
Every system has-pluralism different-from 'interactions'.
Every system has-pluralism equal-to 'purpose'.
Every system has-pluralism different-from 'transformation'.
Every system has-policy equal-to 'governance'.
Every system has-policy different-from 'interactions'.
Every system has-policy equal-to 'purpose'.

APPENDIX E

technology design ONTOLOGY ENCODING (continued)

Every system has-policy different-from 'transformation'.
Every system has-wholeness different-from 'governance'.
Every system has-wholeness different-from 'interactions'.
Every system has-wholeness equal-to 'purpose'.
Every system has-wholeness different-from 'transformation'.
Every requirement has-attribute equal-to 'condition'.
Every requirement has-attribute equal-to 'necessary'.
Every requirement has-constraint equal-to 'condition'.
Every requirement has-constraint equal-to 'necessary'.
Every requirement has-function equal-to 'condition'.
Every requirement has-function equal-to 'necessary'.
Every requirement has-level equal-to 'condition'.
Every requirement has-level equal-to 'necessary'.
Every requirement has-value equal-to 'condition'.
Every requirement has-value equal-to 'necessary'.
Every part has-composition equal-to 'type'.
Every part has-composition equal-to 'unit'.
Every part has-form equal-to 'type'.
Every part has-form equal-to 'unit'.
Every part has-substance equal-to 'type'.
Every part has-substance equal-to 'unit'.
Every product has-entities equal-to 'assemblage'.
Every product has-entities equal-to 'function'.
Every product has-entities equal-to 'performance'.
Every product has-interactions equal-to 'assemblage'.
Every product has-interactions equal-to 'function'.
Every product has-interactions equal-to 'performance'.
Every use has-method equal-to 'accomplish'.
Every use has-method equal-to 'purpose'.
Every use has-objectives equal-to 'accomplish'.
Every use has-objectives equal-to 'purpose'.
Every develop has-change equal-to 'evolution'.
Every develop has-change different-from 'innovation'.
Every develop has-create different-from 'evolution'.
Every develop has-create equal-to 'innovation'.
Every develop has-new equal-to 'evolution'.
Every develop has-new equal-to 'innovation'.
Every develop has-purpose different-from 'evolution'.
Every develop has-purpose equal-to 'innovation'.
Every process has-activity equal-to 'actions'.
Every process has-activity equal-to 'course'.

APPENDIX E

technology design ONTOLOGY ENCODING (continued)

Every process has-activity different-from 'intention'.
Every process has-event equal-to 'actions'.
Every process has-event equal-to 'course'.
Every process has-event different-from 'intention'.
Every process has-mode equal-to 'actions'.
Every process has-mode equal-to 'course'.
Every process has-mode different-from 'intention'.
Every process has-path equal-to 'actions'.
Every process has-mode equal-to 'course'.
Every process has-mode different-from 'intention'.
Every process has-purpose different-from 'actions'.
Every process has-purpose different-from 'course'.
Every process has-purpose equal-to 'intention'.
Every information has-assertion different-from 'facts'.
Every information has-assertion equal-to 'understanding'.
Every information has-interpretation different-from 'facts'.
Every information has-interpretation equal-to 'understanding'.
Every information has-meaning different-from 'facts'.
Every information has-meaning equal-to 'understanding'.
Every information has-proposition different-from 'facts'.
Every information has-proposition equal-to 'understanding'.
Every information has-realization equal-to 'facts'.
Every information has-realization different-from 'understanding'.

Comment: 'Primitive axioms specifications'.

Every system be-moderately-correlated-with requirement.
Every system be-strongly-correlated-with technology.
Every system be-strongly-correlated-with model.
Every system be-moderately-correlated-with product.
Every system be-strongly-correlated-with use.
Every system be-strongly-correlated-with develop.
Every system be-strongly-correlated-with process.
Every system be-moderately-correlated-with information.
Every requirement be-moderately-correlated-with system.
Every requirement be-strongly-correlated-with design.
Every requirement be-strongly-correlated-with cost.
Every requirement be-strongly-correlated-with product.
Every requirement be-moderately-correlated-with use.
Every requirement be-strongly-correlated-with develop.
Every requirement be-moderately-correlated-with process.

