2018

# Design of Complex Engineered Systems and the Effectiveness of Organizational Networks

Bryan Still
*University of Rhode Island*, bryanstill@icloud.com

# DESIGN OF COMPLEX ENGINEERED SYSTEMS AND THE EFFECTIVENESS OF ORGANIZATIONAL NETWORKS

BY

BRYAN STILL

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

INDUSTRIAL AND SYSTEMS ENGINEERING

UNIVERSITY OF RHODE ISLAND

2018

DOCTOR OF PHILOSOPHY DISSERTATION

OF

BRYAN STILL

APPROVED:

Dissertation Committee:

Major Professor: Valerie Maier-Speredelozzi

Jyh-Hone Wang

Michael Barrus

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2018

## ABSTRACT

Cost and schedule overruns have become increasingly common in projects that set out to design and deliver complex engineered systems. Noting the well-established relationship between products and the organizations that design them, this study evaluates the effectiveness of different organizational networks at designing complex engineered systems using agent-based modeling. Specifically, it compares matrix and military staff organizational networks to random and multiscale networks, modeling design as an activity that requires organizations to create design artifacts and share information. It examines the nature of design, the role of product architecture, the nature of complexity and how it affects projects, and the characteristics that improve organizational robustness to congestion. Results indicate matrix organizations are particularly susceptible to congestion failure, while military staff and multiscale networks are more robust to congestion failure, with military staff networks having performance comparable to multiscale networks over a range of scenarios. Results further indicate simple changes to organizational behavior improve performance and robustness to congestion, with decentralization being especially beneficial. Finally, results confirm the utility of agent-based modeling for understanding the dynamics of complex systems.

**ACKNOWLEDGMENTS**

I gratefully acknowledge the help of those who have helped make this dissertation possible. First, my advisor, Professor Valerie Maier-Speredelozzi, Ph. D., and members of my committee: Professors Jyh-Hone Wang, Ph. D., Michael Barrus, Ph. D., and Christopher Hunter, Ph.D. Second, individuals who provided recommendations and guidance on possible research topics: John Dickman and Andy Stoddard. Third, colleagues and mentors who have trained, guided and inspired me over the years, including Will Bundy, Dick Crowell, Phil McLaughlin, Michael Jabaley, and Chuck Merkel. Finally, my wife, Jennifer, whose love and support make all things possible—and worthwhile.

# DEDICATION

For my Mother, who dreamt I would be a Doctor someday.  She meant physician, but she also encouraged me to make my own way.

# TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

Cost and schedule overruns have become increasingly common in large defense programs that attempt to build systems with improved performance and lifecycle characteristics, often using novel, untested, and complex product architectures. (Murray, et al., 2011) Given the well documented relationship between product architecture and the structure of the product development organization, it is logical to examine organizational structure for causes and factors explaining the inability of design organizations to manage the complexity associated with the design of large engineered systems. This study will therefore examine the effectiveness of different organizational networks at designing complex engineered systems, modeling design as an activity that requires the creation of design products and the sharing of information and comparing the performance of real-world organizational networks to ideal ones in order to identify ways real-world networks could be modified to improve performance.

## Research Motivation

A 2011 report prepared for the Defense Advanced Research Projects Agency (DARPA) concluded cost and scheduled overruns in defense programs result from "systematic mismanagement of the inherent complexity associated with the design of these systems." (Murray, et al., 2011) Sinha and de Weck (2013) reported 13 aerospace projects reviewed by the Government Accountability Office between 2008 and 2013 experienced cost growth of 55% or more. (Sinha & de Weck, IDETC/CIE

2013, 2013) More recently, major shipbuilding programs have experienced similar cost and schedule overruns. A 2015 GAO report noted the Ford-class aircraft carrier was more than $2 billion over budget and was unlikely to achieve promised performance with regard to aircraft launch and recovery rates due to unreliability of systems. (Government Accountability Office, 2015) Such problems are not unique to the defense sector. General Motors posted a $4.3 billion loss in the fourth quarter of 2009 as the cost of its new Chevy Volt approached $40,000 per car, doubling initial estimates. (Simpson & Martins, 2012)

The Nature of Design

Herbert Simon (1996) described design as the process of devising "courses of action aimed at changing existing situations into preferred ones," observing engineers and other designers are concerned with how things ought to function in order to accomplish goals, and arguing synthetic or artificial objects, i.e., *artifacts*, are "the central objective of engineering activity and skill." (Simon, 1996) A key step in the design of engineered system is establishing product architecture, the scheme that translates functions and objectives into physical components. Product architecture drives decision-making and affects product performance and defining product architecture involves three inter-related activities: identification of functional requirements and arrangement of functional elements; mapping functional requirements to physical systems or components; and defining physical interfaces between systems or components. (Ulrich K. , 1995)

Organizational Structure and Product Architecture

Researchers have long recognized the interplay between products and the organizations that design them. Conway (1968) argued organizations produce designs that reflect their communication structures, thus design efforts should be organized according to the need for communication. (Conway, 1968) Henderson and Clark (1990) examined the nature of innovation and concluded changes to product architecture challenge traditional firms by destroying existing knowledge embedded in the firms' organizational and communication structures. During periods of innovation, firms require the ability to develop knowledge and synthesize designs, but once a dominant design is established, firms stop investing in learning about alternative configurations and instead invest in refinements. They argue the effect of architectural innovation depends on how organizations learn and suggest the "fashion for cross-functional teams and open organizational environments" may be a response to perceptions on the challenges of architectural innovation. (Henderson & Clark, 1990)

Organizational structure defines how people work together to accomplish objectives and create value, and includes formal hierarchy, the decomposition of the organization into functional elements, such as directorates, departments, divisions, work centers, and individuals; reporting relationships and lines of authority; and informal teaming relationships that cross both vertical and horizontal hierarchical lines. Given the well-established relationship between product architecture and organizational structure, one might expect firms would align the two in order to create products that better meet objectives, but in practice, firms consider a variety of business and management imperatives when setting organizational structure.

Robust Organizations

Dodds, Watts and Sabel (2003) examined the dynamics of information exchange in organizational networks and introduced an organizational network model that incrementally adds links to a hierarchical backbone according to a stochastic rule. They identified a class of networks, which they call “multiscale networks,” that exhibit “ultra-robustness,” meaning they simultaneously reduce the likelihood an individual node will fail because of congestion *and* the likelihood the overall network will fail if congestion failures do occur at individual nodes. Multiscale networks exhibit these properties with the addition of relatively few links, which suggests “ultra-robust organizational networks can be generated in an efficient and scalable manner.” (Dodds, Watts, & Sabel, 2003)

Economists have long studied organizational structure, emphasizing efficiency over robustness and focusing on multilevel hierarchies, which offer advantages for exercising control, accumulating knowledge, and making decisions. These advantages assume tasks can be easily decomposed into smaller subtasks that can be accomplished independently, but modern organizations face multidimensional problems characterized by complexity and ambiguity, where problem solving becomes a collective activity characterized by collaboration among individuals, teams and organizations. Under these conditions, the chief concern is not efficiency, achieved by minimizing costly links, but robustness, achieved by preventing individual nodes from being overwhelmed and protecting the network from catastrophic failure when congestion does occur. (Dodds, Watts, & Sabel, 2003)

Understanding Complexity and Attempts to Measure It

Sinha and de Weck argue "today's large-scale engineered systems are becoming increasingly complex" due to demands for increased performance and improved lifecycle characteristics, but *complexity* is hard to quantify. (Sinha & de Weck, 2013) Mitchell (2009) identifies several characteristics of complex systems, including complex collective behaviors, such as self-organization and adaptation through learning or evolution, but notes no single science or theory of complexity yet exists, despite the many books and articles written on the subject. (Mitchell, 2009) Page (2009) provides a useful framework for understanding complexity, defining complex *adaptive* systems in terms of four necessary characteristics of the agents or elements in the system: diversity, connectedness, interdependence, and adaptation, arguing adaptation is the key characteristic separating complex systems from merely complicated ones. (Page, 2009) In fact, much of the confusion about the meaning of complexity stems from this question about what separates complex from complicated systems.

In common usage, when someone says a thing is "complex," they most often mean hard, challenging or complicated, but for complex systems, the term is also used to describe a variety of rich and unexpected behaviors, including self-organization, emergence, robustness, susceptibility to large events, and non-linear dynamics. In *Micromotives and Macrobehavior*, Schelling (2006) describes how individual choices affect the overall behavior of complex systems in non-obvious ways, observing: "it is not easy to tell from the aggregate phenomenon just what the motives are behind individual decisions or how strong they are." (Schelling, 2006) This kind of micro-macro disconnect is central to the idea of emergence in complex systems, but a similar disconnect can occur in "merely complicated" systems when connections and dependencies are poorly understood.

In complex engineered systems, such as automobiles, aircraft, and ships, the number of connections and dependencies can quickly challenge the limits of human cognition. Even though individual elements of the system may perform in predictable ways, interactions among elements can lead to unexpected macro behaviors. Such behaviors may be predictable in theory, but not in any meaningful or practical way, thus merely complicated, large engineered systems often exhibit *quasi-emergent* behaviors comparable to complex adaptive systems.

Several authors have proposed methods or measures to quantify complexity, but there is no single, widely accepted metric, nor even universal agreement that complexity can be measured. Mitchell surveys different approaches and identifies several categories, including counting methods; entropy-based methods, notably Shannon entropy; algorithmic information content; logical and thermodynamic depth; statistical methods; fractal dimension; and degree of hierarchy. She concludes different measures individually capture something about the notion of complexity but have practical limitations that make them not useful for characterizing real systems. (Mitchell, 2009)

### Summary

To meet demands for improved performance, designers of large engineered systems create new products with increasingly complex architectures that strain the capabilities of the design organization. Unprepared to manage the design of complex engineered systems, organizations built for efficiency may find themselves overwhelmed, leading to the kinds of cost and schedule overruns documented by DARPA and the GAO. Since multiscale organizational networks have been shown to be robust to failure, it is appropriate to compare them to other organizational networks commonly used by design organizations in order to better understand the performance of design organizations and identify ways to improve their ability to manage the development of complex engineered systems. This study will therefore compare the performance of matrix organizations and military staffs, two real-world organizational networks, to random and multiscale networks, two idealized organizational networks, using agent-based modeling (ABM).

The remainder of this dissertation is arranged in four additional chapters: review of literature, methodology, findings, and conclusions. Chapter 2 presents a review of literature, which further develops the concepts and ideas introduced earlier in this introduction. Chapter 3 presents methodology and describes the phased, building block approach used to develop and implement agent-based models to examine the effectiveness of real-world and ideal organizational networks. Chapter 4 presents findings resulting from the implementation and analysis of models of organizational networks. Chapter 5 presents conclusions and recommendations.

# CHAPTER 2

## REVIEW OF LITERATURE

The following literature review addresses a variety of topics related to the design and development of complex engineered systems, and the proposed use of Agent-Based Modeling (ABM) to evaluate the effectiveness of different organizational networks at designing complex engineered systems. It begins by examining the nature of design, the elements of product development, and the role of product architecture, and then turns to organizational structure, organizational networks, and the relationship between organizational structure and product architecture. It then describes robust networks, a special class of organizational network that is simultaneously robust to congestion and connectivity failures, before exploring definitions of complexity and complex systems, as well as efforts to understand and cope with complexity, including qualitative and quantitative measures of complexity. The literature review concludes with a discussion of opportunities for improving project performance, a brief review of design structure matrices and their application to modeling products and organizations, and a description of agent-based modeling.

### The Nature of Design

Herbert Simon, declared: "everyone designs who devises courses of action aimed at changing existing situations into preferred ones." (Simon, 1996) Engineering schools have traditionally taught students how do design and make artifacts with desired characteristics, but Simon argued the mental activity that designs material artifacts is the same fundamental activity that devises plans or policies, concluding design is the foundation of professional training, separating professions from the sciences. Simon was acutely concerned by the damage to professional competence that occurred in the years following World War II, when engineering, business and other professional schools moved toward natural science and away from the "sciences of the artificial." (Simon, 1996)

Simon recognized the problem lay in the notion of "artifical science," and derogatory connotations around the term "artificial." He identified four essential features of artificial things: that they are synthesized by humans; that they may imitate natural appearance; that they are characterized in terms of functions, goals, and adaptation; and that they are often described in terms of design imperatives. Engineers and other designers are concerned with how things ought to function in order to accomplish goals, and synthetic or artificial objects, i.e. *artifacts* are "the central objective of engineering activity and skill." (Simon, 1996)

The design of artifacts involves three related considerations: the purpose or goal to be achieved, the nature of the artifact itself, and the environment in which the artifact functions. An artifact can thus be considered the interface between its own internal structure and function and its surroundings, what Simon called the "inner" and "outer" environments." Simon claimed: "description of an [artifact] in terms of its organization and functioning—its interface between inner and outer environments—is a major objective of invention and design activity." (Simon, 1996) Goals link the inner and outer systems, with the inner system representing one of several functionally equivalent sets of capabilities that can accomplish the goals and the outer environment setting the conditions required for goal achievement. Of course, this is a bit of a simplification, which Simon recognizes, acknowledging that artifacts must obey natural laws and noting we will often have to be satisfied with designs that only partially meet their objectives.

Design problems are often framed as making a choice from among fixed alternatives, where the best, or optimum, solution is selected. Simon notes, however, that actual design decisions frequently involve finding satisfactory, rather than optimal solutions, introducing the term "satisficing" to describe such decision methods. Satisficing methods search for solutions in a way that yields acceptable results with only modest search. Real-world problem solving and design methods must search for appropriate solutions, thus design involves the allocation of resources to ensure designers focus efforts on the most promising lines of inquiry. With satisficing goals, solutions are rarely unique, and the design effort seeks sufficent, rather than necessary, answers. (Simon, 1996)

Simon describes a typical approach to search, in which possible paths are explored, with results stored in a "tree" structure that reflects the value assigned to each branch. The values guide further exploration, and the search process gathers information on problem structure that can be used to discover a solution. The search process therefore serves two complementary purposes: finding a solution and understanding problem structure. Simon identifies decomposition as a powerful tool for solving complex problems. This technique, which is foundational to systems engineering, breaks complex systems into distinct parts, often along functional lines, allowing each part to be designed somewhat independently. Simon notes, however, that "there is no reason to expect that the decomposition of the complete design into functional components will be unique," identifying organizational theory as a field keenly concerned with the "issue of alternative decompositions of a collection of interrelated tasks." (Simon, 1996)

Simon also addresses the topic of problem representation, noting the importance of representations that make solutions more obvious, and the need for a better taxonomy for describing and classifying different classes of problem representations. He concludes by presenting the elements a program in design that incorporates the preceding topics, noting a number of well established design processes that refute any notion that desgin can be reduced to cookbook approaches, the same notion that once threatened to force design from the curricula of engineering and other professional schools. (Simon, 1996)

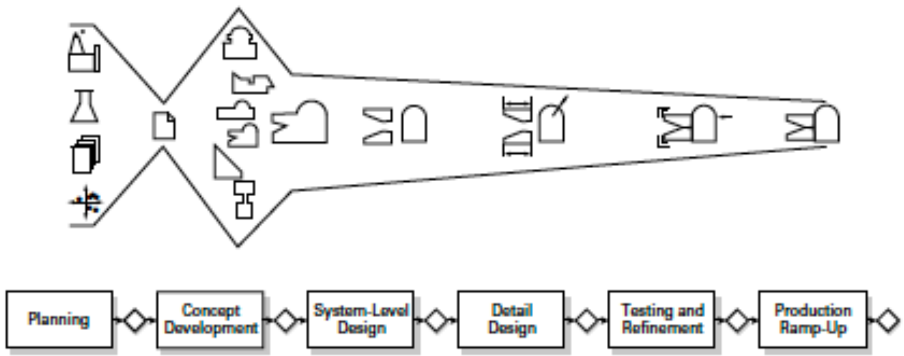
Product Design and Development

A *product* is anything sold to a customer, and *product development* is the set of activities that bring the product to market. By its nature, product development is cross-functional, requiring contributions from numerous functions in a firm, including marketing, design, engineering, and manufacturing. (Ulrich & Eppinger, 1995) Figure 1 presents a generic product development process showing the major activities required to transform a concept into a finished product. Of course, every organization follows a different process, but having a well defined proess offers benefits in terms of quality, coordination, planning, management and process improvement. The generic product development process has five phases:

1. Concept development, which identifies alternative concepts (descriptions of form, function and features) to meet market and customer requirements, evaluates those alternatives, and selects one for further development;
2. System-level design, which defines the product architecture and divides the product into sub-systems and components;
3. Detail design, which provides a complete specification in the form of *control documentation* (e.g., drawings of parts and production tooling, specifications, and fabrication plans) for all unique parts to be manufactured or purchased;
4. Testing and refinement, which evaluates prototypes to verify compliance with customer requirements; and
5. Production, which makes the intended product. (Ulrich & Eppinger, 1995)

For the present study, we are primarily interested in the system and detail design phases and the interaction and communication that must occur in the design organization to create the required detail design products, termed *control documentation* or *artifacts*.

*Figure 1 - Generic Product Development Process and the Major Activities Involved (Ulrich K. , 1995)*




| | Planning | Concept Development | System-Level Design | Detail Design | Testing and Refinement | Production Ramp-Up |
|---|---|---|---|---|---|---|
| **Marketing** | • Articulate market opportunity.<br>• Define market segments. | • Collect customer needs.<br>• Identify lead users.<br>• Identify competitive products. | • Develop plan for product options and extended product family. | • Develop marketing plan. | • Develop promotion and launch materials.<br>• Facilitate field testing. | • Place early production with key customers. |
| **Design** | • Consider product platform and architecture.<br>• Assess new technologies. | • Investigate feasibility of product concepts.<br>• Develop industrial design concepts.<br>• Build and test experimental prototypes. | • Develop product architecture.<br>• Define major sub-systems and interfaces.<br>• Refine industrial design.<br>• Preliminary component engineering. | • Define part geometry.<br>• Choose materials.<br>• Assign tolerances.<br>• Complete industrial design control documentation. | • Test overall performance, reliability, and durability.<br>• Obtain regulatory approvals.<br>• Assess environmental impact.<br>• Implement design changes. | • Evaluate early production output. |
| **Manufacturing** | • Identify production constraints.<br>• Set supply chain strategy. | • Estimate manufacturing cost.<br>• Assess production feasibility. | • Identify suppliers for key components.<br>• Perform make-buy analysis.<br>• Define final assembly scheme. | • Define piece-part production processes.<br>• Design tooling.<br>• Define quality assurance processes.<br>• Begin procurement of long-lead tooling. | • Facilitate supplier ramp-up.<br>• Refine fabrication and assembly processes.<br>• Train workforce.<br>• Refine quality assurance processes. | • Begin full operation of production system. |
| **Other Functions** | • Research: Demonstrate available technologies.<br>• Finance: Provide planning goals.<br>• General Management: Allocate project resources. | • Finance: Facilitate economic analysis.<br>• Legal: Investigate patent issues. | • Finance: Facilitate make-buy analysis.<br>• Service: Identify service issues. | | • Sales: Develop sales plan. | • General Management: Conduct postproject review. |

<u>The Role of Product Architecture</u>

Eppinger and Browning (2012) define product or system architecture as “the arrangement of components interacting to perform specified functions,” noting that architecture is represented by individual components, their relationships to one another and the environment, and principles guiding design. (Eppinger & Browning, 2012)

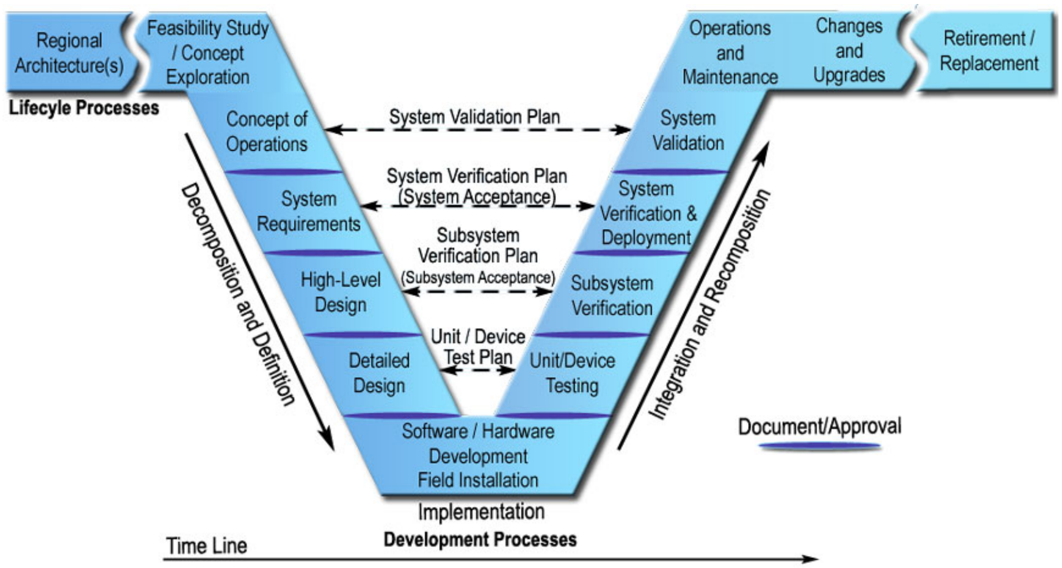
When designing products or engineered systems, one commonly decomposes the product or system into smaller elements, such as subsystems, modules, and components, that must be integrated to work together and achieve performance objectives. The discipline of systems engineering focuses on planning and controlling component interactions to deliver system-level performance. The Systems Engineering "V," shown in Figure 2, illustrates the process of designing and developing engineered systems.

Ulrich (1995) provides a comprehensive survey of product architectures and articulates how architecture affects areas critical to product development. He draws on concepts from a range of fields, including design theory and operations management, and provides a useful framework for understanding the design trade-offs affected by product architecture. Ulrich defines product architecture as "the scheme by which the function of a product is allocated to physical components," and argues product architecture's importance to decision making, noting that product architecture drives performance and that manufacturing firms have flexibility when choosing product architecture. Product architecture considers three inter-related activities: identification of functional requirements and arrangement of functional elements; mapping functional requirements to physical systems or components; and defining physical interfaces between systems or components. *Modular* architectures have a one-to-one mapping of functional requirements to systems or components *and* de-coupled interfaces, while *integral* architectures have a complex (e.g., one-to-many) mapping of functional requirements to systems or components *or* coupled interfaces.

A coupled interface exists when a change to one system or component requires a change to the related (i.e., coupled) system or component. (Ulrich K. , 1995)

Modular architectures can be further divided into slot, bus or sectional types. In a *slot* architecture, components have different interfaces such that components cannot be interchanged with one another. For example, a car radio has a different interface than the car's speedometer. *Bus* architectures provide a common bus to which other components connect or attach using the same kind of interface. Examples include expansion slots in personal computers and shelving systems. Finally, in *sectional* architectures, components use the same kind of interface, but there is no single element to which all others connect. Examples include piping systems and sectional sofas. Of course, these descriptions all represent ideal types—real products may use multiple types of architectures simultanesouly, or blur lines of distinction. Ulrich notes manufacturing firms have significant flexibility when choosing product architecture and argues architecture may result more from incremental evolution rather than deliberate choice. He also notes many authors have argued the superiority of modular architectures, but suggests no architecture should be considered ideal. (Ulrich K. , 1995)

*Figure 2 - The Systems Engineering V (Department of Transportation, 2007)*




<u>Organizational Structure</u>

Successful product development requires an effective development process and effective development staff. Ulrcih and Eppinger (1995) define "product development organizations" as "the scheme by which individual designers and developers are linked together into groups," noting that links can be formal or informal, and can include reporting relationships, financial arrangements, and physical layout. (Ulrich & Eppinger, 1995) Individuals in the product development organization can be classified by either function or project. Functions are areas of responsibility that generally require specialized training or skills, such as marketing, design, engineering, operations management, and manufacturing. Regardless of function, individuals use their expertise on different projects. (Ulrich & Eppinger, 1995)

Organizational strcuture identifies the people in an organization, their relationships to one another and the organization's environment, and the principles governing its purpose and development. The effective development of products and engineered systems depends on the efficient and effective flow of information between people and across organizational divisions. Leaders may want to enable "more and better communication, the free flow of ideas, and the open sharing of issues and concerns, with hopes of building consensus and preempting problems," but the free flow of information can go too far, creating information overload that actually impedes effective communication. Leaders therefore seek to manage the flow of information to facilitate effective execution of complex projects through purposeful organizational structures. Rational organization design enables effective communication by improving team structure and providing insight on the application of integrative or coordination mechanisms. (Eppinger & Browning, 2012)

Organizational structure defines how people work together to accomplish objectives and create value. Organizational structure includes formal hierarchy, the decomposition of the organization into functional elements, such as directorates, departments, divisions, work centers, and individuals; reporting relationships and lines of authority; and informal teaming relationships that cross both vertical and horizontal hierarchical lines. Given the well established relationship between product architecture and organizational structure, one might expect firms would align the two in order to create products that better meet objectives, but in practice, firms consider a variety of business and managerial imperatives when setting organizational structure. The next section examines elements of organizational structure, including descriptions of structures found in real-world organizations.

Types of Organizational Networks

The Role and Nature of Hierarchies. Herbert Simon examined the relationship and interplay between hierarchies—systems composed of inter-related subsystems, which are themselves hierarchical until reaching some elemental structure—and argued hierarchy is one of the "central structural schemes that the architect of complexity uses." (Simon, 1996) Hierarchic systems explicitly include those *not* based on subordination; examples include formal organizations, such as firms, businesses, and government entities; societies, divided into units like families, villages, tribes, or nations; biological and physical systems, including products and complex engineered systems; and symbolic systems.

Hierarchies decompose the whole into modular parts or subsystems, where one can distinguish interactions within a subsystem from interactions between or among subsystems. In the context of the present study, this feature is seen in both the decomposition of products and engineered systems described by product architecture, as well as the decomposition of organizations into elements such as directorates or divisions.

A key property of hierarchies is *near decomposability*, which refers to the idea that intra-component linkages and interactions are generally stronger than inter-component interactions. This feature separates high-frequency dynamics related to internal structure from low-frequency interactions among components. In a nearly decomposable system, inter-component interactions are weak, but not negligible. (Simon, 1996) In fact, it is these weak interactions, which are often poorly understood, that give rise to complexity, a topic explored in greater depth in a subsequent section.

In Chapter 9 of *Six Degrees of Separation: The Science of a Connected Age*, Duncan Watts describes how today's models and theories of organizational structure trace to Adam Smith's *The Wealth of Nations*, which describes the division of labor principle Smith inferred from his observations of workers. Smith noted workers performed better when collective tasks were broken into specialized subtasks, a benefit termed returns on specialization. The division of labor harnesses returns on specialization, but does not explain why production must be accomplished by firms or why hierarchical organizations emerged as the dominant type assoicated with mass production. Nevertheless, many firms did organize that way, and the consensus of economic theory has long been that hierarchies represent the optimal organizational form. (Watts, Six Degrees, 2003)

Traditional economic theory argues that firms grow through the process of vertical integration, the periodic absorbing or jettisoning of hierarchies, but Sabel and Poire (1984) challege that theory, noting that it came about only after vertical integration had become the dominant organizational design. They argue, instead, flexible specialization, which exploits economies of scope using general prupose machinery and skilled workers, is beginning to replace vertical integration, and further argue such economies of scope are optimal when uncertainty and rapid change favor adaptability over scale. (Poire & Sabel, 1984)

<u>Random and Small World Networks</u>. Much has been written about random and small world networks. This section briefly reviews key features and concepts that

inform, or are otherwise relevant to, the study of organizational networks. The so-called "small world" phenomenon formalizes the anecdotal notion that "you are only ever six 'degrees of separation' away from anybody else on the planet." (Watts, Small Worlds, 1999) Watts and Strogatz (1998) coined the term "small-world networks" to describe networks that occupy the "middle ground" between completely regular and completely random, exhibiting short characteristic path lengths associated with random networks and high degrees of clustering associated with ordered networks. They explored simple models that can be tuned through this "middle ground" and demonstrated that real world networks exhibit small world properties. (Watts & Strogatz, 1998)

The study of small-w orld networks, and of networks in general, illustrates basic concepts from graph theory. A graph, *G(N, m)*, is a set of *N* vertices or nodes and *m* edges or links. The study of small-world networks was limited to undirected and unweigthed networks, meaning links had no direction or relative weight, and to sparse graphs, meaning the number of links, $m \ll \frac{N(N-1)}{2}$, where the right-hand quantity represents the maximum possible number of links in a network of *N* nodes. Distance between nodes can be characterized by a *characteristic path length*, *L(G)*, such as the median of the means of the shortest distances between each node. Clustering is the extent to which vertices adjacent to any vertex are connected to one another.

A common theme in the study of graphs is the comparison of network properties to those of random graphs. A random graph of order *N* consists of *N* vertices with an edge set of *m* randomly chosen edges, where *m* usually depends on *N* and *G(N, p)*, a graph of *N* vertices where everyone of the $\binom{N}{2}$ edges exists with probability *p*, ($0 < p < 1$). Random graph theory defines conditions under which a random graph contains some property Q, for example, it is connected, in the limit where $N \rightarrow \infty$. A common feature of random graphs is that most monotone properties appear suddenly at some value or function of *N*. (Watts, Small Worlds, 1999)

Matrix and Project-Based Organizations. The defining characteristic of a matrix organization is the existence of a dual chain of command, with responsibilities assigned to functional departments, such as engineering, production and marketing, and to product or project departments. Functional departments provide specialized, internal resources, while project or product departments focus on outputs. Davis and Lawrence (1978) argue a matrix organization is more than just a matrix structure: "it must be reinforced by matrix systems, such as dual control and evaluation systems, by leaders who operate comfortably with lateral decision making, and by a culutre that can negoitate open conflict and a balance of power." (Davis & Lawrence, 1978)

Ford and Randolph (1992) note terms like *matrix*, *matrix organization*, and *project organization* are often used interchangeably to refer to a cross-functional organizations that bring together people from different functional areas "to undertake a task on either a temporary basis (as in a project team) or on a relatively more permanent basis (as in a matrix organization)." The common characteristic is a hybrid organization form in which a traditional functional hierarchy is "overlayed" by a lateral project-based authority, as shown in Figure 3. Ford and Randolph note most authors place matrix organizations towards the center of a continuum, between purely functional organizations on the one hand, and purely project organizations on the other. (Ford & Randolph, 1992)  Figure 4 illustrates typical functional, product, and matrix organizations showing how matrix organizations are a hybrid of the other two.

In general, matrix organizations can be classified as heavyweight or lightweight.  In a heavyweight project organization, individual project managers report directly to the General Manager and are responsible and accountable for the success of assigned projects.  Functional managers also report to the general manager and are responsible for technical excellence.  Project managers control budgets and allocate resources and therefore have significant authority.  In a lightweight project organization, the project manager plays more of a coordination and administrative role, but has little authority. (Ulrich & Eppinger, 1995)

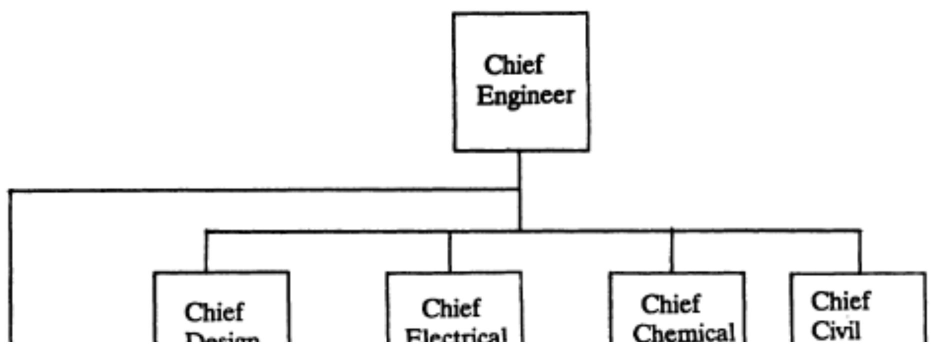



*Figure 3 - Typical Matrix Organization (Ford & Randolph, 1992)*

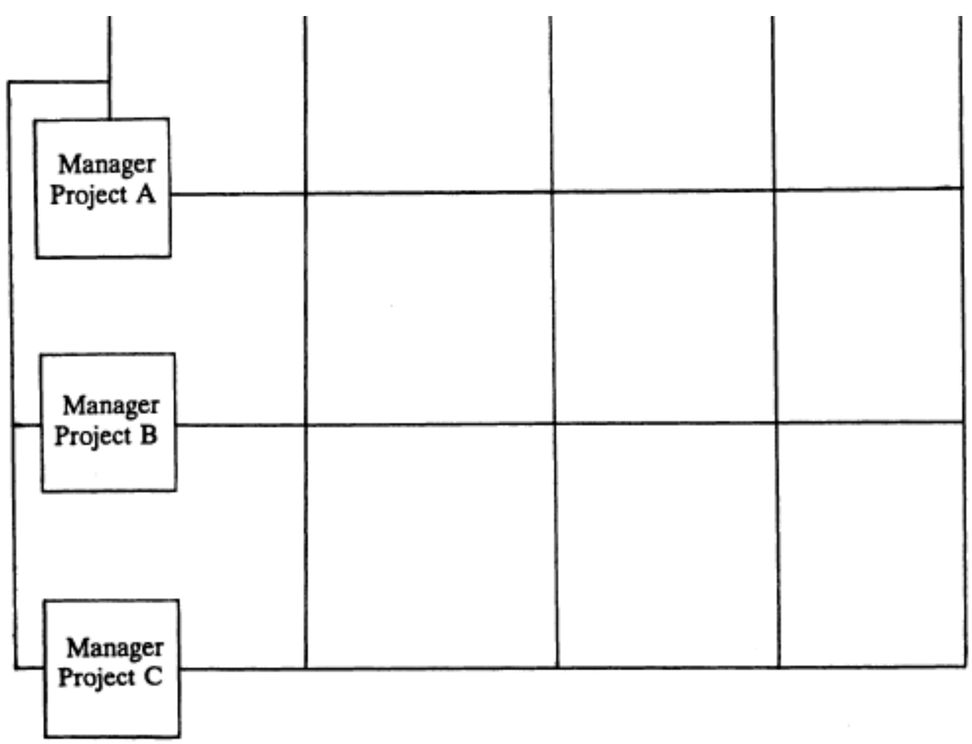
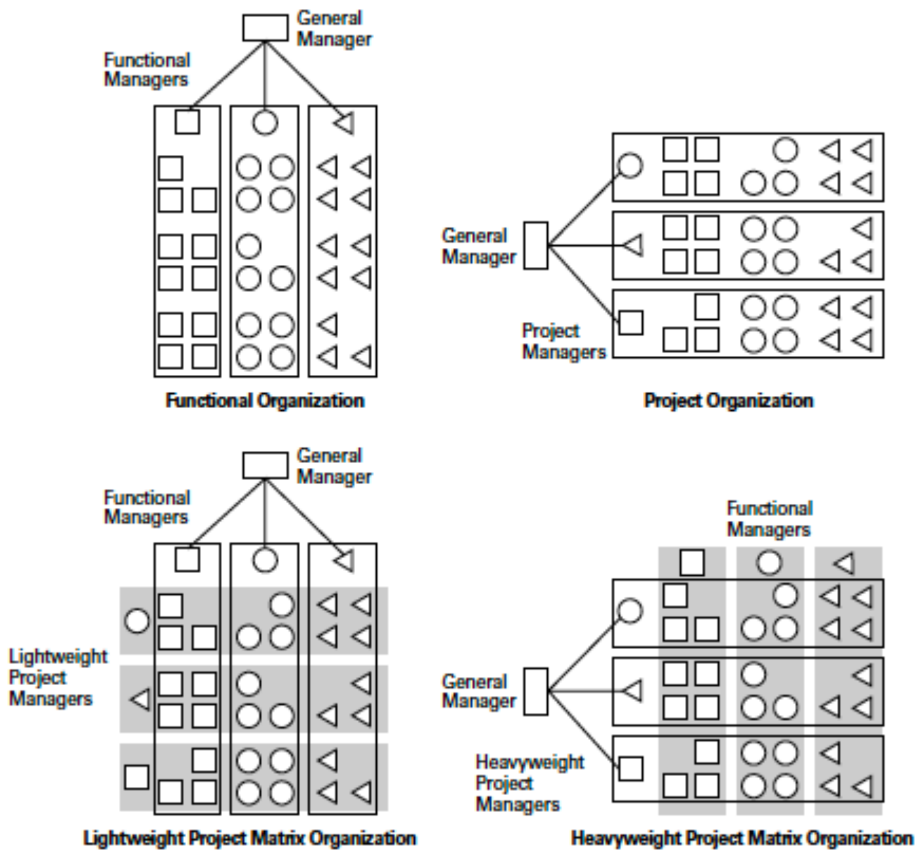


*Figure 4 - Typical Functional, Project and Matrix Organizations (Ulrich & Eppinger, 1995)*




Kerzner (2003) argues matrix organizations "attempt to create synergism through shared responsibility between project and functional management," but notes that achieving such synergy is often quite difficult in practice. Since no two working environments are the same, no two matrix organizations will be the same. (Kerzner, 2003) Advantages of matrix organizations include improved control over resources, independent policies and procedures for individual projects, quick adaptation to change, ability to develop a strong technical base, shared responsibility and authority, and improved ability to solve complex problems. Disadvantages include multidimensional work and information flow, dual reporting, changing priorities, potential for conflict, and role ambiguity. (Kerzner, 2003) Situations favoring matrix organizations include having a mix of products, plants and markets; short business cycles; complex and rapidly changing environments; and high technology products where scarce talent must be spread across multiple projects. (Wintermantel, 2003)

Miterev, Mancini and Turner (2017) identify options available for the design of project-based organizations and explore key factors affecting those options compared to traditional organizations. They define a project-based organization as one that decides to use project management businesses practices to manage work. They distinguish a program as being a collection of related projects, but note both projects and programs are temporary organizations. They argue an unpredictable and rapidly changing business environment drives firms to adopt "temporary organizational forms, such as projects and programs," noting the "management of innovation in the car industry now requires a project-led or project-supported organization." (Miterev, Mancini, & Turner, 2017)

Reflecting on holistic models of organization design, such as the McKinsey 7-S framework and Galbriath's star model, Miterev, Mancini and Turner argue organizational designers must consider a range of factors, including "internal coherence and external fit." Noting the tendency towards disaggregation in large firms, they further argue decentralization can improve performance when searching for solutions to non-decomposable problems. They propose the design of project-based organizations should consider five related elements: orientation, the strategic decision to be project-based; project organization, which defines the relationship between projects, programs and functions; business processes, which should be project-based; culture, which should be project-oriented; and project working practices that recognize and accommodate the churn created when projects are formed and disbanded. (Miterev, Mancini, & Turner, 2017)

Early investigations suggested matrix organizations should improve information processing by “formalizing lateral communication channels and legitimizing informal communication,” with a corresponding increase in formal communication and decrease in informal communication. Ford and Randolph argue matrix organizations should have greater information processing capacity and the ability to handle increased information loads compared to functional organizations because “increased contact among departments allows information to ‘permeate’ the organization, improving decision making and response time, which translates into an organization that can quickly and felxibly adapt to a dynamic situation.” (Ford & Randolph, 1992)

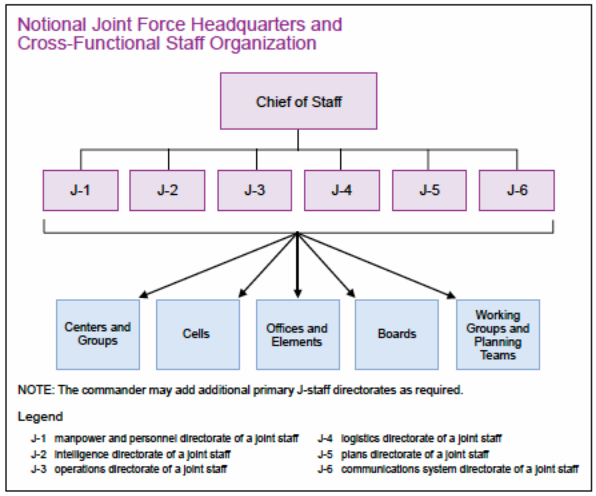
Schnetler, Steyn and van Staden (2015) investigated the effect of communication, collaboration, trust and other characteristics on success of projects and found increased communication in matrix organizations improved both the quality of communication and overall team performance. They argue better communication improves trust and collaboration, which in turn improve team performance and promote project success. They conclude “the matrix structure lends itself to an increase in the frequency of communication” and recommend managers “should facilitate and promote both the frequency and the quality of communication” through co-location of team members and opportunities for greater communication. (Schnetler, Steyn, & van Staden, 2015)

Matrix Organizations in Integrated Product Development. Beginning in the 1990s, large companies, especially those in the aerospace domain, started to move towards integrated product development (IPD), a philosophy that seeks to lower overhead costs, shorten development time and increase flexibility through cross-functional collaboration. IPD brings together representatives from relevant functions to capture collective input during the design phase, when changes can be made at relatively low cost. Integrated product teams (IPTs), composed of designers and representatives from other functions, design systems, subsystems, and components and own a product throughout its lifecycle. IPTs use a variety of integrative tools and mechanisms, including systems engineering, interface optimization, training, co-location, town hall meetings, manager or participant mediation, interface groups, and interface scorecards. The concept of design for integration provides a framework for achieving integration. Design for integration principles inclue knowing the system architecture, assigning IPTs to system elements, grouping IPTs, applying integrative mechanisms, managing interfaces and assessing status. (Browning, 1996)

Military Staffs: Boards, Centers, Cells and Working Groups. To outside observers, military organizations, with their well-defined chains of command and lines of authority and responsibility, may seem to be the embodiment of hierarchical organizational structure. However, successful execution of complex military operations requires close coordination, synchronization, and information sharing, and military staffs achieve this sort of cross-functional collaboration by forming boards, centers, bureaus, offices, working groups, cells, and other temporary and permanent teams to manage specific tasks or functions. These teams, sometimes shortened to “boards, centers, cells and working groups” (BCCWG), facilitate planning, decision-making and execution. (Wade, 2012) Figure 5 illustrates a typical U.S. Joint Task Force organization, illustrating the use of BCCWG for cross-functional collaboration. BCCWG teams are generally led by a senior individual from the cognizant directorate, but draw members from across the organization, depending on the role or function they perform. (Joint Chiefs of Staff, 2017)

The arrangement of BCCWG teams reflects a key principle of military staffs, and of military organizations more generally, in that individuals and organizations have both an administrative chain of command, responsible for a wide range of administrative and logistic functions, and an operational chain of command, responsible performing specific tasks and executing operations. Reflecting on the definition offered by Miterev, Mancini and Turner, it is clear military staffs are a kind of project-based organization. Military staffs have a clear functional structure built around directorates with specific, enduring responsibilities and capabilities, but they also have cross-functional organizations that exist to accomplish specific tasks or projects. Military operations are, by nature, temporary and thus project-like. The principal difference between a matrix organization and a military staff is that a matrix organization relies on a lateral, project-based authority separate from the functional structure, while a military staff embeds the project authority within the existing functional structure.

*Figure 5 - Typical Joint Force Organization (Joint Chiefs of Staff, 2017)*




Organizational Structure and Product Architecture

Conway's Law. Reserachers have long recognized the interplay between products or systems and the organizations that design them. In 1968, Melvin Conway articulated what has come to be known as Conway's Law:

> Organizations which design systems (in the broad sense used here) are constrained to produce designs which are copies of the communication structures of these organizations. …This fact has important implications for the management of system design. Primarily, we have found a criterion for the structuring of design organizations: a design effort should be organized according to the need for communication. (Conway, 1968)

Similar to Simon, Conway defined design as an intellectual activity that creates systems from varied parts. He viewed design in broad terms, including a range of activites, from the design of weapon systems to the creation of public policy. The output of design is the "structured body of information" needed to achieve the stated objective. (Conway, 1968)

Conway lays out the general stages of design, which include establishing boundaries, selecting a preliminary concpet, organizing the design activity, delegating tasks based on concept, coordinating tasks, and consolidating subsystem or component designs into a final, single design. He then examines the relationship between the structure of the design organization and the architecture of the system it designs. He argues that for any node (i.e., component, sub-system) in the system, one can identify a node or group of nodes in the design organization responsible for its design.

Similarly, any link in the system design defines an interface between two nodes, necessitating communication and coordination between the responsible organizational entities. Conway concludes a structure-preserving relationship exists between system architecture and organizational structure. He asserts many alternative designs can satisfy requirements, and argues "the choice of design organization influences the processes of selection of a systems design" from those alternatives. Since the organization is not completely flexible in terms of communication structure, it will "stamp out an image of itself in every design it produces." This phenomenon is more prominent in larger, less flexible organizations. (Conway, 1968)

Conway explores the management of design and questions why design efforts fail, or "disintegrate," as he calls it. He identifies two principal problems, the tendency to "overpopulate" the design effort and "fragmentation of the design organization communication structure." Overpopulation occurs when the perceived complexity of the design exceeds limits of comprehension, leading to subdivision and delegation of tasks. Pressure to maintain schedule incentivizes managers to bring additional resources to bear, leading to further subdivision and delegation. One fallacy contributing to overpopulation is the perceived linearity of resources, the idea that 100 designers working for one week are of equal value to two designers working for a year since both have approximately equal cost in terms of man-hours, and therefore dollars expended.

Conway notes these resource allocations result in radically different organizational structures, which necessarily leads to different designs because of the structure-preserving relationship between organizational structure and system design. Delegation and overpopulation lead to fragmentation of the communication structure. The number of possible communication paths in a design organization is approximately equal to the square of the number of people in the organization divided by two. For design organizations of even modest size, communication must be restricted to allow time for "work." Hierarchical organizations limit communication to defined links along lines of organization and command, but the need to communicate depends on system concept. As a result, Conway argues design organizations should be "lean and flexible," and further argues in favor of management philosophies that do not equate manpower with productivity. (Conway, 1968)

Architectural Innovation and the Failure of Established Firms. Henderson and Clark (1990) examine the nature of innovation and conclude that changes to product architecture, including some perceived as minor technological improvements, challenge traditional firms by destroying existing knowledge embedded in the firms' organizational and communication structures. They focus on product development and take as their unit of analysis products sold to end users that are designed, engineered and manufactured by a single development organization. They acknowledge the distinction between the product as a whole—the system—and its physically distinct components and argue that successful development requires knowledge of component design concepts and knowledge of product architecture, which defines how individual components are integrated into a coherent system. (Henderson & Clark, 1990)

Examining innovation, Henderson and Clark argue simple distinctions between radical and incremental innovation are incomplete and instead propose a two-dimensional model that examines the effect of innovation and technological change on components and the linkages between them. Incremental innovation corresponds to changes that improve or affect components without affecting architecture, while radical innovation corresponds to changes that affect both component design and architecture. Henderson and Clark identify two additional categories: modular innovation, where components adopt new technologies, such as the change from analog to digital, without changing basic architecture, and the category of interest, architectural changes, where the dominant change lies in the architecture or arrangement of components, linking them in new ways. They acknowledge differences among categories are a matter of degree.

Radical changes are readily recognized because they are "radical," while incremental changes tend to reinforce or enhance existing core competencies. Architectural changes, on the other hand, are subtle and therefore hard to recognize. Technical evolution is usually characterized by periods of experimentation followed by the acceptance or emergence of dominant desings that establish basic design decisions not reconsidered in each subsequent design. "Once a dominant design is established, the initial set of components is refined and elaborated, and the progress takes the shape of improvements in the components within the framework of a stable architecture." (Henderson & Clark, 1990)

During periods of innovation, firms require the ability to develop knowledge and synthesize designs, but once a dominant design is established, firms stop investing in learning about alternative configurations and instead invest in refinements. Architectural knowledge becomes embedded in the firms' organizational structure. Henderson and Clark use the idea of channels, filters and strategies to describe how architectural knowledge becomes embedded. *Channels* refer to formal and informal reporting and teaming structures and reflect knowledge about architecture since the organization tends to be arranged and connected in the same way as the product and its components. Organizations establish *filters* to determine what information is important, and tend to eliminate or ignore information irrelevant to the dominant design. Designers develop *strategies* to solve problems based on experience. Organizations use channels, filters and strategies to cope with complexity, and their operation becomes implicit within the organization. (Henderson & Clark, 1990)

Architectural changes present two problems: the need to recognize them, and the need to apply new knowledge effectively. Such changes put a premium on exploration and integration of new knowledge, and established firms often struggle to adapt. Henderson and Clark examine the challenge of architectural innovation through a study of the development of photolithographic equipment that collected data during a two-year field study that included interviews with product development teams and reviews of internal records. They conclude that architectural innovations challenge firms because they render useless existing knowledge contained in the organization's structure and are hard to recognize because the established organizational strucuture filters out critical indicators, delaying recognition. In addition, they argue the effect of architectural innovation depends on how organizations learn, and suggest the "fashion for cross-functional teams and open organizational environments" may be a response to perceptions on the challenges of architectural innovation. (Henderson & Clark, 1990)

Testing the Mirroring Hypothesis. McCormack, Rusnak, and Baldwin (2008) explore the relationship between product architecture and organizational structure and test the mirroring hypothesis, which predicts organizations with different structures will produce products with different architectures. Using examples from the software industry, they find solid evidence supporting the mirroring hypothesis and claim important managerial implications because product architecture is an important predictor of organizational performance. (McCormack, Rusnak, & Baldwin, 2008)

McCormack, Rusnak, and Baldwin use design structure matrices (DSM) to compare software products created by different organizational structures, using degree of modularity to characterize system designs. Broadly speaking, modular designs exhibit interdependence within modules and independence between modules. Designs with a high degree of modularity are often said to be "loosely coupled" in the sense that changes made in one module have little impact on others. They note product architecture is critical to successful development of new products, competitiveness, and the evolution of organizational capabilities. However, several architectures may satisfy a given set of functional requirements, and different designs will have different performance in terms of cost, quality, reliability and adaptability.

Recalling previous studies, McCormack, Rusnak and Baldwin describe the relationship between product architecture and organizational structure, noting the technical dependencies that drive the need for communication within the design organization result from managerial choices. They also note competing perspectives. The first asserts the need to align communication to the technical dependencies among system components derived from system functionality. The second, first articulated by Conway and illustrated by Henderson and Clark, asserts organizational structure is fixed in the short term, so organizational strcuture impacts the resulting design. These competing perspectives can be evaluated by comparing software products with like functional requirements created by organizations with different structures. (McCormack, Rusnak, & Baldwin, 2008)

McCormack, Rusnak and Baldwin find strong support for the mirroring hypothesis, noting loosely-coupled software design organizations produced products with higer degrees of modularity than those developed by tightly-coupled design organizations. They note surprisingly large differences in modularity for products of similar size and function, finding direct dependencies give rise to many more indirect dependencies in tightly-coupled organizations. They further find product architecture is influenced by both functional requirements *and* contextual factors, a result with important managerial implications given that the search for new designs is constrained by the nature of the organization in which the search occurs. They identify two potential causal mechanisms. One one hand, designs may "evolve to reflect their development environments," with differences in communication between tightly- and loosely-coupled organizations leading to differences in modularity. On the other, differences may result from purposeful choices. For example, loosely-coupled organizations may require highly modular designs to succeed. In practice, both mechanisms likely play a role. (McCormack, Rusnak, & Baldwin, 2008)

Managers must understand how decisions on organizational structure affect design choices in non-explicit ways related to the interplay between problem-solving methods and the scope of the design space that must be searched to find an acceptable solution. In addition, managers must recognize the cognitive problem stemming from the critical dependence of system architecture on indirect dependencies that are often difficult to see in simple "black box" representations. (McCormack, Rusnak, & Baldwin, 2008)

<u>Interplay Between Product Architecture and Organizational Structure</u>. Ulrich analyzes the relationship between product architecture and the management of product development. He argues that modular architectures require greater emphasis on system level design to ensure interfaces and associated standards, performance requirements, and acceptance criteria are well defined. Detail design for individual systems or components can proceed independently, with design activities assigned to specialized design teams that have structured but infrequent interaction. In contrast, integral architectures require greater emphasis on detail design. System level design establishes system-level performance requirements and divides the overall system into a few subsystems. Detailed component design relies on a core team of designers who interact constantly to manage interactions. (Ulrich K. , 1995)

Modular designs allow a more traditional, bureaucratic organization built around specialized groups with deep experience, but require teams with strong system engineering and planning skills. For well-understood technologies, modular design may dramatically reduce the difficulty of managing product development, and these benefits may outweigh any system performance penalties associated with a modular architecture. However, modular designs can create organizational barriers to innovation. In contrast, integral designs may offer improved performance, but require teams with strong coordination and integration skills. For this reason, integral designs often prove more difficult to manage. (Ulrich, 1995)

Sinha, James, and de Weck (2012) examine how innovations, which change product architecture, affect the product development organization, demonstrating a feedback effect. They assert improvements to product performance or functional features often increase the product's complexity. Recalling Conway's Law, they note that changes to product architecture necessitate changes to orgnaizational structure and work processes, but also note organizational changes often lag technical changes. Aligning organizational structure with product architecture should improve a product's technical performance and should also provide benefits to business objectives, such as reduced cycle times. (Sinha & de Weck, 2012)

To evaluate the impact of innovation on organizational structure, Sinha, James, and de Weck compared two jet engine designs using design structure and multi-domain matrix techniques and found the new design required a significant increase in both intra- and inter-team interactions. They observe new connections between functional groups not previously connected improved communication and problem discovery, and note the largest changes occurred in groups outside the traditional "core" disciplines, in groups playing supporting roles. The latter result suggests such support functions provide increasing benefits to overall system performance. (Sinha, James, & de Weck, 2012)

Robust Organizations

Poire and Sabel challenged the notion, implicit in theories of firms, that the accomplishment of complex tasks is somehow centralized and controlled from above, considering this a “convenient fiction.” Instead, they argue when firms embark on new projects, the people involved know little about how to accomplish it, so design, innovation and production must occur simultaneously, and in a decentralized manner. When the environment becomes more ambiguous and uncertain, learning and design must occur in parallel. (Poire & Sabel, 1984) When confronted by ambiguity, organizations compensate by exchanging information, thus the problem of coping with ambiguity becomes a problem of distributed communication, which involves the transmission of information in connected systems. However, organizations are intrinsically hierarchical and individual members of the organization are limited in the amount of work they can accomplish. Networks are costly in terms of time and energy, so a robust information processing network must balance production (i.e., work) and information redistribution. (Watts, 2003)

Dodds, Watts and Sabel examined the dynamics of information exchange in organizationl networks and introduced an organizational network model that incrementally adds links to a hierarchical backbone according to a stochastic rule. They identified a class of networks, which they call “multiscale networks,” that exhibits “ultrarobustness,” meaning they simultaneously reduce the likelihood that an individual node will fail because of congestion and the likelihood that the overall network will fail if congestion failures do occur at individual nodes. In addition, they found multiscale networks achieve "ultrarobustness" with the addition of relatively few links, which suggests "ultrarobust organizational networks can be generated in an efficient and scalable manner." (Dodds, Watts, & Sabel, 2003)

Economists have long studied organizational structure, emphasizing efficiency over robustness and focusing on multilevel hierarchies. Hierarchies offer advantages for exercising control, accumulating knowledge, performing decentralized or distributed processing, and making decisions, but these advantages assume the organization's tasks can be easily decomposed into smaller subtasks that can be accomplished independently. Modern organizations "face problems that are not only large and multifaceted but also ambiguous: objectives are specified approximately and typically change on the same time scale as production itself, often in light of knowledge gained through the very process of implementing a solution." Problem solving becomes a collective activity characterized by simultaneous design and collaboration among individuals, teams, and organizations. Under these conditions, the chief concern is not efficiency, achieved by minimizing costly organizational links, but robustness, achieved by preventing individual nodes from being overwhelmed and protecting the overall network from catastrophic collapse when individual failures do occur. (Dodds, Watts, & Sabel, 2003)

Dodds, Watts, and Sabel propose a model (DWS model) of organizational networks with four components: a construction algorithm, a description of the task environment, an algorithm for passing messages, and specific measures for congestion and connectivity robustness. They begin with hierarchical organizational structure defined by branching ratio, $B$, and number of levels, $L$, which yields a network with $N = (B^L - 1)/(B - 1)$ nodes. The construction algorithm then adds $m$ nodes according to a stochastic rule , that governs the probability, $P(i, j)$, that links will be added between nodes $i$ and $j$:

$$P(i,j) \propto e^{-\frac{D_{ij}}{\lambda}} e^{-\frac{x_{ij}}{\varsigma}}$$

The algorithm chooses additional links without replacement. The hierarchical backbone represents the organization's formal structure, while additional links represent teaming arrangements that transmit information. The stochastic rule uses two key parameters, depth, $D_{ij}$, of the lowest common node, $a_{ij}$, between nodes $i$ and $j$, and organizational distance between nodes $i$ and $j$, given by $x_{ij} = \sqrt{d_i^2 + d_j^2 - 2}$, which is valid for $d_i + d_j \geq 2$. The rule also uses two tuning parameters, λ and ς, which represent characteristic lengths for $D_{ij}$ and $x_{ij}$ respectively. Figure 6 identifies and illustrates elements of the stochastic rule.

Figure 7 illustrates four classes of organizational networks for limiting values of λ and ς:

- Random networks, R, $(\lambda, \zeta) \rightarrow (\infty,\infty)$, in which links are added uniformly at random, without regard to lowest common ancestor rank or organizational distance;
- Local Team, LT, $(\lambda, \zeta) \rightarrow (\infty,0)$, in which links are added only between node pairs who share the same immediate supervisor;
- Random Interdivisional, RID, $(\lambda, \zeta) \rightarrow (0,\infty)$, in which links are added between nodes whose lowest common ancestor is the node at the top of

the hiearchy, i.e., between nodes in different major divisions of the hierarchical organization; and

- Core-Periphery, CP, (λ, ζ) → (0, 0), in which links are added only between subordinates of the top node, resulting in a fully connected central core with pure branching hierarchies below.

Multiscale networks, MS, correspond to moderate values of λ and ς (i.e., λ = ς = 0.5) and combine features of the four other network classes. Multiscale network connectivity is not dominated by a single factor or scale. Instead, they show connectivity at mulitples scales at the same time, but do not show uniform density at all scales, which distinguishes them from small-world networks. These features improve information exchange compared to hierarchical networks, which tend to put the burden of information sharing on nodes at higher ranks.

