

**THE EMERGENCE OF DOMINANT DESIGN(S) IN LARGE SCALE CYBER-
INFRASTRUCTURE SYSTEMS**

by

Eirini Ilana Diamanti

BA, Aristotle University, 2001

MSI, University of Michigan, 2003

Submitted to the Graduate Faculty of
The Joseph M. Katz Graduate School of Business in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Pittsburgh

2012

UNIVERSITY OF PITTSBURGH

This dissertation was presented

by

Eirini Ilana Diamanti

It was defended on

March 30, 2012

and approved by

Dissertation Advisor: Laurie J. Kirsch, Ph.D., Professor of Business Administration

Katz Graduate School of Business, University of Pittsburgh

Sandra Slaughter, Ph.D. Alton M. Costley Professor of IT Management

Georgia Institute of Technology

Dennis Galletta, Ph.D. Professor of Business Administration

Katz Graduate School of Business, University of Pittsburgh

Sherae Daniel, Ph.D. Associate Professor of Business Administration

Katz Graduate School of Business, University of Pittsburgh

Phyllis D. Coontz, Ph.D. Associate Professor, Sociologist

University of Pittsburgh

Copyright © by Eirini Ilana Diamanti

2012

THE EMERGENCE OF DOMINANT DESIGN(S) IN LARGE SCALE CYBER- INFRASTRUCTURE SYSTEMS

Eirini Ilana Diamanti, Ph.D.

University of Pittsburgh, 2012

Cyber-infrastructure systems are integrated large-scale IT systems designed with the goal of transforming scientific practice by enabling multi-disciplinary, cross-institutional collaboration. Their large scale and socio-technical complexity make design decisions for their underlying architecture practically irreversible. Drawing on three alternative theories of IT adoption (path dependence, project management, technology framing) and on a qualitative study of archival and interview data I examine how design and development influence the adoption trajectory of four competing cyber-infrastructure systems comprising the Global Environment for Network Innovations (www.geni.net) over a period of ten years (2001-2011).

Findings indicate that a) early design decisions, particularly those related to similar pre-existing systems set a path of adoption in motion leading to the early dominance of one system, b) coordination of milestones led to increased adoption for the high-performing teams, and c) the framing of technology presentations and demos as a social influence strategy was less effective in “breaking” the dominant system’s adoption path in the long term but enabled most of the development teams to challenge that dominance and increase the adoption of their systems in the short term. While studies in path dependence and dominant design assume that adoption and dominance occurs through users’ actions after development is completed, this study’s findings show that developers and managers of competing systems can also influence adoption and even “break” the dominant system’s adoption path while it’s still under development. Understanding

how cyber-infrastructure systems are developed is key to promoting their adoption and use. This research has import for understanding the ramifications of early-stage design decisions, as well as the impact of project coordination and technology presentation strategies such as framing for the adoption of such systems.

TABLE OF CONTENTS

<u>TABLE OF CONTENTS.....</u>	<u>VI</u>
<u>LIST OF TABLES.....</u>	<u>VII</u>
<u>LIST OF FIGURES.....</u>	<u>VIII</u>
<u>ACKNOWLEDGEMENTS.....</u>	<u>IX</u>
<u>1.0 INTRODUCTION.....</u>	<u>1</u>
<u>2.0 BACKGROUND.....</u>	<u>3</u>
<u>3.0 RESEARCH PROBLEM.....</u>	<u>6</u>
<u>4.0 LITERATURE REVIEW.....</u>	<u>8</u>
<u>5.0 SITE DESCRIPTION.....</u>	<u>34</u>
<u>6.0 METHODS.....</u>	<u>39</u>
<u>7.0 RESULTS.....</u>	<u>55</u>
<u>8.0 DISCUSSION.....</u>	<u>102</u>
<u>9.0 LIMITATIONS.....</u>	<u>116</u>
<u>10.0 CONCLUSION.....</u>	<u>122</u>
<u>BIBLIOGRAPHY.....</u>	<u>124</u>

LIST OF TABLES

TABLE 1. ADOPTION AND DIFFUSION IN TRADITIONAL VS. CYBER-INFRASTRUCTURE SYSTEMS.....	15
TABLE 2. DESCRIPTION OF GENI CLUSTERS.....	36
TABLE 3. CONSTRUCT DEFINITIONS AND MEASURES.....	45
TABLE 4. DATA OVERVIEW.....	49
TABLE 5. ADOPTION RATES OF EACH CLUSTER.....	57
TABLE 6. ADOPTION OF ALL CLUSTERS.....	58
TABLE 7. CLUSTER DOMINANCE IN PROJECTS.....	59
TABLE 8. CLUSTER DOMINANCE IN SITES.....	60
TABLE 9. GENI TIMELINE.....	69
TABLE 10. MECHANISMS DRIVING PATH DEPENDENCE.....	71
TABLE 11. PROJECT PERFORMANCE RESULTS.....	97
TABLE 12. TECHNICAL AND USE FRAMES FOR EACH CLUSTER	99
TABLE 13. THEORY FIT FOR EACH CLUSTER.....	101

LIST OF FIGURES

FIGURE 1. GENI AND ITS INITIAL 5 CLUSTERS.....35

FIGURE 2. PLANETLAB'S ADOPTION RATES.....61

FIGURE 3. PROTOGENI'S ADOPTION RATES.....62

FIGURE 4. ORCA'S ADOPTION RATES.....64

FIGURE 5. ORBIT'S ADOPTION RATES.....65

FIGURE 6. GENI'S DEVELOPMENT TIMELINE.....76

ACKNOWLEDGEMENTS

I would like to thank my dissertation committee for supporting me in completing this dissertation, especially my advisor Laurie Kirsch, PhD. The feedback I received from all committee members helped make this study more focused and organized.

Many thanks to the PhD Office, especially Carrie Woods, who provided guidance to the particulars of navigating through the doctoral program.

Finally, many thanks to family and friends who provided support throughout the Ph.D. program.

1.0 INTRODUCTION

The nature of large-scale research projects, the need to draw experts from multiple fields, and the need to produce innovative research outcomes have led to the formation of multi-disciplinary research groups that often span institutional, disciplinary and communities-of-practice boundaries. Large scale cyber-infrastructure (CI) systems or collaboratories are being deployed to support such research efforts (Olson 2008). Cyber-infrastructure has been defined as “infrastructure based upon distributed computer, information, and communication technology” (Atkins et. al 2003). There is also increased involvement on the part of industry experts and practitioners; university research centers are forming partnerships with the practitioner community while deploying an IS infrastructure to support joint research projects (e.g., Lohr 2007).

The deployment of cyber-infrastructure systems often constitutes an innovation in the research community in which they are introduced because such systems involve changes in the way that research activities have been traditionally structured in offline environments, and for that reason their value is less obvious to potential users. For example, contributing and reusing other researchers’ raw data, and making progress and evaluation reports accessible to all members with access to the system are practices that are not traditionally part of researchers’ work norms. The cyber-infrastructure makes them possible on a large scale but unless users can foresee and recognize the value of those practices, they will not be motivated to use an IS that

requires them to overcome the inertia of long held practices and change the way they conduct their work. Moreover, the value of a cyber-infrastructure lies not only in the data and technical support it can provide but also in the fact that there are sufficient users, a critical mass of adopters, to contribute data and knowledge resources. The innovative nature of such systems, the high costs of development and maintenance that they require, and the fact that their value depends on having a critical mass of contributors of data and other resources all make their adoption and diffusion challenging, both for individual scientists and their organizations.

The adoption and diffusion of technology has been examined in the IS literature, but those studies involved organizational or corporate systems whose users were subject to a common organizational culture, rewards structure and policies. Common organizational norms and culture affect whether the system is seen as valuable and beneficial by organizational members. Large scale cyberinfrastructures, however, span organizational boundaries and are not subject to a single set of work norms. Potential users in the community will tend to have different backgrounds and expectations about the system, as well as different priorities and goals associated with their being employed and affiliated with different organizations.

The present study is a comparative examination of four cyber-infrastructure systems and the processes underlying the emergence of a dominant technology. Chapters 2 and 3 provide some background on the development of cyber-infrastructure systems and a literature review with hypotheses about the emergence of a dominant design in such systems. Chapter 4 describes the site of this research, Chapter 5 describes the methods used to test the hypotheses, Chapter 6 discusses the results of the hypothesis testing, Chapter 7 discusses the implications of the results for researchers and practitioners and Chapter 8 is a conclusion of this study.

2.0 BACKGROUND

The development of CI systems tends to follow one of three high-level design paradigms. One of them is grid computing, which involves centrally organized and owned supercomputing clusters that are connected by dedicated, high-bandwidth networks. Grid systems support “*coordinated resource sharing among dynamic collections of individuals, institutions... and geographically distributed hardware, software, and information resources*” (Foster and Kesselman 2001). Grid-based systems are built on an underlying architecture that supports the participating labs’ or universities’ subsystems such as the Open Grid Services Architecture (Foster et al. 2002).

A competing paradigm is that of public-resource or peer-to-peer networks, with supercomputing resources being decentralized and distributed depending on the participating users’ capacity to support and share resources (Foster and Kesselman 1997). An example of peer-to-peer infrastructure is SETI@Home which is based on user-configurable applications that run in parallel and are managed in a decentralized fashion (Anderson et al. 2002). Those two paradigms for CI systems are considered to be irreconcilable; Grid-based middleware cannot be easily deployed for public-resource computing (Foster and Kesselman 1997). Grid computing is based on a symmetric relationship between participating users and institutions: participants can either share or use resources, whereas the peer-to-peer networks impose an asymmetry between providers of resources and those using them: participating users and institutions are required to share each others’ computing resources in order to be able to use the network’s resources.

Joining a peer-to-peer network is often at the discretion of individual users whereas joining a grid network is determined by the institutions supporting the research lab or project with which the users are affiliated.

A third paradigm is that of cloud computing which refers to service-oriented supercomputing resources that can be accessed and utilized over the Internet. The cloud computing infrastructure resides in large data centers and is usually managed by a third party (Jaeger et al 2008; Hand 2007). An example of cloud-computing infrastructure is the Google-IBM Academic Cloud Computing Initiative (Lohr 2007). Cloud infrastructures are designed around market-oriented incentives for participation and utilization of the cloud's resources (Buyya et al. 2008). Users, whether individuals or institutions, purchase or lease access to the cloud's computing resources for a limited period of time, and pricing is determined based on the intensity and scope of the computational resources that they are accessing. The organization of the resources under a cloud vs. a grid architecture still represent different models for data management (data is distributed in the cloud vs. distributed among remote and local academic sites in grid and peer-to-peer computing) and also for resource utilization (market-based or partnership-based incentives with cloud computing vs. academic consortia and community-based incentives with grid and peer-to-peer computing).

In addition to those three design paradigms that are primarily technical, the design space for CI systems is also characterized by different disciplinary paradigms depending on the scientific and computer science disciplines that develop and use the CI system. CI systems are developed largely by research scientists and reflect models and ideas that are prevalent in IT and computer science at the time that the CI is built, but might not be shared or valued by the domain scientists that are the intended users. An example is the tension between experimentation and

production-oriented systems. The logic of production systems emphasizes system reliability, stability, and efficient development methodologies. The logic of experimentation emphasizes research-oriented development, the use of experimental methods, and implementation of open-ended architectures. Experimentation takes precedence over concerns about stability, reliability and efficiency. Domain scientists that are the intended users of a given CI tend to prefer production-oriented designs whereas computer scientists and IT experts that are charged with developing the CI tend to value experimental designs.

3.0 RESEARCH PROBLEM

The various design paradigms for CI systems tend to be irreconcilable with each other, and switching between functionalities that represent different paradigms is impractical. The selection of a particular design for CI systems is more often than not too costly to reverse, both in terms of financial cost (total development cost for CIs range in millions of dollars) and in terms of the coordination costs associated with design decisions made by multiple experts and stakeholders. Even though peripheral functionality and software can be added onto the system at various points after it is built, the underlying architecture and design paradigm of the CI are not easily modifiable. Technical interoperability issues and social sources of conflict among stakeholders that might make them unwilling to share and reuse data and the tools and services embedded in a CI once it is built. The irreversible nature of CI systems increase the risk of decisions regarding their design, while the diversity of users, developers and organizations that have a stake in the CI system increase the social complexity of design decisions.

Practically, the challenge in the development of CI systems is the selection of a design that can serve multiple constituencies while assuming a particular form and structure that will invariably be too costly to replace. Theoretically, the research problem associated with this practical challenge is explicating the process by which a CI design is selected or emerges as a function of the social dynamics among stakeholders, predicting how the design selection process will unfold and what will likely drive the selection of a particular design. The goal of this study

is to examine the process by which a dominant design emerges for the architecture of a cyber-infrastructure system. To that end, the following research questions will be examined:

RQ1: *Why does a technology design achieve dominance over another?*

RQ2: *What are the behavioral factors that explain why a design dominates others?*

4.0 LITERATURE REVIEW

4.1.1 The development of cyber-infrastructure systems

Cyber-infrastructure projects are complex socio-technical projects focused on the development of integrated hardware and software for the purposes of supporting and transforming scientific research (Olson et al. 2008). Scientific collaboration – the sharing of tools, data and other resources that support knowledge generation - has become “essential to researchers” (Lucio-Arias and Leydesdorff 2009). Advanced computing infrastructures make possible the use of digital technologies that expand scientists’ research capabilities (e.g., use of visualization and simulation software) and are also changing the nature of the projects and research questions that scientists pursue. In the MIS literature, the field of cyber-infrastructure studies is relatively new, as such systems have only recently become common in areas such as the natural and health sciences, and less so in the social sciences. Below is a review of recent studies of CI development.

Studies of cyber-infrastructure development have followed two approaches. Some studies have focused on socio-technical issues such as system breakdowns, how individuals and institutions respond when the development efforts don’t go as planned, how scientists, developers and other stakeholders of a cyber-infrastructure system align their interests and priorities and coordinate the development effort. Such an approach, focusing on design and

development activities has been termed infrastructural inversion (Bowker 1994), because its focus is on those activities that constitute and support the creation and functioning of infrastructure. A second approach has been that of examining activities that the cyber-infrastructure supports, such as conducting scientific work, sharing data, collaborating remotely with other scientists, etc. This second approach differs in that it takes the development of the cyber-infrastructure system as a starting point, or as a given, and focuses instead on what happens after the system has been funded, designed, developed and adopted by scientists. Below is a review of studies that have followed each of those two approaches, followed by a description of the present study's approach.

Studies that follow the first approach, of infrastructural inversion, include Bietz et al. 2010; Kee and Browning 2010; Faniel and Jacobsen 2010; Karasti et al, 2010; Kirsch et al, 2009. Karasti et al. (2010) look at issues of conflict among developers and information managers during the development of a metadata language. Theirs is a study of two CI systems: the Long-Term Ecological Research Network (LTER Network) and the National Center for Ecological Analysis and Synthesis (NCEAS). LTER was designed to support comparative research across sites and ecosystems and among other related research programs, in the US and internationally. The NCEAS uses existing information to address important questions in ecology and related disciplines. Karasti et al (2010) conducted a longitudinal study and identified two temporal development orientations: project orientation and infrastructure orientation. For example, developers with a project orientation tend to focus on meeting of deadlines of short-term projects, while information managers with an infrastructure orientation aim to provide a reliable information environment and resources. The study shows how these two orientations,

“infrastructure time” and “continuing design”, can be useful in settings of long-term collaborative infrastructure development.

Kee and Browning (2010) explore funding issues in cyber-infrastructure development and trace their emergence at the institutional, individual, and ideological levels. Their study is based on the TeraGrid, an open scientific discovery infrastructure with resources at eleven partner sites that aims to create an integrated computational environment. Kee and Browning (2010) show how cyber-infrastructure development is supported by a range of institutions, individuals, and ideologies. Moreover, the study identified tensions in the development of TeraGrid, on the three levels of institutions, individuals, and ideologies. At the institutional level, the tensions include: funding either science or the development of technology; and juggling the priorities of NSF and local states, home universities, and federal agencies. At the individual level, the tensions are: providing either unrewarded service to the scientific community or building a tenure case; and spending time both on virtual organizations and a local supercomputer center. At the ideological level, there is one tension of developing the cyber-infrastructure system either for one’s theory/methodology or for one’s competitors’ theory/methodology.

Bietz et al. (2010) introduced the concept of synergizing to highlight the collaborative strategies in building and maintaining productive relationships among scientists, organizations, and technologies that are part of a certain cyber-infrastructure system. Their study is based on the Community Cyber-infrastructure for Advanced Marine Microbial ecology Research and Analysis (CAMERA), which serves the needs of the microbial ecology research community by creating a data repository and bioinformatics tools. The study identified embeddedness as a key characteristic of cyber-infrastructure development, in the sense that the cyber-infrastructure is

“sunk” or embedded into other structures, social arrangements, and technologies. Embeddedness is not only a result of developing cyber-infrastructure, but is also an important resource for development activities.

Kirsch et al. (2009) examined issues of conflict during the development of the Network for Earthquake Engineering Simulation (NEES) from the perspective of faultlines. Faultlines are hypothetical attributes (such as professional affiliation, gender, or expertise) that divide groups, and potentially trigger conflict within and between them (Lau and Murnighan 1998). NEES is comprised of 14 experimental facilities, collaborative tools, a centralized data repository, and earthquake simulation software. The study followed the development of NEES over the course of six years and found that conflict among developers, scientists and the funding agency was the result of faultlines that formed due to deep diversity issues between stakeholders, mistaken assumptions about technology capabilities and misperceptions of who is a stakeholder in the cyber-infrastructure’s development.

Studies that follow the second approach, taking the development of the CI system as a given and focusing instead on activities it supports, include de la Flor et al. (2010); Monteiro, 2010; and Faniel and Jacobsen (2010). More specifically, de la Flor et al. (2010) examined the Virtual Research Environment for the Study of Documents and Manuscripts (VRE-SDM), which was a pilot project designed to address the user needs of documentary, textual and manuscript scholars. The study focused on the function of CI tools and techniques in promoting the development of new forms of scientific practices and collaboration by embedding technologies within research institutes. De la Flor et al (2010) showed how a CI application can support scientific collaboration and emphasized how crucial it is for CI systems to adequately support large-scale collaboration among remotely distributed researchers.

Monteiro (2010) demonstrated how scientists working in a multidisciplinary team produce scientific evidence through the use of scientific visualizations. Advanced imaging technologies as a type of CI application were used by the scientists to produce computer models of heat transfer in living tissues. The study shows that scientists derive observations from digital objects, which they consider to be part of what is ‘empirical’ to their work. Scientific evidence is produced not only through observation of natural objects, but also through scientists’ use of and interaction with digital objects. The use of cyber-infrastructure systems is thus changing scientists’ notions of what counts as empirical evidence. Monteiro (2010) thus demonstrates the important function of CI systems in enabling and altering scientists’ work and in changing the meaning of CI tools and systems as a “third pillar of science” (in addition to theory and observation).

In another study of NEES (Network for Earthquake Engineering Simulation), Faniel and Jacobsen (2010) examined how earthquake engineering researchers reuse colleagues’ experimental scientific data. While scientists have traditionally assessed the reusability of small scale scientific data by looking at the data’s relevance, trustworthiness and intelligibility, the emergence of cyber-infrastructure systems has enabled the collection and reuse of large-scale data. That means that in order to assess the large-scale data’s reusability in the same way that the reusability of smaller-scale data is assessed, CI systems will need to provide specific technological capabilities to those producing the data and to those reusing it. For example, as Faniel and Jacobsen (2010) discuss, CI systems will need to include documentation tools, visual representations and contextual information that help scientists not only produce and document their own data but also assess and reuse their colleagues’ data.

As the above studies indicate, scientists rely on the ability of CI systems to support the production of scientific evidence (Monteiro, 2010), scientists' short-term and long-term information needs (Karasti et al. 2010), remote collaboration (de la Flor et al 2010), interdisciplinary collaboration and career advancement (Kirsch et al. 2009; Bietz et al. 2010; Kee and Browning 2010), and data reuse (Faniel and Jacobsen 2010). For those reasons, it is crucial to understand how the design, development, and deployment of CI systems unfolds with the goal of identifying successful practices and interventions that can ensure that the CI system meets the different needs of its stakeholders.

4.1.2 The Emergence of Dominant Design(s)

The literature on IT adoption, diffusion and innovation discusses the development of IT designs with the metaphor of “dominant design”. An IT design can be a technology, a standard, a format, or a design concept (Abernathy and Utterback 1978). A dominant design for a technology “emerges” out of a number of competing alternatives when it gains wide acceptance by developers, users, or consumers at the exclusion of its alternatives. Dominant designs, once they are established in an industry or field, can be extended and modified incrementally but cannot be readily substituted nor integrated with their alternatives. Further, dominant designs have been conceptualized as primarily technological innovations that are selected over their alternatives not because of their superior functionality but through processes of social influence and market selection (Utterback 1996). Some studies emphasize the path- dependent nature of the development and adoption of IT systems, and tend to assume that alternative designs are available in the market that can be evaluated by their potential adopters (Tushman and Murmann 1998).

Overall, the IT adoption and innovation literature has examined the emergence of “dominant designs” for systems that are relatively homogenous, stand-alone technologies, or easily unbundled into their complements. CI systems differ in that they are a collection of heterogenous technologies, which must be integrated and interoperate for the CI system to be considered as such, and cannot be easily unbundled into their components. Moreover, designs in the IT diffusion and innovation literature tend to be conceptualized as static, having a fixed set of functionality and features. CI designs can be static or dynamic, incomplete and evolving during the process of their “selection” by users. A dominant design emerges as a function of critical mass of adopters (Arthur 1989), however, that is tractable only after the technology is built and users have adopted or rejected it. Dominant CI designs on the other hand, tend to consist of abstract models, prototypes, and incomplete and evolving specifications. A CI design is likely to dominate users’ preferences before it is completely built. Users are less likely to have had experience with systems of similar reach (embedded in work practices) and scope (accommodating multiple scientific communities) and might not have the benefit of trial with a system when evaluating alternatives. The differences between the kinds of IT designs examined in the IT adoption and innovation literature and the characteristics of CI designs imply that the process by which a CI design becomes dominant over competing designs might be different from what the literature on non-CI designs suggests. Different factors might be influential at different points in time leading up to the emergence of a particular design.

Table 1 summarizes the differences in how adoption and diffusion of technology has been typically studied in (non-cyber-infrastructure) information systems development projects as opposed to cyber-infrastructure development projects.