APPENDIX E

technology design ONTOLOGY ENCODING (continued)

Every requirement be-moderately-correlated-with information.
Every design be-strongly-correlated-with requirement.
Every design be-moderately-correlated-with model.
Every design be-moderately-correlated-with cost.
Every design be-moderately-correlated-with part.
Every design be-moderately-correlated-with use.
Every design be-moderately-correlated-with develop.
Every design be-moderately-correlated-with process.
Every design be-moderately-correlated-with information.
Every technology be-strongly-correlated-with system.
Every technology be-moderately-correlated-with develop.
Every model be-strongly-correlated-with system.
Every model be-moderately-correlated-with design.
Every model be-strongly-correlated-with cost.
Every model be-strongly-correlated-with product.
Every model be-moderately-correlated-with use.
Every model be-strongly-correlated-with process.
Every model be-moderately-correlated-with information.
Every cost be-strongly-correlated-with requirement.
Every cost be-moderately-correlated-with design.
Every cost be-strongly-correlated-with model.
Every cost be-moderately-correlated-with product.
Every cost be-moderately-correlated-with use.
Every part be-moderately-correlated-with design.
Every product be-moderately-correlated-with system.
Every product be-strongly-correlated-with requirement.
Every product be-strongly-correlated-with model.
Every product be-moderately-correlated-with cost.
Every product be-moderately-correlated-with use.
Every product be-moderately-correlated-with develop.
Every product be-moderately-correlated-with process.
Every use be-strongly-correlated-with system.
Every use be-moderately-correlated-with requirement.
Every use be-moderately-correlated-with design.
Every use be-moderately-correlated-with model.
Every use be-moderately-correlated-with cost.
Every use be-moderately-correlated-with product.
Every use be-strongly-correlated-with develop.
Every use be-strongly-correlated-with process.
Every use be-moderately-correlated-with information.
Every develop be-strongly-correlated-with system.

APPENDIX E

technology design ONTOLOGY ENCODING (continued)

Every develop be-strongly-correlated-with requirement.
Every develop be-moderately-correlated-with design.
Every develop be-moderately-correlated-with technology.
Every develop be-moderately-correlated-with product.
Every develop be-strongly-correlated-with use.
Every develop be-strongly-correlated-with process.
Every develop be-strongly-correlated-with information.
Every process be-strongly-correlated-with system.
Every process be-moderately-correlated-with requirement.
Every process be-moderately-correlated-with technology.
Every process be-strongly-correlated-with model.
Every process be-moderately-correlated-with product.
Every process be-strongly-correlated-with use.
Every process be-strongly-correlated-with develop.
Every process be-strongly-correlated-with information.
Every information be-moderately-correlated-with system.
Every information be-moderately-correlated-with requirement.
Every information be-moderately-correlated-with design.
Every information be-moderately-correlated-with model.
Every information be-moderately-correlated-with use.
Every information be-moderately-correlated-with develop.
Every information be-moderately-correlated-with process.

APPENDIX F

technology insertion design ONTOLOGY ENCODING

Title: 'technology insertion design'.
Author: 'Kevin J Michael'.
Namespace: 'http://ontorion.com/namespace'.

Comment: 'Primitive concept definitions'.

Every system is a primitive-concept.
Every technology is a primitive-concept.
Every design is a primitive-concept.
Every cost is a primitive-concept.
Every insert is a primitive-concept.
Every product is a primitive-concept.
Every use is a primitive-concept.
Every develop is a primitive-concept.
Every process is a primitive-concept.
Every plan is a primitive-concept.
Every requirement is a primitive-concept.
Every capability is a primitive-concept.
Every evaluate is a primitive-concept.

Comment: 'Primitive concepts existential attribute specifications'.

