Challenges of Scientist-Developers and Adopters of Existing Cyberinfrastructure Tools for Data-Intensive Collaboration, Computational Simulation, and Interdisciplinary Projects in Early e-Science in the U.S.

Kerk F. Kee
Chapman University
One University Drive, Orange, CA 92866
kerk.kee@gmail.com

Larry D. Browning
University of Texas, Austin
2504A Whitis Ave., Austin, TX 78712
lbrowning@mail.utexas.edu

ABSTRACT
While some scientists engage in co-production of CI tools with technologists for data intensive collaboration, computational simulation, and interdisciplinary projects, this position paper focuses on two other groups of users: (a) scientists who develop their own CI tools and (b) scientists who use existing tools developed by pioneering scientists for large-scale research activities. Based on qualitative methods guided by grounded theory and 70 interviews, we identify that both of these groups experience challenges that stemmed from the meta and complexity natures of cyberinfrastructure and infrastructures for big data. Scientist-developers (or self-developers) face two challenges in adopting the meta/complex infrastructures for big data through developing their own tools for data-intensive collaboration and computational simulation. First, they have to know the nitty gritty details of CI hardware. Second, they need to become highly skilled in computational science as master programmers. On the other hand, scientists who simply adopt existing tools face two challenges as a result of the meta/complex natures of cyberinfrastructure. First, existing tools often have low usability. Second, because CI projects are funded on short-term grants, their long-term availability/sustainability is uncertain. We argue that in order to promote effective large-scale and data-intensive collaboration in science and engineering, these socio-technical challenges need to be resolved.

Author Keywords
Cyberinfrastructure; e-science; meta and complexity of infrastructures; socio-technical challenges.

ACM Classification Keywords
K.4.3 [Organizational Impacts]—Computer-supported collaborative work, H.5.3 [Group and Organization Interfaces]—Organizational design, Computer-supported cooperative work.

CSCW'12, February 11–15, 2012, Seattle, Washington, USA.

INTRODUCTION
The grand vision of cyberinfrastructure (CI) is to enable large-scale research and data-intensive collaboration using aggregated computational resources and combined datasets through the Internet, high-performance networks, and local machines and to be able to mine publicly-funded datasets accumulated over time. Data intensiveness is a key defining characteristic of CI and e-science projects [2]. When the grand vision described is achieved, there will be increased productivity and breakthrough discoveries in research. However, like most innovations, an ambitious endeavor such as CI adoption and implementation for data-intensive collaboration, computational simulation, and interdisciplinary projects often encounters challenges. While some scientists engage in co-production of CI tools with computational technologists [the focus of another paper submitted to *W12: Mastering Data-Intensive Collaboration Through the Synergy of Human and Machine Reasoning*], this paper focuses on two other groups of users: (a) scientists who develop their own CI tools, and (b) scientists who use existing tools developed by pioneering scientists for similar large-scale research activities. However, both of these groups experience challenges that stemmed from the meta and complexity natures of cyberinfrastructure and infrastructures for big data.

THE META AND COMPLEXITY NATURES OF CYBERINFRASTRUCTURE

As previously sated, this paper examines CI adoption for data-intensive collaboration, computational simulation, and interdisciplinary projects among pioneering scientists who do not directly co-produce CI tools with computational technologists. For scientists who develop their own CI tools for large-scale research activities, they also must possess advanced knowledge and skills for a range of hardware and software related to supercomputing in addition to the computer, software, and scientific instruments in their own labs. A technologist in Indiana explains, "You need to be a

software developer… have domain knowledge… know how to take advantage of the middleware… that actually implements the middleware part of the cyberinfrastructure. It's pulling together knowledge from a lot of different areas". Furthermore, for scientists who use existing tools, they must understand the tools developed, the datasets collected, and the measurements used by the creators, as well as the scientific instruments employed. Because of all these criteria, Kee and colleagues maintain that infrastructures for big data appear as a meta and complex innovation in the eyes of scientists as users [5].

DATA COLLECTION AND METHOD

We conducted interviews over a period of 32 months, from November 2007 to June 2010. The data set includes 70 interviews with 66 participants from across 17 U.S. states and three other countries. The interviews were spread across four years with 10 participants in 2007, 42 in 2008, 16 in 2009, and two in 2010. Because most of the interviews were conducted in 2008, the analysis primarily reflects CI implementation for data-intensive collaboration, computational simulation, and interdisciplinary projects during this period. The shortest interview was 15 minutes and the longest was 2 hours and 16 minutes. The interviews averaged approximately one hour each and were conducted in person with 19 of the participants and over the phone with the remaining 51 participants. All the interviews were audio recorded except for two, due to technical difficulty and following one participant's request. However, notes were taken immediately after these two interviews.