Table 1. Adoption and Diffusion in Traditional vs. Cyber-infrastructure Systems

	IS Development Projects	Cyber-infrastructure Development Projects
Nature of IT adoption /diffusion	Individuals or organizations adopt a completely developed organizational system or end-user application.	Individuals or organizations form a consortium or a network of multiple research labs and adopt an incomplete system, shared by members of the consortium. The CI system is being developed as users join the consortium or research network.
Nature of the IT system	Complete, stand-alone software system or application, might interoperate with other systems.	Incomplete, complex combination of inter-related software and hardware, must interoperate with each other to be part of the CI system. CI system includes data center(s), large scale databases, equipment related to the users/scientists' work.
System users	Organizations in an industry and their employees, end-users / consumers of IT products.	Research scientists and their employing institutions, typically university departments and labs, from various disciplines.
Measures of adoption/diffusion	Number of users / buyers of a certain IT product, market share of an IT system compared to competing systems.	Number of institutions, research labs or departments joining the cyber-infrastructure consortium or network. Number of research projects using the CI system.

4.2 ANTECEDENTS TO THE EMERGENCE OF DOMINANT DESIGN(S)

There are three alternative explanations in the IS and organizational literatures that might explain how an IT system achieves dominance over competing alternatives. The goal of the present study is to examine the fit of each theory regarding the adoption / dominance of cyber-infrastructure systems. The study will contribute to the research field of cyber-infrastructure studies by identifying social processes driving the adoption /dominance of such systems in the presence of competing alternatives. The site of the present study is a network or program of four cyber-infrastructure systems, comprising GENI (www.geni.net). The four CI systems were funded by the NSF, and are competing for becoming the definitive design for the overall GENI system. Initial funding and design of all four CI systems began in 2007 and continues until the time of the study. Moreover, all four CI systems have a common management structure, with the same management team (the GENI Project Management Office) overseeing their development efforts. That common context and the competing nature of the four system's relationship makes GENI an appropriate site for a comparative study of multiple CI systems and their adoption trajectories over time. The three theories from the literature that might explain differences in their adoption and dominance are discussed in greater detail below.

4.2.1 Path Dependence

Path dependence theory stresses the importance of past events for future action. Current and future events, actions or decisions depend on the path of previous events actions, or decisions. Sydow et al (2009) define path dependence as a rigidified, potentially inefficient action pattern built up by the unintended consequences of former decisions and positive feedback processes.

That means that for an organization's history or development to be considered path-dependent, there must exist a sequence (or path) of events, decisions or other developments that indicate a narrowing or constriction of choices in that organization's development. Those choices can be decision alternatives, choices in the organization's management, or any other kind of options that are available to members of that organization at the beginning of its history and which over time constrict into a progressively narrower set of choices. Eventually a path emerges out of the narrow set of choices that links together key events, decisions or activities so that the organization's development or history can be considered path-dependent.

Studies of path dependence model path-dependent processes as adoption patterns: social actors adopt technologies, standards or make decisions, and some of this adoption paths end up dominating others, eventually resulting in lock-in. For example, the adoption of the QWERTY keyboard design is considered path-dependent (David 1985). The more users that adopted the QWERTY keyboard, the more likely it was that prospective users would also choose that keyboard because of its increased popularity. Also, from the user's perspective, getting accustomed to QWERTY keyboards made typing easier and quicker, so that the more skilled one became in QWERTY, the more utility they derived from it and the harder (costlier) it became for them to switch to another keyboard design. If the QWERTY design was the result of random selection among alternative keyboards made repeatedly over time and by an increasing number of users, despite not being the best design standard, then the history of keyboard standards can be considered path-dependent.

Different mechanisms have been proposed to explain the effects of path dependence on technology adoption: increasing returns, positive feedbacks, self-reinforcement, and lock-in (David 1985). Increasing returns imply that the more a choice is made or an action is taken, the

greater its benefits. For example, a technology's value increases as the number of adopters increases. With positive feedbacks, an action or choice creates positive externalities when that same choice is made by others. (In theory, positive feedbacks are similar to increasing returns, but they differ mathematically). Lock-in implies that one choice becomes better or more attractive because a sufficient number of people have already made that choice.

More specifically, the creation of a path starts with an initial triggering event: an event, decision or action. A decision or a choice that is made by individuals, organizations, or markets may amount to a small event that unintentionally sets off a self-reinforcing process of actions, choices or decisions that eventually lead to lock-in. If a process is path dependent, then lock-in will occur on one of the possible outcomes. Lock-in can happen on any path or choice, not necessarily on the optimal one, and is the result of self-reinforcing mechanisms, a set of mechanisms that increase the attractiveness of a choice relative to others (Vergne and Durand 2011). David (1985) proposed that path dependence involves self-reinforcing dynamics that are beyond the control of the individual actor and that eventually lead to an irreversible state of inflexibility or lock-in. David (1985) also notes that the logic behind technological path dependence is grounded in the fact that repeated decisions or choices (e.g., RD investments in a technology) create sunk costs, which result in irreversibilities along the path of technology development.

Early studies of path dependence (Arthur 1989, 1994) developed models of a formal economic theory of path dependence and proposed that increasing returns were the major mechanism behind its effects. According to Arthur's view, the process of becoming path dependent can be characterized by four general properties:

1. Non predictability: there is an indeterminacy of outcome.

2. Nonergodicity: several outcomes are possible (in economics that implies the existence of multiple equilibria), and history selects among the possible outcomes.

3. Inflexibility: the actors are locked-in or entrapped to an alternative, so a shift to another alternative is impossible.

4. Inefficiency: actions resulting from the path lock the market into an inferior outcome (e.g., an inferior technology, standard, or design).

In the organizational literature, path dependence has been used to explain the development of dynamic capabilities through repeated investment in the organization's resources (Helfat 1994). Helfat hypothesized that R&D investments are path dependent, and that capabilities emerge from a series of path-dependent learning experiences. However, path dependence was not directly operationalized, instead correlations in R&D investment in the past and present (the time of study) were used to demonstrate the presence of path dependence.

Sterman and Witemberg (1999) simulated the path-dependent emergence of scientific paradigms and found that it is driven by situational characteristics of the environment. They developed an agent-based model in which theories compete for resources and multiple positive feedback processes cause the self-reinforcing rise of a new theory. The study showed that the likelihood of any new theory (paradigm) to become dominant depends on the environment into which it is launched, which in turn depends on the history of the theories or paradigms preceding it. The authors explain this as the presence of path dependence for emerging theories. What determines the survival of new theories is the presence of positive feedback loops in the external environment in which they operate, not the inherent characteristics of the theories.

Karim and Mitchell (2000) proposed a path dependence explanation for whether firms participating in acquisitions change more than firms that do not undertake acquisitions. They examined two kinds of organizational change: one resulting from resource deepening and a second one resulting from resource extension. Resource deepening is the retention of product lines that overlap with current product lines. Resource extension involves retaining product lines that are distinct from a firm's current product lines. They argue that resource deepening represents path-dependent change, while resource extension represents path-breaking change. In that study, path dependence was distinguished from path-creation, but path dependence was not operationalized directly, rather the operational variables reflected resource deepening and extension.

Sydow et al (2009) propose a theoretical framework of *organizational* path dependence and identify characteristics of an organization's history (events or decisions or anything that is considered a key development) that would suggest the presence of path dependence. Those characteristics are: the persistence of a decision alternative, a choice or an option in an organization's development showing that members or stakeholders or decision-makers have been selecting the same option over time despite the presence of alternatives. Those can be technology or design alternatives, such as the four clusters in GENI, or more generally decision alternatives in other organizational domains (e.g., alternatives in financial decisions, in management, in policy aspects, etc.) depending on the study's focus.

Moreover, the persistence of a certain choice or option implies that a set of choices or options were available or accessible to that organization's decision-makers, and that the repeated selection of the same option could lead to potential inefficiency. Choices or decisions may lead to inefficiency when alternatives are ignored, not considered or examined as possible solutions,

such as alternative technologies, organizational policies, corporate strategies or management approaches. Sydow et al. (2009) also emphasize that decision-makers may not be aware that their choices can lead to potential inefficiencies at some point in the future, but rather, inefficiency is something that is detected retrospectively with the benefit of hindsight (otherwise decision-makers would not have chosen something that they knew would prove inefficient over time). Path dependence, then, may lead to decisions, or choices more generally, that are initially optimal or efficient, but might become inefficient when made repeatedly and without the consideration of alternatives.

In addition to the persistence of a choice and the presence of potential inefficiencies down the road, Sydow et al. (2009) suggest that a third characteristic in an organization's history that might indicate path dependence is a triggering event, or events, that set the decision-making process, or path, in motion. An example of a triggering event noted by Sydow et al is a corporate strategic decision, such as Matsushita's agreement with Hollywood studios regarding content delivery that gave Matsushita and its VHS design an initial advantage over Sony's Beta, and eventually led to VHS becoming the dominant standard (Cusumano et al. 1992).

Furthermore, another characteristic, or rather, necessary condition for path dependence according to Sydow et al. (2009), is the presence of self-reinforcing dynamics that lead to a narrower set of options or decision alternatives over time. Such dynamics can be group dynamics among key decision-makers, or any organizational or social mechanism that could motivate individuals or groups to make certain decisions and not others and thus 'reinforce' the selection of certain alternatives over time. Sydow et al (2009) identify four such social mechanisms that are common in organizational contexts: the first one is complementarity, which makes some

alternatives more attractive or efficient because of the presence of complementary options, such as complementary technologies (hardware and compatible software).

A second mechanism is learning effects, which refers to individuals or organizations selecting an alternative because they have developed relevant skills over time. A classic example of self-reinforcing learning effects is organizations' focus on exploitation-oriented strategies, developing strong capabilities or an advantage in that domain over time, but ultimately losing sight of exploratory strategies that could direct the organization into new markets (March 1991).

A third mechanism leading to path dependence is what Sydow et al (2009) term 'adaptive expectations', which refers to instances where an alternative is selected because decision-makers expect that others will also make the same choice in the future. An example of adaptive expectations is technology users purchasing a certain software or system because they expect it to become popular in the near future (this is similar to a self-fulfilling prophecy). In other words, decision-makers (or consumers or technology users) 'adapt' their expectations and choose not the most cost-effective or efficient alternative, but that which they expect to become the norm in the future.

A fourth mechanism leading to path dependence is coordination, resulting from rules, policies or practices that if adopted widely can make interactions among organizations and their members more efficient by reducing coordination costs. An example of coordination effects is the adoption of certain R&D practices in an organization which ensures efficient collaboration among departments and individuals but might make adaptation to new market conditions costly or prevent the organization from adjusting their practices and developing new competencies, as was the case with Polaroid (Tripsas and Gavetti 2000).

In summary, path dependence theory explains the high switching costs and lock-in that often characterize technology adoption as outcomes of the technology's history of development. Examining a CI system's history of development then can be informative for explaining its dominance over similar CI systems. CI systems will be more likely to dominate others not because of their technical superiority or usefulness but because they have attracted users that are locked-in to that system and would prefer using it rather than switching to a competitor system.

Proposition 1: The dominance of a cluster (cyber-infrastructure technology) is path-dependent.

Hypothesis 1: The adoption rate of a cluster (cyber-infrastructure technology) in a given spiral (period of development) depends on its adoption rate on the previous spiral.

4.2.2 Project Performance

The management of IS development projects is challenging; a recent Standish Group report indicates that only 32% of software development projects are completed on time and within budget, and meet their requirements (The Standish Group International 2009). The failure rate of IS projects is worse in the public sector (e.g., Collins and Bicknell 1997). Given the failure likelihood of development projects, project management techniques are often adopted as an attempt to safeguard against failure in new IS development projects (Slaughter et al. 2006; Harter and Slaughter 2003). Project management techniques increase a team's or an organization's project management maturity, and that, in turn, results in enhanced project performance (Kwak and Ibbs 2000).

Most project management approaches or techniques involve following a schedule of milestones (e.g., Jelinek and Schoonhoven 1990; Wheelwright and Clark 1992). Project performance is then monitored and tracked by examining a team's ability to meet milestones on time. Milestones are deadlines for project tasks or sets of tasks and are useful in that they help translate a project's high-level strategy into analyzable technical, budgetary, and time-related objectives (Jelinek and Schoonhoven 1990). Milestones help managers track the development team's progress and adjust project resources as needed (Wheelwright and Clark 1992). Milestones also serve as guides for developers; they keep development teams aware of the scarce resources available to them, and they offer a sense of structure while the team is experimenting with alternative designs (Wheelwright and Clark 1992).

Superior project performance, such as meeting milestones on times, is particularly important when developers are competing with other teams that are developing similar technologies. Superior project performance in terms of met milestones could enhance a system's competitiveness and/or dominance of similar systems by allowing developers and project managers to better allocate resources and to adjust the system's design to make it more competitive while the system is still under development. Project managers can also use information from milestones to make adjustments to the development schedule itself and to release a system in the marketplace sooner or with different features. That, in turn, can speed the development and ensure that the system reaches the market or becomes available to users at the same time as competing systems. When a development team is missing milestones, managers are less able to make adjustments to the teams' resources or to the development schedule, and as a result the technology might take longer to develop and reach the market. Project performance information then is a valuable indicator of whether a given system is likely to reach the market as

scheduled, and as such, whether it can successfully compete with similar systems that are released around the same time.

As mentioned earlier, a way in which enhanced project performance can influence a system's competitiveness and dominance of similar systems is by reducing the time-to-market. Studies of IS development efforts have shown that for certain kinds of technologies, such as collaboration systems, there is a window of opportunity in which a newly released system gains popularity and stands to become highly competitive relative to incumbent systems (Stalk 1988, Stalk and Hout 1990). Technologies that exhibit such characteristics are packaged software, collaboration systems, computer hardware, and consumer electronics (Krubasik 1988). The sooner a certain technology becomes available for adoption or prototyping, the greater the likelihood that its adoption will surpass the adoption of similar technologies that might be released later on, and the greater the likelihood that that technology will come to dominate others (Utterback 1994). Missing that window of opportunity, that is, releasing a new technology late in the market, may result in reduced user interest and eventually might mean that that technology has lost the opportunity to dominate others in the marketplace despite its value and usefulness. In other words, being early-to-market reduces the power of substitute technologies. Early adopters will have fewer substitute technologies to choose from, and the first technology that becomes available for sale or experimental use (e.g. beta versions) will likely gain a critical mass of users before alternative technologies enter the market.

A second way in which project performance may increase dominance is by reducing control and coordination costs (Kirsch 1996, 1997; Kraut and Streeter 1995). Control in IS development projects refers to management's attempts to influence developers to behave in accordance with team- or organizational goals (Kirsch 1996, 1997). Controlling IS projects

involves setting deadlines and milestones (outcome control) and also modeling expected or desired behaviors, such as expertise and knowledge sharing among developers (behavior control). The use of milestones represents a formal, outcome control mechanism and as mentioned earlier it stands to benefit both managers and developers by providing feedback about the development team's available resources and capabilities. Development teams that meet project milestones are then better able to adjust and/or request resources as needed, and to utilize their expertise and capabilities in an efficient as well as effective manner (e.g., Boehm 1986). Development teams that miss milestones on the other hand, are less likely to use the resources at their disposal in an efficient manner either because of lack of planning or because of the delayed development of key features that the missed milestones reflect. In other words, meeting milestones on time indicates that that team is exercising at least minimal outcome control and coordination in their development efforts, which in turn better positions them to finish the development early or on time and thus reach the marketplace earlier than competing systems. That, in turn, places them in an advantageous position relative to competitors and increases the likelihood that that technology will come to dominate competing systems.

In summary, control and coordination mechanisms, such as formal milestones allow the IS team to monitor their project performance, make adjustments in resources and goals as needed, and deliver the technology on time. IS projects that meet their milestones are more likely to be completed earlier than and thus reach the market earlier, which increases their chances of dominance over competing systems.

According to the project management perspective, a technology could achieve dominance over another (in terms of adoption) through superior project management. A dominant technology emerges out of a number of competing alternatives when it gains wide acceptance by

developers, users, or consumers at the exclusion of its alternatives. In cyber-infrastructure development projects, such as the four technology clusters that comprise GENI, a dominant technology can emerge as new projects join the GENI clusters and adopt the technologies developed by the clusters' teams. Because each project can only use one of the four alternative designs as the basis for its development, the PIs (project investigators) and their teams can only select or adopt one of the four GENI technologies. The GENI cluster that has attracted the most projects will dominate the other three alternatives. If project management has a positive influence on technology adoption and dominance, then we would expect that the cluster with the most projects will also have superior project performance.

Proposition 2: A dominant cyber-infrastructure technology is the outcome of superior project management of the technology's development project.

Hypothesis 2: The adoption rate of a cluster (cyber-infrastructure technology) in a given spiral is positively associated with the performance of that cluster's projects in the previous spiral.

4.2.3 Social Influence

A stream of research within the larger literature on technology adoption suggests that the process by which a design becomes dominant is a social influence process among developers, users and organizations that have a stake in the technology (Tushman and Murmann 1998;

Utterback 1996; Miller et al 1995). Social influence tactics and organizational politics can shape user preferences for some design alternatives and not others. The ability to shape user preferences gives one power to affect adoption decisions, by individuals, teams or organizations, which collectively can make a given design into a popular or a niche system. Dominant designs can thus emerge through a combination of social influence actions by developers, current users, and organizations that have a stake in a given design (Dagwell and Weber 1983; Miller et al 1995; Tushman and Rosenkopf 1992).

At the interpersonal level, social influence has been defined as intentional change, either psychological or behavioral, that results from the behavior of other people (Turner 1991). In the IS literature, social influence among peers, coworkers and other reference groups has been shown to affect technology adoption decisions (Venkatesh et al 2003). Assuming that dominant designs reflect adoption decisions, whether individual or collective, we would expect that a technology is likely to achieve dominance over another when its stakeholders (developers and current users) engage in social influence actions that attempt to promote that technology (Robey et al. 1989; Markus 1983; Venkatesh et al. 2003). Such social influence actions may involve presentations of the technology to various audiences of potential adopters, descriptions of its technical features and functionality, demonstrations of its possible uses and benefits, and discussions of its usability (e.g., highlighting its user-friendly aspects besides its technical specifications). Moreover, developers and current users of a system might attempt to influence prospective adopters by presenting the technology in various occasions and via different media, such as offline in live public demos, in conferences, workshops or meetings, as well as online in websites, wikis and other spaces that publish information about the technology's development.

However, the above IS studies, and subsequent research based on the technology adoption model have dealt with systems that are stand-alone, fully developed, and commercially available, rather than with systems that are interdependent or integrated with other systems, incomplete or under development, and not widely available in the marketplace (Iacono and Orlikowski 2000). CI systems, as mentioned earlier, consist of disparate component technologies that have been integrated to form an infrastructure, and, rather than being commercially available in a completed static form, are often adopted while still evolving and in an experimental stage of development. Social influence processes might unfold in a different manner between stakeholders of a CI system and its potential adopters than between stakeholders of a non-infrastructure system and its potential adopters. That is, developers and domain experts might attempt to promote a CI system employing different tactics than users of non-CI systems. CI developers and domain experts might rely more on peer relationships and networks of colleagues to promote a particular CI technology. They might rely on their professional reputation or academic status to influence newcomers in a field or those doubting the value of CI-enabled research, thus reducing legitimacy barriers to adoption. They might also attempt to influence users of other systems by demonstrating a CI's interoperability with other systems and by presenting its integrated features and functionality, thus reducing the switching costs for those users. Finally, they might frame a CI as a practically useful resource with short-term benefits by demonstrating its usability (e.g., its mobile add-ons, communication tools, data storage capacity, privacy protocols, etc.), thus reducing cognitive barriers to adoption.

Furthermore, social influence processes might not even be an individual-level phenomenon and instead involve collective action, such as teams of developers and domain experts collaborating to construct presentations and demonstrations of the various CI

technologies. Social influence processes might also involve cooperation among institutions that would otherwise be competitors in a domain, an academic discipline or a narrow research field but depend on the CI becoming a standard, or widely used by other experts, and have thus the motive to cooperate to influence how the CI is perceived by prospective adopters and funding agencies.

The literature on power and politics in IS development has addressed those issues in relation to the adoption of new systems (e.g., Markus 1983). If a new system affects the power balance among its users, such as giving less discretion and authority over scheduling and decision-making to managers or administrators and more so to their subordinates, then, according to that literature, managers, administrators and those with higher power will resist or reject the system despite its superior functionality (Robey et al. 1989; Markus 1983). The implication for studies of CI development projects is that alternative designs that redistribute decision-making authority or discretionary control over resources among stakeholders will be resisted by those who stand to lose any of those privileges once the CI is introduced into their work. Those stakeholders will also be less motivated to participate in design decisions in ways that move the development efforts forward; rather, they will likely attempt to obstruct or change the course of the development effort and the decision-making process to prevent losing power and status that they previously had. Power and status need not be only about IT resources. Because the introduction of CI is implicated in the way in which research outputs and the research process itself are evaluated, power and status concerns will be career-related such as academic promotions and the maintenance of a distinctive research identity for domain scientists. For IT specialists, power and status concerns can similarly involve career and research-related concerns rather than just concerns about the system's resources.

Taken together, these observations suggest that the examination of how CI designs are selected should take into account the behavior of multiple stakeholder groups and their interactions. More specifically, social influence theory would predict that certain kinds of social dynamics, such as communication between developers, managers and users, and competition among organizations and individual stakeholders, are critical to the process by which CI designs emerge.

4.2.4 Technology Frames

Technology Frames

Research in the field of social construction of technology has examined how shared information or knowledge, such as knowledge shared among groups of scientists, users, and manufacturers influences their understanding of a technology's functions and possible uses (Bijker et al. 1987). Multiple studies have shown that the developers' assumptions and knowledge influence decisions made about the system's design and deployment (Hirschheim and Klein 1989; Markus and Bjørn-Anderson 1987).

More specifically, Boland (1978) defined frames as a set of assumptions, understandings, and expectations that individuals have about a new information system (Boland 1978). A frame emerges through the interactions among people, technology and context; therefore, it's possible that different groups or individuals within the same context (e.g. an organization) have different frames about a system. Frames are used as a sense-making and problem-solving devices, and once constructed are hard to change.

Orlikowski and Gash (1994) examined how technological frames affect stakeholders' interpretations and actions related to IS development. They define frames as the knowledge

structures that individuals or groups use to make sense of a new technology, to solve problems related to the development or use of a new system, and more generally, "the assumptions, expectations, and knowledge they use to understand technology in organizations. This includes not only the nature and role of the technology itself, but the specific conditions, applications, and consequences of that technology in particular contexts" (p. 178). Orlikowski and Gash (1994) identified three kinds of frames that developers and users of a new technology (Lotus Notes) relied on to make sense of that technology: the nature of the technology (e.g., features), the organization's technology strategy, and technology-in-use (the uses and usability of the technology). In addition, Davidson (2002) developed a process model of how frames and changes in frames over time affect developers' sense-making during the requirements determination stage of ISD. Repeated changes in framing the requirements for a new system actually disrupted the participants' understanding of the technology and impeded their sense-making process.