Every governance is a system.
Every interaction is a system.
Every purpose is a system.
Every transformation is a system.
Every application is a technology.
Every engineering is a technology.
Every knowledge is a technology.
Every realization is a technology.
Every scientific is a technology.
Every creation is a design.
Every devise is a design.
Every purpose is a design.
Every expenditure is a cost.
Every between is a insert.
Every placement is a insert.
Every within is a insert.
Every assemblage is a product.
Every function is a product.
Every performance is a product.

APPENDIX F

technology insertion design ONTOLOGY ENCODING (continued)

Every accomplish is a use.
Every purpose is a use.
Every evolution is a develop.
Every innovation is a develop.
Every action is a process.
Every course is a process.
Every intention is a process.
Every sequence is a plan.
Every steps is a plan.
Every condition is a requirement.
Every necessary is a requirement.
Every boundary is a capability.
Every limit is a capability.
Every assess is a evaluate.
Every measure is a evaluate.

Comment: 'Primitive concepts state modification attribute specifications'.

Every system has-boundary greater-or-equal-to 'governance'.
Every system has-boundary different-from 'interactions'.
Every system has-boundary that-matches-pattern 'purpose'.
Every system has-boundary greater-or-equal-to 'transformation'.
Every system has-coordination lower-or-equal-to 'governance'.
Every system has-coordination equal-to 'interactions'.
Every system has-coordination that-matches-pattern 'purpose'.
Every system has-coordination equal-to 'transformation'.
Every system has-complexity different-from 'governance'.
Every system has-complexity equal-to 'interactions'.
Every system has-complexity different-from 'purpose'.
Every system has-complexity different-from 'transformation'.
Every system has-coupling different-from 'governance'.
Every system has-coupling equal-to 'interactions'.
Every system has-coupling different-from 'purpose'.
Every system has-coupling equal-to 'transformation'.
Every system has-dynamic different-from 'governance'.
Every system has-dynamic equal-to 'interactions'.
Every system has-dynamic different-from 'purpose'.
Every system has-dynamic equal-to 'transformation'.
Every system has-environment greater-than 'governance'.
Every system has-environment greater-than 'interactions'.
Every system has-environment greater-than 'purpose'.

APPENDIX F

technology insertion design ONTOLOGY ENCODING (continued)

Every system has-environment greater-than 'transformation'.
Every system has-homeostasis different-from 'governance'.
Every system has-homeostasis equal-to 'interactions'.
Every system has-homeostasis different-from 'purpose'.
Every system has-homeostasis equal-to 'transformation'.
Every system has-inputs different-from 'governance'.
Every system has-inputs different-from 'interactions'.
Every system has-inputs equal-to 'purpose'.
Every system has-inputs equal-to 'transformation'.
Every system has-interdependency different-from 'governance'.
Every system has-interdependency different-from 'interactions'.
Every system has-interdependency equal-to 'purpose'.
Every system has-interdependency different-from 'transformation'.
Every system has-niche different-from 'governance'.
Every system has-niche different-from 'interactions'.
Every system has-niche equal-to 'purpose'.
Every system has-niche different-from 'transformation'.
Every system has-outputs different-from 'governance'.
Every system has-outputs different-from 'interactions'.
Every system has-outputs different-from 'purpose'.
Every system has-outputs equal-to 'transformation'.
Every system has-pluralism equal-to 'governance'.
Every system has-pluralism different-from 'interactions'.
Every system has-pluralism equal-to 'purpose'.
Every system has-pluralism different-from 'transformation'.
Every system has-policy equal-to 'governance'.
Every system has-policy different-from 'interactions'.
Every system has-policy equal-to 'purpose'.
Every system has-policy different-from 'transformation'.
Every system has-wholeness different-from 'governance'.
Every system has-wholeness different-from 'interactions'.
Every system has-wholeness equal-to 'purpose'.
Every system has-wholeness different-from 'transformation'.
Every technology has-capacity equal-to 'application'.
Every technology has-capacity equal-to 'knowledge'.
Every technology has-capacity equal-to 'realization'.
Every technology has-capacity equal-to 'function'.
Every technology has-capacity different-from 'requirement'.
Every technology has-performance equal-to 'application'.
Every technology has-performance equal-to 'knowledge'.
Every technology has-performance equal-to 'realization'.

