The Work of Developing Cyberinfrastructure Middleware Projects

Matthew J. Bietz
University of California, Irvine
Department of Informatics
Irvine, CA 92697-3440
mbietz@uci.edu

Drew Paine
University of Washington
Department of Human Centered Design & Engineering
Seattle, WA 98195
pained@uw.edu

Charlotte P. Lee
University of Washington
Department of Human Centered Design & Engineering
Seattle, WA 98195
cplee@uw.edu

ABSTRACT
Middleware software, which provides an abstraction layer between low-level computational services and domain-specific applications, is a key component of cyberinfrastructure. This paper presents a qualitative study of how cyberinfrastructure middleware development is accomplished in two supercomputing centers. Our investigation highlights key development phases in the lives of middleware projects. Middleware development is typically undertaken as part of collaborations between technologists and domain scientists, and middleware developers must balance the pressure to meet specific scientific needs and the desire to explore their own R&D agendas. We explore how developers work to sustain an ongoing development trajectory by aligning their own work with particular domain science projects and funding streams. However, we find that the key transition from being a component in a domain-specific project to a stand-alone system that is useful across domains is particularly challenging for middleware development. We provide organizational and national policy implications for how to better support this transition.

Author Keywords
Cyberinfrastructure; middleware; sustaining development; qualitative methods.

ACM Classification Keywords
H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces - Computer-supported cooperative work.

INTRODUCTION
The large-scale, advanced computational infrastructures known by terms such as collaboratories, cyberinfrastructures (CI), or e-Science, have been sites of study for CSCW for well over a decade [7,11,20]. In this paper we explore the development of a key element of cyberinfrastructure that has remained understudied: middleware. Middleware are software systems that act as an abstraction layer between low-level computational services and domain-specific applications. Understanding how to support the development of CI requires also understanding how middleware development is sustained over time.

Cyberinfrastructures are created to support collaboration in science and other academic disciplines. However the creation of cyberinfrastructure is also a collaborative enterprise, bringing together researchers in various academic domains with computer scientists and professional development teams. Cyberinfrastructures can be seen as large-scale, collaborative enterprises distributed over space and time, in other words, virtual organizations [8]. In this paper we examine the strategies developers undertake in order to enact collaborative cyberinfrastructures. Understanding the processes that shape the development of large-scale information infrastructures continues to be an important concern for CSCW [15].

Creating sustainable CI remains an important challenge [19]. Given the significant investment required to build cyberinfrastructures, they are expected to operate over decade-long (if not longer) time scales. Additionally, there are concerns about long-term preservation of the scientific record [12]. Here, however, we draw on an understanding of sustainability as "how to maintain the persistent human and technological arrangements that comprise cyberinfrastructure" [3]. We are primarily concerned with the ongoing work of sustainability; that is, the ways in which developers sustain (or sometimes don't) a development trajectory through significant social and technical disruptions and breakdowns. In this qualitative, interview-based study of software engineers at two supercomputing centers we ask: *how do developers sustain middleware projects in the context of cyberinfrastructure development?*

The sites for this study are two supercomputing centers in the United States. Formed in the mid-1980s to provide high-performance computing services to the nation's scientists, these organizations are central players in the development of cyberinfrastructure. Located at large

CSCW '13, February 23–27, 2013, San Antonio, Texas, USA.

research universities, these organizations bring together a wide range of domain stakeholders and systems developers to produce CI projects ranging from domain-specific science gateways & workflow systems down to the large-scale computing systems upon which computational scientific research relies.

The remainder of this paper is structured as follows: to position our work we first introduce cyberinfrastructure development, including previous work in CSCW, along with studies of the development of middleware. We then describe our research sites and methods before presenting our findings through a narrative that tells the story of the development of one successful middleware, along with analysis, commentary, and other examples from our data. We discuss three key aspects of CI middleware development and the work that supports this process. Finally, we reflect on the challenges of sustaining the development of CI middleware.

LITERATURE REVIEW

In our literature review we provide an introduction to the concept of cyberinfrastructure, define middleware and its role in cyberinfrastructure, and discuss the theoretical lens we use to examine our data on cyberinfrastructure middleware development.

Cyberinfrastructure Development

Cyberinfrastructure refers to a class of information and computation infrastructures in support of scientific and knowledge work. The term cyberinfrastructure was coined by a 2003 NSF Blue-Ribbon panel that examined the future of scientific computing [2]. Concepts like cyberinfrastructure, e-Science, and collaboratories are fundamentally concerned with distributed collaborative science which grows out of a trend in scientific research toward larger, collaborative projects studying large-scale phenomena through shared data sets [1,9]. Cyberinfrastructure refers to the infrastructural facilities needed to support this kind of science, and is based on information technology resources including high-performance computing and high-speed networking. Cyberinfrastructure relies not only on advanced technologies, but also on "human infrastructure," the rich and extensive arrangements of people, organizations, and communities that support the CI [14].

The cyberinfrastructure paradigm makes explicit comparisons between these new information infrastructures and traditional infrastructures such as roads and electricity distribution systems. This way of thinking focuses on the interconnectedness of the information systems more than on the individual software projects of which they are partially composed. Star & Ruhleder [24] emphasize that infrastructures are characterized by the relationships among the infrastructural components, they are embedded inside of other systems and infrastructures, and they are learned as part of membership in a community. Infrastructures form when separate components are consolidated into linked "networks" or "webs" of systems [6]. Infrastructure works because it structures the relationships among various entities in specific ways. Therefore to understand the development of cyberinfrastructures it is important to study the relationships between the various stakeholders of CI projects.

Lee et al. [14] posit the notion of the "human infrastructure" of cyberinfrastructure as a way to understand the myriad relationships that affect the work of building CI. Human infrastructure is the collaborative partnerships of researchers, software developers, managers, and other stakeholders who develop and maintain cyberinfrastructure. The notion of human infrastructure takes a perspective beyond that of teams to include organizational structures and personal networks as well. Using the notion of human infrastructure, the fluctuations in the relationships which support cyberinfrastructure development may be made visible and recognizable as multimorphous and constantly changing.