In summary, research in both the IS development stream and in the stream of technology adoption has developed the concept of frames as an analytic tool for examining the expectations, meanings and visions that individuals and groups hold about a new technology. Studies in the dominant design literature have not specifically addressed the issue of frames and frame changes as an operative variable influencing adoption decisions. However, based on the larger body of research on frames and IS development we can hypothesize that social influence, and specifically the frames that the developers construct about their systems, will predict their system's dominance.

As mentioned earlier, Orlikowski and Gash (1994) have identified three types of frames that developers and users of new technologies rely on to make sense of those technologies:

technical frames that involve features of the technology, use frames that are about the use of new technologies, and their organization's technology strategy which is about the corporate IT strategy of their employing organizations. Based on the Orlikowski and Gash (1994) findings, and since there haven't been any newer studies to identify more or different types of frames, the hypothesis of the current study will concern the same types of frames: technical and use frames. Corporate strategy frames cannot be used in the current study since the GENI developers (and users/experimenters using the GENI technologies) are not employed by organizations that might have corporate IT strategies, rather, they are affiliated with research labs in various university departments, so their descriptions and presentations of their technologies would be unlikely to involve corporate strategy concerns. The technical and use frames, however, are relevant to all GENI developers since the features and the possible uses of the technologies they are working on are a primary concern for prospective users, for their project's funding agencies, and can affect a technology's adoption, competitiveness and therefore its dominance over other similar technologies.

Proposition 3: A dominant cluster (*cyber-infrastructure* technology) emerges through processes of social influence among its stakeholders.

Hypothesis 3: The number of (technical and usefulness) frames that developers (stakeholders) use in presenting their technologies during a Spiral will be positively associated with their cluster's adoption rates in the following Spiral.

5.0 SITE DESCRIPTION

The site of the present study is the Global Environment for Network Innovations (GENI) initiative funded by the National Science Foundation. GENI was designed for the purpose of enabling large-scale networking experimentation. Its goal is to create a cyber-infrastructure that would enable running experiments and tests to create innovations in networking. GENI's development began in 2007, and oversight of the overall project was granted to BBN Technologies who formed the GENI Project Management Office (GPO). Initially, GENI included five technology development clusters, however, one cluster was not as successful and was merged with one of the other four clusters early on in the development process. GENI is now comprised of four major technology clusters, which represent different networking technologies.

Each cluster is responsible for designing and developing systems based on different networking technologies (table 2). Each cluster includes multiple technology development projects and project teams, and can potentially support multiple research experiments and users from non-profit (academic) and for profit research institutions. Development teams receive funding by submitting proposals which are evaluated by the GPO. Research experimenters can adopt or use a cluster's technology and run experiments funded by NSF research grants.

GENI was selected for this research because it is a program of technology development projects; because all four clusters are supported by the same "umbrella organization" (GENI),

having a single project management office (the GPO) and the same funding agency (NSF), external variability is minimized in the comparison of the four clusters. At the same time, internal-to-GENI variability is maximized as the four clusters differ in the type of technologies under development, their history of development prior to GENI, the number of sub-projects they support and the number and kind of experiments and experimenters that are associated with each framework. All those internal factors vary along dimensions that allow us to compare the fit of each theory to a given framework’s characteristics and trajectory of development. Moreover, GENI was conceived as a collection of competing designs, so the four clusters are competing to become the final design for GENI; that makes them an appropriate research site for examining the processes by which cyber-infrastructure systems become dominant.

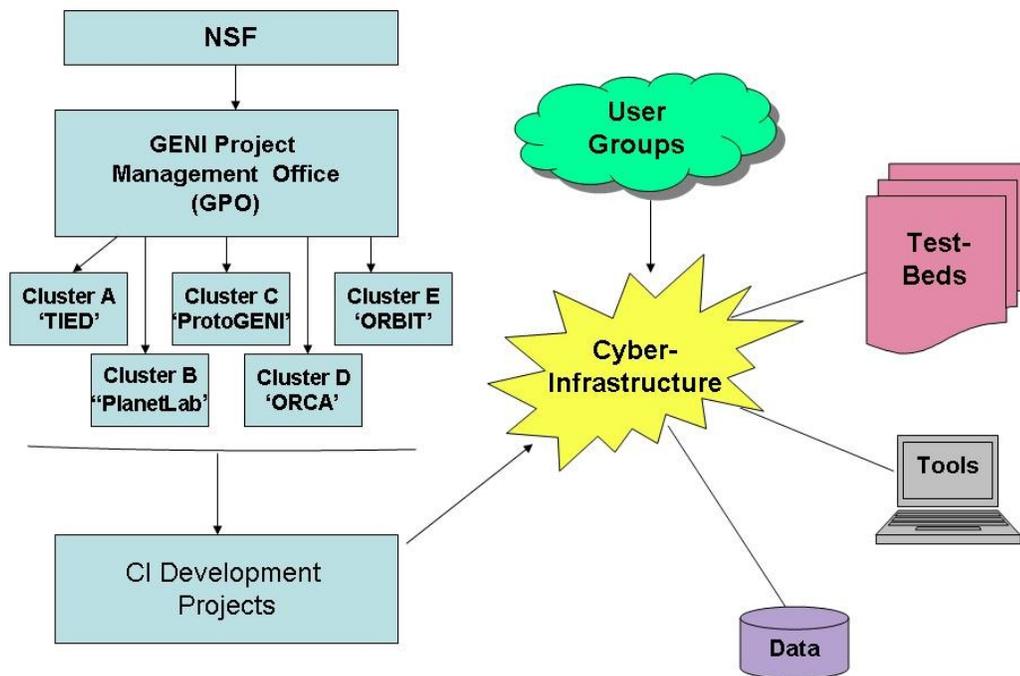


Figure 1. GENI and its initial 5 Clusters

Table 2. Description of GENI Clusters

Cluster Name	PlanetLab	ProtoGeni	ORCA	ORBIT	GENI-Wide
<i>Description</i>	Development of control framework reference implementations and integrating a range of component technologies and tools.	Network control and measurement, and integrated experiment environments.	Development of resource allocation strategies and integration of sensor networks.	Wireless networks development projects.	Projects that cut across clusters; e.g., Security, Operations.

5.1.1 GENI's Method of Development

All technology clusters in GENI follow the spiral development method. Spiral development is a systems development method used in information technology development projects. According to this method, a system goes through four stages of development before it can be considered completed: first, the user needs and requirements for the system are defined. Second, alternative designs are considered and a tentative design of the system is constructed. Third, a prototype is constructed out of the alternative designs. Prototypes are scaled-down approximations of the final system. Fourth, the first prototype is evaluated in terms of its strengths and weaknesses, and user requirements are again defined based on that prototype; then a second prototype is created and evaluated as to how well it meets the users' requirements. This process is repeated in cycles (spirals) until a system emerges that best matches the users' requirements. The spiral model is generally used in large, expensive and complicated IT development projects that involve shifting or hard to define user requirements (Boehm, 1986).

5.1.2 GENI Stakeholders

Each cluster in GENI has multiple IT projects associated with it. Stakeholders of a GENI cluster include: developers in the cluster's IT projects, the experimenters using the technologies developed in the projects, the sponsoring agency (NSF) and the project management office (GPO) that oversees the cluster, and prospective users and experimenters of the technologies developed in the projects. If a project initiates collaboration with another project in GENI, then those projects will have common stakeholders, the expanded group of developers, experimenters, sponsors, GPO and new collaborators and users. If a project or cluster initiates collaboration with a development team or another CI development project outside GENI, then, similarly, the group of stakeholders expands to include the new developers, users, experimenters and oversight and sponsoring committees of the two projects.

Stakeholder groups differ in terms of the interests they hold with respect to the technology. Developers are computer scientists who tend to value experimentation over system performance and reliability, which is something that users and experimenters value greatly. The developers' interests might differ from those of users and the project managers: computer scientists and experimenters (the domain scientists) in academia are interested in GENI's technologies as an opportunity to do research that will lead to tenure and/or other promotion in their respective fields. However, to achieve that goal, computer scientists are required to build new, experimental technologies that haven't been tried elsewhere, whereas domain scientists and other experimenters need to have a reliable, high-performing cyber-infrastructure on which to run new studies.

Those interests and values are often in conflict even when the two stakeholder groups have the same goal, using the CI system for research and ultimately gaining tenure and promotion through their research. Similarly, the values and interests of the project management office might conflict with those of the developers, as the former aims to keep the CI development project within schedule and budgetary constraints and the latter (the developers) strive for technical excellence but not necessarily superior project performance.

The interests of the sponsoring agency and those of users might also differ since the sponsoring agency funds CI projects that are supposed to last for an extended number of years and is interested in having users adopt those technologies and continue to use them over the course of their career, with the goal of making CI-enabled research in academia the norm or a standard. Users, on the other hand, while they may share the goal of using CI technologies for research purposes, they still might interpret the CI differently, as an opportunity to use cutting-edge technologies for only part of their research but not over an extended period of time, or might not be interested in using CI systems at all if that implies sharing their research data and making it widely available to other users through repositories and databases that are designed to last multiple years. The large-scale and long-term orientation of CI systems then, accounts for the diversity of interests and values that might exist among the system's stakeholders, and as the GENI's clusters develop different kinds of technologies, their stakeholder groups and their interests need to be taken into account when examining the development process and competition among the cluster's technologies.

6.0 METHODS

6.1 THIS STUDY'S APPROACH

This study follows the approach of infrastructural inversion (Bowker 1994) mentioned earlier, by focusing on design- and development-related events and activities that took place during the development of GENI. The main advantage of examining design-related events is that it enables us to look at how individuals and teams of developers think about design alternatives, to trace how one alternative becomes more popular or desirable than others, and to imagine how a CI system might have been designed or developed differently had the actions and decisions of its developers been different. In a sense, this approach looks at the CI's development from the inside out, "inverting" the process of development and exposing the antecedent events, actions and considerations that led to the CI system's eventual form, structure and organization. Some examples of development-related activities are: the collaboration among GENI's design and prototyping community in order to agree on key system specifications and interfaces; the meetings, workshops and other communication within members of each GENI cluster that address issues of development, project management, and promotion of the technologies developed by each cluster; the organization of GENI conferences and the presentations and other promotion efforts made by each cluster to generate interest in their technologies and attract more participants, experimenters and potential collaborators; and the project reviews and evaluations

of the development efforts of each cluster team by the project managers and the impact of those evaluations on the development trajectory of each cluster.

A qualitative study design was chosen to examine the technology development process over time for the four clusters that comprise GENI. Alternative methodologies, such as field experiments, simulation, and survey methodology are unsuitable because of the longitudinal nature of the study (Yin 1994; Galliers, 1992). A case study design attempts to describe relationships which exist in reality in a single or multiple organizations and allows for the examination of ill-defined variables which are common in real-world contexts. The goal of this research design is to use social behavior occurring in real-world context in order to understand a little known phenomenon such as the competition of multiple technology designs while they are still in early stages of development (Cavaye 1996).

6.2 CONSTRUCT DEFINITIONS AND MEASUREMENT

6.2.1 Dominance of a technology (cluster)

Dominance of a technology is the adoption, use or deployment of that technology (Abernathy and Utterback 1978). In the literature of technology diffusion a technology “emerges” as dominant out of a number of competing alternatives when it gains wide acceptance by developers, users, or consumers (Utterback 1996). In the present study, a technology becomes more/less dominant when it gets adopted or implemented by more/fewer sites and experimental projects. Dominance reflects an increase in the technology’s competitiveness over alternative

technologies. A technology can increase its dominance through its technical development, superior performance of its development project, or through its stakeholders' attempts to influence the terms of competition in favor of their cluster. A GENI cluster can be considered "dominant" over another if it has a greater number of projects or sites that have deployed its technologies. To determine whether a cluster has gained dominance over another, I used a measure of dominance-as-adoption, assessed at the cluster level (assessed at the end of each spiral from 2007 to 2011). Dominance as adoption has two indicators:

- Number of new projects in a cluster
- Number of new sites implementing components of a cluster or joining a cluster

To construct a measure of dominance for each cluster, I calculated the relative percentage of sites and projects (separately) that each cluster has gained at the end of each Spiral, relative to the ones it had at the beginning of that Spiral. That is the number of new sites and projects that have deployed the technologies of a cluster in a Spiral, divided by the number of sites and projects that the same cluster had at previous Spiral. This is a relative growth measure, indicating how much a cluster has 'grown' in dominance over the course of each Spiral relative to its dominance in the previous Spiral. Because it is a relative measure, Spiral 1 does not have any scores for this measure since it is the first phase of development for each cluster.

I also used a second calculation to construct a second measure of dominance: the absolute percentage of sites and projects (separately calculated) that each cluster has in each Spiral; that is, the number of sites and projects that have adopted the technologies of a cluster in each Spiral, divided by the total number of sites and projects that exist/were funded in that Spiral. This is a measure of dominance that is relative not to a cluster's earlier dominance level

but to the dominance of the rest three clusters in that same Spiral. These two indicators, the relative growth score and the absolute dominance score are two different measures of the same construct; they describe a different aspect of each cluster's dominance (dominance relative to the cluster's earlier dominance, and dominance relative to the other three clusters' dominance in the same Spiral).

6.2.2 Path Dependence

Sydow et al (2009) define path dependence as “a rigidified, potentially inefficient action pattern built up by the unintended consequences of former decisions and positive feedback processes”. Path dependence was assessed by examining each technology's history such as its stage of development when GENI was funded and at the start of spiral one, the early experiments and projects using that technology prior to GENI's funding, and the linkages among events, actions or decisions that led to experimenters, developers or project managers to select or prioritize one technology over the others over time.

6.2.3 Project Performance

The project performance factors that are most relevant to cyber-infrastructure development projects are the technology development schedule and deadlines, such as meeting milestones on time. Because the cyber-infrastructure technologies that are the focus of this study (ProtoGeni, PlanetLab, ORCA, ORBIT) are not complete systems at the time of the study, this study will focus only on the progress performance aspect which is reflected in the cluster's developer teams' met or unmet milestones. The development schedules and milestones are set by

the project office in all clusters, so there is a common management structure and a common set of formal coordination mechanisms (quarterly project reports, and quarterly working group meetings that discuss the milestones in the GENI engineering conferences). All four development teams have been using the method of tracking their progress through project milestones, which they assess as being late, early, or on time.

Project performance reflects how advanced a cluster or project in GENI is in its development of various cyber-infrastructure technologies, and is indicated by project management metrics such as milestones and development problems and the speed and effectiveness with which the development team addresses them. First, a project performance score was constructed at the project-level by adding the number of on-time and early milestones its developers had reached, and then subtracting the number of its late milestones. A score for project performance was then constructed for each GENI cluster by aggregating (adding) the project performance scores of each cluster's projects.

6.2.4 Social Influence and Framing

To determine whether developers and stakeholders engage in competitive moves to promote the technology I examined data from presentations at the GENI Engineering Conferences (GECs), the email communication in the development's archives, and the clusters' websites. Social influence at the interpersonal level has been traditionally defined as intentional change, either psychological or behavioral, that results from the behavior of other people (Turner 1991). Social influence was assessed through the frames that the GENI developers used in presentations and descriptions of their technologies. As mentioned earlier, frames have been defined as "the assumptions, expectations, and knowledge they use to understand technology in

organizations” (Orlikowski and Gash 1994). Frames are expressed through discourse; that includes verbal descriptions, written materials such as reports and presentations, visual images, and stories that individuals or groups construct or employ to describe or make sense of a technology (Orlikowski and Gash 1994; Moch and Fields 1985). This discourse may involve multiple media and may take place in various contexts (e.g., group meetings, organizational reports). Data on frames, then, should be collected from multiple sources and materials that include descriptions or presentations of the technology of interest.

For each cluster, two kinds of frames were identified, technical and usefulness frames based on Orlikowski and Gash (1994). The sums of a cluster’s technical and usefulness frames were calculated separately for each Spiral, and then the total sum of both kinds of frames was calculated. Greater number of frames indicated higher levels of social influence of a given cluster.

Table 3. Construct Definitions and Measures

Construct/Theory	Definition	Measurement	Data Source
Organizational Path Dependence	A “potentially inefficient action pattern built up by the unintended consequences of former decisions and positive feedback processes” (Sydow et al, 2009).	Identification of a sequence or pattern of decisions or actions that are reinforced by certain social mechanisms.	Interviews with GENI participants, developers’ mailing list archive.
Path Dependence: Self-Reinforcing Mechanisms	Social mechanisms/processes that drive path dependence.	Identification of a number of self-reinforcing mechanisms (coordination, complementarity, learning, adaptive expectations).	Interviews with GENI participants, developers’ mailing list archive.
Path-Dependence: Inefficiencies	Potential or actual inefficiency of an individual or organizational decision or action resulting from path-dependence.	Identification of decisions or actions that could be potentially inefficient (or have been actually inefficient).	Interviews with GENI participants, developers’ mailing list archive.
Project Performance	Average Number of milestones reached in that cluster.		Project reports and project retrospectives documents in clusters’ wiki pages.
Social Influence	Social influence attempts on the part of a cluster’s stakeholders to influence the development process.	Frequency of frames used to present a cluster’s technologies: how many times each frame (technical or use) appears in a cluster’s presentations, demos, reports.	Cluster and project presentations in the GENI wiki , project websites, developer listserv, observation data on GEC conference demos.

Social Influence – Frames	A frame is the way a cluster’s technology is presented and/or described by its stakeholders.	Identify references made to the cluster’s technology in reports, presentations and demos. Code the technology’s description with what kind of frame is used. A frame can be ‘derived’ by summarizing an object’s (e.g., the technology’s) description until a single tag/label emerges that describes the content in a few (one-two) words. Aggregate measure at cluster level by adding the number of technical and use frames in a cluster’s projects.	Cluster and project presentations in the GENI wiki, project websites, developer listserv archive, observation data on GEC conference demos.
---------------------------	--	--	---

6.3 DATA COLLECTION

The research design is a case study design (Yin 1994) in which a GENI cluster is the unit of analysis. Data was collected from archives of communication among GENI stakeholders, such as email listservs, meeting notes, progress reports, and other project documents from each GENI cluster. In addition, interview data with twenty GENI members were used to supplement the archival data. Interviewees were GENI developers, project managers and members of the funding agency that supports GENI. The archival data was also supplemented by non-participant observation from GENI demos and presentations that take place at the GENI Engineering

Conferences (GECs). That allows for triangulation of the data, as suggested by Eisenhardt (1989).

The focus of the study is the first three spirals of development, which began in 2007. The data collection followed the start and end dates of each spiral (2007-2011). The start and end dates of spirals are usually indicated in the meeting notes, progress reports, and are also reflected in the milestone reports. For example, the beginning of the second spiral is indicated by the start of integration development (integrating the main technology of a cluster with other projects across clusters). From that it can be inferred that the end date of the first spiral for that cluster is the last instance of recorded communication (or announcement in GEC archives) prior to the beginning of the integration development efforts.

The archival data is available online through the GENI wiki (www.geni.net/wiki) and the developer listservs allow anybody with online access to register and access the archives of past communication. The main source of data is the GENI wiki. Each cluster has its own wiki, which includes documents such as project reports, project data (milestones, deadlines), technical notes such as reported errors in the system and whether they have been addressed, meeting minutes of the cluster's working groups, access to the developers' listservs, slide presentations from the GENI conferences (GECs). A second source of data is the project websites; not all projects in a cluster have websites, but those that do include information about the technology's history of development that often predates GENI, and other information related to the project's history.

The path dependence hypothesis was tested based on interview data and data from the GENI developers' mailing list. Interviews were conducted during Spiral Three (2010-2011) and respondents were twenty in total and included GENI developers, project managers and members of GENI's funding agency. The developers' mailing list data is archived in the GENI wiki.

Data about project milestones were collected from the same online archive, the GENI wiki (www.geni.net/wiki). IS development projects in all clusters make available their quarterly project reports, project review documents, and track their milestones on their webpages in the GENI wiki. All those three data sources were used to gather information about each project team's met or unmet milestones in spirals one and two. Each milestone was classified as being early, late or on time. Data about project milestones were aggregated at the cluster level by summing all milestones in a cluster's projects. Those scores were then converted to percentages to make them comparable across clusters as the four clusters did not have the same number of scheduled milestones throughout the two spirals. That data were then aggregated at the level of spiral, so that each cluster has three milestone indicators in each spiral (% of early milestones, % of late milestones, % of on-time milestones). A single score of project performance was computed by adding a cluster's number of on-time milestones to its early milestones and then subtracting its late milestones.

Data on frames was collected from materials such as posters and slides presentations that the clusters' developers gave at the GENI conferences, descriptions of the technologies in the clusters' websites and wikis and in the email discussions they had on the developer listserv. The majority of the frames data came from the GENI conference presentations (more than 80% of distinct frames); the project websites and wikis had a limited number of technology descriptions. In total, approximately 280 instances of technology descriptions were identified that could be considered as frames. Some of those were duplicated across the GENI conferences, so they were counted only once (only the first instance of the presentations or description). Table 4 describes each data type and the volume of data that was used in the assessment and measurement of each construct and theory.

Table 4. Data Overview

Theory	Data Type	Data Volume
Path Dependence	Interviews	20 respondents (Project PIs, managers, developers, experimenters)
	GENI wiki	Project wikis within the larger GENI wiki
	Developer listservs	4 listservs - one per cluster, ~100 emails per year (2 each week on average)
	Project Websites	29 project websites
Social Influence, Frames, Project Management	GEC project presentations and demos	11 conferences in 2007-2011, ~30 slide and poster presentations and demos in each
	Meeting Notes	3-4 cluster team meetings per GEC conference
	Project reports	4 project reports annually per project in each cluster (Total ~50 across all clusters)

6.4 DATA CODING

The collected data was coded following Strauss and Corbin's (1990) open, selective coding procedure. Only the qualitative part of the data was coded, that is, the data that was used to test the path dependence and the social influence hypotheses (the project performance and dominance data are quantitative). First, the relevant constructs from the literature were coded in the raw data and then a range of values were identified for each construct which allows comparing the GENI clusters along those constructs. The coding procedure for each construct and theory is described in greater detail below.

6.4.1 Path Dependence

Path dependence theory was assessed with data regarding development-related incidents. Data sources include: the GENI wiki and each cluster's wiki pages (in geni.net), quarterly project reports that each cluster and its projects submitted to the project management office (the GPO), interview data with some of GENI's project managers, experimenters and developers, the developers' mailing list, meeting notes from project team meetings (that are uploaded by developers in the projects' wiki pages), and project websites.

The coding of the data proceeded as follows:

- 1) Key incidents to the development of the four clusters were identified in archival documents that described, reported or discussed the design and development of projects in each cluster. Incidents were coded as events, decisions, or activities, based on the coding scheme and suggestions in the studies of Langley and Truax (1994), and Langley (1990). More specifically, Langley (1990) discusses several approaches for analyzing

qualitative data; the coding of events, activities and decisions is proposed as an approach that's appropriate for the purpose of creating a map (or timeline) of key incidents in the history of an organization. This "visual mapping" approach helps researchers compare the evolution or development of an organization's history and helps identify, and present, linkages among key incidents.