APPENDIX F

technology insertion design ONTOLOGY ENCODING (continued)

Every technology has-performance equal-to 'function'.
Every technology has-performance different-from 'requirement'.
Every technology has-robustness equal-to 'application'.
Every technology has-robustness equal-to 'knowledge'.
Every technology has-robustness equal-to 'realization'.
Every technology has-robustness equal-to 'function'.
Every technology has-robustness different-from 'requirement'.
Every technology has-stability equal-to 'application'.
Every technology has-stability equal-to 'knowledge'.
Every technology has-stability equal-to 'realization'.
Every technology has-stability equal-to 'function'.
Every technology has-stability different-from 'requirement'.
Every design has-action equal-to 'creation'.
Every design has-action equal-to 'devise'.
Every design has-action that-matches-pattern 'purpose'.
Every design has-action different-from 'system'.
Every design has-action different-from 'requirement'.
Every design has-function equal-to 'creation'.
Every design has-function equal-to 'devise'.
Every design has-function that-matches-pattern 'purpose'.
Every design has-function different-from 'system'.
Every design has-function different-from 'requirement'.
Every design has-pattern equal-to 'creation'.
Every design has-pattern equal-to 'devise'.
Every design has-pattern that-matches-pattern 'purpose'.
Every design has-pattern different-from 'system'.
Every design has-pattern different-from 'requirement'.
Every design has-representation equal-to 'devise'.
Every design has-representation equal-to 'creation'.
Every design has-representation that-matches-pattern 'purpose'.
Every design has-representation different-from 'system'.
Every design has-representation different-from 'requirement'.
Every cost has-amount equal-to 'expenditure'.
Every cost has-amount different-from 'requirement'.
Every cost has-denomination equal-to 'expenditure'.
Every cost has-denomination different-from 'requirement'.
Every cost has-time equal-to 'expenditure'.
Every cost has-time different-from 'requirement'.
Every insert has-interaction equal-to 'between'.
Every insert has-interaction equal-to 'placement'.
Every insert has-interaction equal-to 'within'.

APPENDIX F

technology insertion design ONTOLOGY ENCODING (continued)

Every insert has-interface equal-to 'between'.
Every insert has-interface equal-to 'placement'.
Every insert has-interface equal-to 'within'.
Every insert has-location equal-to 'between'.
Every insert has-location equal-to 'placement'.
Every insert has-location equal-to 'within'.
Every insert has-interaction different-from 'design'.
Every insert has-interaction different-from 'plan'.
Every insert has-interaction different-from 'requirement'.
Every insert has-interface different-from 'design'.
Every insert has-interface different-from 'plan'.
Every insert has-interface different-from 'requirement'.
Every product has-entities equal-to 'assemblage'.
Every product has-entities equal-to 'function'.
Every product has-entities equal-to 'performance'.
Every product has-interactions equal-to 'assemblage'.
Every product has-interactions equal-to 'function'.
Every product has-interactions equal-to 'performance'.
Every use has-method equal-to 'accomplish'.
Every use has-method equal-to 'purpose'.
Every use has-objectives equal-to 'accomplish'.
Every use has-objectives equal-to 'purpose'.
Every develop has-change equal-to 'evolution'.
Every develop has-change different-from 'innovation'.
Every develop has-create different-from 'evolution'.
Every develop has-create equal-to 'innovation'.
Every develop has-new equal-to 'evolution'.
Every develop has-new equal-to 'innovation'.
Every develop has-purpose different-from 'evolution'.
Every develop has-purpose equal-to 'innovation'.
Every process has-activity equal-to 'actions'.
Every process has-activity equal-to 'course'.
Every process has-activity different-from 'intention'.
Every process has-event equal-to 'actions'.
Every process has-event equal-to 'course'.
Every process has-event different-from 'intention'.
Every process has-mode equal-to 'actions'.
Every process has-mode equal-to 'course'.
Every process has-mode different-from 'intention'.
Every process has-path equal-to 'actions'.
Every process has-mode equal-to 'course'.