In the same vein, Bietz et. al [4] define the concept of synergizing to highlight the work of creating infrastructural relationships among individuals, technologies, and institutions. In this context, synergizing is the work necessary to enact productive infrastructural relationships. Synergy refers to the bringing together of two or more already existing entities to produce an effect greater than any of the single entities could produce alone. The overall lens of synergizing reveals that a major part of the work of developing cyberinfrastructure is concerned with building and managing the connections among people, technologies, and organizations. In other words, cyberinfrastructure development is inherently collaborative work. It is a sociotechnical process that relies upon the changing, multi-faceted relationships of software developers, domain-scientists, funding bodies, and institutions. Studying middleware development reveals the processes by which human infrastructure enables and supports cyberinfrastructure.

Middleware

Given this relational view of infrastructure, middleware provides a rich site for understanding cyberinfrastructure development. Middleware systems are a type of software that support distributed information systems. Middleware systems provide a common platform upon which domain-specific applications may be built. Fundamentally, middleware enables communication between disparate systems. Middleware, sometimes referred to as "glue software," are systems that "mediate communication and interaction between heterogeneous networked technologies" and provide "a kind of lingua franca that enables inscriptionally diverse systems to interact" [18]. In CI, middleware systems sit between lower-level systems like databases or authorization services and the applications designed for domain-specific user groups.

The term middleware dates to the original 1968 NATO conference report defining Software Engineering as a field [17]. Here d'Agapeyeff presented an inverted pyramid structure of software systems, with middleware placed between application programs and service routines. d'Agapeyeff stated that middleware is necessary because "no matter how good the manufacturer's software for items like file handling it is just not suitable; it's either inefficient or inappropriate" [17] (p.14). As software systems have grown in scale and complexity, middleware have maintained their role as an integral part of many systems that support a multitude of hardware platforms at low-level and domain-specific applications at the high-level.

Newman [18], in a study of corporate software development, finds that middleware development is subject to a number of organizational and business complexities. The development that we study takes place in an academic space where many of the organizational constraints differ, but many of Newman's insights apply. In particular, Newman finds that middleware development is particularly important because it "involves choices about standardization of new and important parts of the technical infrastructure that links parts of highly distributed enterprises together." Middleware not only allows technologies to interact; it also synergistically links people and organizations within collaborative enterprises.

Academic software development, at least in the contexts we studied, has key differences to corporate development. For example, Newman discusses how developers operate within a relatively hierarchical arrangement, where projects are subject to the whims of high-level managers and concerns of profitability and efficiency. In contrast, the middleware project teams we observed were relatively small and autonomous. Rather than being told what to work on, they depended on making successful proposals to grant agencies or other sources of funding, and metrics of success had more to do with providing novel solutions (or enabling transformative science) than profit. Given the importance of cyberinfrastructure in many areas of science, and the key role of middleware in enabling that cyberinfrastructure, understanding how development proceeds in this area will have important implications for scientific policy makers, developers, and other stakeholders.

We next introduce the research sites and methods of our study before presenting our findings on the middleware development process.

RESEARCH SITES & METHODS

Research that studies cyberinfrastructure development typically does so by studying particular domain-specific research and development projects. Our study is distinct from these approaches as we focus on how cyberinfrastructure middleware development crosses and transcends cyberinfrastructure domain project collaborations. We investigated projects and the groups developing them at two of the United States' most established supercomputing organizations, the National Center for Supercomputing Applications (NCSA) at the University of Illinois Urbana-Champaign and the San Diego Supercomputer Center (SDSC) at the University of California, San Diego.

Each center was founded in the mid-1980s when the US National Science Foundation established five national supercomputing centers to provide the nation's scientists with access to high-performance computing resources [5]. While originally funded through large facilities grants, today most development and research activities at these two centers are funded through project-level competitive grants. With their focus on scientific computing, the centers have been vital hubs for the development of cyberinfrastructure. Studying the middleware development activities at these centers affords the opportunity to discover commonalities between the two institutions that may be common to computing organizations throughout the United States, and around the world. Additionally the knowledge gained from studying the work to build these middleware projects may be transferable to researchers developing systems at smaller scales and in different research contexts so as to enhance and support their development activities.

Both SDSC and NCSA work with multiple scientific communities coming from various academic disciplines. Cyberinfrastructure developers at NCSA and SDSC are more often than not involved with more than one project at a given point in time. These projects range in size from one or two people up to the hundreds of people who are involved in projects like TeraGrid (www.teragrid.org) and its successor XSEDE (www.xsede.org). In addition to varying in size, projects also vary in the distribution of members, from entirely localized at a single center to those with members distributed around the United States and elsewhere in the world.

We conducted a qualitative, interview-based study with two rounds of semi-structured interviews. The first round of interviews was designed to elicit an understanding of the roles of each interviewee along with the projects that they work on and how CI development takes place in the organization that they work in. The second round was designed to provide depth to our understanding of the design and development practices of an individual's CI projects. The first round of interviews took place in the Spring and Summer of 2009 and the second in the Summer of 2011 after multiple rounds of data analysis were completed. We interviewed 20 participants at NCSA and 12 participants at SDSC during the first round of data collection, for a total of 32 participants. The second round of interviews included 8 participants at NCSA alone based upon access to our research sites. Of the 20 participants in the first round at NCSA and the 8 in the second round, three were interviewed both times providing elaboration and triangulation of our initial interviews.

The 37 individual participants in this study included upper-level management and directors (10 participants), software engineers and project developers (15 participants), and domain scientists and humanities scholars (12 participants) who work with NCSA or SDSC on cyberinfrastructure development. The participants in our study work on software rather than hardware development. We attempted to obtain a breadth of roles in the cyberinfrastructure development process who could convey perspectives from different positions in the organization. While our sampling focused on interviewing a breadth of the organization the interviews themselves resulted in discussion of several common projects. This provides us with multiple perspectives on projects that cut across the individuals involved in the development process. Many of the participants in our study act as principal investigators on funded projects enabling us to obtain data regarding this important aspect of project work. The first round of interviews were semi-structured, typically scheduled for an hour, and lasted between 15 and 86 minutes depending on participant availability and how the discussion developed (median: 56 min). Those conducted in the second round were again semi-structured and scheduled for an hour and lasted between 20 and 95 minutes (median: 53 min). Participants and projects have been assigned pseudonyms to ensure privacy.