- 2) A timeline of GENI-wide events, decisions, activities was constructed.
- 3) The events, decisions, and activities in GENI's timeline were coded as to whether they involved or represented one of the three characteristics that according to Sydow et al (2009) indicate path dependence (persistence, potential inefficiency, triggering event(s).
- 4) The events, decisions and activities in the GENI timeline were coded according to whether they referred to any of the self-reinforcing mechanisms (proposed by Sydow et al 2009).

6.4.2 Social Influence and Frames

As mentioned earlier, the data sources for social influence and the technology frames were the GENI conference presentations and demos, and the project websites. Frames were identified in each document (slide presentation, demo poster, website description of a project) by first identifying descriptions of the cluster technologies. Those are chunks of text (a sentence or a few sentences, which might include images) in a slide, poster or in an online document. In theory, a technology can be 'framed' in various ways when it is presented or described, so each document (slide presentation, website, poster) may include multiple descriptions of the same technology and thus multiple frames. Each description of a technology was coded or labeled with a frame.

A frame can:

1. describe the technology's features (the technology's development might have been completed)
2. describe the technology's purpose and goals if its development has not been completed
3. describe the technology's technical advantages over other systems
4. describe the technology as a proposed solution to a technical problem
5. describe the technology as a tool that might be useful to experimenters or researchers or students
6. describe the technology as a proposed solution to an education or research problem
7. describe possible uses of the technology
8. describe past impacts that the technology had on certain groups of users

As the above list indicates, two kinds of technology descriptions emerged from the data. While the first four kinds of descriptions are purely technical frames, the last four frames are about the technology's use, usability or usefulness (past or future). Moreover, the same technology (or cluster) may be described in both technical and use terms, either in the same presentation (e.g., in the same poster) or in different documents at different times. Below are some examples of each kind of frame:

- Technical frame (description of the technology's features):

“This demo shows several aspects of GpENI: -the basic operational configuration and monitoring capabilities of the GpGENI global infrastructure, -PlanetLab-based slice creation: the demo allows users to create an arbitrary overlay loop, shows and overlay topology using the google maps GUI and measures the latency of probes sent along the loop, - Interconnection with MAX demonstrating the transfer of LHC scientific data from the Prairiefire supercomputer at UNL to a PlanetLab node at MAX using ION” (GpENI project, GEC9 conference presentation).

- Technical frame (the technology is described as a solution to a problem):

“Developers have long lamented the programming interface provided by the network. It is clear that low-level sockets are not the ideal way to communicate and reason about execution on remote machines. Seattle makes developing networked applications easy and makes the power of cloud computing accessible to all developers” (Million Node GENI project, GEC9 conference presentation)

- Usefulness frame (description of some of the technology’s benefits and uses)

“Seattle [system] is ideal for students, researchers, and companies that want to prototype and test code on testbeds that have varying scale, diversity, and topologies. The same code may easily be run on a variety of operating systems, architectures, and network environments in order to understand the performance and dynamics of a distributed system” (Million Node GENI project, GEC9 conference presentation)

- Usefulness frame (technology is described as a tool useful for researchers)

“Cron is a cyber-infrastructure of reconfigurable optical networks environment that can provide multiple virtual networking testbeds consisting of routers, delay links and high end workstation operating up to 1gigs per sec bandwidth 1x faster). Different application developers and networking researchers can use those virtually created high speed networking and computing environments WITHOUT technical knowledge of network hardware and softwares” (CRON project, GEC9).

6.5 DATA ANALYSIS

Each of the three proposed theories was examined as a potential explanation for variation in the clusters' dominance scores. The unit of analysis is the cluster. For each cluster, a matrix was constructed to organize the observations regarding the main constructs and theories. Construct validity and reliability were established by relying on multiple data sources for each construct and theory (Yin 1989). Construct validity can also be established by demonstrating the “chain of events” that are used as evidence for a theory (Yin 1989), such as the timelines showing the clusters' development history. Each cluster ultimately has values of low/med/high in each theory, indicating how well that theory explains that cluster's development. The theory that best describes the most clusters and their development will be the best fit for the data.

Since the data collection begins at the project level, with each cluster having multiple projects, this makes the study an “embedded” case study (Yin 1989). Data collected at a lower level, such as the project level, had to be aggregated at a higher level, that of a cluster, in order for the theories to be comparable. The data analysis for each theory is described in the next section.

7.0 RESULTS

The results for each hypothesis are described in detail below.

7.1 DOMINANCE

Tables 5 and 6 below describe each cluster's dominance (or adoption rates), in relative and absolute scores, assessed in terms of sites and projects that have adopted the technologies of that cluster. Spiral One has only absolute indicators of dominance /adoption because there is no prior period of development for calculating growth of a cluster's adoption rate over time. In Spiral Three no new GENI projects were funded, so there are no indicators for dominance in terms of new projects for that period. Tables 5 and 6 present the dominance/adoption scores for each cluster in each Spiral (percent scores are in table 5, and numerical scores are in table 6).

The relative growth measure indicates how much a cluster has 'grown' in dominance relative to its dominance level in the previous Spiral (or how much its adoption has increased over the same period). For example, PlanetLab has grown in dominance 93.3% in terms of new sites that deployed its technologies during Spiral 2, and 71.4% in terms of new projects that were funded/ started in Spiral 2 that adopted its technologies.

The absolute score indicates what percentage of GENI's projects or sites have adopted a particular cluster's technologies at any given time. For example, PlanetLab had a 28%

dominance over the other clusters in Spiral 1, calculated in terms of the number of sites that have adopted PlanetLab's technologies (ie., 28% of all GENI sites in Spiral 1 have adopted PlanetLab's technologies). Similarly, 30.4% of all GENI projects in Spiral 1 have adopted PlanetLab's technologies.

The cluster with the greater adoption rates during Spiral One that can be considered "dominant" for that period is ProtoGeni in terms of projects (35%), and ORBIT in terms of sites (31%). These are absolute scores of adoption and as such they indicate each cluster's share of GENI projects or sites that have adopted its technologies (e.g., 35% of GENI projects during Spiral One have adopted ProtoGeni's technologies, while 31% of GENI research sites during Spiral One have adopted ORBIT technologies). That way no single cluster emerges as completely dominant of both projects and sites during Spiral One.

In Spiral Two ProtoGeni emerges as the dominant cluster in terms of both projects and sites with the exception of relative growth in projects: ORCA has the greatest growth of adoption rates in projects (140%) while ProtoGeni has 137% growth in projects. Both clusters had a much larger number of projects adopting their technologies in Spiral Two relative to the number of projects they had in the previous Spiral. Similarly Two ProtoGeni dominates the other clusters in terms of absolute scores of adoption by sites and projects; that is, the largest share of GENI projects and sites adopted ProtoGeni's technologies during Spiral Two.

No new projects were funded in Spiral Three, so the four clusters can be compared only by the number of research sites that adopted their technologies. PlanetLab emerges as the dominant cluster in terms of both relative and absolute indicators of adoption. That is, PlanetLab had the greatest growth in number of sites that adopted its technologies in Spiral Three compared to the sites that were using its technologies in the previous Spiral. Similarly, PlanetLab had also

the largest share of GENI research sites deploying its technologies compared to the number of sites that chose different clusters.

Table 5. Adoption Rates of each Cluster

	Spiral 1		Spiral 2				Spiral 3	
	Absolute score – Adoption by Sites	Absolute score – Adoption by Projects	Relative Growth in Sites	Relative Growth In Projects	Absolute score – Adoption by Sites	Absolute score – Adoption by Projects	Relative Growth in Sites	Absolute score – Adoption by Sites
PlanetLab	28%	30.4%	93.3%	71.4%	28%	25.5%	193%	42%
ProtoGeni	20.5%	35%	209%	137%	33%	40.5%	27%	21%
ORCA	20.5%	22%	118%	140%	23%	25.5%	142%	28.6%
ORBIT	31%	13%	0%	33%	16%	8.5%	0%	8.4%

Table 6. Adoption of all Clusters

	PlanetLab	ProtoGeni	ORCA	ORBIT	Total
Spiral 1 Projects	7	8	5	3	23
New Projects- Spiral 2	5	11	7	1	24
Total Proj. Spiral 2	12	19	12	4	47
Sites Spiral 1	15	11	11	17	54
New Sites Spiral 2	14	23	13	0	50
Total Sites Spiral 2	29	34	24	17	105
New Sites Spiral 3	56	9	34	0	99
Total Sites Spiral 3	85	43	58	17	203

Table 7 below lists all clusters in order of dominance, from highest to lowest, for each Spiral. For example, in terms of projects, the most dominant cluster in Spiral 1 was ProtoGeni, followed by PlanetLab. In Spiral 2 the most dominance cluster was ORCA in relative growth terms, and ProtoGeni in absolute terms.

Table 7. Cluster Dominance in Projects

	Spiral 1 - Projects		Spiral 2 - Projects	
	<i>Relative</i>	<i>Absolute</i>	<i>Relative</i>	<i>Absolute</i>
	<i>Growth</i>	<i>Dominance</i>	<i>Growth</i>	<i>Dominance</i>
Highest	--	ProtoGeni	ORCA	ProtoGeni
	--	PlanetLab	ProtoGeni	PlanetLab, ORCA (tie)
	--	ORCA	PlanetLab	ORBIT
Lowest	--	ORBIT	ORBIT	--

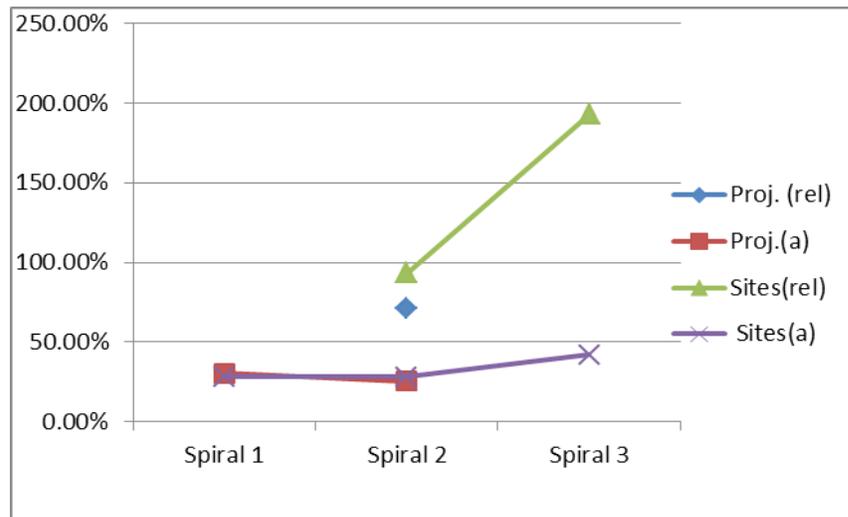
In terms of research sites that deployed cluster technologies (table 8), most sites in Spiral One adopted ORBIT’s technologies, making that cluster dominant for Spiral One. In Spiral Two, ProtoGeni emerges as the dominant cluster, both in absolute and relative terms. In Spiral Three, PlanetLab emerges as the dominant cluster, followed by ORCA, both in absolute and relative terms.

Table 8. **Cluster Dominance in Sites.**

	Spiral 1 - Sites		Spiral 2 - Sites		Spiral 3 - Sites	
	<i>Relative</i>	<i>Absolute</i>	<i>Relative</i>	<i>Absolute</i>	<i>Relative</i>	<i>Absolute</i>
	<i>Growth</i>	<i>Dominance</i>	<i>Growth</i>	<i>Dominance</i>	<i>Growth</i>	<i>Dominance</i>
Highest	--	ORBIT	ProtoGeni	ProtoGeni	PlanetLab	PlanetLab
	--	ProtoGeni, ORCA (tie)	ORCA	PlanetLab	ORCA	ORCA
	--	PlanetLab	PlanetLab	ORCA	ProtoGeni	ProtoGeni
Lowest	--		ORBIT	ORBIT	ORBIT	ORBIT

The graphs below show the adoption trajectories of each cluster, for projects and sites that adopted that cluster, both in absolute and relative terms. The graphs were created by plotting the dominance results (percent of adoption) of each cluster during the three Spirals. Each cluster’s adoption trajectory can be seen in terms of projects and research sites that deployed its technologies starting in Spiral One up until Spiral Two.

Figure 2. PlanetLab's Adoption Rates



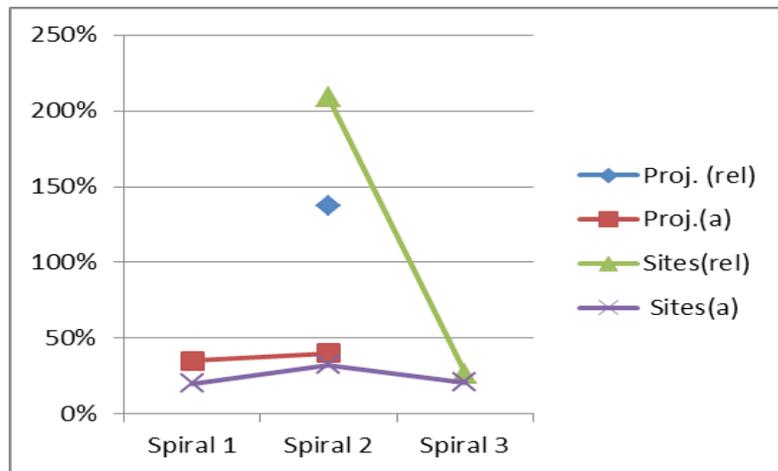
PlanetLab: The above graph shows PlanetLab's adoption rates in projects and sites in absolute and relative scores. PlanetLab is the cluster that has the highest level of dominance (or the greatest amount of growth in sites in both Spiral 2 and Spiral 3. More specifically, in terms of relative growth in sites (new sites deploying PlanetLab technologies), PlanetLab has a 93% growth from Spiral 1 to Spiral 2, and also the same amount of growth (93%) in sites from Spiral 2 to Spiral 3. That is the highest level of growth among the four GENI clusters in both periods.

In terms of absolute scores of dominance, PlanetLab starts out with the highest number of sites (15 in Spiral 1), which is 28% dominance over the rest of the clusters in terms of new sites deploying PlanetLab technologies. In Spiral 2, it remains at that level of dominance, 28%, over the other clusters in terms of new sites, and in Spiral 3 it increases to 42% dominance, which is still the highest level compared to the other tree clusters.

In terms of experimental projects, PlanetLab starts out with 7 projects in Spiral 1 and gains 5 new projects in Spiral 2, which reflects a 71% growth in dominance. In absolute terms,

that is a 30% dominance in Spiral 1, and PlanetLab is second to ProtoGeni which has 35% dominance. In Spiral 2, and in terms of new projects joining the cluster, PlanetLab decreases in dominance from 30 % to 25% (that is in absolute terms). In Spiral 3 none of the four GENI clusters had any new projects so there is no data in that aspect of dominance. Overall, PlanetLab has an increasing trend in dominance, measured by number of sites deploying its technologies, and also has the highest score in dominance of sites relative to the other three clusters except for the fact that ProtoGeni started out with more sites than PlanetLab at the beginning of Spiral 1.

Figure 3. ProtoGeni’s Adoption Rates

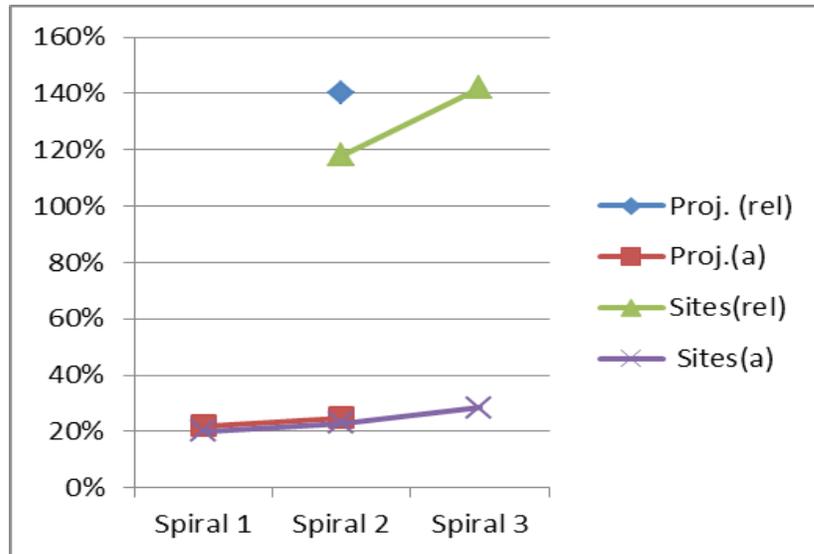


ProtoGeni: The above graph shows ProtoGeni’s dominance in projects and sites in absolute and relative scores. ProtoGeni starts out with 11 sites in Spiral 1 and gains 23 new ones in Spiral 2, which reflects a 209% growth. In terms of relative scores, that is the highest level of growth in sites among the four clusters in Spiral 2. In absolute scores, ProtoGeni starts out with 20% dominance in sites over the other three clusters in Spiral 1, increases to 33% dominance in

Spiral 2, but drops to 21% in Spiral 3. Both in relative and absolute scores then, ProtoGeni has the highest level of dominance in Spiral 2 among the four clusters.

In terms of new projects joining the GENI clusters, ProtoGeni starts out with an absolute score of 35% of dominance, and that increases to 40.4% in Spiral 2. In both Spirals, those are the highest absolute scores of dominance among the four clusters. In relative terms, however, the new projects that join ProtoGeni in Spiral 2 reflect a 137% growth, which is the second highest score (after ORCA's 140% relative growth in projects in Spiral 2). In Spiral 3, 9 new sites deploy ProtoGeni technologies, which is a 27% growth from Spiral 2. That is lower than both ORCA's and PlanetLab's relative growth in sites in Spiral 3. Similarly, in absolute terms, ProtoGeni also has a lower dominance score in Spiral 3, 21%, than ORCA and PlanetLab. Overall, ProtoGeni seems to have a decreasing trend in dominance of both sites and projects in Spiral 3 despite its highest scores in Spiral 2.

Figure 4. ORCA's Adoption Rates

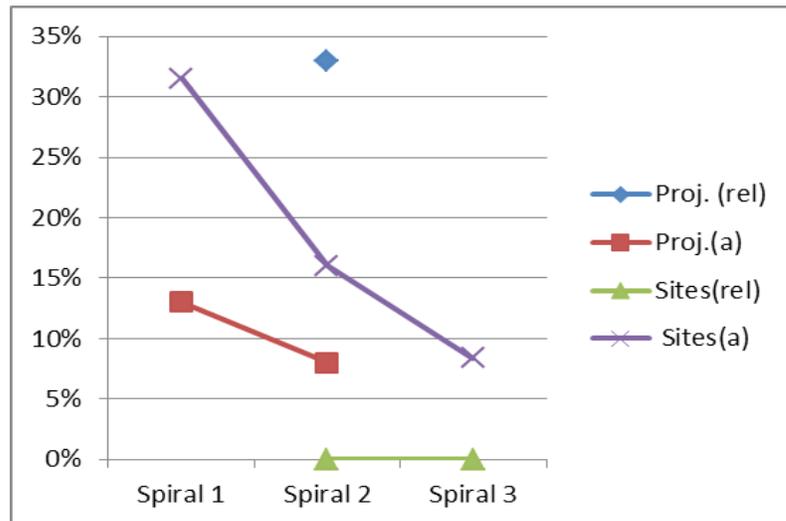


ORCA: The above graph shows ORCA's adoption rates in projects and sites. In Spiral 1, ORCA starts out with the same number of sites as ProtoGeni (11), but gains fewer new sites in Spiral 2 (13), which is a 118% growth. In absolute terms, however, ORCA remains stable, at the same level of dominance over the other clusters, 20.4%, from Spiral 1 to Spiral 2. In terms of new projects, ORCA has the highest growth among the four clusters as it starts with 5 projects and gains 7 new ones in Spiral 2, which is 140% growth. In absolute terms, this reflects an increase from 21.7% to 25.5% dominance over the other three clusters.

In Spiral 3, 34 new sites deploy ORCA's technologies, which is a growth of 142% from Spiral 2. This is a lower growth score than PlanetLab but higher than ProtoGeni's for the same

spiral. Overall, while ORCA doesn't "dominate" PlanetLab and ProtoGeni, it has an increasing trend in dominance in sites and in projects, both in relative growth scores and in absolute scores.

Figure 5. ORBIT's Adoption Rates



ORBIT: The above graph shows ORBIT's adoption rates in projects and sites. ORBIT starts out with the highest number of sites in GENI, 17, but doesn't gain any new sites in Spiral 2, neither in Spiral 3. In absolute scores, that reflects a decrease in dominance from 31.5% in Spiral 1, to 16.4% in Spiral 2, to 8.4 % in Spiral 3. In terms of new projects, ORBIT starts out with 3 projects in Spiral 1 and gains only 1 new project in Spiral 2, which is a relative growth of 33%. In absolute scores, that reflects a decrease in dominance from 13% to 8.5%. Overall, ORBIT has a decreasing trend in dominance of projects and sites despite starting out with the highest number of sites.

In summary, ORBIT has the highest absolute number of sites and ProtoGeni the highest absolute number of projects in Spiral 1. In Spiral 2, ORCA has the greatest growth in projects and ProtoGeni the highest absolute score of dominance in sites. In Spiral 3, PlanetLab has the highest growth in sites and the highest absolute score of dominance.

7.2 PATH DEPENDENCE

A timeline of development-related incidents is shown in table 9, indicating the time and type of incident (event, activity, decision), and its source in the dataset. The timeline presents a series of incidents related to GENI's development, starting from the earliest noted event that led to the creation of Emulab (a precursor system of PlanetLab and ProtoGeni) up until the end of Spiral Three.

Table 10 shows the timeline with additional coding of incidents into self-reinforcing mechanisms that should in theory make technology adoption path-dependent according to Sydow et al (2009). As in the timeline (table 9), each incident in table 10 is an event, decision or activity, and also is coded as a mechanism that promoted or inhibited path dependence. That way, the timing and sequence of events that promoted or inhibited the creation of a path of adoption can be considered for each cluster.

The coded incidents are also arranged in a flowchart showing GENI's timeline of events, activities and decisions along with the self-reinforcing mechanisms (figure 6). Each band in the graph represents a cluster and shows the sequence of development-related events, activities or decisions that reinforced or inhibited the creation of adoption paths. Arrows link those incidents

that were explicitly connected in the data, by being mentioned as such by interviewees or in messages in the developers' listserv. Incidents that were not explicitly connected but for which a relationship can be inferred indirectly from the data are presented without arrows. Each incident is labeled with its code that corresponds to the same incident in table 10, e.g., T1 is the first triggering event and it is placed in PlanetLab's band in the flowchart because it involved scientists and other participants that would later create PlanetLab. Some events that involve two clusters are placed in both bands, e.g. the integration of PlanetLab and ORCA was a decision made by developers of both clusters so a box representing that decision is placed in both PlanetLab's and ORCA's band in the flowchart. Those incidents that inhibited or counteracted the development of an adoption path are marked in blue (e.g., all integration efforts), while incidents that led to potential or actual inefficiencies (according to Sydow's et al framework) are marked in red.