*Figure 6 Schematic Illustration of the Construction Algorithm (Dodds, Watts, & Sabel, 2003)*

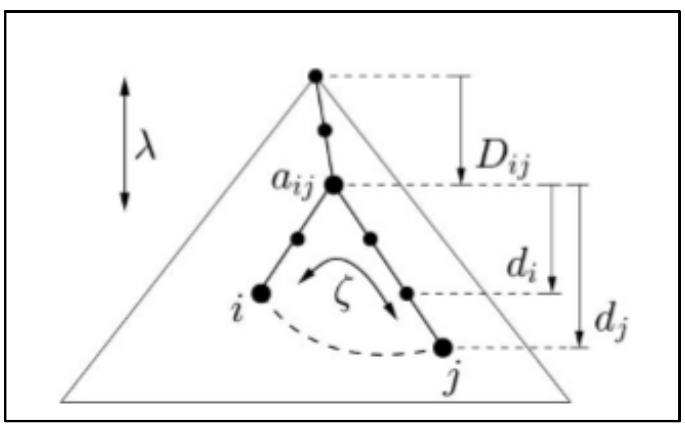
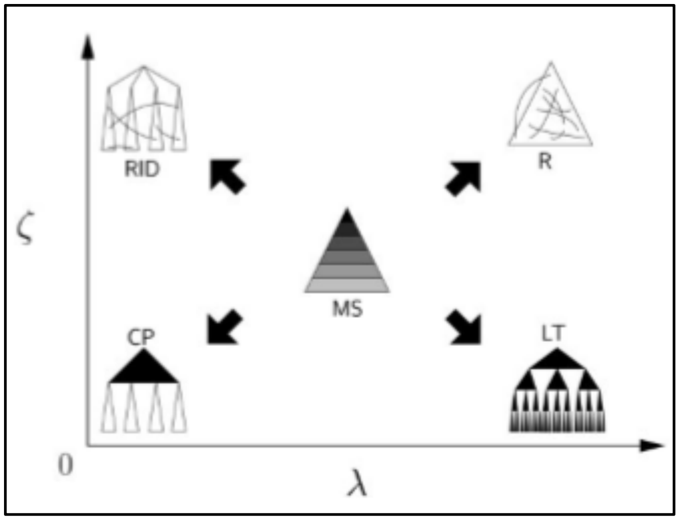


*Figure 7 Classes of Organizational Networks (Dodds, Watts, & Sabel, 2003)*




The DWS model represents the task environment based on the rate and distribution of messages exchanged in the process of completing a global task. Stable environments have low rates of information exchange, μ, defined as the average number of messages initiated by a node per time step. The task environment also allows different degrees of task decomposability. Tasks with a high degree of decomposibility only require message passing within the same group, that is nodes with the same immediate supervisor, while tasks that cannot be decomposed require communication with distant nodes. For a given source node, *s*, transmitting messages at rate μ, the task environment model selects a target node, *t*, at random by weighting all nodes at distance *x* using the factor $e^{-\frac{x}{\xi}}$. When $\xi = 0$, tasks display a high degree of decomposability and all messages are passed locally. When $\xi \rightarrow \infty$, tasks are not decomposable and the target is chosen at random. Messages are passed from source to target through intermediaries, with each node in the chain passing the message to an immediate neighbor who has the lowest common ancestor with the target. This method assumes each node has complete information on its own location and the locations of its neighbors, a condition called "pseudoglobal knowledge." (Dodds, Watts, & Sabel, 2003)

The DWS model uses two measures of network robustness, congestion centrality and connectivity robustness. Congestion centrality of an individual node is the probability that any message will be processed by that node. The rate of information processed by node *i* is therefore given by $r_i = \mu N \rho_i$. A node will remain free of failure only if its capacity, $R_i$, exceeds $r_i$. Dodds, Watts, and Sabel argue a robust organizational structure reduces congestion centrality, thus they associate congestion robustness with structures that reduce $\rho_{max}$. When individual node failures do occur, the network can continue to function if it remains connected. Dodds, Watts, and Sabel therefore adopt fractional size of the largest connected component as a measure of connectivity robustness: $C = \frac{S}{N - N_r}$, where *S* is the size of the largest connected component remaining after the removal of $N_r$ nodes. (Dodds, Watts, & Sabel, 2003)

Figure 8 presents congestion results for a hierarchy defined by branching ratio *B* = 5 and depth *L* = 6, and a task environment with moderate decomposability, $\xi = 1$. The upper contour plot demonstrates multiscale networks correspond to moderate values of $\lambda$ and $\zeta$, while the lower plot demonstrates that multiscale networks reduce maximum congestion centrality with fewer team links, *m*, than other networks. Core-periphery networks exhibit lower values of maximum congestion centrality, but also exhibit greater variability and sensitivity to initial conditions. Multiscale networks do not exhibit this volatility, making them a more reliable solution for improving congestion robustness.

Figure 9 illustrates the scaling of congestion centrality with network size and demonstrates that congestion centrality continues to decrease as network size increases for multiscale networks, while for other networks, congestion centrality decreases only to a plateau or limiting value. Figure 10 presents connectivity robustness results and shows that random and random interdivisional networks have the best connectivity robustness. However, multiscale networks have comparable connectivity robustness, and significantly better congestion robustness, making them the overall most robust choice. (Dodds, Watts, & Sabel, 2003)

Multiscale networks "display a remarkable combination of properties," including low likelihood of congestion failures over a range of environmental conditions, resilience to disconnection if node failures occur, and ultrarobustness, meaning simultaneious congestion and connectivity robustness not exhibited by any other network class. In addition, multiscale networks achieve these benefits when only a small number of additional team links are added to the hierarchical backbone. One should therefore expect to find networks resembling multiscale networks in real-world organizations. (Dodds, Watts, & Sabel, 2003)

*Figure 8 Congestion Centrality as a Function of Network Parameters and Number of Team Links (Dodds, Watts, & Sabel, 2003)*

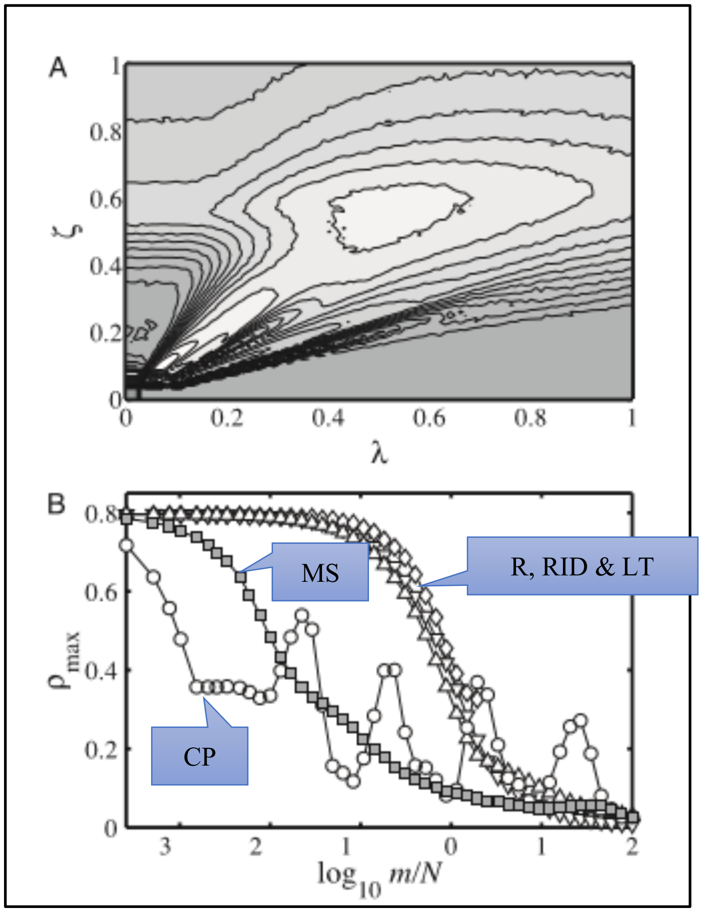



*Figure 9 Scaling of Congestion Centrality with Increasing Network Size (Dodds, Watts, & Sabel, 2003)*

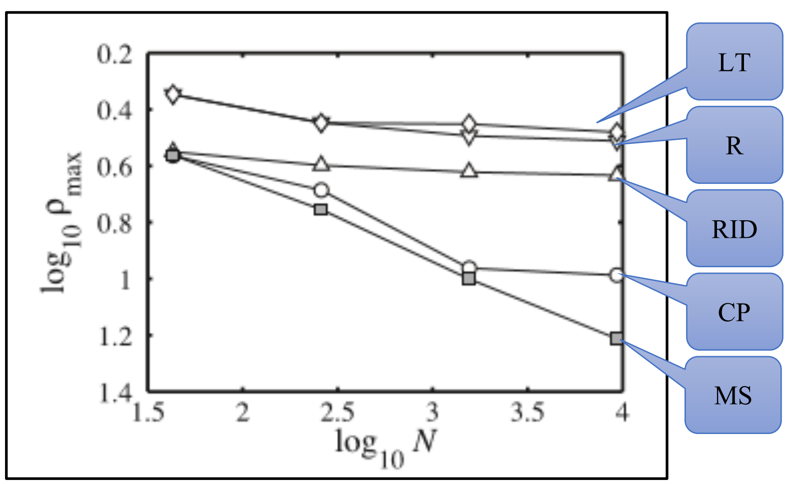
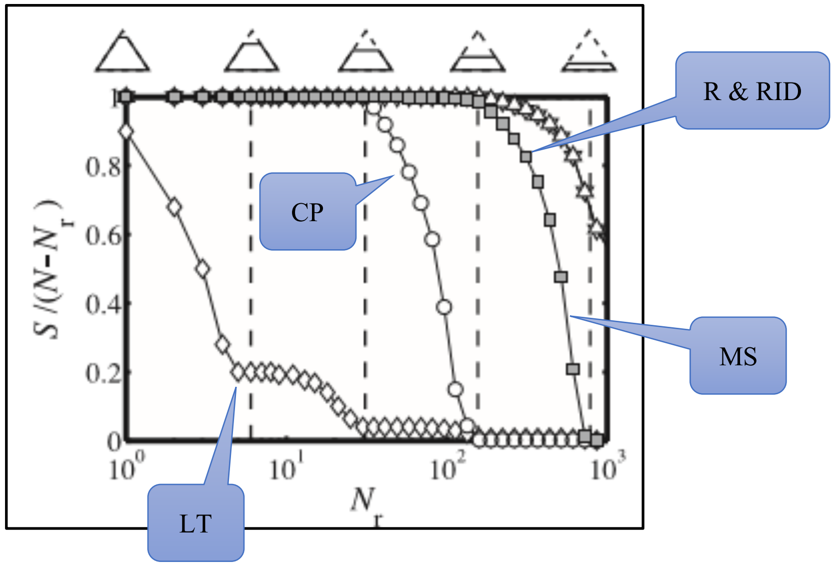


*Figure 10 Connectivity Robustness of Networks (Dodds, Watts, & Sabel, 2003)*




Magee (2010) assesses the DWS model and notes the model describes the relationship between the structure of organizational networks and their robustness properties, but does not describe the mechanisms by which such networks are formed. He argues two questions should be considered when evaluating the practical relevance of the DWS model: whether real organizations must address non-decomposable problems requiring collaboration at large organizational distances and how DWS organizational networks might be created in practice. Drawing on personal experience, Magee identifies several widely-used approaches to solving non-decompoable problems, including co-location, use of cohorts, rewarding cross-functional knowledge, and use of matrix management. He also suggests directions for future research on organizational networks, including an exploration of how well matrix organizational networks compare to multiscale networks. (Magee, 2010)

Understanding Complexity

A 2011 report prepared for the Defense Advanced Research Projects Agency (DARPA) concluded cost and schedule overruns in defense programs result from "systematic mismanagement of the inherent complexity associated with the design of these systems." The report notes complexity is hard to quantify, but argues complexity is related to the number of design parameters and the interactions among them, which are often poorly understood. (Murray, et al., 2011) Building on the 2011 DARPA report's conclusions, two of its contributors, Kaushik Sinha and Oliver de Weck 2013), argue "today's large-scale engineered systems are becoming increasingly complex" due to demands for increased performance and improved lifecycle properties. (Sinha & de Weck, 2013) They report 13 aerospace projects reviewed by the Government Accountability Office (GAO) showed cost growth of 55% or more, and attribute such cost overruns to "our current inability to characterize, quantify and manage complexity." (Sinha & de Weck, 2013) They assert complexity results from the number and variety of elements in a system and their connectivity, and further assert complexity is a "measureable system characteristic." (Sinha & de Weck, 2013)

The 2011 DARPA report is correct when it says the term complexity is "difficult to quantify and often abused." (Murray, et al., 2011) Melanie Mitchell, External Professor at the Santa Fe Institute, notes in her book *Complexity: A Guided Tour* that no single science or theory of complexity yet exists, despite the many books and articles written on the subject. She identifies common properties of complex systems, including complex collective behaviors, such as self-organization and emergence; signalling and information processing; and adaptation through learning or evolution.  Her definition of complex system incorporates these characteristics:  a complex system is one "in which large networks of components with no central control and simple rules of operation give rise to complex collective behavior, sophisticated information processing, and adaptation via learning or evolution."  She further describes self-organizing systems as those where organized behaviors arise without an internal or external controller or leader, and emergent behaviors as those that arise from simple rules in unpredictable ways. (Mitchell, 2009)  As the title suggests, her book provides a guided tour of the subjects and ideas central to complexity, including dynamics and chaos, information and computation, evolution, genetics, cellular automata, and networks.

Scott Page provides a useful framework for undestanding complexity, defining complex systems in terms of four necessary characteristics: diversity, connectedness, interdependence, and adapation.  Diversity refers to the number and variety of different agents or elements in the system.  These agents are connected and interdependent, that is the actions and behaviors of individual agents affect and are affected by those of other agents.  Finally, complex systems change over time due to adaptation and selection.  Page argues adaptation is the key characteristic separating complex systems from complicated ones.  As an example, he says a watch is complicated because it has diverse, connected and interdependent parts, but it is not complex because those parts do not adapt.  The watch operates in a fixed and predictable manner and does not exhibit the kinds of behaviors associated with complex systems.  (Page, 2009)

Mitchell likewise notes that some definitions of complexity omit adaptation, with the term *complex adaptive system* being used to distinguish systems in which adaption plays an important role. (Mitchell, 2009) In fact, much of the confusion about the meaning of complexity stems from two related questions: whether to include adaptation in definitions of complexity and how to differentiate complex systems from those that are merely complicated. In common use, when someone says a thing is "complex," they most often mean it is hard, challenging, or complicated, but as we have already seen, the term complex is also used to describe a variety of unexpected or "complex" system behaviors. Peter Senge (1990) addresses this disparity in *The Fifth Discipline*, where he distinguishes detail complexity, the usual kind characterized by many variables, from dynamic complexity, in which cause and effect are subtle, and the effects of interventions over time are not obvious. (Senge, 1990)

Using Senge's categories, detail complexity would equate to complicated systems, while dynamic complexity would equate to complex systems or complex adaptive systmes, those specifically characterized by adaptation. The definition offered by Sinha and de Weck focused on the number of elements and their dependencies and therefore represents a form of detail complexity. Later in their article, they adopt the term "structural complexity" to emphasize they are principally interested in non-adaptive characteristics. This paper will use *structural complexity* to mean the detail complexity associated with the architecture of a system, characterized by the number and variety of elements in the system and their connections and interdependencies, and *complex adaptive system* to mean systems additionally characterized by adaptation and selection when the difference is important.

While this distinction seems clear, confusion can also occur when trying to separate the characteristics of complex systems from their behaviors. Mitchell identified self-organization and emergence as behaviors that distinguish complex adaptive systems. (Mitchell, 2009) To this list, Page adds several additional items, including robustness, susceptibility to large events, and non-linear dynamics. (Page, 2009) He argues complex adaptive systems are robust, meaning they can withstand disturbances. Returning to his watch analogy, he notes that a watch, while complicated, will cease to function if elements are removed. In contrast, a complex adaptive system will continue to function because it is adaptive. Paradoxically, complex adaptive systems often produce the kinds of "large events" to which they are robust. Nassim Taleb famously called such events "Black Swans." He defines a Black Swan as an event with three charactersitics: "rarity, extreme impact, and retrospective predictability." The third characteristic refers to the human tendency to identify, *post facto,* explanations that would have made the event predictable. (Taleb, 2010) Complex adaptive systems also exhibit non-linear dynamics such as phase transitions, the sudden change from one condition to another sometimes called a tipping point. Among the behaviors of complex adaptive systems, emergence is perhaps the most important. Emergence, or emergent behavior, refers to the situation where macro behavior differs from, and cannot be easily predicted by, the micro behaviors of agents in the system. One common type of emergence is self-organization, which happens when macro patterns or structures arise from the bottom up without centralized control. Classic examples include schooling of fish and crystalline structures of materials. (Page, 2009)

Thomas Schelling's transformational work, *Micromotives and Macrobehavior*, explores, as the title suggests, how individual choices affect overall behavior in non-obvious ways: "though people may care how it all comes out in the aggregate, their *own* decisions and their *own* behavior are typically motivated toward their *own* interests, and often impinged on by only a local fragment of the overall pattern." (Schelling, 2006) Chapter 4 famously demonstrates how a slight and non-malicious preference towards having neighbors of the same race ultimately leads to segregated neighborhoods. Despite a relatively high degree of tolerance at the micro level, the overall result is segregation. Schelling observes: "it is not easy to tell from the aggregate phenomenon just what the motives are behind the individual decisions or how strong they are." (Schelling, 2006) This kind of micro-macro disconnect is central to the idea of emergence in complex adaptive systems.

Complexity in Projects

A similar disconnect can occur in "merely complicated" systems when connections and dependencies are poorly understood. Sinha and de Weck note "a perpetually occurring theme" affecting the design of large engineered systems is the idea that designers create more complex product architectures when they "stretch the limits of efficiency and attempt to design more robust systems." (Sinha & de Weck, 2013) In large engineered systems, such as automobiles, aircraft and ships, the number of connections and dependencies can quickly challenge the limits of human cognition. Even though individual elements of the system may perform in predictable ways, interactions among elements can lead to unexpected macro behaviors. Such behavior may be predictable in theory, but not in any meaningful or practical way. As a result, merely complicated, large engineered systems often exhibit *quasi-emergent* behavior comparable to complex adaptive systems.

Patanakul, *et al* (2016), analyzed 39 public projects undertaken in the United States, United Kingdom and Australia and identified six key characteristics affecting project performance. Among these, they identified project complexity as a root cause of poor performance, noting a positive correlation between project size and complexity. They argue project complexity results from both tehnical challenges and from an array of “ambiguous and uncertain external and internal forces.” They identify improper governance structures and poor project management approaches as key factors leading to poor project performance. (Patanakul, Kwak, Zeikael, & Liu, 2016) Floricel, Michela and Piperca investigated how complexity affects project performance and provide a theoretical basis for understanding the relationship between complexity and project performance. They propose a framework that characterizes project complexity using structural-dynamic and intrinsic-representational dichotomies as illustrated at Table 1. The structural-dynamic dichotomy corresponds to previous definitions, with structural complexity referring emergent behaviors that result from poorly understood interactions among system entities and dynamic complexity referring to temporal behaviors that produce sudden changes that can be radical and unpredictable. The intrinsic-represnetational dichotomy refers to differening perspectives around whether complexity is an intrinsic characteristic of reality or results from our inability to recognize and represent it. The intrinsic-representational distinction implies “planners see complexity aspects as intrinisc in the ‘world out there’ or as resulting from imperfections in their own representations." Applying these distinctions results in the indicators of project complexity shown in the four quadrants of Table 1 (Floricel, Michela, & Piperca, 2016)

*Table 1 - Types of Project Complexity and Their Indicators and Effects (Floricel, Michela, & Piperca, 2016)*

| | Structural | Dynamic |
|---|---|---|
| Intrinsic | Non-additive aggregation or interactions<br>Multi-level frameworks<br><br>Effect: unpredictable form | Number and interdependence of variable<br>Evolutionary and dialectic frameworks<br><br>Effect: path-dependent dynamics |
| Representational | Abstraction and computational difficulties<br>Systematic trial and error<br><br>Effect: unintended properties | Hidden interdependencies<br>Contingency planning and early tests<br><br>Effect: repeated and significant surprises |

Floricel, Michela and Piperca (2016) identify four strategies that planners use to cope with complexity: use of existing knowledge, creation of new knowledge, separated organization, and integrated organization. The first two categories represent a choice between using existing knowledge as captured in databses, models and rules and creating new knowledge through experimentation, simulation and prototyping. The second two categories represent a choice between "decomposing relevant objects and tasks into stand-alone blocks and allocating the execution to distinct organizations or teams" and increasing "the density and strength of communication tieks throughout a project organization by stimulating collaborative work." (Floricel, Michela, & Piperca, 2016)

Floricel, Michela and Piperca analyzed data from 81 projects from across a range of sectors and found project complexity negatively affects completion performance as expected. Specifically, they found technical complexity negatively effects project performance, but also found mixed results for other performance aspects, including innovation and value creation. They argue for a “more careful consideration of complexity effects,” noting “perceptions of high complexity may generate more intense representation efforts, followed by implementation of special strategies.” (Floricel, Michela, & Piperca, 2016)

## Measuring Complexity

Several authors have proposed methods or measures to quantify complexity, but there is no single, widely accepted metric, nor even universal agreement that complexity can be measured. Melanie Mitchell surveys different approaches, taking as her point of departure a 2001 paper in which physicist Seth Lloyd proposed three features affecting the complexity of an object or process: the difficulty describing it, the difficulty creating it, and its degree of organization. Lloyd identified forty-odd measures of complexity from dynamical systems theory, thermodynamics, information theory, and computation. (Lloyd, 2001)

Counting Methods. Size is the simplest, and perhaps most commonly used, measure of complexity. For engineered systems, counting methods, that describe characteristics like the number of components in the system, provide insight, but size is generally not a good measure of complexity. For example, the human genome has 250 times more DNA base pairs than the yeast genome, but single-celled amoeba have

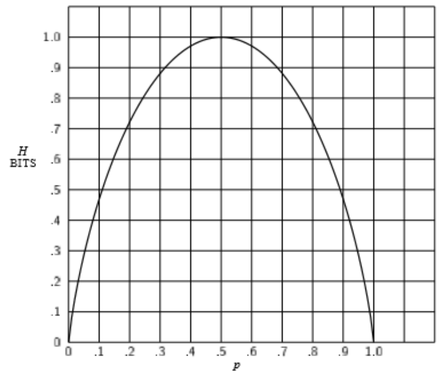
250 times more base pairs than humans. Clearly, counting the number of DNA base pairs would tell you little about why humans are more complex than amoeba.

Shannon Entropy. A second commonly proposed measure of complexity is Shannon Entropy, defined as the average information content in a series of messages between source and receiver. Shannon(1948) proposed entropy as a measure to quantify how much information is produced, or at what rate, by an information source. For a discrete, noiseless channel, the Shannon Entropy, *H*, is given by:

$$H = -K \sum_{i=1}^{n} p_i \log_2 p_i$$

Where *K* is a constant to account for units of measure and the base 2 logarithm is used to quantify information in binary digits, or *bits*. Shannon concluded measures of this form "play a central role in information theory as measures of information, choice and uncertainty." (Shannon, 1948) The form of *H* recalls formulations from statistical mechanics, and is identical to the form proposed by Boltzman. Shannon entropy has several interesting properties. It tends to zero when the probability of a particular outcome approaches unity, and has maximum value when all possible conditions have equal value equal to 1/n. Figure 11 presents a plot of Shannon Entropy versus probability, *p*, for the case of two probabilities, *p* and (1-*p*) and demonstrates that Shannon Entropy takes on a maximum value when either condition is equally likely. (Shannon, 1948)

*Figure 11 Shannon Entropy (bits) in the case of two possibilities with probabilities p and (1-p) (Shannon, 1948)*




Shannon Entropy seems appealing as a measure of complexity because the behavior illustrated in Figure 11 appears consistent with the intuitive sense that maximum complexity should occur somewhere in the transition between order and disorder. However, Mitchell notes it also has drawbacks that challenge its use as a measure of complexity. First, it is not always possible to describe a system as a series of messages. For example, it is not clear how one might use Shannon Entropy to measure the complexity of the human brain. Second, maximum entropy corresponds to a random system, where all conditions are equally likely. Mitchell concludes Shannon Entropy fails to capture the intuitive concept of complexity because the most complex systems are neither the most ordered nor fully random, falling instead somewhere between. (Mitchell, 2009)

Wilhelm and Hollunder (2007) propose a similar information theoretic approach for classifying networks based on the metric medium articulation (MA), which is a measure of network complexity. Medium articulation is given by:

$$MA = I(A, B)\, R(A, B)$$

Where *R(A, B)* is redundancy, given by:

$$R(A,B) = \sum_i \sum_j T_{ij} \log \frac{T_{ij}^2}{\sum_k T_{ik} \sum_l T_{lj}}$$

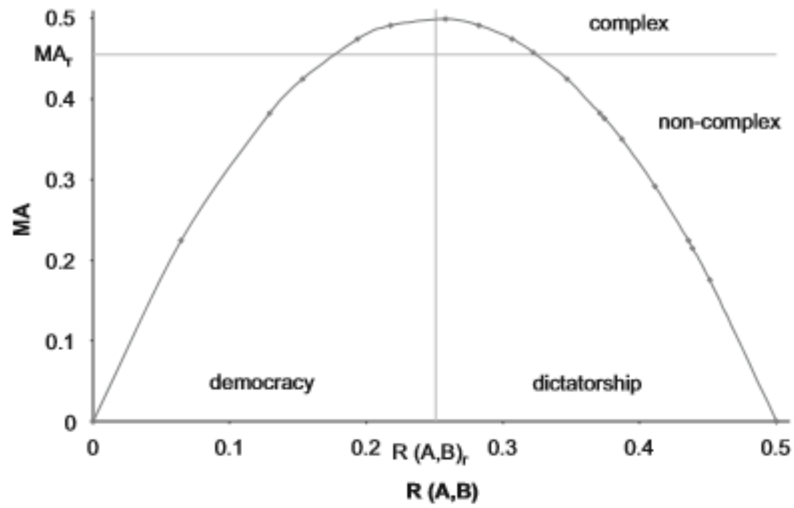
And *I(A, B)* is mutual information, given by:

$$I(A,B) = \sum_i \sum_j T_{ij} \log \frac{T_{ij}}{\sum_k T_{ik} \sum_l T_{lj}}$$

And $T_{ij}$ is the normalized flow from node *I* to J. (Wilhelm & Hollunder, 2007)

They demonstrate that networks with a medium number of links, $L \sim n^{1.5}$, show maximum MA of $\frac{(\log n)^2}{2}$, where *n* is the number of nodes in the network. They consider a network complex if its MA is larger than the MA of a randomized network and also differentiate democracy networks, in which information cycles, from dictatorship networks, in which information flows from sources to sinks. Figure 12 plots MA versus *R(A, B)* for all undirected and unweighted networks of $n = 6$ nodes and $L = 6$ links, and illustrates the classification scheme. All networks above the $MA_r$ line are considered complex, while those below are considered non-complex. Networks left of the vertical $R(A, B)_r$ line are democracy networks while those to the right are dictatorship networks. The proposed classification scheme applies to directed and undirected netowrks, as well as weighted and unweighted networks. (Wilhelm & Hollunder, 2007)

*Figure 12 Plot of Medium Articulation versus Redundancy for all networks with n = 6 nodes and L = 6 links (Wilhelm & Hollunder, 2007)*




Wilhelm and Hollunder establish a clear criteria for classifying a network as complex, but the distinction appears abitrary. Interestingly, they investigate food webs and neural networks, two classic examples of complex adaptive systems that exhibit emergent behavior, yet classify them as non-complex, illustrating the challenge of differentiating complex structure from complex behavior. (Wilhelm & Hollunder, 2007)

<u>Algorithmic Information Content, Logical and Thermodynamic Depth</u>. As an alternative to simple entropy, Kolmogorov, Chaitin and Solomonoff independently proposed algorithmic information content, the size of the shortest computer program that could generate a complete description of the system, as a measure of complexity. For example, a repeating sting of characters, such as “ACACACAC…” could be generated more simply than a ramdom string, such as “ATCTGCAAC…” The first string is said to be compressible, but the second is not and therefore contains more

information content.  Similar to simple entropy, algorithmic information content allots higher content to random systems than those one would intuitively consider complex.

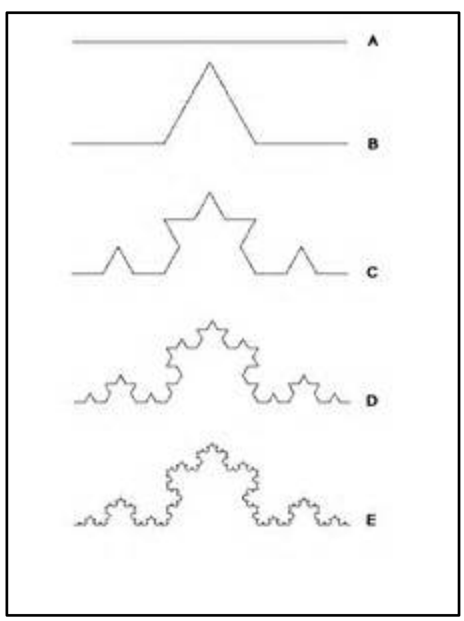
Physicist Murray Gell-Mann proposed a similar measure, "effective complexity," that characterizes a system in terms of regularities and randomness.  For example, the first string above has simple regularity, but the second, random string has none.  To calculate effective complexity, one must find the best description of the regularities; effective complexity is then the information content of the regularities.

A related pair of complexity measures, logical depth and thermodynamic depth, relate complexity of a system to the difficulty of creating it.  Such methods equate complexity with either the amount of information processed, or the thermodynamic or information resources required to create it.  (Mitchell, 2009) Measures of this sort hold intuitive and theoretical appeal, but they tend to be arbitrary in the sense that they depend on subjective descriptions of a system.  In addition, they are more a process to characterize a system than a measure in the truly quantitative sense.

Statistical Complexity.  James Crutchfield and Karl Yound defined statistical complexity as "the minimum amount of information about the past behavior of a system that is needed to optimally predict the *statistical behavior* of the system in the future."  Like Shannon Entropy, statistical complexity quantifies system behavior in terms of discrete messages.  To predict future behavior, a model of the system is created such that the behavior of the model is statistically indistinguishable from the system's behavior.  Statistical complexity matches intuitive expectations in that it is low for ordered and random systems and high for those in between.  However, like other measures already discussed, it is difficult to apply if the system cannot be easily represented as a message source. Still, investigators have successfully measured statistical complexity of complicated crystals and other phenomena. (Mitchell, 2009)

Fractal Dimension. Unlike previous measures that rely on concepts from information or computation theory, fractal dimension relies on concepts from dynamical systems theory. French mathematician Benoit Mandelbrot coined the term fractal to describe real-world objects, such as coastlines, trees, and snowflakes, with self-similar structues. In general, a fractal is a geometric shape that has the same structure at every scale of observation. For example, coastlines have similar, rugged structure at all scales of observation. Mathematicians have proposed numerous fractal models. For example, the Koch curve is created by application of the following rule: starting with a straight line, at each step, replace the middle third of the line with two sides of a triangle. Figure 13 illustrates the result.

*Figure 13 Koch Curve. At each step, replace the middle third of each line segment with two sides of a triangle.*



Fractals challenge traditional notions of spatial dimension. For example, if you repeatedly bisect a line *n* times, you get $2^n$ smaller copies after *n* steps. In general, when you repeatedly divide an object into *x* new objects, each level is made up $x^{dimension}$ copies of the previous level. For the example of bisecting a line, the dimension is 1, but for the Koch curve, the dimension is 1.26.[1] To summarize, "fractal dimension quantifies the number of copies of a self-similar object at each level of magnification of that object. Equivalently, fractal dimension quantifies how the total size (or area, or volume) of an object will change as the magnification level changes." Fractal dimension finds appeal as a measure of complexity because it captures the idea that complex systems have interesting details at all levels of observation, and it provides a way to quantify how interesting that detail is. However, level of detail is only one interesting aspect of complex systems, so fractal dimension is only a partial measure of complexity. (Mitchell, 2009)

Degree of Hierarchy. Herbert Simon argued hierarchy is one of the "central structural schemes" of complex systems, noting "the frequency with which complexity takes the form of hierarchy—the complex system being composed of subsystems that in turn have their own subsystems, and so on." (Simon, 1996) Simon identified a number of social, biological, physical and symbolic systems with hierarchic structures. For example, biological systems are often described using cells as the fundamental building block, with cells organized into tissues, tissues into organs, organs into

[1] The Koch curve divides a line into 3 segments at each step, thus $x = 3$, but each step creates 4 new objects, thus $3^{dimension} = 4$ and *dimension* = 1.26.

systems, and so forth. The cell is likewise composed of structured subsystems, such as the nucleus, cell membrane, and mitochondria.

Simon examines the dynamics of hierarchical systems and identifies a key property of hierarchic systems: near decomposability, defining *nearly decomposable* systems as ones "in which the interactions among the subsystems are weak but not negligible." (Simon, 1996) He offers two key propositions regarding nearly decomposable systems. First, the short-run behavior of individual subsystems is approximately independent of the behavior of other subsystems. Second, the long-run behavior of an individual subsystem depends only on the aggregate behavior of the other subsystems. Because hierarchies exhibit this property of near-decomposability, one can separate high-frequency dynamics related to the internal structure of subsystems from the low-frequency dynamics related to interactions among subsystems.

Hierarchic representations provide information about the relationships among the major elements of a systems, as well as information about the relationships among the parts that make up each element. Information about relationships between parts in different elements is lost, but this loss of information is not signficant because elements interact in an aggregate manner. Hierarchic representations also enable our ability to recognize, describe, and comprehend complex systems. (Simon, 1996) Mitchell notes several authors have explored the use of hierarchy to measure complexity. For example, Daniel McShea proposed to measure the complexity of biological organisms using a hierarchic measure based on *nestedness*, the idea that one entity contains as its parts entities at the next lower level. He showed organisms become more hierarchic as they evolve, but noted the challenge of objectively determining what constitutes a part or level. (Mitchell, 2009)

Analysis and Critique. Mitchell notes the large number of complexity measures that have been proposed and concludes "each of these measures captures something about our notion of complexity but all have theoretical and practical limitations, and have so far rarely been useful for characterizing any real-world systems." Like the idea of complexity itself, the variety of measures suggests complexity has many different dimensions not readily captured by a single metric. (Mitchell, 2009)

Feldman and Crutchfield (1998) reached a similar conclusion in their review of several measures of statistical complexity. They note many functions satisfy the intuitive criteria for measures of complexity, that they vanish at the extremes of order and disorder, and conclude this property is not sufficient. They then suggest two criteria for measures of complexity. First, the measure must have clear interpretation, that is it must specify what precisely is being measured. Second, it must consider motivation and define how it will be used and what questions it will answer. Many individual measures of complexity meet these criteria, but no single measure fully captures the nature or behavior of complex systems. (Feldman & Crutchfield, 1998)

Vincent Vesterby (2007) offers a stronger critique of efforts to measure complexity, arguing no current method is up to the task because the nature and magnitude of complexity render quantitative and qualitative methods inadequate. He further argues methods that simplify will fail because they "ignore what complexity is." Since current measures are inadequate, he recommends an approach that focuses effort on understanding complex systems rather than measuring them, suggesting that knowledge about how a system operates simplifies the task of measurement, making it more practical and specific to attributes like prediction and management. (Vesterby, 2007)

A Practical Measure of Structural Complexity. A common theme in the critiques of complexity measures is the insufficiency of individual measures, that is the idea that no single measure fully captures the nature or behavior of complex systems. However, individual measures provide useful insight, so it is important to follow the advice of Feldman and Crutchfield and focus on what is being measured and why. From an engineering perspective, useful measures would correlate complex characteristics or behaviors to properties of interest, such as cost. Sinha and de Weck proposed a practical measure for quantifying the structural complexity of engineered systems and demonstrated that cost varies non-linearly with complexity. In addition, they discuss the distribution of structural complexity across system architecture, as well as implications for system development efforts. (Sinha & de Weck, 2013)

The measure of structural complexity proposed by Sinha and de Weck takes the functional form: $C = C_1 + C_2C_3$, where $C_1$ represents the complexity of individual components, $C_2$ represents the number and complexity of pair-wise interactions among components, and $C_3$ represents the effect of architecture or the arrangement of interfaces. The terms $C_2$ and $C_3$ are mutually independent in the sense that the same number of interfaces can be arranged in a number of patterns, and the number of interfaces does not dictate arrangement. In addition, $C_3$ represents the global effect of architecture, which is often realized at the time of system integration. Sinha and de

Weck note their proposed measure of structural complexity has the same functional form as measures used in quantum mechanical analysis of molecular systems where the system's Hamiltonian (total energy) is the matrix of interest. Using this analogy, they propose topological complexity is captured by the graph or matrix energy of the adjacency matrix *A*, representing the system architecture.

The adjacency matrix of a network, here the network defined by system architecture, is the n x n matrix, *A*, where $A_{ij} = 1$ when nodes *i* and *j* are connected and 0 otherwise. The associated matrix energy of the network is defined by the sum of the singular values of the adjacency matrix, obtained from singular value decomposition:

$$E(A) = \sum_{i=1}^{n} \sigma_i$$

where $\sigma_i$ represents the i^th^ singular value. Sinha and de Weck note the matrix energy represents the "intricateness" of the structural dependencies among system components. They also note topological complexity increases as architecture moves from centralized to distributed structures. Distributed architectures cannot be reduced easily, but may offer improved performance and robustness. The full form of the proposed measure of structural complexity is given by:

$$C = \sum_{i=1}^{n} \alpha_i + \left[ \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} A_{ij} \right] \gamma E(A)$$

where $\alpha_i$ estimates the complexity of individual components, $\beta_{ij}$ estimates the complexity of each component-to-component interface, and $\gamma \sim 1/n$ is a scaling factor for graph energy. Sinha and de Weck suggest Technology Readiness Level (TRL) or similar measures could be used to estimate individual component complexities, and postulate interfaces with multiple connection types (such as load transfer, material flow, or control action) should have larger interface complexities.

Sinha and de Weck used their measure of structural complexity to compare two jet engine architectures, an older, dual-spool direct drive turbofan architecture and a newer, geared turbofan architecture. Experts assigned component complexities, and equal interface complexities were assumed. They found a 40% increase in complexity and noted that simple counting of components or interfaces grossly underestimated the increase in complexity, reinforcing the importance of architecture and toplogical complexity on overall structural complexity. Sensitivity analysis revealed thrust-generating components contributed significantly to component complexity, while supporting systems, such as lubrication and engine control, were principal contributors to topological complexity, with corresponding impacts to system integration efforts. They conclude simple components can have a greater effect than complex components due to their impact on overall system architecture.

Sinha and de Weck note the need for empirical validation of their proposed measure of structural complexity and also note the lack of direct measures of complexity. As a result, validation must rely on indirect measures or observables, such as development cost. They hypothesize that development cost should increase super-linearly with structural complexity and test their hypothesis using literature data for simple and complex systems. They demonstrate that development cost follows a power-law relationship, $Y = aX^b$ but caution their findings are based on limited data. They also conducted simple experiments in which human subjects were asked to build ball and stick models of molecules and found assembly time, a surrogate for cost, followed a power law relationship.

Sinha and de Weck also explored the factors affecting the distribution of structural complexity and found modular architectures do not necessarily reduce structural complexity, contrary to conventional wisdom. In fact, structural complexity can increase even as modularity increases. They conclude "knowledge of overall system architecture is absolutely critical to be able to quantify and track the complexity during the system development activity." Taking development of the Boeing 787 Dreamliner as a case study, they note Boeing outsourced much of the deveopment work and lost control of the development process. As a result, Boeing failed to understand total structural complexity as the system evolved. In order to successfully manage the development of large engineered systems, design teams must track evolving architectures to ensure subsystem complexities remain within sustainable limits. (Sinha & de Weck, 2013)

The measure of structural complexity proposed by Sinha and de Weck is useful because it provides a logical framework for understanding structural complexity and the role of individual components, interfaces, and system architecture. In addition, they demonstrate, based on preliminary data, that development cost should follow a power law relationship with structural complexity, and that increasing modularity may not decrease structural complexity. However, the lack of objective ways to estimate component and interface complexity, and the reliance on expert assessments for practical applications, may limit the measure's utility.

Improving Project Performance

Zhu and Mostafavi (2017) propose a framework for understanding complexity and managing emergence in projects. Drawing on contingency theory, they argue "the efficiency of a project is contingent on congruence between the project system's capability to cope with complexity (i.e., project characteristics) and the level of complexity." They characterize complexity using the framework proposed by Senge, that is in terms of detail and dynamic complexity, and identify three capacities that improve the project system's ability to cope with complexity: absorptive capacity, which relates to the ability to mitigate the effects of disruptions in advance; adaptive capacity, which relates to the ability to react to disruptions; and restorative capacity, which relates to the ability to recover from disruptions. (Zhi & Mostafavi, 2017)

Reinersten (2009) claims "the dominant paradigm for managing product development is fundamentally wrong" and recommends a new paradigm that aims to achieve flow in the product development process similar to that achieved in lean manufacturing. He identifies twelve problems with the current "product development orthodoxy:"

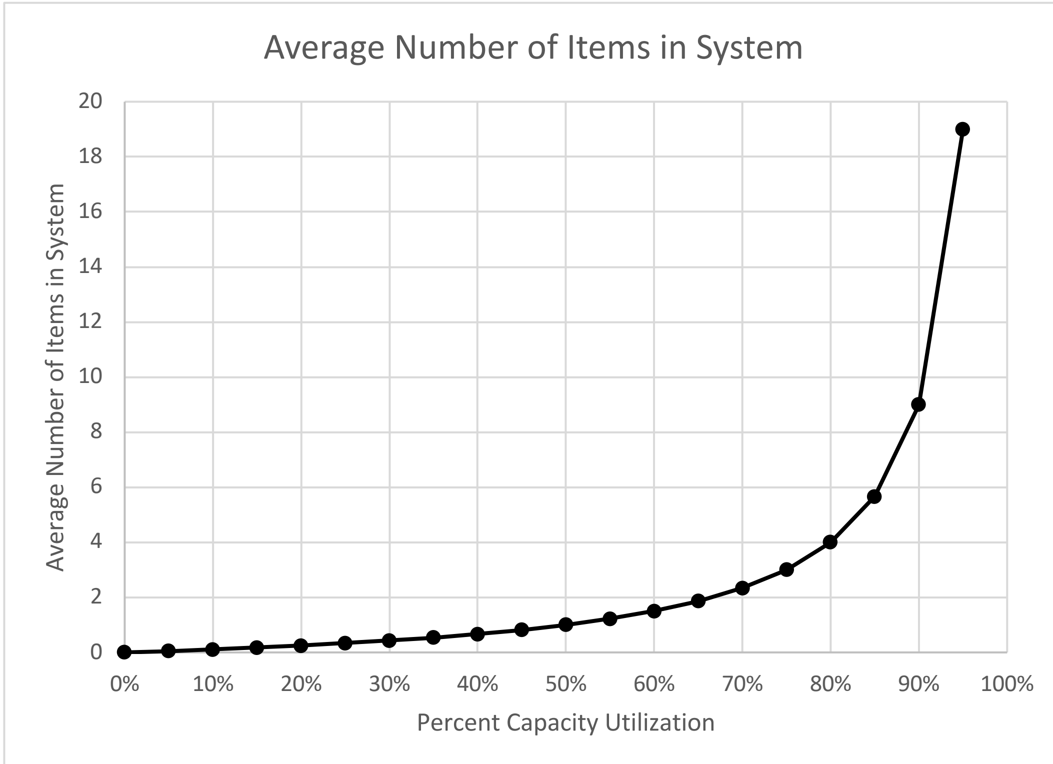
1. Use of the wrong economic objectives, that is a focus on proxy measures, like cycle times, rather than life-cycle profits;
2. Failure to recognize the importance of or measure queues, which lead to high volumes of in-process design "inventory;"
3. Inappropriate focus on efficiency, which leads to processes loaded to unreasonable utilization factors;

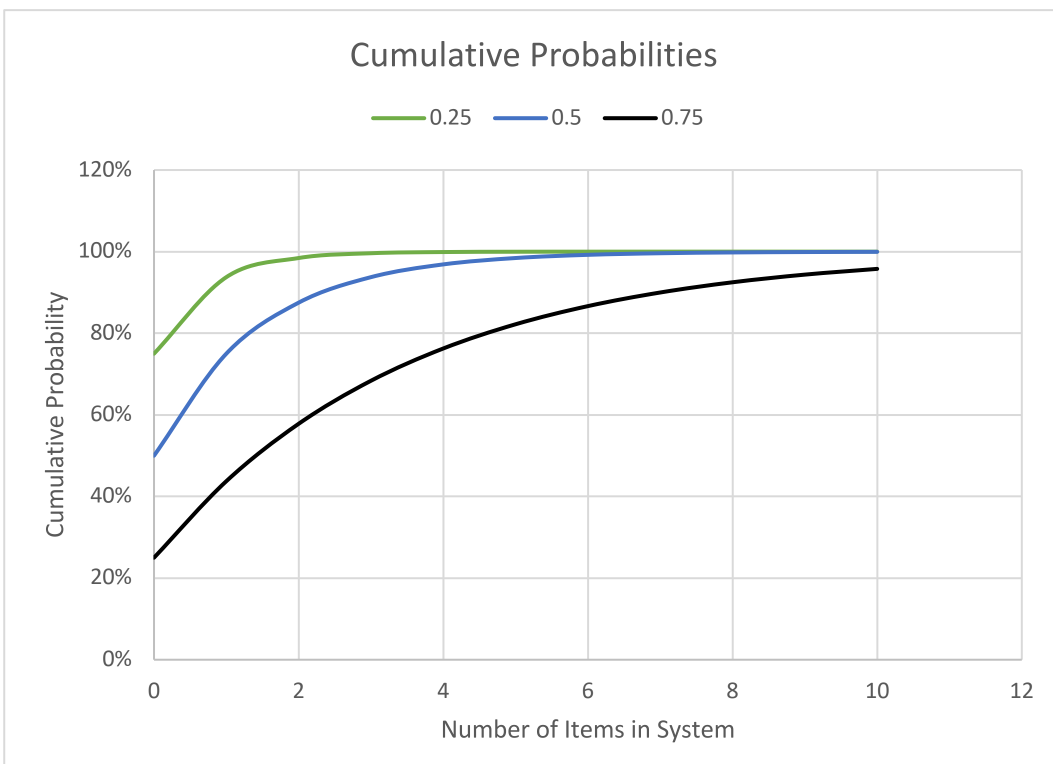
4. Failure to understand the role and value of variability, a practice that impedes innovation;
5. Overemphasis on conformance to plans at the expense of understanding new information;
6. Processes that institutionalize large batch sizes, such as phase-gate processes;
7. Failure to use cadence and synchronization;
8. Managing to timelines instead of managing queues, and failing to appreciate the implications and effects of variability;
9. Absence of limitations on work in process (WIP), as seen in lean manufacturing;
10. Inflexibility of resources, people and processes, which hinders responsivemeness to variability;
11. Failure to appreciate the cost of delay; and
12. Centralized control built on centralized information systems.

Drawing on concepts and ideas from a number of sources, including lean manufacturing, economics, queueing theory, statistics, control engineering, and military doctrine, he identifies 175 principles to address these problems. (Reinersten, 2009)

Reinersten identifies queues as the most important factor causing poor performance in product development. Unlike manufacturing, where inventory queues of physical items are obvious and have known costs associated with them, product development queues are invisible and appear to have no cost. Recalling basic principles of queueing theory, Reinersten notes capacity utilization, $\rho$, is the most important factor affecting queue size. For an M/M/1/∞ queue, items arrive according to a random Poisson process with rate $\lambda$ and are and have exponentially distributed service times with rate parameter $\mu$. Capacity utilization is the ratio of arrival rate to service time, $\rho = \frac{\lambda}{\mu}$, and the queue is stable only when $\lambda < \mu$. The average number of items in queue is $n = {}^{\rho}/_{(1-\rho)}$ and the probability that the queue is in state *n* (i.e., there are *n* items in queue) is $(1- \rho)\rho^n$. Figure 14 illustrates M/M/1/∞ queue behavior, showing the average number of items in the system versus capacity utilization and the cumulative probabilities of different queue states for different capacity utilization.

*Figure 14 - Plots Showing the Average Number of Items in a Queueing System versus Percent Capacity Utilization and the Cumulative Probabilities of Queue States for Different Capacity Utilizations*



Reinersten observes large queues increase variability, risk and cycle times while decreasing efficiency and quality. Queue size (the number of items in the system) increases expoentially with capacity utilization, and large queues create delays, although the cost of delay is often poorly understood. High queue states are rare, but have significant impact, amplifying delay and delay cost. Operating at high capacity utilization amplifies variability because small changes have greater effect at high capacity utilization (i.e., the slope of the upper plot in Figure 14 is much steeper at 90% utilization than at 50%). Turning to the economics of queues, Reinersten observes one can trade queue size against capacity using the theoretical optimum capacity, which for an M/M/1/∞ queue is given by

$$\mu_0 = \lambda + \sqrt{\frac{C_D \lambda}{C_C}}$$

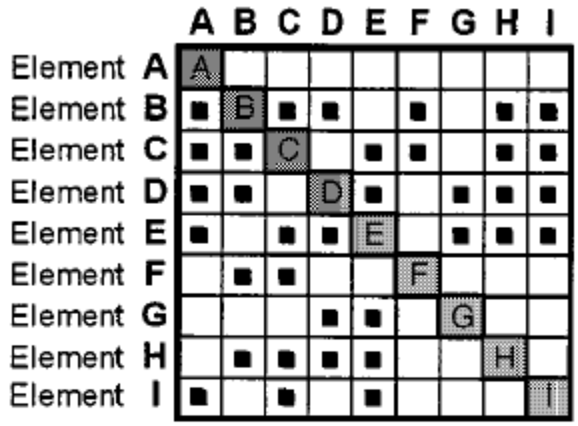
where $C_C$ and $C_D$ are the costs of capacity and delay respectively. He recommends several principles for managing queues, chief among them two imperatives: first, to monitor and control queue size rather than capacity utilization because neither demand nor capcity can usually be estimated accurately in product development; and second, to take prompt action to resolve high queue states because they are so damaging. (Reinersten, 2009)

Reinersten offers principles and recommendations across a broad range of topics, but some of his most interesting ideas relate to the value of decentralized control and maintaining alignment, achieved through application of principles from military doctrine. He notes military organizations rely on the initiative of subordinates to respond effectively to changing conditions and offers two key principles for balancing centralized versus decentralized control. First, he recommends decentralized control for problems and opportunitiies that require prompt action, an approach he compares to fire fighting. Second, he recommends centralized control for infrequent and large problems and for opportunities that benefit from economies of scale. For example, a centralized purchasing organization will often out-perform a decentralized one. With regard to alignment, he recommends an approach based on the conept of "mission orders" which describe end state, purpose, intent and a minimum number of constraints or limitations, and rely on clearly established roles and responsibilitieis. (Reinersten, 2009)

Modeling Systems with Design Structure Matrices

Design Structure Matrices (DSM) are a simple but powerful engineering management tool for analyzing the elements of a system and their interactions, and highlighting system architecture or structure. DSM use a square, i.e., $N \times N$, matrix to represent systems and map interactions. Figure 15 shows a simple DSM, showing how they illustrate interactions between generic system elements. DSM find application in the design of complex engineered systems, and can be used to model system architecture, organizational structure, and process arrangement. The primary benefit of DSM is their ability to represent information in a graphical, easy to understand format. Different terms have been associated with the DSM moniker, such as dependency structure matrix or dependency system model, to emphasize particular DSM aspects. The key criteria for DSM is that the DSM is a square matrix with identically ordered and labeled rows and columns and off-diagonal elements that indicate or describe relationships between elements. A similar tool, domain mapping matrices (DMM), link DSM across multiple domains, such as product architecture to organizational structure. DMM are rectangular, but not necessarily square, matrices since the linked DSM could be of different size. (Eppinger & Browning, 2012)

*Figure 15 Simple DSM Showing How DSM Represent Interactions Between System Elements (Browning, 2001)*




DSM can be classified into four types with three main categories. The first cateogry includes static architecture models, usually used to represent products or artifacts whose components interact with one another, or organizations whose members interact with each other. To reduce potential confusion, this report will use product or system architecture to refer to the physical arrangement of components in a product or complex engineered system, and organizational structure to refer to the arrangement of personnel within an organization, such as the engineering and design team responsible for the design and development of a product or engineered system. The second category includes temporal flow models that represent processes where system elements change or interact over time. The third category includes multi-domain matrices (MDM) that combine multiple DSM, such as product architecture and organizational strucutre, in a single matrix. (Eppinger & Browning, 2012) The following paragraphs examine each type of DSM in greater detail.

Product Architecture DSM Models. DSM aid both the design of system architecture, the "down" side of the Systems Engineering V, and the integration of components and subsystems, the "up" side of the V. (Eppinger & Browning, 2012) Figure 16 shows a product architecture DSM for a climate control system. The process for creating a product or system arcthitecture DSM involves decomposing the system into subsystems or components; laying out the elements on a square DSM, grouping subsystems or modules when appropriate; and identifying and marking interactions among elements.

When modeling system architecture with DSM, the user should consider several factors. First, the limits of the system may be poorly understood, so system boundaries should include the relevant components and interactions to be modeled. Second, the user must clearly identify the types of relationships and interactions relevant to the system, such as physical adjacency or spatial arrangement, material or energy flow, heat transfer, electrical interference, or environmental effects. The DSM may use different marks or colors to represent different types of interactions. Third, the DSM should quantify or qualify the strength of different interactions using binary (+/-), numerical or similar representations. Fourth, the user must establish the level of granularity modeled, trading off richness of detail against simplicity of the model. Finally, the user should consult multiple sources, including subject matter experts, when creating the system architecture DSM. (Eppinger & Browning, 2012)

*Figure 16 Component DSM Illustrating Material Flows in a Climate Control System (Browning, 2001)*

| | | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Radiator | A | A | 2 | | | | | | | | | | | | | | |
| Engine Fan | B | 2 | B | | | 2 | | | | | | | | | | | |
| Heater Core | C | | | C | | | | | | | | | | | | | 2 |
| Heater Hoses | D | | | | D | | | | | | | | | | | | |
| Condenser | E | | 2 | | | E | 2 | | 2 | | | | | | | | |
| Compressor | F | | | | | 2 | F | | 2 | 2 | | | | | | | |
| Evaporator Case | G | | | | | | | G | | | | | | | | | 2 |
| Evaporator Core | H | | | | | 2 | 2 | | H | 2 | | | | | | | 2 |
| Accumulator | I | | | | | | 2 | | 2 | I | | | | | | | |
| Refrigeration Controls | J | | | | | | | | | | J | | | | | | |
| Air Controls | K | | | | | | | | | | | K | | | | | |
| Sensors | L | | | | | | | | | | | | L | | | | |
| Command Distribution | M | | | | | | | | | | | | | M | | | |
| Actuators | N | | | | | | | | | | | | | | N | | |
| Blower Controller | O | | | | | | | | | | | | | | | O | 2 |
| Blower Motor | P | | | 2 | | | | 2 | 2 | | | | | | | 2 | P |

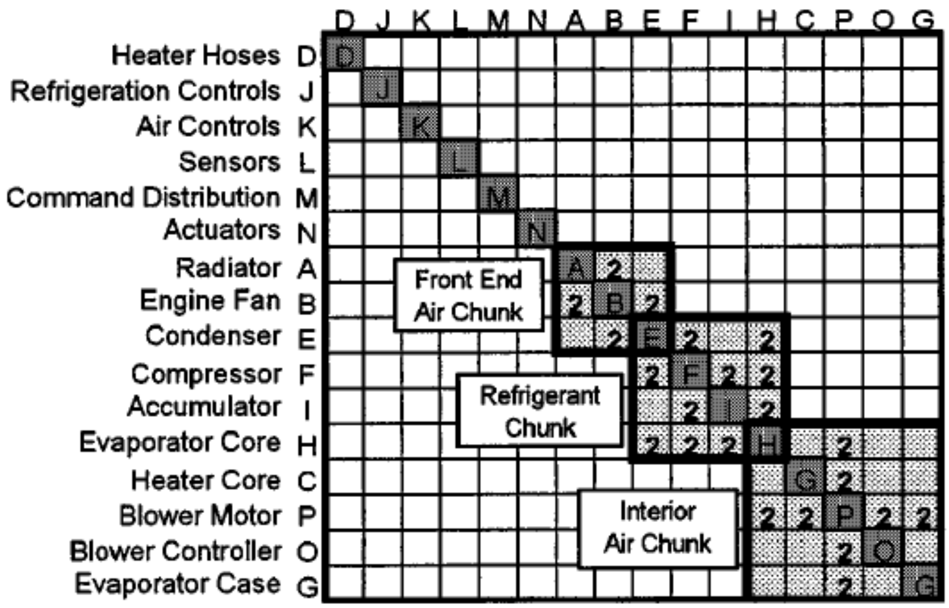
Building the DSM provides insight, but the real benefit comes from the analysis of system architecture using clustering techniques that reorder or group system elements according to some objective, often related to the number and strength of interactions. In that regard, clustering is a type of assignment problem that seeks to optimize the allocation of $N$ elements to $M$ clusters using objective functions that trade off competing goals of minimizing the number or strength of interactions outside clusters against cluster size. For example, the following objective function could be used:

$$z = \alpha \sum_{i=1}^{M} C_i^2 + \beta I_0$$

Where α and β are constants, $C_i$ is the size of cluster $i$, and $I_0$ is the number of interactions outside a cluster. In addition, clustering techniques generally try to choose modules that are as independent as possible, although complex engineered systems often exhibit both modular and integrative subsystems. (Eppinger &

Browning, 2012) Figure 17shows a clustered DSM for the climate control system shown in Figure 16.

*Figure 17 Clustered DSM for a Climate Control System (Browning, 2001)*



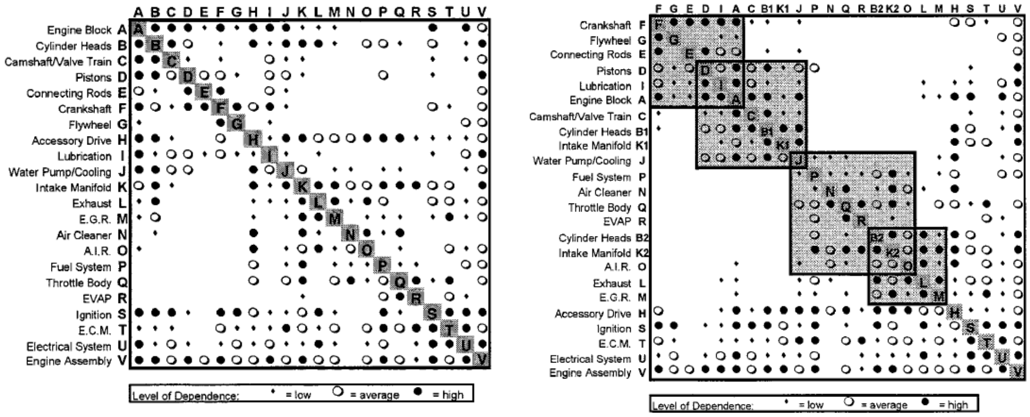

Eppinger and Browning argue product architecture DSM provide effective representations of components and their relationships, illustrating decomposition and interactions. Clustering analysis identifies alternative groupings of components into modules, improving understanding and facilitating innovation. DSM are particularly helpful for large systems where system complexity "makes it impossible for any single individual to have a complete, detailed, and accurate mental model of the entire system." (Eppinger & Browning, 2012)

<u>Organizational Structure DSM Models</u>. Organizational structure DSM capture organizational elements, such as individuals, groups or departments, as rows and columns, and interactions and communication pathways in off-diagonal cells. The process for creating an organizational structure DSM involves decomposing the

organization into elemental units, such as departments, divisions or individuals; laying out the DSM with organizational elements along the rows and columns, grouped as higher-level elements if appropriate; and identifying and marking actual or desired interactions between elements in the off-diagonal cells.  The considerations for organizational structure DSM are similar to those for product architecture DSM (Eppinger & Browning, 2012)

Similar to product architecture DSM, the analysis of organizational structure DSM relies on clustering techniques that typically focus on grouping people with the greatest need to communicate since the need to communicate often suggests the application of integrative mechanisms, like co-location, meetings, or distribution lists. Analysis of an organizational DSM may explore several scenarios and trade off advantages and disadvantages of different potential structures, including both political and practical considerations related to group size, location, or composition. Organizational structure DSM provide intuitive visualization and facilitate discussions around the flow of information, while clustering analysis generates alternative perspectives to improve understanding, facilitate innovation, and inform use of integrative mechanisms. (Eppinger & Browning, 2012)  Figure 18 shows an original and clustered DSM for an automobile engine product development team.