The interviews were semi-structured to ensure consistency across interviews and also to allow for deeper questioning to draw out new insights regarding the projects each participant works on. All interviews were recorded and professionally transcribed for analysis. The second round of interviews was designed to elicit further depth as to the processes and strategies participants use when designing cyberinfrastructure systems. Data were analyzed using a grounded-theory approach. We began with closed coding of the initial interviews based on the semi-structured protocol, whereupon a codebook was iteratively developed and refined based on discussions within the research team. A series of memos were generated to express the thematic relationships the team found and to support further open coding of the data. This analysis process saw the emergence of themes surrounding the building of middleware systems which informed our semi-structured protocol for the second round of interviews.

FINDINGS: MIDDLEWARE Z AND OTHER MIDDLEWARE PROJECTS

In this section, we detail the development of a particular system, Middleware Z (a pseudonym) and other middleware projects, in order to illustrate the work of attempting to sustainably develop middleware for cyberinfrastructure. Based on our analysis of multiple middleware projects at both supercomputing centers, we believe Middleware Z followed a typical development trajectory. We heard about Middleware Z from several different interview participants, allowing us to construct a narrative about this project that covers its entire lifetime. The stages of development that we highlight create a common trajectory across the projects we studied. Of course, not every detail is shared by every project, and the stages and transitions may occur along different time scales. But combining the story of Middleware Z with details from other projects at NCSA and SDSC provides a rich set of examples that both gives a sense of the lifecycle of a middleware project and illuminates our analysis of cross-cutting themes. This narrative illustrates the complexities and challenges inherent in sustaining a middleware development project, one of the key activities participants in our study undertake.

Middleware Z is designed to provide seamless access to multiple database platforms for domain specific applications. Scientists often need to access multiple databases which each might have their own query language and constructs. Middleware Z provides an abstraction layer between the application and the databases that makes it easier to search across multiple databases and aggregate the results. Middleware Z has been developed over approximately the last decade, starting from the initial system conception to the second full version that is in use as part of multiple projects today. The developers of Middleware Z are part of a group that eventually became known as the Cyberinfrastructure Middleware and Applications (CIMA, a pseudonym) group. The software engineers and project managers in CIMA develop tools that integrate distributed computing resources and data sources for scientific purposes in various domains. The CIMA group, like the other groups at the centers in our study, must work with stakeholders in multiple scientific domains to write grants and obtain funding to develop systems. Beyond developing applications for domain projects the CIMA group works to develop middleware that they can use as a common base for any specific domain application they may end up developing. This middleware supports the group's domain applications by providing uniform access to various computational, data storage, and user management services, or by acting as a common platform for end-user portals. The CIMA group builds collaborations with stakeholders in each domain to obtain funding that is tied to that specific domain's research needs. This funding model typically only supports middleware development to the extent that it serves a particular domain application.

The narrative we construct below detailing the development of Middleware Z over time is compiled from our interviews with several participants, including the lead developer (John) and other members of his group. We use the narrative as a way to illuminate particular features of the story that are representative of middleware development in CI centers. In telling the story, **we use typographic distinctions, telling the Middleware Z story in italics and providing analysis, commentary, and other examples from our data in plain text.**

In the next sections we present key moments in the middleware development process that illustrate the overarching trajectory of the work being conducted throughout a project's lifecycle. Specifically, we discuss:

- Conceiving of a Problem and Solution: The Seeds of a Project Idea
- Pursuing Development through Domain Science Collaborations
- Developing a Standalone Project
- Managing Breakdown and Change

These four sections highlight salient phases and transitions that comprise the work of cyberinfrastructure middleware development.

Conceiving of a Problem and Solution: The Seeds of a Project Idea

Like infrastructures themselves, middleware as concepts and functioning systems tend to emerge gradually. They do not appear as fully-formed or obvious solutions to an immediate problem. Instead, a project's birth can often be traced to the emergent awareness of an unmet need or problem and a rough idea for a potential solution. Developers conceive of project ideas when stakeholders in a CI project are not in agreement about either the need or the solution or both. Before the formal naming and development of Middleware Z, a developer identified a problem and unsuccessfully proposed forming a group to solve it.

In the late 1990s, John, a developer and project lead, was working on an information management system and was frustrated by that project's inability to easily handle multiple different types of data. John states that the client for this project wanted to "merge two extremely different kinds of information." At the time, tools to handle such data queries were not readily available. John was part of a group tasked to "buil[d] a whole set of tools for performing distributed searches, so that when you issued a search, it would fan the search out to a bunch of different sub-searches, and then the sub-searches could translate the query into what was necessary to pull it from the different sources and then present a unified view." But John believed that the group was "doing it [the development] wrong." He proposed to the client that the group be funded to develop a new version that would be more capable and stable in the long-term. At this stage however John only had an initial concept for this new type of system, and importantly, nothing was yet implemented. The client felt the "technology [wasn't] really ready to go" and the utility of such a system was not apparent. This led John to "shelve" the idea for the time being.

John's project idea, for what later becomes Middleware Z, grows out of a recognition of a particular, perhaps better, way of meeting an information management need that could be applied to the existing application. We heard similar stories at SDSC. For example, one participant told us about the genesis of a middleware project: "This is something that we started a while ago because we realized that what we were doing for our research group—in dealing with how to move data and how to get it in one spot for people to analyze it…—these things are all coupled in it" (Roland, domain scientist). The technologists working with Roland realized, as John did, that there was a better way to deal with their data problems, and began building a middleware system to link their various archives and tools.