As is shown in the flowchart, an adoption path emerges for some clusters during certain periods but not for all clusters and not consistently throughout the time examined in the data (2001-2011). For example, a path of adoption emerges for PlanetLab during the period preceding the design of GENI until the end of Spiral One. That path starts with two "triggering" events, labeled T1 and T2 in the flowchart that establish the need for a large-scale infrastructure system to support research in networking. The two triggering events led to the creation of Emulab and PlanetLab. By 2002, PlanetLab had already been adopted by several research projects, starting with Princeton's, which made coordination of research efforts and coordination of management of the overall cyber-infrastructure more effective (creating thus "coordination effects" according to Sydow et al's terminology). Those are shown in the flowchart as CO9. The effects of improved coordination led to project investigators choosing to work with PlanetLab repeatedly a

number of times and drawing more new users (creating a “persistent” choice according to Sydow et al. 2009 (P5a and P5b in the flowchart)).

During Spiral One, PlanetLab continues to have a path of adoption, as its already large user base expands through promotion efforts of its technologies by PlanetLab developers and administrators (EXP19 in the flowchart). At the same, developers with prior experience in PlanetLab have an advantage over developers in other clusters that are just starting to design their cluster technologies or need to modify them to make them usable. That creates “learning effects” in Sydow et al’s terminology (L17 in the flowchart) that further reinforces PlanetLab’s adoption and success in gaining new users and grant funding. After Spiral One, it is not as obvious that PlanetLab’s adoption proceeded according to a “path” or a clear trajectory of interrelated events, activities and decisions.

Table 9. GENI Timeline

Entry Number	Year	Cluster	Event/Activity/Decision	Source
1	2002	PlanetLab	PlanetLab created	Interview with GENI funder
2	2003	ORCA	Paper called “sharp” co-authored with [colleagues was published]. ORCA is a derivative of that”	ORCA developer /PI
3	2006	---	GENI project announced with a ~\$370 million budget through the year 2013	http://groups.geni.net/geni/attachment/wiki/OldGPGDesignDocuments/GDD-06-07.pdf
4	2006	ORCA	“GENI specing out control framework ideas – aligned with where ORCA is going”	ORCA developer /PI
5	2007	---	“BBN Technologies, an advanced technologies solutions firm, announced today it has been selected by the National Science Foundation to serve as the project office for the Global Environment for Network Innovations (GENI), responsible for managing the planning and design of an advanced network facility spanning the United States”.	http://www.bbn.com/news_and_events/press_releases/2007_press_releases/pr_may_21_2007_geni
6	2007	GENI-wide	“The GENI Project Office (GPO) announced today that it is actively soliciting proposals to help design GENI and begin its prototyping efforts.”	http://www.bbn.com/news_and_events/press_releases/2007_press_releases/pr_geni_121807
7	2007	ORBIT	“ORBIT wireless testbed project has own community of users”	ORBIT developers /Pis
8	2007	ORCA	[ORCA team] “so disconnected they just spent time to learn about each other”	ORCA developer /PI
9	2009	ProtoGeni	“The ProtoGeni team at the University of Utah has built a prototype GENI control framework, complete with software and a wide range of hardware.”	http://www.geni.net/?p=1246
10	2009	PlanetLab	GENI wrapper customized for EntGENI [EntGeni is ahead in development]	http://www.openflowswitch.org/wk/index.php/GENILight
11	2009	PlanetLab, ORBIT	PlanetLab, ORBIT integration	http://groups.geni.net/geni/attachment/wiki/ORBIT/ORBIT_PL_integration_report_9.08.pdf
12	2009	PlanetLab, ORCA	Gush structured to work with both PlanetLab and ORCA	http://lists.geni.net/pipermail/dev/2009-March/000042.html
13	2009	ORCA	Had to spend last 1.5 years [2009-2010] modifying to fit GENI. Not at all as mature as PG & PL	GENI Project office manager
14	2009	ORCA, IMF	When GENI Spiral 2 came out, decided to	ORCA developer /PI

			transition FIND grant into GENI effort to create GENI IMF	
15	2009	ORCA, LEARN	The Learn Project was without a cluster and saw Iliia's presentation, and said let's join that one	ORCA developer /PI
16	2009	ProtoGeni	In the second round, our cluster grew the most. It's pretty visible that we're building something that works"	ProtoGeni developer /PI
17	2009	ORCA, Plab, ProtoGeni	"Plab and Pgeni have a user base, T shirts, promotion, marketing." "Plaband ProtoGeni created an entire community – NSDI conference – could not do that style of research without Plab or PG"...[ORCA has] no established testbed with users, no armies of engineers"	ORCA developer /PI
18	2009	ORBIT	Activity/claim: "Experimenters need to know very little to use ORBIT"	http://lists.geni.net/pipermail/dev/2009-March/000045.html
19	2010	ProtoGeni	Decision: GENI will use "ProtoGeni Rspec V2 as the format on the wire, Aggregates can use Translators to convert between formats"	http://lists.geni.net/pipermail/dev/2010-April/000270.html
20	2010	ORCA	\$12M [available] for the meso-scale experiments – ORCA Pis didn't know anything [about it].... It was in a presentation from Chip. Stanford group tied in very powerful at NSF – put in GENI proposal to BBN via the back door"	ORCA developer /PI
21	2010-2011	PlanetLab, ProtoGeni	"Trying to merge PG and PL because similar ...were developed based on the same architecture document"	GENI project office manager

Table 10. Mechanisms Driving Path Dependence

	Construct	Cluster	Quote	Description (<i>action, event, decision</i>)	Source
T1	Triggering event	--	Initial need for “network-wide peer to peer solutions”	Event	Interview respondent – GENI project manager
T2	Triggering event	PlanetLab	“[developer/scientist was] hired in 1998 – wanted to do large scale research in networks – but couldn’t get published. No method, no tools. Plab and Emulab created to facilitate this. Those tools created and people jumped on them like “fleas on a dog”.	Event	Interview respondent -ORCA developer /PI
P3	Persistence	ProtoGeni	“didn’t have a good understanding of the Control Frameworks. It was semi random that [I]chose Pgeni	Example of “semi-random” event leading up to Pgeni’s adoption	Interview respondent –ProtoGeni developer
P4	Self-reinf. mechanism -learning	PlanetLab	“[developer/PI] brought up Emulab long time before virtualization things, grew as needed (8 years already)”	Activities (emulab’s development)	Interview respondent –Emulab developer
P5	Persistence	ProtoGeni	“Independently of GENI they wanted to use emulab so had already gone down that path”	Event (emulab already built)	Interview respondent –ProtoGeni developer
L6	Self-reinf. mechanism -learning	ProtoGeni	“In Spiral 1 – people already working with our cluster – many had their own emulab installations, some have now created their own. In Spiral 2, brought in new collaborators”	Decision	Interview respondent –ProtoGeni developer/PI
CO7	Self-reinf. mechanism -coord.	PlanetLab	[in 2003 Intel and HP] “put together consortium for PlanetLab..... and put Plab all over the place. Did more and more projects on Plab.	Events	Interview respondent - PlanetLab developer
P8	Persistence	ORCA	“we started doing infrastructure as a service	Decisions: ORCA is not being	Interview respondent – ORCA

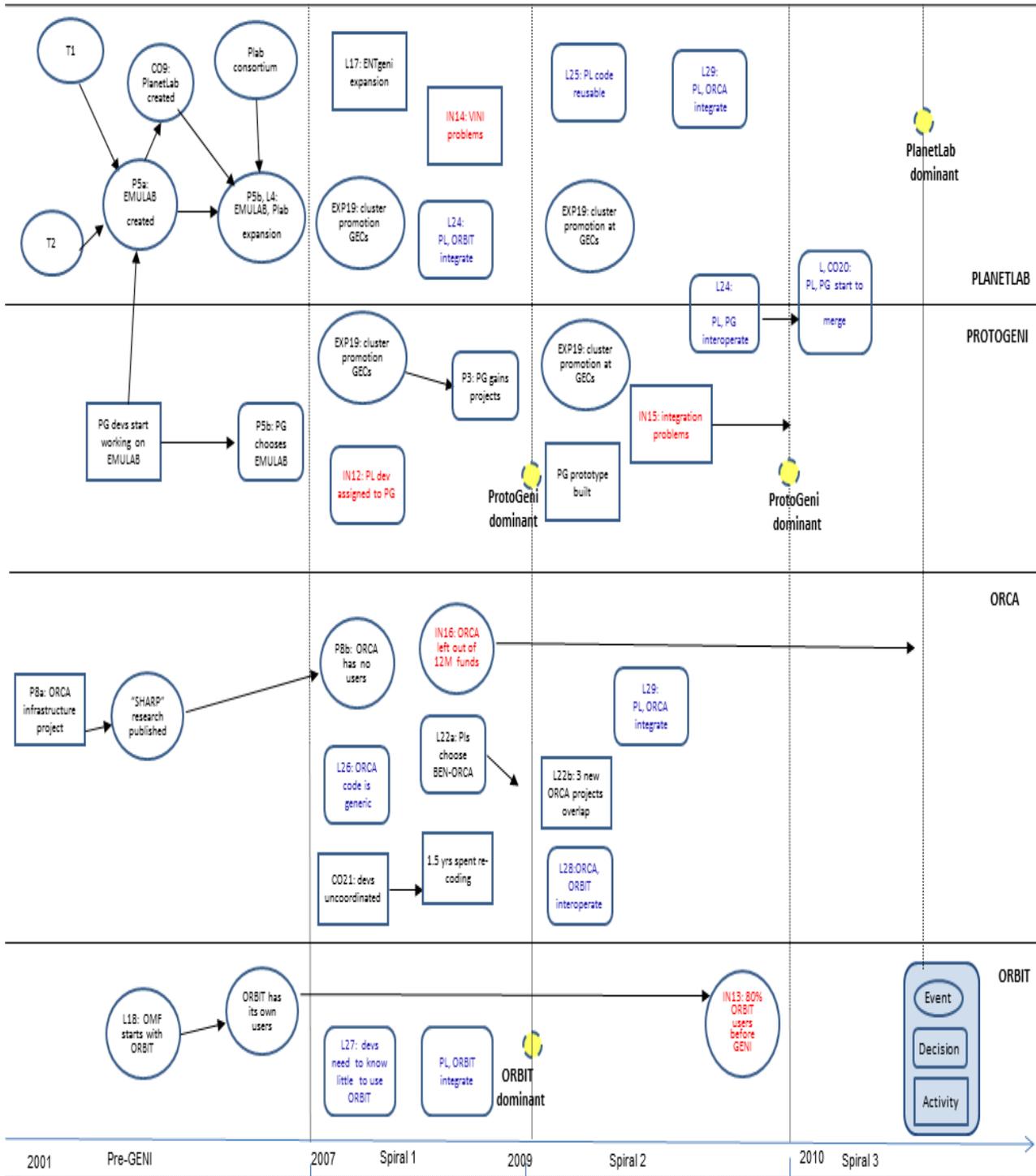
			in 2001, That was a research project, smaller, didn't get as many grants, no users"	selected/adopted by users and not a priority for funding agencies early on	developer /PI
CO9	Self-reinf. mechanism –coord.	PlanetLab	[PlanetLab was] “up and running at Princeton by Fall of 2002. All that stuff built and then institutionalized it so it was too late for people to wreck it”	Activity	Interview respondent - PlanetLab developer
CO10	Self-reinf. mechanism (learning, coord)	ProtoGeni	“We’ve developed this emulab software [for 10 yrs now]–We naturally became the center of this”	Past actions leading to ProtoGeni’s dominance	Interview respondent- ProtoGeni developer/PI
P11	Persistence	PlanetLab	“Emulab/Plab had large user communities before BBN in the arena”	Past event(s) leading to PlanetLab’s dominance	Interview respondent – GENI project manager
IN12	Potentially inefficient choice	ProtoGeni	[I] had been working with Vinnie at Princeton [PlanetLab]...submitted the spiral thing but was put in cluster B (i.e., ProtoGeni). PI (from PlanetLab) called me and said why are you in B, you’ve been working with us. It didn’t make sense”	Decision	Interview respondent –ProtoGeni developer
IN13	Potentially inefficient choice	ORBIT	“80% of ORBIT was from before... Most users are ORBIT users not GENI.... Our existing user community doesn’t feel a need for anything beyond what [they] already have”	Event/decision: 80% ORBIT users came from before GENI/ locked-in to ORBIT before other projects/options became available	Interview respondents –ORBIT Pis
IN14	Potentially inefficient choice	PlanetLab	“We started down that path with Vini ... We made this available to researchers, but they didn’t come running.”	Activity/decisions (VINI testbed, PI)	Interview respondent - GENI project manager
IN15	Potentially inefficient choice	ProtoGeni	“we picked ProtoGeni because we were assured that integration was as painless as possible – this	Decision proved inefficient	Interview respondent- ProtoGeni developer

			turned out not to be true”		
IN16	Potentially inefficient choice	ORCA	[in 2010 funding available / opportunity] “\$12M for the meso-scale experiments – ORCA Pis didn’t know anything [about it]”	Decision (funding decision inefficient as it excludes one framework – ORCA hasn’t heard about the funding opportunity. Without funding, fewer projects/opportunities for prospective users/adopters/developers/experimenters	Interview respondent – ORCA developer /PI
L17	Self-reinf. mechanism learning effects	PlanetLab	“we got a small grant to do enterpriseGENI from GPO – before we do large nationwide GENI let’s see if we can do it on a campus... Wrote another proposal to GPO – now that we can do it on our campus, why not do it on other campuses as well”	Activity/series of activities	Interview respondent – PlanetLab developer/PI
L18	Self-reinf. mechanism -learning	ORBIT	“OMF tools and software are not tied to a specific testbed technology. Indeed, OMF has been deployed and maintained on multiple testbeds with many different types of technologies”	Activity: this should reduce /prevent path dependence (prior to 2007)	http://ORBITlab.org/wiki/Documentation/OML#ForExperimenters http://omf.mytestbed.net/wiki/omf/Introduction
EXP19	Self-reinf. mechanism expectation	PlanetLab ProtoGeni	“Plab and Pgeni have a user base, T shirts, promotion, marketing”	Activities: Plab and Pgeni promote their technologies, can influence users’ expectations,	ORCA developer /PI
CO20	Self-reinf. mechanism -coord.	PlanetLab ProtoGeni	“Now Trying to merge PG and PL because [they are] similar... were developed based on the same architecture document”	Activity (funding and management of the similar 2 clusters more efficient for GPO)	Interview respondent - GENI project manager
CO21	Self-reinf. mechanism -coord.	ORCA	“in 1 st year [ORCA developers] so disconnected they just	Events: Coordination setbacks in ORCA	Interview respondent – ORCA developer /PI

			spent time to learn about each other”		
L22	Self-reinf. mechanism - learning	ORCA	“We wanted to find a cluster to fit in – and we found ORCA (D) – they have a BEN network in North Carolina that fits us. Keren found the cluster to join. Came up with unified measurement framework to connect into that cluster. Last summer we started the IMF and LEARN projects in Spiral 2. ERM, IMF and LEARN – there are lots of overlaps among the projects”	Decisions	Interview respondent - ORCA developer /PI
L23	Self-reinf. mechanism -learning	ProtoGeni PlanetLab	“The GPO has just completed and posted an initial draft of “GENI Aggregate Manager API v1.0”. This API represents convergence of interface designs used by the PlanetLab and ProtoGeni control framework projects. Once those projects have completed their migration to the API, any aggregate that supports this API will be interoperable with both.”	Event (this should reduce path dependence since same API can be used in both clusters)	http://lists.geni.net/pipermail/dev/2010-September/000165.html
L24	Self-reinf. mechanism -learning	PlanetLab ORBIT	“The proxy enables PlanetLab users to include ORBIT nodes in their experiments. PlanetLab users can log into their slivers and start their experiments.”	Activity: PlanetLab, ORBIT integration,	http://groups.geni.net/geni/attachment/wiki/ORBIT/ORBIT_PL_integration_report_9.08.pdf
L25	Self-reinf. mechanism -learning	PlanetLab	“PlanetLab is an excellent example of an ‘AM’ which hosts multiple services (Codeen, OpenChord?) which other experimenters rely on. Obviously, everyone can roll their own management API, but why couldn’t they re-use one (and all its infrastructure) if there is one, especially for environments which need to make resource allocation decisions and	Activity (PlanetLab’s AM code can be reused)	http://lists.geni.net/pipermail/dev/2010-September/000177.html

			where some form of resource accounting is necessary.”		
L26	Self-reinf. mechanism -learning	ORCA	“The ORCA framework is designed to be substrate-independent, ... Substrate independence ensures that the control framework is generic”	Activity	https://geni-ORCA.renci.org/ORCA-doc/current/guides/devel/substrate.html
L27	Self-reinf. mechanism -learning	ORBIT	“Experimenters need to know very little to use ORBIT “	Activity	http://lists.geni.net/pipermail/dev/2009-March/000045.html
L28	Self-reinf. mechanism -learning	ORBIT, ORCA	“User-Defined Experiments for DOME Systems Supporting WiMAX. We are adding WiMAX 4G clients to some of the DOME buses and installing a WiMAX base station on the Umass campus”	Event, activity (ORBIT, ORCA interoperate)	http://geni.cs.umass.edu/dome/docs.php
L29	Self-reinf. mechanism -learning	PlanetLab ORCA	ORCA- GUSH Integration	Activity	http://lists.geni.net/pipermail/dev/2009-March/000042.html

Figure 6. GENI's Development Timeline



According to path-dependence theory, and to the framework proposed for *organizational* path dependence by Sydow et al (2009), a path is initiated by certain “triggering” events that promote or motivate the technology’s adoption. In GENI’s case, participants identified two related triggering events leading to the creation of Emulab (T1 and T2 in the flowchart): a research lab’s or researcher’s need for large-scale network infrastructure and later on, the recognition of that need by the wider academic community and GENI’s funding agency: “[scientist I.V. was] hired in 1998 – wanted to do large scale research in networks but couldn’t get published. No method, no tools. PlanetLab and Emulab were created to facilitate this. Those tools created and people jumped on them like “fleas on a dog” (interview respondent’s note, T2 in table 10).

In addition, path-dependence creates potential and/or eventually actual inefficiencies as the result of repeated adoption decisions. In theory, inefficiency could be created by the adopted technology’s sub-optimal design and the availability in the market of superior alternatives which repeated and widespread adoption of the dominant technology makes costly or impossible (e.g., QWERTY keyboard design). Sydow et al (2009) point out that while inefficiencies are easily detected in retrospect, after a design or a technology has become dominant in the market, they are not as easily detected at the outset during the technology’s introduction or development. They propose that since organizational decisions and practices are not easily characterized as completely (in)efficient but can be both in different aspects or increase/decrease in efficiency over time, potential not actual inefficiency should be a sufficient characteristic to distinguish a path-dependent process (and to also predict one prospectively rather than retrospectively).

The period preceding GENI’s development does not include any potential or actual inefficiencies (either decisions or activities). All four clusters have a pre-existing technology or

theoretical design on which their subsequent development is based (in the flowchart those are P5a for PlanetLab, P5b for ProtoGeni, P8a for ORCA, and L18 for ORBIT). The availability of multiple alternative designs for GENI even prior to its conception can in theory mitigate the risks of developing a large-scale cyber-infrastructure based on a single design and the risks of investing in its development. If a design were to prove unworkable, an alternative could be selected from the pre-existing options and be built upon or expanded, minimizing the likelihood of failure or the need for additional funding to develop completely new designs. Even though that would prevent GENI's development from becoming path-dependent on a single design and thus 'inefficient', it does not necessarily prevent the clusters themselves from becoming path-dependent, that is, it might prevent path-dependence at the level of GENI (the overall infrastructure) but not at the cluster level.

All four clusters were based on a single pre-existing technology which sometimes seemed the "natural" choice for them (respondent's note in CO10 in table10) and it is unclear what alternative options their developers considered. In some cases developers had little knowledge of cyber-infrastructure technologies: as a ProtoGeni respondent mentioned "[I] didn't have a good understanding of the Control Frameworks. It was semi random that [I] chose ProtoGeni" (P3 in table 10). Other developers' choices were pre-determined by past experience even before all four alternatives had been available: "Independently of GENI they wanted to use Emulab so had already gone down that path", "We've developed this emulab software [for 10 yrs now]—We naturally became the center of this" (ProtoGeni developer, CO10 in table 10).

While none of the above observations indicate any actual downsides or inefficiencies with developers' choices, the manner in which the choices were made, with limited consideration of alternatives suggest that they could create or lead to potential inefficiencies in

the future. The scale and complexity of the infrastructure technology itself creates “sunk costs” early on for its developers and users in terms of specialized expertise required to build the system, and time and effort needed to modify or adapt it for different research purposes. As ProtoGeni and PlanetLab were both based on the same pre-existing technology, Emulab, any potential failure, lack of fit or unworkable mechanism in Emulab would have had to be addressed by both clusters, increasing the potential costs of such efforts for both developer teams.

Similarly, ORCA was based on a single research project, “SHARP”, pre-dating GENI while choices were also limited by the highly specialized nature of its technology (optical systems). While none of those events were actual downsides for its developers, the fact that its development started on a given path in a field with a narrow focus such as optical networks could in theory make it path-dependent and irreversible once alternative options became available and its developers could work with multiple systems (e.g, as ORCA integrated with other clusters in later Spirals).

ORBIT’s development was also based on pre-existing technologies, in fact “80% of ORBIT was from before” GENI as its PIs have noted (IN13 in the flowchart). That could lock-in its users to ORBIT even at the outset of GENI, and make switching or deploying other systems too costly for its project teams, much like Emulab and ORCA’s initial project success could prove a limiting choice for its developers at a later time.

While early technology choices made by cluster developers could only be considered potentially and partially inefficient, certain decisions made subsequently at the project level led to actual inefficiencies. For example, some projects based on early PlanetLab technologies misestimated its usefulness to prospective adopters: “We started down that path with Vini – We

made this available to researchers, but they didn't come running" (project manager's note, IN14 in table 10). Management's decisions sometimes were inefficient or perceived as such for developers themselves: a PlanetLab developer seeking funding during Spiral One was assigned to the ProtoGeni cluster, removing thus a PlanetLab contributor from his team and creating dissatisfaction for both the individual and the PlanetLab team: "[I] had been working with VINI [testbed] at Princeton [PlanetLab]...but was put in cluster B (i.e., ProtoGeni).[PI] from PlanetLab called me and said why are you in B, you've been working with us. It didn't make sense" (ProtoGeni developer, IN12 in table 10). Finally, a cluster's technologies could prove inefficient at later stages of development (e.g., Spiral Two) despite its early history, progress and visibility: as a developer notes, his project's choice of ProtoGeni was driven by unrealistic expectations: "we picked PG because we were assured that integration was as painless as possible – this turned out not to be true" (IN15 in table 10).