*Figure 18 Organizational Structure DSM for an Automobile Engine Product Development Team: Original and Clustered (Browning, 2001)*




<u>Process Architecture DSM and Multi-Domain Matrices</u>. DSM can also be used to model temporal processes. Such DSM represent the activities in a process and their interactions, and are known by many names: process architecture DSM, process DSM, process flow DSM, activity-based DSM, and task-based DSM. The process for building a process flow DSM involves decomposing the process into activities; laying out the DSM with activities on the rows and columns, grouped into subprocesses or states, if appropriate; and identifying interactions between activities. A unique feature of process flow DSM is the use of markings and designators to represent one of four fundamental types of relationships: sequential activities; parallel activities; coupled activities, meaning those that must converge to a mutually satisfactory result; and conditional activities that depend on upstream activities. The analysis of process flow DSM relies on sequencing methods that reorder activities to minimize iterations, or cycles. Sequencing methods seek to eliminate or minimize feedback, or shorten feedback that cannot be eliminated, recognizing that long feedbacks create a situation where interim activities can proceed with incomplete or inaccurate information, leading to costly and time consuming rework. (Eppinger & Browning, 2012)

Multi-Domain Matrices (MDM) extend DSM by simultanesouly representing two or more DSM from different domains. MDM are closely related to domain mapping matrices (DMM), which are rectangular (i.e., $n$ x $m$) matrices that map relationships between two DSM, one of size $n$, and the other of size $m$. The two can be easily confused because of the similarity in names and acronyms. DMM can hold binary values, indicating the presence or absence of a relationship between two domains, or may use symbols or numbers to indicate the strength, degree, or type of relationship. DSM and MDM can be used together to analyze the influence of one domain on another. Possible applications include identifying the need for cross-functional interactions based on product component or process activity interactions; inferring interactions in other domains; and evaluating how project domains affect one another. (Eppinger & Browning, 2012)

Agent-Based Modeling[2]

Wilensky and Rand (2015) describe agent-based modeling (ABM) as a computational approach in which phenomena are modeled "in terms of agents and their interactions," and argue ABM represents a transformational technology that enables better understanding of familiar topics while facilitating exploration of previously unexplored topics. Taking predator-prey interactions as an example, they note such interactions can be modeled using a system of coupled differential equations. Though relatively straigthforward to solve, the equation-based approach provides no insight into individual behavior, and embeds a simplifying assumption

---

[2] This section incorporates information presented by William (Bill) Rand in the online course *An Introduction to Agent-Based Modeling*, offered through the Santa Fe Institute's Complexity Explorer program. For additional information, visit the Complexity Explorer website: http://complexityexplorer.org.

that agents are sufficiently homogeneous to permit use of average quantities. Agent-based representations accommodate heterogeneity and may be simpler to understand: "agent-based representations are easier to understand than mathematical representations of the same phenomenon," because agent-based models are built from individual objects (i.e., agents) and simple behavior rules. (Wilensky & Rand, 2015)

Wilensky and Rand also explore the challenge of understanding complex systems and emergence, noting the need for both integrative and differential understanding. Integrative understanding relates to discerning aggregate patterns when individual behaviors are known, while differential understanding relates to discerning individual behaviors when the aggregate pattern is known. Agent-based modeling addresses both challenges because it provides a way to explore how the actions and interactions of individual agents affect aggregate system behavior. (Wilensky & Rand, 2015)

<u>Description of Agent-Based Models</u>. Agent-based models are based on the idea that many phenomena can be represented by agents, their environment, and rules governing agent-to-agent and agent-to-environment interactions. Agents are autonomous entities with properties, actions and goals, while the environment is the terrain where the agents interact. Agent behaviors can change over time based on interactions, information exchange, or other factors. Wilensky and Rand argue agent-based models offer several advantages. First, agent-based models accommodate heterogeneous populations and discrete interactions. Second, they do not require knowledge of aggregate behavior, relying instead on simple behavior rules. Third, they closely match real-world behavior, making them easier to understand. Fourth,

they simultaneously provide both individual- and aggregate-level detail. Finally, they readily accommodate randomness and random behaviors. (Wilensky & Rand, 2015) Wilensky and Rand acknowledge there are situations for which the costs of creating agent-based models exceed their benefit and provide guidelines for selecting situations where agent-based models provide the greatest benefit:

- Systems with a moderate (tens to millions) of interacting agents;
- Systems comprised of heterogeneous agents;
- Systems characterized by complex, history- or property-dependent, and local agent-to-agent interactions;
- Systems involving rich environments, such as social networks or geographical systems;
- Systems that exhibit time-dependent, i.e., step-wise, behavior;
- Systems where agents adapt over time, such that future behavior depends on past behavior and agents change behavior based on experience.

Among these criteria, time dependence is considered a necessary condition, while adaptation is considered a sufficient condition. Virtually all agent-based models evaluate system behavior in discrete time steps, making time-dependence a necessary condition. Furthermore, few other approaches accommodate adaptive agents, making adaptation a sufficient condition for using agent-based models. (Wilensky & Rand, 2015)

Despite their power, however, agent-based models have important limitations. First, agent-based models can be computationally expensive, requiring extensive computational power to simulate many individual agents. Second, the modeler must use judgment when deciding which variables to model, so must have some knowledge about how the system operates. Finally, most agent-based models require some knowledge of individual agent behaviors. (Wilensky & Rand, 2015)

Creating Agent-Based Models. Wilensky and Rand explore issues related to designing, building, and examining agent-based models. They identify two major categories of models: phenomena-based and exploratory. Phenomena-based models start with a phenomenon that exhibits a known, or reference, pattern, and then create a model—a set of agents and the rules governing their behavior—that generates the reference pattern. Exploratory models start with agents and their behaviors and then explore the patterns that emerge. A related feature of modeling methodology is the degree to which the model seeks to answer a specific question. At one extreme, one might formulate a specific research questions, such as "How do organizations effectively manage the design of complex engineered systems?" At the other extreme, one might start with only a desire to model organizational structures. Another dimension of agent-based modeling is the relationship between the conceptual model and the code written to implement it. In some cases, a top-down approach is appropriate. In top-down models, the conceptual model is fully specified—agents, environment, rules govnerning behavior and interactions—before writing any code to implement it. In other cases, a bottoms-up approach is better. In bottoms-up models, the conceptual model and code evolve together.

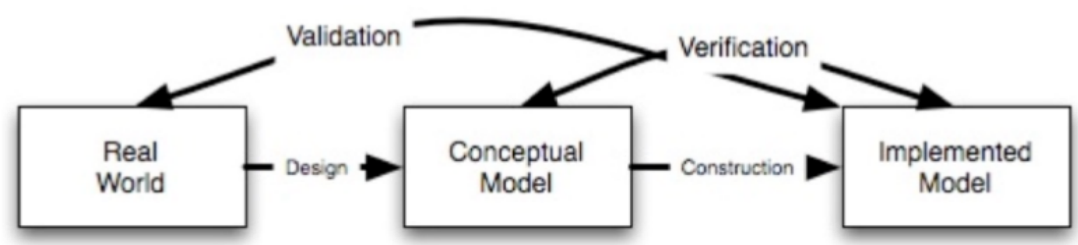
Wilensky and Rand identify an essential design principle for agent-based models: "start simple and build toward the question you want to answer." An agent-based model should start with the simplest set of agents and rules possible, and should avoid adding anything that detracts from answering the question motivating the model. To help modelers implement this principle, Wilensky and Rand identify seven critical design choices. The first choice involves identifying the question to be answered. Noted before, models and questions often co-evolve, but it is important to confirm the phenomenon and system are suited to agent-based modeling using the guidelines given before. The second choice involves identifying the agents to be used in the model. Since every entity can be subdivided into several smaller entities, it is important to match the granularity or scale of agents to the temporal scale of interest. In addition, the need for any proto-agents should be identified. Proto-agents do not have their own rules or behaviors. Instead, they take on characteristics from a global agent type. The third and fourth choices are related and involve identifying agent properties and behaviors respectively. Properties describe individual agents and help distinguish one from another. Behaviors describe what agents do, including how they interact with one another and the environment. The fifth choice involves describing the environment in which the agents exist. Some environments can be described using stationary agents. For example, the environment may consist of a grid, where each block has its own properties and rules governing its interactions with other elements of the environment and the agents that exist in it.

The sixth choice involves selecting the amount of time associated with each step of the model and describing what happens during each time step. The seventh and final choice involves selecting model inputs and outputs. Inputs include global parameters that affect how the model behaves and might include items like the initial number of agents or their initial arrangement in the environment. Outputs include the measures required to answer the question of interest. It is often wise to limit the number of measures used to prevent data overload. (Wilensky & Rand, 2015)

Analyzing Agent-Based Models. Agent-based models present unique analsysis and interpretation challenges compared to equation-based models because agent-based models allow users to control many agent characteristics, which often results in large numbers of inputs and outputs. While this flexibility is one feature giving agent-based models their power, it also creates concerns. For example, using more inputs means there are more parameters to validate against real-world data, while more outputs can lead to data overload and make it difficult for users to discern clear patterns of behavior since modelers must often examine many different relationships between inputs and outputs to identify key relationships. Wilensky and Rand identify four classes of data commonly associated with agent-based models: statistical, graphical, network, and spatial. Statistical results include standard measures like mean, variance, median and other measures. An important consideration for analyzing agent-based models is the need for multiple runs and statistical analysis of results because agents commonly exhibit stochastic behavior. Graphical results present outputs in the form of plots and graphs, rendering them more understandable. Network measures, like clustering coefficient and path length, are useful for network-based models. Finally, spatial measures help identify patterns in one-, two- or higher dimensional space. (Wilensky & Rand, 2015)

Verification and Validation. George Box famously said, "all models are wrong, but some are useful." Verification and validation evaluate the accuracy of models to ensure they adequately represent real-world behavior and provide outputs useful to the model's user. *Verification* confirms the implemented model corresponds to the conceptual model, to ensure that you built the model you meant, while *validation* confirms that the implemented model explains and corresponds to real-world phenomena, that you built the "right" model. Figure 19 illustrates these relationships. Verification and validation increase confidence in the "correctness and exploratory power of both the conceptual and implemented models." (Wilensky & Rand, 2015)

*Figure 19 Relationship between model verification and validation (Rand, 2016)*




Rand and Rust propose guidelines for rigorous verification and validation of agent-based models, arguing both activities should be performed to the extent necessary to convince the target audience of the model's accuracy. They identify three key elements of verification: documentation, programmatic testing, and test cases. Documentation refers to descriptions of the conceptual and implemented models, which should provide sufficient detail to facilitate comparison and should include choices related to model design and comments within the model's code to identify how the code implements the conceptual model. (Rand & Rust, 2011)

Programmatic testing ensures the implemented model does what the programmer expects and includes unit testing, code walkthroughs, debugging, and formal testing. Unit testing refers to tests on individual sections of codes. For example, if the code performs calucations, then those calculations should be checked. Code walktrhoughs refer to peer or group reviews to compare the code to the conceptual model. Debugging refers to systematic evaluations to verify proper outputs, usually using a debugging tool. Formal testing refers to the use of logic to demonstrate model correctness, although formal testing of complicated agent-based models is often difficult. (Rand & Rust, 2011)

Test cases use artificial data to ensure proper model function. Rand and Rust identify four categories of test cases: corner cases, sampled cases, specific scenarios, and relative value testing. Corner cases use extreme input values to check for abberant behavior. Sampled cases use subsets of input values to check for abberant behavior. Specific scenarios test input parameters for which outputs are known from previous work, subject matter expertise, or insight into the conceptual model. Relative value testing uses known relationships between inputs and outputs, such as when increasing an input will increase the output. (Rand & Rust, 2011)

Rand and Rust identify four key elements of model validation: micro-face validation, macro-face validation, empirical input validation, and empirical output validation. The first two categories confirm the model makes sense "on face," while the latter two confirm the model's results match real-world data. Micro-face validation confirms the model's methods and properties match those of the phenomenon modeled. For example, it confirms the model's agents accurately represent real-world agents. Macro-face validation confirms the aggregate patterns generated by the implemented model correspond to real world patterns. For example, it confirms the model generates dynamic behaviors consistent with real-world behavior. Both micro- and macro-face validation focus on demonstrating and explaining correspondence between model and real-world behaviors, and it is usually sufficient “to describe the relationship between the model and the real world to show that it has been validated ‘on face.’” (Rand & Rust, 2011)

Empirical validation confirms input or output data corresponds to real world data and facts. Empirical input validation ensures the accuracy of relevance of data used as an input to the model. It is often sufficient to explain how input data was derived and demonstrate its correspondence to the real world. When possible, models should be calibrated and tested using real-world input data. Emprical output validation ensures the output of the implemented model corresponds to the real world data and facts and is the key test of model validity because it tests the author’s hypothesis. Rand and Rust identify three approaches to empirical output validation: stylized facts, real-world data, and cross-validation. Stylized facts are general ideas about a system obtained from expert opinion and are generally used for thought experiements. Real-world data validation compares model outputs to historical data to validate model accuracy. Cross-validation compares ouputs from a new model to those from a prior model to increase confidence in the new model’s validity. (Rand & Rust, 2011)

## CHAPTER 3

## METHODOLOGY

### Overall Approach and Research Questions

This study will investigate the effectiveness of different organizational structures (organizational networks) at designing complex engineered systems. Specifically, it will evaluate and compare the ability of different organizational networks to deliver design products and share information in the presence of complexity using agent-based modeling (ABM). A phased, building block approach will be followed. Phase 1 will examine information exchange models and implement the model of information exchange proposed by Dodds, Watts and Sabel to confirm the model can be successfully implemented using ABM. Phase 2 will examine artifact models and extend the information exchange model to include the processing of work products, termed artifacts. Phase 3 will examine smart team models, which include alternate network construction algorithms and alternative methods for processing work products. Phase 4 will apply information exchange and artifact models to a real-world organization. The following research questions will be answered:

- How do random, multiscale, military staff and matrix organizational networks perform in the information exchange and artifact task environments and how does increasing the degree of complexity affect performance?
- How do military staff and matrix organizational networks (real organizations) perform compared to one another and to random and multiscale networks (ideal

organizations)? How does increasing degree of complexity affect performance and which structure is preferred for organizations that design complex engineered systems?

- How can organizational networks be modified to improve performance?

Organizational Structures and Networks Examined

Organizational structure defines how people work together to accomplish objectives and create value and includes the hierarchical structure that defines an organization's functional decomposition, lines of authority and responsibility, and formal reporting relationships, as well as the teaming structures that cross horizontal and vertical lines and exist to facilitate communication, problem solving, and task accomplishment. Given the well documented relationship between product architecture and the structure of the product development organization, it is logical and appropriate to examine organizational structure for causes and factors explaining why design organizations sometimes fail to effectively manage the design of complex engineered systems. Dodds, Watts, and Sabel identified a class of networks, *multiscale networks*, that simultaneously reduce the likelihood an individual node will fail because of congestion and the likelihood the overall network will fail if congestion failures do occur at individual nodes. (Dodds, Watts, & Sabel, 2003) Because of their robustness to failure, multiscale networks represent an ideal type and provide a basis for comparing and evaluating real-world organizational networks. Random networks represent another ideal type and likewise provide a basis for evaluating and comparing real-world organizational networks. This study will compare the effectiveness of matrix and military staff organizational networks to multiscale and random networks in order to understand the factors affecting the ability of design organizations to manage the design of complex engineered systems, and to identify ways performance can be improved.

Agent-Based Modeling and NetLogo

Agent-based models represent phenomena using agents, their environment, and rules governing agent-to-agent and agent-to-environment interactions. Organizational networks satisfy the criteria for selecting ABM proposed by Wilensky and Rand, thus ABM is an appropriate tool for evaluating the effectiveness of organizational networks. (Wilensky & Rand, 2015) Specifically, organizational networks have a moderate number of heterogeneous, interacting agents; are characterized by local agent-to-agent interactions; exhibit time-dependent behavior; and adapt over time.

The NetLogo ABM environment will be used to implement models. NetLogo is an open source, cross-platform modeling environment authored by Uri Wilensky in 1999. It has been continuously developed at the Center for Connected Learning and Computer-Based Modeling at Northwestern University since then. (Northwestern University, 2017)

Phase One: Information Exchange Models

The first phase implements the Dodds, Watts and Sabel (DWS) model of information exchange to confirm the model can be successfully implemented using the NetLogo ABM environment. The model starts with a pure hierarchy defined by number of levels, $L$, and branching ratio, $b$, with the total number of nodes given by

$$N = \frac{b^L - 1}{b - 1}$$

The model then adds $m$ additional team links according to a stochastic rule in which the probability that a new link forms between two nodes, $i$ and $j$, $P(i,j)$, depends on the organizational distance between the two nodes, $x_{ij}$, and the rank of the two nodes' lowest common ancestor, $D_{ij}$. The model employs two tunable parameters, $\lambda$ and $\zeta$, which correspond to ancestor rank and organizational distance respectively. The resulting stochastic rule:

$$P(i,j) \propto e^{-x_{ij}/\zeta} e^{-D_{ij}/\lambda}$$

In addition to the construction algorithm, the DWS model includes a description of the task environment, a method of information exchange, and a measure of performance. (Dodds, Watts, & Sabel, 2003)

The DWS model describes the task environment in terms of the rate and distribution of messages to be exchanged between individual nodes in the organizational network. The information exchange rate, $\mu$, is the average number of messages originated by each node at each time step, and $\mu N$ is the total number of messages originated across the network at each time step. Message routing considers task decomposability. Tasks that are nearly decomposable require communication only within the same team, meaning nodes with the same immediate superior, whereas tasks that are decomposable require communication across the network. For a given source node, *s*, a target node, *t* is selected based on the distance between the two nodes, $x_{st}$, using the following stochastic rule:

$$P(s,t) \propto e^{\frac{-x_{st}}{\xi}}$$

When $\xi = 0$, local dependencies prevail; when $\xi = \infty$, global dependencies prevail. (Dodds, Watts, & Sabel, 2003)

Messages pass from source to target through a chain of intermediate nodes. During each time step, nodes pass messages they initiate or receive to an immediate neighbor with the lowest common ancestor with the target node. This method reflects an assumption termed "pseudo-global knowledge," which assumes individual nodes understand their own locations, and the locations of their immediate neighbors, and have general information about nodes beyond their immediate neighborhood. (Dodds, Watts, & Sabel, 2003)

The DWS model adopts congestion centrality as a measure of network performance. Assuming each node can process up to $R_i$ messages per time step, an organizational network will, on average, remain free of congestion when $R_i > r_i = \mu N \rho_i$, where $\rho_i$, the congestion centrality of node $i$, is the probability that any given message will be processed by node $i$. Maximum congestion centrality across the organizational network, $\rho_{max}$, is a measure of robustness to congestion failure. (Dodds, Watts, & Sabel, 2003)

Phase one extends the DWS model to matrix and military staff organizational networks by altering the network construction algorithm. These networks begin with the same underlying hierarchical network but employ different methods to add team links. The task environment, method of information exchange, and use of maximum congestion centrality to measure network performance remain unchanged.

For matrix organizational networks, the model adds an additional major branch to the hierarchy to represent a project management organization. This new branch has only two levels, with the first level representing the manager of the project management organization and the second representing the project managers. After creating the project management organization, the model adds $m$ team links at random between project manager nodes and worker nodes in the main hierarchical network. The underlying hierarchical network represents the functional organization, while team links represent the assignment of workers from the functional organization to project teams.

For military staff organizational networks, the model first identifies team leads at random from among nodes at the top levels of the hierarchical network. The model then adds *m* team links at random between team leader nodes and worker nodes at the bottom levels of the hierarchy. This arrangement represents the fact that in military staffs, individuals perform both functional tasks according to their position in the hierarchy and cross-functional tasks according to teams to which they are assigned.

Phase one further extends the DWS model to account for the effect of complexity on how information is exchanged. Complexity affects the decomposability of tasks performed by the design organization. When the system being designed is more complex, tasks tend to be less decomposable because the system has more interactions, which are often poorly understood. As a result, tasks require greater cross-functional collaboration. Conversely, when the system being designed is less complex, tasks tend to be decomposable and require little cross-functional collaboration.

The model implements the effect of complexity by adding a complexity input that allows the user to rate complexity on a scale of 1 to 10. When the model creates new messages, it compares a random number to the complexity rating. If the random number is less than the complexity rating, the situation is considered complex and the target node is selected at random from other nodes across the hierarchy. If the random number is greater than or equal to the complexity rating, the situation is considered routine and the target node is selected at random from other nodes in the same major branch of the hierarchy (i.e., same functional organization). Although the complexity rating employs a numerical scale, it is meant to provide a qualitative, not quantitative, representation of complexity. Recalling the task environment of the DWS model, a high complexity rating corresponds to global dependencies, $\xi \to \infty$, while a low complexity rating corresponds to local dependencies, $\xi \to 0$.

Phase one will implement the DWS information exchange model in NetLogo for random and multiscale networks; verify and validate the information exchange model, to include cross-validation against a model implemented in MATLAB; extend the information exchange model to include matrix and military staff organizational networks and the effect of complexity; and characterize and compare the performance of random, multiscale, matrix and military staff organizational networks.

Phase Two: Artifact Models

The second phase extends the DWS model to include the processing of work products, often called artifacts. The *artifact* model implemented in phase two uses the same network construction algorithms as the information exchange models, but modifies the task environment, methods of information exchange and measure of performance used in the information exchange networks implemented in phase one.

Design organizations develop a wide range of artifacts to develop and deliver complex engineered systems. Examples include drawings, specifications, analyses, reports, correspondence, and other documents. Workers, supervisors and managers in the design organization must create, review and approve artifacts to meet design and schedule objectives. In general, the review and approval of artifacts follows functional, hierarchical lines. The creation of artifacts also requires cross-functional collaboration, thus design organizations must process artifacts and share information.

The artifact model describes the task environment in terms of the rate and distribution of artifacts to be processed and the rate and distribution of messages that must be exchanged to accomplish cross-functional collaboration. The artifact rate, $\mu_A$, is the average number of artifacts originated by each node at each time step, and $\mu_A N$ is the total number of artifacts originated across the network at each time step. Artifact routing follows the functional hierarchy. Workers at the lowest level of the hierarchy originate artifacts and then pass them up the functional chain of command to a manager near the top of the hierarchy for approval. For simple tasks, the originating worker likely has sufficient information to complete the artifact without the need for cross-functional collaboration. For complex tasks, however, the worker likely lacks sufficient information and requires additional information from other workers. In this case, the originating worker places the artifact on hold and originates a request for information (RFI) to acquire the additional information required to complete the artifact. RFIs pass from source to target through a chain of intermediate nodes as with messages in the information exchange model. Upon receipt, the RFI target provides the information requested and returns the RFI directly to the originator. When the originator receives an answered RFI, he completes the associated artifact and routes if for approval.

Complexity affects the rate and distribution of RFIs. At low complexity, few RFIs are created, and because tasks are decomposable, RFIs are routed to other workers in the same functional organization. At high complexity, many RFIs are created. Since tasks are not decomposable, RFIs are routed to other workers across the organization. The *Artifact* model uses the same qualitative complexity scale used in the information exchange models implemented in phase one.

During a given time step, nodes process a number of artifacts and information requests up to their capacity. If a given node has only RFIs or artifacts available, it processes them, but if both are available, it decides which to process by comparing a random number to an artifact preference rating, in the range [0,1]. When the artifact preference rating is higher, it is more likely the node will select an artifact than an RFI. An artifact rating of 0.5 represents a "coin flip," with the node choosing RFIs half the time, and artifacts the other half.

The artifact model adopts artifact completion rate (number of artifacts completed divided by the total number of artifacts created) as a measure of organizational network performance. If the organizational network is able to keep pace with the demand for artifact processing and information sharing, artifact completion rate will tend to unity, with a small deviation resulting from the number of artifacts being processed during any particular time step. However, if the organizational network fails to keep pace with demands for artifact processing and information sharing, the artifact completion rate will drop and the organization will fall further and further behind.

Congestion centrality remains an important indicator of network performance, but separate centralities must be considered. For any node:

$$r_A = \mu_A N \rho_A$$

$$r_{RFI} = \mu_{RFI} N \rho_{RFI}$$

The "A" subscript refers to artifacts, while the RFI subscript refers to RFIs. In addition, an overall or effective congestion centrality, $\rho_{eff}$, can be defined. On average, the network will remain free of congestion when, for any node, $R > r_A + r_{RFI}$, where $R$ is the node's capacity. Noting that total RFI rate is proportional to artifact arrival rate, $\mu_{RFI} N = k\mu_A N$ leads to:

$$r_A + r_{RFI} = \mu_A N \rho_A + k\mu_A N \rho_{RFI}$$

Letting $\rho_{eff} = \rho_A + k\rho_{RFI}$ leads to:

$$\rho_{eff} = \frac{r_A + r_{RFI}}{\mu_A N}$$

For the artifact model, maximum effective congestion centrality is a measure of organizational network robustness to congestion failure.

Phase Three: Smart Team Models

The third phase extends the artifact model to test different network construction algorithms, task environments and methods of artifact routing. The resulting model is called the "Smart Team" model referring to the idea that performance can be improved through smart decisions around organizational design.

Network Construction. In previous models, the network construction algorithm for matrix and military staff organizational networks first identified team leaders and then added $m$ team links between team leaders and workers in the hierarchical organization. The resulting teams are linked through the team leader but have no internal links between team members. In reality, a team would have these sorts of internal links in addition to the links between members and the leader. The smart team model uses a stochastic rule to determine how team links are added to matrix and military staff organizational networks. The rule captures the idea that as a team grows in size, it is more likely that new links will be added between existing team members. The rule is:

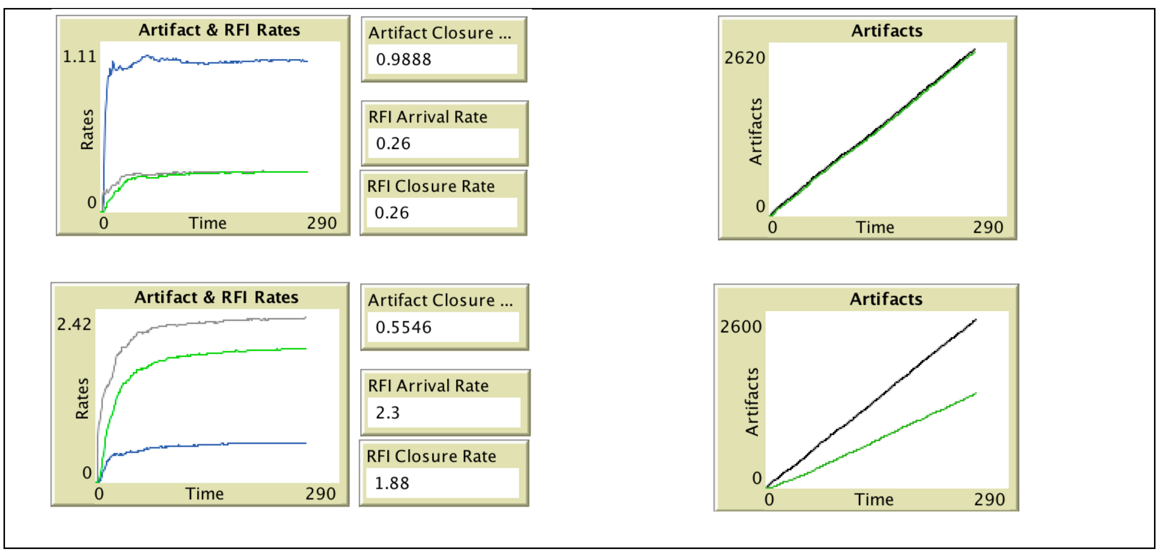
$$p_{new} = e^{-S/\chi}$$

where $p_{new}$ is the probability that a new member will be added to the team, $S$ is the current size of the team, and $\chi$ is a scaling factor for team size. As team size increases in relation to the scaling factor, it is increasingly less likely that new members will be added and instead a new link will be added between existing team members.

Task Environment and Artifact Routing. The smart team model carries forward from the artifact model the idea of artifact preference and tests whether an RFI, neutral, or artifact preference improves organizational performance. Noting that previous models have demonstrated the positive impact of decentralizing congestion, the smart team model provides an option to allow approval of artifacts at the supervisor vice manager level. The model implements decentralized artifact approval with a decentralized approval preference rating. When decentralized approval is enabled, the model allows supervisor approval of an artifact when a random number is less than the decentralized approval preference.

Testing for Congestion. The smart team model implements a specific test for congestion. When an organizational network is free of congestion, artifact completion rate stabilizes near unity and plot of total artifacts and completed artifacts converge to the same slope. Conversely, when an organizational network experiences congestion failure, artifact completion rate diverges from unity and plots of total artifacts and completed artifacts also diverge. Figure 20 illustrates this behavior using plots from NetLogo. The upper plots show a network free of congestion, while the lower plots show a congested network.

*Figure 20 - NetLogo Plots Illustrating Artifact Behavior for Congested Networks and Networks Free of Congestion (the upper plots show a network free of congestion, while the lower plots show a congested network)*




The difference between total artifacts and completed artifacts can be used as a test for congestion. The smart team model defines the variable delta slope, $\delta$, as

$$\delta = \frac{\sum(y - \bar{y})(t - \bar{t})}{\sum(t - \bar{t})^2}$$

where $t$ refers to time step and $y$ is the difference between total artifacts and completed artifacts at a given time step. Readers will recognize delta slope is the least-squares slope of a line fit to a plot of $y$ versus $t$. When a network is free of congestion, $\delta$ will tend to zero. The smart team model calculates delta slope over the last ten time ticks.

PW-4098 Case Study

The fourth and final phase applies information exchange and artifact models to real-world organizations in order to evaluate how well models represent actual behavior and likewise understand how real-world organizational networks behave under model conditions. Phase 4 examines the integrated product development organization used at Pratt and Whitney during design of the PW4098 turbofan engine.

Background. Rowles applied principles of design structure matrices to analyze information flow in the integrated product development organization used by Pratt and Whitney during development of the PW4098 turbofan engine. After a decade of developing concurrent engineering capabilities, Pratt & Whitney evolved a complex matrix organization to support integrated product development. A hierarchy of cross-functional teams replaced purely functional organizations with teams grouped around engine program, systems and subsystems, specific parts, and manufacturing operations. Discipline centers replaced traditional functional organizations, with the resulting organization being characterized as a heavyweight project matrix organization. (Rowles, 1999)

Figure 21 depicts the PW4098 design organization’s structure and shows division of the PW4098 design team into component or sub-system teams, or interface groups, while Figure 22 presents a design structure matrix showing strong and weak interactions between individual elements of the organization. Rowles found approximately one-third of interactions occurred outside of the component or sub-system teams or integration groups, and that approximately one-fourth of interactions did not correspond known or planned design relationships, with the majority of these unplanned interactions being with three teams: burner, main shaft, and HPT Case/OAS. (Rowles, 1999)

*Figure 21 - Organization Structure for Design of the Pratt & Whitney PW4098 Turbofan Engine (Rowles, 1999)*

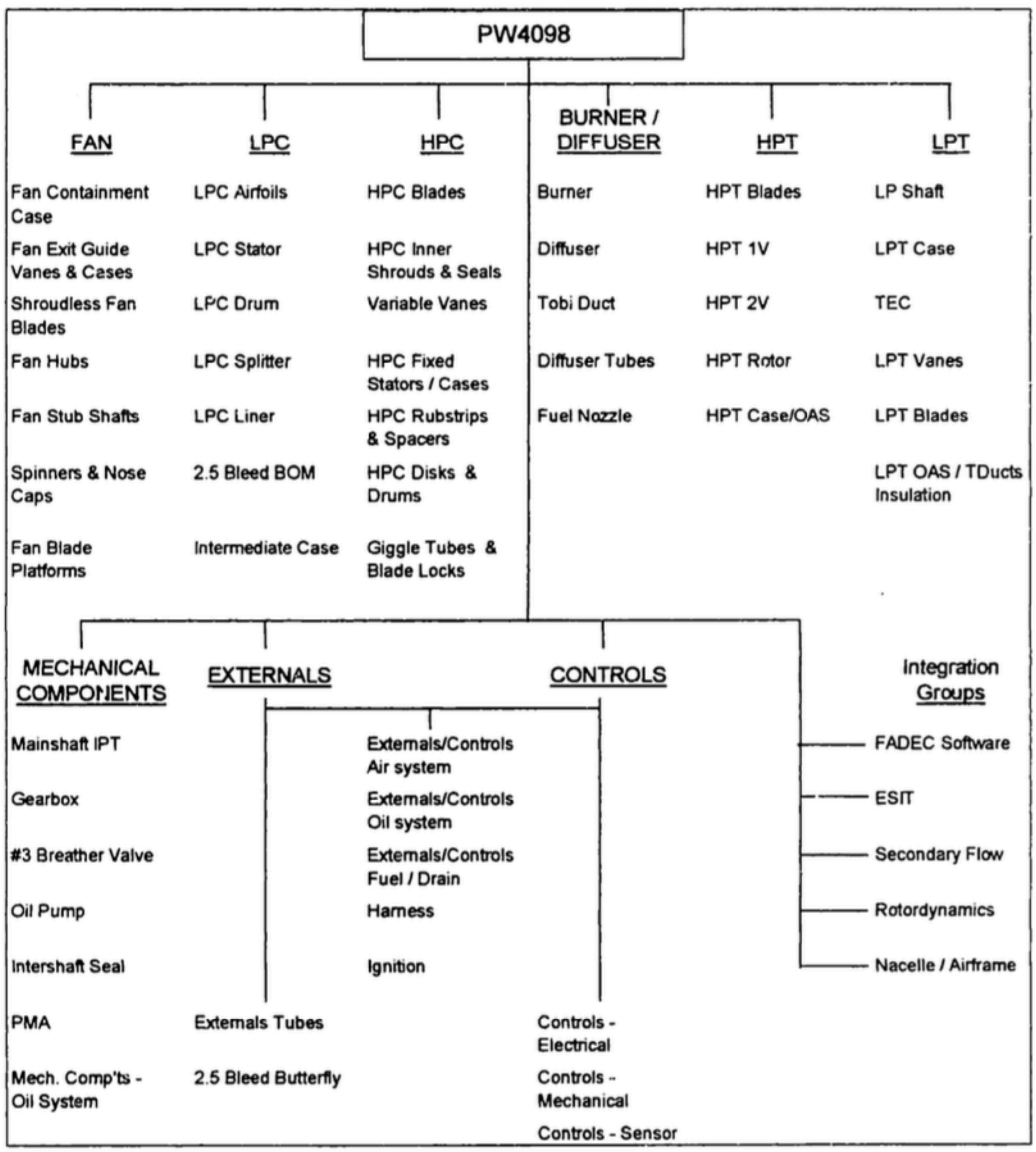
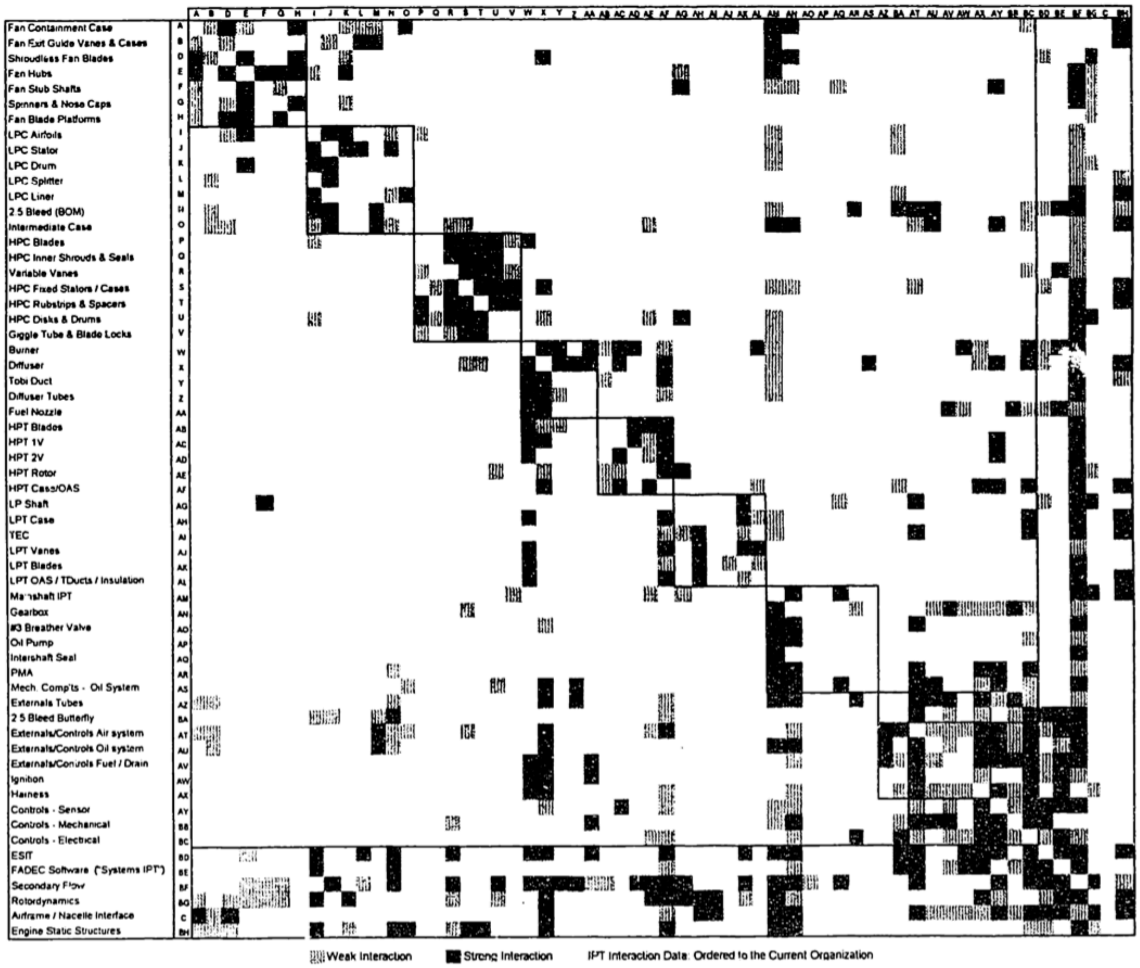


*Figure 22 - Design Structure Matrix Showing Strong and Weak Interactions Among Elements of the PW4098 Design Organization (Rowles, 1999)*




<u>Modeling the PW4098 Design Organization</u>. Two models extend the information exchange and smart team artifact models to the PW4098 organization. The design structure matrix of Figure 22 illustrates cross-functional relationships among teams. Models assume interactions occur between individual team members, thus the models modify the organizational structure shown in Figure 21 by adding five team members to each organization. Models allow testing of all links or only strong links.

# CHAPTER 4

# FINDINGS

Chapter 4 presents findings resulting from the implementation and analysis of models of organizational networks, and follows the phased approach outlined in Chapter 3. Appendices 1 through 4 supplement the sections that follow by providing detailed results in tabular and graphical formats, copies of model code, and evidence obtained during model verification and validation.

## Information Exchange Networks

Development of information exchange networks followed four main steps: implementation of the DWS information exchange model in NetLogo for random and multiscale networks; validation of the NetLogo model against a model implemented in MATLAB; extension of the information exchange model to include matrix and military staff organizational networks; and extension of the information exchange model to incorporate the effect of complexity. A series of NetLogo models was developed, culminating in two final products. The first, *Information Exchange, Version 1*, implements the DWS information exchange model for random, and multiscale networks and then extends the model to matrix and military staff organizational networks. The second, *Information Exchange, Version 2*, extends the information exchange model to incorporate the effect of complexity.

Verification and Validation. Verification and validation of the information exchange models focused on cross-validation of the information exchange model implemented using NetLogo against a model implemented in MATLAB. A multilevel, full factorial experiment compared results obtained NetLogo and MATLAB for multiscale and random networks. Table 2 summarizes the experimental design, while Table 3 summarizes analysis of variance results obtained using a general linear model in Minitab. Analysis of variance results indicate software, i.e., NetLogo or MATLAB, is not a significant factor affecting maximum congestion centrality. This result suggests successful cross-validation of the NetLogo implementation of the DWS model. Analysis of variance results also indicate network type and number of team links *are* significant factors affecting maximum congestion centrality. This result confirms results presented by Dodds, Watts and Sabel, which showed maximum congestion centrality decreased with increasing numbers of team links added, with multiscale networks decreasing much sooner, i.e., with fewer team links added.

*Table 2 - Multilevel Factorial Design for Validating the Information Exchange Model*

| | |
|---|---|
| Experiment Type | Multilevel, full factorial |
| Response Variable | Maximum Congestion Centrality, $\rho_{max}$ |
| Factors and Levels | Software Used: NetLogo or MATLAB<br>Network Type: Multiscale or Random<br>Team Links Added: 1, 4, 6, 9, 14, 22, 35, 55, 86, 136, 216, 341 |
| Replicates | 10<br>Basis: detect a $\rho_{max}$ difference of 0.1 with a confidence of 0.95 and target power of 0.9 using standard deviation estimates for $\rho_{max}$ obtained from preliminary investigations. |

*Table 3 - Analysis of Variance Results for Information Exchange Validation Experiment*

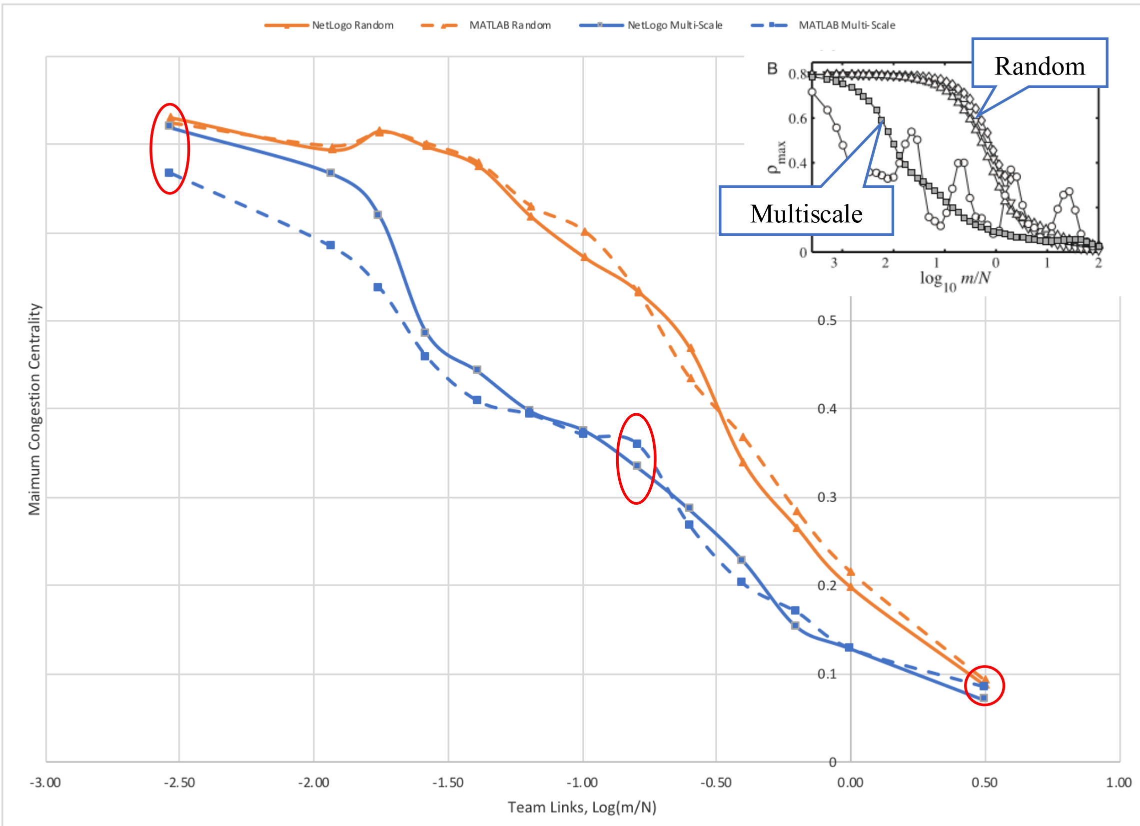
| Factor | Degrees of Freedom | F-Value | P-Value |
|---|---|---|---|
| Software | 1 | 2.02 | 0.156 |
| Network Type | 1 | 678.2 | 0.000 |
| Team Links Added | 12 | 406.6 | 0.000 |
| Interactions | | | |
| Software-Network Type | 1 | 9.12 | 0.003 |
| Software-Team Links | 12 | 1.75 | 0.053 |
| Network-Team Links | 12 | 24.42 | 0.000 |

Figure 23 summarizes results from the validation experiment, plotting maximum congestion centrality versus number of team links added, as log m/N, for random and multiscale networks as implemented in both NetLogo and MATLAB. Hypothesis testing confirmed NetLogo maximum congestion centralities for random networks equaled MATLAB maximum congestion centralities across all points tested. Hypothesis testing also confirmed NetLogo maximum congestion centralities for multiscale networks equaled MATLAB maximum congestion centralities for the majority of points tested, with exceptions circled in red.

In addition, maximum congestion centrality demonstrated behavior similar to that observed by Dodds, Watts and Sabel (see inset at figure). In particular, multiscale networks exhibited an earlier decrease in maximum congestion centrality as team links were added, which is a key feature of multiscale networks. Note that the results presented by Dodds, Watts and Sabel were for a larger network, thus precise values obtained from the validation experiment are not expected to equal those presented by Dodds, Watts and Sabel.

In summary, validation experiment results confirm proper implementation of the DWS model in NetLogo. The model is considered valid for further development to evaluate and compare the performance of organizational networks.

*Figure 23 - Comparison of NetLogo and MATLAB Results for Random and Multiscale Networks (Note: inset shows results presented in (Dodds, Watts, & Sabel, 2003))*




Evaluation of congested node data yielded an unexpected result in that multiscale networks tend to push the congested node lower in the hierarchy and do so with fewer team links than random networks. Table 4 compares depths of congested nodes for random and multiscale networks, showing the number of congested nodes at each depth. Table 5 compares maximum congestion centralities for congested nodes at different depths. Deeper congested nodes tend to have lower congestion centralities, which drives the average for any number of team links, *m*, to a lower value. This result suggests decentralization of congestion is a significant factor for improving network performance. It further suggests that even within multiscale networks, networks with particular configurations that decentralize congestion will out-perform other networks.

*Table 4 - Comparison of Depth of Congested Nodes for Random and Multiscale Networks*

| Team Links Added | | Multiscale | | | Random | | |
|---|---|---|---|---|---|---|---|
| m | log m/N | 0 | 1 | 2 | 0 | 1 | 2 |
| 1 | -2.53 | 10 | | | 10 | | |
| 4 | -1.93 | 10 | | | 10 | | |
| 6 | -1.75 | 9 | 1 | | 10 | | |
| 9 | -1.58 | 6 | 4 | | 10 | | |
| 14 | -1.39 | 1 | 9 | | 10 | | |
| 22 | -1.19 | 2 | 8 | | 10 | | |
| 35 | -0.99 | 1 | 9 | | 10 | | |
| 55 | -0.79 | 1 | 9 | | 9 | 1 | |
| 86 | -0.60 | | 10 | | 9 | 1 | |
| 136 | -0.40 | | 10 | | 7 | 3 | |
| 216 | -0.20 | | 8 | 2 | 2 | 8 | |
| 341 | 0.00 | | 2 | 8 | | 9 | 1 |
| 1079 | 0.50 | | 1 | 9 | | 2 | 8 |

*Table 5 – Comparison of Maximum Congestion Centralities for Congested Nodes for Random and Multiscale Networks*

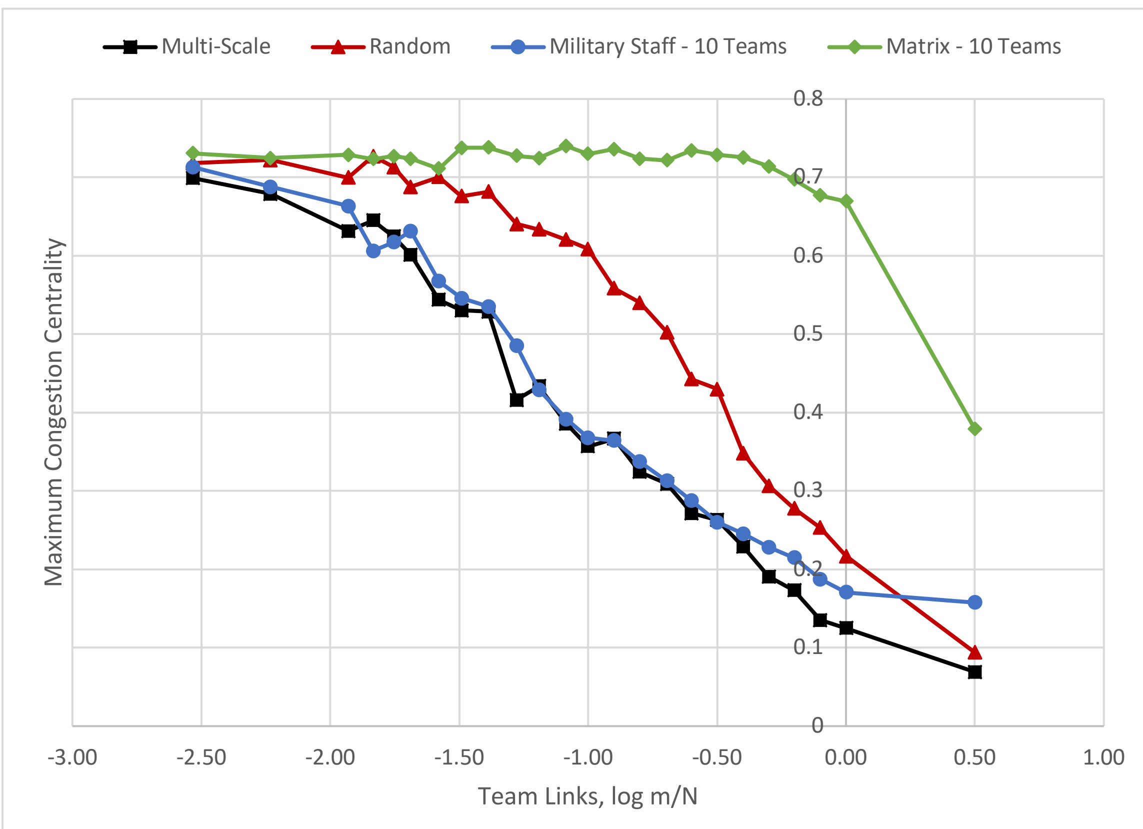
| Team Links Added | | Multiscale | | | Random | | |
|---|---|---|---|---|---|---|---|
| m | log m/N | 0 | 1 | 2 | 0 | 1 | 2 |
| 1 | -2.53 | 0.7189 | | | 0.7305 | | |
| 4 | -1.93 | 0.6663 | | | 0.6939 | | |
| 6 | -1.75 | 0.6375 | 0.4436 | | 0.7154 | | |
| 9 | -1.58 | 0.5336 | 0.4131 | | 0.6976 | | |
| 14 | -1.39 | 0.6574 | 0.4184 | | 0.6757 | | |
| 22 | -1.19 | 0.4431 | 0.3858 | | 0.6183 | | |
| 35 | -0.99 | 0.5000 | 0.3612 | | 0.5718 | | |
| 55 | -0.79 | 0.3505 | 0.3317 | | 0.5597 | 0.3010 | |
| 86 | -0.60 | | 0.2856 | | 0.4867 | 0.3129 | |
| 136 | -0.40 | | 0.2275 | | 0.3648 | 0.2838 | |
| 216 | -0.20 | | 0.1552 | 0.1431 | 0.2596 | 0.2673 | |
| 341 | 0.00 | | 0.1337 | 0.1272 | | 0.2062 | 0.1347 |
| 1079 | 0.50 | | 0.0673 | 0.0705 | | 0.0882 | 0.0878 |

Matrix and Military Staff Organizational Networks. *Information Exchange, Version 1*, extends the DWS model to matrix and military staff organizational networks by altering the network construction algorithm. A NetLogo Behavior Space experiment characterized the performance of information exchange networks. Behavior Space is a NetLogo feature that allows the user to vary input parameters and run experiments in a batch-wise manner, randomizing the order in which runs are performed. Table 6 summarizes the experimental design and Figure 24 summarizes the results, plotting maximum congestion centrality for each network type against the number of team links added, as log m/N.

*Table 6 - Summary of Experimental Design for Characterizing the Performance of Information Exchange Networks*

| | |
|---|---|
| Experiment Type | Randomized NetLogo Behavior Space |
| Response Variable | Maximum Congestion Centrality, $\rho_{max}$ |
| Factors and Levels | Network Type: Random, Multiscale, Matrix or Military Staff<br>Number of Teams (for Matrix and Military Staff): 5 or 10<br>Number of Team Links Added, *m*: 1, 2, 4, 5, 6, 7, 9, 11, 14, 18, 22, 28, 34, 43, 54, 69, 86, 108, 136, 171, 215, 271, 341, 1078 |
| Fixed Inputs | RFI Arrival Rate: 10<br>Worker Capacity: 10 |
| Additional Data | Congested Node, RFI Completion Rate, Mean RFI Age, Mean Path Length, and Global Clustering Coefficient |
| Replicates | 10 |
| NOTE: RFI refers to "requests for information," the name given in the model to messages passed within the network. | |

*Figure 24 - Comparison of Maximum Congestion Centrality for all Information Exchange Networks*



Two results stand out. First, matrix networks performed poorly, showing no meaningful reduction in maximum congestion centrality until a large number of team links had been added. Second, military staff networks performed well, comparable to multiscale networks. In fact, hypothesis testing confirmed military staff networks had maximum congestion centralities equal to multiscale networks for the majority of points tested. Note that Figure 24 shows military staff and matrix results for scenarios where 10 teams were added as both military staff and matrix networks tended to perform better when 10 teams were added compared to 5.

Evaluating the Impact of Complexity. *Information Exchange, Version 2*, extends the DWS model to account for the effect of complexity. A NetLogo Behavior Space experiment characterized the performance of information exchange networks at low, moderate and high complexity. Table 7 summarizes the experimental design.

*Table 7 - Summary of Experimental Design for Characterizing the Performance of Information Exchange Networks at Low, Moderate and High Complexity*

| | |
|---|---|
| Experiment Type | Randomized NetLogo Behavior Space |
| Response Variable | Maximum Congestion Centrality, $\rho_{max}$ |
| Factors and Levels | Network Type: Random, Multiscale, Matrix or Military Staff<br>Number of Teams (for Matrix and Military Staff): 5 or 10<br>Complexity: Low, Medium or High<br>Number of Team Links Added, *m*: 1, 2, 4, 5, 6, 7, 9, 11, 14, 18, 22, 28, 34, 43, 54, 69, 86, 108, 136, 171, 215, 271, 341, 1078 |
| Fixed Inputs | RFI Arrival Rate: 10<br>Worker Capacity: 10 |
| Additional Data | Congested Node, RFI Completion Rate, Mean RFI Age, Mean Path Length, and Global Clustering Coefficient |
| Replicates | 10 |

Figure 25 compares maximum congestion centralities for all information exchange networks at low complexity, while Figure 26 compares maximum congestion centralities for all information exchange networks at high complexity, plotting maximum congestion centrality versus number of team links added, as log m/N. At low complexity, all information exchange networks have low maximum congestion centralities that vary over a relatively narrow range as team links are added. At high complexity, maximum congestion centralities vary over a greater range, with greater differences between the performance of different organizational networks.

Multiscale networks perform well across the range of team links added. Military staff organizational networks also perform well, and have maximum congestion centralities comparable to multiscale networks over a range of team links added from -2.5 to approximately -0.8 as log m/N. Interestingly, multiscale and military staff organizational networks diverge as more team links are added, with military staff networks converging with random networks as $m$ tends to $N$. Hypothesis testing confirms the equality of maximum congestion centralities in the ranges indicated in Figure 26.

Congested node results indicate multiscale and military staff organizational networks push the congested node down the hierarchy, with multiscale networks achieving this affect with fewer team links added. Table 8 shows the number of congested nodes at each level of the hierarchy over the range of team links added. Multiscale networks decentralize congestion more quickly, and the faster and more extensive decentralization in multiscale networks as $m$ tends to $N$ helps to explain why the maximum congestion centrality of military staff organizational networks diverges from multiscale networks in this range.

*Figure 25 - Maximum Congestion Centrality for All Information Networks at Low Complexity*

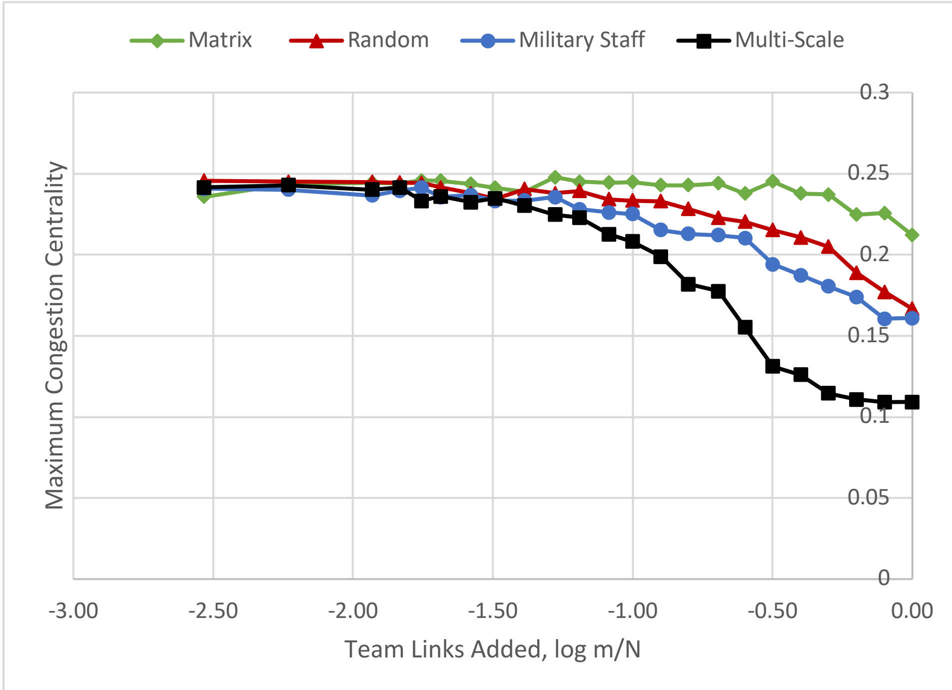



*Figure 26 - Maximum Congestion Centrality of All Information Exchange Networks at High Complexity*

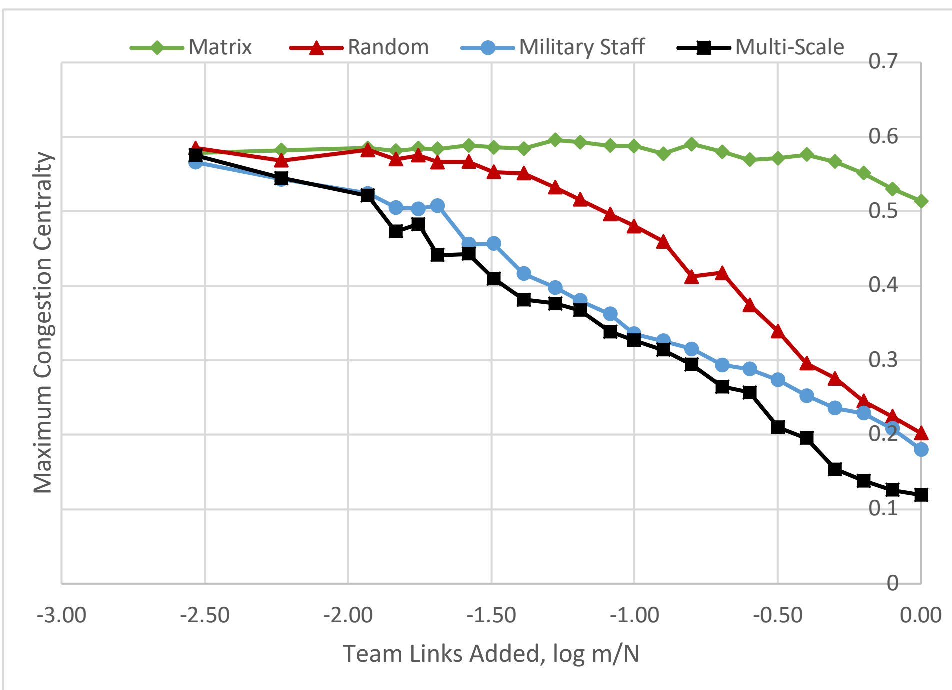


*Table 8 – Depth of Congested Nodes for Multiscale and Military Staff Organizational Networks at High Complexity – Number of Congested Nodes at Each Level*

| Team Links | | Military Staff | | | Multiscale | | |
|---|---|---|---|---|---|---|---|
| m | log m/N | 0 | 1 | 2 | 0 | 1 | 2 |
| 1 | -2.53 | 10 | | | 10 | | |
| 2 | -2.23 | 10 | | | 10 | | |
| 4 | -1.93 | 10 | | | 10 | | |
| 5 | -1.83 | 10 | | | 9 | 1 | |
| 6 | -1.75 | 10 | | | 9 | 1 | |
| 7 | -1.69 | 10 | | | 8 | 2 | |
| 9 | -1.58 | 10 | | | 8 | 2 | |
| 11 | -1.49 | 10 | | | 6 | 4 | |
| 14 | -1.39 | 9 | 1 | | 5 | 5 | |
| 18 | -1.28 | 6 | 4 | | 3 | 7 | |
| 22 | -1.19 | 5 | 5 | | 6 | 4 | |
| 28 | -1.09 | 5 | 5 | | 1 | 9 | |
| 34 | -1.00 | 4 | 6 | | | 10 | |
| 43 | -0.90 | 1 | 9 | | | 10 | |
| 54 | -0.80 | 1 | 9 | | | 10 | |
| 69 | -0.69 | | 10 | | | 10 | |
| 86 | -0.60 | | 10 | | | 10 | |
| 108 | -0.50 | | 10 | | | 10 | |
| 136 | -0.40 | | 10 | | | 10 | |
| 171 | -0.30 | | 10 | | | 6 | 4 |
| 215 | -0.20 | | 10 | | | 4 | 6 |
| 271 | -0.10 | | 10 | | | 1 | 9 |
| 341 | 0.00 | | 9 | 1 | | | 10 |

Artifact Networks

Development of artifact networks followed three major steps: implementation of the artifact model, verification and validation of the artifact model, and characterization of organizational network performance using the artifact model. The *Artifacts Network* model implements the artifact processing model for random, multiscale, matrix and military staff organizational networks.