The middleware Roland described could be developed within an already-funded project, but when John went to the client to fund his new concept, he was unable to convince the client of both the viability and the utility of the project. This simple, seemingly mundane example, of an unmet need and modest failed proposal, indicates a fairly common scenario in which the ideas for new projects are conceived. Developers may decide not to pursue the realization of the idea, but they may also decide to find ways to continue pursuing development.

Pursuing Development through Domain Science Collaborations

Cyberinfrastructure developers and researchers alike bemoan that projects tend to be funded on relatively short time scales (from one to five years). The pressure to consistently come up with attractive, fundable projects results in a climate where developers are required to work on multiple projects simultaneously or start new ones. This offers both challenges and opportunities.

New projects can open up opportunities for developers to pursue their own interests in a new context. The middleware developers in our study are able to pursue their development activities in the context of domain-specific projects as a strategy to support their development work. Pursuing development within the context of domain science projects enables relationships to be formed that may be synergistically relied upon to support the transition from a concept into a functioning system.

As the initial information management system project ended, John began working on a different cyberinfrastructure development project in civil engineering. Importantly he was placed in the role of data architect for the entire project. John states that many of the other project members were not as "data-oriented" as himself, and thus from his perspective did not fully grasp the complexity of the data handling task at hand. From his point of view the larger group did not understand the need to standardize data handling which is why the project "wasn't really getting very far." As development work progressed, both CIMA and the domain science stakeholders began to realize that the data issues were in fact more difficult than originally thought. Given this opening, John was able to push to pursue his idea within the context of this project. The larger project team became convinced that the data storage portion of the software system was one of the more difficult problems and needed

more resources, and John was able to take his idea off the shelf and pursue it within the mandate of this new project.

In the case of Middleware Z, John was able to find a way to pursue his idea within an existing cyberinfrastructure project. More often, senior developers within the centers must actively search for collaborations with domain scientists that can support their work. In order to pursue their own R&D agendas, the computer scientists in the supercomputing centers typically must find domain-specific application development projects with which to collaborate. Developers must both align with the current to near-future needs of the funding source and ensure that they can continue their own middleware development activities. In order to identify potential collaborators, the developers worked to maintain an awareness of domain science activities both inside and outside their host universities. They would attend seminars and conferences, follow trends in various domains, and watch the funding calls from various agencies for potential opportunities. Sometimes finding a collaborator seems serendipitous, as in the case of Middleware Z. However, a developer on another middleware project described actively seeking opportunities for collaboration:

"The other thing that I do is I go out and I look for new opportunities to fund applications that can drive our development.... To make [Middleware Q] viable and exciting to researchers in the application domains, I have collaborations with computer science departments, graduate schools, the library information sciences, bioinformatics groups that are developing new data management analysis and visualization techniques that we feed into this development environment.... So I'm looking for applications that can help fund this development and some of this research." (Mick, project lead, NCSA)

These supercomputing centers can also be seen as centers of expertise for e-science projects, and the depth of that expertise also attracts collaborators.

"A lot of e-science projects have a community portal and data management infrastructure, data analysis infrastructure, visualization, whatever is needed. These are the ingredients to build it up. We have the expertise here for all these different pieces…. We have a lot of PIs here who could work with domain scientists…. We're also [providing] the middleware infrastructure with our research groups to put this together" (Irene, project lead, SDSC).

For Irene, collaborating with domain scientists is a crucial driver in her work: "On a given day we have to have the computer science researcher that you're thinking how you can make this better, but it's also thinking about working with scientists, understanding what they want, and then turning them into requirements for what they're doing." Maintaining the right pool of expertise to serve the scientists' requirements helps to build those necessary collaborations.

Despite the opportunities, the strategy of pursuing a longer-term research and development agenda within the context of multiple domain science projects also presents significant challenges. Regardless of the length of a project that is scoped out as part of a collaboration, middleware developers must grapple with a tension between building systems that can be used across multiple projects (the middleware goal) while also ensuring that their software meets the specific requirements of a particular funded project. As we saw, John had recognized that his idea could be adapted to solve some of the data problems for the civil engineers, and that their project could provide a venue for his middleware development agenda. However, aligning the agenda to the specific domain project was difficult, as John explained when discussing an early version of Middleware Z.

Development work proceeded on the first version of what would become Middleware Z. But responding to the needs of the domain scientists resulted in software that "had an absurd amount of complexity for what it did because I had all these requirements that were being dumped on me." Development took place in the context of a domain-specific CI project with requirements drawn from a large and diverse scientific community. While there may have been scientific reasons for the requirements, for John's technological interests, they were "completely arbitrary." Collaborating with scientists was a mixed blessing: while it provided funding and a context to test the ideas, supporting functionality that was very domain specific led to complexity in the design and less focus on pushing the development of transformative technologies. And even though the middleware approach was showing promise, the domain project's funding was time limited and it came to an end.

The scientific needs of the stakeholders in a domain science project can place many changing and varied demands upon the developers building middleware. However, in order to maintain their collaborations, software developers must be proactive about aligning the existing and potential functionality of their middleware with scientific requirements. Ronnie, a project lead working with an astronomy community, described how his group ensured that their development would satisfy the domain scientists: "so, first of all, we set up some science priorities and we do that every year and things that we're going to deliver to the user." However, prioritizing scientific agendas over development agendas can make it difficult to create broadly applicable middleware.

The short-term nature of project funding is also a continual challenge for middleware developers. Rex, a senior research programmer in the CIMA group, explained that for one of the main projects that he works on they are funded on a yearly basis with no guarantee that they will receive continued funding the next year. To keep the money flowing, they work hard to keep the funder happy and

deliver systems as close to what was promised in the grant proposal. By building a reputation as a group that delivers what they promise, they increase their opportunities for future projects.