The above examples suggest that technology adoption decisions were sometimes less than optimal for the developers and their project teams, but do not indicate the presence of systematic inefficiency by anyone of GENI's project participants or managers. They seem to be instead isolated decisions that collectively created setbacks for GENI but their impact did not extend beyond their clusters. Nevertheless, at the cluster level, their potential and actual effects often could persist throughout the Spirals despite that cluster's increasing adoption, or were not sufficiently powerful to influence a cluster's adoption trajectory, which suggests that that adoption had been path-dependent on the cluster's early history. Misguided expectations, changes in a project's development team (moving developers across clusters) and lack of interest in a technology have long-lasting effects on a cluster's internal coordination processes and its visibility and legitimacy in the scientific community. Developers and project managers need to

invest additional time and effort to counteract those effects, which might involve re-directing resources across projects, spending time modifying code and falling behind schedule, changing project plans and the project's direction of development. Overall, the presence of potential and actually inefficient decisions is consistent with path-dependence theory, and some clusters such as PlanetLab and ProtoGeni have become dominant despite such decisions. However, since their decision-making cannot be characterized as systematically or predominantly inefficient, its effect on adoption and their dominance was likely weak or less powerful than that of the learning, coordination and promotion effects that can make adoption path-dependent.

In addition to the presence of inefficient decisions, path dependent adoption is characterized by the persistence of choices that lead to a technology's dominance. By definition, an adoption "path" consists of the repeated selection of a certain technology or cluster by developers, managers or funding agents supporting its development. Evidence for repeated adoption decisions includes Emulab's persistent use and deployment by ProtoGeni and PlanetLab projects (P5a, P5b in the flowchart); the option to use its code was available since its creation in 2001, and that option became an actual decision soon thereafter (by PlanetLab in 2002) and repeatedly by both cluster's Spiral One and Spiral Two projects. Similarly, ORBIT's trajectory of adoption was formed by repeated adoption decisions over time, as 80% of its Spiral Three users had adopted its systems prior to Spiral One. Finally, the technical overlap of some of ORCA's newer projects (L22b in the flowchart) despite its lack of users early on indicates an increased or sustained adoption rate during Spiral Two, in which ORCA grows in projects more than any other cluster. In summary, evidence for path-dependence in the adoption or dominance of certain GENI clusters exists to some extent in terms of potentially inefficient decision-making and in terms of persistent technology adoption choices made by developers and project

managers. As no exact measure or amount of such decisions or adoption choices exists or is proposed in the literature, it is difficult to determine whether the evidence in the GENI case is sufficient to establish the presence of path dependence; rather, path dependence is the result of subjective interpretation which limits its explanatory power of the clusters' different adoption rates.

7.2.1 Mechanisms Driving Path Dependence

Pre-GENI (2001-2006)

As shown in GENI's timeline, PlanetLab is the oldest and has the longest history of all GENI clusters. It was created in 2002 based on the design and architecture of an earlier infrastructure, Emulab. While both ProtoGeni and PlanetLab are based on Emulab, and while some of ProtoGeni's developers had also been working on Emulab, PlanetLab was built earlier than ProtoGeni. This series of developments, starting with a lab's (or researcher's) need for large-scale infrastructure in 1998, then spreading among other academic researchers and funding agencies and leading to Emulab's creation, then to PlanetLab's design and becoming progressively more popular among researchers was a path that led to PlanetLab's high adoption rates at the beginning of GENI.

For a system's adoption to be considered path-dependent, certain self-reinforcing mechanisms need to exist that could motivate its adoption in ways that early adoption decisions would "reinforce" subsequent ones made by potential users, developers and other supporters or stakeholders. That way, its adoption over time can be considered "self-reinforced" and its

development path dependent on early adoption decisions. One such mechanism, proposed in theory by Sydow et al (2009) and also identified by interview respondents is coordination. As mentioned earlier, coordination refers to the existence of certain practices, rules, or routines that are adopted by multiple individuals or organizations in a field, and as they are replicated over time, new entrants to that field are more likely to conform to them and also adopt them because of their popularity. The fact that a practice has been chosen by the majority of players in a field reduces the coordination costs among them and among potential entrants into that field. In theory, coordination effects can pull a technology's adoption toward a certain direction in a consistent manner so that an adoption trajectory emerges that makes that technology's adoption seem path-dependent (Sydow et al. 2009).

In the case of PlanetLab, the existence of a single cyber-infrastructure (Emulab) prior to its creation, and the successful deployment and adoption of that infrastructure by researchers across institutions facilitated the "coordination" among its developers and potential users by functioning like a standard, or a common design tested by Emulab's adopters. Researchers who wished to use a large-scale infrastructure for new projects could use Emulab's tested design, could share resources with other PlanetLab users, or collaborate with PlanetLab's developers to modify and improve its design. The absence of Emulab or the existence of multiple similar systems at the time that PlanetLab was introduced would have created increased coordination costs among developers wishing to collaborate, share resources, or start new projects based on a reliable infrastructure. At the same time, industry support also functioned as a coordination mechanism, making PlanetLab more attractive and a more efficient choice for potential adopters who could have the support of PlanetLab's consortium that Intel and HP had helped create. Intel and HP, through that consortium, "put PlanetLab all over the place" and "did more and more

projects” on it as noted by an interview respondent (CO7 in table 10). Their support, along with Emulab’s pre-existing user-base helped expand PlanetLab substantially before GENI was even conceived by funding agencies and developers themselves.

In addition to coordination, another mechanism that gave PlanetLab sustained advantage and helped “build” a path of adoption over time was the learning advantage of its developers created by Emulab. The fact that PlanetLab was based on Emulab’s specifications and overall architecture made it easier for those experienced with Emulab to work with PlanetLab’s code without having to gain expertise in a different kind of technology. With Emulab at its core, new features and resources could be easier added onto PlanetLab depending on the needs of researchers, making it more usable or attractive to new adopters. Those two mechanisms, learning and coordination, could have promoted or “reinforced” PlanetLab’s adoption over time, making its adoption trajectory path-dependent even prior to GENI’s introduction.

Spiral One (2007-2009)

Learning and coordination effects for PlanetLab persisted during Spiral One, as more projects based on it could benefit from developers’ ongoing support and their coordinated efforts to modify, expand and improve its technologies. Spiral One projects whose developers had had experience with PlanetLab, such as EnterprizeGENI, gained sustained funding and quickly expanded across US universities (L17 in table 10 and in the flowchart). At the same time, PlanetLab developers had repeated opportunities to present their projects in the GENI conferences (EXP19 in the flowchart). PlanetLab projects collectively had the strongest presentations during Spiral One in terms of demonstrating or describing the advantages of their technologies (they used technical and usefulness frames in their presentations more so than the

rest tree clusters). Promoting the projects and their technologies increases their visibility and builds legitimacy among prospective adopters, a social influence effect that can, in theory, attract new users or experimenters to PlanetLab.

Sydow et al (2009) and theories of network effects suggest that such social influence can be a powerful mechanism driving the adoption of a technology by creating expectations among new users (or at least those exposed to social influence effects) that the technology being promoted will become popular and possibly dominate the market or the field in which they operate. Irrespective of its actual advantages over competing technologies, the one being promoted, advertised or presented in an effective manner, might eventually become popular and dominate the market because its sustained promotion leads individuals to expect that it is or will be adopted by others, thus motivating them to adopt it themselves. Sydow et al (2009) use the term “adaptive expectations” to describe this effect, indicating that individuals’ and organization’s expectations can change over time and “adapt” in response to social influence. A technology’s adoption then can become path-dependent as new users are swayed by promotion events and by a technology’s visibility, and adapt their expectations (or form favorable expectations) regarding its advantages over time. Repeated promotions, increased advertising and in general events that are designed to influence prospective users can predict the formation of a “path” of increasing adoption despite the availability of competing technologies in the market.

Despite the effects of coordination, learning and social influence, PlanetLab was not the dominant cluster by the end of Spiral One (2009). ProtoGeni had by that time more projects than any of the rest three clusters, thus “dominating” GENI in terms of projects, and ORBIT was also dominating GENI in terms of sites deploying GENI technologies. A possible explanation for

ProtoGeni's dominance according to path-dependence theory is that similarly to PlanetLab, ProtoGeni was also subject to the influence of multiple self-reinforcing mechanisms such as learning, coordination and expectations management. Its developers had extensive prior experience with Emulab before GENI which lowered the learning barriers to working with ProtoGeni technologies; as a developer and project investigator noted, in Spiral One "people [were] already working with our cluster – many had their own Emulab installations" (L6 in table 10). ProtoGeni developers also had opportunities to present and promote their projects and technologies at multiple GENI conferences (GECs); in fact, their presentations seem to have been strongest compared to the other clusters' in terms of demonstrating or explaining the usefulness of their technologies (ProtoGeni projects had collectively the most 'usefulness' frames) during Spiral One. At the same time, ProtoGeni developers made significantly greater efforts to promote their cluster with "T shirts, promotion, marketing" (EXP19 in table 10 and in the flowchart) while the other clusters did not invest in such marketing strategies beyond their conference presentations. Moreover, ProtoGeni, like PlanetLab, had substantial coordination advantages driving its adoption as its developers had been "already working with our [ProtoGeni] cluster – many had their own emulab installations" before the start of Spiral One (developer's note, L6). A common infrastructure design facilitated resource sharing, collaboration and modifications to the code so that prospective adopters could effectively use ProtoGeni's technology and easily coordinate with current developers who had been working with that cluster's code for ten years.

The effects of coordination, learning and expectations management could in theory explain ProtoGeni's dominance of GENI projects during Spiral One (assessed in absolute counts). Starting with its developers' work on Emulab ten years prior to Spiral One and their

decision to build ProtoGeni based on Emulab's code, and then continuing through their repeated demonstrations of ProtoGeni's usefulness to a wide audience, a path of increasing adoption (or diffusion) emerges from 2001 to 2009 leading up to ProtoGeni's dominance at the end of Spiral One. Path dependence, however, can only partially hold during Spiral One, as ProtoGeni dominates GENI only in terms of projects (the largest share of GENI sites is using ORBIT technologies during Spiral One).

Similar to PlanetLab and ProtoGeni, ORBIT had an early start and its own community of users before the start of Spiral One in 2007. For example, OMF, a project whose code was later used by the rest of the clusters, was originally an ORBIT technology (adopted first by developers that later built ORBIT, L18 in table 10 and in the flowchart). According to its developers, ORBIT and OMF were purposely built in a way that makes them easily deployable ("OMF tools and software are not tied to a specific testbed technology. Indeed, OMF has been deployed and maintained on multiple testbeds with many different types of technologies", developer's note in GENI listserv, L18 in table 10). Moreover, ORBIT required little prior experience in wireless networking from experimenters as well: "experimenters need to know very little to use ORBIT"(developer's note). These observations suggest that ORBIT would dominate GENI during Spiral One, its adoption driven by learning and coordination effects on prospective users. The results indicate that while it has the largest share of sites deploying GENI technologies (31%), it is not similarly dominant in terms of projects (ProtoGeni has the largest share of projects during that period). That suggests that path-dependence can only partially explain the emergence of a dominant cluster during Spiral One.

Overall, while path dependence is a plausible explanation for the dominance results during Spiral One, it does not seem to be a strong one since, as mentioned above, no single

cluster emerges as dominant for both projects and sites, and in addition, it can also apply to PlanetLab (which is not the dominant cluster in Spiral One). PlanetLab, is however, the second most dominant cluster for that period, with a relatively small difference between its share of projects and sites from that of the dominant clusters (ProtoGeni is 5% more dominant than PlanetLab in terms of projects and ORBIT is 3% more dominant in terms of sites). That suggests that while path-dependence does not exclusively explain the dominance of ProtoGeni and ORBIT in Spiral One, it has a partial explanatory value for the dominance of the first two most dominant clusters considered together.

Spiral Two (2009-2010)

In contrast to Spiral One, a single dominant cluster emerges in Spiral Two (2009-2010). ProtoGeni has the largest share of both projects and sites in GENI (measured in absolute % adoption) and also the greatest growth in sites relative to its adoption or dominance in the previous Spiral. This is not a complete dominance, at least not in relative growth terms, as ORCA has the greatest growth in projects relative to its adoption in Spiral One (the difference in their growth rates is however small, as ORCA grows by 140% while ProtoGeni grows by 137%).

Mechanisms that might explain ProtoGeni's dominance are, similar to Spiral One, learning effects due to its deployment of Emulab code since the beginning of its construction, and potential coordination effects persisting through the two Spirals (even though there isn't much evidence of new coordination effects different from those of Spiral One). The effects of expectations management (EXP19 in the flowchart and in table 10) also persist as ProtoGeni developers make new presentations and demonstrations of their projects (ProtoGeni and ORCA have the strongest, or potentially most influential, presentations during Spiral Two in terms of

the way they frame their technologies). Nevertheless, as there are no new coordination or learning mechanisms in Spiral Two acting on ProtoGeni's users, path-dependence theory would suggest that its dominance was driven by repeated promotion activities (creating favorable expectations) and by its early, pre-GENI, history: events such as Emulab's development and decisions to base its design on Emulab might have created sustained coordination and learning effects with a long-lasting influence through Spiral Two.

ORCA's growth in projects during Spiral Two, the largest among all clusters (140% from its Spiral One share of projects), can be better explained by its developers' activities and decisions during the same period than by its early history. Several new projects that joined ORCA in Spiral Two (IMF, LEARN, ERM: L22b in the flowchart and in table 10) had overlaps in their code as their PIs had decided to use the same base of ORCA/BEN core technology built in Spiral One. That decision, together with earlier decisions to keep ORCA's code generic (L26 in the flowchart) lowered any potential learning barriers for new adopters. At the same time, ORCA and PlanetLab are integrated during Spiral Two, which makes it easier, or more efficient, for PlanetLab users to also use ORCA's technologies (and vice versa). ORCA also integrated its systems with ORBIT's (L28 in the flowchart and in table 10), which similarly facilitates the adoption of both clusters by each other's users. Integration activities eventually lead to the two clusters' being able to deploy each other's technologies, giving thus a learning advantage to their developers and prospective adopters over those of non-integrated clusters. ORCA's integration with other clusters, together with its developers' conscious efforts to create overlaps across some of its projects might explain its having the greatest growth in projects during Spiral Two.

Nevertheless, if path dependence is to be considered a highly likely explanation, the "path" of events, activities and decisions driving ORCA's adoption would be relatively short,

covering mostly Spiral Two, as ORCA's early history does not promote the creation of such a path. ORCA's team was not well coordinated during Spiral One, spending over a year making changes to the code, while having decided to make ORCA generic enough to work with multiple technologies. The developers' coordination problems, and the fact that ORCA started out without a user community that could potentially support new adopters, discourage or delay adoption, and rather than create a path, would act as path-breakers for as long as their effects persist. Spiral One, then, would be unlikely to be dominated by ORCA, which is the case in the results, and ORCA's dominance in Spiral Two's would be limited or mitigated by the lingering effects of Spiral One (lack of funding and the generic quality of its technologies).

At the same time, the integration strategy might not just draw new users from other clusters, but also limit the extent to which those clusters' adoption is path dependent. Path dependence implies that users are "locked" in a particular technology because the option of adopting an alternative one is inefficient or impossible. The integration of clusters could counter-act or prevent such lock-in because technically, it expands the options for the users of those clusters that can use each other's code (it gives them an additional option). From the perspective of path-dependence theory, integration might increase the adoption of a technology (or cluster) but it does not necessarily make that adoption path-dependent but rather, can actually 'break' the adoption path and expand the choice of technologies for some users. On the other hand, lock-in is still possible; considered together as a composite system, the integrated clusters can lock-in their users much like a single technology or cluster can create lock-in. Depending on the presence of self-reinforcing mechanisms, and on whether those act similarly on both clusters at the same time, the adoption of the integrated clusters can in theory become path-dependent.

That, however, is not the case with ORCA since the Spiral One mechanisms acting on its users (generic code, overlap among some projects) are unique to that cluster, while new mechanisms that could potentially increase its adoption do not necessarily create lock-in for all users (integration). Overall, path dependence can only partially explain the dominance or adoption results of Spiral Two: the lingering effects of learning and expectations management can create a path of adoption extending ProtoGeni's dominance in Spiral One to Spiral Two. ORCA's dominance however, has a short-term path starting and ending with Spiral Two and would not affect all users similarly because its integration with other clusters can expand rather than limit the technology choices.

Spiral Three (2010 - 2011)

Integration as a strategy, while being driven by GENI's funding agency and project managers, is ultimately determined by the clusters' development teams and their willingness to overcome technical and coordination/collaboration barriers to achieve that goal. As a result, integration proceeds at a different pace in different clusters; PlanetLab, most likely because of its longer history and advanced stage, is more effective, or faster, in leading integration efforts. That, in theory, would give it an advantage over ProtoGeni in attracting users (and prospective adopters) from other clusters, but it does not make it the dominant cluster at the end of Spiral Two. Integration activities might instead have longer-term impact, influencing adoption during Spiral Three, as PlanetLab emerges as the dominant cluster at the end of that period. That, however, does not necessarily imply that path-dependence theory explains its dominance, since, as mentioned earlier, integration can lock-in new adopters to the integrated clusters as a whole while expanding the current users' choices (by creating the option of an additional cluster).

Path dependence could help explain PlanetLab's dominance in Spiral Three only by considering that cluster's early history. Self-reinforcing mechanisms present in its early stages of development would have to also be active during Spiral Three or their effects should persist until the end of Spiral Three. Coordination and learning effects that led to PlanetLab's dominance prior to Spiral One could have persisted through Spiral Three even though they were likely limited by the projects' increasing social and technical complexity: the addition of new members to the development team and experimenters that would need to coordinate their efforts, the complexity of the technology itself as new projects join the cluster and as PlanetLab is being modified to fit new sites and to interoperate with other clusters, and changes in policies or management practices by the management team and funding agency. On the other hand, those socio-technical issues would affect all clusters, however PlanetLab only was dominant among them, which suggests that its earliest adoption path might have had lingering effects on Spiral Three, counter-acting more recent developments, at least in terms of sites deploying GENI (as GENI did not have any new projects funded during that time).

In summary, the assumptions of path dependence theory must hold for the researcher to "assert" path dependence; if one or more of the assumptions do not hold that indicates that there might be a better explanation for the phenomenon than path dependence. For example, if there are two or more technologies with similar widespread adoption, then there is no single technology that "dominates" the field. Sydow et al (2009) note that path dependence does not have to lead to dominance, in other words a technology can have a path-dependent adoption trajectory that leads to its becoming less popular. If there is dispute about which technology is dominant, then path dependence can still hold independently of the outcome as long as the adoption history of the technology the theory refers to satisfies the four assumptions in theory. In

the case of GENI PlanetLab's adoption was path-dependent early on and its adoption history satisfied the assumptions (or necessary conditions) proposed in theory by Sydow et al (2009). ProtoGeni also had a path-dependent adoption history up to Spiral Two and the theory's necessary conditions were present during that time frame. The implication is that path-dependence theory seems to hold and also lead to dominance rather than lack of dominance, but only for the two largest clusters.

Overall, path dependence is a partial or weak explanation for the entire period considered in the present study, starting around 2001 and ending in 2011 at the closing of Spiral Three. While an adoption path seems to exist for PlanetLab and is consistent with its widespread adoption prior to Spiral One, no concrete data exist showing the exact extent of its adoption before Spiral One. Starting in Spiral One, that adoption path becomes less clear as PlanetLab becomes second most-dominant cluster after ProtoGeni in projects and ORBIT in sites. GENI's dominance by any single cluster becomes less path-dependent in Spirals Two and Three as similar self-reinforcing mechanisms act on PlanetLab and ProtoGeni (large user base, early learning and coordination advantages, user expectations) and to a lesser extent on ORCA and ORBIT. While learning effects also benefit ORCA and ORBIT they are deliberately created by their development teams and build over time, as opposed to PlanetLab and ProtoGeni's early reliance on a widely used infrastructure (Emulab).

Similarly, technology promotion events and activities occur repeatedly, and all clusters attempt to influence user expectations. ProtoGeni developers make a greater effort to present their cluster as useful (in addition to being technically advanced), which might have reinforced ProtoGeni's advantage throughout all three Spirals. ProtoGeni is dominant in Spirals One and Two. An adoption path for it could have started with Emulab's creation, continuing through

decisions to base its projects on Emulab, the expansion of its use base and promotion activities during GECs leading to ProtoGeni's dominance at the end of Spiral Two. The adoption path becomes even less clear after that, as no new self-reinforcing mechanisms act on it (or any other cluster). PlanetLab's dominance in Spiral Three could have been predicted by early events and decisions prior to Spiral One, but not by any recent developments during the last two Spirals. Early history, then, might predict and explain dominance but only partially (or approximately) and its predictive and explanatory power decreases over time.

7.3 PROJECT PERFORMANCE

The results of the project performance hypothesis are shown on table 11. In Spiral One the ProtoGeni cluster has the highest score in project performance (94.8%), followed by PlanetLab (88.8%), ORBIT (80%), and ORCA (61.3%). In Spiral Two, PlanetLab has the highest score in project performance (65%), followed by ORCA (52.3%), followed by ProtoGeni (16.6%), and ORBIT which has negative score (-16%), indicating that that cluster has not made any progress in its milestones but rather has fallen behind in its development.

The project performance hypothesis suggested that a cluster's project performance will positively influence its dominance over other clusters in the following Spiral. That hypothesis was evaluated by examining the four clusters' adherence to milestones, that is, the number of milestones that each cluster reached on time or early minus the number of milestones that they had missed or met late. Each cluster has a score of project performance that represents the

number of on-time or early milestones minus the number of late milestones. Higher scores reflect superior project performance.

The results show a pattern that supports this hypothesis for some clusters but not all. The clusters with the highest dominance scores in Spiral Two (ProtoGeni with 209% relative dominance in sites, 137% relative dominance in projects, 33% absolute dominance in sites, 40.5% absolute dominance in projects) and Spiral 3 (PlanetLab with 193% absolute dominance in sites, 42% relative dominance in sites) also had the highest performance in the preceding Spirals (ProtoGeni had reached 94.8% of its milestones in Spiral One; PlanetLab reached 65% of its milestones in Spiral Two).

However, the positive relationship between performance and dominance does not hold for the second most dominant cluster in Spiral Two (ORCA is the second most dominant cluster in Spiral Two but did not have the second-best project performance in Spiral One: 61.3%, lower than PlanetLab's 88.8%). Similarly, the second-best performing project in Spiral Two (PlanetLab, with 65% reached milestones) is not the second-most dominant but rather the most dominant cluster in Spiral Three. Furthermore, the cluster with the third highest Spiral One performance (80%, ORBIT) is not the third most dominant cluster in Spiral Two, and the cluster with the worst performance in Spiral One (61.3%, ORCA) is not the least dominant cluster in Spiral Two.