Verification and Validation. Verification and validation of the *Artifacts Network* model focused on changes from the information exchange model and confirmation of network behavior. A multi-level, full factorial experiment compared the artifact processing performance of random, multiscale, matrix and military staff organizational networks at low and high complexity. Table 9 summarizes the experimental design, while Table 10 summarizes analysis of variance results obtained using a general linear model in Minitab. Analysis of variance results indicate network type, complexity and number of team links added are all significant factors affecting artifact completion rate. Figure 27 summarizes validation experiment results and compares artifact completion rates for all networks at low and high complexity, plotting artifact completion rates versus team links added, as log m/N. At low complexity, all networks exhibit similar performance, with completion rates approaching unity. At high complexity, all networks have completion rates less than unity across the range of team links evaluated. Results indicate all networks experience congestion failure at high complexity.

In summary, the validation experiment confirms that different organizational networks behave differently at low and high complexity. The model is considered valid for comparing the behavior of organizational networks.

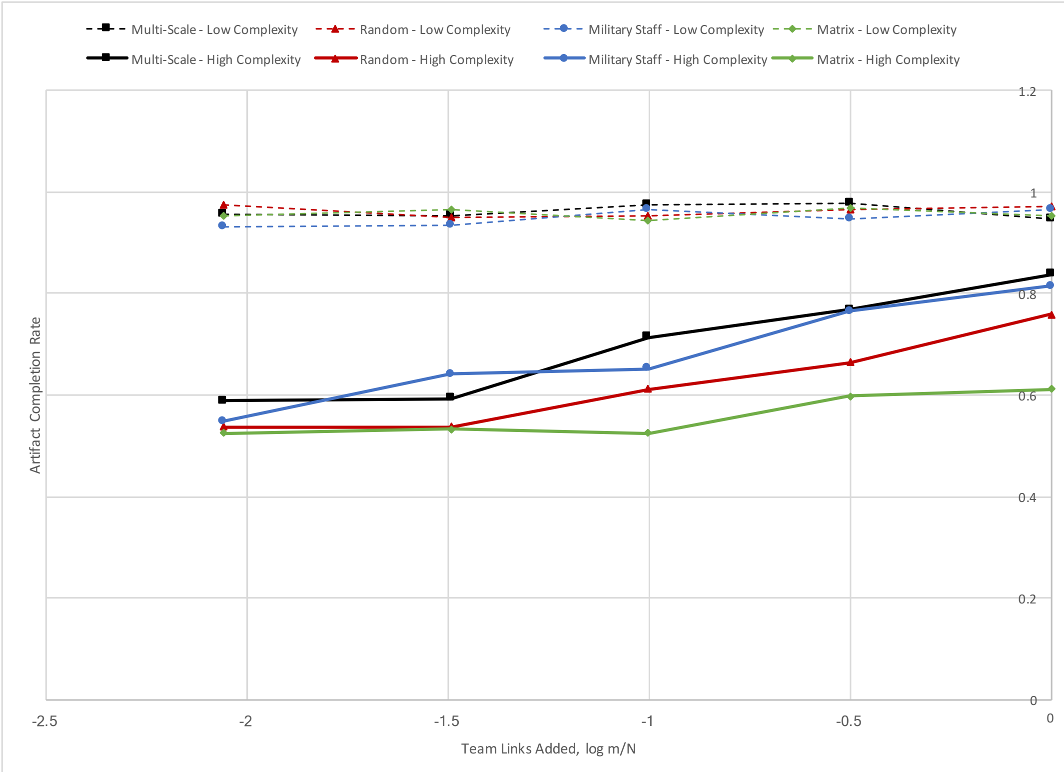
*Table 9 - Multilevel Factorial Design for Evaluating and Validating the Artifacts Network Model*

| | |
|---|---|
| Experiment Type | Multilevel, full factorial |
| Response Variable | Artifact Completion Rate |
| Factors and Levels | Network Type: Random, Multiscale, Matrix or Military Staff<br>Complexity: Low or High<br>Team Links Added: 3, 11, 34, 108, 341 |
| Replicates | 4<br>Basis: detect an artifact completion rate difference of 0.1 with a confidence of 0.95 and target power of 0.9 using standard deviation estimate obtained from preliminary results. |

*Table 10 - Analysis of Variance Results for Artifacts Network Experiment*

| Factor | Degrees of Freedom | F-Value | P-Value |
|---|---|---|---|
| Network Type | 3 | 9.99 | 0.000 |
| Complexity | 1 | 966.79 | 0.000 |
| Team Links Added | 4 | 15.57 | 0.000 |
| Interactions | | | |
| Network-Complexity | 3 | 24.56 | 0.000 |
| Network-Team Links | 12 | 2.27 | 0.012 |
| Complexity-Team Links | 4 | 30.06 | 0.000 |

*Figure 27 - Comparison of Artifact Completion Rates for All Networks at Low and High Complexity (Validation Experiment)*



<u>Characterization of Artifact Networks</u>. A NetLogo Behavior Space experiment further characterized the performance of random, multiscale, matrix and military staff organizational networks at low, moderate and high complexities. Table 11 summarizes the experimental design. At low and moderate complexity, all organizational networks perform well, with artifact completion rates approaching unity. However, at high complexity, all organizational networks experience congestion failure. Figure 28 compares artifact completion rates for all organizational networks at high complexity. Multiscale and military staff organizational networks out-perform random and matrix networks. Hypothesis testing confirmed multiscale and military staff organizational networks achieve equal artifact completion rates across a broad range of team links added but diverge in the neighborhood of log m/N = -0.5. By comparison, matrix organizations perform poorly, showing no improvement in artifact completion rates until a relatively large number of team links has been added.

*Table 11 - Artifact Network Characterization Experimental Design*

| | |
|---|---|
| Experiment Type | Randomized NetLogo Behavior Space |
| Response Variable | Artifact Completion Rate |
| Factors and Levels | Network Type: Random, Multiscale, Matrix or Military Staff<br>Complexity: Low, Moderate or High<br>Number of Team Links Added, *m*: 1, 2, 4, 5, 6, 7, 9, 11, 14, 18, 22, 28, 34, 43, 54, 69, 86, 108, 136, 171, 215, 271, 341 |
| Fixed Inputs | Artifact Arrival Rate: 10<br>Worker Capacity: 10<br>Number of Teams: 10<br>Artifact Rating: 0.5 |
| Additional Data | RFI Arrival and Completion Rates; RFI, Artifact and Effective Congestion Centralities and Congested Nodes; and Mean Path Length and Global Clustering Coefficient |
| Replicates | 10 |

*Figure 28 - Artifact Completion Rates for All Organizational Networks at High Complexity*

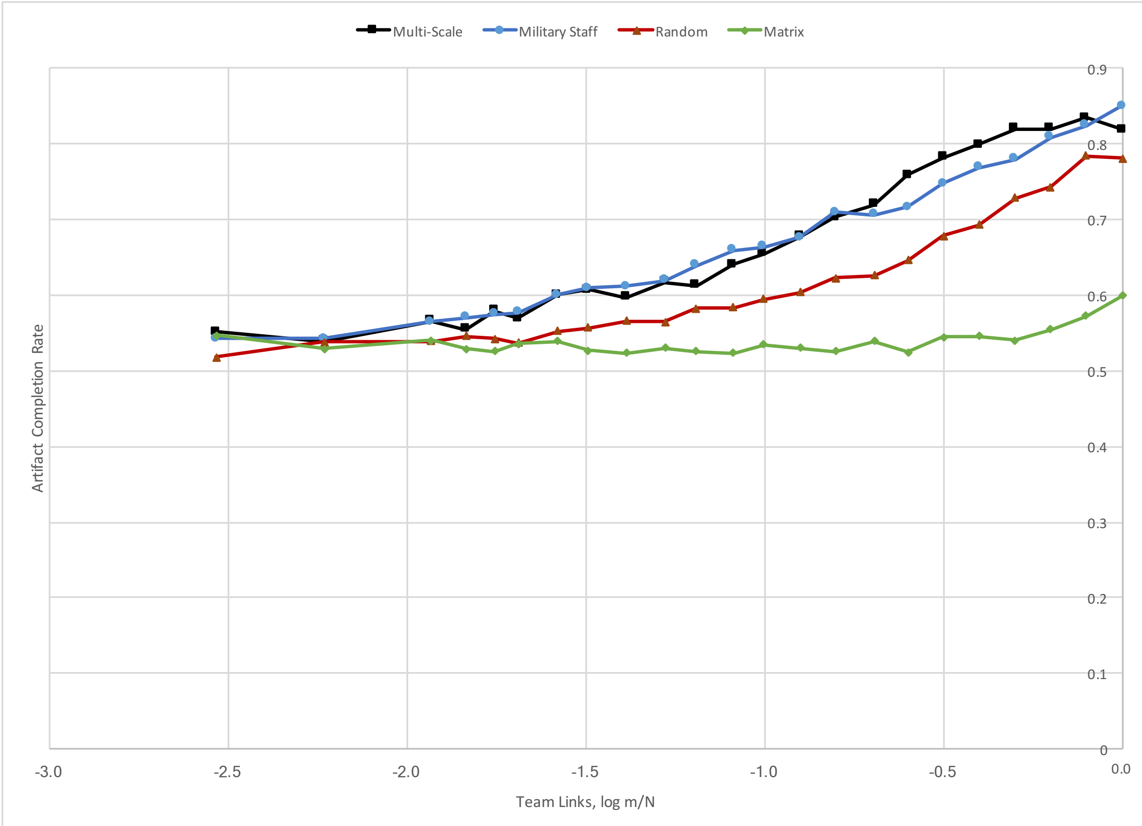



Figure 29 compares effective and RFI congestion centralities for all organizational networks at high complexity, and Figure 30 compares artifact, RFI and effective congestion centralities for multiscale networks at high complexity to demonstrate the interaction among them. Effective congestion centrality exhibits behavior similar to that seen for maximum congestion centrality in information exchange networks in that it decreases as the number of team links added increases, with a sharp decrease across a narrow range of team links added. Multiscale networks exhibit this decrease sooner, that is with fewer team links added, than other organizational networks.

This result is also comparable to that seen in information exchange networks. Military staff organizational networks do not exhibit the same sharp decrease in effective congestion centrality as multiscale networks, and this divergence corresponds to the divergence in in artifact completion rates described above. Still, military staff organizational networks out-perform random and matrix organizational networks.

Figure 30 demonstrates a complicated relationship among RFI, artifact and effective congestion centralities. RFI congestion centrality decreases as the number of team links added increases, while artifact congestion centrality increases as the number of team links added increases. As team links increase, networks become more effective at exchanging information. RFIs are answered more quickly, which allows artifacts to be processed more quickly, leading to an increase in artifact congestion centrality. The point at which effective congestion centrality begins to decrease rapidly corresponds to the crossover point at which RFI congestion centrality equals artifact congestion centrality, which suggests RFI congestion is the key factor leading to congestion failure in organizational networks at high complexity.

*Evaluation of congested node results confirmed multiscale and military staff organizational networks achieved decentralization of RFI congestion comparable to that seen in information exchange networks.*

Table 12 compares the depths of artifact and RFI congested nodes for multiscale and military staff artifact networks at high complexity and shows that both achieve decentralization with respect to RFIs, with multiscale networks achieving decentralization more quickly, and to a greater extent, than military staff networks. Note that the artifact congested node is always at level 1 because all artifacts must be approved by the manager at level one, thus the manager is the natural congestion point for artifacts.

*Figure 29 - Effective and RFI Congestion Centralities for All Organizational Networks at High Complexity*

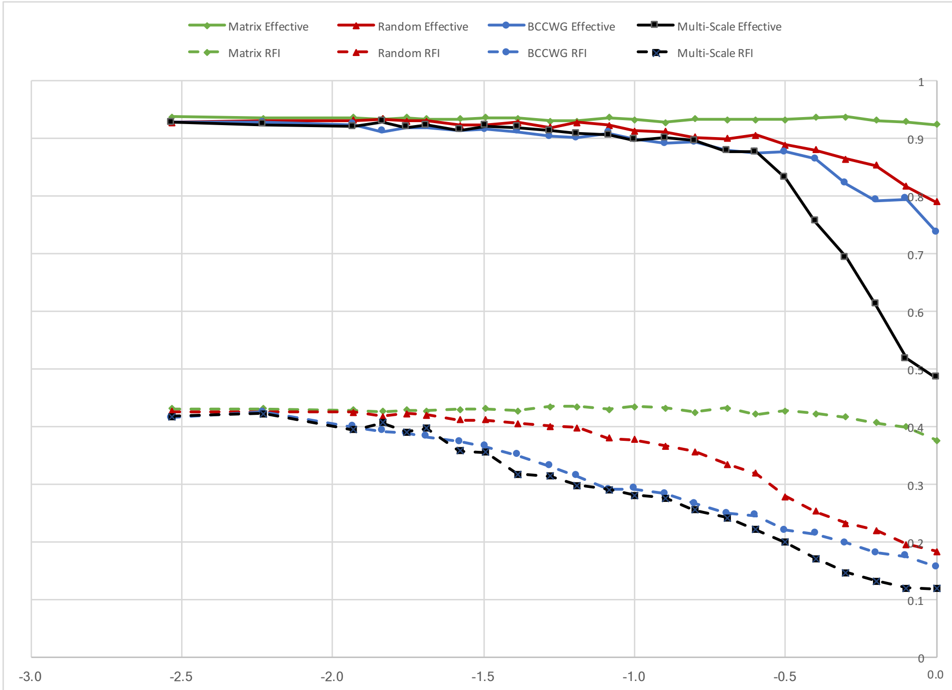



*Figure 30 - Comparison of Artifact, RFI and Effective Congestion Centralities for Multiscale Networks at High Complexity*

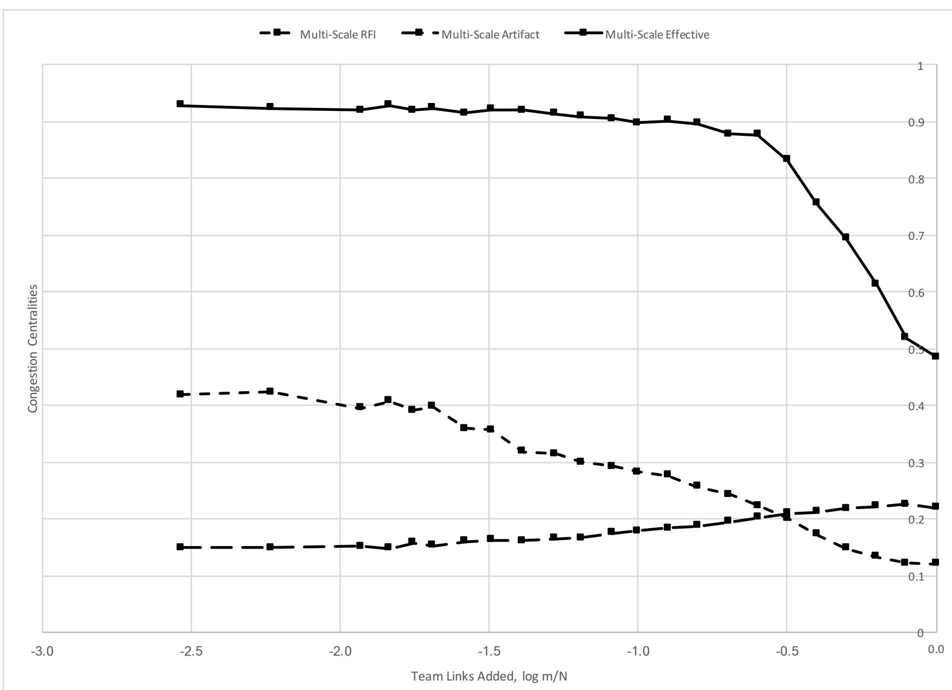


*Table 12 - Depths of Artifact and RFI Congested Nodes in Multiscale and Military Staff Artifact Networks at High Complexity*

| Team Links | | Multi-Scale Networks | | | | Military Staff Networks | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Artifact Congested Node | RFI Congested Node | | | Artifact Congested Node | RFI Congested Node | | |
| m | log m/N | 1 | 0 | 1 | 2 | 1 | 0 | 1 | 2 |
| 1 | -2.53 | 10 | 10 | | | 10 | 10 | | |
| 2 | -2.23 | 10 | 10 | | | 10 | 10 | | |
| 4 | -1.93 | 10 | 10 | | | 10 | 10 | | |
| 5 | -1.83 | 10 | 10 | | | 10 | 10 | | |
| 6 | -1.75 | 10 | 10 | | | 10 | 10 | | |
| 7 | -1.69 | 10 | 10 | | | 10 | 10 | | |
| 9 | -1.58 | 10 | 8 | 2 | | 10 | 9 | 1 | |
| 11 | -1.49 | 10 | 8 | 2 | | 10 | 10 | | |
| 14 | -1.39 | 10 | 4 | 6 | | 10 | 9 | 1 | |
| 18 | -1.28 | 10 | 2 | 8 | | 10 | 8 | 2 | |
| 22 | -1.19 | 10 | 2 | 8 | | 10 | 6 | 4 | |
| 28 | -1.09 | 10 | 2 | 8 | | 10 | 6 | 4 | |
| 34 | -1.00 | 10 | | 10 | | 10 | 4 | 6 | |
| 43 | -0.90 | 10 | 1 | 9 | | 10 | 4 | 6 | |
| 54 | -0.80 | 10 | | 10 | | 10 | 4 | 6 | |
| 69 | -0.69 | 10 | | 10 | | 10 | | 10 | |
| 86 | -0.60 | 10 | | 10 | | 10 | 2 | 8 | |
| 108 | -0.50 | 10 | | 10 | | 10 | | 10 | |
| 136 | -0.40 | 10 | | 8 | 2 | 10 | 1 | 9 | |
| 171 | -0.30 | 10 | | 8 | 2 | 10 | 1 | 9 | |
| 215 | -0.20 | 10 | | 3 | 7 | 10 | 1 | 8 | 1 |
| 271 | -0.10 | 10 | | | 10 | 10 | | 10 | |
| 341 | 0.00 | 10 | | | 10 | 10 | | 6 | 4 |

Figure 31 illustrates the aggregate effect of complexity on artifact networks, plotting artifact completion rate against complexity for all organizational networks. It shows all organizational networks perform well at low and moderate complexity but experience a sharp decrease in artifact completion rate as complexity increases to high. This sort of tipping point behavior is commonly seen in complex systems.

*Figure 31 - Aggregate Effect of Complexity on Artifact Networks*

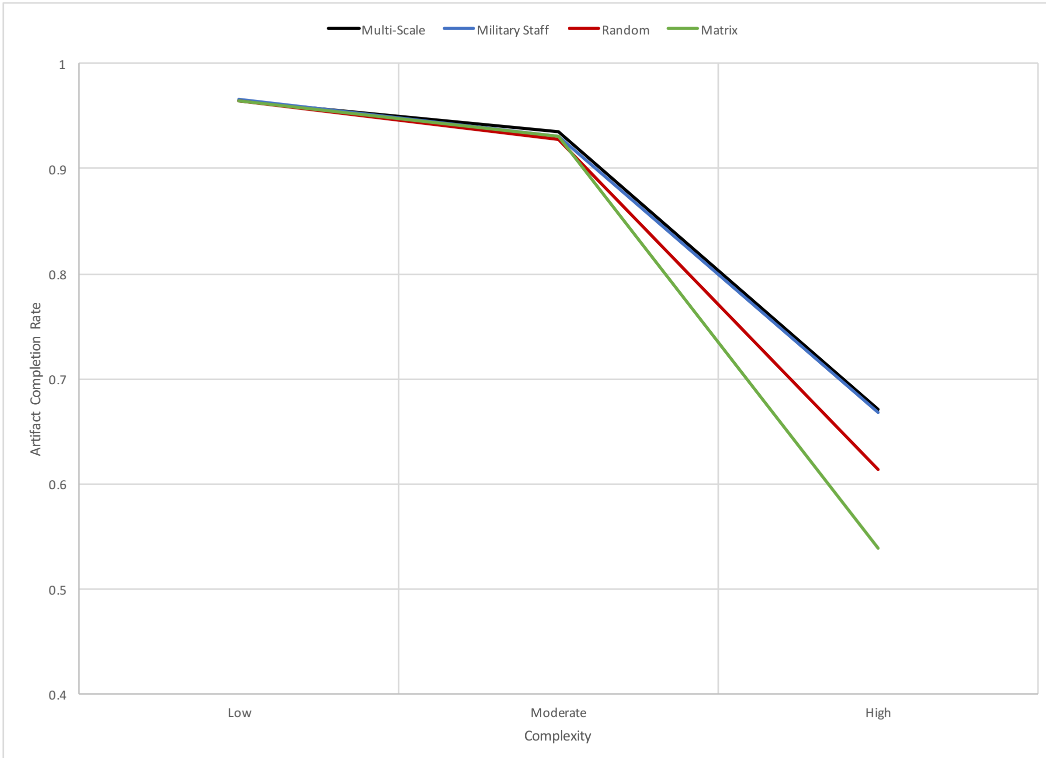
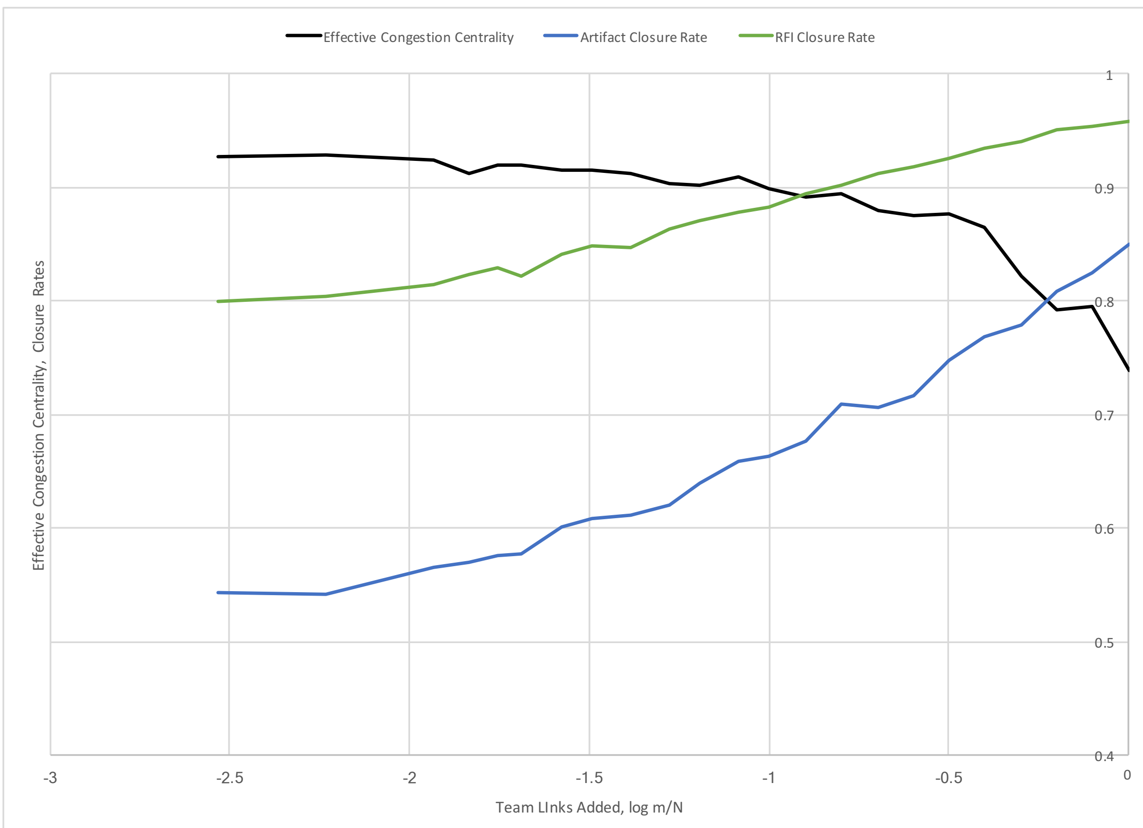



Figure 32 compares RFI and artifact closure rates to effective congestion centrality for military organizational networks at high complexity and illustrates how each change as team links are added to the network. Initially, effective congestion is relatively high and artifact closure rate is low, while RFI closure rate is relatively high. As team links added, effective congestion decreases and RFI closure rates increase, with artifact closure rates increasing in turn. As the number of team links added increases beyond -1, as log *m/N*, effective congestion centrality decreases more rapidly, RFI closure rate begins to stabilize, and artifact closure rate begins to rise more sharply.

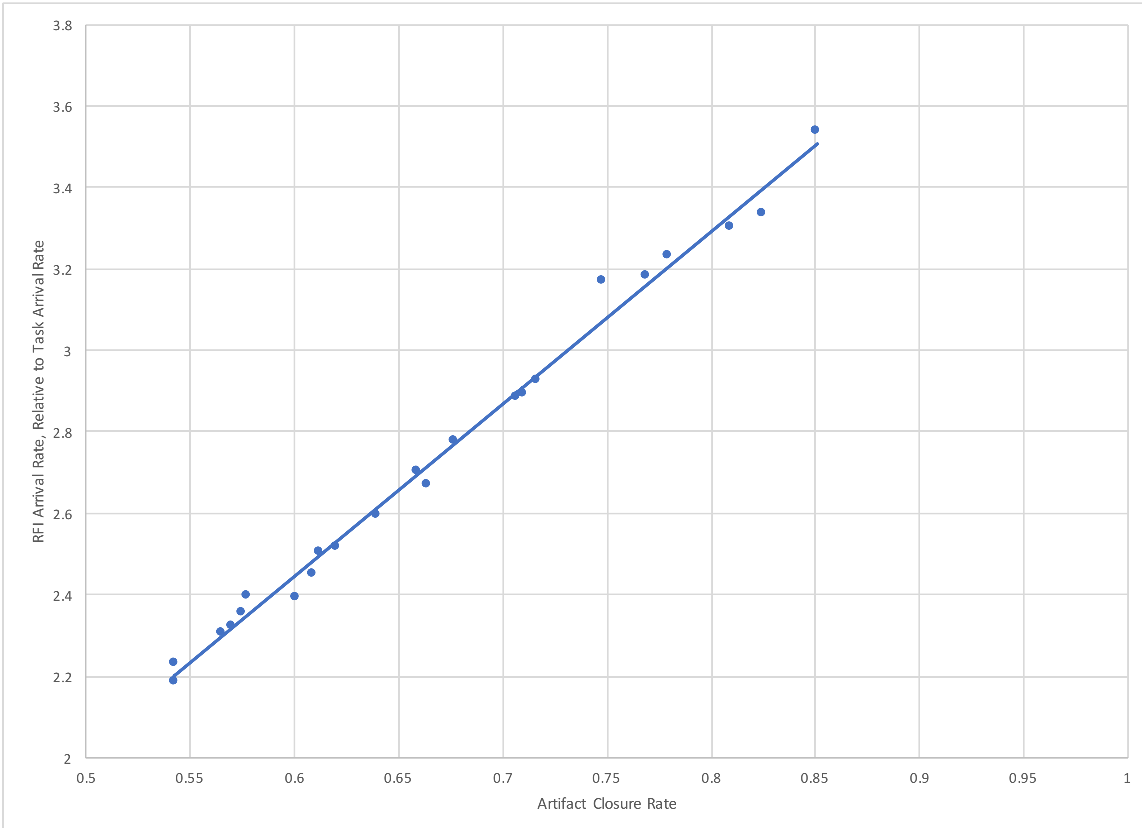
Interestingly, as artifact closure rate rises, so does the RFI arrival rate. Figure 33 shows that RFI arrival rates increase linearly with artifact closure rates.[3] As congestion decreases, RFIs and artifacts are processed more quickly, but faster processing of artifacts mean that more artifacts are in the system, thus there is greater likelihood RFIs will be created. This result demonstrates a positive feedback with regard to RFIs in that reduced congestion and improved processing of RFIs leads to greater demand for RFI processing.

*Figure 32 - RFI and Artifact Closure Rates with Effective Congestion Centrality for Military Organizational Networks at High Complexity*




[3] Note that RFI arrival rate is expressed in terms of task arrival rate. For example, RFI arrival rate of 2 corresponds to 20 RFIs per time interval when task arrival rate is 10.

*Figure 33 - RFI Amplification in Military Organizational Networks at High Complexity*



Smart Team Networks

Development of smart team networks followed three major steps: implementation of the smart team model, verification and validation of the model, and characterization of organizational network performance using the model. The *Smart Team Network* model implements the smart team model for random, multiscale, matrix and military staff organizational networks. Since all networks exhibit satisfactory performance at low and moderate complexity, meaning they remain free of congestion, analysis focused on network performance at high complexity.

Verification and Validation. Verification and validation of the *Smart Team Network* model focused on changes from the artifact model. A multilevel, full factorial experiment compared the artifact processing effectiveness of random, multiscale, matrix and military staff organizational networks considering the following additional factors: team size scaling factor, $\chi$, artifact preference, centralized versus decentralized approvals, and number of team links added. Table 13 summarizes the experimental design and Table 14 presents analysis of variance results. Artifact preference was not a significant factor affecting artifact completion rate, but all other factors were significant factors.

Figure 34 provides further insight by comparing artifact completion rates at different values of team links added for each network type, taking team scaling factor, artifact preference and approval method in turn. Highlighting draws attention to low and high values of artifact completion rate and demonstrates that multiscale, military staff and random networks out-performed matrix networks by a wide margin. Results also suggest matrix networks perform better for smaller values of team size scaling factor, while military staff networks perform better for larger values. In other words, matrix networks perform better when teams are small, while military staff networks perform better when teams are large.

*Figure 35 evaluates the utility of delta slope as an indicator of congestion failure, with the left-hand bar showing the range of delta slope values for congested networks and the right-hand bar showing the range of values for networks free of congestion. The two bars overlap significantly, which suggests delta slope is not an effective indicator of congestion failure. Delta slope uses only the last ten time ticks, but observation of network behavior showed networks exhibit dynamic behavior with respect to artifact processing, with relatively large swings in the difference between total number of artifacts and the number of completed artifacts, even in networks free of congestion.*

Figure 36 shows a plot of total team links minus completed team links for a multiscale network with 341 team links at high complexity. The difference varies widely over time even though the network is free of congestion, which illustrates how organizational networks exhibit interesting dynamic behaviors over time.

Delta slope could possibly be improved by using a longer ranger of time to smooth out variations, but examination of validation experiment results indicates artifact completion rate is a reliable indicator of congestion. All networks with artifact completion rates less than 0.85 were congested, while all networks with artifact completion rates greater than 0.90 were free of congestion. This suggests a simple scheme for characterizing congestion. If artifact completion rate is greater than 0.90, the network is free of congestion. If the artifact completion rate is between 0.85 and 0.90, the network is on the verge of congestion. If the artifact completion rate is less than 0.85, the network is congested.

*Table 13 - Experimental Design for Smart Team Network Validation*

| | |
|---|---|
| Experiment Type | Multilevel, full factorial |
| Response Variable | Artifact Completion Rate |

| | |
|---|---|
| Factors and Levels | Network Type: Random, Multiscale, Matrix or Military Staff<br>Team size scaling factor, $\chi$: 5 or 50<br>Artifact Preference: RFI, Balanced, or Artifact Preference<br>Approval Method: Centralized or Decentralized<br>Team Links Added: 34, 108, 341 |
| Fixed Inputs | Complexity: High<br>Artifact Arrival Rate and Worker Capacity: 10 |
| Replicates | 4<br>Basis: detect an artifact completion rate difference of 0.1 with a confidence of 0.95 and target power of 0.9 using standard deviation estimate obtained from previous results. |

*Table 14 - Analysis of Variance Results for Smart Team Network Validation Experiment*

| Factor | Degrees of Freedom | F-Value | P-Value |
|---|---|---|---|
| Network Type | 3 | 477.71 | 0.000 |
| Team Size Scaling Factor | 1 | 7.5 | 0.000 |
| Artifact Preference | 2 | 0.05 | .951 |
| Approval | 1 | 22.32 | 0.000 |
| Team Links Added | 2 | 446.24 | 0.000 |
| Interactions | | | |
| Network-Team Size | 3 | 38.92 | 0.000 |
| Network-Preference | 6 | 0.46 | 0.839 |
| Network-Approval | 3 | 4.79 | 0.003 |
| Network-Team Links | 6 | 64.02 | 0.000 |
| Team Size-Preference | 2 | 0.74 | 0.478 |
| Team Size-Approvals | 1 | 0.12 | 0.729 |
| Team Size-Team Links | 2 | 13.12 | 0.000 |
| Preference-Approval | 2 | 1.14 | 0.321 |
| Preference-Team Links | 4 | 1.05 | 0.383 |
| Approval-Team Links | 2 | 4.35 | 0.013 |

*Figure 34 - Comparison of Artifact Completion Rate Results from Smart Team Network Validation Experiment*

| Artifact Rate Comparison - Chi | | | | | | | |
|---|---|---|---|---|---|---|---|
| Team Links | | Matrix | | Military Staff | | Multi-Scale | Random |
| m | log m/N | 5 | 50 | 5 | 50 | | |
| 34 | -1.00 | 0.5787 | 0.5778 | 0.7294 | 0.7316 | 0.7323 | 0.6454 |
| 108 | -0.50 | 0.6211 | 0.5901 | 0.8047 | 0.8484 | 0.8827 | 0.7578 |
| 341 | 0.00 | 0.6866 | 0.6554 | 0.7541 | 0.9228 | 0.9330 | 0.9065 |

| Artifact Rate Comparison - Decentralized Approval | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Team Links | | Matrix | | Military Staff | | Multi-Scale | | Random | |
| m | log m/N | FALSE | TRUE | FALSE | TRUE | FALSE | TRUE | FALSE | TRUE |
| 34 | -1.00 | 0.5757 | 0.5808 | 0.7131 | 0.7480 | 0.7117 | 0.7528 | 0.6342 | 0.6567 |
| 108 | -0.50 | 0.6044 | 0.6069 | 0.8110 | 0.8421 | 0.8626 | 0.9028 | 0.7338 | 0.7817 |
| 341 | 0.00 | 0.6735 | 0.6685 | 0.8364 | 0.8405 | 0.9309 | 0.9350 | 0.8920 | 0.9211 |

| Artifact Rate Comparison - Artifact Preference | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Team Links | | Matrix | | | Military Staff | | | Multi-Scale | | | Random | | |
| m | log m/N | Artifact | Balanced | RFI | Artifact | Balanced | RFI | Artifact | Balanced | RFI | Artifact | Balanced | RFI |
| 34 | -1.00 | 0.5706 | 0.5826 | 0.5816 | 0.7238 | 0.7388 | 0.7290 | 0.7299 | 0.7315 | 0.7353 | 0.6387 | 0.6420 | 0.6556 |
| 108 | -0.50 | 0.6087 | 0.5994 | 0.6088 | 0.8325 | 0.8223 | 0.8248 | 0.8864 | 0.8810 | 0.8806 | 0.7660 | 0.7542 | 0.7532 |
| 341 | 0.00 | 0.6611 | 0.6832 | 0.6687 | 0.8266 | 0.8392 | 0.8496 | 0.9380 | 0.9304 | 0.9305 | 0.9152 | 0.9038 | 0.9006 |

*Figure 35 - Evaluation of Delta Slope as a Measure of Network Congestion*

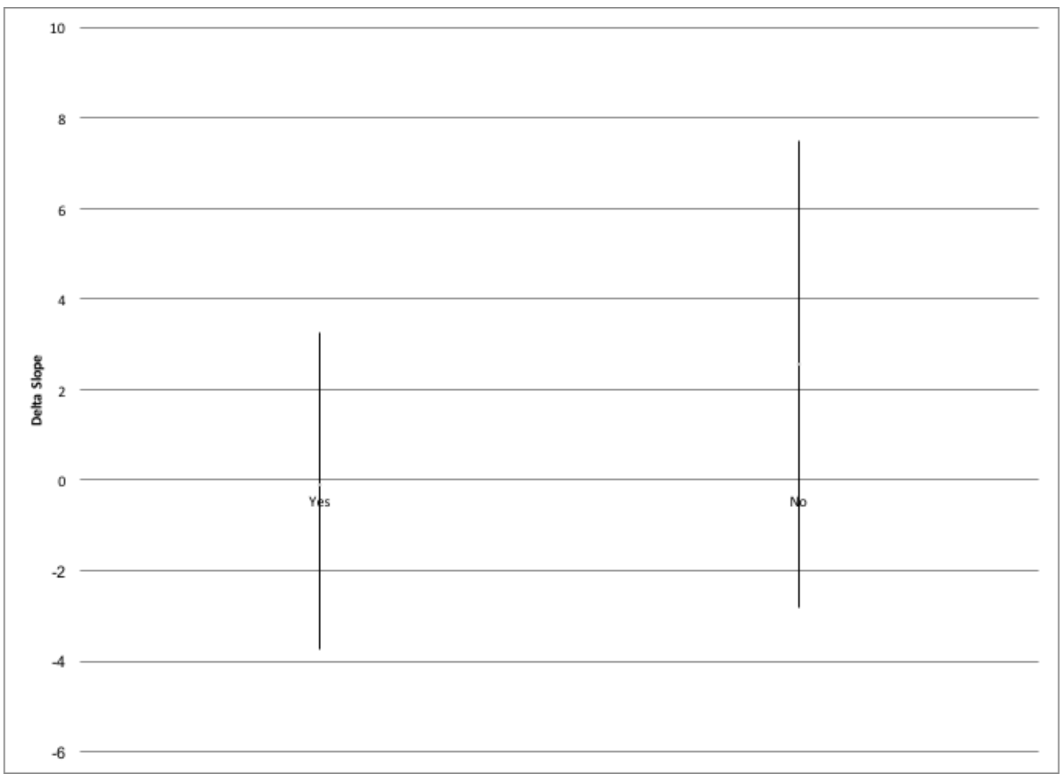
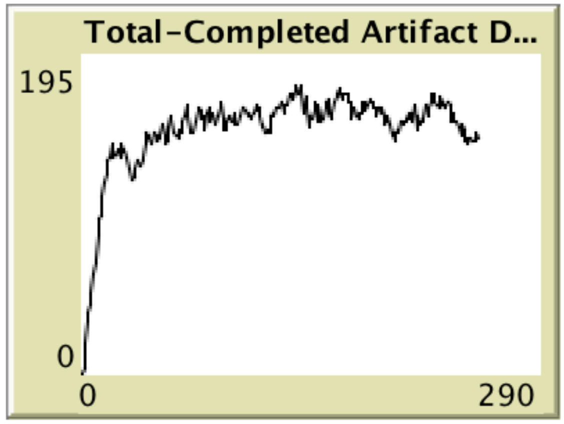



*Figure 36 - Plot of Total Artifacts Minus Completed Artifacts for a Multiscale Network at High Complexity and 341 Team Links*




Characterization of Smart Team Networks. A NetLogo Behavior Space experiment further characterized the behavior of random, multiscale, matrix and military staff organizational networks at high complexity. Table 15 summarizes the experimental design.

*Table 15 - Design of NetLogo Behavior Space Experiment to Characterize Smart Team Network Behavior*

| | |
|---|---|
| Experiment Type | Randomized NetLogo Behavior Space |
| Response Variable | Artifact Completion Rate |
| Factors and Levels | Network Type: Random, Multiscale, Matrix or Military Staff<br>Artifact Preference: RFI-Preference, Neutral, Artifact-Preference<br>$\chi$: 5 or 50<br>Decentralize Preference: Low, Moderate, or High<br>Number of Team Links Added, *m*: 3, 7, 11, 20, 34, 61, 108, 192, 341, 541, 681 |
| Fixed Inputs | Artifact Arrival Rate: 10<br>Worker Capacity: 10<br>Number of Teams: 10<br>Complexity: High |
| Additional Data | RFI Arrival and Completion Rates; RFI, Artifact and Effective Congestion Centralities and Congested Nodes; Mean Path Length and Global Clustering Coefficient; and Delta-Slope |
| Replicates | 10 |

Figure 37 compares the performance of random, multiscale, matrix and military staff organizational networks for centralized and decentralized approvals and RFI, balanced, and artifact processing preference. All organizational networks exhibit similar behavior regardless of processing preference, but decentralized approvals improve performance. Overall, decentralized networks with an RFI processing preference perform best. In this group, multiscale and military staff organizational networks exhibit robustness to congestion around log m/N = -0.5. Figure 38 compares effective congestion centrality for random, multiscale, matrix and military staff organizational networks at high complexity for centralized and decentralized approvals and demonstrates that decentralized approvals improve robustness to congestion.

*Figure 37 - Comparison of Organizational Network Performance for Centralized and Decentralized Approvals for RFI, Balance and Artifact Processing Preference*

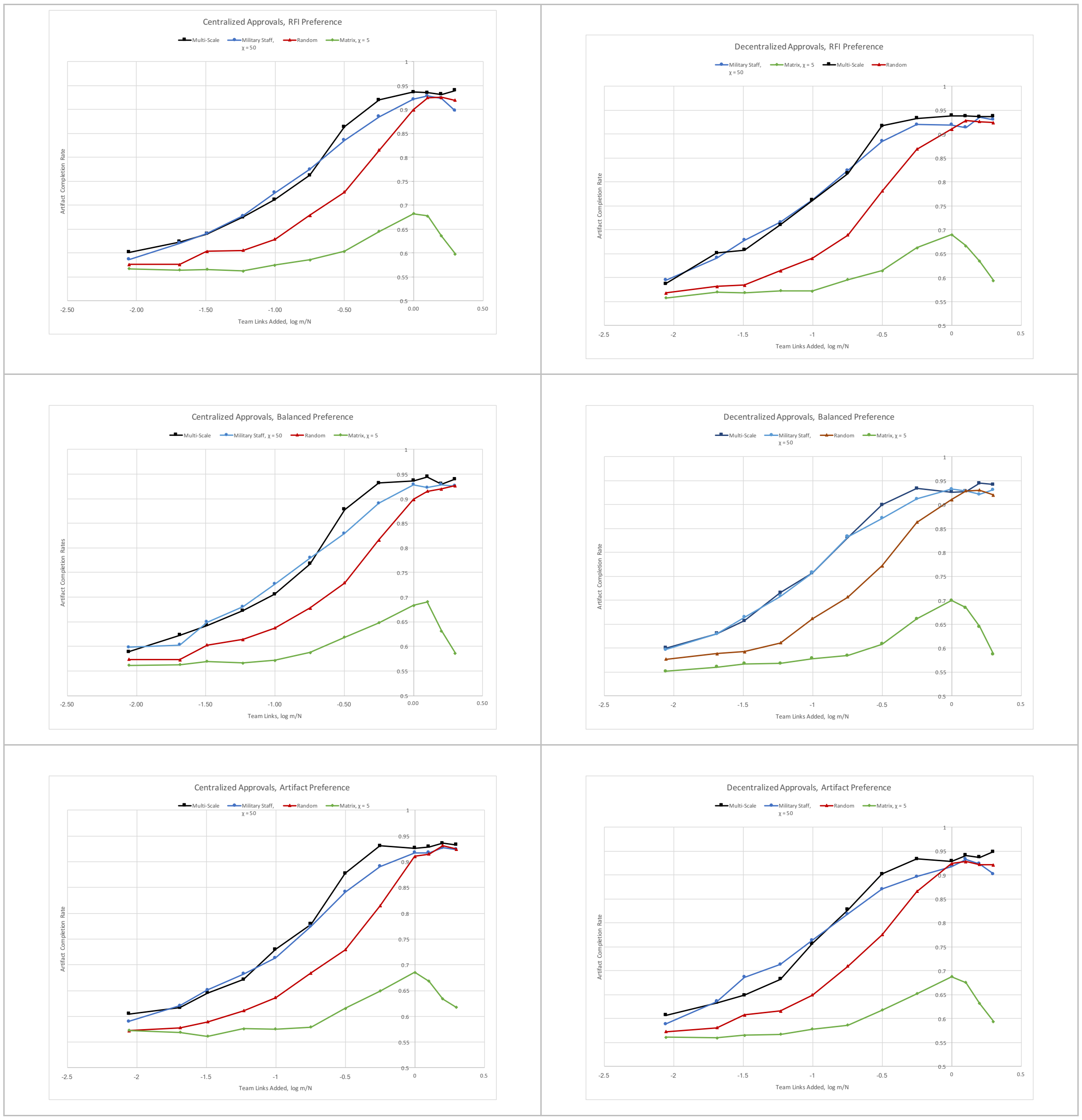
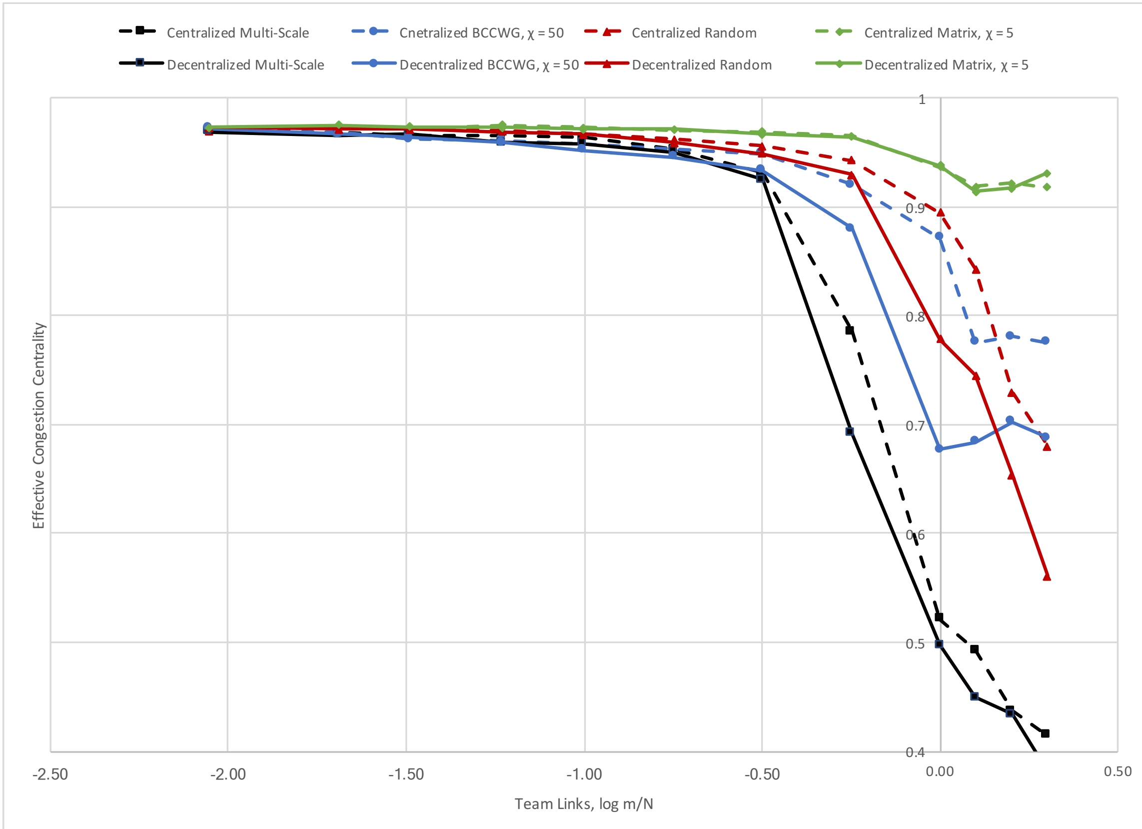


*Figure 38 - Summary of Effective Congestion Centrality for All Networks at High Complexity, Comparing Centralized and Decentralized Approvals*



The PW4098 Case Study

A NetLogo Behavior Space experiment characterized the performance of the PW4098 design organization using the information exchange and smart team artifact models. Table 16 summarizes the experimental design for performance characterization using the information exchange model, and Table 17 summarizes the experimental design for performance characterization using the artifact model.

*Table 16 - Design for Information Network Experiment on the PW4098 Design Organization*

| | |
|---|---|
| Experiment Type | Randomized NetLogo Behavior Space |
| Response Variable | Maximum Congestion Centrality |
| Factors and Levels | Team Links: Strong Links Only or All Links<br>Complexity: Low or High |
| Fixed Inputs | Network Type: PW4098 Design Organization<br>RFI Arrival Rate and Worker Capacity: 10 |
| Replicates | 10 |

*Table 17 - Design for Artifact Experiment on the PW4098 Design Organization*

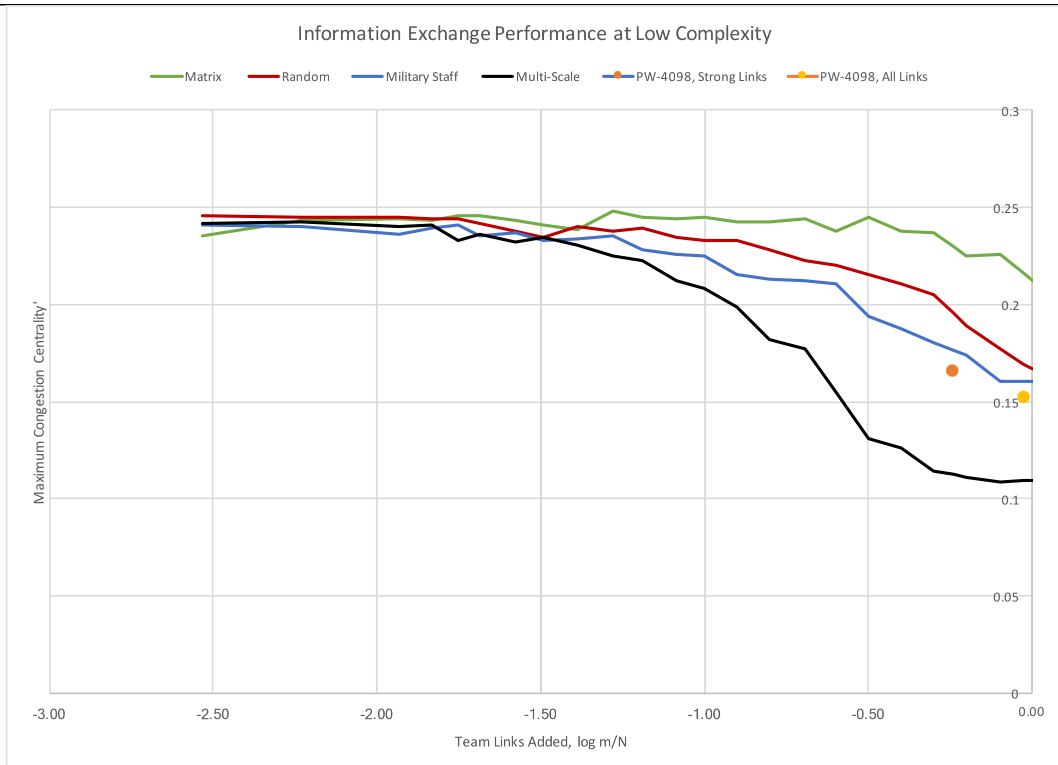
| | |
|---|---|
| Experiment Type | Randomized NetLogo Behavior Space |
| Response Variable | Artifact Completion Rate |
| Factors and Levels | Team Links: Strong Links Only or All Links<br>Artifact Preference: RFI, Balanced, or Artifact Preference<br>Complexity: Low, Moderate or High<br>Approval Method: Centralized or Decentralized |
| Fixed Inputs | Network Type: PW4098 Design Organization<br>Artifact Arrival Rate and Worker Capacity: 10<br>Decentralization Preference: 0.5 |
| Replicates | 10 |

Information Exchange Performance. Figure 39 compares the information exchange performance of the PW4098 design organization to random, multiscale, matrix, and military staff organizational networks at low and high complexity. At low complexity, the PW4098 design organization exhibits performance comparable to random and military staff organizational networks. Structurally, the PW4098 design organization is comparable to a random network. The PW4098 design organization was evaluated for only two numbers of team links added, corresponding to all links or strong links only, so full comparison to random, multiscale, matrix and military staff organizational networks across the range of team links added is not possible. At high complexity, however, the PW4098 design organization performs poorly, with maximum congestion centralities higher than other organizational networks. This result suggests the PW4098 design organization will be susceptible to congestion failure at high complexity.

Figure 40 summarizes the artifact performance of the PW4098 design organization for RFI, balanced, and artifact processing preference; centralized and decentralized approvals; and low, moderate, and high complexity. Figure 40 presents artifact completion rates with overall groups based on artifact preference, with further subdivisions for approvals and complexity. The PW4098 design organization performs well at low and moderate complexity, but experiences congestion failure at high complexity. Furthermore, the PW4098 design organization achieves artifact completion rates well below those of all other organizational networks at high complexity, which suggests the PW4098 design organization is unprepared to manage the design of a complex engineered system and is therefore susceptible to the kinds of cost and schedule overruns that often plague programs that attempt to deliver them. This result is concerning because the PW4098 design organization reflects mainstream thinking around the design of engineered systems. First, it is a matrix organization composed of cross-functional teams. In fact, Rowles reported Pratt & Whitney had abandoned functional organization, having replaced them with discipline centers to maintain technical expertise. Second, Rowles characterized the PW4098 design organization as a heavyweight project matrix organization, which is wholly consistent with mainstream project management practice. (Rowles, 1999)

Rowles provides a key insight into the susceptibility of the PW4098 design organization to congestion failure, noting approximately one-third of integrated product team interactions occurred outside of the team's hierarchical group, and that approximately one-fourth of integrated product team interactions did not correspond to design relationships. (Rowles, 1999) The PW4098 design organization's structure implicitly assumes knowledge of the design relationships. The composition and arrangement of integrated product teams reflects these assumed relationships, but complex engineered systems are considered complex because system interactions, and therefore design relationships, are poorly understood. It is not surprising, then, that a relatively large number of interactions would occur outside a hierarchical arrangement based on known or predicted design relationships. The ability of an organizational network to withstand complexity depends on its ability to cope with these unanticipated relationships and the interactions that result. As complexity increases, these unexpected interactions become more frequent, putting strain on the organizational network.

*Figure 39 - Comparison of PW4098 Design Organization Information Exchange Performance at Low and High Complexity*



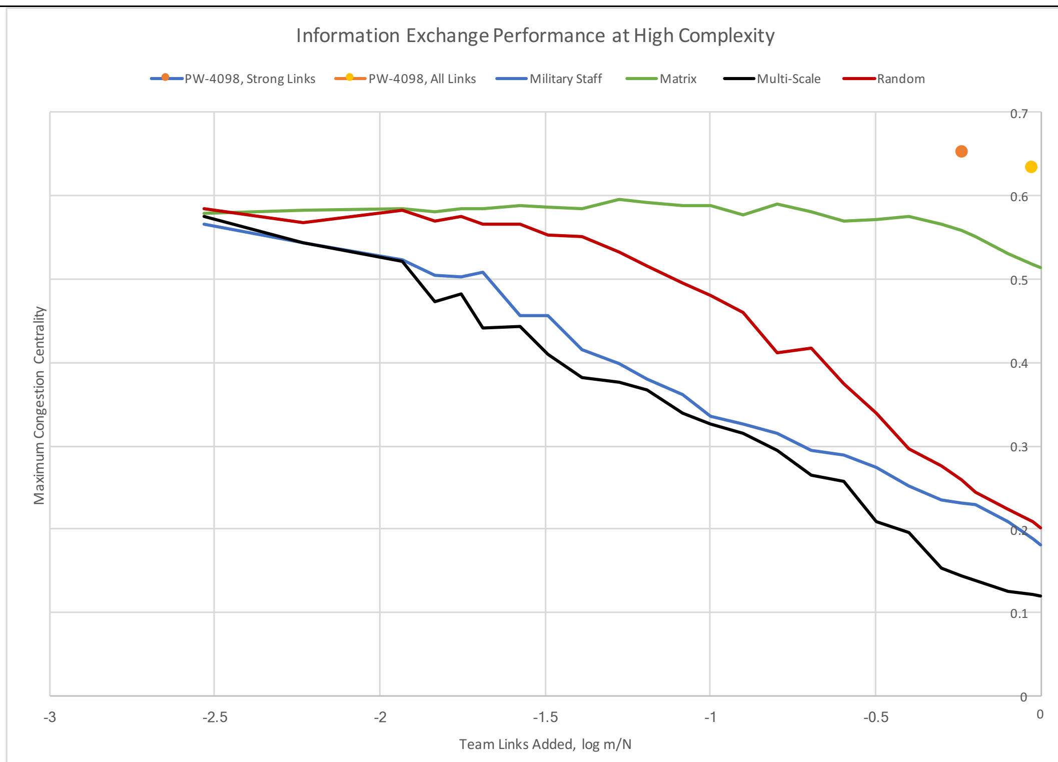



*Figure 40 - Artifact Performance of the PW4098 Design Organization for RFI, Balanced and Artifact Preference; Centralized and Decentralized Approvals; and Low, Moderate and High Complexity*

| RFI Preference | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Team Links | | | Centralized | | | Decentralized | | |
| | m | log m/N | Low | Moderate | High | Low | Moderate | High |
| Strong Link | 129 | -0.46 | 0.9941584 | 0.9540594 | 0.5375248 | 0.9670297 | 0.9494059 | 0.5523762 |
| All Links | 321 | -0.06 | 0.9888119 | 0.9409901 | 0.5248515 | 0.969505 | 0.9666337 | 0.5282178 |

| Balanced RFI-Artifact Processing | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Team Links | | | Centralized | | | Decentralized | | |
| | m | log m/N | Low | Moderate | High | Low | Moderate | High |
| Strong Link | 129 | -0.46 | 0.9737624 | 0.9557096 | 0.5412211 | 0.9747525 | 0.9639604 | 0.5531683 |
| All Links | 321 | -0.06 | 0.9841254 | 0.9493069 | 0.5215842 | 0.9847525 | 0.9539604 | 0.5280198 |

| Artifact Preference | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Team Links | | | Centralized | | | Decentralized | | |
| | m | log m/N | Low | Moderate | High | Low | Moderate | High |
| Strong Link | 129 | -0.46 | 0.9543564 | 0.960297 | 0.5436634 | 0.9724752 | 0.9573267 | 0.5431683 |
| All Links | 321 | -0.06 | 0.9846535 | 0.9546535 | 0.5139604 | 0.9847525 | 0.9509901 | 0.509505 |

Case Study Implications for Model Utility and Validity. Results indicate the information exchange and artifact models are useful tools for predicting performance of real-world organizations. At low complexity, both models indicate the PW4098 design organization will perform well and remain free of congestion failure, but at high complexity, both models predict the PW4098 design organization is susceptible to congestion failure. Results also indicate models are fairly reflective of real world behavior. At low complexity, in particular, PW4098 results for maximum congestion centrality and artifact completion rate fall in the same range as the corresponding results for random, multiscale, matrix and military staff organizational networks. At high complexity, PW4098 results diverge from those of other organizational networks, suggesting the models may under-predict the extent of the susceptibility of networks to failure at high complexity. At high complexity, maximum congestion centrality of the

PW4098 design organization diverged from that of matrix networks—the next worse-performing organizational network—by about 20%. A comparable divergence in artifact completion rates was seen at high complexity. It is worth recalling that the complexity scale used in the information exchange and artifact models is qualitative despite its use of a numerical scale. From a qualitative and predictive perspective, the information and artifact models represent real-world behavior in a useful manner.