With development activities typically arranged around a specific, short-term application context it can be difficult to enlist the resources for a long-term middleware vision. As was the case with Middleware Z, having to emphasize the needs of the immediate domain project may not leave room in the near term to focus on the middleware development. On the other hand, it does support working towards tangible artifacts that may be leveraged in future projects to obtain support for the work. For John, the opportunity to work on his own research agenda inside of the civil engineering project set up the possibility of future funding. Even though the concept became overly complex and did not play out as he envisioned in this project, the incomplete version 1 of Middleware Z offered a functioning example of the concept that he could carry forward into the next project. This use in a deployed system allowed John to illustrate the utility of the concept behind this data middleware and to argue that he was well-prepared for the next project.

Developing a Standalone Project

Once a middleware has been developed within the context of a domain science project, and if it shows promise as a system applicable to multiple projects such that enough collaborators, colleagues, and funders can be convinced, then a middleware may be able to become a standalone project. Middleware Z until this point had only existed as a piece of other cyberinfrastructure projects for specific domain uses. Based on the work that occurred in the context of a civil engineering CI, Middleware Z's next developmental point was to become its own project. This transition enabled Middleware Z to obtain an identity that is recognizable as a singular system outside of the context of any one domain-specific project.

John wanted to continue development of Middleware Z, but new funding sources would be required. John told us, "The result of that whole [civil engineering] process was kind of a half-baked version of [Middleware Z]. I sustained the funding for it. I called it [Middleware Z] after I left - after the [civil engineering] project ended.... It was just the data component of [the civil engineering project], and that provided the entire funding for the development of that first rev.... Then, after the [civil engineering] project ended, I had some support through [a grid computing] project.... I put [Middleware Z] in there as a data management component, but it was really dis-integrated from the rest of the package. It was a strange relationship, where I just sort of hung around with them, and helped them make sure my stuff would install correctly... but it provided bridge funding."

An important shift in middleware development occurs when the middleware develops its own identity. In our investigations, the seed idea for middleware typically was born from an attempt to solve a problem in a particular cyberinfrastructure project. Then, in part because developers, either individually or as a group, are always involved with multiple different domains, there is an emerging realization that the same approach could be useful across a range of domain contexts.

At SDSC, Irene told us about a middleware project that gained shape when several projects realized they were dealing with the same issue: "[The projects] were basically trying to do a similar thing…. We sort of put the streams together so we could learn from each other." At first, they began developing Middleware W as an informal package of shared software that was shared across several domain projects, but there was a need to create a more sustainable solution.

"MWC (Middleware W Central) is a specific project we put together… to lead Middleware W into something more shareable, more robust. Until MWC, Middleware W didn't have that driving force. Projects were taking a lot of the – [ProjectX] mainly was a big, big driving core for Middleware W until the end of the project. Projects end, and Middleware W didn't have this community self-sustainable structure. MWC is the project to bring it in and provide an infrastructure for Middleware W."

It was time for the next step, to move development out of the context of specific applications and into its own project. We saw this same progression—from a domain-specific solution to a stand-alone cross-domain middleware—in projects that ranged across data processing workflows, image processing, pattern analysis, and portal systems. Returning to Middleware Z, when our story began, there was no middleware. Instead, what became Middleware Z began as a simple idea for how to solve a particular problem of merging data - an idea that was not compelling enough to pursue. During development for the civil engineering project, it was just the "data component." Now, Middleware Z has a name (or, in our story, a pseudonym). Now it is possible for Middleware Z to be "dis-integrated" from the domain projects that fund it. While usefulness in particular contexts will remain a criteria for continued funding, developing this stand-alone identity makes it possible to consider Middleware Z as a project in itself.

As the grid project funding ended the CIMA group obtained a technology research "umbrella" grant. Timothy, the project manager of CIMA, told us that this umbrella grant funded John's group to "actually design and develop new capabilities" for Middleware Z so that it could be used by projects being built for multiple communities. John was now able to lead a small team focused just on developing Middleware Z. The umbrella grant also removed the constraints placed on the software by the original civil engineering project and the grid project. John states that Middleware Z "didn't need 50% of the features that [version 1]" had. The Middleware Z team could "really push it to the point of maturity to where it became useful."

John said, "I can't quite remember the chronology, but at some point, I decided to completely rewrite Middleware Z from the ground up." Over the next two to three years, Middleware Z became an important item in CIMA's cyberinfrastructure toolbox. Middleware Z was incorporated into several projects in multiple domains including education, environmental research, and the humanities.

Middleware Z found funding as a standalone project through an umbrella grant, but other middleware projects were able to make this transition using other models. For example, one workflow middleware project fostered an active open-source community to support the development, while another project spun off their middleware and founded a commercial start-up to take over the project. Regardless of the model, however, this progression from local problem solution to widely applicable middleware was a key transition in all of the projects.

Managing Breakdown and Change

Middleware development does not follow a purely linear trajectory, however, and any infrastructural middleware will be faced with moments of change and breakdown. Middleware Z became a popular tool inside the center where it was developed. It had been incorporated into a number of other systems developed by the CIMA group. However, Middleware Z had not gained traction in a wider community and there were changes on the horizon which led to support for further development to wane. This final point in the middleware development process in our data highlights the relational nature of CI and the constant synergistic activities that the developers behind a system must undertake throughout the development of a cyberinfrastructure middleware system.

When the umbrella grant funding ended Middleware Z had been through its second major re-write and was at a stable development point. John decided to leave the center for a position at another research institution. Before he left, the team conducted a code review with him ensuring that they had an overall grasp of the middleware system's design and implementation. The group also performed an internal analysis that revealed that CIMA's projects would soon run up against Middleware Z's "performance wall" with larger data sets. Due to this performance wall the group sought an "exit strategy" that would allow them to support existing projects while finding a solution for larger datasets. Timothy, the CIMA group manager now responsible for Middleware Z, explained that the CIMA group as a whole could maintain Middleware Z since they "have got enough experience on the team [with Middleware Z] that if [they] have to debug something, [they] will debug it." Timothy, Jerry, and the developers in the group decided to lock down a subset of Middleware Z's features as "Middleware Z Light." The Light version would strip out all but the essential and stable functions, since "when we were creating this, we were experimenting with all these different back ends to see which ones would behave best, you know, give us different characteristics and such" (Timothy). Any experimenting would cease, and Middleware Z would not be further developed. Instead of continuing development, the CIMA group began to evaluate open source middleware projects that could perform similar functions. "The reality is, [Middleware A] and [Middleware B] have become de facto middleware data store environments" (Timothy), and they decided to adopt one of those platforms rather than continue Middleware Z development.