APPENDIX F

technology insertion design ONTOLOGY ENCODING (continued)

Every process has-mode different-from 'intention'.
Every process has-purpose different-from 'actions'.
Every process has-purpose different-from 'course'.
Every process has-purpose equal-to 'intention'.
Every plan has-arrangement equal-to 'sequence'.
Every plan has-arrangement equal-to 'steps'.
Every plan has-series equal-to 'sequence'.
Every plan has-series equal-to 'steps'.
Every plan has-location equal-to 'sequence'.
Every plan has-location equal-to 'steps'.
Every plan has-arrangement different-from 'system'.
Every plan has-series different-from 'system'.
Every plan has-location different-from 'system'.
Every plan has-arrangement different-from 'design'.
Every plan has-series different-from 'design'.
Every plan has-location different-from 'design'.
Every plan has-arrangement different-from 'requirement'.
Every plan has-series different-from 'requirement'.
Every plan has-location different-from 'requirement'.
Every plan has-arrangement different-from 'use'.
Every plan has-series different-from 'use'.
Every plan has-location different-from 'use'.
Every requirement has-attribute equal-to 'condition'.
Every requirement has-attribute equal-to 'necessary'.
Every requirement has-constraint equal-to 'condition'.
Every requirement has-constraint equal-to 'necessary'.
Every requirement has-function equal-to 'condition'.
Every requirement has-function equal-to 'necessary'.
Every requirement has-level equal-to 'condition'.
Every requirement has-level equal-to 'necessary'.
Every requirement has-value equal-to 'condition'.
Every requirement has-value equal-to 'necessary'.
Every capability has-degree equal-to 'boundary'.
Every capability has-degree equal-to 'limit'.
Every capability has-demarcation equal-to 'boundary'.
Every capability has-demarcation equal-to 'limit'.
Every capability has-extent equal-to 'boundary'.
Every capability has-extent equal-to 'limit'.
Every capability has-termination equal-to 'boundary'.
Every capability has-termination equal-to 'limit'.
Every capability has-degree different-from 'design'.

APPENDIX F

technology insertion design ONTOLOGY ENCODING (continued)

Every capability has-demarcation different-from 'design'.
Every capability has-extent different-from 'design'.
Every capability has-termination different-from 'design'.
Every capability has-degree different-from 'use'.
Every capability has-demarcation different-from 'use'.
Every capability has-extent different-from 'use'.
Every capability has-termination different-from 'use'.
Every capability has-degree different-from 'plan'.
Every capability has-demarcation different-from 'plan'.
Every capability has-extent different-from 'plan'.
Every capability has-termination different-from 'plan'.
Every evaluate has-estimate equal-to 'assess'.
Every evaluate has-estimate equal-to 'measure'.
Every evaluate has-classification equal-to 'assess'.
Every evaluate has-classification equal-to 'measure'.
Every evaluate has-determination equal-to 'assess'.
Every evaluate has-determination equal-to 'measure'.
Every evaluate has-amount equal-to 'assess'.
Every evaluate has-amount equal-to 'measure'.

Comment: 'Primitive axioms specifications'.

Every system be-moderately-correlated-with technology.
Every system be-moderately-correlated-with design.
Every system be-moderately-correlated-with cost.
Every system be-moderately-correlated-with insert.
Every system be-moderately-correlated-with product.
Every system be-moderately-correlated-with use.
Every system be-moderately-correlated-with develop.
Every system be-moderately-correlated-with process.
Every system be-moderately-correlated-with plan.
Every system be-moderately-correlated-with requirement.
Every system be-moderately-correlated-with evaluate.
Every technology be-moderately-correlated-with system.
Every technology be-moderately-correlated-with cost.
Every technology be-strongly-correlated-with insert.
Every technology be-moderately-correlated-with product.
Every technology be-moderately-correlated-with develop.
Every technology be-moderately-correlated-with plan.
Every technology be-strongly-correlated-with capability.
Every design be-moderately-correlated-with system.