# CHAPTER 5

## CONCLUSION

This chapter discusses findings, and presents conclusions and recommendations, including opportunities for further research.

<u>Task Environment Matters</u>

All organizational networks performed reasonably well when the task environment was limited to information exchange. Although matrix networks performed poorly compared to other organizational networks, all demonstrated satisfactory performance and remained free of congestion, even at high complexity. All organizational networks also remained free of congestion when the task environment was modified to include artifact processing, at low and moderate complexity. However, at high complexity, all organizational networks experienced congestion failure. This finding demonstrates how a simple change to the task environment alters network dynamics in important and unexpected ways. These kinds of subtle changes to network dynamics are a hallmark of complex systems.

Simon argued the creation of artifacts was the central activity of design organizations. (Simon, 1996) Information exchange is essential to the function of a design organization, but it is through the creation of artifacts that design organizations achieve their purpose. Organizations may be effective at information exchange, but that matters little if they are not effective at delivering artifacts. In other words, effective information exchange is a necessary, but not sufficient, condition for success in design.

*Findings provide insight into why organizational networks experience congestion failure at high complexity. For the artifact task environment,*

Table 12 shows military staff and multiscale networks decentralize information exchange to an extent comparable to that achieved in the information exchange environment. However, it also shows that the congested node for artifact processing is always at level one, that is artifact approvers (managers) are the congested node for artifacts and limiting factor for artifact performance. Decentralizing information exchange improves the processing of information requests, but it does not change the fact that all artifacts have to go to a manager for approval. At low and moderate complexity, the centralized approval of artifacts does not cause congestion failure, but at high complexity, the combination of centralized artifact approval and increased demand for information exchange leads to congestion. These findings support recommendations for decentralized authority in design organizations.

Findings also confirm the damaging effects of high queue states. Figure 29 demonstrates that congested nodes in all organizational networks have high effective congestion centrality until the number of team links added approaches the number of nodes, that is $m$ tends to *N*. Recalling that $\rho_{eff} = \frac{r_A + r_{RFI}}{\mu N}$, one sees $\rho_{eff}$ indicates capacity utilization because the right-hand side is the ratio of work done (rate of artifacts and information requests processed) to arrival rate. At high complexity, congested nodes are operating at capacity utilization factors above 0.9. As shown in Figure 14, this corresponds to high queue states. In other words, at high complexity, congested nodes are operating at high capacity utilization factors and high queue states.

The Pernicious Nature of Complexity

Despite variations, all organizational networks exhibit satisfactory performance at low to moderate complexity but suffer congestion failure at high complexity. This sort of tipping point behavior, shown in Figure 31, is another hallmark of complex systems and illustrates the pernicious nature of complexity. In the artifact task environment, increasing complexity of the system being designed has two compounding effects, both related to the concept of decomposability. At high complexity, it is less likely the task can be neatly decomposed and assigned to a single organization, so it is also less likely the individual responsible for the artifact, the originator, has sufficient information to complete the artifact alone. As a result, the originator puts the artifact on hold while soliciting assistance from others. Because the task is not decomposable, it is more likely information is needed from another worker outside the originator's department or immediate neighborhood, which means it will take longer for the information request to reach its target and be answered. The combined effect is high queue states, extended service times, and ultimately increased congestion.

When complexity is low to moderate, decisions on organizational structure are less important, from a congestion perspective, because a range of possible organizational structures will remain free of congestion and therefore have satisfactory performance. Of course, it is still possible to have poor organizational design and corresponding poor performance, but that poor performance would not be the result of an inherent susceptibility to congestion. However, at high complexity, an organizational structure that otherwise works perfectly well at low to moderate complexity can easily experience congestion failure, leading to the kinds of cost and schedule overruns that are increasingly common in projects that set out to design and deliver complex engineered systems.

Organizations that work reasonably well at low to moderate complexity may find themselves unprepared for high complexity. This situation is similar to the one described by Henderson and Clark, where organizations may find themselves unprepared for the effects of architectural innovation. Interestingly, they suggest the trend towards cross-functional organizations may reflect an understanding of the challenges of architectural innovation. In fact, cross-functional organizations, especially matrix organizations, may find themselves unprepared for innovation when that innovation increases complexity. Not surprisingly, organizational networks exhibit properties of complex adaptive systems, with two examples having already been noted, namely the noteworthy change in network dynamics resulting from a simple change to the task environment and the tipping point behavior exhibited by artifact closure rate in response to increasing complexity. In addition, Figure 33 demonstrated a positive feedback affecting RFIs. As team links are added, effective congestion centrality is reduced, which results in improved RFI and artifact processing, but as artifact processing improves, there are more artifacts in the system and greater opportunity for RFIs to be generated. RFI arrival rate increases in proportion to artifact closure rate.

Emergence, the idea that complex systems exhibit collective behavior not easily discerned from the behavior of individual system elements, is generally considered the defining characteristic of complex systems. Results demonstrate organizational networks exhibit emergent behavior. Agents in the organizational networks follow simple behavioral rules. In a given time period, workers examine their RFI and artifact queues and flip a coin to decide whether to process an RFI or artifact when both are present. The interesting dynamics, tipping point, and positive feedback effect already described could not be predicted from this simple behavioral rule.

In comparison, the co-called "complex" engineered system the organizational network is designing would be considered, strictly speaking, a merely complicated system because the elements in the engineered system are not adaptive. It was previously argued that complex engineered systems exhibit *quasi*-emergent behavior because the number and nature of system interactions are often poorly understood or exceed the limits of human cognition. From a practical perspective, this is an accurate characterization, and when engineers are being careful with their terminology, they will clarify that they mean structural complexity when referring to complex engineered systems. Of course, the design organization is inextricably linked to the engineered system being designed. Introduction of an adaptive agent, namely the human designers, necessarily makes the design organization, represented by an organizational network, a complex adaptive system.

Susceptibility of Matrix Organizations to Congestion Failure

The defining characteristics of a matrix organization are, in the first instance, the dual assignment of individual workers to both functional and project chains of command, and in the second instance, the assignment of project managers to their own branch in the overall organizational hierarchy. Conway's Law argues design organizations should be organized around the need for communication, and matrix organizations implicitly assume knowledge of communication requirements. In the specific case of a design organization, the matrix structure assumes the need for communication correlates to product architecture since architecture describes the relationship among components in the system being designed. For example, Browning describes the trend toward integrated product development, which brings together representatives from relevant functions using integrated product teams that own a product throughout its lifecycle. He describes the design for integration principles, which include assigning integrated product teams to system elements based on knowledge of system architecture. (Browning, 1996)

Ford and Randolph argue matrix organizations should improve information processing capability due to increased cross-functional collaboration. (Ford & Randolph, 1992). Schetler, Steyn and van Staden argue increased communication in matrix organizations improves both the quality of communication and overall team performance. (Schnetler, Steyn, & van Staden, 2015) While it is true matrix organizations improve communication relative to pure functional hierarchies, results indicate matrix organizations do not improve performance to the same extent as other organizational networks, even random networks. Results further demonstrate matrix organizations are particularly susceptible to congestion failure, and that the performance of matrix organizational networks is not improved by the smart team remedies explored.
First, results from phase 1 demonstrate matrix organizations are not efficient at information exchange, especially when compared to other organizational networks. Second, results from phase 2 demonstrate that when the task environment is extended to artifact processing, ineffective information exchange leads to artifact backlogs with corresponding poor artifact completion rates at high complexity. Finally, results from phase 3 demonstrate that smart team remedies do not improve the performance of matrix organizational networks to the same extent as other networks.

The PW-4098 case study provides critical insights. Rowles noted one-third of integrated product team interactions occurred outside the team's hierarchical group, and that one-fourth of interactions did not correspond to design relationships. (Rowles, 1999) Recall also Sinha and de Weck compared two jet engine designs and found the newer and more complex design required a significant increase in both intra- and inter-team interactions, including new connections between groups not previously connected. (Sinha & de Weck, 2012)

In complex engineered systems, interactions are poorly understood, so one would expect a large number of interactions would occur outside a structure based on known or predicted design relationships. Increasing complexity only exacerbates the problem because it increases the need for cross-functional collaboration and implies tasks are not decomposable, which means collaboration must occur beyond the hierarchical or team arrangements defined by the matrix structure. Matrix organizations improve communication within the teams formed, but do not facilitate the more extensive cross-functional and cross-team communication needed when designing complex engineered systems. As a result, matrix organizations are particularly susceptible to congestion failure when complexity is high.

Military Staff Organizational Networks Exhibit Multiscale Properties

Military staff organizational networks exhibit performance comparable to multiscale networks across a range of situations. Phase 1 demonstrated military staff organizational networks exhibited performance comparable to multiscale networks in the information exchange environment. For example, Figure 24 demonstrates military staff organizational networks had maximum congestion centrality comparable to multiscale networks for a broad range of team links added, up to the point that the number of team links, *m*, approached the number of nodes, *N*. However, as *m* continues to increase, military staff organizational networks diverge from multiscale networks. This kind of divergence is a recurring theme and will be further explored shortly. Table 8 demonstrates military staff organizational networks decentralize congestion similar to multiscale networks, although decentralization is not as effective in military staff organizational networks.

Phase 2 demonstrated military staff organizational networks had performance similar to multiscale organizational networks. For example, Figure 28 demonstrates military staff organizational networks have artifact completion rates comparable to multiscale networks at high complexity. Interestingly, however, Figure 29 shows effective congestion centrality for military staff organizational networks did not track with multiscale networks and instead tracked more closely to random networks. In addition, as *m* tends to *N*, RFI congestion centrality for military staff organizational networks diverged from that of multiscale networks and began to converge to that of random networks.

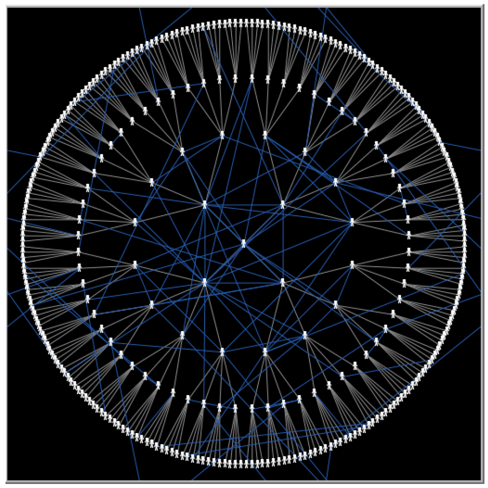
Phase 3 demonstrated smart team remedies improved the performance of military staff organizational networks to an extent comparable to multiscale networks. In particular, decentralization of approval made them robust to congestion as *m* tended to *N*. An examination of military staff organizational network structure provides useful insight into their behavior. Figure 41 compares the structures of multiscale, military staff and random networks for 108 team links, log $m/N = -0.5$. Although multiscale and military staff organizational networks have different construction algorithms, they have similar structures, with high central connectivity in the spatial sense when the networks are laid out in a radial fashion around the central node. By comparison, random networks have greater peripheral connectivity. Strictly speaking, multiscale and military staff organizational networks have distinct connection patterns. Multiscale organizational networks achieve central connectivity through direct links between workers, whereas military staff organizational networks achieve it by links through intermediate nodes, namely the team leads, shown in red.

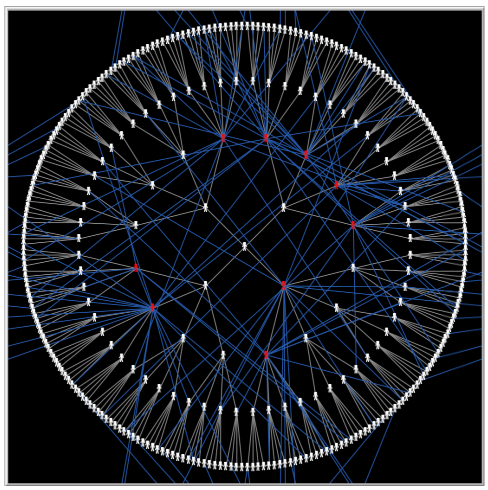
A key difference between military staff and matrix organizational networks is that military staff networks embed the team leaders in the existing functional hierarchy, while matrix networks place them in a separate branch of the functional hierarchy. Matrix organizations overlay a project management hierarchy on top of an existing functional hierarchy, while military staff organizations create a structure with multiscale qualities. This structural difference likely explains much about the performance difference between military staff and matrix networks.

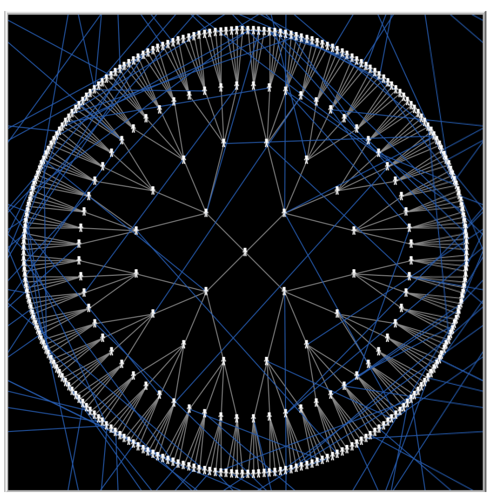
Despite their different construction method, military staff organizational networks have structural similarities sufficient to give them performance characteristics comparable to multiscale networks over a broad range of team links added. However, results also demonstrate military staff organizational networks begin to diverge from multiscale networks as $m$ tends to $N$ and log $m/N$ tends to 0. For example, maximum congestion centrality in the information exchange environment diverges just as RFI congestion centrality diverges in artifact environment. This suggests that as $m$ tends to $N$, the benefit of structural similarity becomes less important. Notably, effective congestion centrality of military staff organizational networks diverges sharply from that of multiscale organizational networks in the artifact environment, suggesting military staff networks are not as effective at relieving the combined congestion associated with artifacts and information requests.

Military staff organizational networks are *not* multiscale networks, but they do have performance properties comparable to multiscale networks over a wide range of situations. This finding suggests an answer to Magee's question of how multiscale networks might be created in practice. (Magee, 2010) Since military staff organizational networks exhibit similar properties, it is likely they can be adjusted and used in ways to realize the robust performance characteristics of multiscale networks. In fact, results from phase 3 illustrate this, showing that minor changes to the task environment improve the robustness of military staff organizational networks to congestion failure. For example, Figure 38 demonstrates military staff organizational networks achieve congestion robustness, with artifact completion rates exceeding 90%, in the range of team links added from log $m/N$ = -0.5 to 0, when artifact approvals are decentralized.

*Figure 41 - Comparison of Multiscale, Military Staff, and Random Organizational Network Structures for 108 Team Links*







Simple Remedies Improved Network Performance

Results from phase 3 demonstrated simple changes to network construction algorithms and the task environment improved the performance of organizational networks. Results demonstrated that matrix organizational networks perform better when teams are small and military organizational networks perform better when teams are large, and that RFI-artifact preference was not a significant factor affecting performance of any organizational network. More importantly, results demonstrate decentralization of artifact approvals improved the performance of all organizational networks.

Results from previous phases predicted the value of decentralization. For example, Table 4 showed multiscale and random organizational networks decentralized the congested node in the information exchange environment, while Table 5 showed decentralized congested nodes had lower maximum congestion centralities—much lower in some cases. Similarly,

Table 12 showed how multiscale and military staff organizational networks decentralized RFI congestion in the artifact environment.

Table 12 also showed neither organizational network decentralized artifact congestion. Adding decentralized artifact approvals to the Smart Teams model facilitated decentralized approvals and improved organizational network performance.

Results confirm Reinersten's assertions regarding the value of decentralized control. Organizations interested in improving performance will be interested in his principles for implementing decentralized control and maintaining organizational alignment.

Value of Agent-Based Modeling

Results confirm the value of agent-based modeling (ABM) for evaluating and understanding complex systems. For example, the validation of information exchange networks using MATLAB demonstrated models of organizational networks could be implemented using either ABM or more traditional programming tools, such as MATLAB. However, the NetLogo interface aids visualization and improves understanding relative to the purely numerical results obtained from MATLAB. To be fair, MATLAB can also be programmed to provide visual depictions, but NetLogo provides them as an inherent feature of its modeling environment. Figure 41 demonstrates the value of visualization because it is the visual comparison of multiscale and military staff organizational networks that suggests the idea that structural similarities between the two networks contributes to the multiscale-like behavior seen in military staff networks. In addition, visual observation of model execution, especially using the "go once" feature, which allows step-wise execution, aids understanding of organizational network behavior. In particular, observation of artifact backlogs at provides understanding of why organizational networks experience congestion failure at high complexity.

*Visualization and the ability to observe temporal behavior also provided insights into network dynamics. Observations showed networks generally did not exhibit equilibrium behavior. For example,*

Figure 36 plotted the difference between open and completed artifacts and showed artifact closure rate did not converge to an equilibrium value. Instead, it continued to vary over time. In this regard, it would be more appropriate to say organizational networks are under control than at equilibrium. This observation confirms one of the key features of ABM. Equation-based models tend to predict average or equilibrium behaviors at the expense of dynamics, while agent-based models illustrate dynamic behaviors. Both types of models are useful, but in this case, use of ABM provided useful insight into the dynamic behavior of organizational networks.

All software tools have advantages and disadvantages. NetLogo provides powerful visualization tools and a syntax that facilitates creation of agent-based models, but it performs some basic computer functions quite poorly. In particular, activities that require loops or recursive searching are not easily implemented in NetLogo or tend to slow model performance significantly. This was particularly evident when first attempting to implement the DWS stochastic rule. This action essentially requires searching through the network for a pair of nodes that has a sufficiently high probability of forming a team link. Since the number of possible links is on the order of $N^2$, hundreds or thousands of node pairs must potentially be tested for each team link added, even for relatively small networks like those tested here. In addition, for each pair tested, the lowest common node between them, $D_{ij}$, must be identified through recursive search. The initial approach repeated this recursive search for every pair tested.

Implementation of the information exchange model in MATLAB yielded the critical insight that $D_{ij}$ is a property defined by the hierarchical structure of any given network, thus the $D_{ij}$ values between every node pair could be calculated in advance and stored in a file as an $N$ x $N$ matrix. MATLAB was able to handle this task with ease and use of the $D_{ij}$ matrix as an input to NetLogo improved model performance significantly. The benefit was two-fold because $D_{ij}$ values are needed to route information requests, thus having the values stored in a matrix prevented the need to calculate them at each step of routing every single information request. The integration of MATLAB and NetLogo proved quite useful, and NetLogo users may find value in a NetLogo-MATLAB application programming interface (API) similar to other APIs (called extensions in NetLogo) already provided.

The PW4098 case study demonstrated the utility of ABM for analyzing and predicting the performance of real organizations. When tested with the information exchange and artifact models, the PW4098 organization exhibited performance comparable to other organizations. Models represent the design process using relatively simple task environments and methods of information exchange, consistent with the "keep it simple" design principle articulated by Wilensky and Rand but deliver meaningful results consistent with real-world behavior. (Wilensky & Rand, 2015)

Variations in Multiscale Networks

The ability of multiscale networks to decentralize congestions has already been mentioned but warrants additional discussion. Dodds, Watts and Sabel demonstrated the robustness of multiscale networks to congestion, using maximum congestion centrality, $\rho_{max}$, as a key indicator of robustness. It is especially noteworthy, then, that even within the multiscale class, different network configurations can have quite different values of $\rho_{max}$ for the same number of team links. Focusing on single line from information network results, $m = 9$ and $\log m/N = -1.6$ for multiscale networks:

| | Groups | | Total |
|---|---|---|---|
| Depth of Congested Node | 0 | 1 | |
| Number | 6 | 4 | 10 |
| Mean | 0.5336 | 0.4131 | |
| Total Contribution | 3.2016 | 1.6524 | 4.854 |
| Overall Average | | | 0.4854 |

Data demonstrates the importance of decentralization. In six of ten runs, the congested node was at level zero, the top of the hierarchy, but in four of ten, the congested node was one level lower. The overall average of $\rho_{max}$ combines these results, but it is clear that the decentralized nodes put downward pressure on the overall average. Decentralized nodes have significant impact on overall maximum congestion centrality for a given value of *m*, and this result demonstrates that even within multiscale networks, there are subclasses of networks with better performance. Military staff organizational networks exhibited the same phenomenon. This matter warrants further investigation.

<u>Conclusions</u>

Referring to the research questions set out in Chapter 3, findings support the following conclusions:

- In the information exchange task environment, all organizational networks perform well and remain free of congestion at low, moderate and high complexity.
- In the artifact task environment, all organizational networks perform well at low to moderate complexity, but all are susceptible to congestion failure at high complexity.
- At low to moderate complexity, military staff and matrix organizational networks perform well, or well enough, and remain free of congestion, but military staff

networks consistently out-perform matrix networks, having lower congestion centralities and higher artifact completion rates for a given number of team links added.

- At high complexity, military staff and matrix organizational networks are susceptible to congestion failure, but military staff networks continue to out-perform matrix networks
- Military staff organizational networks exhibit performance comparable to multiscale networks over a range of situations.
- Matrix organizational networks tended to exhibit poor performance compared to all other networks, often being out-performed by even random networks, especially at high complexity.
- Since military staffs have performance comparable to multiscale networks across a range of situations, they are the preferred organizational form for organizations that design complex engineered systems.
- Changing team size improved performance of military staff and matrix organizational networks, and decentralizing artifact approval authority improved the performance of all networks. Military staff organizations perform better when team sizes are large and matrix organizations perform better when team sizes are small. Decentralizing artifact approvals made military staff and multiscale networks robust to congestion failure in the artifact task environment at high complexity.

Summary and Recommendations

This study set out to understand why some organizations fail to effectively manage the design of complex engineered systems. It used agent-based modelling to evaluate and compare the effectiveness of random, multiscale, matrix and military staff organizational networks, modelling design as an activity that requires organizations to balance competing demands to complete artifacts and share information. Complexity—strictly speaking, structural complexity—results from the number and diversity of elements in the system being designed, and their interactions, which are often poorly understood. Increasing complexity challenges the design organization's ability to keep artifacts and information-sharing in balance by increasing the frequency and extent of cross-functional collaboration required. The study found all organizational networks perform well, or at least well enough, at low to moderate complexity, but also found that all are susceptible to congestion failure at high complexity. As congestion builds, the organization falls further and further behind, leading to the cost and schedule overruns that seem to plague projects that set out to design complex engineered systems like ships and aircraft.

Conventional wisdom argues projects should be organized around matrix organizations because they improve communication and cross-functional collaboration relative to traditional, functional hierarchies. However, results indicate matrix organizations are particularly susceptible to congestion failure. Compared to multiscale, military staff and even random organizations, matrix organizations are not effective at exchanging information because they overlay a project management hierarchy on top of an existing functional hierarchy. The resulting structure fails to create the conditions for effective cross-functional communication when increasing complexity requires collaboration outside established channels. As a consequence, matrix organizations experience congestion failure when challenged by high complexity.

Military staff organizational networks demonstrated performance properties comparable to multiscale networks over a range of conditions. They are not multiscale networks but have structural similarities to them. They therefore represent a practical approach to creating an organization with multiscale properties. Unlike matrix organizations, military staff organizations embed team leaders in the functional hierarchy, which makes them more effective at cross-functional communication.

Conway argued design organizations should be structured around the need to communicate (Conway, 1968), but the essence of complexity is the inability to fully appreciate the interactions in the system being designed, which likewise makes it impossible to predict in advance which elements of the organization need to communicate. Sinha and de Weck examined how changes to product architecture affect design organizations, demonstrating a feedback effect. Performance and feature improvements often increase a product's complexity, necessitating organizational changes, but those organizational changes often lag behind design changes. (Sinha & de Weck, 2012) Poire and Sabel argued organizations know little about how to accomplish a project when they first embark on it, so learning and design must occur in parallel. (Poire & Sabel, 1984) Watts argued organizations compensate for ambiguity by exchanging information. Because networks are costly in terms of time and energy, robust networks must balance production and information exchange. (Watts, 2003) As complexity increases, organizations must communicate outside their usual hierarchical and teaming arrangements. Military staff organizational networks are better at accommodating such demands for increased cross-functional communication than matrix organizations and therefore represent a preferred solution for organizations that design complex engineered systems. Decentralization of congestion and approval authority further improves performance.

Organizations that set out to design complex engineered systems should organize themselves around a structure similar to that used by military staffs by embedding project managers in the functional hierarchy and should decentralize control to the maximum extent possible. Project-based organizations may find it challenging to implement this recommendation because the current project management orthodoxy emphasizes a separate organizational role for project managers. Success will depend on having project managers able to balance functional and project management roles, but these are often seen as distinct areas of specialization, especially in design organizations where design demands specific technical expertise. Military organizations are more comfortable blending project and functional roles because the concept of dual chains of command is built into their culture.

The separation of project management and functional roles is reminiscent of the division of labor described by Adam Smith, which exploits returns on specialization. The division of labor leads to hierarchical organizations, and the separation of project management and functional roles helps explain why matrix organizations essentially embed two hierarchies on one another. By comparison, one can argue military staff organizations do a better job of achieving the kind of flexible specialization, which exploits economies of scope and general-purpose capabilities, recommended by Poire and

Sabel. (Poire & Sabel, 1984) Organizations that design complex engineered systems should organize themselves around the military staff model, but implementation will require cultural change, and that is no trivial task. Success will depend on having personnel capable of performing, and comfortable with, project and technical roles, and that capability must be developed and encouraged over time. Organizations that invest in such capabilities will reap rewards in terms of organizational resilience to congestion.

<u>Opportunities</u>

Several opportunities for further research stand out. First, variations in maximum congestion centrality for multiscale and military staff organizational networks should be further explored. Such an exploration might include evaluation and comparison of other network parameters or comparison of adjacency matrices to identify and potentially correlate characteristics of network configurations that decentralize the congested node. An exploration should also examine ways to preferentially generate networks that decentralized the congested node since they have improved robustness to congestion. Second, actual military staff organizations could be characterized to evaluate their performance and confirm they are robust to congestion, especially compared to matrix organizations. Finally, the models developed for this study could be extended to evaluate other variations in network construction algorithm, task environment and routing method, or even to other similar activities. For example, different artifact preference models could be explored. As implemented, artifact preference was shown to not be a significant factor affecting network performance, but different rules, especially those that choose preference dynamically, could yield different results. In addition, variations in worker capacity could be explored, including variations resulting from the number of team links a particular node has.  Maintaining team links can be time and resource intensive and can detract from a worker's ability to get work done.  Think, for example, of time spent in meetings and other collaborative activities.  If team links had a capacity cost, then the network would balance worker capacity against cross-functional collaboration.

# APPENDICES

# APPENDIX 1: Robust Networks Information Exchange Model

Elements of the Information Exchange, Version 1, Model

| Agents |
|---|
| Workers, representing the individuals within the hierarchy.<br>Depth (level);<br>Department (major division of the hierarchy);<br>Supervisor (immediate superior in the hierarchy);<br>Team (team assignment, for matrix and military staff organizational networks);<br>Capacity (the amount of work the worker can perform in one time step);<br>RFI queue (list of RFIs to be processed, i.e., worker's "in box"); and<br>RFI count (number of RFIs processed by worker).<br><br>Requests for Information (RFIs), representing messages passed between workers.<br>Originator (worker who originated the RFI);<br>Target (worker to whom the RFI was sent);<br>Status (status of the RFI: open, answered, or complete); and<br>Age (age of RFI). |
| Links |
| Organization Links, representing the hierarchical structure.<br>Team Links, representing the cross-functional team links added to the hierarchy. |
| Environment |
| The environment is defined by the backbone hierarchical network, the team links added to the hierarchical backbone, and the task environment. The DWS model describes the task environment in terms of the rate and distribution of messages to be exchanged.<br>The information exchange rate, $\mu$, is the average number of messages originated by each node at each time step, and $\mu N$ is the total number of messages originated across the network at each time step.<br>Message routing considers task decomposability. Tasks that are nearly decomposable require communication only within the same team, meaning nodes with the same immediate superior, whereas tasks that are decomposable require communication across the network. For a given source node, $s$, a target node, $t$ is selected based on the distance between the two nodes, $x_{st}$, using the following stochastic rule:<br>$P(s,t) \propto e^{\frac{-x_{st}}{\xi}}$<br><br>When $\xi = 0$, local dependencies prevail; when $\xi = \infty$, global dependencies prevail.<br>*Information Exchange, Version 3-0* assumes global dependencies. |
| Time Behavior |
| At each time step, workers create and/or process RFIs.<br>RFIs arrive according to a random Poisson process with mean equal to the user-specified RFI arrival rate. RFIs are assigned source (originator) and target nodes at random.<br>Messages pass from source to target through a chain of intermediate nodes. At each |

| |
|---|
| time step, worker nodes pass RFIs they initiate or receive, up to their capacity, by selecting an immediate neighbor with the lowest common ancestor with the target node |
| Inputs |
| Network parameters: Levels, branching ratio;<br>Network type: random, multiscale, matrix or military staff (BCCWG);<br>Number of Teams, for matrix and military staff organizational networks;<br>$D_{ij}$ Name, the file containing a matrix of the depths of lowest common ancestors;<br>Team links added, $m$;<br>RFI Arrival Rate; and<br>Worker Capacity |
| Outputs |
| The principal output and measure of performance is maximum congestion centrality, $\rho_{max}$. Assuming each node can process up to $R_i$ messages per time step, an organizational network will, on average, remain free of congestion when $R_i > r_i = \mu N \rho_i$, where $\rho_i$, the congestion centrality of node $i$, is the probability that any given message will be processed by node $i$. Maximum congestion centrality across the organizational network, $\rho_{max}$, is a measure of robustness to congestion failure.<br>Additional outputs include:<br>Identity of the most congested node;<br>RFI completion rate;<br>Average RFI age; and<br>Network parameters: mean path length and global clustering coefficient. |

Verification & Validation Worksheets *Information Exchange 1*

| Verification of the *Information Exchange 1* Model |
|---|
| Documentation |
| Model documentation consistent with conceptual model<br>Code comments adequately identify implementation of conceptual model |
| Comments |
| Documentation and code comments sufficient. |
| Programmatic Testing |

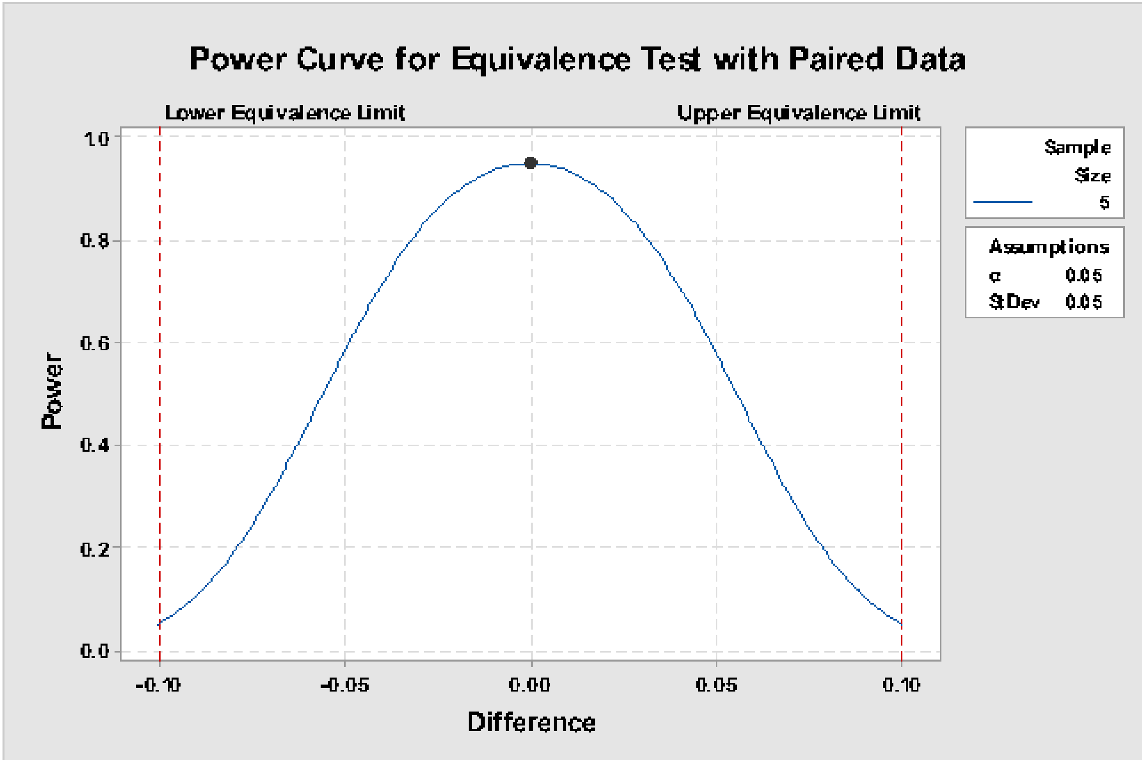
| Item Tested | Method | Result | Comments |
|---|---|---|---|
| Hierarchy matches user inputs for branching ratio and number of levels | Observe structure in viewer for different values of $b$ and $L$ | Satisfactory | |
| Calculation of *P(i,j)* for random and multiscale networks | Show and record intermediate values during implementation and compare to hand calculation | Satisfactory. | Verification uncovered and corrected error in identification of lowest common node. |
| Proper creation of matrix and military staff organizational networks | Observe structure and confirm correct number of team leaders and team links | Satisfactory | |
| Proper routing of RFIs | Follow individual RFIs using "watch me" functionality | Satisfactory | Because the "ask" command in NetLogo selects agents in random order, an individual RFI can move multiple steps in one time increment. |

| Test Cases | | |
|---|---|---|
| Method | Description | Result |
| Edge cases: random networks represent an edge case with extreme values of $\lambda$ and $\varsigma$ | Calculate $\rho_{max}$ for different numbers of team links and plot results. | Networks demonstrate behavior similar to results presented by Dodds, Watts, and Sabel. |

| Conclusion |
|---|
| Model correctly implements DWS stochastic rule and creates organizational networks with behavior consistent with predictions. |

<table>
<tr><td colspan="2">Validation of the Information Exchange 1 Model</td></tr>
<tr><td colspan="2">Face Validation</td></tr>
<tr><td>Micro-Face Validation</td><td>Macro-Face Validation</td></tr>
<tr><td>Principal elements of model are the agents representing workers in and organization, and the organizational and team links that connect them. Organizations with these characteristics are ubiquitous across any number of disciplines. The model uses values of λ and ζ corresponding to different classes of organizational structures. The model assumes RFIs arrive according to a random Poisson process, consistent with queueing theory. The model assumes tasks are not decomposable, which is a reasonable and limiting case.</td><td>The model realistically depicts the flow of information in organizational networks, combining both formal passing of information up and down a hierarchy, and informal passing through team relationships.<br><br>The model realistically represents matrix and military staff organizational networks, two networks found in real-world organizations.</td></tr>
<tr><td colspan="2">Empirical Validation</td></tr>
<tr><td>Empirical Input Validation</td><td>Empirical Output Validation</td></tr>
<tr><td>The hierarchical backbone is described by number of levels and branching ratio. This is an idealization in that real organizations exhibit irregularities in both level and branching, but the idealization is reasonable.</td><td>For this model, empirical validation is accomplished by cross-validation against a model implemented in MATLAB, along with comparison of results to those previously published by Dodds, Watts and Sabel.</td></tr>
<tr><td colspan="2">Conclusion</td></tr>
<tr><td colspan="2">The multiscale and random network behavior is consistent with reference data and results obtained from an alternate implementation in MATLAB; the model therefore considered valid for further development to explore the effectiveness of organizational networks.</td></tr>
</table>

Validation Experiment Results from Minitab




## Factor Information

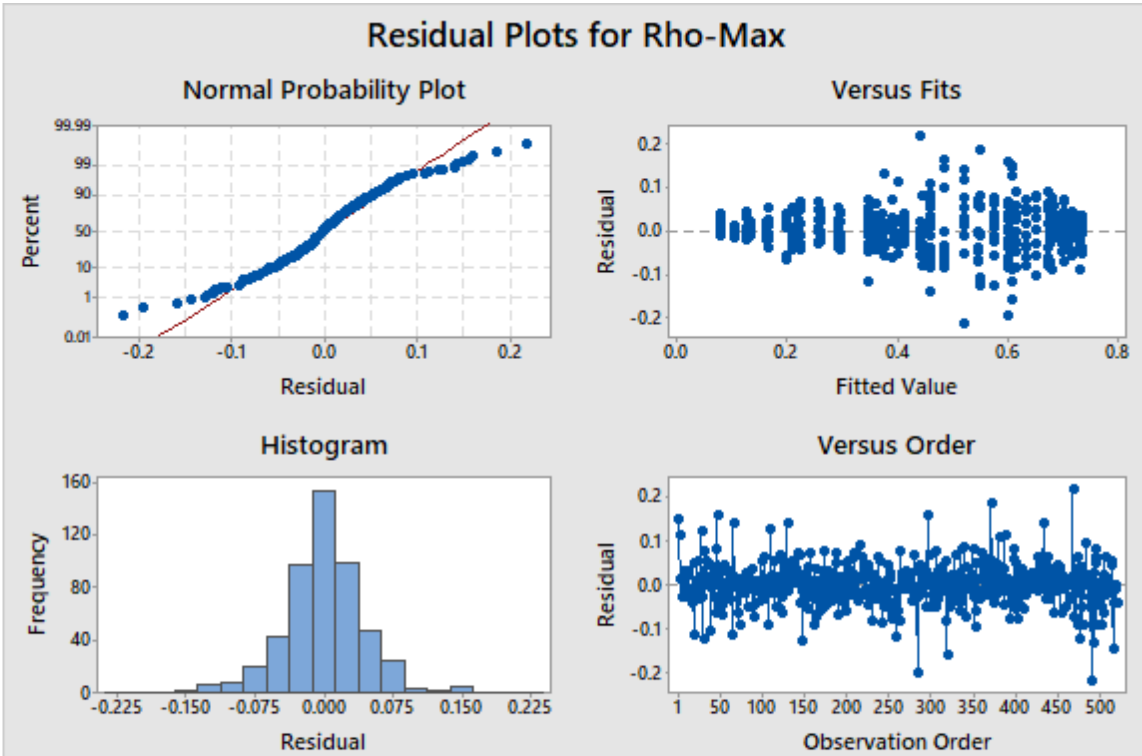
| Factor | Type | Levels | Values |
|---|---|---|---|
| Software | Fixed | 2 | NetLogo, MATLAB |
| Network Type | Fixed | 2 | Multi-Scale, Random |
| Team Links Added (m) | Fixed | 13 | 1, 4, 6, 9, 14, 22, 35, 55, 86, 136, 216, 341, 1079 |

## Analysis of Variance

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Software | 1 | 0.0080 | 0.00801 | 3.21 | 0.074 |
| Network Type | 1 | 2.6900 | 2.69001 | 1078.65 | 0.000 |
| Team Links Added (m) | 12 | 19.3533 | 1.61278 | 646.70 | 0.000 |
| Software*Network Type | 1 | 0.0227 | 0.02274 | 9.12 | 0.003 |
| Software*Team Links Added (m) | 12 | 0.0525 | 0.00437 | 1.75 | 0.053 |
| Network Type*Team Links Added (m) | 12 | 0.7308 | 0.06090 | 24.42 | 0.000 |
| Error | 480 | 1.1971 | 0.00249 | | |
| Lack-of-Fit | 12 | 0.0367 | 0.00306 | 1.23 | 0.257 |
| Pure Error | 468 | 1.1604 | 0.00248 | | |
| Total | 519 | 24.0544 | | | |

## Model Summary

| S | R-sq | R-sq(adj) | R-sq(pred) |
|---|---|---|---|
| 0.0499386 | 95.02% | 94.62% | 94.16% |



Hypothesis Testing for Validation Experiment

The following hypothesis testing evaluates equality of maximum congestion centralities for random and multiscale networks using the paired data test:

| | |
|---|---|
| Hypothesis | $H_0$: $\mu_D = 0$<br>$H_1$: $\mu_D \neq 0$<br>($\mu_D$ refers to mean of differences) |
| Test Statistic | $t_0 = m\ \bar{d} / s_D/\sqrt{n}$ |
| Criteria to reject $H_0$ | $\lvert t_o \rvert > t(\frac{\alpha}{2}, n-1)$ |

| Hypothesis Testing for Multiscale Networks NetLogo vs. MATLAB | | | | | |
|---|---|---|---|---|---|
| m | $\bar{d}$ | $S_D$ | $\lvert t_0 \rvert$ | t | Reject |
| 1 | -0.05201 | 0.05903943 | 2.7857664 | 2.26215716 | TRUE |
| 4 | -0.08313 | 0.133780683 | 1.96500822 | 2.26215716 | FALSE |
| 6 | -0.08142 | 0.179578722 | 1.43375921 | 2.26215716 | FALSE |
| 9 | -0.0272 | 0.108138296 | 0.79540695 | 2.26215716 | FALSE |
| 14 | -0.03386 | 0.067341636 | 1.59002257 | 2.26215716 | FALSE |
| 22 | -0.00375 | 0.057687304 | 0.20556588 | 2.26215716 | FALSE |
| 35 | -0.00494 | 0.052347179 | 0.29842395 | 2.26215716 | FALSE |
| 55 | 0.02604 | 0.034563282 | 2.38246214 | 2.26215716 | TRUE |
| 86 | -0.01853 | 0.056932241 | 1.02924114 | 2.26215716 | FALSE |
| 136 | -0.02601 | 0.056011456 | 1.46846462 | 2.26215716 | FALSE |
| 216 | 0.01653 | 0.035564997 | 1.46977237 | 2.26215716 | FALSE |
| 341 | -0.00021 | 0.022683104 | 0.02927634 | 2.26215716 | FALSE |
| 1079 | 0.01451 | 0.009799371 | 4.68240769 | 2.26215716 | TRUE |

| Hypothesis Testing for Random Networks, NetLogo vs. MATLAB | | | | | |
|---|---|---|---|---|---|
| m | $\bar{d}$ | $S_D$ | $\lvert t_0 \rvert$ | t | Reject |
| 1 | -0.0070 | 0.0354 | 0.6248 | 2.2622 | FALSE |
| 4 | 0.0036 | 0.0163 | 0.6903 | 2.2622 | FALSE |
| 6 | -0.0018 | 0.0258 | 0.2243 | 2.2622 | FALSE |
| 9 | 0.0027 | 0.0539 | 0.1572 | 2.2622 | FALSE |
| 14 | 0.0030 | 0.0785 | 0.1189 | 2.2622 | FALSE |
| 22 | 0.0109 | 0.0888 | 0.3874 | 2.2622 | FALSE |
| 35 | 0.0288 | 0.0448 | 2.0335 | 2.2622 | FALSE |
| 55 | -0.0022 | 0.1473 | 0.0468 | 2.2622 | FALSE |
| 86 | -0.0350 | 0.1160 | 0.9535 | 2.2622 | FALSE |
| 136 | 0.0274 | 0.0525 | 1.6500 | 2.2622 | FALSE |
| 216 | 0.0179 | 0.0270 | 2.0995 | 2.2622 | FALSE |
| 341 | 0.0166 | 0.0435 | 1.2077 | 2.2622 | FALSE |
| 1079 | 0.0051 | 0.0126 | 1.2749 | 2.2622 | FALSE |

Elements of the Information Exchange, Version 2, Model

| Agents |
|---|
| Workers, representing the individuals within the hierarchy.<br>Depth (level);<br>Department (major division of the hierarchy);<br>Supervisor (immediate superior in the hierarchy);<br>Team (team assignment, for matrix and military staff organizational networks);<br>Capacity (the amount of work the worker can perform in one time step);<br>RFI queue (list of RFIs to be processed, i.e., worker's "in box"); and<br>RFI count (number of RFIs processed by worker<br><br>Requests for Information (RFIs), representing messages passed between workers.<br>Originator (worker who originated the RFI);<br>Target (worker to who the RFI was sent);<br>Status (status of the RFI: open, answered, or complete); and<br>Age (age of RFI). |
| **Links** |
| Organization Links, representing the hierarchical structure.<br>Team Links, representing the cross-functional team links added to the hierarchy. |
| **Environment** |
| The environment is defined by the backbone hierarchical network, the team links added to the hierarchical backbone, and the task environment. The DWS model describes the task environment in terms of the rate and distribution of messages to be exchanged.<br>The information exchange rate, $\mu$, is the average number of messages originated by each node at each time step, and $\mu N$ is the total number of messages originated across the network at each time step.<br>Message routing considers task decomposability, which depends on complexity. When the system being designed is more complex, tasks are less decomposable and require greater cross-functional collaboration. Thus, at high complexity, message target nodes are selected at random from across the hierarchy. At low complexity, tasks are decomposable, and message target nodes are selected at random from among other nodes in the same major branch as the source. |
| **Time Behavior** |
| At each time step, workers create and/or process RFIs.<br>RFIs arrive according to a random Poisson process with mean equal to the user-specified RFI arrival rate. RFIs are assigned source (originator) and target nodes at random.<br>Messages pass from source to target through a chain of intermediate nodes. At each time step, worker nodes pass RFIs they initiate or receive, up to their capacity, by selecting an immediate neighbor with the lowest common ancestor with the target node. |
| **Inputs** |
| Network parameters: Levels, branching ratio;<br>Network type: random, multiscale, matrix or military staff (BCCWG); |

| |
|---|
| Number of Teams, for matrix and military staff organizational networks;<br>$D_{ij}$ Name, the file containing a matrix of the depths of lowest common ancestors;<br>Team links added, $m$;<br>Complexity;<br>RFI Arrival Rate; and<br>Worker Capacity |
| Outputs |
| The principal output and measure of performance is maximum congestion centrality, $\rho_{max}$. Assuming each node can process up to $R_i$ messages per time step, an organizational network will, on average, remain free of congestion when $R_i > r_i = \mu N \rho_i$, where $\rho_i$, the congestion centrality of node $i$, is the probability that any given message will be processed by node $i$. Maximum congestion centrality across the organizational network, $\rho_{max}$, is a measure of robustness to congestion failure.<br>Additional outputs include:<br>Identity of the most congested node;<br>RFI completion rate;<br>Average RFI age; and<br>Network parameters: mean path length and global clustering coefficient. |

Verification & Validation Worksheets-*Information Exchange 2*

| Verification of the *Information Exchange 2* Model | | | |
|---|---|---|---|
| Documentation | | | |
| Model documentation consistent with conceptual model<br>Code comments adequately identify implementation of conceptual model | | | |
| Comments | | | |
| Documentation and code comments sufficient. | | | |
| Programmatic Testing | | | |
| Item Tested | Method | Result | Comments |
| Proper selection of target nodes based on complexity rating. | Confirm selection of target nodes based on complexity test. | Satisfactory | |

| Test Cases | | |
|---|---|---|
| Method | Description | Result |
| Edge cases: low and high complexity | Calculate $\rho_{max}$ for high and low complexity and plot results | Networks demonstrate a narrow range of $\rho_{max}$ for low complexity, and a wider range for high complexity |
| Conclusion | | |
| Model extends previous model to account for complexity and correctly implements the method selected to model complexity. Other features were previously verified. | | |

| Validation of the *Information Exchange 2* Model | |
|---|---|
| Face Validation | |
| Micro-Face Validation | Macro-Face Validation |
| The model implements complexity in a way that increases the need for cross-functional communication as complexity increases, consistent with the notion that complexity decreases task decomposability. | Complex engineered systems are *complex* because they have numerous and varied elements whose interactions are poorly understood. As complexity increases, design tasks are likely to require greater cross-functional communication and collaboration. |
| Empirical Validation | |
| Empirical Input Validation | Empirical Output Validation |
| The model rates complexity on a scale of 1 to 10. Use of a simple scale is not meant to represent a quantitative comparison of system complexity, but instead differentiates systems of low and high complexity in a numerical fashion | Empirical output validation relies on stylized facts, i.e., the expectation that high complexity will increase congestion centralities because non-decomposable tasks require greater cross-functional routing. |

| | |
|---|---|
| that is easy to implement in a model. | |
| Conclusion | |
| The model provides a reasonable representation of the difference in information exchange network behavior at low and high complexity. | |

Hypothesis Testing-Information Exchange Characterization

The following hypothesis testing evaluates equality of maximum congestion centralities for random and multiscale networks compared to military staff and matrix networks at high complexity using the paired data test, as above:

| Multiscale-Military Comparison Hypothesis Test | | | | | |
|---|---|---|---|---|---|
| m | $\bar{d}$ | $S_D$ | $\lvert t_0 \rvert$ | t | Reject |
| 1 | 0.0055 | 0.0510 | 0.3437 | 2.2622 | FALSE |
| 2 | 0.0159 | 0.0605 | 0.8337 | 2.2622 | FALSE |
| 4 | 0.0247 | 0.0488 | 1.5991 | 2.2622 | FALSE |
| 5 | (0.0355) | 0.0726 | 1.5475 | 2.2622 | FALSE |
| 6 | (0.0329) | 0.0937 | 1.1092 | 2.2622 | FALSE |
| 7 | (0.0670) | 0.0709 | 2.9892 | 2.2622 | TRUE |
| 9 | (0.0111) | 0.0772 | 0.4543 | 2.2622 | FALSE |
| 11 | (0.0432) | 0.0847 | 1.6116 | 2.2622 | FALSE |
| 14 | (0.0136) | 0.0637 | 0.6735 | 2.2622 | FALSE |
| 18 | 0.0117 | 0.0440 | 0.8399 | 2.2622 | FALSE |
| 22 | 0.0203 | 0.0599 | 1.0716 | 2.2622 | FALSE |
| 28 | 0.0011 | 0.0374 | 0.0922 | 2.2622 | FALSE |
| 34 | (0.0066) | 0.0460 | 0.4561 | 2.2622 | FALSE |
| 43 | 0.0079 | 0.0295 | 0.8480 | 2.2622 | FALSE |
| 54 | 0.0028 | 0.0302 | 0.2903 | 2.2622 | FALSE |
| 69 | (0.0040) | 0.0145 | 0.8660 | 2.2622 | FALSE |
| 86 | (0.0004) | 0.0535 | 0.0234 | 2.2622 | FALSE |
| 108 | (0.0267) | 0.0239 | 3.5345 | 2.2622 | TRUE |
| 136 | (0.0201) | 0.0330 | 1.9249 | 2.2622 | FALSE |
| 171 | (0.0628) | 0.0383 | 5.1773 | 2.2622 | TRUE |
| 215 | (0.0740) | 0.0245 | 9.5452 | 2.2622 | TRUE |
| 271 | (0.0523) | 0.0297 | 5.5713 | 2.2622 | TRUE |
| 341 | (0.0384) | 0.0334 | 3.6344 | 2.2622 | TRUE |

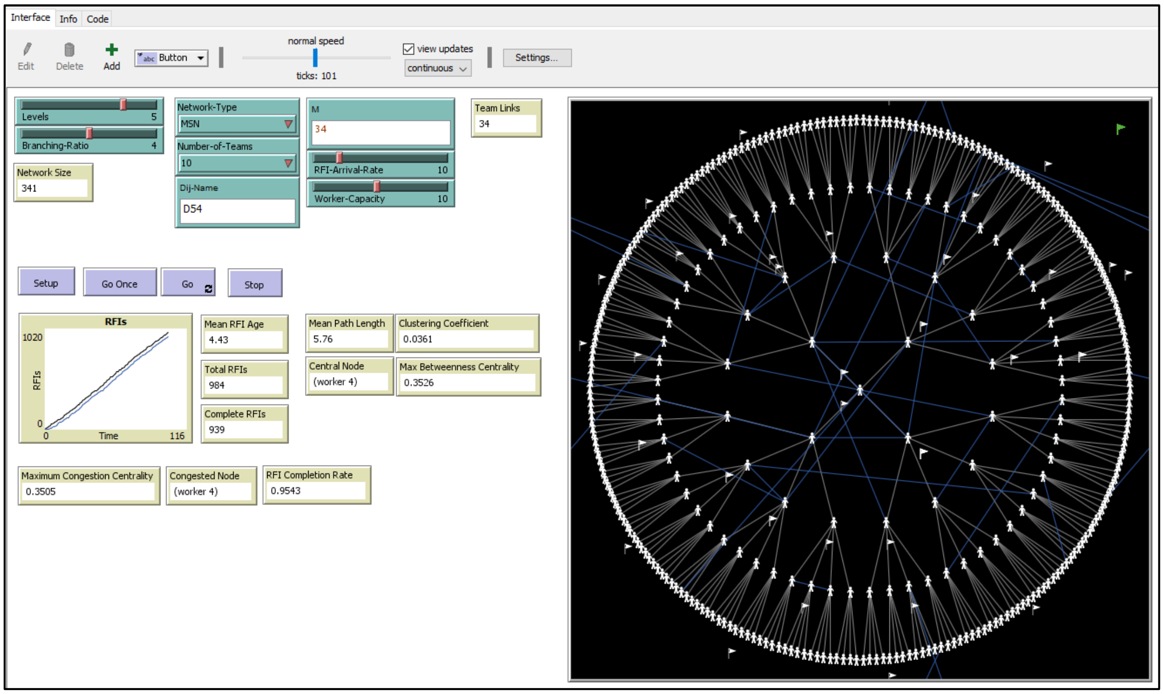
| Multiscale-Matrix Comparison Hypothesis Test | | | | | |
|---|---|---|---|---|---|
| m | $\bar{d}$ | $S_D$ | $\lvert t_0 \rvert$ | t | Reject |
| 1 | 0.0008 | 0.0526 | 0.0476 | 2.2622 | FALSE |
| 2 | (0.0226) | 0.0310 | 2.3056 | 2.2622 | TRUE |
| 4 | (0.0516) | 0.0423 | 3.8578 | 2.2622 | TRUE |
| 5 | (0.1170) | 0.0593 | 6.2359 | 2.2622 | TRUE |
| 6 | (0.1295) | 0.0393 | 10.4112 | 2.2622 | TRUE |
| 7 | (0.1345) | 0.0658 | 6.4643 | 2.2622 | TRUE |
| 9 | (0.1366) | 0.0662 | 6.5243 | 2.2622 | TRUE |
| 11 | (0.1774) | 0.0537 | 10.4580 | 2.2622 | TRUE |
| 14 | (0.1903) | 0.0489 | 12.2944 | 2.2622 | TRUE |
| 18 | (0.2151) | 0.0366 | 18.5998 | 2.2622 | TRUE |
| 22 | (0.2144) | 0.0504 | 13.4531 | 2.2622 | TRUE |
| 28 | (0.2472) | 0.0469 | 16.6680 | 2.2622 | TRUE |
| 34 | (0.2638) | 0.0334 | 24.9404 | 2.2622 | TRUE |
| 43 | (0.2758) | 0.0187 | 46.6760 | 2.2622 | TRUE |
| 54 | (0.2969) | 0.0422 | 22.2621 | 2.2622 | TRUE |
| 69 | (0.3166) | 0.0288 | 34.7517 | 2.2622 | TRUE |
| 86 | (0.3032) | 0.0363 | 26.3844 | 2.2622 | TRUE |
| 108 | (0.3628) | 0.0300 | 38.2035 | 2.2622 | TRUE |
| 136 | (0.3874) | 0.0371 | 32.9975 | 2.2622 | TRUE |
| 171 | (0.4265) | 0.0235 | 57.5019 | 2.2622 | TRUE |
| 215 | (0.4273) | 0.0267 | 50.6990 | 2.2622 | TRUE |
| 271 | (0.4145) | 0.0226 | 58.0469 | 2.2622 | TRUE |
| 341 | (0.4170) | 0.0210 | 62.7764 | 2.2622 | TRUE |

| Random-Military Comparison Hypothesis Test | | | | | |
|---|---|---|---|---|---|
| m | $\bar{d}$ | $S_D$ | $\lvert t_0 \rvert$ | t | Reject |
| 1 | 0.0205 | 0.0363 | 1.7852 | 2.2622 | FALSE |
| 2 | 0.0293 | 0.0408 | 2.2729 | 2.2622 | TRUE |
| 4 | 0.0795 | 0.0398 | 6.3167 | 2.2622 | TRUE |
| 5 | 0.0691 | 0.0423 | 5.1623 | 2.2622 | TRUE |
| 6 | 0.0808 | 0.0585 | 4.3663 | 2.2622 | TRUE |
| 7 | 0.0483 | 0.0209 | 7.2949 | 2.2622 | TRUE |
| 9 | 0.1085 | 0.0473 | 7.2619 | 2.2622 | TRUE |
| 11 | 0.1034 | 0.0500 | 6.5312 | 2.2622 | TRUE |
| 14 | 0.1371 | 0.0730 | 5.9396 | 2.2622 | TRUE |
| 18 | 0.1755 | 0.0549 | 10.1051 | 2.2622 | TRUE |
| 22 | 0.1576 | 0.0505 | 9.8652 | 2.2622 | TRUE |
| 28 | 0.1733 | 0.0475 | 11.5286 | 2.2622 | TRUE |
| 34 | 0.1477 | 0.0527 | 8.8663 | 2.2622 | TRUE |
| 43 | 0.1548 | 0.0493 | 9.9344 | 2.2622 | TRUE |
| 54 | 0.1385 | 0.0403 | 10.8777 | 2.2622 | TRUE |
| 69 | 0.1539 | 0.0652 | 7.4604 | 2.2622 | TRUE |
| 86 | 0.1220 | 0.0713 | 5.4116 | 2.2622 | TRUE |
| 108 | 0.0897 | 0.0561 | 5.0585 | 2.2622 | TRUE |
| 136 | 0.0783 | 0.0257 | 9.6517 | 2.2622 | TRUE |
| 171 | 0.0607 | 0.0434 | 4.4224 | 2.2622 | TRUE |
| 215 | 0.0421 | 0.0313 | 4.2527 | 2.2622 | TRUE |
| 271 | 0.0546 | 0.0286 | 6.0355 | 2.2622 | TRUE |
| 341 | 0.0394 | 0.0312 | 3.9907 | 2.2622 | TRUE |

| Random-Matrix Comparison Hypothesis Test | | | | | |
|---|---|---|---|---|---|
| m | $\bar{d}$ | $S_D$ | $\lvert t_0 \rvert$ | t | Reject |
| 1 | 0.0157 | 0.0425 | 1.1703 | 2.2622 | FALSE |
| 2 | (0.0092) | 0.0494 | 0.5890 | 2.2622 | FALSE |
| 4 | 0.0033 | 0.0471 | 0.2192 | 2.2622 | FALSE |
| 5 | (0.0124) | 0.0729 | 0.5370 | 2.2622 | FALSE |
| 6 | (0.0158) | 0.0686 | 0.7300 | 2.2622 | FALSE |
| 7 | (0.0191) | 0.0745 | 0.8108 | 2.2622 | FALSE |
| 9 | (0.0170) | 0.0609 | 0.8836 | 2.2622 | FALSE |
| 11 | (0.0309) | 0.0795 | 1.2281 | 2.2622 | FALSE |
| 14 | (0.0396) | 0.0512 | 2.4442 | 2.2622 | TRUE |
| 18 | (0.0513) | 0.0280 | 5.7991 | 2.2622 | TRUE |
| 22 | (0.0770) | 0.0532 | 4.5800 | 2.2622 | TRUE |
| 28 | (0.0750) | 0.0353 | 6.7188 | 2.2622 | TRUE |
| 34 | (0.1094) | 0.0295 | 11.7420 | 2.2622 | TRUE |
| 43 | (0.1290) | 0.0325 | 12.5604 | 2.2622 | TRUE |
| 54 | (0.1612) | 0.0500 | 10.1954 | 2.2622 | TRUE |
| 69 | (0.1588) | 0.0288 | 17.4453 | 2.2622 | TRUE |
| 86 | (0.1808) | 0.0591 | 9.6795 | 2.2622 | TRUE |
| 108 | (0.2463) | 0.0273 | 28.4852 | 2.2622 | TRUE |
| 136 | (0.2890) | 0.0154 | 59.3630 | 2.2622 | TRUE |
| 171 | (0.3031) | 0.0229 | 41.7754 | 2.2622 | TRUE |
| 215 | (0.3113) | 0.0258 | 38.2066 | 2.2622 | TRUE |
| 271 | (0.3076) | 0.0160 | 60.8294 | 2.2622 | TRUE |
| 341 | (0.3392) | 0.0198 | 54.2634 | 2.2622 | TRUE |