Middleware Z had reached a turning point: would it become infrastructure or not? Infrastructure is a relational concept [24]. Inside this center, Middleware Z had become infrastructural. Many of their domain-specific projects were using Middleware Z as an underlying, black-boxed technology. But faced with two forms of impending infrastructural breakdown—John leaving the center and the performance limits of the software—the CIMA group was forced to reconsider the role that Middleware Z played in their activities. Change is not uncommon in these middleware projects; people come and go, funding streams end, organizations are restructured. Sometimes, as in the case of Middleware Z, these changes may lead to the end of the project. For other projects, this provides a chance to restructure the project or go in a new direction (as they did earlier in the Middleware Z project when the developers were able to do a complete rewrite of the software). These moments provide new opportunities for heedful synergizing by creating new infrastructural alignments or reinforcing those that are already in place.

The synergizing lens frames the work of infrastructure as a process of developing and managing infrastructural relationships among people, technologies, and institutions, and is useful for understanding how infrastructures deal with and manage change. [3] finds that "A change in one component of the infrastructure cannot be understood without understanding the way that the component is connected to others." In this case, Middleware Z has infrastructural relationships with a number of other middleware projects and domain-specific applications. In order to sustain the larger infrastructure that has been built up at NCSA, it would be necessary to either fix the breakdowns (for example, by assigning a new development lead and re-architecting Middleware Z to avoid its performance limitations), or find another way to provide the required functionality. In the end, the decision was made to turn to external middleware efforts to meet the same infrastructural needs. However, in order to manage the pace of that change and smooth the transition, the group decided to maintain the "light" version of Middleware Z until the replacement components were ready.

Overall the moments of articulation work that groups at NCSA and SDSC enact enable them to sustain work on many projects for specific domains while working to produce middleware that can be applied in many situations.

We find that middleware development typically begins in response to an emergent need within a domain science project. As developers work with different scientist collaborators, if they face similar needs in multiple contexts they tend to reuse and expand on what they have already developed and the middleware takes shape and grows. Eventually, if the need is broad enough, the middleware can become a project in itself and begin to stand alone from the particular contexts from which it emerged. Throughout the life of these projects, developers work collaboratively and must continually align their own interests with domain science needs. The growth of Middleware Z and the other examples provided here—from initial ideas to a functioning middleware leveraged as a key components of multiple domain science projects—demonstrates the work cyberinfrastructure developers must undertake to accomplish infrastructure.

DISCUSSION

Based on the findings presented we now discuss three key themes that emphasize the synergizing work necessary to support cyberinfrastructure middleware development in the supercomputing centers in our study. The first is that of the importance of research & development agendas to the work of building these systems. The second theme is the relationship among the multiple projects that provide the context for development over time. Finally, the third theme is that of recognizing that sustaining middleware development requires the marshaling of multiple stakeholders' interests.

R&D Agendas in Cyberinfrastructure Development

In all of the middleware projects in our study, it was clear that there was an individual project leader who played a key role in sustaining the middleware development. John is a dominant figure in the story of Middleware Z because John and his research interests were such an important part of its trajectory. Of course, John was working with other people in the various scientific projects, and then led a small team when the middleware was funded as a standalone project. What we see as important, however, is the role of people like John in sustaining the middleware across the gaps and transitions between funded projects.

Others have pointed out the importance of good leadership in cyberinfrastructure projects [23], but we believe there are structural properties to middleware development that go beyond having a strong or charismatic leader. In particular, what is important is that the developer has his or her own research and development agenda. This agenda is not, at least initially, tied to a particular system or approach. Instead, the agenda revolves around solving a particular class of problems, as in the case of Middleware Z, where John was interested in the challenges of data management when working with large and heterogeneous datasets. Prior research argues that cyberinfrastructure developers, like the domain scientists with whom they collaborate, have their own research agendas and face many of the same challenges related to pursuing a research agenda in the face of short-term grant-funded projects [13].

Our point here is similar but slightly different. In the stories we heard from our participants, a developer (or development group) with an agenda was *necessary* to shepherd the concept and prototypes through the multiple phases and moments of alignment necessary to realize a functioning middleware. With each of the middleware projects we encountered, there was an individual or small group of individuals who, at least in the early phases of the project, served as champions for the systems. These centers are working on cutting edge technologies, and most (although not all) of the people we spoke to leading development projects considered themselves technology researchers as much as software developers, and they modeled their groups on academic research laboratories. These individuals take responsibility for the necessary synergizing work to enable sustainable software. They identify the concepts in their current project that could have broader applicability. They advocate for their vision, and they gather the necessary resources and foster collaborations to develop prototypes. As the middleware matures, they work with other projects to convince them to use the middleware and help them incorporate it into their systems, all the while continuing development work to support additional use cases or make the middleware more stable. They monitor the system, anticipating breakdowns and repairing them when they occur. Without an interested party to carry the concept through prototypes and into a working system, middleware would not happen.

The Relationships Among Projects

The work of middleware infrastructure development cannot be understood without looking at the multiple and complex relationships among projects. Middleware is thought of as a technological component that sits in the 'middle' of and links other components together. If we take a sociotechnical approach, we see that it is not just a matter of interoperable software, but that there is a meshing of work practices that must also occur simultaneously. This work requires collaboration among developers, and between developers and domain scientists. Schmidt and Simone [22] point out that collaboration takes place within a "common field of work." The common field of work is the set of things that the cooperating individuals are working on, in other words, "the collective of things upon which an ensemble is enacting state changes" [4]. However, our data suggests that middleware development occurs in a complex, multiple, and sometimes ill-defined project space.