Overall, the results suggest that project performance during a Spiral alone does not explain variation in all the cluster's subsequent adoption. Superior project performance is positively associated with dominance only in the case of the top two dominant clusters (ProtoGeni and PlanetLab). Low-performing clusters (ORBIT and ORCA) have low dominance in the subsequent Spirals but not in a consistent manner.

In addition, project performance does not explain variation in adoption and dominance when its effects are related to adoption and dominance in the same time period. During Spiral One PlanetLab had the highest project performance but ProtoGeni and ORCA emerged as the dominant cluster at the end of Spiral One. Moreover, ORCA had the lowest project performance but contrary to the second hypothesis it emerged as the dominant cluster together with ProtoGeni. During Spiral Two, PlanetLab again had the highest performance scores but ProtoGeni was dominant at the end of that Spiral.

Other factors might be at play influencing technological dominance besides the development teams' ability to meet milestones. One such factor might be the cluster's prior adoption rates, such that adoption rates in Spiral One influence adoption rates in Spiral Two. The results, however, support this alternative explanation only partially: whereas ProtoGeni and ORCA increased their adoption rates in Spiral Two, PlanetLab and ORBIT had decreased adoption rates in Spiral Two.

Another interesting observation in this data is that all clusters have decreased project performance in Spiral Two compared to their Spiral One performance. The leading cluster in Spiral One, ProtoGeni, shows deterioration in its performance from 94.8% to 16.6%. The largest difference in performance between the two spirals is that of ORBIT's, who initially had an 80% score but then stopped making progress and fell behind by 16% in its development schedule. A likely reason for this trend across all clusters is the fact that in Spiral Two new projects joined all clusters, and those new projects might have slowed down the development of the cluster that they joined. A second likely reason is that the development goals had changed in Spiral Two, emphasizing integration across the technologies rather than just incremental development. Whereas all clusters had managed to perform well in their own development, when they

attempted to make their technologies compatible with those of other clusters they started missing milestones and falling behind schedule.

Table 11. Project Performance Results

Spiral 1		GENI Cluster			
		Cluster B PlanetLab	Cluster C ProtoGeni	Cluster D ORCA	Cluster E ORBIT
Project Performance	Milestones on Time	94.4%	97.4%	48.4%	30%
	Late Milestones	5.6%	2.6%	19.4%	10%
	Early Milestones	0%	0%	32.3%	60%
	(OnTime+Early- Late) Milestones	88.8%	94.8%	61.3%	80%
Spiral 2		GENI Cluster			
		Cluster B PlanetLab	Cluster C ProtoGeni	Cluster D ORCA	Cluster E ORBIT
Project Performance	Milestones on Time	82.5%	52.8%	58.2%	29%
	Late Milestones	17.5%	41.7%	23.8%	58%
	Early Milestones	0%	5.5%	17.9%	13%
	(OnTime+Early- Late) Milestones	65%	16.6%	52.3%	16%

7.4 SOCIAL INFLUENCE

The results of the frame analysis for each cluster are presented in table 12. Each column includes the numbers of technical or use frames that each cluster used in a particular Spiral. For example,

PlanetLab's presentations, demos, and project documents (e.g., reports and slides) during Spiral One included 19 technical frames and 20 use frames. The presentations, demos and project documents that PlanetLab developers created in Spiral 2 included 13 technical and 10 use frames.

Table 12. Technical and Use Frames for each Cluster

	Technical frames-Spiral 1	Use frames-Spiral 1	Total -Spiral 1	Technical frames-Spiral 2	Use frames-Spiral 2	Total-Spiral 2
PlanetLab	19	20	39	13	10	23
ProtoGeni	8	24	32	29	28	57
ORCA	18	8	26	41	33	74
ORBIT	9	4	13	5	1	6
<i>Total</i>	54	56	110	88	72	160

In Spiral 1, PlanetLab is the cluster that used the greatest number of frames in its presentations, demos, and descriptions of its systems and technologies. However, contrary to H3, PlanetLab is not the dominant cluster in Spiral 2, neither in terms of projects nor in terms of sites deploying its technologies. Instead, ProtoGeni dominates the other clusters in terms of absolute number of new projects and new sites in Spiral 2. ProtoGeni also has a high number of frames in its presentations, higher than those of ORCA and ORBIT but not as high as those of PlanetLab. With the exception of ProtoGeni, there is a reverse trend for the rest three clusters to the one hypothesized in H3: high numbers of frames seem to be associated with lower dominance scores, and low numbers of frames with higher dominance scores.

If we examine the type of frames used by a cluster separately, based on whether a frame includes only a technical description of the cluster vs. a description of its possible uses and its

usefulness for experimenters and researchers, a different picture of the results emerges. The number of technical frames is positively associated with growth in dominance in Spiral 2 for ORCA and ORBIT, but the inverse is true for PlanetLab and ProtoGeni. ProtoGeni dominates the other clusters in terms of relative growth in sites and in absolute terms, but its presentations of its systems have few technical frames. The cluster with the most technical frames, PlanetLab, has a high dominance relative to ORCA and ORBIT but does not dominate ProtoGeni.

The number of use frames (descriptions of the technology's usefulness) is also inversely related to a cluster's subsequent dominance for ORCA and PlanetLab but not for ProtoGeni and ORBIT, both in terms of sites and projects. Similarly, ORCA's and PlanetLab's number of frames in Spiral 2 is inversely related to their relative dominance scores; that's true for technical frames and for use frames. If we look at absolute dominance scores, however, the number of use frames is positively related to a cluster's dominance in Spiral 2 for all clusters.

The social influence hypothesis is also not supported if we consider the impact of frames on adoption and dominance in the same time period instead of over time. During Spiral One PlanetLab has the highest number of frames but the dominant cluster at the end of Spiral One is ProtoGeni. ORBIT has the lowest number of frames but ORCA is the least dominant cluster at the end of Spiral One. Similarly, during Spiral Two, ORCA has the highest number of frames but ProtoGeni is the dominant cluster at the end of Spiral Two.

Overall, each theory can explain only partially some of the clusters' adoption rates and dominance, and only when the impact of the independent variables is examined over time across Spirals. None of the three theories completely explains any cluster's adoption, instead, different processes seem to account for variation in adoption and dominance. Table 13 summarizes the results for each cluster.

Table 13. Theory Fit for each Cluster

	Path Dependence	Project Management	Social Influence
Cluster B PlanetLab	Partially (pre-GENI, Spiral One, Spiral Three)	Yes	No
Cluster C ProtoGeni	Partially (for Spiral One)	Yes	Partially (only for use frames in Spiral Two)
Cluster D ORCA	No	No	Partially (only for technical frames in Spiral Two)
Cluster E ORBIT	No	No	Partially (for technical and use frames in Spiral Two)

8.0 DISCUSSION

This study examined the development of four inter-related cyber-infrastructure systems that comprise the GENI network. It focused on the question of why some of those systems (or their designs which are under development) seem to dominate others in terms of adoption by developers and researchers. Three theories were identified based on the literature of technology adoption, and data was collected and analyzed to determine whether each theory had any predictive power in suggesting which of the four systems might dominate others.

The results regarding path dependence theory suggest that that theory can only partially explain variation in the four clusters' adoption rates and dominance over time. While the theory suggests a positive impact of path dependence on adoption rates, the opposite trend emerged for some clusters, especially during the later development phase (Spiral 3). That shows that cluster developers and other project participants were able to maintain increased adoption rates over time despite the fact that the adoption of their technologies was not path dependent. Other factors, such as social influence, might have promoted technology adoption over time as the effects of path dependence decreased.

Similarly, the project performance results suggest that superior project performance in the technology development efforts did not fully account for differences in the clusters' adoption rates. While a reverse trend seems to describe the development of ORBIT projects, which collectively had decreasing project performance and decreasing adoption rates over time, no

positive relationship between project performance and dominance emerged over the whole period of the three Spirals for any cluster. That indicates that project performance was not a driver of adoption or dominance, and that GENI clusters were able to increase adoption of their technologies regardless of the efficiency of their project management.

The findings related to the frame analysis and the social influence hypothesis do not reveal any consistent patterns in the four clusters' dominance in the period from which data was collected (Spirals 1 through 3, 2007-2011). That suggests that social influence alone, at least measured in terms of technology frames, did not account for the variation over time in the four clusters' adoption or dominance.

Overall, while none of the proposed theories completely explained variation in adoption rates across the four clusters over time, each had partial explanatory value depending on the development stage (Spiral) and the cluster in which it was assessed. That suggests that additional factors might be at play, such as the cluster's size, as the largest clusters, PlanetLab and ProtoGeni, show similar trends. The type of technology developed in each cluster might also be a driving factor, as ORCA and ORBIT which developed more specialized types of networks (optical and wireless) had slower adoption rates. To examine the potential impact of such factors a larger sample size is needed so that cyber-infrastructure development projects can be reliably compared along dimensions in which they differ while also belonging to the same overall framework like GENI.

8.1 IMPLICATIONS FOR RESEARCH

A general position of this study is that CI systems have dominant designs that reflect competing organizing logics to which stakeholders ascribe. While there are multiple theories in the IS literature that could predict the emergence of dominant designs, their limitation to stand-alone, complete technologies that are already in the market does not make them easily generalizable to CI systems. This study's consideration of three such theories in the context of competing CI systems can contribute to the research stream of dominant designs and IT adoption while also extending the literature of cyber-infrastructure studies with a comparative examination of four CI systems. More specific implications of this study are discussed in greater detail below in relation to each theory considered in the analysis.

8.1.1 IT adoption and dominance

This study considered dominance along the three Spirals of GENI's development; if a shorter timeframe had been used to test the hypotheses, for example if the study had ended with Spiral Two, then the results would have been different. Without considering Spiral Three, PlanetLab will not re-emerge as the dominant cluster, as ProtoGeni was dominant at the end of Spiral Two. In that case, path dependence theory would be less powerful and ProtoGeni's prior social influence and project performance would seem to "break" PlanetLab's initial dominance. That would suggest that those two theories have greater effects on adoption as they enabled another system to dominate PlanetLab, and the assumption would be that their effect would hold over time. Including Spiral Three in the analysis shows that we can't assume that the effects of those

theories will hold over time, and that PlanetLab's path dependence or at least early history had a lasting influence that enabled it to eventually re-emerge as the dominant cluster.

8.1.2 Path dependence

While the study's findings do not suggest any significant or consistent influences on technology dominance over time, none of the theories can be rejected either, as each can partially account for differences in adoption rates at different times. More specifically, path dependence seems to have influenced the dominance of the two largest clusters and in the early stages of GENI's development, while social influence seems to have had an influence in later stages for the smaller clusters and in earlier stages for the larger clusters.

A contribution can be made to the research stream of dominant design in the IT adoption and diffusion literature. Empirical findings from past studies are generalizable in reference to an industry, a market niche or a single organization. Whether those findings apply to inter-organizational CI systems whose underlying design, such as the architecture or data model, is selected before the system is widely available for adoption and through multiple stakeholders' inputs cannot be inferred from the existing literature.

Consistent with findings in the IT adoption literature, this study shows that the adoption or dominance of inter-dependent, incomplete and large-scale cyber-infrastructure systems may depend on early events that set an adoption path like PlanetLab's. Where CI systems differ from standard IS, and a contribution of this study, is that their incomplete, ongoing development and interdependence can counteract the effects of path dependence, especially as the systems become more integrated and interoperable over time. A dominant design for CI systems might emerge early on, as this study and the literature on IT adoption suggest, but its dominance might not

persist over time as the CI system is being developed, modified and integrated with competing systems.

Moreover, different implications can be drawn depending on the timeframe in which dominance is considered. When each Spiral is considered separately, different clusters seem to be dominant at different times. That implies that different factors or social processes drive adoption in each stage of development. As a result, no single theory or social process can explain differences in adoption and dominance across clusters consistently over time. While theoretical frameworks such as Sydow et al's (2009) propose various processes that might lead to dominance, it is unclear from the theory whether those processes have effects that persist over time. If a technology emerges as dominant at a certain point in time, the assumption in the dominant design literature is that it should continue to be dominant if its adoption is path dependent because according to path dependence theory early adoption leads to future adoption.

This study's findings do not support that assumption of persistent dominance over time, as different clusters were dominant in each Spiral. That implies that in the context of CI projects, predictions derived from the literature regarding dominance will need to be adjusted to take into account changes in dominance over time: additional social mechanisms and theories need to be identified that could cause such changes.

A more general implication is that the three theories examined in this study seem to interact in affecting adoption and dominance over time. This study considered three explanations of GENI's development independently of each other, however it is possible that these theories interact and their combined effects on adoption might have greater explanatory power. Path dependence, in particular, might reinforce the effects of social influence theory in the sense that

developers or a dominant cluster can use different frames to prolong their technology's dominance.

Path dependence might be also reinforced or sustained through superior project performance: Sydow et al (2009) propose coordination as a mechanism reinforcing path dependence over time. For example, in ProtoGeni's case, presentations and promotions of a technology using different frames, and superior team performance together seem to have led to its dominance in Spiral 2. These effects were short-term; in the longer term PlanetLab became dominant again (Spiral 3), so the interactive effects of team performance and framing might have a declining strength over time.

While there is a such a connection between project performance and dominance for PlanetLab and ProtoGeni, a closer look at their teams suggests that framing attempts were also influential in the adoption of those clusters. The combined effects of project performance and framing sustained high adoption rates for both clusters which suggests that dominance can be extended or sustained over time through better team coordination and different technology frames. Future studies can explore this research avenue by examining development teams of dominant and less dominant systems with variable project performance that also made attempts to influence their users through technology frames. Another avenue for future research involves considering those effects at different time frames along a system's adoption trajectory. The effects of project performance and framing independently or interactively might become less or more powerful in the long term. Future studies can identify factors or processes that explain their decreasing or increasing effect on adoption as a system advances through spirals or other stages of development.

In future studies, the theories of dominance and path dependence might need to include or be combined with social theories or mechanisms that can explain changes in dominance over time, or at least identify how path dependence ends and how long its effects should be expected to last. In this study, the integration of certain clusters seemed to increase their adoption while expanding the choices of CI technologies for the developers of the integrated clusters. That strategy, if pursued consistently might “break” the adoption path and eventually end path dependence. That is a possible outcome in theory but cannot be inferred from the present results, rather, more extensive studies will need to test the persistence of path dependence and how dominance may shift over time as a result of that.

When the period of the three Spirals is considered as a whole, including events predating GENI’s design (2001-2011), the same cluster that was dominant early on in 2002 (PlanetLab) emerges as the dominant design at the end of Spiral Three in 2011. That implies that dominance can be predicted by early history and that it persists in the longer term even though it might shift in the intervening years between the earliest presence of a CI design (PlanetLab in 2002) and the time of the study (PlanetLab is dominant again in 2011). That implication however is more likely to be a unique characteristic of GENI and a result of the timeframe chosen for the current study rather than a generalizable statement. A future study with different start and end dates, e.g., considering a shorter timeframe or a timeframe extending beyond 2011 will not necessarily have the same finding. The value of examining a system’s development over time is in the fact that different mechanisms and shifts in adoption can be considered at multiple points in time as opposed to considering only the start and end dates of the CI’s development project.

As mentioned earlier, studies in technology adoption and diffusion have typically focused on systems that are complete in their development, small scale and stand-alone but not on cyber-

infrastructure systems that are large-scale, incomplete as scientists begin to use them while they are being developed and expanded, and have a complex structure by including hardware, large scale databases, and equipment used by research scientists. The findings suggest that path dependence theory might not fully apply to cyber-infrastructure systems, even though early events in a system's development trajectory have a powerful impact in its adoption at later stages. Early history still matters, but not necessarily or not completely in the way specified by path dependence theory. As early events prior to a CI system's design prepare the ground for a path of decisions or actions that might make its adoption dependent on that path, other factors and processes, specific to CI systems can counteract the effects of early events.

Path dependence can be counteracted or prevented by social influence tactics and efforts made by the system's developers, funders, managers and other stakeholders to increase its visibility and gain legitimacy among scientists. Sydow et al (2009) discuss how decision paths can be changed or stopped by "path breakers", events or intentional actions taken by an organization's members to adapt long-established decision-making patterns or routines to better serve the organization. This study's findings suggest that such path-breaking actions might involve social influence: developers can play an active role in breaking or altering an adoption path by promoting their projects and technologies and managing the expectations of prospective adopters. Past studies of path dependence typically did not consider the role that developers or project managers of a technology might play in forming or breaking an adoption path since their focus was on complete systems. An adoption path began forming only after a system's development was finished and after that system had entered the marketplace and competed with similar technologies. Participants in the system's design and development could

not influence the adoption path; it was only users or other adopters that through their buying decisions could choose to switch to another system or keep using the current one.

An additional implication of the present study then is that developers and other project participants can set an adoption path through early design decisions and efforts to not only build a technically advanced system but also to promote its visibility even while the system is under development. Cyber-infrastructure systems are a good site for examining how path dependence might interact with other social processes to affect adoption patterns, and how developers and other project participants can actively influence the adoption path before the system is fully developed. Future studies in this line of research can focus specifically on the role of developers or other system stakeholders and examine additional factors or social processes beyond social influence that might interact with path dependence.

New studies can also focus on the role of other kinds of users besides developers and project managers, such as CIOs or educators who might have different expectations from cyber-infrastructure systems (e.g., users who would only adopt a CI system after it reaches production quality rather than during its experimental or early development stage). Past studies of path dependence tended to assume that a technology's user base was homogenous, and that users made a binary adopt/reject decision that led to the formation of an adoption path. CI systems offer an opportunity to examine the motivations and adoption decisions of a diverse group of users, who might have different needs, motives and expectations, and whose adoption decisions at different times during the system's development might have variable impacts on its adoption path.

Further, cyber-infrastructure systems that share a management structure like the four clusters in GENI might be a good setting for comparative studies of path dependence theory and

for identifying additional self-reinforcing mechanisms in addition to those suggested by Sydow et al (2009) and also for identifying counteracting forces to their effects, such as the integration strategies of the GENI clusters. The theoretical framework proposed by Sydow et al (2009) has not been empirically tested yet, so new studies could examine its fit to various kinds of cyber-infrastructure systems, and potentially identify additional mechanisms that sustain or break technology adoption paths.

More specifically, in the present study, system integration as a coordinated strategy seemed to inhibit the formation of an adoption path in the short term (during Spirals 2 and 3). If the effects of systems integration are examined in longer term periods they might not necessarily work against the formation of an adoption path. Future studies can focus on different kinds of technology strategies such as systems integration, and examine their impact in the longer term. That way additional mechanisms, and more specifically technical aspects specific to CI systems can be identified that promote or “break” the formation of adoption paths.

8.1.3 Project Performance

With regard to the project performance results, this study could potentially make a contribution to research on project management in cyber-infrastructure and more generally IS development projects. Past studies of project management aspects of standard IT systems have shown that project management imposes significant costs to the development effort and that systems maintenance involves long term costs and can have significant impacts to the organization’s IT strategy. The present study considered only one aspect of project management, the development team’s success in meeting project milestones. An examination of additional aspects of project

management that includes multiple performance indicators besides met milestones would be a more comprehensive test of the project management hypothesis.

The adoption rates of low-performing CI clusters were not influenced by the cluster's project performance; project performance led to increased dominance only in the case of the two highest-performing clusters (PlanetLab, ProtoGeni). That suggests that project performance does not determine the competition among CI systems but rather it only has a partial impact on those systems with superior project performance in their design and development efforts. Low-performing CI development teams that seek to increase their system's competitiveness might be more effective by focusing on different factors that promote adoption more directly such as building a user community, promoting the system consistently and managing user expectations. While met milestones did not completely influence the adoption and dominance of all clusters during all Spirals, additional aspects of project management might have a greater impact when considered in a comprehensive manner. Future studies can focus on multiple aspects of project management, both technical such as the development team's project management approach, and social such as the role of project leaders or project managers in influencing team dynamics.

8.1.4 Social influence and frames

The focus on frame analysis represents one only approach of understanding the fit of social influence theory on technology adoption. Other aspects of social influence, such as social identity dynamics among scientists and other stakeholders, might play a role in influencing adoption and dominance. Moreover, the present study identified two kinds of frames (use and technical) that were consistent with Orlikowski and Gash's (1994) study. New kinds of frames might be identified in future studies by examining multiple modes of technology promotion

discourse besides presentations, demos and written materials (e.g., visual or interactive media presentations of CI projects, or presentations specifically designed for educators vs. researchers). Given the diversity of potential adopters and stakeholders of CI systems, an examination of various promotion efforts specifically targeted to different audiences could make both a research and practical contribution to the study of social influence and adoption of CI systems.

Future studies can also complement the present analysis of frames by examining not only the presence of frames but also changes in frames over time, how frames are constructed in the first place, and how they are shared and interpreted by groups of developers and users. The impact of frame dynamics can then be examined on different stages of the CI system's development or on phenomena that have a significant socio-cognitive aspect and have not yet been studied in the context of CI systems, such as escalation of commitment to failing projects (Newman and Sabherwal, 1996), and power dynamics (Markus 1983; Newman and Nobel 1990).

8.2 IMPLICATIONS FOR PRACTICE

Developing large-scale cyber-infrastructure systems is a time-consuming and challenging process that involves diverse stakeholders, multiple user groups, and inter-related technologies that are dispersed across different labs in academia or in some cases in the industry. Understanding the problems and success factors in the development of such systems is similarly challenging because of the scale of the systems involved and the diversity of the participants. This study represents only an initial attempt to understand four such systems in a comparative case study context. Further studies of CI systems are needed in order to derive 'lessons learned' that can be generalized to multiple kinds of CI development cases.

The practical implications of the current study, such as understanding how developers of CI systems frame the technologies that they develop and how differences in framing affect the technology's subsequent adoption or dominance might be useful to project managers and developers of CI systems. As the findings of the social influence hypothesis suggest, the manner of presentation can make a difference in a system's later adoption. Practically, the framing analysis suggests that presentations and in general promotion of CI systems are more effective when they include information about the system's usability and usefulness to current and prospective users.

Some clusters, such as ProtoGeni, also presented certain technologies as solutions to existing problems in the scientific community or among system developers. Framing a new system as a solution to a recognizable problem, or more generally emphasizing its usefulness for scientists, educators and students, seems to be an effective strategy and should be a practical concern for members of the development team. Developers and other project members should be consistently interacting with current users not only for the purposes of identifying technical problems and usability issues but also for discovering new possible uses of the CI technologies and different ways in which they can be useful to various user groups (e.g., researchers vs. educators). That will enable them to better promote certain technologies that are part of a CI system and make the case for their value to different audiences. Framing the CI system as a useful tool in multiple ways, together with more technical demonstrations can build interest among a wider range of users and eventually increase its adoption and popularity.