NetLogo and MATLAB Code and Screen Shots

Information Exchange, Version 1, User Interface




Information Exchange, Version 1, Code

```
extensions [ nw cf csv matrix]
globals [
  Dij-File              ;;name of the file where the Dij matrix is stored
(string)
  Age                   ;;variable used to limit search for new links
  Threshold             ;;scaling factor to adjust threshold for adding
team links
  Nodes                 ;;total number of nodes in the network
  Lambda                ;;network scaling factor related to depth of
common node
  Zeta                  ;;network scaling factor related to distance
between nodes
  Dij                   ;;matrix containing depth of common node between
nodes i and j
  Rho-Max               ;;maximum congestion centrality of network
  Congested-Node        ;;node with maximum congestion centrality
  BC-Max                ;;maximum normalized betweenness centrality of
network
  Central-Node          ;;node with maximum betweenness centrality
  GCC                   ;;global clustering coefficient of network
  MPL                   ;;mean path length of network
]
```

```
breed [workers worker]      ;;Agent breed representing workers in the
organization
breed [RFIs RFI]            ;;Meta-agent for information requests

;;Workers represent the individuals in the organizational hierarchy,
i.e., the individual
;;engineers, designers, supervisors, managers, etc who make up the
organization
workers-own [
  Depth           ;;Worker's depth in hierarchy
  My-Department   ;;Worker's department, equal to worker number at
level 1
  My-Supervisor   ;;Worker's supervisor, common node one level higher
in hierarchy
  My-Team .       ;;Worker’s team assignment, relevant for matrix and
BCCWG organizations
  Capacity        ;;Worker's capacity to perform work in a single tick
  RFI-Queue       ;;Worker's RFI in box (list)
  RFI-Count       ;;Number of RFIs completed by worker
]

;;Requests for Information (RFIs) represent questions or requests to
other workers
RFIs-own [
  R-Originator    ;;Worker who originated the RFI [Agentset]
  R-Target        ;;Worker to whom the RFI was sent [Agentset]
  R-Status        ;;Status of RFI: Open, Answered, or Complete
  R-Age           ;;Age of RFI
]

;;organizational links represent the hierarchical backbone structure of
the organization, defined by levels and branching ratio
undirected-link-breed [organizations organization]

;;team links represent the links added to the organization, according
to network construction algorithm
undirected-link-breed [teams team]

to setup
  clear-all
  reset-ticks

  ;;calculate number of node
  set Nodes (Branching-Ratio ^ Levels - 1) / (Branching-Ratio - 1)

  ;;create organizational hierarchy based on number of levels and
branching
  ;;ratio specified by user
  foreach n-values Levels [ [?1] -> ?1 ] [ [?1] ->
    ifelse ?1 = 0
    [
      ;;create worker 0
      create-workers 1 [
        set Depth 0
        set My-Department 0
        set My-Supervisor worker 0
        set My-Team 0
```

```
      set Capacity Worker-Capacity
      set RFI-Queue [ ]
      set RFI-COunt 0
    ]
  ]
  [
    ;;make new level in hierarchy
    make-level ?1
  ]
]

;;ask workers to set their shape to person and color to white
ask workers [
  set shape "person"
  set color white
]

;;arrange workers in radial layout with worker 0 at center
layout-radial workers organizations (worker 0)

;;create Dij matrix
set Dij matrix:make-constant Nodes Nodes 0
ifelse Network-Type = "Matrix"
[
  set Dij-File (word Dij-Name "_" Number-of-Teams ".csv")
]
[
  set Dij-File word Dij-Name ".csv"
]
show Dij-File
file-open Dij-File
set Dij matrix:from-row-list csv:from-file Dij-File
file-close

;;create team structure based on user inputs
cf:when
cf:case [Network-Type = "Matrix"] [
  ;;create the manager of the project management organization
  create-workers 1 [
    set shape "person"
    set color orange
    set Depth 1
    set My-Department who
    set My-Supervisor worker 0
    set Capacity Worker-Capacity
    set RFI-Queue []
    set RFI-Count 0
    create-organization-with worker 0
  ]
  ;;create project managers, the heads of each project team
  foreach range Number-of-Teams [ [?the-team] ->
    create-workers 1 [
      set Depth 2
      set My-Department Branching-Ratio + 1
      set My-Supervisor worker Nodes
      set My-Team (?the-team + 1)
      set Capacity Worker-Capacity
```

```
        set RFI-Queue []
        set RFI-Count 0
        set shape "person"
        set color red
        create-organization-with worker Nodes
      ]
    ]
    ;;create links to team members
    while [count teams < M] [
      let source one-of workers with [color = red]
      let target one-of workers with [Depth = Levels - 1]
      ask target [
        set My-Team [My-Team] of source
        create-team-with source [set color blue]
      ]
    ]
    layout-radial workers organizations (worker 0)
  ]
  cf:case [Network-Type = "BCCWG"] [
    ;;identify team leads
    foreach range Number-of-Teams [ [?the-team] ->
      ask one-of workers with [My-Team = 0 and (Depth = 1 or Depth =
2)] [
        set My-Team (?the-team + 1)
        set color red
      ]
    ]
    ;;add team members
    while [count teams < M] [
      let source one-of workers with [color = red]
      let target one-of workers with [color != red and Depth > 2]
      ask target [
        set My-Team [My-Team] of source
        create-team-with source [set color blue]
      ]
    ]
  ]
  cf:case [Network-Type = "Random"] [
    set Threshold 0.9999
    set Lambda 1000000
    set Zeta 1000000
    while [count teams < M] [
      let source one-of workers
      let target one-of workers with [who != [who] of source]
      construct-team source target
    ]
  ]
  cf:else [
    ;;default case, use multiscale network
    set Threshold 0.3768
    set Lambda 0.5
    set Zeta 0.5
    while [count teams < M ] [
      let source one-of workers
      let target one-of workers with [who != [who] of source]
      ;;watch target
      construct-team source target
```

```
      ;;if age counter reaches number of Nodes, reduce threshold to
lower value
      set Age Age + 1
      if Age > Nodes ^ 2 [
        set Threshold Threshold ^ 2
        show Threshold
        set Age 0
      ]
    ]
  ]

  ;;calculate and show network parameters
  set BC-Max 2 * max [nw:betweenness-centrality] of workers / ( (Nodes
- 1) * (Nodes - 2) )
  ;;show BC-Max
  set Central-Node max-one-of workers [nw:betweenness-centrality]
  ;;show Central-Node
  set GCC global-clustering-coefficient
  ;;show GCC
  set MPL nw:mean-path-length
  ;;show MPL

end

to go

  ask workers [
    ;;show who
    repeat Capacity [
      if not empty? RFI-Queue [
        ;;If I have RFIs in my RFI Queue, then take RFI actions
        set RFI-Count RFI-Count + 1
        let r first RFI-Queue
        let source [R-Originator] of RFI r
        let target [R-Target] of RFI r
        let here worker who
        ifelse here = target
        [
          ;;I am the target of RFI, so close RFI
          ask RFI r [
             set R-Status "Complete"
             set xcor 23
             set ycor 23
             set color green
          ]
          ;;remove RFI from my RFI queue
          set RFI-Queue remove r RFI-Queue
        ]
        [
          ;;I am not the target of RFI, so pass to next worker on path
          ;;using the Dodds Watts Sabel assumption regarding pseudo-
global knowledge
          ;;show target
          ;;let d-here Depth
          let next one-of organization-neighbors with [Depth < [Depth]
of here]
```

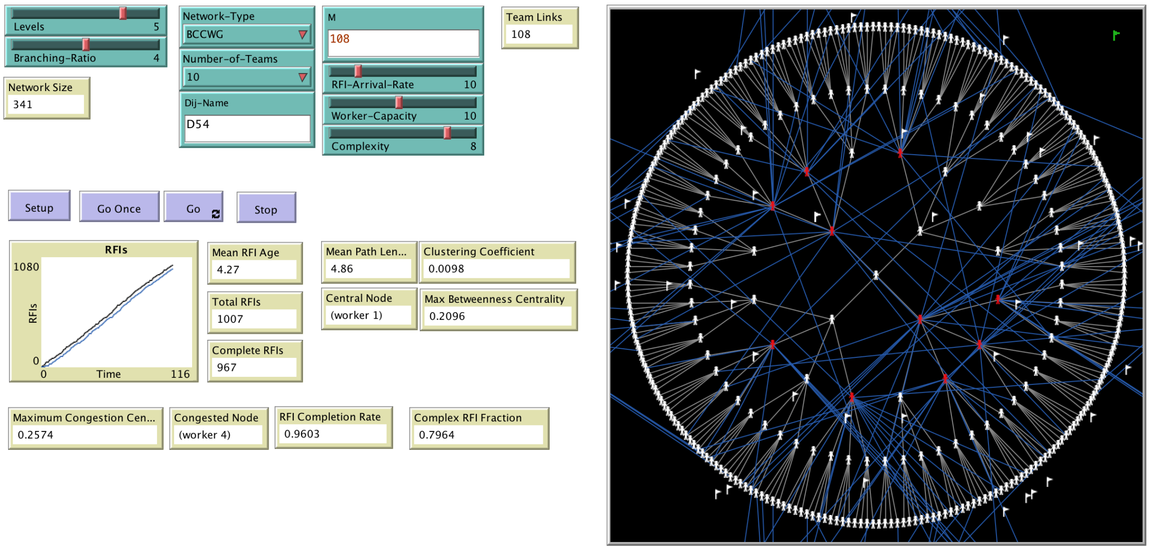
```
        ;;show next
        let t [who] of target
        let da-kt matrix:get Dij who t
        foreach sort link-neighbors [ [?the-neighbor] ->
          if Network-Type = "Matrix" or Network-Type = "BCCWG" [
            ask ?the-neighbor [
              if team-neighbor? target [
                watch-me
                set next ?the-neighbor
                set da-kt [Depth] of ?the-neighbor
              ]
            ]
          ]
          let k [who] of ?the-neighbor
          let dd matrix:get Dij k t
          ;;show k
          ;;show dd
          if dd > da-kt [
            set next ?the-neighbor
            ;;show next
            set da-kt dd
          ]
        ]
        ;;add RFI to next worker's RFI queue
        ask next [
          ifelse empty? RFI-Queue
          [ set RFI-Queue (list r) ]
          [ set RFI-Queue lput r RFI-Queue ]
        ]
        ask RFI r [
          set xcor [xcor] of next
          set ycor [ycor] of next
          face worker 0
          fd -2
        ]
        ;;remove RFI from my RFI queue
        set RFI-queue remove r RFI-Queue
      ]
    ]
  ]
]

;;create new RFIs
make-new-RFIs

;;increment counters
increment

tick
if ticks > 100 [
  ;;calculate congestion probability, rho
  set rho-max maximum-congestion-centrality
  set Congested-Node max-one-of workers [RFI-Count]
  show rho-max
  show Congested-Node

  ;;stop execution
```

```
    stop
  ]
end

;;procedure to add levels (i.e., rows) to hierarchical organizational
structure
to make-level [row]
  let b Branching-Ratio                ;;let b equal the branching
ratio
  let W b ^ (row - 1)                  ;;let W equal the number of
workers in pervious row
  let N (b ^ (row - 1) - 1) / (b - 1)  ;;let N equal number of workers
in all previous rows of hierarchy
  ;;for each of the workers in the previous row
  foreach n-values W [ [?1] -> ?1 ] [ [?1] ->
    ;;show ? + N
    ;;create b new workers and link to worker ?+N in previous row
    create-workers b [
      set Depth row
      ifelse row = 1 [
        set My-Department who
      ]
      [
        set My-Department [My-Department] of worker (?1 + N)
      ]
      set My-Supervisor worker (?1 + N)
      set My-Team 0
      set Capacity ceiling (Worker-Capacity / row)
      set RFI-Queue [ ]
      create-organization-with worker (?1 + n)
    ]
  ]

end

;;procedure to create team links based on Watts' stochastic rule
to construct-team [source target]
  let i [who] of source
  let d-source [Depth] of source
  let j [who] of target
  let d-target [Depth] of target
  let da matrix:get Dij i j
  let d1 abs (d-source - da)
  let d2 abs (d-target - da)
  ;; if d1+d2 < 2, then no new link created by procedure
  if (d1 + d2) >= 2 [
    let x12 (d1 ^ 2 + d2 ^ 2 - 2) ^ (1 / 2)
    let P exp ((- da) / Lambda) * exp ((- x12) / Zeta)
    ;;show P
    if random-float Threshold < P [
      ask source [
        if not team-neighbor? target [
            create-team-with target [
              set color blue
            ]
            set Age 0
        ]
```

```
      ]
    ]
  ]

end

;;procedure to make new RFIs
;;New RFIs arrive according to a random Poisson process with mean equal
to RFI-Arrival-Rate
;;Each new RFI is randomly assigned to an originating and target worker
to make-new-RFIs
  create-RFIs random-poisson RFI-Arrival-Rate [
    set R-Status "Open"
    let r who
    let s one-of workers with [color = white]
    let t one-of workers with [who != [who] of s and color = white]
    set R-Originator s
    set R-Target t
    set R-Age 0
    ;;Put artifact in originator's Artifact Queue
    ask s [
      ifelse empty? RFI-Queue
      [ set RFI-Queue (list r) ]
      [ set RFI-Queue lput r RFI-Queue ]
    ]
    ;;Place artifact near originator
    set xcor [xcor] of s
    set ycor [ycor] of s
    face worker 0
    fd -1
    set color white
    set shape "flag"
  ]

end

;;procedure to increment counters
to increment

  ;;Increment age of open RFIs
  ask RFIs with [R-Status != "Complete"] [
    set R-Age R-Age + 1
  ]
end

;;procedure to calculate and report global clustering coefficient,
using method in nw extensions documentation
to-report global-clustering-coefficient
  let closed-triplets sum [ nw:clustering-coefficient * count my-links
* (count my-links - 1 ) ] of workers
  let triplets sum [ count my-links * (count my-links - 1) ] of workers
  report closed-triplets / triplets
end

;;procedure to calculate and report rho-max, the maximum congestion
centrality
to-report maximum-congestion-centrality
```

```
  let r-max max [RFI-Count] of workers
  report r-max / (RFI-Arrival-Rate * ticks)
end
```

## Information Exchange, Version 2, User Interface




## Information Exchange, Version 2, Code

```
extensions [ nw cf csv matrix]
globals [
  Dij-File            ;;name of the file where the Dij matrix is stored
(string)
  Age                 ;;variable used to limit search for new links
  Threshold           ;;scaling factor to adjust threshold for adding
team links
  Nodes               ;;total number of nodes in the network
  Lambda              ;;network scaling factor related to depth of
common node
  Zeta                ;;network scaling factor related to distance
between nodes
  Dij                 ;;matrix containing depth of common node between
nodes i and j
  Rho-Max             ;;maximum congestion centrality of network
  Congested-Node      ;;node with maximum congestion centrality
  BC-Max              ;;maximum normalized betweenness centrality of
network
  Central-Node        ;;node with maximum betweenness centrality
  GCC                 ;;global clustering coefficient of network
  MPL                 ;;mean path length of network
]

breed [workers worker]       ;;Agent breed representing workers in the
organization
breed [RFIs RFI]             ;;Meta-agent for information requests

;;Workers represent the individuals in the organizational hierarchy,
i.e., the individual
```

```
;;engineers, designers, supervisors, managers, etc who make up the
organization
workers-own [
  Depth           ;;Worker's depth in hierarchy
  My-Department   ;;Worker's department, equal to worker number at
level 1
  My-Supervisor   ;;Worker's supervisor, common node one level higher
in hierarchy
  My-Team .       ;;Worker’s team assignment, relevant for matrix and
BCCWG organizations
  Capacity        ;;Worker's capacity to perform work in a single tick
  RFI-Queue       ;;Worker's RFI in box (list)
  RFI-Count       ;;Number of RFIs completed by worker
]

;;Requests for Information (RFIs) represent questions or requests to
other workers
RFIs-own [
  R-Originator    ;;Worker who originated the RFI [Agentset]
  R-Target        ;;Worker to whom the RFI was sent [Agentset]
  R-Status        ;;Status of RFI: Open, Answered, or Complete
  R-Age           ;;Age of RFI
]

;;organizational links represent the hierarchical backbone structure of
the organization, defined by levels and branching ratio
undirected-link-breed [organizations organization]

;;team links represent the links added to the organization, according
to network construction algorithm
undirected-link-breed [teams team]

to setup
  clear-all
  reset-ticks

  ;;calculate number of node
  set Nodes (Branching-Ratio ^ Levels - 1) / (Branching-Ratio - 1)

  ;;create organizational hierarchy based on number of levels and
branching
  ;;ratio specified by user
  foreach n-values Levels [ [?1] -> ?1 ] [ [?1] ->
    ifelse ?1 = 0
    [
      ;;create worker 0
      create-workers 1 [
        set Depth 0
        set My-Department 0
        set My-Supervisor worker 0
        set My-Team 0
        set Capacity Worker-Capacity
        set RFI-Queue [ ]
        set RFI-COunt 0
      ]
    ]
    [
```

```
    ;;make new level in hierarchy
    make-level ?1
  ]
]

;;ask workers to set their shape to person and color to white
ask workers [
  set shape "person"
  set color white
]

;;arrange workers in radial layout with worker 0 at center
layout-radial workers organizations (worker 0)

;;create Dij matrix
set Dij matrix:make-constant Nodes Nodes 0
ifelse Network-Type = "Matrix"
[
  set Dij-File (word Dij-Name "_" Number-of-Teams ".csv")
]
[
  set Dij-File word Dij-Name ".csv"
]
;;show Dij-File
file-open Dij-File
set Dij matrix:from-row-list csv:from-file Dij-File
file-close

;;create team structure based on user inputs
cf:when
cf:case [Network-Type = "Matrix"] [
  ;;create the manager of the project management organization
  create-workers 1 [
    set shape "person"
    set color orange
    set Depth 1
    set My-Department who
    set My-Supervisor worker 0
    set Capacity Worker-Capacity
    set RFI-Queue []
    set RFI-Count 0
    create-organization-with worker 0
  ]
  ;;create project managers, the heads of each project team
  foreach range Number-of-Teams [ [?the-team] ->
    create-workers 1 [
      set Depth 2
      set My-Department Branching-Ratio + 1
      set My-Supervisor worker Nodes
      set My-Team (?the-team + 1)
      set Capacity Worker-Capacity
      set RFI-Queue []
      set RFI-Count 0
      set shape "person"
      set color red
      create-organization-with worker Nodes
    ]
```

```
    ]
    ;;create links to team members
    while [count teams < M] [
      let source one-of workers with [color = red]
      let target one-of workers with [Depth = Levels - 1]
      ask target [
        set My-Team [My-Team] of source
        create-team-with source [set color blue]
      ]
    ]
    layout-radial workers organizations (worker 0)
  ]
  cf:case [Network-Type = "BCCWG"] [
    ;;identify team leads
    foreach range Number-of-Teams [ [?the-team] ->
      ask one-of workers with [My-Team = 0 and (Depth = 1 or Depth =
2)] [
        set My-Team (?the-team + 1)
        set color red
      ]
    ]
    ;;add team members
    while [count teams < M] [
      let source one-of workers with [color = red]
      let target one-of workers with [color != red and Depth > 2]
      ask target [
        set My-Team [My-Team] of source
        create-team-with source [set color blue]
      ]
    ]
  ]
  cf:case [Network-Type = "Random"] [
    set Threshold 0.9999
    set Lambda 1000000
    set Zeta 1000000
    while [count teams < M] [
      let source one-of workers
      let target one-of workers with [who != [who] of source]
      construct-team source target
    ]
  ]
  cf:else [
    ;;default case, use multiscale network
    set Threshold 0.3768
    set Lambda 0.5
    set Zeta 0.5
    while [count teams < M ] [
      let source one-of workers
      let target one-of workers with [who != [who] of source]
      ;;watch target
      construct-team source target

      ;;if age counter reaches number of Nodes, reduce threshold to
lower value
      set Age Age + 1
      if Age > Nodes ^ 2 [
        set Threshold Threshold ^ 2
```

```
      show Threshold
      set Age 0
    ]
  ]
]

;;calculate and show network parameters
set BC-Max 2 * max [nw:betweenness-centrality] of workers / ( (Nodes
- 1) * (Nodes - 2) )
;;show BC-Max
set Central-Node max-one-of workers [nw:betweenness-centrality]
;;show Central-Node
set GCC global-clustering-coefficient
;;show GCC
set MPL nw:mean-path-length
;;show MPL

end

to go

  ask workers [
    ;;show who
    repeat Capacity [
      if not empty? RFI-Queue [
        ;;If I have RFIs in my RFI Queue, then take RFI actions
        set RFI-Count RFI-Count + 1
        let r first RFI-Queue
        let source [R-Originator] of RFI r
        let target [R-Target] of RFI r
        let here worker who
        ifelse here = target
        [
          ;;I am the target of RFI, so close RFI
          ask RFI r [
             set R-Status "Complete"
             set xcor 23
             set ycor 23
             set color green
          ]
          ;;remove RFI from my RFI queue
          set RFI-Queue remove r RFI-Queue
        ]
        [
          ;;I am not the target of RFI, so pass to next worker on path
          ;;using the Dodds Watts Sabel assumption regarding pseudo-
global knowledge
          ;;show target
          ;;let d-here Depth
          let next one-of organization-neighbors with [Depth < [Depth]
of here]
          ;;show next
          let t [who] of target
          let da-kt matrix:get Dij who t
          foreach sort link-neighbors [ [?the-neighbor] ->
            if Network-Type = "Matrix" or Network-Type = "BCCWG" [
              ask ?the-neighbor [
```

```
              if team-neighbor? target [
                set next ?the-neighbor
                set da-kt [Depth] of ?the-neighbor
              ]
            ]
          ]
          let k [who] of ?the-neighbor
          let dd matrix:get Dij k t
          ;;show k
          ;;show dd
          if dd > da-kt [
            set next ?the-neighbor
            ;;show next
            set da-kt dd
          ]
        ]
        ;;add RFI to next worker's RFI queue
        ask next [
          ifelse empty? RFI-Queue
          [ set RFI-Queue (list r) ]
          [ set RFI-Queue lput r RFI-Queue ]
        ]
        ask RFI r [
          set xcor [xcor] of next
          set ycor [ycor] of next
          face worker 0
          fd -2
        ]
        ;;remove RFI from my RFI queue
        set RFI-queue remove r RFI-Queue
      ]
    ]
  ]
]

;;create new RFIs
make-new-RFIs

;;increment counters
increment

tick
if ticks > 100 [
  ;;calculate congestion probability, rho
  set rho-max maximum-congestion-centrality
  set Congested-Node max-one-of workers [RFI-Count]
  show rho-max
  show Congested-Node

  ;;stop execution
  stop
]
end

;;procedure to add levels (i.e., rows) to hierarchical organizational
structure
to make-level [row]
```

```
  let b Branching-Ratio                ;;let b equal the branching
ratio
  let W b ^ (row - 1)                  ;;let W equal the number of
workers in pervious row
  let N (b ^ (row - 1) - 1) / (b - 1)  ;;let N equal number of workers
in all previous rows of hierarchy
  ;;for each of the workers in the previous row
  foreach n-values W [ [?1] -> ?1 ] [ [?1] ->
    ;;show ? + N
    ;;create b new workers and link to worker ?+N in previous row
    create-workers b [
      set Depth row
      ifelse row = 1 [
        set My-Department who
      ]
      [
        set My-Department [My-Department] of worker (?1 + N)
      ]
      set My-Supervisor worker (?1 + N)
      set My-Team 0
      set Capacity ceiling (Worker-Capacity / row)
      set RFI-Queue [ ]
      create-organization-with worker (?1 + n)
    ]
  ]

end

;;procedure to create team links based on Watts' stochastic rule
to construct-team [source target]
  let i [who] of source
  let d-source [Depth] of source
  let j [who] of target
  let d-target [Depth] of target
  let da matrix:get Dij i j
  let d1 abs (d-source - da)
  let d2 abs (d-target - da)
  ;; if d1+d2 < 2, then no new link created by procedure
  if (d1 + d2) >= 2 [
    let x12 (d1 ^ 2 + d2 ^ 2 - 2) ^ (1 / 2)
    let P exp ((- da) / Lambda) * exp ((- x12) / Zeta)
    ;;show P
    if random-float Threshold < P [
      ask source [
        if not team-neighbor? target [
            create-team-with target [
              set color blue
            ]
            set Age 0
        ]
      ]
    ]
  ]

end

;;procedure to make new RFIs
```

```
;;New RFIs arrive according to a random Poisson process with mean equal
to RFI-Arrival-Rate
;;new RFIs are assigned at random to an originator
;;but target depends on complexity.  If complexity is higher, task is
less likely to
;;be decomposable, and target selected at random from other workers.
;;when complexity is lower, task is decomposable and target selected
from other
;;workers in same department
to make-new-RFIs
  create-RFIs random-poisson RFI-Arrival-Rate [
    set R-Status "Open"
    let r who
    let s one-of workers with [color = white and who != 0]
    let t []
    ifelse random 10 < Complexity
    [
      set t one-of workers with [who != [who] of s and color = white]
    ]
    [
      set t one-of workers with [My-Department = [My-Department] of s]
    ]
    set R-Originator s
    set R-Target t
    set R-Age 0
    ;;Put artifact in originator's Artifact Queue
    ask s [
      ifelse empty? RFI-Queue
      [ set RFI-Queue (list r) ]
      [ set RFI-Queue lput r RFI-Queue ]
    ]
    ;;Place artifact near originator
    set xcor [xcor] of s
    set ycor [ycor] of s
    face worker 0
    fd -1
    set color white
    set shape "flag"
  ]

end

;;procedure to increment counters
to increment

  ;;Increment age of open RFIs
  ask RFIs with [R-Status != "Complete"] [
    set R-Age R-Age + 1
  ]
end

;;procedure to calculate and report global clustering coefficient,
using method in nw extensions documentation
to-report global-clustering-coefficient
  let closed-triplets sum [ nw:clustering-coefficient * count my-links
* (count my-links - 1 ) ] of workers
  let triplets sum [ count my-links * (count my-links - 1) ] of workers
```

```
  report closed-triplets / triplets
end

;;procedure to calculate and report rho-max, the maximum congestion
centrality
to-report maximum-congestion-centrality
  let r-max max [RFI-Count] of workers
  report r-max / (RFI-Arrival-Rate * ticks)
end
```

MATLAB Code

```
DoddsWattsSabel_Dij.m
Calculate and return the Dodd-Watts-Sabel Dij matrix
% Source: Dodds, Watts, Sabel, "Information exchange and the robustness
of organizational networks", PNAS 100 (21): 12516-12521
% INPUTs: number of levels in hierarchy (L) and branching ratio (b)
% OUTPUTs: Dij matrix containing depth of highest common node between
nodes
% i and j, an N x N matrix where N=(b^L-1)/(b-1)
% Other routines used: edgeL2adj.m, canonical_nets.m, dijkstra.m

function Dij_Matrix = DoddsWattsSabel_Dij(L,b)

% calculate number of nodes
N = (b^L - 1)/(b - 1);

% construct a tree with N nodes and branch factor b
adj0=edgeL2adj( canonical_nets(N,'tree',b) ); % backbone adjacency
matrix

adj=adj0;
edges=0;

for i=1:N
    for j=i:N
        % find di, dj and Dij
        [d1,path1]=dijkstra(adj0,i,1); % adjacency, source, target
        [d2,path2]=dijkstra(adj0,j,1);

        for p=1:length(path1)
            p1=path1(p);
            p2=find(path2==p1);
            if length(p2)>0 % if p1 in path2
                % di+dj is the distance from i to j
                % Dij is level of highest common node on path
                di=p-1; dj=p2-1;
                Dij=length(path1(p:length(path1)))-1;
                break
            end
        end
        Dij_Matrix(i,j)=int16(Dij);
        Dij_Matrix(j,i)=int16(Dij);
    end
end
```

```
DoddsWattsSabel2.m
% Creates a Dodds-Watts-Sabel network of team links on a pure tree
% hierarchy
% Non-backbone edges are added with probability P(i,j)=e^(-
Dij/lambda)*e^(-xij/ksi),
%     where Dij is the level of the lowest common ancestor and xij is
the "organizational" distance
% Source: Dodds, Watts, Sabel, "Information exchange and the robustness
of organizational networks", PNAS 100 (21): 12516-12521
% Note: alternative method using Dij matrix as input
% INPUTs
%   Number of levels, L
%   Branching ratio, b
%   Dij matrix, Dij_Matrix
%   Number of team links, m
%   Model parameters lamda and ksi in [0,inf)
% OUTPUTs: adjacency matrix of randomized hierarchy, NxN
% Other routines used: edgeL2adj.m, canonical_nets.m, dijkstra.m

function adj = DoddsWattsSabel2(L,b,Dij_Matrix,m,lamda,ksi)

% calculate number of nodes, N
N=(b^L-1)/(b-1);

% construct a tree with N nodes and branch factor b
adj0=edgeL2adj( canonical_nets(N,'tree',b) ); % backbone adjacency
matrix

adj=adj0;
edges=0;
while edges<m
    % pick two nodes at random
    ind1=randi(N); ind2=randi(N);

    % if same node or already connected, keep going
    if ind1==ind2 | adj(ind1,ind2)>0 | adj(ind2,ind1)>0; continue; end

    % find di,dj and Dij
    [d1,path1]=dijkstra(adj0,ind1,1);  % adjacency, source, target
    [d2,path2]=dijkstra(adj0,ind2,1);
    Dij=Dij_Matrix(ind1,ind2);
    di=Dij-d1;
    dj=Dij-d2;

    % calculate distance Xij
    xij=sqrt(di^2+dj^2-2);

    % connect ind1 and ind2 with prob. e^(-Dij/lam)e^(-xij/ksi))
    if rand<exp(-Dij/lamda)*exp(-xij/ksi)
        adj(ind1,ind2)=1;
        adj(ind2,ind1)=1;
        edges=edges+1;
    end

end
```

DWS_Model2.m

```
% Calculates the maximum congestion centrality and identifies the
congested
% node for a network described by adjacency matrix, using pseudo-global
% routing algorithm
% Inputs
%   Adjacency matrix, adj
%   Dij matrix, Dij
%   RFI Arrival Rate, RFI_Arrival_Rate
% Outputs
%   Maximum congestion centrality, rho_max
%   Congested Node, C_Node
% Other routines used: *****

function [rho_max,C_Node,RFI_Completion] =
DWS_Model2(adj,Dij,RFI_Arrival_Rate)

% calculate number of nodes, N
N = length(adj);
% create graph of adjacency matrix
G = graph(adj);

Count_RFIs = 0;
% initialize Nodes data structure
%   Node(i).Q is the RFI queue for node i
%   Node(i).C is the number of RFIs processed by node i
for i = 1:N
    Node(i).Q = [];
    Node(i).C = 0;
end
% RFI Data Structure
%   RFI(r).Source is the source node for RFI r
%   RFI(r).Target is the target node for RFI r
%   RFI(r).Status is the status of RFI r
%       0 = open, 1 = closed
%   RFI(r).Age is the age of RFI r
%       age is no longer incremented once RFI is closed

for ticks = 1:100
    % create RFIs
    for r = 1:poissrnd(RFI_Arrival_Rate)
        % new RFIs arrive according to Poisson distribution
        % select source and target nodes, s & t
        s = randi(N);
        t = randi(N);
        if s==t
            continue
        end
        Count_RFIs = Count_RFIs+1;
        % set initial parameters for RFI
        RFI(Count_RFIs).Source = s;
        RFI(Count_RFIs).Target = t;
        RFI(Count_RFIs).Status = 0;
```

```
        RFI(Count_RFIs).Age = 0;
        % assign rfi to source node
        l = length(Node(s).Q);
        Node(s).Q(l+1) = Count_RFIs;
    end

    % process RFIs
    for n = 1:N
        while length(Node(n).Q)>0
            % increment Node(n) RFI count
            Node(n).C = Node(n).C+1;
            % get next RFI from Node(n).Q
            r = Node(n).Q(1);
            s = RFI(r).Source;
            t = RFI(r).Target;
            if n==t
                % Node(n) is target of RFI => mark RFI complete
                RFI(r).Status = 1;
                % remove RFI from queue
                Node(n).Q(1) = [];
            else
                % Node(n) is not target of RFI => pass to next node
                % using pseudo-global knowledge algorithm
                MyNeighbors = neighbors(G,n);
                Near = nearest(G,n,1);
                next = Near(1);
                dd = Dij(next,t);
                for k=1:length(MyNeighbors)
                    Neighbor_k = MyNeighbors(k);
                    if Dij(Neighbor_k,t)>dd
                        dd = Dij(Neighbor_k,t);
                        next = MyNeighbors(k);
                    end
                end
                l = length(Node(next).Q);
                Node(next).Q(l+1) = r;
                % remove RFI from queue
                Node(n).Q(1) = [];
            end
        end
    end

    % increment RFI age
    for r=1:Count_RFIs
        if RFI(r).Status==0
            RFI(r).Age = RFI(r).Age+1;
        end
    end
end

% calculate rho-max and congested node (C_Node)
for n=1:N
    R_Count(n)=Node(n).C;
end
[R_max,Indices]=max(R_Count);
rho_max = R_max/(RFI_Arrival_Rate * ticks);
C_Node = Indices(1);
```

```
% calculate RFI completion
RFI_Done = 0;
for r=1:Count_RFIs
    if RFI(r).Status==1
        RFI_Done = RFI_Done+1;
    end
end
RFI_Completion=RFI_Done/Count_RFIs;
```

# APPENDIX 2: Artifact Processing Model

Elements of the *Artifacts* Model

| Agents |
|---|
| Workers, representing the individuals within the hierarchy.<br>Depth (level);<br>Department (major division of the hierarchy);<br>Supervisor (immediate superior in the hierarchy);<br>Team (team assignment, for matrix and military staff organizational networks);<br>Capacity (the amount of work the worker can perform in one time step);<br>Artifact Queue (list of artifacts to be processed);<br>Artifact Count (number of artifacts processed by worker);<br>Hold Queue (list of artifacts placed on hold while awaiting RFI response);<br>RFI queue (list of RFIs to be processed); and<br>RFI count (number of RFIs processed by worker).<br><br>Artifacts representing work products.<br>Originator (worker who originated the artifact)'<br>Status (status of the artifact: open, hold or complete); and<br>Age (age of artifact).<br><br>Requests for Information (RFIs), representing messages passed between workers.<br>Artifact (the artifact to which the RFI is related);<br>Originator (worker who originated the RFI);<br>Target (worker to whom the RFI was sent);<br>Status (status of the RFI: open, answered, or complete); and<br>Age (age of RFI). |
| **Links** |
| Organization Links, representing the hierarchical structure.<br>Team Links, representing the cross-functional team links added to the hierarchy. |
| **Environment** |
| The environment is defined by the backbone hierarchical network, the team links added to the hierarchical backbone, and the task environment. The Artifact model describes the task environment in terms of the rate and distribution of artifacts to be processed and messages that must be exchanged to accomplish cross-functional collaboration.<br>The artifact rate, $\mu_A$, is the average number of artifacts originated by each node at each time stem, and $\mu_A N$ is the total number of artifacts originated across the network at each time step.<br>Artifact routing follows the functional hierarchy. Workers at the lowest level of the hierarchy originate artifacts and then pass them up the functional chain of command to a manager near the top of the hierarchy for approval.<br>For simple tasks, the originating worker likely has sufficient information to complete the artifact without the need for cross-functional collaboration. For complex tasks, however, the worker likely lacks sufficient information and requires additional |

information from other workers. In this case, the originating worker places the artifact on hold and originates a request for information (RFI) to acquire the additional information required to complete the artifact.

RFIs pass from source to target through a chain of intermediate nodes as with messages in the information exchange model. Upon receipt, the RFI target provides the information requested and returns the RFI directly to the originator. When the originator receives an answered RFI, he completes the associated artifact and routes if for approval.

Complexity affects the rate and distribution of RFIs. At low complexity, few RFIs are created, and because tasks are decomposable, RFIs are routed to other workers in the same functional organization. At high complexity, many RFIs are created. Since tasks are not decomposable, RFIs are routed to other workers across the organization. The *Artifact* model uses the same qualitative complexity scale used in the information exchange models implement in phase one.

**Time Behavior**

At each time step, workers process artifacts and information requests up to their capacity.

If a given node has only RFIs or artifacts available, it processes them, but if both are available, it decides which to process by comparing a random number to an artifact preference rating, in the range [0,1].

When the artifact preference rating is higher, it is more likely the node will select an artifact than an RFI. An artifact rating of 0.5 represents a "coin flip," with the node choosing RFIs half the time, and artifacts the other half.

**Inputs**

Network parameters: Levels, branching ratio;
Network type: random, multiscale, matrix or military staff (BCCWG);
Number of Teams, for matrix and military staff organizational networks;
$D_{ij}$ Name, the file containing a matrix of the depths of lowest common ancestors;
Team links added, *m*;
Task Arrival Rate;
Worker Capacity;
Complexity; and
Artifact Preference

**Outputs**

The principal output is artifact completion rate, defined as the number of artifacts completed divided by the total number of artifacts. If the organizational network is able to keep pace with the artifact and information processing work load, the artifact completion rate will tend to unity with a small deviation resulting from the artifacts in process at any given time step. Additional outputs include:
RFI arrival and completion rates;
Mean age of RFIs and artifacts;
RFI, artifact and effective congestion centralities and congested nodes;
Network parameters: mean path length and global clustering coefficient.

Verification and Validation of the *Artifacts* Model

<table>
<tr><th colspan="4">Verification of the Artifacts Model</th></tr>
<tr><th colspan="4">Documentation</th></tr>
<tr><td colspan="4">Model documentation consistent with conceptual model<br>Code comments adequately identify implementation of conceptual model</td></tr>
<tr><td colspan="4">Comments</td></tr>
<tr><td colspan="4">Documentation and code comments sufficient.</td></tr>
<tr><th colspan="4">Programmatic Testing</th></tr>
<tr><th>Item Tested</th><th>Method</th><th>Result</th><th>Comments</th></tr>
<tr><td>Proper creation and assignment of artifacts</td><td>Observe artifact arrival in viewer.</td><td>Satisfactory</td><td></td></tr>
<tr><td>Proper placing on hold of artifacts when complexity high</td><td>Watch artifacts and worker hold queues.</td><td>Satisfactory</td><td></td></tr>
<tr><td>Proper creation of RFIs and selection of target nodes.</td><td>Watch artifacts and RFIs.</td><td>Satisfactory</td><td></td></tr>
<tr><td>Proper routing of artifacts along functional hierarchy</td><td>Observe artifact routing in viewer.</td><td>Satisfactory</td><td></td></tr>
<tr><td>Proper routing of RFIs</td><td>Watch artifacts in viewer.</td><td>Satisfactory</td><td></td></tr>
</table>

<table>
<tr><th colspan="3">Test Cases</th></tr>
<tr><th>Method</th><th>Description</th><th>Result</th></tr>
<tr><td>Edge cases: low and high complexity</td><td>Calculate artifact completion rates at low and high complexity.</td><td>Networks demonstrate comparable performance and are congestion free at low complexity but exhibit divergent behavior and experience congestion failure at high complexity.</td></tr>
<tr><th colspan="3">Conclusion</th></tr>
<tr><td colspan="3">Model correctly implements artifact creation and routing and also correctly implements the relationship between artifacts and RFIs. Elements common to information exchange models previously verified.</td></tr>
</table>

<table>
<tr><th colspan="2">Validation of the Artifacts Model</th></tr>
<tr><th colspan="2">Face Validation</th></tr>
<tr><th>Micro-Face Validation</th><th>Macro-Face Validation</th></tr>
<tr><td>Principal inputs to model are the organizational networks and the task environment. The organizational networks are based on real-world organizations or ideal classes described in the literature (i.e., random and multiscale). Model implements the creation of artifacts and sharing of information in design organizations.</td><td>The model realistically depicts the flow of artifacts and information in organizational networks, combining both formal passing of information up and down a hierarchy, and informal passing through team relationships.</td></tr>
<tr><th colspan="2">Empirical Validation</th></tr>
<tr><th>Empirical Input Validation</th><th>Empirical Output Validation</th></tr>
<tr><td>The hierarchical backbone is described by number of levels and branching ratio. This is an idealization in that real organizations exhibit irregularities in both level and branching, but the idealization is reasonable.<br><br>The model rates complexity on a scale of 1 to 10. Use of a simple scale is not meant to represent a quantitative comparison of system complexity, but instead differentiates systems of low and high complexity in a numerical fashion that is easy to implement in a model.<br><br>The model uses an artifact preference rating to control worker selection between artifacts and RFIs when both are present. This is a reasonable representation of real-world behavior.</td><td>Empirical validation relies on stylized facts, primarily the expectation that organizational networks will exhibit different performance at low and high complexity. A designed experiment confirms that all networks perform well at low complexity, but experience congestion failures at high complexity, with multiscale and military staff organizational networks out-performing random and matrix organizational networks.</td></tr>
<tr><th colspan="2">Conclusion</th></tr>
<tr><td colspan="2">The model is considered valid for the purpose of evaluating the factors and causes leading to the inability of design organizations to manage the complexity associated with the development of large engineered systems.</td></tr>
</table>

Validation Experiment Results from Minitab

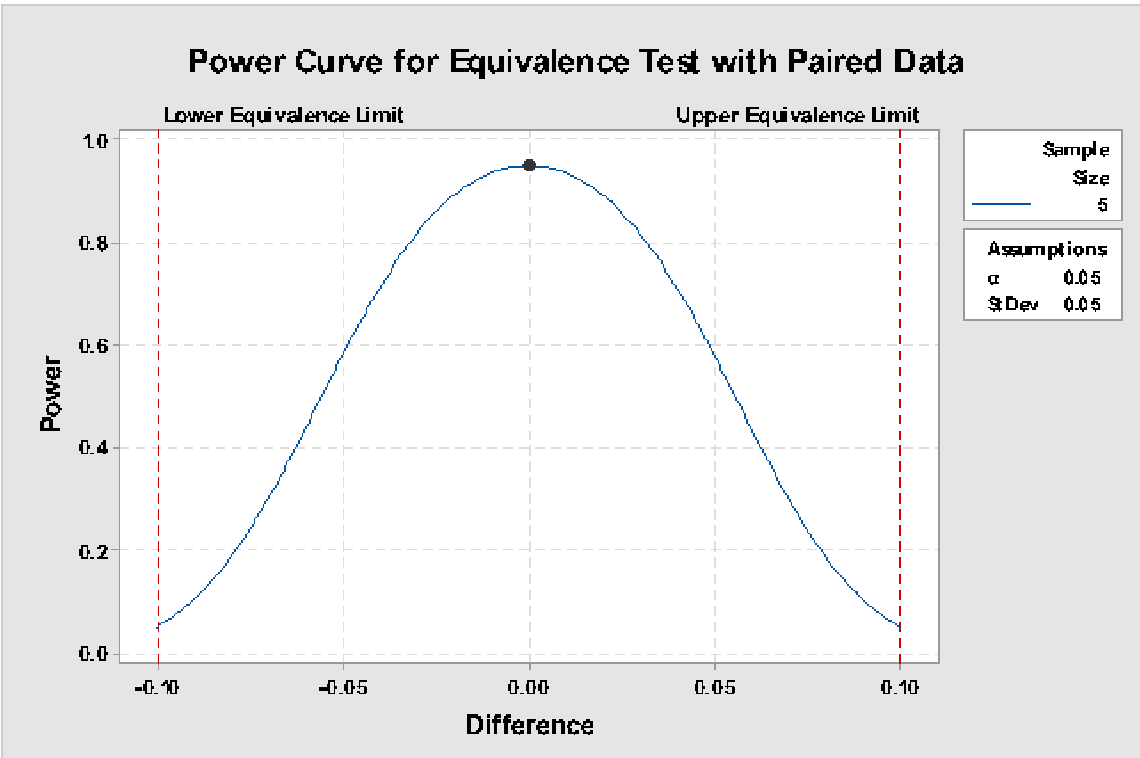
## Power and Sample Size

2-Sample t Test
Testing mean 1 = mean 2 (versus ≠)
Calculating power for mean 1 = mean 2 + difference
α = 0.05  Assumed standard deviation = 0.03

### Results

| Difference | Sample Size | Target Power | Actual Power |
|---|---|---|---|
| 0.1 | 4 | 0.9 | 0.972660 |

*The sample size is for each group.*




## Factor Information

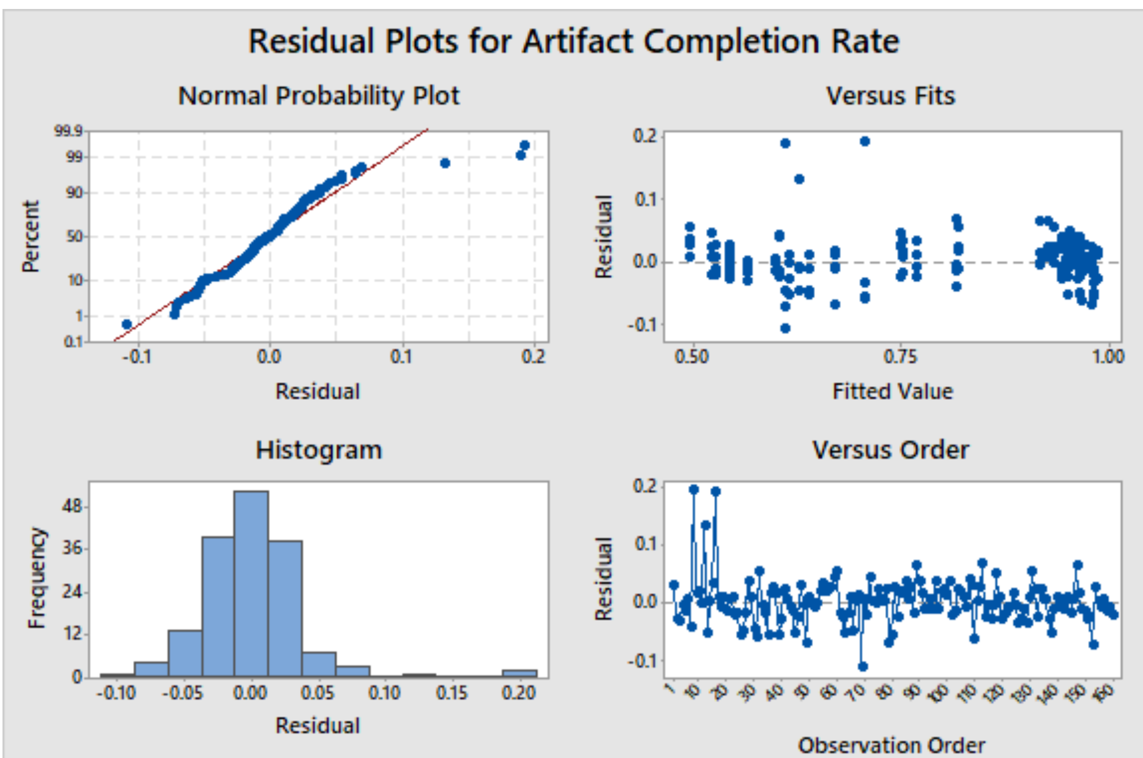
| Factor | Type | Levels | Values |
|---|---|---|---|
| Network Type | Fixed | 4 | Random, Multi-Scale, Matrix, BCCWG |
| Complexity | Fixed | 2 | Low, High |
| Team Links | Fixed | 5 | 3, 11, 34, 108, 341 |

## Analysis of Variance

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Network Type | 3 | 0.12420 | 0.04140 | 23.36 | 0.000 |
| Complexity | 1 | 4.00797 | 4.00797 | 2261.27 | 0.000 |
| Team Links | 4 | 0.25816 | 0.06454 | 36.41 | 0.000 |
| Network Type*Complexity | 3 | 0.13059 | 0.04353 | 24.56 | 0.000 |
| Network Type*Team Links | 12 | 0.04832 | 0.00403 | 2.27 | 0.012 |
| Complexity*Team Links | 4 | 0.21312 | 0.05328 | 30.06 | 0.000 |
| Error | 132 | 0.23396 | 0.00177 | | |
| Lack-of-Fit | 12 | 0.03225 | 0.00269 | 1.60 | 0.101 |
| Pure Error | 120 | 0.20171 | 0.00168 | | |
| Total | 159 | 5.01632 | | | |

## Model Summary

| S | R-sq | R-sq(adj) | R-sq(pred) |
|---|---|---|---|
| 0.0421004 | 95.34% | 94.38% | 93.15% |



Hypothesis Testing for Artifact Characterization Experiment

The following hypothesis tests compare artifact completion rates for random and multiscale networks to military staff and matrix networks at high complexity using the paired data test, as before.

| Multiscale-Military Comparison Hypothesis Test | | | | | |
|---|---|---|---|---|---|
| m | $\bar{d}$ | $S_D$ | $\lvert t_0 \rvert$ | t | Reject |
| 1 | 0.0081 | 0.0309 | 0.8299 | 2.2622 | FALSE |
| 2 | (0.0034) | 0.0449 | 0.2372 | 2.2622 | FALSE |
| 4 | 0.0013 | 0.0397 | 0.1025 | 2.2622 | FALSE |
| 5 | (0.0147) | 0.0462 | 1.0019 | 2.2622 | FALSE |
| 6 | 0.0040 | 0.0447 | 0.2803 | 2.2622 | FALSE |
| 7 | (0.0080) | 0.0394 | 0.6442 | 2.2622 | FALSE |
| 9 | (0.0001) | 0.0282 | 0.0111 | 2.2622 | FALSE |
| 11 | (0.0020) | 0.0344 | 0.1821 | 2.2622 | FALSE |
| 14 | (0.0146) | 0.0369 | 1.2480 | 2.2622 | FALSE |
| 18 | (0.0027) | 0.0437 | 0.1934 | 2.2622 | FALSE |
| 22 | (0.0267) | 0.0315 | 2.6860 | 2.2622 | TRUE |
| 28 | (0.0190) | 0.0387 | 1.5536 | 2.2622 | FALSE |
| 34 | (0.0091) | 0.0450 | 0.6408 | 2.2622 | FALSE |
| 43 | 0.0010 | 0.0294 | 0.1065 | 2.2622 | FALSE |
| 54 | (0.0059) | 0.0423 | 0.4436 | 2.2622 | FALSE |
| 69 | 0.0128 | 0.0451 | 0.8952 | 2.2622 | FALSE |
| 86 | 0.0426 | 0.0366 | 3.6830 | 2.2622 | TRUE |
| 108 | 0.0340 | 0.0511 | 2.1010 | 2.2622 | FALSE |
| 136 | 0.0291 | 0.0272 | 3.3797 | 2.2622 | TRUE |
| 171 | 0.0406 | 0.0321 | 4.0042 | 2.2622 | TRUE |
| 215 | 0.0107 | 0.0460 | 0.7356 | 2.2622 | FALSE |
| 271 | 0.0103 | 0.0330 | 0.9874 | 2.2622 | FALSE |
| 341 | (0.0319) | 0.0441 | 2.2842 | 2.2622 | TRUE |

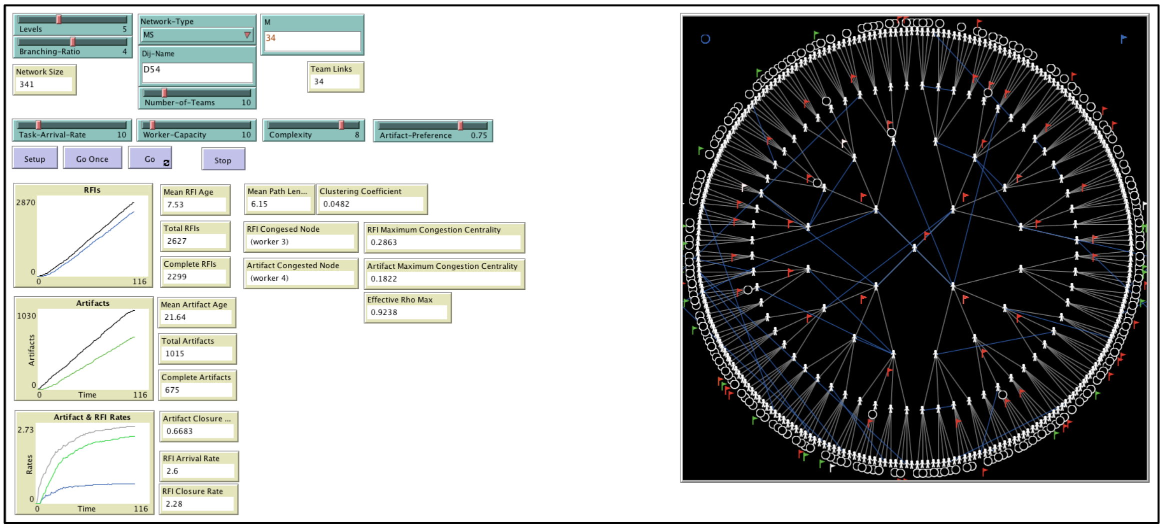
| Multiscale-Matrix Comparison Hypothesis Test | | | | | |
|---|---|---|---|---|---|
| m | $\bar{d}$ | $S_D$ | $\lvert t_0 \rvert$ | t | Reject |
| 1 | 0.0045 | 0.0447 | 0.3153 | 2.2622 | FALSE |
| 2 | 0.0101 | 0.0443 | 0.7217 | 2.2622 | FALSE |
| 4 | 0.0262 | 0.0266 | 3.1176 | 2.2622 | TRUE |
| 5 | 0.0258 | 0.0290 | 2.8146 | 2.2622 | TRUE |
| 6 | 0.0528 | 0.0378 | 4.4162 | 2.2622 | TRUE |
| 7 | 0.0337 | 0.0411 | 2.5873 | 2.2622 | TRUE |
| 9 | 0.0622 | 0.0247 | 7.9645 | 2.2622 | TRUE |
| 11 | 0.0797 | 0.0305 | 8.2724 | 2.2622 | TRUE |
| 14 | 0.0740 | 0.0225 | 10.3900 | 2.2622 | TRUE |
| 18 | 0.0865 | 0.0286 | 9.5727 | 2.2622 | TRUE |
| 22 | 0.0868 | 0.0248 | 11.0784 | 2.2622 | TRUE |
| 28 | 0.1162 | 0.0363 | 10.1400 | 2.2622 | TRUE |
| 34 | 0.1198 | 0.0371 | 10.2037 | 2.2622 | TRUE |
| 43 | 0.1473 | 0.0206 | 22.6695 | 2.2622 | TRUE |
| 54 | 0.1777 | 0.0369 | 15.2357 | 2.2622 | TRUE |
| 69 | 0.1803 | 0.0349 | 16.3407 | 2.2622 | TRUE |
| 86 | 0.2345 | 0.0332 | 22.3493 | 2.2622 | TRUE |
| 108 | 0.2364 | 0.0233 | 32.0975 | 2.2622 | TRUE |
| 136 | 0.2527 | 0.0402 | 19.8729 | 2.2622 | TRUE |
| 171 | 0.2793 | 0.0249 | 35.4572 | 2.2622 | TRUE |
| 215 | 0.2655 | 0.0498 | 16.8506 | 2.2622 | TRUE |
| 271 | 0.2620 | 0.0274 | 30.2186 | 2.2622 | TRUE |
| 341 | 0.2199 | 0.0335 | 20.7602 | 2.2622 | TRUE |

| Random-Military Comparison Hypothesis Test | | | | | |
|---|---|---|---|---|---|
| m | $\bar{d}$ | $S_D$ | $\lvert t_0 \rvert$ | t | Reject |
| 1 | (0.0244) | 0.0379 | 2.0334 | 2.2622 | FALSE |
| 2 | (0.0046) | 0.0304 | 0.4738 | 2.2622 | FALSE |
| 4 | (0.0260) | 0.0313 | 2.6271 | 2.2622 | TRUE |
| 5 | (0.0244) | 0.0320 | 2.4076 | 2.2622 | TRUE |
| 6 | (0.0330) | 0.0449 | 2.3217 | 2.2622 | TRUE |
| 7 | (0.0403) | 0.0269 | 4.7419 | 2.2622 | TRUE |
| 9 | (0.0486) | 0.0315 | 4.8849 | 2.2622 | TRUE |
| 11 | (0.0518) | 0.0333 | 4.9156 | 2.2622 | TRUE |
| 14 | (0.0461) | 0.0389 | 3.7467 | 2.2622 | TRUE |
| 18 | (0.0554) | 0.0398 | 4.4034 | 2.2622 | TRUE |
| 22 | (0.0574) | 0.0338 | 5.3662 | 2.2622 | TRUE |
| 28 | (0.0752) | 0.0374 | 6.3545 | 2.2622 | TRUE |
| 34 | (0.0685) | 0.0306 | 7.0834 | 2.2622 | TRUE |
| 43 | (0.0724) | 0.0491 | 4.6633 | 2.2622 | TRUE |
| 54 | (0.0864) | 0.0430 | 6.3595 | 2.2622 | TRUE |
| 69 | (0.0809) | 0.0307 | 8.3317 | 2.2622 | TRUE |
| 86 | (0.0697) | 0.0426 | 5.1687 | 2.2622 | TRUE |
| 108 | (0.0688) | 0.0417 | 5.2127 | 2.2622 | TRUE |
| 136 | (0.0763) | 0.0463 | 5.2146 | 2.2622 | TRUE |
| 171 | (0.0504) | 0.0486 | 3.2794 | 2.2622 | TRUE |
| 215 | (0.0656) | 0.0188 | 11.0328 | 2.2622 | TRUE |
| 271 | (0.0399) | 0.0290 | 4.3580 | 2.2622 | TRUE |
| 341 | (0.0700) | 0.0391 | 5.6674 | 2.2622 | TRUE |

| Random-Matrix Comparison Hypothesis Test | | | | | |
|---|---|---|---|---|---|
| m | $\bar{d}$ | $S_D$ | $\|t_0\|$ | t | Reject |
| 1 | (0.0280) | 0.0317 | 2.7942 | 2.2622 | TRUE |
| 2 | 0.0089 | 0.0290 | 0.9706 | 2.2622 | FALSE |
| 4 | (0.0011) | 0.0147 | 0.2351 | 2.2622 | FALSE |
| 5 | 0.0161 | 0.0234 | 2.1851 | 2.2622 | FALSE |
| 6 | 0.0158 | 0.0282 | 1.7758 | 2.2622 | FALSE |
| 7 | 0.0014 | 0.0439 | 0.0998 | 2.2622 | FALSE |
| 9 | 0.0137 | 0.0414 | 1.0443 | 2.2622 | FALSE |
| 11 | 0.0299 | 0.0289 | 3.2676 | 2.2622 | TRUE |
| 14 | 0.0424 | 0.0347 | 3.8592 | 2.2622 | TRUE |
| 18 | 0.0338 | 0.0218 | 4.8876 | 2.2622 | TRUE |
| 22 | 0.0561 | 0.0308 | 5.7669 | 2.2622 | TRUE |
| 28 | 0.0600 | 0.0356 | 5.3327 | 2.2622 | TRUE |
| 34 | 0.0604 | 0.0265 | 7.2000 | 2.2622 | TRUE |
| 43 | 0.0740 | 0.0351 | 6.6695 | 2.2622 | TRUE |
| 54 | 0.0972 | 0.0250 | 12.2947 | 2.2622 | TRUE |
| 69 | 0.0866 | 0.0391 | 7.0130 | 2.2622 | TRUE |
| 86 | 0.1222 | 0.0348 | 11.1104 | 2.2622 | TRUE |
| 108 | 0.1337 | 0.0292 | 14.4661 | 2.2622 | TRUE |
| 136 | 0.1472 | 0.0431 | 10.8011 | 2.2622 | TRUE |
| 171 | 0.1883 | 0.0403 | 14.7864 | 2.2622 | TRUE |
| 215 | 0.1892 | 0.0364 | 16.4277 | 2.2622 | TRUE |
| 271 | 0.2118 | 0.0234 | 28.6322 | 2.2622 | TRUE |
| 341 | 0.1818 | 0.0404 | 14.2299 | 2.2622 | TRUE |