Consider the story of Middleware Z. It started out as an idea in an information management project, but did not become a prototype until it was picked up in a second (civil engineering) project. Then it had a "strange relationship" with the grid computing project before getting funding from the umbrella grant. As Middleware Z matured, it was adopted by many other projects within the CIMA group,

requiring significant internal collaboration among the projects. The diversity of collaborations was a factor in all of the middleware projects, and all of the developers in our study were involved in developing components of multiple different domain projects at any one time. Rather than having a single common field of work, middleware developers are always participating simultaneously in multiple overlapping fields of work. That developers are working in multiple fields of work at once is not particularly new or remarkable in itself. What is important, however, is how the fields of work shift and blend over time.

Within a common field of work, cooperation requires that workers "articulate (divide, allocate coordinate, schedule, mesh, interrelate, etc.) their distributed individual activities" [21]. However, much of the work of middleware development goes beyond this sense of articulation work to involve defining the fields of work themselves and managing the relationships among fields of work. Building and managing these links across projects and infrastructural components draws more on the processes that Bietz, Lee and Baumer have called "synergizing" [4].

The point at which a middleware goes from existing only within other fields of work to being able to define its own field of work is a key transition. For example, in the case of Middleware Z, it was not until after John consolidated his work on the concept and named it "Middleware Z" that others viewed it as a specific project and area of focus. When middleware is defined as a project in itself, it becomes a target of a significant amount of synergizing work. The middleware is now a distinct entity, leverageable by other projects since it is a visible, known system. The alignment work which takes place to integrate middleware into other projects further strengthens its position in the larger infrastructure as other projects come to rely on its services.

The Sustainability of Infrastructure

Sustaining infrastructure requires an understanding of the organizational, individual, and technological aspects together. Middleware development occurs within cyberinfrastructure, domain science, and organizational contexts that shape long-term development trajectories. In the Middleware Z story, and all of the stories we heard, middleware development relies on successful marshaling of R&D agendas, domain science needs, and institutional interests.

Once a system has reached a point such that it is embedded in multiple projects it reaches a point that it is viewed as infrastructural middleware, at least from the perspective of the group relying upon it in multiple projects. The individual developers and the organizational group that they are a part of are able to make the case that supporting the continued development of the middleware is crucial to the needs of the many domain projects that it has been embedded within. Moments that may have major impacts on the sustainability of a middleware project include when the intellectual owner leaves for a new position and does not take the system with them or when it is determined that a performance wall has been reached. It is at points like these in a project's development that the opportunity to pull back and examine opportunities for alignment with other outside development projects arise. It may be the case that continuing to develop the internally developed middleware may no longer be the most sensible approach.

IMPLICATIONS

Cyberinfrastructures are complex sociotechnical collaborative systems that are comprised of and embedded in other systems. As the corpus of CSCW research in the area of cyberinfrastructure continues to grow in breadth and depth, CSCW researchers, without losing a necessary view of larger systems, are able to focus on key elements of cyberinfrastructure development such as the particular relationship and interaction between middleware developers, middleware, and larger organizations such as funded projects and supercomputing centers.

Implications for Middleware Development Organizations

Our data suggest that the ideas that eventually grew into fully fledged middleware systems all began as solutions to specific challenges that arose directly out of the scientific needs. However, we saw that the crucial transition from a system-specific component into a fully capable middleware was poorly supported. There is a need to envision new sustainability models that allow for earlier identification of candidate middleware systems and ease the transition from research prototype to sustainable system. Organizations can create procedures that help developers to detect when similar needs arise in multiple domains. For example, periodic cross-project reviews of development work and new technologies would allow for awareness of emergent trends. Also, while we recognize the constraints of current academic funding models, finding ways to give middleware developers the time and flexibility to pursue their own R&D interests would be productive. One developer we interviewed stated that she felt that she had to account for every working hour to the projects that funded her. However, allowing for some individually-directed work time (perhaps similar to Google's "twenty-percent time" [10]) could create space for innovation and encourage developers to take a seed concept from a domain-specific project and begin to develop it into its own distinct middleware system at an earlier stage in its lifecycle.

Implications for National Funding Policy

Our findings also suggest that there are important policy concerns around providing better support for the R&D agendas and careers of the individuals who lead cyberinfrastructure development efforts. Middleware is clearly an essential component of cyberinfrastructure. To achieve the stated goal of developing systems and infrastructures that can provide a broad array of services to scientists in multiple domains [2], it may be necessary to implement funding structures that can directly fund

middleware development and provide greater stability and support for the long-term R&D agendas of middleware developers. The US National Science Foundation's (NSF) *Software Infrastructure for Sustained Innovation (SI^2)* program is a new program with the goal of transforming innovations in research and education into sustained software resources that are an integral part of the cyberinfrastructure [16]. This program is a good first step, but the program is currently limited to chemical science collaborations bridging the US and UK. A broader program is needed.

Implications for CSCW

Our research finds that the developers of middleware do not stand outside of infrastructure, but are integrally embedded in it, which is key to sustaining these projects. Lee et al. [14] refers to this as the "human infrastructure of cyberinfrastructure." We find that individual technology researchers pursuing their own career-level R&D agendas provide important continuity of expertise as well as a driving force that helps make the transition from domain-specific solutions to generalizable middleware systems. Thinking about enabling individual careers and building stable organizations must be as much of the discussion as creating a flexible and long lasting technological artifact.

Understanding how elements in the technical and human infrastructure interact over time is a step towards creating an increasingly sophisticated conceptual model that provides insight into two layers of sociotechnical design: *supporting collaborative development* of cyberinfrastructures and *developing cyberinfrastructure systems themselves* that support collaboration. Future work will further explore and qualitatively model how these layers interact and co-evolve and how cyberinfrastructures are created and maintained by multiple sociotechnical elements that are by turns stable and dynamic.

CONCLUSION

This paper is part of a growing corpus of work that is taking seriously the notion that cyberinfrastructures are relational, sociotechnical systems that are comprised of and embedded in other systems. The development of cyberinfrastructures not only necessarily spans geographic space, but also communities that have differing priorities and ways of working, and—importantly—time. Designing something as complex as a collaborative cyberinfrastructure is itself a complex collaborative activity that resembles a program that is more akin to a public works development project with a myriad of stakeholders and a changing political-economic environment.