APPENDIX F

technology insertion design ONTOLOGY ENCODING (continued)

Every design be-strongly-correlated-with cost.
Every design be-strongly-correlated-with insert.
Every design be-strongly-correlated-with product.
Every design be-strongly-correlated-with use.
Every design be-moderately-correlated-with plan.
Every design be-strongly-correlated-with evaluate.
Every cost be-moderately-correlated-with system.
Every cost be-moderately-correlated-with technology.
Every cost be-strongly-correlated-with design.
Every cost be-strongly-correlated-with insert.
Every cost be-strongly-correlated-with product.
Every cost be-strongly-correlated-with use.
Every cost be-moderately-correlated-with plan.
Every cost be-strongly-correlated-with evaluate.
Every insert be-moderately-correlated-with system.
Every insert be-strongly-correlated-with technology.
Every insert be-strongly-correlated-with design.
Every insert be-strongly-correlated-with cost.
Every insert be-moderately-correlated-with develop.
Every insert be-moderately-correlated-with plan.
Every insert be-strongly-correlated-with capability.
Every product be-moderately-correlated-with system.
Every product be-moderately-correlated-with technology.
Every product be-strongly-correlated-with design.
Every product be-strongly-correlated-with cost.
Every product be-strongly-correlated-with use.
Every product be-moderately-correlated-with plan.
Every product be-moderately-correlated-with requirement.
Every product be-strongly-correlated-with evaluate.
Every use be-strongly-correlated-with system.
Every use be-strongly-correlated-with design.
Every use be-strongly-correlated-with cost.
Every use be-strongly-correlated-with product.
Every use be-strongly-correlated-with plan.
Every use be-moderately-correlated-with requirement.
Every use be-strongly-correlated-with evaluate.
Every develop be-moderately-correlated-with system.
Every develop be-moderately-correlated-with technology.
Every develop be-moderately-correlated-with insert.
Every develop be-strongly-correlated-with process.
Every develop be-moderately-correlated-with requirement.

APPENDIX F

technology insertion design ONTOLOGY ENCODING (continued)

Every process be-moderately-correlated-with system.
Every process be-strongly-correlated-with develop.
Every process be-moderately-correlated-with requirement.
Every plan be-moderately-correlated-with system.
Every plan be-moderately-correlated-with technology.
Every plan be-moderately-correlated-with design.
Every plan be-moderately-correlated-with cost.
Every plan be-moderately-correlated-with insert.
Every plan be-moderately-correlated-with product.
Every plan be-strongly-correlated-with use.
Every plan be-moderately-correlated-with requirement.
Every requirement be-moderately-correlated-with system.
Every requirement be-moderately-correlated-with product.
Every requirement be-moderately-correlated-with use.
Every requirement be-moderately-correlated-with develop.
Every requirement be-moderately-correlated-with process.
Every requirement be-moderately-correlated-with plan.
Every requirement be-moderately-correlated-with evaluate.
Every capability be-strongly-correlated-with technology.
Every capability be-strongly-correlated-with insert.
Every evaluate be-moderately-correlated-with system.
Every evaluate be-strongly-correlated-with design.
Every evaluate be-strongly-correlated-with cost.
Every evaluate be-strongly-correlated-with product.
Every evaluate be-strongly-correlated-with use.
Every evaluate be-moderately-correlated-with plan.

References:
[84TC3YFU] 'D:/Fluent_Ontologies/technology_design/technology_design_ontology' ('D:/Fluent_Ontologies/technology_design/technology_design_ontology.encnl').

VITA

Kevin J. Michael received a B.S. Computer Science from the University of Maryland and a Master of.Science in Management Information Systems from the University of Maryland. Kevin has worked as in engineer at the Naval Surface Warfare Center in Dahlgren, Virginia, and is employed as an engineer for the Census Bureau in Washington, D.C.