<u>NetLogo Code and Screenshots</u>

Artifacts Model, Screen Shot



## Artifacts Model Code

```
extensions [nw cf csv matrix]
globals [
  Dij-File           ;;file name for file containing the Dij matrix
  Age                ;;variable used to limit search for new links at
given Threshold value
  Threshold          ;;threshold for adding team links, links added
when random-float Threshold < P
  Lambda             ;;network parameter for multiscale networks
  Zeta               ;;network parameter for multiscale networks
  Dij                ;;matrix containing depth of common nodes between
nodes i and j
  Nodes              ;;total number of nodes in the network
  RFI-Rho-Max        ;;maximum RFI congestion centrality of network
  RFI-Congested-Node ;;node with maximum RFI congestion centrality
  A-Rho-Max          ;;maximum artifact congestion centrality
  A-Congested-Node   ;;node with maximum artifact congestion centrality
  BC-Max             ;;maximum normalized betweenness centrality of
network
  Central-Node       ;;node with maximum betweenness centrality
  GCC                ;;global clustering coefficient of network
  MPL                ;;mean path length of network
  E-Rho-Max          ;;maximum effective congestion centrality
  A-Counter
  B-Counter
]

breed [workers worker]      ;;Agent breed representing workers in the
organization
breed [artifacts artifact]  ;;Meta-agent representing work products
breed [RFIs RFI]            ;;Meta-agent for information requests

;;Workers represent the individuals in the organizational hierarchy,
i.e., the individual
;;engineers, designers, supervisors, managers, etc who make up the
organization
workers-own [
  Depth           ;;Worker's depth in hierarchy
  My-Department   ;;Worker's department, the main branches of hierarchy
  My-Supervisor   ;;Worker's immediate superior in hierarchy
  My-Team         ;;Worker's team assignment, used for matrix and BCCWG
networks
  Capacity        ;;Worker's capacity to perform work in a single tick
  Artifact-Queue  ;;Worker's artifact in box (list)
  Artifact-Count  ;;Number of artifacts processed by worker
  Hold-Queue      ;;List of artifacts placed on hold pending resolution
of RFI (list)
  RFI-Queue       ;;Worker's RFI in box (list)
  RFI-Count       ;;Number of RFIs processed by worker
]

;;Artifacts represent design products, such as drawings, calculations,
specifications, or
;;other documents created by the design organization
artifacts-own [
```

```
  A-Originator    ;;Worker to whom artifact initially assigned
[Agentset]
  A-Status        ;;Artifact status: Open, Hold or Complete
  A-Age           ;;Age of the artifact, incremented until artifact
complete
]

;;Requests for Information (RFIs) represent questions or requests to
other workers
;;for infomration
RFIs-own [
  R-Artifact      ;;Artifact the RFI relates to [Agentset]
  R-Originator    ;;Worker who originated the RFI [Agentset]
  R-Target        ;;Worker to whom the RFI was sent [Agentset]
  R-Status        ;;Status of RFI: Open, Answered, or Complete
  R-Age           ;;Age of RFI
]

;;organizational links represent the hierarchical backbone structure of
the organization,
;;defined by levels and branching ratio
undirected-link-breed [organizations organization]

;;team links represent the links added to the organization, according
to network construction algorithm
undirected-link-breed [teams team]



to setup
  clear-all
  reset-ticks

  set Nodes (Branching-Ratio ^ Levels - 1) / (Branching-Ratio - 1)
  set A-Counter 0
  set B-Counter 0

  ;;create organizational hierarchy based on number of levels and
branching
  ;;ratio specified by user
  foreach n-values Levels [ [?1] -> ?1 ] [ [?1] ->
    ifelse ?1 = 0
    [
      ;;create worker 0
      create-workers 1 [
        set Depth 0
        set My-Department 0
        set My-Supervisor 0
        set My-Team 0
        set Capacity Worker-Capacity
        set Artifact-Queue [ ]
        set Artifact-Count 0
        set Hold-Queue [ ]
        set RFI-Queue [ ]
        set RFI-COunt 0
      ]
    ]
```

```
  [
    ;;make new level in hierarchy
    make-level ?1
  ]
]
ask workers [
  set shape "person"
  set color white
]
;;arrange workers in radial layout with worker 0 at center
layout-radial workers organizations (worker 0)

;;create the Dij matrix from file
set Dij matrix:make-constant Nodes Nodes 0
ifelse Network-Type = "Matrix" [
  set Dij-File (word Dij-Name "_" Number-of-Teams ".csv")
]
[
  set Dij-File word Dij-Name ".csv"
]
file-open Dij-File
set Dij matrix:from-row-list csv:from-file Dij-File
file-close

;;create team structure based on user inputs
cf:when
cf:case [Network-Type = "Random"] [
  set Threshold 0.9999
  set Lambda 1000000000
  set Zeta 1000000000
  while [count teams < M] [
    let source one-of workers
    let target one-of workers with [who != [who] of source]
    construct-team source target
  ]
]
cf:case [Network-Type = "Matrix"] [
  ;;create matrix-organization teams
  ;;create manager of the project organization
  create-workers 1 [
    set shape "person"
    set color orange
    set Depth 1
    set My-Department who
    set My-Supervisor worker 0
    set Capacity Worker-Capacity
    set Artifact-Queue [ ]
    set Artifact-Count 0
    set Hold-Queue [ ]
    set RFI-Queue [ ]
    set RFI-Count 0
    create-organization-with worker 0
  ]
  ;;create project managers, heads of each of the project teams
  foreach range Number-of-Teams [ [?1] ->
    create-workers 1 [
      set Depth 2
```

```
        set My-Department Branching-Ratio + 1
        set My-Supervisor Nodes
        set My-Team (?1 + 1)
        set Capacity Worker-Capacity
        set Artifact-Queue []
        set Artifact-Count 0
        set Hold-Queue []
        set RFI-Queue []
        set RFI-Count 0
        set shape "person"
        set color red
        create-organization-with worker Nodes
      ]
    ]
    ;;create links to team members
    while [count teams < M] [
      let source one-of workers with [color = red]
      let target one-of workers with [Depth = Levels - 1]
      ask target [
        set My-Team [My-Team] of source
        create-team-with source [
          set color blue
        ]
      ]
    ]
    layout-radial workers organizations (worker 0)
  ]
  cf:case [Network-Type = "BCCWG"] [
    ;;create team around boards, centers, cells and working groups
model
    ;;first, identify team leads
    foreach range Number-of-Teams [ [?1] ->
      ask one-of workers with [My-Team = 0 and Depth = 1 or Depth = 2]
[
        set My-Team (?1 + 1)
        set color red
      ]
    ]
    ;;add team members
    while [count teams < M] [
      let source one-of workers  with [color = red]
      let target one-of workers with [color != red and Depth > 2]
      ask target [
        set My-Team [My-Team] of source
        create-team-with source [
          set color blue
        ]
      ]
    ]
  ]
  cf:else [
    ;;create a multiscale network
    set Threshold 0.3768
    set Lambda 0.5
    set Zeta 0.5
    while [count teams < M ] [
      let source one-of workers
```

```
      let target one-of workers with [who != [who] of source]
      construct-team source target
      ;; if age counter reaches Nodes^2, reduce threshold to lower
value
      set Age Age + 1
      if Age > Nodes ^ 2 [
        set Threshold Threshold ^ 2
        show Threshold
        set Age 0
      ]
    ]
  ]

  ;;calculate network parameters
  set GCC global-clustering-coefficient
  ;;show GCC
  set MPL nw:mean-path-length
  ;;show MPL
  ;;set BC-Max 2 * max [nw:betweenness-centrality] of workers / ((Nodes
- 1) * (Nodes - 2))
  ;;set Central-Node max-one-of workers [nw:betweenness-centrality]


end


to go

  ask workers [
    ;;perform work at each tick up to worker capacity
    repeat Capacity [
      cf:when
      cf:case [empty? RFI-Queue and not empty? Artifact-Queue] [
        ;;process artifact
        process-Artifact worker who
        set A-Counter A-Counter + 1
      ]
      cf:case [not empty? RFI-Queue and empty? Artifact-Queue] [
        ;;process RFI
        process-RFI worker who
        set B-Counter B-Counter + 1
      ]
      cf:case [not empty? RFI-Queue and not empty? Artifact-Queue] [
        ;;process an artifact or RFI according to coin toss
        ifelse random-float 1 < Artifact-Preference [
          ;;artifact selected
          process-Artifact worker who
          set A-Counter A-Counter + 1
        ]
        [
          ;;RFI selected
          process-RFI worker who
          set B-Counter B-Counter + 1
        ]
      ]
      cf:else [
        ;;queues empty, take no action
```

```
      ]
    ]
  ]

  ;;create new Artifacts
  make-new-Artifacts

  ;;increment counters
  increment

  tick
  if ticks > 100 [
    ;;calculate congestion probability
    set RFI-Rho-max maximum-RFI-congestion-centrality
    set RFI-Congested-Node max-one-of workers [RFI-Count]
    set A-Rho-Max maximum-A-congestion-centrality
    set A-Congested-Node max-one-of workers [Artifact-Count]
    ;;show rho-max
    ;;show Congested-Node
    set E-Rho-Max effective-rho-max

    ;;stop execution
    stop
  ]
end

;;procedure to process RFIs
to process-RFI [wrkr]
        let r first RFI-Queue
        let source [R-Originator] of RFI r
        let target [R-Target] of RFI r
        let here worker who
        let next []
        ifelse here = source and [R-Status] of RFI r = "Answered"
        [
          ;;I am the originator of an answered RFI

          ;;remove related Artifact from hold queue and return to top
of
          ;;artifact queue and set artifact status to Open
          let a [R-Artifact] of RFI r
          set Hold-Queue remove a Hold-Queue
          ifelse empty? Artifact-Queue
          [ set Artifact-Queue (list a) ]
          [ set Artifact-Queue fput a Artifact-Queue]
          ask Artifact a [
            set A-Status "Open"
          ]
          ;;close RFI
          ask RFI r [
            set R-Status "Complete"
            set xcor 23
            set ycor 23
            set color blue
          ]
          ;;and remove RFI from my RFI queue
          set RFI-Queue remove r RFI-Queue
```

```
        ]
        [
          ;;I am NOT the originator of an answered RFI...
          ;;increment RFI counter
          set RFI-Count RFI-Count + 1

          ifelse here = target [
            ;;I am the target of the RFI
            ;;set RFI status to Answered
            ask RFI r [
              set R-Status "Answered"
            ]
            ;;return RFI to originator and put on top of RFI queue
            ask source [
              ifelse empty? RFI-Queue
              [ set RFI-Queue (list r) ]
              [ set RFI-Queue fput r RFI-Queue ]
            ]
            ask RFI r [
              set xcor [xcor] of source
              set ycor [ycor] of source
              face worker 0
              fd -2
              set color green
            ]
            ;;remove RFI from my RFI Queue
            set RFI-Queue remove r RFI-Queue
          ]
          [
            ;;I am neither the source of an answered RFI
            ;;nor the target of the RFI, so
            ;;pass RFI to next worker on route to target
            set next one-of organization-neighbors with [Depth <
[Depth] of here]
            let t [who] of target
            let da-kt matrix:get Dij who t
            foreach sort link-neighbors [ [?the-neighbor] ->
              if Network-Type = "Matrix" or Network-Type = "BCCWG" [
                ask ?the-neighbor [
                  if team-neighbor? target [
                    set next ?the-neighbor
                    set da-kt [Depth] of ?the-neighbor
                  ]
                ]
              ]
              let k [who] of ?the-neighbor
              let dd matrix:get Dij k t
              if dd > da-kt [
                set next ?the-neighbor
                set da-kt dd
              ]
            ]
            ;;add RFI to bottom of next worker's queue
            ask next [
              ifelse empty? RFI-Queue
              [ set RFI-Queue (list r) ]
              [ set RFI-Queue lput r RFI-Queue ]
```

```
            ]
            ask RFI r [
              set xcor [xcor] of next
              set ycor [ycor] of next
              face worker 0
              fd -2
            ]
            ;;remove RFI from my RFI Queue
            set RFI-Queue remove r RFI-Queue
          ]
        ]
end

;;procedure to process Artifacts
to process-Artifact [wrkr]
          let n first Artifact-Queue
          let here worker who
          let d Depth
          let originator [A-Originator] of artifact n

          ifelse d = 1 [
            ;;I am the approver, i.e., worker (Manager) at depth 1
            ;;then approve artifact and close
            ask Artifact n [
              Set A-Status "Complete"
              set xcor -23
              set ycor 23
              set color blue
            ]
            ;;remove Artifact from my Artifact Queue
            set Artifact-Queue remove n Artifact-Queue
            ;;increment Artifact Count
            set Artifact-Count Artifact-Count + 1
          ]
          [
            ;;I am not the approver
            ifelse here = originator [
              ;;I am the originator of Artifact
              ;;Evaluate complexity
              ;;Only Artifact originator's evaluate complexity
              ifelse random 10 < Complexity
              [
                ;;Situation complex, initiate RFI
                hatch-RFIs 1 [
                  set R-Artifact n
                  set R-Originator here
                  ;;Select RFI based on complexity.  At low complexity,
tasks are decomposable
                  ;;and are passed in same department.  At high
complexity, tasks are not
                  ;;decomposable and are assigned to another worker at
random
                  let t []
                  ifelse random 10 < Complexity
                  [
                    set t one-of workers with [who != [who] of here and
color = white and Depth = Levels - 1]
```

```
          set shape "flag"
          set color red
        ]
        [
          set t one-of workers with [who != [who] of here and
My-Department = [My-Department] of here and Depth = Levels - 1]
          set shape "flag"
          set color white
        ]
        set R-Target t
        set R-Status "Open"
        ;;set shape "flag"
        ;;set color white
        set xcor [xcor] of here
        set ycor [ycor] of here
        face worker 0
        fd -2
        let r who
        ;;put RFI at bottom of my RFI Queue
        ask here [
          ifelse empty? RFI-Queue
          [ set RFI-Queue (list r) ]
          [ set RFI-Queue lput r RFI-Queue ]
        ]
      ]
    ;;move artifact to Hold Queue
    ask here [
      ifelse empty? Hold-Queue
      [ set Hold-Queue (list n) ]
      [ set Hold-Queue lput n Hold-Queue ]
      set Artifact-Queue remove n Artifact-Queue
    ]
    ]
    [
      ;;Situation NOT complex, process Artifact
      ;;Identify next worker up chain of command
      let next one-of organization-neighbors with [Depth < d]
      ;;and move Artifact to bottom of next worker's Artifact
Queue
      ask next [
        ifelse empty? Artifact-Queue
        [ set Artifact-Queue (list n) ]
        [ set Artifact-Queue lput n Artifact-Queue ]
      ]
      ask Artifact n [
        set xcor [xcor] of next
        set ycor [ycor] of next
        face worker 0
        fd -1
      ]
      ;;remove Artifact from my Artifact Queue
      set Artifact-Queue remove n Artifact-Queue
      ;;increment Artifact Count
      set Artifact-Count Artifact-Count + 1
    ]
  ]
  [
```

```
              ;;I Am neither the originator nor approver of Artifact
              ;;so pass Artfiact to next worker in the organizational
              ;;chain of command

              ;;Identify next worker up chain of command
              let next one-of organization-neighbors with [Depth < d]
              ;;move artifact to bottom of next worker's Artifact Queue
              ask next [
                ifelse empty? Artifact-Queue
                [ set Artifact-Queue (list n) ]
                [ set Artifact-Queue lput n Artifact-Queue ]
              ]
              ask Artifact n [
                set xcor [xcor] of next
                set ycor [ycor] of next
                face worker 0
                fd -1
              ]
              ;;remove Artifact from my Artifact Queue
              set Artifact-Queue remove n Artifact-Queue
              ;;increment Artifact Count
              set Artifact-Count Artifact-Count + 1
            ]
          ]
end

;;procedure to add levels (i.e., rows) to hierarchical organizational
structure
to make-level [row]
  let b Branching-Ratio                ;;let b equal the branching
ratio
  let W b ^ (row - 1)                  ;;let W equal the number of
workers in pervious row
  let N (b ^ (row - 1) - 1) / (b - 1)  ;;let N equal number of workers
in all previous rows of hierarchy
  ;;for each of the workers in the previous row
  foreach n-values W [ [?1] -> ?1 ] [ [?1] ->
    ;;show ? + N
    ;;create b new workers and link to worker ?+N in previous row
    create-workers b [
      set Depth row
      ifelse row = 1 [
        set My-Department who
      ]
      [
        set My-Department [My-Department] of worker (?1 + N)
      ]
      set My-Supervisor worker (?1 + N)
      set My-Team 0
      set Capacity Worker-Capacity
      set Artifact-Queue [ ]
      set Artifact-Count 0
      set Hold-Queue [ ]
      set RFI-Queue [ ]
      set RFI-Count 0
      create-organization-with worker (?1 + n)
    ]
```

```
  ]

end

;;procedure to create team links based on Watts' stochastic rule
to construct-team [source target]
  let i [who] of source
  let d-source [Depth] of source
  let j [who] of target
  let d-target [Depth] of target
  let da matrix:get Dij i j
  let d1 abs (d-source - da)
  let d2 abs (d-target - da)
  ;;if d1+d2 <2, no team link created
  if (d1 + d2) >= 2 [
    let x12 (d1 ^ 2 + d2 ^ 2 - 2) ^ (1 / 2)
    let P exp ((- da) / Lambda) * exp ((- x12) / Zeta)
    if random-float Threshold < P [
      ask source [
        if not team-neighbor? target [
          create-team-with target [
            set color blue
          ]
          set Age 0
        ]
      ]
    ]
  ]

end

;;procedure to make new Artifacts
;;New Artifacts arrive according to a random Poisson process with mean
equal to Task-Arrival-Rate
;;Each new Artifact is randomly assigned to a worker at the lowest
level
;;of the organizational hierarchy
to make-new-Artifacts
  create-artifacts random-poisson Task-Arrival-Rate [
    set A-Status "Open"
    let n who
    let w one-of workers with [Depth = Levels - 1]
    set A-Originator w
    ;;Put artifact at bottom of originator's Artifact Queue
    ask w [
      ifelse empty? Artifact-Queue
      [ set Artifact-Queue (list n) ]
      [ set Artifact-Queue lput n Artifact-Queue ]
    ]
    ;;place Artifact near originator
    set xcor [xcor] of w
    set ycor [ycor] of w
    face worker 0
    fd -1
    set color white
    set shape "circle 2"
  ]
```

```
end

;;procedure to increment counters
to increment

  ;;Increment age of open RFIs
  ask RFIs with [R-Status != "Complete"] [
    set R-Age R-Age + 1
  ]

  ;;Increment age of open Artifacts
  ask Artifacts with [A-Status != "Complete"] [
    Set A-Age A-Age + 1
  ]
end



;;procedure to calculate and report global clustering coefficient,
using method in nw extensions documentation
to-report global-clustering-coefficient
  let closed-triplets sum [ nw:clustering-coefficient * count my-links
* (count my-links - 1 ) ] of workers
  let triplets sum [ count my-links * (count my-links - 1) ] of workers
  report closed-triplets / triplets
end

;;procedures to calculate and report rho-max, the maximum congestion
centrality, for RFIs and Artifacts
to-report maximum-RFI-congestion-centrality
  let r-max max [RFI-Count] of workers
  report r-max / count RFIs
end

to-report maximum-A-congestion-centrality
  let A-max max [Artifact-Count] of workers
  report A-max /  (Task-Arrival-Rate * ticks)
end

;;procedure to calculate and report the maximum work rate of workers
to-report effective-rho-max
  let er-max max [RFI-Count + Artifact-Count] of workers
  ifelse ticks > 0 [
    report er-max / (ticks * Task-Arrival-Rate)
  ]
  [
    report 0
  ]
end

;;procedure to calculate Artifact Rate
to-report Artifact-Rate
  ifelse ticks < 1 [
    report 0
  ]
  [
```

```
    report count Artifacts with [A-Status = "Complete"] / (ticks *
Task-Arrival-Rate)
  ]
end

;;procedure to calculate RFI arrival rate
to-report RFI-Arrival-Rate
  ifelse ticks < 1 [
    report 0
  ]
  [
    report count RFIs / (ticks * Task-Arrival-Rate)
  ]
end

;;procedure to calculate RFI closure rate
to-report RFI-Closure-Rate
  ifelse ticks < 1 [
    report 0
  ]
  [
    report count RFIs with [R-Status = "Complete"] / (ticks * Task-
Arrival-Rate)
  ]
end
```

## MATLAB Code for Creating $D_{ij}$ Matrices

```
DoddsWattsSabel_Dij2
% Calculate and return the Dodd-Watts-Sabel Dij matrix for a matrix
team
% arrangement where project managers are added to hierarchy
% Source: Dodds, Watts, Sabel, "Information exchange and the robustness
of organizational networks", PNAS 100 (21): 12516-12521
% INPUTs: number of levels in hierarchy (L), branching ratio (b), and
% number of teams (teams)
% OUTPUTs: Dij matrix containing depth of highest common node between
nodes
% i and j
% Other routines used: edgeL2adj.m, canonical_nets.m, dijkstra.m

function Dij_Matrix = DoddsWattsSabel_Dij2(L,b,teams)

% calculate number of nodes, before adding PMO staff
N = (b^L - 1)/(b - 1);

% construct a tree with N nodes and branch factor b
adj0=edgeL2adj( canonical_nets(N,'tree',b) ); % backbone adjacency
matrix

% add the PMO manager
adj0(N+1,1) = 1; adj0(1,N+1) = 1;
% add project managers
for k=1:teams
    % add project managers, subordinate to PMO Manager
    adj0(N+1+k,N+1) = 1; adj0(N+1,N+k+1) = 1;
end
adj=adj0;
edges=0;

for i=1:N+1+teams
    for j=i:N+1+teams
        % find di, dj and Dij
        [d1,path1]=dijkstra(adj0,i,1); % adjacency, source, target
        [d2,path2]=dijkstra(adj0,j,1);

        for p=1:length(path1)
            p1=path1(p);
            p2=find(path2==p1);
            if length(p2)>0 % if p1 in path2
                % di+dj is the distance from i to j
                % Dij is level of highest common node on path
                di=p-1; dj=p2-1;
                Dij=length(path1(p:length(path1)))-1;
                break
            end
        end
        Dij_Matrix(i,j)=int16(Dij);
        Dij_Matrix(j,i)=int16(Dij);
    end
end
```

DoddsWattsSabel_Dij3

```
% Calculate and return the Dodd-Watts-Sabel Dij matrix from adjacency
% matrix entered by user
% Source: Dodds, Watts, Sabel, "Information exchange and the robustness
of organizational networks", PNAS 100 (21): 12516-12521
% INPUTs: adjacency matrix (adj0)
% OUTPUTs: Dij matrix containing depth of highest common node between
nodes
% i and j
% Other routines used: edgeL2adj.m, canonical_nets.m, dijkstra.m

function Dij_Matrix = DoddsWattsSabel_Dij3(adj0)

% calculate number of nodes, before adding PMO staff
N = length(adj0);

% construct a tree with N nodes and branch factor b
% adj0=edgeL2adj( canonical_nets(N,'tree',b) ); % backbone adjacency
matrix


for i=1:N
    for j=i:N
        % find di, dj and Dij
        [d1,path1]=dijkstra(adj0,i,1); % adjacency, source, target
        [d2,path2]=dijkstra(adj0,j,1);

        for p=1:length(path1);
            p1=path1(p);
            p2=find(path2==p1);
            if length(p2)>0; % if p1 in path2
                % di+dj is the distance from i to j
                % Dij is level of highest common node on path
                di=p-1; dj=p2-1;
                Dij=length(path1(p:length(path1)))-1;
                break
            end
        end
        Dij_Matrix(i,j)=int16(Dij);
        Dij_Matrix(j,i)=int16(Dij);
    end
end
```

# APPENDIX 3: Smart Teams Model

Elements of the *Smart Team* Model

| Agents |
| --- |
| Workers, representing the individuals within the hierarchy.<br>Depth (level);<br>Department (major division of the hierarchy);<br>Supervisor (immediate superior in the hierarchy);<br>Team (team assignment, for matrix and military staff organizational networks);<br>Capacity (the amount of work the worker can perform in one time step);<br>Artifact Queue (list of artifacts to be processed);<br>Artifact Count (number of artifacts processed by worker);<br>Hold Queue (list of artifacts placed on hold while awaiting RFI response);<br>RFI queue (list of RFIs to be processed); and<br>RFI count (number of RFIs processed by worker).<br><br>Artifacts representing work products.<br>Originator (worker who originated the artifact)'<br>Status (status of the artifact: open, hold or complete); and<br>Age (age of artifact).<br><br>Requests for Information (RFIs), representing messages passed between workers.<br>Artifact (the artifact to which the RFI is related);<br>Originator (worker who originated the RFI);<br>Target (worker to whom the RFI was sent);<br>Status (status of the RFI: open, answered, or complete); and<br>Age (age of RFI). |
| Links |
| Organization Links, representing the hierarchical structure.<br>Team Links, representing the cross-functional team links added to the hierarchy. |
| Environment |
| The environment is defined by the backbone hierarchical network, the team links added to the hierarchical backbone, and the task environment. For matrix and military staff organizational networks, the Smart Team model extends the Artifacts model to account for teams of different size. The model uses a stochastic rule to determine if a new link will be added to a given team, or whether a new link will be created between existing team members. The rule:<br><br>$P_{new} = e^{-S/\chi}$<br><br>where S is the size of the team and $\chi$ is a scaling factor. As the size of a team increases relative to the scaling factor, it is more likely intra-team links will be added.<br><br>The Smart Team model describes the task environment in terms of the rate and distribution of artifacts to be processed and messages that must be exchanged to |

| |
|---|
| accomplish cross-functional collaboration.<br>The artifact rate, $\mu_A$, is the average number of artifacts originated by each node at each time stem, and $\mu_A N$ is the total number of artifacts originated across the network at each time step.<br>Artifact routing follows the functional hierarchy. Workers at the lowest level of the hierarchy originate artifacts and then pass them up the functional chain of command to a manager near the top of the hierarchy for approval. The Smart Team model extends the Artifact model to account for decentralized approvals. When the decentralized approvals option is selected, artifacts can be approved by a supervisor, one level below the manager. The model includes an input called decentralized preference which controls the probability a supervisor will approve an artifact. When the preference is higher, it is more likely a supervisor will approve an artifact.<br>For simple tasks, the originating worker likely has sufficient information to complete the artifact without the need for cross-functional collaboration. For complex tasks, however, the worker likely lacks sufficient information and requires additional information from other workers. In this case, the originating worker places the artifact on hold and originates a request for information (RFI) to acquire the additional information required to complete the artifact.<br>RFIs pass from source to target through a chain of intermediate nodes as with messages in the information exchange model. Upon receipt, the RFI target provides the information requested and returns the RFI directly to the originator. When the originator receives an answered RFI, he completes the associated artifact and routes if for approval.<br>Complexity affects the rate and distribution of RFIs. At low complexity, few RFIs are created, and because tasks are decomposable, RFIs are routed to other workers in the same functional organization. At high complexity, many RFIs are created. Since tasks are not decomposable, RFIs are routed to other workers across the organization. The *Artifact* model uses the same qualitative complexity scale used in the information exchange models implement in phase one. |
| Time Behavior |
| At each time step, workers process artifacts and information requests up to their capacity.<br>If a given node has only RFIs or artifacts available, it processes them, but if both are available, it decides which to process by comparing a random number to an artifact preference rating, in the range [0,1].<br>When the artifact preference rating is higher, it is more likely the node will select an artifact than an RFI. An artifact rating of 0.5 represents a "coin flip," with the node choosing RFIs half the time, and artifacts the other half. |
| Inputs |
| Network parameters: Levels, branching ratio;<br>Network type: random, multiscale, matrix or military staff (BCCWG);<br>Number of Teams, for matrix and military staff organizational networks;<br>$D_{ij}$ Name, the file containing a matrix of the depths of lowest common ancestors;<br>Team links added, *m*;<br>Task Arrival Rate;<br>Worker Capacity; |

| |
|---|
| Complexity; and<br>Artifact Preference |
| Outputs |
| The principal output is artifact completion rate, defined as the number of artifacts completed divided by the total number of artifacts. If the organizational network is able to keep pace with the artifact and information processing work load, the artifact completion rate will tend to unity with a small deviation resulting from the artifacts in process at any given time step. Additional outputs include:<br>RFI arrival and completion rates;<br>Mean age of RFIs and artifacts;<br>RFI, artifact and effective congestion centralities and congested nodes;<br>Difference between total and completed artifacts and delta-slope; and<br>Network parameters: mean path length and global clustering coefficient. |

<u>Verification and Validation of the *Smart Team* Model</u>

| Verification of the *Smart Team* Model | | | |
|---|---|---|---|
| Documentation | | | |
| Model documentation consistent with conceptual model<br>Code comments adequately identify implementation of conceptual model | | | |
| Comments | | | |
| Documentation and code comments sufficient. | | | |
| Programmatic Testing | | | |
| Item Tested | Method | Result | Comments |
| Proper application of the stochastic rule for adding team links either as new members or intra-team links | Observation of team links during network creation. | Satisfactory | |
| Proper creation and assignment of artifacts | Observe artifact arrival in viewer. | Satisfactory | |
| Proper application of artifact preference. | Monitor number of RFIs and artifacts processed at each time step. | Satisfactory | |
| Proper placing on hold of artifacts when complexity high | Watch artifacts and worker hold queues. | Satisfactory | |
| Proper creation of RFIs and selection of target nodes. | Watch artifacts and RFIs. | Satisfactory | |

<table>
<tr><td>Proper routing of artifacts along functional hierarchy</td><td>Observe artifact routing in viewer.</td><td>Satisfactory</td><td></td></tr>
<tr><td>Supervisor approval of artifacts when decentralized approval enabled.</td><td>Observe (watch) artifacts and their approval in viewer.</td><td>Satisfactory</td><td></td></tr>
<tr><td>Proper routing of RFIs</td><td>Watch artifacts in viewer.</td><td>Satisfactory</td><td></td></tr>
</table>

<table>
<tr><th colspan="3">Test Cases</th></tr>
<tr><th>Method</th><th>Description</th><th>Result</th></tr>
<tr><td>Edge cases: high complexity</td><td>Calculate artifact completion rates at high complexity.</td><td>Networks demonstrate performance comparable to artifacts model.</td></tr>
<tr><th colspan="3">Conclusion</th></tr>
<tr><td colspan="3">Model correctly implements stochastic rule for team links in matrix and military staff organizations, the selection of RFIs and artifacts for processing, and decentralized approvals. Elements common to information exchange and artifact models previously verified.</td></tr>
</table>

<table>
<tr><th colspan="2">Validation of the Artifacts Model</th></tr>
<tr><th colspan="2">Face Validation</th></tr>
<tr><th>Micro-Face Validation</th><th>Macro-Face Validation</th></tr>
<tr><td>Principal inputs to model are the organizational networks and the task environment. The organizational networks are based on real-world organizations or ideal classes described in the literature (i.e., random and multiscale). Model implements the creation of artifacts and sharing of information in design organizations.</td><td>The model realistically depicts the flow of artifacts and information in organizational networks, combining both formal passing of information up and down a hierarchy, and informal passing through team relationships.</td></tr>
<tr><th colspan="2">Empirical Validation</th></tr>
<tr><th>Empirical Input Validation</th><th>Empirical Output Validation</th></tr>
<tr><td>The hierarchical backbone is described by number of levels and branching ratio. This is an idealization in that real organizations exhibit irregularities in both level and branching, but the idealization is reasonable.<br><br>The model rates complexity on a scale of</td><td>Empirical validation relies on stylized facts, primarily the expectation that organizational networks will exhibit different performance for different combinations of smart team parameters. A designed experiment confirms Smart Team factors except artifact preference affect artifact completion rates.</td></tr>
</table>

| | |
|---|---|
| 1 to 10. Use of a simple scale is not meant to represent a quantitative comparison of system complexity, but instead differentiates systems of low and high complexity in a numerical fashion that is easy to implement in a model.<br><br>The model uses an artifact preference rating to control worker selection between artifacts and RFIs when both are present. This is a reasonable representation of real-world behavior.<br><br>The model uses a selector to enable decentralized approvals. Organizations often allow supervisors and managers at different levels in the organization to approve work products. | |
| Conclusion | |
| The model is considered valid for the purpose of evaluating ways organizational networks can be modified to improve their performance. . | |

Validation Experiment Results from Minitab

## Power and Sample Size

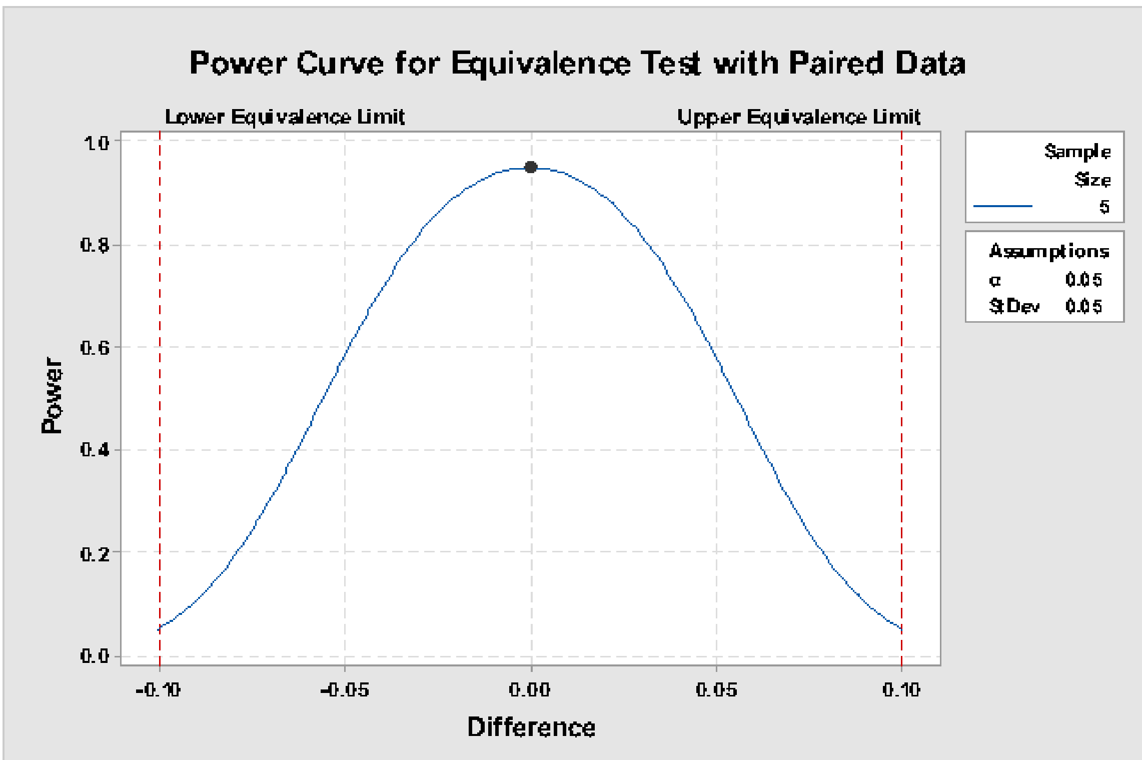
2-Sample t Test
Testing mean 1 = mean 2 (versus ≠)
Calculating power for mean 1 = mean 2 + difference
α = 0.05  Assumed standard deviation = 0.03

### Results

| Difference | Sample Size | Target Power | Actual Power |
|---|---|---|---|
| 0.1 | 4 | 0.9 | 0.972660 |

*The sample size is for each group.*

### Power Curve for 2-Sample t Test




## Factor Information

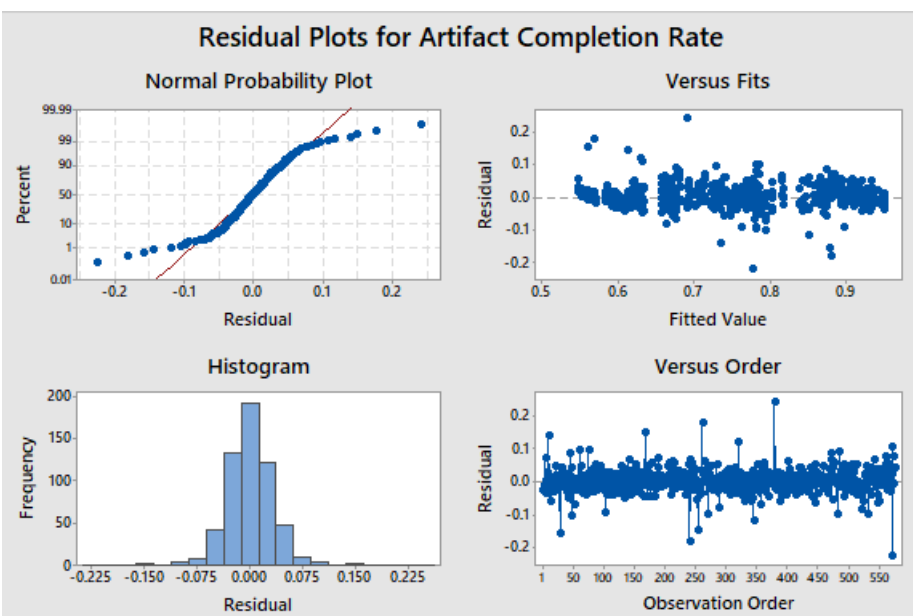
| Factor | Type | Levels | Values |
|---|---|---|---|
| Network Type | Fixed | 4 | Random, Multi-Scale, Matrix, Military Staff |
| Chi | Fixed | 2 | 5, 50 |
| Artifact Preference | Fixed | 3 | RFI, Balanced, Artifact |
| Decentralized Approval | Fixed | 2 | True, False |
| Team Links Added (m) | Fixed | 3 | 34, 108, 341 |

## Analysis of Variance

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Network Type | 3 | 4.26709 | 1.42236 | 925.03 | 0.000 |
| Chi | 1 | 0.02232 | 0.02232 | 14.52 | 0.000 |
| Artifact Preference | 2 | 0.00030 | 0.00015 | 0.10 | 0.908 |
| Decentralized Approval | 1 | 0.06647 | 0.06647 | 43.23 | 0.000 |
| Team Links Added (m) | 2 | 2.65732 | 1.32866 | 864.09 | 0.000 |
| Network Type*Chi | 3 | 0.17953 | 0.05984 | 38.92 | 0.000 |
| Network Type*Artifact Preference | 6 | 0.00423 | 0.00070 | 0.46 | 0.839 |
| Network Type*Decentralized Approval | 3 | 0.02210 | 0.00737 | 4.79 | 0.003 |
| Network Type*Team Links Added (m) | 6 | 0.59060 | 0.09843 | 64.02 | 0.000 |
| Chi*Artifact Preference | 2 | 0.00227 | 0.00114 | 0.74 | 0.478 |
| Chi*Decentralized Approval | 1 | 0.00019 | 0.00019 | 0.12 | 0.729 |
| Chi*Team Links Added (m) | 2 | 0.04035 | 0.02018 | 13.12 | 0.000 |
| Artifact Preference*Decentralized Approval | 2 | 0.00350 | 0.00175 | 1.14 | 0.321 |
| Artifact Preference*Team Links Added (m) | 4 | 0.00643 | 0.00161 | 1.05 | 0.383 |
| Decentralized Approval*Team Links Added (m) | 2 | 0.01339 | 0.00669 | 4.35 | 0.013 |
| Error | 535 | 0.82264 | 0.00154 | | |
| Lack-of-Fit | 103 | 0.31384 | 0.00305 | 2.59 | 0.000 |
| Pure Error | 432 | 0.50880 | 0.00118 | | |
| Total | 575 | 8.69873 | | | |

## Model Summary

| S | R-sq | R-sq(adj) | R-sq(pred) |
|---|---|---|---|
| 0.0392128 | 90.54% | 89.84% | 89.04% |



Using Artifact Completion Rate to Identify Congestion

Cross-over from always congested to sometimes congested:

| Rate | Congested |
|---|---|
| 0.8398 | Yes |
| 0.8414 | Yes |
| 0.8422 | Yes |
| 0.8446 | Yes |
| 0.8458 | Yes |
| 0.849 | No |
| 0.8502 | Yes |
| 0.8518 | Yes |
| 0.8534 | No |
| 0.8562 | Yes |

Cross-Over from sometimes congested to free of congestion:

| Rate | Congested |
|---|---|
| 0.8992 | Yes |
| 0.8996 | Yes |
| 0.9 | No |
| 0.9008 | No |
| 0.9016 | No |
| 0.9016 | Yes |
| 0.9028 | No |
| 0.9032 | No |
| 0.9032 | No |
| 0.9032 | No |

# Breakpoints for Stable Artifact Environments

| Artifact Completion Rates at High Complexity, Centralized Approvals | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Team Links | | RFI Preference | | | | Balanced Preference | | | | Artifact Preference | | | |
| Row Labels | log m/N | BCCWG, χ = 50 | Matrix, χ = 5 | Multi-Scale | Random | BCCWG, χ = 50 | Matrix, χ = 5 | Multi-Scale | Random | BCCWG, χ = 50 | Matrix, χ = 5 | Multi-Scale | Random |
| 3 | -2.06 | 0.58625498 | 0.567290837 | 0.601593625 | 0.575976096 | 0.598087649 | 0.561752988 | 0.588605578 | 0.573466135 | 0.589960159 | 0.572868526 | 0.604621514 | 0.572430279 |
| 7 | -1.69 | 0.620517928 | 0.564462151 | 0.623505976 | 0.576135458 | 0.602988048 | 0.562589641 | 0.622868526 | 0.57310757 | 0.62059761 | 0.568964143 | 0.616812749 | 0.577609562 |
| 11 | -1.49 | 0.640717131 | 0.565697211 | 0.639880478 | 0.603784861 | 0.649163347 | 0.569243028 | 0.642390438 | 0.602749004 | 0.650876494 | 0.560996016 | 0.645099602 | 0.589243028 |
| 20 | -1.23 | 0.67752988 | 0.562390438 | 0.67501992 | 0.605816733 | 0.680796813 | 0.566095618 | 0.672350598 | 0.614262948 | 0.682231076 | 0.576135458 | 0.671394422 | 0.61059761 |
| 34 | -1.00 | 0.726374502 | 0.575059761 | 0.711832669 | 0.628884462 | 0.72625498 | 0.5712749 | 0.705976096 | 0.637450199 | 0.713227092 | 0.575537849 | 0.729601594 | 0.636414343 |
| 61 | -0.75 | 0.77498008 | 0.585776892 | 0.762549801 | 0.678804781 | 0.779641434 | 0.587211155 | 0.767569721 | 0.67812749 | 0.774422311 | 0.578924303 | 0.778446215 | 0.683705179 |
| 108 | -0.50 | 0.835776892 | 0.603984064 | 0.863864542 | 0.727609562 | 0.829721116 | 0.618804781 | 0.877689243 | 0.728844622 | 0.840876494 | 0.615179283 | 0.877450199 | 0.729322709 |
| 192 | -0.25 | 0.885298805 | 0.645537849 | 0.919760956 | 0.81498008 | 0.89059761 | 0.647729084 | 0.931553785 | 0.815936255 | 0.890876494 | 0.648804781 | 0.93063745 | 0.814143426 |
| 341 | 0.00 | 0.921115538 | 0.682629482 | 0.936414343 | 0.89940239 | 0.928486056 | 0.683067729 | 0.936613546 | 0.898326693 | 0.917091633 | 0.68498008 | 0.926334661 | 0.910557769 |
| 430 | 0.10 | 0.928366534 | 0.677729084 | 0.935298805 | 0.92501992 | 0.922589641 | 0.690478088 | 0.944262948 | 0.915059761 | 0.917649402 | 0.668247012 | 0.928844622 | 0.914940239 |
| 541 | 0.20 | 0.923625498 | 0.635936255 | 0.931832669 | 0.925498008 | 0.928247012 | 0.631513944 | 0.929561753 | 0.919960159 | 0.926613546 | 0.634023904 | 0.936095618 | 0.931474104 |
| 681 | 0.30 | 0.897450199 | 0.597888446 | 0.940159363 | 0.919123506 | 0.925936255 | 0.585976096 | 0.939681275 | 0.926733068 | 0.923625498 | 0.617250996 | 0.932749004 | 0.924342629 |

| Artifact Completion Rates at High Complexity, Decentralized Approvals | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Team Links | | RFI Preference | | | | Balanced Preference | | | | Artifact Preference | | | |
| Row Labels | log m/N | BCCWG, χ = 50 | Matrix, χ = 5 | Multi-Scale | Random | BCCWG, χ = 50 | Matrix, χ = 5 | Multi-Scale | Random | BCCWG, χ = 50 | Matrix, χ = 5 | Multi-Scale | Random |
| 3 | -2.06 | 0.594462151 | 0.55749004 | 0.587051793 | 0.567768924 | 0.596613546 | 0.551633466 | 0.600039841 | 0.576972112 | 0.588486056 | 0.560956175 | 0.607250996 | 0.572549801 |
| 7 | -1.69 | 0.641035857 | 0.570159363 | 0.651633466 | 0.581832669 | 0.629601594 | 0.560517928 | 0.630318725 | 0.588804781 | 0.635537849 | 0.56 | 0.633227092 | 0.581115538 |
| 11 | -1.49 | 0.678007968 | 0.568645418 | 0.657729084 | 0.584661355 | 0.664342629 | 0.567450199 | 0.656932271 | 0.592430279 | 0.686294821 | 0.565179283 | 0.649003984 | 0.608047809 |
| 20 | -1.23 | 0.716095618 | 0.572709163 | 0.710239044 | 0.614900398 | 0.708645418 | 0.568366534 | 0.716294821 | 0.611035857 | 0.712988048 | 0.567689243 | 0.68247012 | 0.615936255 |
| 34 | -1.00 | 0.762788845 | 0.57250996 | 0.761992032 | 0.640956175 | 0.758047809 | 0.578486056 | 0.757171315 | 0.662191235 | 0.763784861 | 0.578247012 | 0.757330677 | 0.649840637 |
| 61 | -0.75 | 0.824302789 | 0.596454183 | 0.818207171 | 0.68940239 | 0.832948207 | 0.584780876 | 0.831354582 | 0.70685259 | 0.818406375 | 0.58689243 | 0.827410359 | 0.71 |
| 108 | -0.50 | 0.88498008 | 0.614661355 | 0.917171315 | 0.780876494 | 0.871553785 | 0.608326693 | 0.899601594 | 0.77187251 | 0.87059761 | 0.617968127 | 0.902151394 | 0.775258964 |
| 192 | -0.25 | 0.920039841 | 0.662589641 | 0.932948207 | 0.868366534 | 0.911752988 | 0.661155378 | 0.933665339 | 0.862868526 | 0.896812749 | 0.652430279 | 0.933824701 | 0.865737052 |
| 341 | 0.00 | 0.919083665 | 0.690278884 | 0.938286853 | 0.910756972 | 0.932151394 | 0.699282869 | 0.925896414 | 0.91059761 | 0.918007968 | 0.68812749 | 0.928685259 | 0.924223108 |
| 430 | 0.10 | 0.913944223 | 0.666533865 | 0.938007968 | 0.928007968 | 0.928565737 | 0.684143426 | 0.927410359 | 0.929083665 | 0.932709163 | 0.675258964 | 0.941314741 | 0.927928287 |
| 541 | 0.20 | 0.936095618 | 0.634183267 | 0.936772908 | 0.926454183 | 0.921314741 | 0.644581673 | 0.944940239 | 0.930398406 | 0.92314741 | 0.631832669 | 0.937051793 | 0.922111554 |
| 681 | 0.30 | 0.93059761 | 0.594302789 | 0.937370518 | 0.924143426 | 0.931195219 | 0.587211155 | 0.942111554 | 0.920159363 | 0.902549801 | 0.594063745 | 0.949043825 | 0.9212749 |

Hypothesis Testing-Smart Teams Characterization

Taking as a specific example the case of balanced preference and high complexity, hypothesis testing confirms the value of decentralized artifact approvals for multiscale, military staff, and random networks.