Coordinated promotion efforts can affect adoption as early as Spiral One (e.g., for PlanetLab and ProtoGeni). Developers and other members of the project team can increase their system's visibility and the chances of adoption even in early stages of its development.

Practically, that implies that rather than focusing team efforts and resources exclusively to technical aspects of the development and management of their project, team members should be also actively working to promote their system and engage in community-building efforts for prospective users. User support is typically considered a secondary concern for developers and managers of conventional IT systems, however for CI systems user support is a central part of the development project, as ProtoGeni and PlanetLab's promotion efforts show. Building a user community, forming a consortium for users or informally supporting users in gaining expertise in a new technology should not follow the development process and begin after a system has reached production quality, rather, should begin as early as the design or Spiral One stage to ensure sustained interest in the new technology.

Overall, the antecedents and drivers of adoption examined in this study suggest that a greater focus on the social aspects of the development of CI systems can be an effective strategy for generating interest and increasing their adoption. Practically that implies that designers, developers and other members of CI project teams have an influential role beyond their technical tasks; managers and project leaders will need to devise strategies to help developers take advantage of their potential to influence the system's adoption. The typical way of assigning tasks and responsibilities by dividing the technical component of the development work from the social one might not be the most effective way to influence its adoption. Developers who engage in both kinds of tasks, technical and social, in parallel might find out that they are more successful in generating interest for their system among prospective users. Standard project management techniques might thus not be the best fit for CI projects; new approaches that either combine current techniques or adapt them to the CI's context will need to be devised by project managers and leaders with the goal of having a sustained impact on the system's adoption.

9.0 LIMITATIONS

The single focus on GENI and the method of this study place limitations on the results that can be drawn from the analysis. Additional studies are needed to examine diverse CI systems, and with additional methods (e.g., social network analysis) in order to triangulate this study's findings. The study's findings are limited mainly by its methodological approach and its small sample size. Limitations that are common in qualitative studies, such as researcher bias in interpreting results, selective reporting of observations by interview respondents, and the recording and documentation of events, decisions and activities by project participants (e.g., meeting notes and progress reports archived in the GENI wiki) are all present in this study and limit the reliability of the findings. In addition, the small sample size limits the generalizability of the findings to other cyber-infrastructure development projects that do not share GENI's social and technical characteristics, method of development and funding structure. To increase the reliability and generalizability of the present study's findings, a larger sample of projects is needed that have a common management structure like GENI's support of four alternative cluster technologies, while varying in their adoption rates, development history, and social influence of prospective adopters.

In addition, a limitation associated with the data sources is the fact that the archival materials and other online documents have been selected by the project participants to be included in the project websites and in the GENI wiki. Other documentation and information that

is not available online might be missing from the analysis. Similarly, the presentations in the GENI conferences are invited, by the GPO, and missing data from projects that were not invited to create demos and presentations cannot be included in the analysis. Those limitations and research biases can be mitigated by grounding interpretations in theoretical frameworks and by stating any conceptual or methodological assumptions that were made between data collection, analysis and their connection to constructs and theories in the literature.

Moreover, path dependence assumes that the competing systems are similar to the extent that one can replace the other, and that users can choose either of them but not both or all of them at the same time without significant costs. The fact that ORBIT is a wireless technology and the other three systems are not is a limitation of the study; ORBIT might be practically comparable to the other three systems in that they all support large scale network experimentation but in theory and specifically in terms of path dependence theory it might not be comparable to them.

Another limitation of this study is that the role of industry stakeholders was not examined. Industry participants played a role in GENI's development since the earliest stages, prior to Spiral One and before GENI was even conceptualized. HP helped organize a consortium of Emulab users in 2003 which led to PlanetLab's having the advantage of a user support community at the beginning of its development in 2007. The role of the industry however is not mentioned in later project reports and presentations that were archived during the four cluster' development, which is a limitation of the study. Industry participants can exert social influence on prospective adopters by promoting certain technologies, by integrating some of their own pre-existing technologies into a cluster, and by providing funding and user support as HP did early on. ORCA's partnering with IBM might have increased ORCA's adoption rates in Spiral 3, and

generally industry support as social influence would be expected to increase the adoption rates of the technology/cluster they are partnering with. However, PlanetLab still emerged as a dominant technology having greater adoption than ORCA during Spiral 3 which suggests that industry support in ORCA's case didn't "break" PlanetLab's adoption path.

This study also did not examine the influence of exogenous factors as the focus was on how factors and processes internal to the GENI teams during development affect dominance.

Possible exogenous influences on adoption include:

- a) the support that development teams received from the project management office throughout GENI's development and changes in team structure and membership. Those factors can affect a team's ability to meet project milestones and affect its performance, which in turn might influence adoption indirectly. Resources and other forms of support can speed development, or hinder it if they are mismanaged, which in turn can influence the team's ability to meet its milestones. That might explain the ORBIT team's declining performance: lack of funding for equipment could have hindered development and as such the team fell behind in its development schedule. In theory, lack of resources would hinder development and decrease team performance, and ultimately impact adoption.
- b) The existence of technologies such as OpenFlow which are compatible with all clusters but not integrated equally across all clusters. That might give an advantage to clusters that can integrate and expand to include such technologies and as a result increase their user base.
- c) The stage of development of each cluster: more advanced clusters might have better project performance in terms of milestones, and ultimately as hypothesized, greater

adoption. The reverse might also be true: less advanced clusters might focus more on meeting milestones and coordinating the project instead of on development and as a result have greater adoption despite their missed milestones.

In addition, differences in stages of development among clusters might have played a role despite the fact that all GENI clusters were required to follow the Spiral method. ProtoGENI and PlanetLab seemed to be more advanced because their development was based on a pre-existing architecture (Emulab) whereas ORCA and ORBIT didn't not use Emulab as a starting point. However, as some interview respondents mentioned, ORBIT had a pre-existing design and a user base before GENI began, so that is a comparable starting point to ProtoGENI's and PlanetLab's. ORCA started out with a theoretical design concept and was less advanced than the other three clusters.

It is possible in theory that less advanced clusters miss more milestones because their development teams might need additional time and resources to coordinate their development. It is also possible that more advanced clusters miss more milestones because their development teams might be less concerned with coordination matters and deadlines and instead focus more on extending, improving or integrating the technology once the initial development phase is over and there is a working prototype on which further extensions can be made.

Past studies of project management in IT teams have not examined technologies at variable stages of development as an antecedent to adoption, but the stage of development can have indirect effects on adoption by influencing how much attention the team pays to milestones, through changes in the team's membership and leadership, and changes in the expertise that's required to complete the project. In GENI, all clusters reported consistently on their progress and

discussed their milestones in GEC meetings, which indicates they paid similar attention to milestones. Changes in team membership and leadership occurred as new developers joined a project – it is likely that some clusters experienced greater changes internally in their teams, and that might have affected their ability to meet milestones. This is a factor that was not measured in this study as the direct effect on adoption was hypothesized to be team performance on milestones and indirect effects such as team membership were not considered. That is a limitation of the analysis and can be a basis for future studies on how team performance affects technology adoption.

With respect to the second hypothesis about social influence effects on dominance, this study only examine one of many mechanisms of social influence. To assess the impact of social influence on adoption in a complete manner more factors should be included in the analysis, such as word of mouth and any additional ways in which developers can communicate with adopters. This study considered only framing effects as those were measurable through the teams' repeated presentations at conferences, as opposed to word of mouth which was not captured in GENI's archives. Further studies can complement this analysis by including an examination of word of mouth effects through interviews and surveys of GENI's adopters.

Another limitation of this study is that data on the independent variables were not collected during Spiral Three, as the goal was to test the hypotheses over time across Spirals. Even though the influence of the independent variables on adoption and dominance within Spirals One and Two was examined (no support was found for the project performance and social influence hypotheses), it is possible that the data analysis during Spiral Three might have found support for either hypothesis.

Finally, the goal of this study was to examine how development influences adoption, so projects external to GENI are not part of its development teams, but they might have an external impact on GENI's team dynamics if communication or any type of collaboration occurred among them. That is a limitation of the study and can be included in further analysis as exogenous variables that might affect adoption.

10.0 CONCLUSION

As scientific work becomes more collaborative across all aspects of the scientific enterprise (e.g., discovery, experimentation, data collection, documentation, evaluation, reuse), the importance of cyber-infrastructure systems in supporting and transforming scientific practice will increase, along with their becoming an integral part of scientific work. Cyber-infrastructure isn't a stand-alone technology but a platform (Baldwin and Clark 2000, Whitney et al 2004) that mediates and structures the interactions of users engaged in collaborative research, data sharing and data reuse. Selection or emergence of a dominant architecture for a CI system has implications not only for what software peripherals can be included in it but also for how data sharing, reuse and collaborative research will take place through the CI. The embeddedness of such systems in scientific collaboration and in institutional (university) policies, along with their large scale, increase both the risk of failure in their development and deployment, and the opportunities they can provide for collaboration across disciplinary and institutional boundaries. For those reasons the emergence or selection of a "dominant" design or underlying architecture for cyber-infrastructure systems has both practical and research implications beyond what can be inferred from studies of less complex end-user technologies.

The goal of this study was to examine the process by which a dominant design emerges for a CI system. The findings can make theoretical contributions to the research stream of IT adoption, and in particular, to studies of how the adoption of large-scale systems is influenced by

actions and decisions of the system's stakeholders while the system is still under development. More specific contributions can be made through the empirical testing of *organizational* path dependence theory and to research on technological frames and social influence with the examination of how the framing of CI technologies influences their adoption. Findings regarding the impact of project performance on adoption can be useful to managers and developers of large-scale system development projects and to funding agencies supporting those projects. While findings are tentative and inconclusive, they can be a starting point for examining the emergence of dominant designs for large-scale systems and the processes influencing their adoption over time. Future studies can extend this research stream by examining the role of early and ongoing history in the adoption of different kinds of large-scale systems, and how specific design decisions and project management approaches might affect the system's development trajectory.

BIBLIOGRAPHY

Abernathy, W. J. and Utterback, J. M. (1978). Patterns of Industrial Restructuring. *Technology Review*, 80 (7), pp. 1-9.

Anderson, D.P., Cobb, J., Korpela, E., Lebofsky, M., and Werthimer, D. (2002). SETI@home: an Experiment in Public-resource Computing. *Communications of the ACM*, 45(11), pp. 56–61.

Arthur, W.B. (1989). Competing Technologies, Increasing Returns, and Lock-in by Historical Events. *The Economic Journal*, 99 (394), pp. 116–131.

Arthur, W. B. (1994). *Increasing Returns and Path Dependency in the Economy*. Ann Arbor, University of Michigan Press.

Atkins. D. (2003). A report from the U.S. national science foundation blue ribbon panel on cyberinfrastructure. In *CCGRID'02: Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, page 16, Washington, DC, USA, 2002. IEEE Computer Society.

Baldwin, C.Y., and Clark. K.B. (2000). *Design Rules: The Power of Modularity*. MIT Press, Cambridge, MA.

Bietz, M. J., Baumer, E. P. S., and Lee, C. P. (2010). Synergizing in Cyber-infrastructure Development. *Journal of Computer Supported Cooperative Work (JCSCW)*. 19, pp. 245-81.

Bijker, W., Hughes, T., and Pinch, T. (eds), (1987). *The Social Construction of Technological Systems*, Cambridge, MA, MIT Press.

Boehm B, (1986). A Spiral Model of Software Development and Enhancement, *ACM SIGSOFT Software Engineering Notes*, 11(4) pp.14-24.

Boland, R. (1978). The Process and Product of System Design. *Management Science* 24(9), pp. 887–898.

Bowker, G. C. (1994). *Science on the Run: Information Management and Industrial Geophysics at Schlumberger, 1920-1940*. Cambridge, Mass MIT.

Buyya, R., Yeo, C.S., and Venugopa. S. (2008). Market-oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities. In Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08) IEEE CS Press, Los Alamitos, CA, USA.

Cavaye, A. (1996) Case Study Research: a Multi-Faceted Research Approach for IS, *Information Systems Journal*, 6, pp. 227-242.

Collins , T. and Bicknell , D. (1997) . *Crash: Learning from the World’s Worst Computer Disasters* . New York : Simon & Schuster.

Cusumano, M.A., Mylonadis, Y., and Rosenbloom, R.S. (1992). Strategic Maneuvering and Mass-market Dynamics: the Triumph of VHS over Beta. *Business History Review*, 66, pp. 51–94.

Dagwell, R., and Weber, R. (1983). System Designers' User Models: A Comparative Study and Methodological Critique. *Communications of the ACM* 26 (11), pp. 987-997.

David, P. A. (1985). Clio and the Economics of QWERTY. *American Economic Review*, 75, pp. 332–337.

David, P. A. (1994). Why are Institutions the “Carriers of History”? Path Dependence and the Evolution of Conventions, Organizations and Institutions. *Structural Change and Economic Dynamics*, 5, pp. 205–220.

Davidson, E.J. (2002). Technology Frames and Framing: A Socio-cognitive Investigation of Requirements Determination. *MIS Quarterly* 26(4), pp. 329-358.

de la Flor, G., Jirotko, M., Luff, M. P., Pybus, J., and Kirkham, R. (2010). Transforming Scholarly Practice: Embedding Technological Interventions to Support the Collaborative Analysis of Ancient Texts. *Journal of Computer Supported Cooperative Work (JCSCW)*, pp 1-26.

Eisenhardt, K.M. (1989). Building Theories from Case Study Research. *Academy of Management Review*, 14, pp. 532-550.

Faniel, I. M., and Jacobsen, T. E. (2010). Reusing Scientific Data: How Earthquake Engineering Researchers Assess the Reusability of Colleagues’ Data. *Journal of Computer Supported Cooperative Work (JCSCW)*.

Foster, I., and Kesselman., C. (1997). Globus: A Metacomputing Infrastructure Toolkit. *International Supercomputer Applications*, 11(2), pp. 115-128.

Foster, I., Kesselman, C., and Tucke, S. (2001) The Anatomy of the GRID: Enabling Scalable Virtual Organization. *International Journal of High Computing Applications* (15: 200).

Foster, I., Kesselman, C., Nick, J.M., and Tuecke, S. (2002). Grid Services for Distributed Systems Integration, *IEEE Computer*, 35 (6).

Galliers, R.D. (1992) Choosing Information Systems Research Approaches, in Galliers, R.D. *Information Systems Research: Issues Methods and Guidelines*, Blackwell Scientific Publications, Oxford.

Hand, E. (2007). Head in the Clouds. *Nature*, pp. 449-963.

Harter, D. and Slaughter, S. (2003). Quality Improvement and Infrastructure Activity Costs in Software Product Development: A Longitudinal Analysis. *Management Science* 49(6) pp. 784-800.

Helfat, C. E. (1994). Evolutionary Trajectories in Petroleum Firm R&D. *Management Science*, 40, pp. 1720–1747.

Hirschheim, R.A. and Klein, H.K. (1989). Four Paradigms of Information Systems Development. *Communications of the ACM*. 32(10), pp. 1199-1216.

Jaeger, P.T., Lin, J., and Grimes, J.M. (2008). Cloud Computing and Information Policy: Computing in a Policy Cloud? *Journal of Information Technology and Politics*, 5(3) pp 269–283.

Jelinek, M. and Schoonhoven, C.B. (1990). *The Innovation Marathon: Lessons from High Technology Firms*. San Francisco: Jossey-Bass.

Karasti, H., Baker, K. S., and Millerand, F. (2010). Infrastructure Time: Long-Term Matters in Collaborative Development. *Journal of Computer Supported Cooperative Work*, 19, pp. 377-415.

Karim, S. and Mitchell. W. (2000). Path-Dependency and Path-breaking Change: Reconfiguring Business Resources following Acquisitions in the US Medical Sector, 1978-1995. *Strategic Management Journal*, 21, pp. 1061-1081.

Kee, K. F., and Browning, L. D. (2010). The Dialectical Tensions in the Funding Infrastructure of Cyber-infrastructure. *Journal of Computer Supported Cooperative Work*, 19, pp. 283-308.

Kirsch, L. J. (1996). The Management of Complex Tasks in Organizations: Controlling the Systems Development Process, *Organization Science* , 7(1), pp.1-21.

Kirsch, L.J. (1997). Portfolios of Control Modes and IS Project Management, *Information Systems Research*, 8(3), pp. 215-239.

Kirsch L.J., Slaughter, S., Diamant, E.I., Ma, L., Haney, M (2009). Understanding Conflict in Large Multi-Party Information Systems Development: The Role of Faultlines. *Organizational*

Communication and Information. Systems (OCIS) Division, Academy of Management Annual Meeting (AoM' 09), Chicago, IL.

Kraut, R., and Streeter, L. (1996). Co-ordination in Software Development. *Communications of the ACM*, 38, pp. 69-81.

Krubasik, E. G. (1988). Customize your Product Development, *Harvard Business Review* 66(6), pp.46–51.

Kwak, Y.H. and Ibbs, C.W.(2000). Calculating Project Management's Return on Investment. *Project Management Journal* 31 (2), pp.38–47.

Langley, A. (1999). Strategies for Theorizing from Process Data. *Academy of Management Review* (24:4), pp. 691-710.

Langley, A. and J. Truax (1994). A Process Study of New Technology Adoption in Smaller Manufacturing Firms. *Journal of Management Studies* 31(5), pp. 619-652.

Lohr, S. (2007). Google and IBM Join in Cloud Computing Research. *New York Times*. Retrieved April 27, 2012, from:
<http://www.nytimes.com/2007/10/08/technology/08cloud.html>

Lucio-Arias, D., and Leydesdorff, L. (2009). The Self-Organization of Scientific Knowledge on the Basis of Exchanges and References among Scientific Texts: The Communication Turn in the Science of Science. *Journal of Informetrics*.

March, J. G. (1991). Exploration and Exploitation in Organizational learning. *Organization Science*, 2, pp.71–87.

Markus, M.L. (1983). Power, Politics and MIS Implementation. *Communications of the ACM*, Vol. 26, pp. 430-444.

Markus, M. L. and Bjorn-Andersen, N. (1987) Power Over Users: its Exercise by System Professionals. *Communications of the ACM* 30(6), pp. 498–504.

Miller, R., Hobday, M., Leroux-Demers, T., and Olleros, X., (1995). Innovation in Complex System Industries: the Case of Flight Simulation. *Industrial and Corporate Change*, 4 (2), pp. 363–400.

Moch, M., and Fields, W. C. (1985). Developing a Content Analysis for Interpreting Language Use in Organizations," *Research in the Sociology of Organizations* (4), pp. 81-126.

Monteiro, M. (2010). Reconfiguring Evidence: Interacting with Digital Objects in Scientific Practice. *Journal of Computer Supported Cooperative Work*, 19, pp. 335-354.

- Newman, M., and Nobel, F. (1990). User Involvement as an Interaction Process: A Case Study," *Information Systems Research* (1:1), pp. 89-113.
- Newman, M., and Sabherwal, R. (1996). Escalation of Commitment to Information Systems Development: A Longitudinal Investigation," *MIS Quarterly* (20:1), pp. 23-54.
- Olson, J.M., B. Nathan, B., and A. Zimmerman, A. (Eds.) (2008). *Science on the Internet*. Cambridge: MIT Press.
- Orlikowski, W.J., and Baroudi, J. (1991). Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research* (2:1) pp 1-28.
- Orlikowski, W., and Gash, D. (1994) Technological Frames: Making Sense of Information Technology in Organizations. *ACM Transactions on Information Systems* (12:1), pp. 174-207.
- Ribes, D., and Finholt, T. A. (2009). The Long Now of Infrastructure: Articulating Tensions in Development. In P. Edwards, G. C. Bowker, S. Jackson, & R. Williams (Eds.), *Special Issue on e-Infrastructure in the Journal of the Association for Information Systems*, 10(5), pp. 375–398.
- Robey, D., Farrow, D.L., and Franz, C.R. (1989). Group Process and Conflict During System Development. *Management Science*, 35 (10), pp. 1172-1191.
- Slaughter, S., Levine, L., Ramesh, B., Baskerville, R., and Pries-Heje, J. (2006). Aligning Software Processes and Strategy. *MIS Quarterly*, 30(4), pp. 891-918.
- Standish Group International, Inc. (2001), *CHAOS 2001: A Recipe for Success*, Standish Group International, Inc, West Yarmouth, MA.
- Sterman, J. and Witemberg, J. (1999). Path Dependence, Competition and Succession in the Dynamics of Scientific Revolutions. *Organization Science* 10(3) pp. 322–341.
- Stalk, G. (1988). Time—The Next Source of Competitive Advantage. *Harvard Business Review* July–August, pp. 41–51.
- Stalk, G., Jr., and Hout, T. M. (1990). *Competing Against Time: How Time-Based Competition is Reshaping Global Markets*. New York: Free Press.
- Strauss A. and Corbin J. (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Sage, Newbury Park, CA.
- Sydow, J., Schreyogg, G. and Koch, J. (2009). Organizational Path Dependence: Opening the Black Box. *Academy of Management Review*, 34, pp. 689–709.
- Tripsas, M., and Gavetti, G. (2000). Capabilities, Cognition and Inertia: Evidence from Digital Imaging. *Strategic Management Journal*, 21, pp. 1147–1161.

- Turner, J.C. (1991). *Social Influence*. Pacific Groves, CA: Brooks/Cole.
- Tushman, M., and Murmann, J. (1998). Dominant Designs, Technology Cycles, and Organizational Outcomes. In B. Staw & L. L. Cummings (Eds.), *Research in Organizational Behavior*, vol. 20: 231–266. Greenwich, CT: JAI Press.
- Tushman, M.L., Rosenkopf, L. (1992). On the Organizational Determinants of Technological Change: Towards a Sociology of Technological Evolution. *Research in Organizational Behavior* 14, pp. 311– 347.
- Utterback, J. (1996) *Mastering the Dynamics of Innovation*. Harvard Business School Press; Rev. Edition, 2006.
- Venkatesh, V., Morris, M. G., Davis, G. B., and Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View, *MIS Quarterly* 27(3), pp. 425-578.
- Vergne, J.P, and Durand , R. (2011). The Path of Most Persistence: An Evolutionary Perspective on Path Dependence and Dynamic Capabilities , *Organization Studies* , 32(3) pp. 365–382.
- Wheelwright, S. C. and Clark., K. B. 1992. *Revolutionizing Product Development*, The Free Press, New York.
- Whitney, D., E. Crawley, O. deWeck, S. Eppinger, C. Magee, J. Moses, J. Seering, W. Schindall, J., and Wallace, D. (2004). *The Influence of Architecture in Engineering Systems*. Engineering Systems Monograph, MIT Engineering Systems Division.
Available at:
<http://esd.mit.edu/symposium/pdfs/monograph/architecture-b.pdf>
- Yin, R. (2003). *Case Study Research*. Beverly Hills, CA: Sage Publications.