This paper also highlights the importance of the everyday, seemingly mundane work, of middleware developers to cultivate and maintain infrastructural relationships over time. We addressed the question of "how do developers sustain middleware projects in the context of cyberinfrastructure development?" Building on the data from a qualitative study of middleware development at two supercomputing centers, we tell the story of the development of a representative project, Middleware Z, enriched with examples from other projects. Focusing on key transitions—from an early concept to a functional prototype to working infrastructure to managing breakdown and change—reveals the work required to nurture and sustain a middleware project through its different stages of development. Infrastructural middleware development requires not just understanding how to build sustainable technologies, but also how to build sustainable human and technological assemblages.

ACKNOWLEDGMENTS

We wish to thank the participants in this research for their time and insights, and the anonymous reviewers for their feedback. This work was supported by National Science Foundation Awards IIS-0712994, IIS-0954088 and OCI-0838601.

REFERENCES

1. Aronova, E., Baker, K. S., & Oreskes, N. Big science and big data in biology: From the International Geophysical Year through the International Biological Program to the Long Term Ecological Research (LTER) Network, 1957-present. *Historical Studies in the Natural Sciences, 40*, 2 (2010), 183-224.
2. Atkins, D. E., Droegemeier, K. K., Feldman, S. I., Garcia-Molina, H., Klein, M. L., et al. *Revolutionizing science and engineering through cyberinfrastructure: Report of the National Science Foundation blue-ribbon advisory panel on cyberinfrastructure*. National Science Foundation, Washington, D.C. (2003).
3. Bietz, M. J., Ferro, T., & Lee, C. P. Sustaining the development of cyberinfrastructure: An organization adapting to change. In Proc. *CSCW 2012*, ACM Press (2012), 901-910.
4. Bietz, M. J., Lee, C. P., & Baumer, E. P. S. Synergizing in cyberinfrastructure development. *Computer Supported Cooperative Work, 19*, 3-4 (2010), 245-281.
5. Branscomb, L., Belytschko, T., Bridenbaugh, P., Chay, T., Dozier, J., et al. *From Desktop To Teraflop: Exploiting the U.S. Lead in High Performance Computing*. National Science Foundation, Washington, D.C. (1993).
6. Edwards, P. N., Jackson, S. J., Bowker, G. C., & Knobel, C. P. *Understanding infrastructure: Dynamics, tensions, and design*. http://hdl.handle.net/2027.42/49353
7. Finholt, T. A. Collaboratories. In *Annual Review of Information Science and Technology*. B. Cronin, Ed., (Vol. 36). Information Today Publishers, Medford, NJ (2002), 73-107.
8. Foster, I., & Kesselman, C. *The Grid: Blueprint for a New Computing Infrastructure* (second ed.). Morgan Kaufmann, Boston (2004).

9. Galison, P. *Big Science: The Growth of Large-Scale Research*. Stanford University Press, Stanford, CA (1992).
10. Iyer, B., & Davenport, T. H. Reverse engineering Google's innovation machine. *Harvard Business Review, 86*, 4 (2008), 58-68.
11. Jirotka, M., Proctor, R., Rodden, T., & Bowker, G. C. Special Issue: Collaboration in e-Research. *Computer Supported Cooperative Work, 15*, 4 (2006), 251-255.
12. Karasti, H., & Baker, K. S. Infrastructuring for the long-term: Ecological information management. In *Proc. HICSS 2004*, IEEE Computer Society (2004), 10020c.
13. Lee, C. P., Bietz, M. J., Derthick, K., & Paine, D. A sociotechnical exploration of infrastructural middleware development. In Proc. *CSCW 2012,* ACM Press (2012), 1347-1350.
14. Lee, C. P., Dourish, P., & Mark, G. The human infrastructure of cyberinfrastructure. In *Proc. CSCW 2006*. ACM Press (2006), 483-492.
15. Monteiro, E., Pollock, N., Hanseth, O., & Williams, R. From artefacts to infrastructures. *Computer Supported Cooperative Work (CSCW)* (2012), 1-33. DOI=10.1007/s10606-012-9167-1.
16. National Science Foundation. (2012). *Software infrastructure for sustained innovation (SI2)*. http://www.nsf.gov/pubs/2012/nsf12576/nsf12576.htm.
17. Naur, P., & Randell, B. Software Engineering Report of a conference sponsored by the NATO Science Committee Garmisch Germany 7th-11th October 1968: Scientific Affairs Division, NATO (1969).
18. Newman, S. E. Here, there, and nowhere at all: Distribution, negotiation, and virtuality in postmodern ethnography and engineering. *Knowledge and Society, 11* (1998), 235-267.
19. Ribes, D., & Finholt, T. A. The long now of technology infrastructure: Articulating tensions in development. *Journal of the Association for Information Systems, 10* (2009), 375-398.
20. Ribes, D., & Lee, C. Sociotechnical studies of cyberinfrastructure and e-research: Current themes and future trajectories. *Computer Supported Cooperative Work (CSCW), 19*, 3 (2010), 231-244.
21. Schmidt, K., & Bannon, L. Taking CSCW seriously: Supporting articulation work. *Computer Supported Cooperative Work, 1*, 1 (1992), 7-40.
22. Schmidt, K., & Simone, C. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *Computer Supported Cooperative Work (CSCW), 5*, 2 (1996), 155-200.
23. Spencer, B. F., Jr., Butler, R., Ricker, K., Marcusiu, D., Finholt, T. A., et al. NEESgrid: Lessions learned for future cyberinfrastructure development. In *Scientific Collaboration on the Internet*. G. M. Olson & A. Zimmerman & N. Bos, Eds. MIT Press, Cambridge, MA (2008), 331-347.
24. Star, S. L., & Ruhleder, K. Steps toward an ecology of infrastructure: Design and access for large information spaces. *Information Systems Research, 7*, 1 (1996), 111-134.