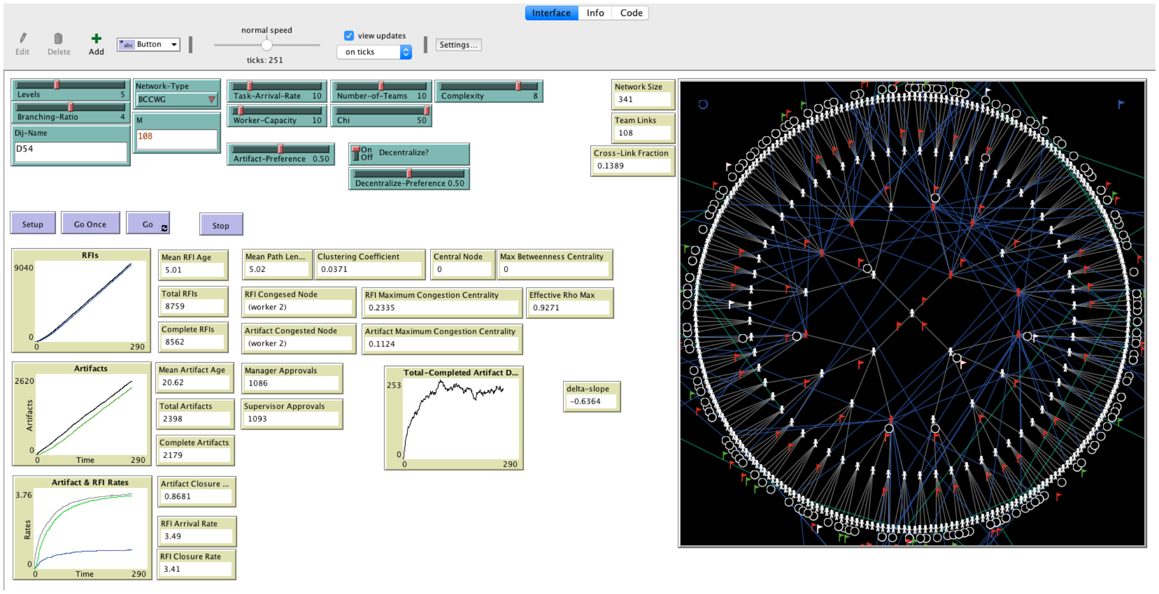
| Multiscale Comparison, Centralized - Decentralized | | | | | | | |
|---|---|---|---|---|---|---|---|
| m | d-bar | ds | t0 | t= | Reject | t< | Reject |
| 3 | -0.0114343 | 0.04221606 | -0.8565061 | 2.26215716 | FALSE | 1.83311293 | FALSE |
| 7 | -0.0074502 | 0.04533673 | -0.5196581 | 2.26215716 | FALSE | 1.83311293 | FALSE |
| 11 | -0.0145418 | 0.04708739 | -0.9765951 | 2.26215716 | FALSE | 1.83311293 | FALSE |
| 20 | -0.0439442 | 0.031715 | -4.381644 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 34 | -0.0511952 | 0.02550855 | -6.3466357 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 61 | -0.0637849 | 0.04916673 | -4.1024783 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 108 | -0.0219124 | 0.02301273 | -3.0110692 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 192 | -0.0021116 | 0.02332411 | -0.2862839 | 2.26215716 | FALSE | 1.83311293 | FALSE |
| 341 | 0.01071713 | 0.01853858 | 1.82810929 | 2.26215716 | FALSE | 1.83311293 | FALSE |

| Military Staff Comparison, Centralized - Decentralized | | | | | | | |
|---|---|---|---|---|---|---|---|
| m | d-bar | ds | t0 | t= | Reject | t< | Reject |
| 3 | 0.0014741 | 0.02963485 | 0.15729877 | 2.26215716 | FALSE | 1.83311293 | FALSE |
| 7 | -0.0266135 | 0.036513 | -2.3049165 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 11 | -0.0151793 | 0.03967093 | -1.2099819 | 2.26215716 | FALSE | 1.83311293 | FALSE |
| 20 | -0.0278486 | 0.033383 | -2.6380197 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 34 | -0.0317928 | 0.02342202 | -4.2924461 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 61 | -0.0533068 | 0.02865746 | -5.8822674 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 108 | -0.0418327 | 0.04039063 | -3.2751786 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 192 | -0.0211554 | 0.03340888 | -2.002437 | 2.26215716 | FALSE | 1.83311293 | TRUE |
| 341 | -0.0036653 | 0.01937255 | -0.5983115 | 2.26215716 | FALSE | 1.83311293 | FALSE |

| Random Comparison, Centralized - Decentralized | | | | | | | |
|---|---|---|---|---|---|---|---|
| m | d-bar | ds | t0 | t= | Reject | t< | Reject |
| 3 | -0.003506 | 0.02331107 | -0.4756054 | 2.26215716 | FALSE | 1.83311293 | FALSE |
| 7 | -0.0156972 | 0.03506439 | -1.4156511 | 2.26215716 | FALSE | 1.83311293 | FALSE |
| 11 | 0.01031873 | 0.03832875 | 0.85133676 | 2.26215716 | FALSE | 1.83311293 | FALSE |
| 20 | 0.00322709 | 0.02959519 | 0.34481824 | 2.26215716 | FALSE | 1.83311293 | FALSE |
| 34 | -0.024741 | 0.03290195 | -2.3779145 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 61 | -0.0287251 | 0.02977224 | -3.051055 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 108 | -0.0430279 | 0.03538323 | -3.8454976 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 192 | -0.0469323 | 0.03048947 | -4.8676762 | 2.26215716 | TRUE | 1.83311293 | TRUE |
| 341 | -0.0134573 | 0.03357964 | -1.2673054 | 2.26215716 | FALSE | 1.83311293 | FALSE |

NetLogo Code and Screen Shots

## Smart Team Model Screen Shot



```
extensions [nw cf csv matrix]
globals [
  Dij-File             ;;file name for file containing the Dij matrix
  Age                  ;;variable used to limit search for new links at
given Threshold value
  Threshold            ;;threshold for adding team links, links added
when random-float Threshold < P
  Lambda               ;;network parameter for multiscale networks
  Zeta                 ;;network parameter for multiscale networks
  Dij                  ;;matrix containing depth of common nodes
between nodes i and j
  Nodes                ;;total number of nodes in the network
  RFI-Rho-Max          ;;maximum RFI congestion centrality of network
  RFI-Congested-Node   ;;node with maximum RFI congestion centrality
  A-Rho-Max            ;;maximum artifact congestion centrality
  A-Congested-Node     ;;node with maximum artifact congestion
centrality
  E-Rho-Max            ;;maximum effective congestion centrality
  BC-Max               ;;maximum normalized betweenness centrality of
network
  Central-Node         ;;node with maximum betweenness centrality
  Manager-Approvals    ;;number of artifacts approved by manager
  Supervisor-Approvals ;;number of artifacts approved by supervisor
  GCC                  ;;global clustering coefficient of network
  MPL                  ;;mean path length of network
  delta-vector
  delta-slope
]

breed [workers worker]      ;;Agent breed representing workers in the
organization
breed [artifacts artifact]  ;;Meta-agent representing work products
breed [RFIs RFI]            ;;Meta-agent for information requests

;;Workers represent the individuals in the organizational hierarchy,
i.e., the individual
;;engineers, designers, supervisors, managers, etc who make up the
organization
workers-own [
  Depth            ;;Worker's depth in hierarchy
  My-Department    ;;Worker's department, the main branches of hierarchy
  My-Supervisor    ;;Worker's immediate superior in hierarchy
  My-Team          ;;Worker's team assignment, used for matrix and BCCWG
networks
  Capacity         ;;Worker's capacity to perform work in a single tick
  Artifact-Queue   ;;Worker's artifact in box (list)
  Artifact-Count   ;;Number of artifacts processed by worker
  Hold-Queue       ;;List of artifacts placed on hold pending resolution
of RFI (list)
  RFI-Queue        ;;Worker's RFI in box (list)
  RFI-Count        ;;Number of RFIs processed by worker
]
```

```
;;Artifacts represent design products, such as drawings, calculations,
specifications, or
;;other documents created by the design organization
artifacts-own [
  A-Originator    ;;Worker to whom artifact initially assigned
[Agentset]
  A-Status        ;;Artifact status: Open, Hold or Complete
  A-Age           ;;Age of the artifact, incremented until artifact
complete
]

;;Requests for Information (RFIs) represent questions or requests to
other workers
;;for information
RFIs-own [
  R-Artifact      ;;Artifact the RFI relates to [Agentset]
  R-Originator    ;;Worker who originated the RFI [Agentset]
  R-Target        ;;Worker to whom the RFI was sent [Agentset]
  R-Status        ;;Status of RFI: Open, Answered, or Complete
  R-Age           ;;Age of RFI
]

;;organizational links represent the hierarchical backbone structure of
the organization,
;;defined by levels and branching ratio
undirected-link-breed [organizations organization]

;;team links represent the links added to the organization, according
to network construction algorithm
undirected-link-breed [teams team]



to setup
  clear-all
  reset-ticks

  set Manager-Approvals 0
  set Supervisor-Approvals 0
  set delta-vector []
  set Nodes (Branching-Ratio ^ Levels - 1) / (Branching-Ratio - 1)

  ;;create organizational hierarchy based on number of levels and
branching
  ;;ratio specified by user
  foreach n-values Levels [ [?1] -> ?1 ] [ [?1] ->
    ifelse ?1 = 0
    [
      ;;create worker 0
      create-workers 1 [
        set Depth 0
        set My-Department 0
        set My-Supervisor 0
        set My-Team 0
        set Capacity Worker-Capacity
        set Artifact-Queue [ ]
        set Artifact-Count 0
```

```
      set Hold-Queue [ ]
      set RFI-Queue [ ]
      set RFI-COunt 0
    ]
  ]
  [
    ;;make new level in hierarchy
    make-level ?1
  ]
]
ask workers [
  set shape "person"
  set color white
]
;;arrange workers in radial layout with worker 0 at center
layout-radial workers organizations (worker 0)

;;create the Dij matrix from file
set Dij matrix:make-constant Nodes Nodes 0
ifelse Network-Type = "Matrix" [
  set Dij-File (word Dij-Name "_" Number-of-Teams ".csv")
]
[
  set Dij-File word Dij-Name ".csv"
]
file-open Dij-File
set Dij matrix:from-row-list csv:from-file Dij-File
file-close

;;create team structure based on user inputs
cf:when
cf:case [Network-Type = "Random"] [
  set Threshold 0.9999
  set Lambda 1000000000
  set Zeta 1000000000
  while [count teams < M] [
    let source one-of workers
    let target one-of workers with [who != [who] of source]
    construct-team source target
  ]
]
cf:case [Network-Type = "Matrix"] [
  ;;create matrix-organization teams
  ;;create manager of the project organization
  create-workers 1 [
    set shape "person"
    set color orange
    set Depth 1
    set My-Department who
    set My-Supervisor worker 0
    set Capacity Worker-Capacity
    set Artifact-Queue [ ]
    set Artifact-Count 0
    set Hold-Queue [ ]
    set RFI-Queue [ ]
    set RFI-Count 0
    create-organization-with worker 0
```

```
    ]
    ;;create project managers, heads of each of the project teams
    foreach range Number-of-Teams [ [?1] ->
      create-workers 1 [
        set Depth 2
        set My-Department Branching-Ratio + 1
        set My-Supervisor Nodes
        set My-Team (?1 + 1)
        set Capacity Worker-Capacity
        set Artifact-Queue []
        set Artifact-Count 0
        set Hold-Queue []
        set RFI-Queue []
        set RFI-Count 0
        set shape "person"
        set color red
        create-organization-with worker Nodes
      ]
    ]
    ;;create links to team members
    while [count teams < M] [
      let source []
      let target []
      let t random Number-of-Teams + 1
      let S count workers with [My-Team = t]
      let P exp(- S / chi)
      ifelse (random-float 1 < P or count workers with [My-Team = t] <
3) and count workers with [My-Team = 0 and Depth > 2] > 1
      [
        set source one-of workers with [color = red]
        set target one-of workers with [Depth > 2 and My-Team = 0]
        ask target [
          if not team-neighbor? source [
            set My-Team [My-Team] of source
            create-team-with source [set color blue]
          ]
        ]
      ]
      [
        set source one-of workers with [My-Team = t and color = white]
        set target one-of workers with [My-Team = t and who != [who] of
source and color = white]
        ask target [
          if not team-neighbor? source [
            create-team-with source [set color turquoise]
          ]
        ]
      ]
    ]
    layout-radial workers organizations (worker 0)
  ]
  cf:case [Network-Type = "BCCWG"] [
    ;;create team around boards, centers, cells and working groups
model
    ;;first, identify team leads
    foreach range Number-of-Teams [ [?1] ->
```

```
      ask one-of workers with [My-Team = 0 and Depth = 1 or Depth = 2]
[
        set My-Team (?1 + 1)
        set color red
      ]
    ]
    ;;add team members
    while [count teams < M] [
      let source []
      let target []
      let t random Number-of-Teams + 1
      ;;show t
      let S count workers with [My-Team = t]
      ;;show S
      let P exp(- S / chi)
      ;;show P
      ifelse (random-float 1 < P or count workers with [My-Team = t] <
3) and count workers with [My-Team = 0 and Depth > 2] > 1
      [
        set source one-of workers  with [color = red]
        set target one-of workers with [color = white and Depth > 2 and
My-Team = 0]
        ask target [
          if not team-neighbor? source [
            create-team-with source [set color blue]
            set My-Team [My-Team] of source
          ]
        ]
      ]
      [
        set source one-of workers with [My-Team = t and color = white]
        set target one-of workers with [My-Team = t and who != [who] of
source and color = white]
        ask target [
          if not team-neighbor? source [
            create-team-with source [set color turquoise]
          ]
        ]
      ]
    ]
  ]
  cf:else [
    ;;create a multiscale network
    set Threshold 0.3768
    set Lambda 0.5
    set Zeta 0.5
    while [count teams < M ] [
      let source one-of workers
      let target one-of workers with [who != [who] of source]
      construct-team source target
      ;; if age counter reaches Nodes^2, reduce threshold to lower
value
      set Age Age + 1
      if Age > Nodes ^ 2 [
        set Threshold Threshold ^ 2
        show Threshold
        set Age 0
```

```
      ]
    ]
  ]

  ;;calculate network parameters
  set GCC global-clustering-coefficient
  ;;show GCC
  set MPL nw:mean-path-length
  ;;show MPL
  ;;set BC-Max 2 * max [nw:betweenness-centrality] of workers / ((Nodes
- 1) * (Nodes - 2))
  ;;set Central-Node max-one-of workers [nw:betweenness-centrality]


end


to go

  ask workers [
    ;;perform work at each tick up to worker capacity
    repeat Capacity [
      cf:when
      cf:case [empty? RFI-Queue and not empty? Artifact-Queue] [
        ;;process artifact
        process-Artifact worker who
      ]
      cf:case [not empty? RFI-Queue and empty? Artifact-Queue] [
        ;;process RFI
        process-RFI worker who
      ]
      cf:case [not empty? RFI-Queue and not empty? Artifact-Queue] [
        ;;process an artifact or RFI according to coin toss
        ifelse random-float 1 < Artifact-Preference [
          ;;artifact selected
          process-Artifact worker who
        ]
        [
          ;;RFI selected
          process-RFI worker who
        ]
      ]
      cf:else [
        ;;queues empty, take no action
      ]
    ]
  ]

  ;;create new Artifacts
  make-new-Artifacts

  ;;increment counters
  increment

  ifelse ticks <= 10
  [
```

```
    let delta count artifacts - count artifacts with [A-Status =
"Complete"]
    set delta-vector lput delta delta-vector
  ]
  [
    set delta-slope simple-slope delta-vector
    set delta-vector remove-item 0 delta-vector
    let delta count artifacts - count artifacts with [A-Status =
"Complete"]
    set delta-vector lput delta delta-vector
  ]

  tick
  if ticks > 250 [
    ;;calculate congestion probability
    set RFI-Rho-max maximum-RFI-congestion-centrality
    set RFI-Congested-Node max-one-of workers [RFI-Count]
    set A-Rho-Max maximum-A-congestion-centrality
    set A-Congested-Node max-one-of workers [Artifact-Count]
    ;;show rho-max
    ;;show Congested-Node
    set E-Rho-Max effective-rho-max

    ;;stop execution
    stop
  ]
end

;;procedure to process RFIs
to process-RFI [wrkr]
        let r first RFI-Queue
        let source [R-Originator] of RFI r
        let target [R-Target] of RFI r
        let here worker who
        let next []
        ifelse here = source and [R-Status] of RFI r = "Answered"
        [
          ;;I am the originator of an answered RFI

          ;;remove related Artifact from hold queue and return to top
of
          ;;artifact queue and set artifact status to Open
          let a [R-Artifact] of RFI r
          set Hold-Queue remove a Hold-Queue
          ifelse empty? Artifact-Queue
          [ set Artifact-Queue (list a) ]
          [ set Artifact-Queue fput a Artifact-Queue]
          ask Artifact a [
            set A-Status "Open"
          ]
          ;;close RFI
          ask RFI r [
            set R-Status "Complete"
            set xcor 23
            set ycor 23
            set color blue
          ]
```

```
          ;;and remove RFI from my RFI queue
          set RFI-Queue remove r RFI-Queue
        ]
        [
          ;;I am NOT the originator of an answered RFI...
          ;;increment RFI counter
          set RFI-Count RFI-Count + 1

          ifelse here = target [
            ;;I am the target of the RFI
            ;;set RFI status to Answered
            ask RFI r [
              set R-Status "Answered"
            ]
            ;;return RFI to originator and put on top of RFI queue
            ask source [
              ifelse empty? RFI-Queue
              [ set RFI-Queue (list r) ]
              [ set RFI-Queue fput r RFI-Queue ]
            ]
            ask RFI r [
              set xcor [xcor] of source
              set ycor [ycor] of source
              face worker 0
              fd -2
              set color green
            ]
            ;;remove RFI from my RFI Queue
            set RFI-Queue remove r RFI-Queue
          ]
          [
            ;;I am neither the source of an answered RFI
            ;;nor the target of the RFI, so
            ;;pass RFI to next worker on route to target
            set next one-of organization-neighbors with [Depth <
[Depth] of here]
            let t [who] of target
            let da-kt matrix:get Dij who t
            foreach sort link-neighbors [ [?the-neighbor] ->
              if Network-Type = "Matrix" or Network-Type = "BCCWG" [
                ask ?the-neighbor [
                  if team-neighbor? target [
                    set next ?the-neighbor
                    set da-kt [Depth] of ?the-neighbor
                  ]
                ]
              ]
              let k [who] of ?the-neighbor
              let dd matrix:get Dij k t
              if dd > da-kt [
                set next ?the-neighbor
                set da-kt dd
              ]
            ]
            ;;add RFI to bottom of next worker's queue
            ask next [
              ifelse empty? RFI-Queue
```

```
              [ set RFI-Queue (list r) ]
              [ set RFI-Queue lput r RFI-Queue ]
            ]
            ask RFI r [
              set xcor [xcor] of next
              set ycor [ycor] of next
              face worker 0
              fd -2
            ]
            ;;remove RFI from my RFI Queue
            set RFI-Queue remove r RFI-Queue
          ]
        ]
end

;;procedure to process Artifacts
to process-Artifact [wrkr]
          let n first Artifact-Queue
          let here worker who
          let d Depth
          let originator [A-Originator] of artifact n

          ifelse d = 1 or (Decentralize? and d = 2 and random-float 1 <
Decentralize-Preference) [
            ;;I am the approver, i.e., worker (Manager) at depth 1
            ;;then approve artifact and close
            ifelse d = 1
            [
              set Manager-Approvals Manager-Approvals + 1
            ]
            [
              set Supervisor-Approvals Supervisor-Approvals + 1
            ]
            ask Artifact n [
              Set A-Status "Complete"
              set xcor -23
              set ycor 23
              set color blue
            ]
            ;;remove Artifact from my Artifact Queue
            set Artifact-Queue remove n Artifact-Queue
            ;;increment Artifact Count
            set Artifact-Count Artifact-Count + 1
          ]
          [
            ;;I am not the approver
            ifelse here = originator [
              ;;I am the originator of Artifact
              ;;Evaluate complexity
              ;;Only Artifact originator's evaluate complexity
              ifelse random 10 < Complexity
              [
                ;;Situation complex, initiate RFI
                hatch-RFIs 1 [
                  set R-Artifact n
                  set R-Originator here
```

```
                  ;;Select RFI based on complexity.  At low complexity,
tasks are decomposable
                  ;;and are passed in same department.  At high
complexity, tasks are not
                  ;;decomposable and are assigned to another worker at
random
                  let t []
                  ifelse random 10 < Complexity
                  [
                    set t one-of workers with [who != [who] of here and
color = white and Depth = Levels - 1]
                    set shape "flag"
                    set color red
                  ]
                  [
                    set t one-of workers with [who != [who] of here and
My-Department = [My-Department] of here and Depth = Levels - 1]
                    set shape "flag"
                    set color white
                  ]
                  set R-Target t
                  set R-Status "Open"
                  ;;set shape "flag"
                  ;;set color white
                  set xcor [xcor] of here
                  set ycor [ycor] of here
                  face worker 0
                  fd -2
                  let r who
                  ;;put RFI at bottom of my RFI Queue
                  ask here [
                    ifelse empty? RFI-Queue
                    [ set RFI-Queue (list r) ]
                    [ set RFI-Queue lput r RFI-Queue ]
                  ]
                ]
              ;;move artifact to Hold Queue
              ask here [
                ifelse empty? Hold-Queue
                [ set Hold-Queue (list n) ]
                [ set Hold-Queue lput n Hold-Queue ]
                set Artifact-Queue remove n Artifact-Queue
              ]
              ]
              [
                ;;Situation NOT complex, process Artifact
                ;;Identify next worker up chain of command
                let next one-of organization-neighbors with [Depth < d]
                ;;and move Artifact to bottom of next worker's Artifact
Queue
                ask next [
                  ifelse empty? Artifact-Queue
                  [ set Artifact-Queue (list n) ]
                  [ set Artifact-Queue lput n Artifact-Queue ]
                ]
                ask Artifact n [
                  set xcor [xcor] of next
```

```
          set ycor [ycor] of next
          face worker 0
          fd -1
        ]
        ;;remove Artifact from my Artifact Queue
        set Artifact-Queue remove n Artifact-Queue
        ;;increment Artifact Count
        set Artifact-Count Artifact-Count + 1
      ]
    ]
    [
      ;;I Am neither the originator nor approver of Artifact
      ;;so pass Artfiact to next worker in the organizational
      ;;chain of command

      ;;Identify next worker up chain of command
      let next one-of organization-neighbors with [Depth < d]
      ;;move artifact to bottom of next worker's Artifact Queue
      ask next [
        ifelse empty? Artifact-Queue
        [ set Artifact-Queue (list n) ]
        [ set Artifact-Queue lput n Artifact-Queue ]
      ]
      ask Artifact n [
        set xcor [xcor] of next
        set ycor [ycor] of next
        face worker 0
        fd -1
      ]
      ;;remove Artifact from my Artifact Queue
      set Artifact-Queue remove n Artifact-Queue
      ;;increment Artifact Count
      set Artifact-Count Artifact-Count + 1
    ]
  ]
end

;;procedure to add levels (i.e., rows) to hierarchical organizational
structure
to make-level [row]
  let b Branching-Ratio                ;;let b equal the branching
ratio
  let W b ^ (row - 1)                  ;;let W equal the number of
workers in pervious row
  let N (b ^ (row - 1) - 1) / (b - 1)  ;;let N equal number of workers
in all previous rows of hierarchy
  ;;for each of the workers in the previous row
  foreach n-values W [ [?1] -> ?1 ] [ [?1] ->
    ;;show ? + N
    ;;create b new workers and link to worker ?+N in previous row
    create-workers b [
      set Depth row
      ifelse row = 1 [
        set My-Department who
      ]
      [
        set My-Department [My-Department] of worker (?1 + N)
```

```
      ]
      set My-Supervisor worker (?1 + N)
      set My-Team 0
      set Capacity Worker-Capacity
      set Artifact-Queue [ ]
      set Artifact-Count 0
      set Hold-Queue [ ]
      set RFI-Queue [ ]
      set RFI-Count 0
      create-organization-with worker (?1 + n)
    ]
  ]

end

;;procedure to create team links based on Watts' stochastic rule
to construct-team [source target]
  let i [who] of source
  let d-source [Depth] of source
  let j [who] of target
  let d-target [Depth] of target
  let da matrix:get Dij i j
  let d1 abs (d-source - da)
  let d2 abs (d-target - da)
  ;;if d1+d2 <2, no team link created
  if (d1 + d2) >= 2 [
    let x12 (d1 ^ 2 + d2 ^ 2 - 2) ^ (1 / 2)
    let P exp ((- da) / Lambda) * exp ((- x12) / Zeta)
    if random-float Threshold < P [
      ask source [
        if not team-neighbor? target [
          create-team-with target [
            set color blue
          ]
          set Age 0
        ]
      ]
    ]
  ]

end

;;procedure to make new Artifacts
;;New Artifacts arrive according to a random Poisson process with mean
equal to Task-Arrival-Rate
;;Each new Artifact is randomly assigned to a worker at the lowest
level
;;of the organizational hierarchy
to make-new-Artifacts
  create-artifacts random-poisson Task-Arrival-Rate [
    set A-Status "Open"
    let n who
    let w one-of workers with [Depth = Levels - 1]
    set A-Originator w
    ;;Put artifact at bottom of originator's Artifact Queue
    ask w [
      ifelse empty? Artifact-Queue
```

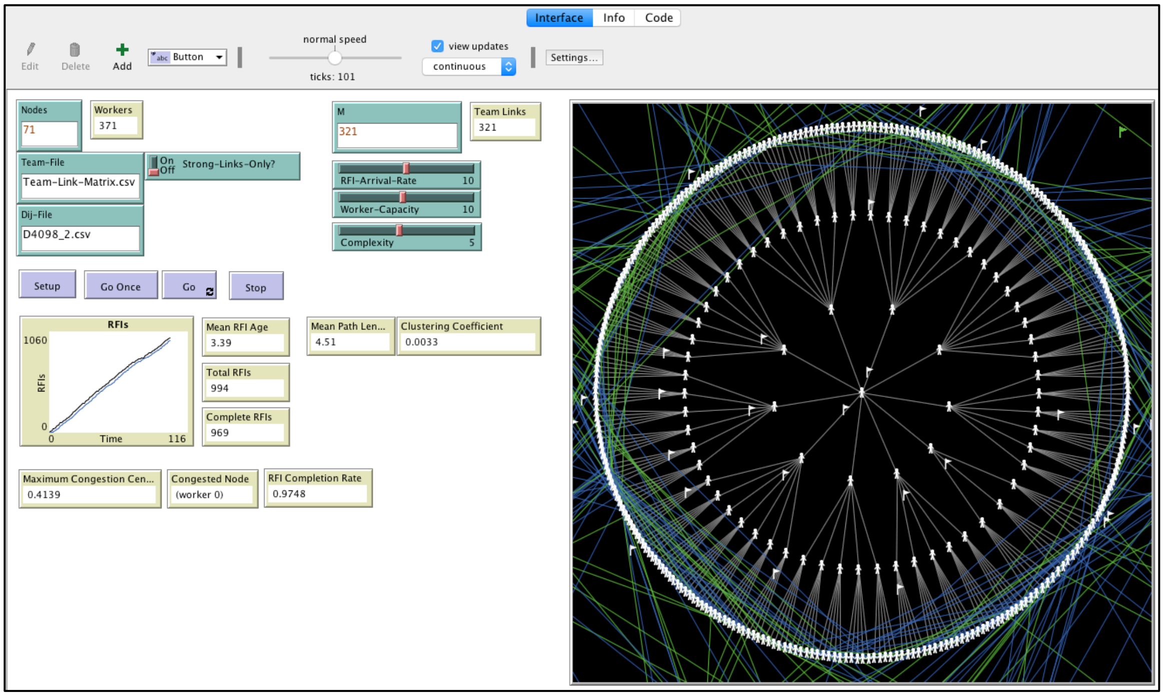
```
      [ set Artifact-Queue (list n) ]
      [ set Artifact-Queue lput n Artifact-Queue ]
    ]
    ;;place Artifact near originator
    set xcor [xcor] of w
    set ycor [ycor] of w
    face worker 0
    fd -1
    set color white
    set shape "circle 2"
  ]

end

;;procedure to increment counters
to increment

  ;;Increment age of open RFIs
  ask RFIs with [R-Status != "Complete"] [
    set R-Age R-Age + 1
  ]

  ;;Increment age of open Artifacts
  ask Artifacts with [A-Status != "Complete"] [
    Set A-Age A-Age + 1
  ]
end



;;procedure to calculate and report global clustering coefficient,
using method in nw extensions documentation
to-report global-clustering-coefficient
  let closed-triplets sum [ nw:clustering-coefficient * count my-links
* (count my-links - 1 ) ] of workers
  let triplets sum [ count my-links * (count my-links - 1) ] of workers
  report closed-triplets / triplets
end

;;procedures to calculate and report rho-max, the maximum congestion
centrality, for RFIs and Artifacts
to-report maximum-RFI-congestion-centrality
  let r-max max [RFI-Count] of workers
  report r-max / count RFIs
end

to-report maximum-A-congestion-centrality
  let A-max max [Artifact-Count] of workers
  report A-max /  (Task-Arrival-Rate * ticks)
end

;;procedure to calculate and report the maximum work rate of workers
to-report effective-rho-max
  let er-max max [RFI-Count + Artifact-Count] of workers
  ifelse ticks > 0 [
    report er-max / (ticks * Task-Arrival-Rate)
  ]
```

```
  [
    report 0
  ]
end

;;procedure to calculate Artifact Rate
to-report Artifact-Rate
  ifelse ticks < 1 [
    report 0
  ]
  [
    report count Artifacts with [A-Status = "Complete"] / (ticks *
Task-Arrival-Rate)
  ]
end

;;procedure to calculate RFI arrival rate
to-report RFI-Arrival-Rate
  ifelse ticks < 1 [
    report 0
  ]
  [
    report count RFIs / (ticks * Task-Arrival-Rate)
  ]
end

;;procedure to calculate RFI closure rate
to-report RFI-Closure-Rate
  ifelse ticks < 1 [
    report 0
  ]
  [
    report count RFIs with [R-Status = "Complete"] / (ticks * Task-
Arrival-Rate)
  ]
end

to-report simple-slope [y-vector]
  let x-vector [1 2 3 4 5 6 7 8 9 10]
  let x-bar mean x-vector
  let y-bar mean y-vector
  let sum-dx2 0
  foreach range 10 [ [?1] ->
    let xi item ?1 x-vector
    set sum-dx2 sum-dx2 + (xi - x-bar) ^ 2
  ]
  let sum-xy 0
  foreach range 10 [ [?1] ->
    let xi item ?1 x-vector
    let yi item ?1 y-vector
    set sum-xy sum-xy + (xi - x-bar) * (yi - y-bar)
  ]
  report sum-xy / sum-dx2
end
```

# APPENDIX 4: The PW4098 Case Study

## NetLogo Code and Screen Shots

### PW4098 Case Study Information Exchange Model – Screen Shot




### PW4098 Case Study NetLogo Code

```
extensions [ nw cf csv matrix]
globals [
  Adj                  ;;adjacency matrix for organizagtional hierarchy
  Tij                  ;;matrix of team links
  Dij                  ;;matrix containing depth of common node between
nodes i and j
  Rho-Max              ;;maximum congestion centrality of network
  Congested-Node       ;;node with maximum congestion centrality
  BC-Max               ;;maximum normalized betweenness centrality of
network
  Central-Node         ;;node with maximum betweenness centrality
  GCC                  ;;global clustering coefficient of network
  MPL                  ;;mean path length of network
]



breed [workers worker]        ;;Agent breed representing workers in the
organization
breed [RFIs RFI]              ;;Meta-agent for information requests
```

```
;;Workers represent the individuals in the organizational hierarchy, i.e., the individual
;;engineers, designers, supervisors, managers, etc who make up the organization
workers-own [
  Depth           ;;Worker's depth in hierarchy
  My-Department   ;;Worker's department, equal to worker number at level 1
  My-Supervisor   ;;Worker's superior in the organizational structure
  Capacity        ;;Worker's capacity to perform work in a single tick
  RFI-Queue       ;;Worker's RFI in box (list)
  RFI-Count       ;;Number of RFIs completed by worker
]

;;Requests for Information (RFIs) represent questions or requests to other workers
RFIs-own [
  R-Originator    ;;Worker who originated the RFI [Agentset]
  R-Target        ;;Worker to whom the RFI was sent [Agentset]
  R-Status        ;;Status of RFI: Open, Answered, or Complete
  R-Age           ;;Age of RFI
]

;;organizational links represent the hierarchical backbone structure of the organization, defined by levels and branching ratio
undirected-link-breed [organizations organization]

;;team links represent the links added to the organization, according to network construction algorithm
undirected-link-breed [teams team]



to setup
  clear-all
  reset-ticks

  ;;create organizational hierarchy
  ;;create worker 0
  create-workers 1 [
    set Depth 0
    set My-Department 0
    set My-Supervisor 0
    set Capacity Worker-Capacity
    set RFI-Queue []
    set RFI-Count 0
  ]
  ;;create workers at level 1
  create-workers 10 [
    set Depth 1
    set My-Department who
    set My-Supervisor 0
    set Capacity Worker-Capacity
    set RFI-Queue []
    set RFI-Count 0
    create-organization-with worker 0
```

```
  ]
  ;;create workers at level 2
  let node-count (list 7 7 7 5 5 6 7 7 3 6)
  foreach (range 1 11) [ [?1] ->
    let num first node-count
    create-workers num [
      set Depth 2
      set My-Department ?1
      set My-Supervisor ?1
      set Capacity Worker-Capacity
      set RFI-Queue []
      set RFI-Count 0
      create-organization-with worker ?1
    ]
    set node-count remove-item 0 node-count
  ]
  ;;create workers at level 3
  foreach (range 11 71) [ [?the-worker] ->
    create-workers 5 [
      set Depth 3
      set My-Department [My-Department] of worker ?the-worker
      set My-Supervisor ?the-worker
      set Capacity Worker-Capacity
      set RFI-Queue []
      set RFI-Count 0
      create-organization-with worker ?the-worker
    ]
  ]

  ;;ask workers to set their shape to person and color to white
  ask workers [
    set shape "person"
    set color white
  ]

  ;;arrange workers in radial layout with worker 0 at center
  layout-radial workers organizations (worker 0)

  ;;create team links from team link file
  file-open Team-File
  set Tij matrix:from-row-list csv:from-file Team-File
  file-close

  foreach range Nodes [ [?row] ->
    foreach (range ?row Nodes) [ [?column] ->
      let link-marker matrix:get Tij ?row ?column
      if link-marker = 3 [
        let s one-of workers with [Depth = 3 and My-Supervisor = [who]
of worker ?row]
        let t one-of workers with [Depth = 3 and My-Supervisor = [who]
of worker ?column]
        ask t [
          create-team-with s [set color blue]
        ]
      ]
      if  link-marker = 2 and not strong-links-only? [
```

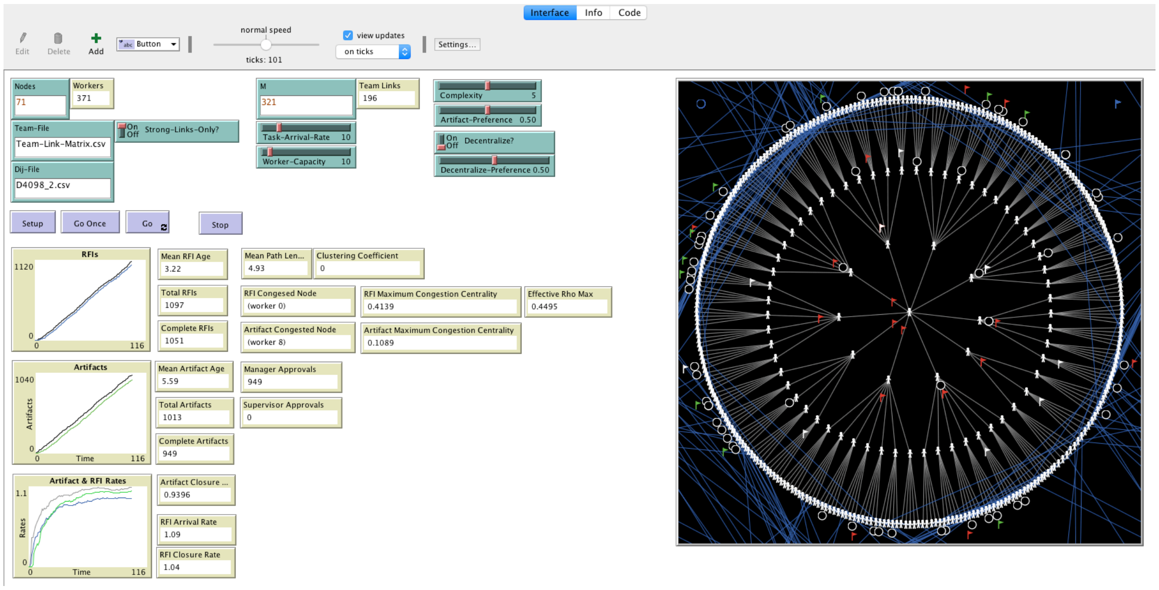
```
        let s one-of workers with [Depth = 3 and My-Supervisor = [who]
of worker ?row]
        let t one-of workers with [Depth = 3 and My-Supervisor = [who]
of worker ?column]
        ask t [
          create-team-with s [set color green]
        ]
      ]
    ]
  ]

  ;;remove links at random until only m links remain
  while [count teams > M] [
    ask one-of teams [die]
  ]



  ;;create Dij matrix
  set Dij matrix:make-constant Nodes Nodes 0

  file-open Dij-File
  set Dij matrix:from-row-list csv:from-file Dij-File
  file-close

  ;;create team structure based on user inputs


  ;;calculate and show network parameters
  set GCC global-clustering-coefficient
  ;;show GCC
  set MPL nw:mean-path-length
  ;;show MPL

end


to go

  ask workers [
    ;;show who
    repeat Capacity [
      if not empty? RFI-Queue [
        ;;If I have RFIs in my RFI Queue, then take RFI actions
        set RFI-Count RFI-Count + 1
        let r first RFI-Queue
        let source [R-Originator] of RFI r
        let target [R-Target] of RFI r
        let here worker who
        ifelse here = target
        [
          ;;I am the target of RFI, so close RFI
          ask RFI r [
             set R-Status "Complete"
             set xcor 23
             set ycor 23
             set color green
```

```
          ]
          ;;remove RFI from my RFI queue
          set RFI-Queue remove r RFI-Queue
        ]
        [
          ;;I am not the target of RFI, so pass to next worker on path
          ;;using the Dodds Watts Sabel assumption regarding pseudo-
global knowledge
          ;;show target
          ;;let d-here Depth
          let next one-of organization-neighbors with [Depth < [Depth]
of here]
          ;;show next
          let t [who] of target
          let da-kt matrix:get Dij who t
          foreach sort link-neighbors [ [?the-neighbor] ->
            let k [who] of ?the-neighbor
            let dd matrix:get Dij k t
            ;;show k
            ;;show dd
            if dd > da-kt [
              set next ?the-neighbor
              ;;show next
              set da-kt dd
            ]
          ]
          ;;add RFI to next worker's RFI queue
          ask next [
            ifelse empty? RFI-Queue
            [ set RFI-Queue (list r) ]
            [ set RFI-Queue lput r RFI-Queue ]
          ]
          ask RFI r [
            set xcor [xcor] of next
            set ycor [ycor] of next
            face worker 0
            fd -2
          ]
          ;;remove RFI from my RFI queue
          set RFI-queue remove r RFI-Queue
        ]
      ]
    ]
  ]

  ;;create new RFIs
  make-new-RFIs

  ;;increment counters
  increment

  tick
  if ticks > 100 [
    ;;calculate congestion probability, rho
    set rho-max maximum-congestion-centrality
    set Congested-Node max-one-of workers [RFI-Count]
    show rho-max
```

```
    show Congested-Node

    ;;stop execution
    stop
  ]
end


;;procedure to make new RFIs
;;New RFIs arrive according to a random Poisson process with mean equal
to RFI-Arrival-Rate
;;Each new RFI is randomly assigned to an originating and target worker
to make-new-RFIs
  create-RFIs random-poisson RFI-Arrival-Rate [
    set R-Status "Open"
    let r who
    let s one-of workers with [who > 0]
    let t []
    ifelse random 10 < Complexity
    [
      set t one-of workers with [who != [who] of s]
    ]
    [
      set t one-of workers with [My-Department = [My-Department] of s]
    ]
    set R-Originator s
    set R-Target t
    set R-Age 0
    ;;Put artifact in originator's Artifact Queue
    ask s [
      ifelse empty? RFI-Queue
      [ set RFI-Queue (list r) ]
      [ set RFI-Queue lput r RFI-Queue ]
    ]
    ;;Place artifact near originator
    set xcor [xcor] of s
    set ycor [ycor] of s
    face worker 0
    fd -1
    set color white
    set shape "flag"
  ]

end

;;procedure to increment counters
to increment

  ;;Increment age of open RFIs
  ask RFIs with [R-Status != "Complete"] [
    set R-Age R-Age + 1
  ]
end
```

```
;;procedure to calculate and report global clustering coefficient,
using method in nw extensions documentation
to-report global-clustering-coefficient
  let closed-triplets sum [ nw:clustering-coefficient * count my-links
* (count my-links - 1 ) ] of workers
  let triplets sum [ count my-links * (count my-links - 1) ] of workers
  report closed-triplets / triplets
end

;;procedure to calculate and report rho-max, the maximum congestion
centrality
to-report maximum-congestion-centrality
  let r-max max [RFI-Count] of workers
  report r-max / (RFI-Arrival-Rate * ticks)
end
```

# PW4098 Artifact Screen Shot




# PW4098 Artifact Model Code from NetLogo

```
extensions [nw cf csv matrix]
globals [
  Adj                    ;;adjacency matrix for organizational hierarchy
  Tij                    ;;matrix of team links
  Dij                    ;;matrix containing depth of common node between
nodes i and j
  RFI-Rho-Max            ;;maximum RFI congestion centrality of network
  RFI-Congested-Node     ;;node with maximum RFI congestion centrality
  A-Rho-Max              ;;maximum artifact congestion centrality
  A-Congested-Node       ;;node with maximum artifact congestion
centrality
  E-Rho-Max              ;;maximum effective congestion centrality
  BC-Max                 ;;maximum normalized betweenness centrality of
network
  Central-Node           ;;node with maximum betweenness centrality
  Manager-Approvals      ;;number of artifacts approved by manager
  Supervisor-Approvals   ;;number of artifacts approved by supervisor
  GCC                    ;;global clustering coefficient of network
  MPL                    ;;mean path length of network
]

breed [workers worker]       ;;Agent breed representing workers in the
organization
breed [artifacts artifact]   ;;Meta-agent representing work products
breed [RFIs RFI]             ;;Meta-agent for information requests

;;Workers represent the individuals in the organizational hierarchy,
i.e., the individual
```

```
;;engineers, designers, supervisors, managers, etc who make up the
organization
workers-own [
  Depth            ;;Worker's depth in hierarchy
  My-Department    ;;Worker's department, the main branches of hierarchy
  My-Supervisor    ;;Worker's immediate superior in hierarchy
  My-Team          ;;Worker's team assignment, used for matrix and BCCWG
networks
  Capacity         ;;Worker's capacity to perform work in a single tick
  Artifact-Queue   ;;Worker's artifact in box (list)
  Artifact-Count   ;;Number of artifacts processed by worker
  Hold-Queue       ;;List of artifacts placed on hold pending resolution
of RFI (list)
  RFI-Queue        ;;Worker's RFI in box (list)
  RFI-Count        ;;Number of RFIs processed by worker
]

;;Artifacts represent design products, such as drawings, calculations,
specifications, or
;;other documents created by the design organization
artifacts-own [
  A-Originator     ;;Worker to whom artifact initially assigned
[Agentset]
  A-Status         ;;Artifact status: Open, Hold or Complete
  A-Age            ;;Age of the artifact, incremented until artifact
complete
]

;;Requests for Information (RFIs) represent questions or requests to
other workers
;;for information
RFIs-own [
  R-Artifact       ;;Artifact the RFI relates to [Agentset]
  R-Originator     ;;Worker who originated the RFI [Agentset]
  R-Target         ;;Worker to whom the RFI was sent [Agentset]
  R-Status         ;;Status of RFI: Open, Answered, or Complete
  R-Age            ;;Age of RFI
]

;;organizational links represent the hierarchical backbone structure of
the organization,
;;defined by levels and branching ratio
undirected-link-breed [organizations organization]

;;team links represent the links added to the organization, according
to network construction algorithm
undirected-link-breed [teams team]



to setup
  clear-all
  reset-ticks

  set Manager-Approvals 0
  set Supervisor-Approvals 0
```

```
;;create organizational hierarchy
;;create worker 0
create-workers 1 [
  set Depth 0
  set My-Department 0
  set My-Supervisor 0
  set My-Team 0
  set Capacity Worker-Capacity
  set Artifact-Queue []
  set Artifact-Count 0
  set Hold-Queue []
  set RFI-Queue []
  set RFI-Count 0
]
;;create workers at level 1
create-workers 10 [
  set Depth 1
  set My-Department who
  set My-Supervisor 0
  set My-Team 0
  set Capacity Worker-Capacity
  set Artifact-Queue []
  set Artifact-Count 0
  set Hold-Queue []
  set RFI-Queue []
  set RFI-Count 0
  create-organization-with worker 0
]
;;create workers at level 2
let node-count (list 7 7 7 5 5 6 7 7 3 6)
foreach (range 1 11) [ [?1] ->
  let num first node-count
  create-workers num [
    set Depth 2
    set My-Department ?1
    set My-Supervisor ?1
    set My-Team 0
    set Capacity Worker-Capacity
    set Artifact-Queue []
    set Artifact-Count 0
    set Hold-Queue []
    set RFI-Queue []
    set RFI-Count 0
    create-organization-with worker ?1
  ]
  set node-count remove-item 0 node-count
]
;;create workers at level 3
foreach (range 11 71) [ [?the-worker] ->
  create-workers 5 [
    set Depth 3
    set My-Department [My-Department] of worker ?the-worker
    set My-Supervisor ?the-worker
    set My-Team 0
    set Capacity Worker-Capacity
    set Artifact-Queue []
    set Artifact-Count 0
```

```
      set Hold-Queue []
      set RFI-Queue []
      set RFI-Count 0
      create-organization-with worker ?the-worker
    ]
  ]

  ask workers [
    set shape "person"
    set color white
  ]
  ;;arrange workers in radial layout with worker 0 at center
  layout-radial workers organizations (worker 0)

  ;;create team links from team link file
  file-open Team-File
  set Tij matrix:from-row-list csv:from-file Team-File
  file-close

  foreach range Nodes [ [?row] ->
    foreach range Nodes [ [?column] ->
      let link-marker matrix:get Tij ?row ?column
      if link-marker = 3 [
        let s one-of workers with [Depth = 3 and My-Supervisor = [who]
of worker ?row]
        let t one-of workers with [Depth = 3 and My-Supervisor = [who]
of worker ?column]
        ask t [
          create-team-with s [set color blue]
        ]
      ]
        if link-marker = 2 and not strong-links-only? [
          let s one-of workers with [Depth = 3 and My-Supervisor =
[who] of worker ?row]
          let t one-of workers with [Depth = 3 and My-Supervisor =
[who] of worker ?column]
          ask t [
            create-team-with s [set color green]
          ]
        ]
    ]
  ]

  ;;create the Dij matrix from file
  set Dij matrix:make-constant Nodes Nodes 0
  file-open Dij-File
  set Dij matrix:from-row-list csv:from-file Dij-File
  file-close

  ;;calculate network parameters
  set GCC global-clustering-coefficient
  ;;show GCC
  set MPL nw:mean-path-length
  ;;show MPL
  ;;set BC-Max 2 * max [nw:betweenness-centrality] of workers / ((Nodes
- 1) * (Nodes - 2))
  ;;set Central-Node max-one-of workers [nw:betweenness-centrality]
```

```
end


to go

  ask workers [
    ;;perform work at each tick up to worker capacity
    repeat Capacity [
      cf:when
      cf:case [empty? RFI-Queue and not empty? Artifact-Queue] [
        ;;process artifact
        process-Artifact worker who
      ]
      cf:case [not empty? RFI-Queue and empty? Artifact-Queue] [
        ;;process RFI
        process-RFI worker who
      ]
      cf:case [not empty? RFI-Queue and not empty? Artifact-Queue] [
        ;;process an artifact or RFI according to coin toss
        ifelse random-float 1 < Artifact-Preference [
          ;;artifact selected
          process-Artifact worker who
        ]
        [
          ;;RFI selected
          process-RFI worker who
        ]
      ]
      cf:else [
        ;;queues empty, take no action
      ]
    ]
  ]

  ;;create new Artifacts
  make-new-Artifacts

  ;;increment counters
  increment

  tick
  if ticks > 100 [
    ;;calculate congestion probability
    set RFI-Rho-max maximum-RFI-congestion-centrality
    set RFI-Congested-Node max-one-of workers [RFI-Count]
    set A-Rho-Max maximum-A-congestion-centrality
    set A-Congested-Node max-one-of workers [Artifact-Count]
    ;;show rho-max
    ;;show Congested-Node
    set E-Rho-Max effective-rho-max

    ;;stop execution
    stop
  ]
end
```

```
;;procedure to process RFIs
to process-RFI [wrkr]
        let r first RFI-Queue
        let source [R-Originator] of RFI r
        let target [R-Target] of RFI r
        let here worker who
        let next []
        ifelse here = source and [R-Status] of RFI r = "Answered"
        [
          ;;I am the originator of an answered RFI

          ;;remove related Artifact from hold queue and return to top
of
          ;;artifact queue and set artifact status to Open
          let a [R-Artifact] of RFI r
          set Hold-Queue remove a Hold-Queue
          ifelse empty? Artifact-Queue
          [ set Artifact-Queue (list a) ]
          [ set Artifact-Queue fput a Artifact-Queue]
          ask Artifact a [
            set A-Status "Open"
          ]
          ;;close RFI
          ask RFI r [
            set R-Status "Complete"
            set xcor 23
            set ycor 23
            set color blue
          ]
          ;;and remove RFI from my RFI queue
          set RFI-Queue remove r RFI-Queue
        ]
        [
          ;;I am NOT the originator of an answered RFI...
          ;;increment RFI counter
          set RFI-Count RFI-Count + 1

          ifelse here = target [
            ;;I am the target of the RFI
            ;;set RFI status to Answered
            ask RFI r [
              set R-Status "Answered"
            ]
            ;;return RFI to originator and put on top of RFI queue
            ask source [
              ifelse empty? RFI-Queue
              [ set RFI-Queue (list r) ]
              [ set RFI-Queue fput r RFI-Queue ]
            ]
            ask RFI r [
              set xcor [xcor] of source
              set ycor [ycor] of source
              face worker 0
              fd -2
              set color green
            ]
```

```
            ;;remove RFI from my RFI Queue
            set RFI-Queue remove r RFI-Queue
          ]
          [
            ;;I am neither the source of an answered RFI
            ;;nor the target of the RFI, so
            ;;pass RFI to next worker on route to target
            set next one-of organization-neighbors with [Depth <
[Depth] of here]
            let t [who] of target
            let da-kt matrix:get Dij who t
            foreach sort link-neighbors [ [?the-neighbor] ->
              let k [who] of ?the-neighbor
              let dd matrix:get Dij k t
              if dd > da-kt [
                set next ?the-neighbor
                set da-kt dd
              ]
            ]
            ;;add RFI to bottom of next worker's queue
            ask next [
              ifelse empty? RFI-Queue
              [ set RFI-Queue (list r) ]
              [ set RFI-Queue lput r RFI-Queue ]
            ]
            ask RFI r [
              set xcor [xcor] of next
              set ycor [ycor] of next
              face worker 0
              fd -2
            ]
            ;;remove RFI from my RFI Queue
            set RFI-Queue remove r RFI-Queue
          ]
        ]
end

;;procedure to process Artifacts
to process-Artifact [wrkr]
          let n first Artifact-Queue
          let here worker who
          let d Depth
          let originator [A-Originator] of artifact n

          ifelse d = 1 or (Decentralize? and d = 2 and random-float 1 <
Decentralize-Preference) [
            ;;I am the approver, i.e., worker (Manager) at depth 1
            ;;then approve artifact and close
            ifelse d = 1
            [
              set Manager-Approvals Manager-Approvals + 1
            ]
            [
              set Supervisor-Approvals Supervisor-Approvals + 1
            ]
            ask Artifact n [
              Set A-Status "Complete"
```

```
            set xcor -23
            set ycor 23
            set color blue
          ]
          ;;remove Artifact from my Artifact Queue
          set Artifact-Queue remove n Artifact-Queue
          ;;increment Artifact Count
          set Artifact-Count Artifact-Count + 1
        ]
        [
          ;;I am not the approver
          ifelse here = originator [
            ;;I am the originator of Artifact
            ;;Evaluate complexity
            ;;Only Artifact originator's evaluate complexity
            ifelse random 10 < Complexity
            [
              ;;Situation complex, initiate RFI
              hatch-RFIs 1 [
                set R-Artifact n
                set R-Originator here
                ;;Select RFI based on complexity.  At low complexity,
tasks are decomposable
                ;;and are passed in same department.  At high
complexity, tasks are not
                ;;decomposable and are assigned to another worker at
random
                let t []
                ifelse random 10 < Complexity
                [
                  set t one-of workers with [who != [who] of here and
color = white and Depth = 3]
                  set shape "flag"
                  set color red
                ]
                [
                  set t one-of workers with [who != [who] of here and
My-Department = [My-Department] of here and Depth = 3]
                  set shape "flag"
                  set color white
                ]
                set R-Target t
                set R-Status "Open"
                ;;set shape "flag"
                ;;set color white
                set xcor [xcor] of here
                set ycor [ycor] of here
                face worker 0
                fd -2
                let r who
                ;;put RFI at bottom of my RFI Queue
                ask here [
                  ifelse empty? RFI-Queue
                  [ set RFI-Queue (list r) ]
                  [ set RFI-Queue lput r RFI-Queue ]
                ]
              ]
```

```
              ;;move artifact to Hold Queue
              ask here [
                ifelse empty? Hold-Queue
                [ set Hold-Queue (list n) ]
                [ set Hold-Queue lput n Hold-Queue ]
                set Artifact-Queue remove n Artifact-Queue
              ]
              ]
              [
                ;;Situation NOT complex, process Artifact
                ;;Identify next worker up chain of command
                let next one-of organization-neighbors with [Depth < d]
                ;;and move Artifact to bottom of next worker's Artifact
Queue

                ask next [
                  ifelse empty? Artifact-Queue
                  [ set Artifact-Queue (list n) ]
                  [ set Artifact-Queue lput n Artifact-Queue ]
                ]
                ask Artifact n [
                  set xcor [xcor] of next
                  set ycor [ycor] of next
                  face worker 0
                  fd -1
                ]
                ;;remove Artifact from my Artifact Queue
                set Artifact-Queue remove n Artifact-Queue
                ;;increment Artifact Count
                set Artifact-Count Artifact-Count + 1
              ]
            ]
            [
              ;;I Am neither the originator nor approver of Artifact
              ;;so pass Artfiact to next worker in the organizational
              ;;chain of command

              ;;Identify next worker up chain of command
              let next one-of organization-neighbors with [Depth < d]
              ;;move artifact to bottom of next worker's Artifact Queue
              ask next [
                ifelse empty? Artifact-Queue
                [ set Artifact-Queue (list n) ]
                [ set Artifact-Queue lput n Artifact-Queue ]
              ]
              ask Artifact n [
                set xcor [xcor] of next
                set ycor [ycor] of next
                face worker 0
                fd -1
              ]
              ;;remove Artifact from my Artifact Queue
              set Artifact-Queue remove n Artifact-Queue
              ;;increment Artifact Count
              set Artifact-Count Artifact-Count + 1
            ]
          ]
end
```

```
;;procedure to add levels (i.e., rows) to hierarchical organizational
structure

;;procedure to make new Artifacts
;;New Artifacts arrive according to a random Poisson process with mean
equal to Task-Arrival-Rate
;;Each new Artifact is randomly assigned to a worker at the lowest
level
;;of the organizational hierarchy
to make-new-Artifacts
  create-artifacts random-poisson Task-Arrival-Rate [
    set A-Status "Open"
    let n who
    let w one-of workers with [Depth = 3]
    set A-Originator w
    ;;Put artifact at bottom of originator's Artifact Queue
    ask w [
      ifelse empty? Artifact-Queue
      [ set Artifact-Queue (list n) ]
      [ set Artifact-Queue lput n Artifact-Queue ]
    ]
    ;;place Artifact near originator
    set xcor [xcor] of w
    set ycor [ycor] of w
    face worker 0
    fd -1
    set color white
    set shape "circle 2"
  ]

end

;;procedure to increment counters
to increment

  ;;Increment age of open RFIs
  ask RFIs with [R-Status != "Complete"] [
    set R-Age R-Age + 1
  ]

  ;;Increment age of open Artifacts
  ask Artifacts with [A-Status != "Complete"] [
    Set A-Age A-Age + 1
  ]
end



;;procedure to calculate and report global clustering coefficient,
using method in nw extensions documentation
to-report global-clustering-coefficient
  let closed-triplets sum [ nw:clustering-coefficient * count my-links
* (count my-links - 1 ) ] of workers
  let triplets sum [ count my-links * (count my-links - 1) ] of workers
  report closed-triplets / triplets
end
```

```
;;procedures to calculate and report rho-max, the maximum congestion
centrality, for RFIs and Artifacts
to-report maximum-RFI-congestion-centrality
  let r-max max [RFI-Count] of workers
  report r-max / count RFIs
end

to-report maximum-A-congestion-centrality
  let A-max max [Artifact-Count] of workers
  report A-max /  (Task-Arrival-Rate * ticks)
end

;;procedure to calculate and report the maximum work rate of workers
to-report effective-rho-max
  let er-max max [RFI-Count + Artifact-Count] of workers
  ifelse ticks > 0 [
    report er-max / (ticks * Task-Arrival-Rate)
  ]
  [
    report 0
  ]
end

;;procedure to calculate Artifact Rate
to-report Artifact-Rate
  ifelse ticks < 1 [
    report 0
  ]
  [
    report count Artifacts with [A-Status = "Complete"] / (ticks *
Task-Arrival-Rate)
  ]
end

;;procedure to calculate RFI arrival rate
to-report RFI-Arrival-Rate
  ifelse ticks < 1 [
    report 0
  ]
  [
    report count RFIs / (ticks * Task-Arrival-Rate)
  ]
end

;;procedure to calculate RFI closure rate
to-report RFI-Closure-Rate
  ifelse ticks < 1 [
    report 0
  ]
  [
    report count RFIs with [R-Status = "Complete"] / (ticks * Task-
Arrival-Rate)
  ]
end
```

# BIBLIOGRAPHY


Browning, T. R. (1996). *Systematic IPT Integration in Lean Development Programs.* Massachussets Institute of Technology, Technology and Policy. Cambridge, Massachusetts: Massachussets Institute of Technology.

Browning, T. R. (2001). Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. *IEEE Transactions on Engineering Management*, 292-306.

Colfer, L. J., & Baldwin, C. Y. (2016). The Mirroring Hypothesis: Theory, Evidence and Excpetions. *Industrial and Corporate Change*, 709-738.

Conway, M. E. (1968). How do Committees Invent? *Datamation, 14*(5), 28-31.

Davis, S. M., & Lawrence, P. R. (1978). Problems of Matrix Organizations. *Harvard Business Review*, 131-142.

Department of Transportation. (2007, January). *Systems Engineering for Intelligent Transportation Systems.* Washington, D.C.: Department of Transportation. Retrieved April 21, 2017, from U.S. Department of Transportation, Federal Highway Administration: https://ops.fhwa.dot.gov/publications/seitsguide/

Dodds, P. S., Watts, D. J., & Sabel, C. F. (2003). Information Exchange and the Robustness of Organizational Networks. *Proceedngs of the National Academny of Sciences of the United States of America, 100*(21), 12516-12521.

Eppinger, S. D., & Browning, T. R. (2012). *Design Structure Matrix Methods and Applications.* Cambbridge, Massachusetts: The MIT Press.

Feldman, D. P., & Crutchfield, J. P. (1998). Measures of Statistical Complexity: Why? *Physics Letters A, 238*, 244-252.

Floricel, S., Michela, J. L., & Piperca, S. (2016). Complexity, Uncertainty-Reduction Strategies, and Project Performance. *International Journal of Project Management, 34*, 1360-1383.

Ford, R. C., & Randolph, W. A. (1992). Cross-Functional Structures: A Review and Integration of Matrix Organization and Project Management. *Journal of Management, 18*(2), 267-294.

Government Accountability Office. (2015). *Poor Outcomes Are the Predictable Consequence of the Prevalent Acquisition Culture.* Washington, D.C.: Government Accountability Office.

Henderson, R., & Clark, K. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Sciences Quarterly*, 9-30.

Kerzner, H. (2003). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling, Eighth Edition.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Lloyd, S. (2001). Measures of Complexity: A Non-Exhaustive Llist. *IEEE Control Systems Magazine*.

Magee, C. L. (2010). *Modeling Organizational Architecture.* Massachusetts Institute of Technology, Engineering Systems Division. Cambridge: Massachusetts Institute of Technology.

McCormack, A., Rusnak, J., & Baldwin, C. (2008). *Exploring the Duality Between Product and Organizational Architectures: a Test of the "Mirroring" Hypothesis.* Cambridge, Massachusetts: Harvard Business School.

Mitchell, M. (2009). *Complexity: A Guided Tour.* New York: Oxford University Press.

Miterev, M., Mancini, M., & Turner, R. (2017). Towards a Design for the Project-Based Organization. *International Journal of Project Management, 35*, 479-491.

Murray, B. T., Pinto, A., Skelding, R., de Weck, O., Zhu, H., Nair, S., . . . Zeidner, L. (2011). *META II: Complex System Design and Analysis (CODA).* Wright-Patterson Air Force Base, Ohio: Air Force Research Laboratory.

Northwestern University. (2017, March 21). *What is NetLogo?* Retrieved April 29, 2017, from NetLogo: https://ccl.northwestern.edu/netlogo/docs/

Page, S. E. (2009). *Understanding Complexity.* Chantilly, Virginia: The Teaching Company.

Patanakul, P., Kwak, Y. H., Zeikael, O., & Liu, M. (2016). What Impacts the Performance of Large-Scale Government Projects. *International Journal of Project Management, 34*, 452-466.

Poire, M. J., & Sabel, C. F. (1984). *The Second Industrial Divide: Possibilities for Prosperity.* New York: Basic Books.

Rand, W. (2016). An Introduction to Agent-Based Modeling. Santa Fe, New Mexico: Santa Fe Institute.

Rand, W., & Rust, R. T. (2011). Agent-Based Modeling in Marketing: Guidelines for Rigor. *International Journal of Research in Marketing*.

Reinersten, D. G. (2009). *The Principles of Product Development Flow, Second Generation Lean Product Development.* Redondo Beach, California: Celeritas Publishing.

Reinersten, D. G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development.* Redondo Beach, California: Celeritas Publishing.

Rowles, C. M. (1999). *System Integration Analysis of a Large Commercial Aircraft Engine.* Massachusetts Institute of Technology, System Design and Management Program. Cambridge: Massachusetts Institute of Technology.

Schelling, T. C. (2006). *Micromotives and Macrobehavior.* New York: W. W. Norton and Company.

Schnetler, R., Steyn, H., & van Staden, P. (2015). Characteristics of Matrix Structures, and Their Effects on Project Success. *South African Journal of Industrial Engineering, 26*(1), 11-26.

Senge, P. (1990). *The Fifth Discipline: The Art and Practice of Learning Organization.* New York: Doubleday.

Shannon, C. E. (1948, July, October). A Mathematical Theory of Communication. *Bell System Technical Journal, 27*, 379-423, 623-656.

Simon, H. A. (1996). *The Science of the Artifical, Third Edition.* Cambrdige, Massachusetts: The MIT Press.

Simpson, T. W., & Martins, J. R. (2012). Multidisciplinary Deisgn Optimization for Complex Engineered Systems: Report from a National Science Foundation Workshop. *53rd AIAA/SME/ASCE/ASC Structures, Structural Dynamics, and Materials Conference 2012* (pp. 3516-3520). Honolulu, Hawaii: Curran Associates, Inc.

Sinha, K. (2014). *Structural Complexity and its Implications for Design of Cyber-Physical Systems.* Cambridge, Massachusetts: Massachusetts Institute of Technology.

Sinha, K., & de Weck, O. L. (2012). Structural Complexity Metric for Engineered Complex Systems and its Applications. In M. M. Moriyoto Onishi (Ed.), *Gain Competitive Advantage by Managing Complexity, Proceedings of the 14th Inernational Dependency and Structure Modeling Conference* (pp. 181-192). Kyoto, Japan: Carl-Hansler-Verlag.

Sinha, K., & de Weck, O. L. (2013, April 15-18). A Networked-Based Strcutural Complexity Metric for Engineered Complex Systems. *Systems Conference (SYSCON), 2013 IEEE International*, 426-430.

Sinha, K., & de Weck, O. L. (2013, August 4-7). Structural Complexity Quantification for Engineered Complex Systems and Implications on System Architecture and Design. *Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 1-13.

Sinha, K., James, D., & de Weck, O. (2012). Interplay Between Product Architecture and Organizational Structure. In M. Onishi, M. Mayrer, K. Kirner, & U. Lindemann (Ed.), *14th International Dependency and Structure Modeling Conference 2012* (pp. 275-288). Kyoto, Japan: Carl-Hanser-Verlag.

Taleb, N. N. (2010). *The Black Swan: The Impact of the Highly Improbable.* New York: Random House.

Ulrich, K. (1995). The Role of Product Architecture in the Manufacturing Firm. *Research Policy, 24*, 419-440.

Ulrich, K. T., & Eppinger, S. D. (1995). *Product Design and Development.* New York: McGraw-Hill, Inc.

Vesterby, V. (2007). Measuring Complexity: Things That Go Wrong and How to Get It Right. *Emergence: Complexity and Organization. 10*, pp. 90-102. Stellenbosch, South Africa: Third International Workshop on Complexity and Philosophy.

Wade, N. M. (2012). *The Joint Forces Operations & Doctrine SMARTbook: Guide to Joint, Multinational and Interagency Operations, Third Edition.* Lakeland, Florida: The Lightning Press.

Watts, D. J. (1999). *Small Worlds: The Dynamics of Networls between Order and Radnomness.* Princeton, New Jersey: Princeton University Press.

Watts, D. J. (2003). *Six Degrees of Separation: The Science of a Connected Age.* New York: W. W. Norton & Company.

Watts, D. J., & Strogatz, S. H. (1998). Collective Dynamics of 'Small-World' Networks. *Nature*, 440-442.

Wilensky, U., & Rand, W. (2015). *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo.* Cambridge, Massachusetts: The MIT Press.

Wilhelm, T., & Hollunder, J. (2007). Information Theoretic Description of Networks. *Physica A: Statistical Mechanics and its Applications, 385*, 385-396.

Wintermantel, R. E. (2003). Application of the Matrix Organization in Industry. *Proceedings of the Tenth Project Management Institute Seminar Symposium* (pp. 493-497). Newton Square, Pennsylvania: Project Management Institute.

Zhi, J., & Mostafavi, A. (2017). Discovering Complexity and Emergent Properties in Project Systems: A New Approach to Understanding Project Performance. *International Journal of Project Management, 35*, 1-12.