The 66 interview participants came from Texas (12), Illinois (11), California (10), Michigan (5), Indiana (4), Massachusetts (3), Arizona (2), Colorado (2), Louisiana (2), Washington (2), DC (1), Maryland (1), New York (1), Virginia (1), Ohio (1), Pennsylvania (1), Delaware (1), as well as Australia (2), Germany (1), and the UK (1). The geographic affiliations refer to the primary locations of the participants at the time of the interviews. Participants include 52 males and 14 females. Participants' primary professional roles were diverse, including domain scientists who used CI to conduct science (15), computational technologists who built CI (12), a range of administrative directors and program managers at supercomputer centers and national research laboratories across the country (21), US NSF program officers who helped allocate funding to CI projects (4), social scientists and policy analysts who studied and participated in CI projects (12), and experts from commercial industry (2).

This paper employs grounded theory in qualitative data collection and analysis. According to Corbin and Strauss, one of the main goals of grounded theory is to seek to uncover relevant conditions under which social phenomena manifest [3]. We argue in this paper that the relevant conditions for adoption challenges experienced by the two groups of scientists are the meta and complexity natures of cyberinfrastructure and infrastructure for big data discussed earlier. The next section unpacks these challenges in details.

SCIENTIST-DEVELOPERS

The first group of scientists, those who developed their own CI tools, are referred to as 'scientist-developers' in this paper. The notion of a scientist-developer has an emphasis on the role of a developer like a technologist. However, they are developers of their own tools. They can be faculty scientists, post-docs, or graduate students working on CI and e-science projects. The faculty scientists often were trained by advisors who did data and computationally intensive research. The graduate students and post-docs acquire the knowledge and skills necessary for developing CI and computational tools by attending training workshops. Sometimes they consult with technologists at supercomputer centers if they have access to them. In this section, the analysis reveals the challenges scientist-developers encounter in the adoption and development process as cyberinfrastructure is a meta/complex innovation.

The practice of self-developing CI tools for data-intensive collaboration has several limitations for faculty scientists. Ideally, faculty scientists would receive big grants to co-produce CI tools with technologists, as alluded to earlier. Frequently in reality, their graduate students and post-docs are the actual scientist-developers due to the limited funding faculty scientists receive. Furthermore, these are funding and learning opportunities for graduate students and post-docs. A chemical engineering professor from Massachusetts shares, "The correct way [is to] have the government pay chemists to hire computer programmers to work with them... But the current size of the NSF grants [is] too small… [So] we're using our chemistry graduate students to write the software".

In order for these graduate students to do this work, faculty scientists often send their graduate students for training to increase their computational science knowledge and skills. Most scientists themselves do not participate in this training. A CI project manager in California shares, "We had a special conference that we convened once a year… to train end-users… We mostly got grad students… We got maybe one or two actual researchers from our community here in the States that were professors". This further shows that graduate students are often the hands-on scientist-developers of CI tools for projects. This project manager also explains that faculty scientists from out of the country participated because they wanted to learn skills and techniques related to the open source software being developed in the U.S. so that they could build on the American effort when they returned to their home countries. Since CI tools are open source, attending the training conference in the U.S. can be the cheapest way to gain technology for their science back home.

Hardware Details

The development of infrastructures for big data is challenged by the first limitation scientist-developers encounter: the need to be knowledgeable about supercomputer hardware details. In order to become a competent scientist-developer, a scientist (or usually a graduate student) has to acquire the advanced knowledge and skills of a professional CI computational technologist. The first skill is the hardware, and a scientist-developer has to understand the technical specifications of the hardware down to the "nitty gritty". A center administrator in Pennsylvania shares, "It's vital … for any code to work right… [you] need to understand the nitty gritty down to a very deep level, ….[you need to] work very closely down to the most intimate details of the hardware". This quote shows that a scientist-developer needs to be as knowledgeable about the hardware as a technologist, down to the finest level of detail.

Master Programmer

The second challenge a scientist-developer encounters is that they also have to become highly skilled at the software aspects of high-performance computing. Not only does a scientist-developer have to know how to write programming codes, he or she has to become a "black belt" or master programmer. In order to utilize the vivid details in the interview data, a few block quotes are provided to illustrate the meta/complex nature of infrastructure for big data during CI adoption and development. An administrator at Indiana shares:

> So to use a supercomputer from a UNIX shell, you've got to know MPI, you've realistically today got to know C, you've got to know… about computer architecture. They are the fastest supercomputers on Earth for a reason. They're built with the newest, most advanced components so in general, they have the least polished interfaces. They're just harder to program. There's no getting around that… If you wanted to be able to use one of these things, it is a reasonable expectation that you are going to be a black belt [MPI] programmer.

Message Passing Interface (MPI) is a standard for writing parallel programs on supercomputers, including mastering processor-to-processor communication routines, monitoring collective operations performed by groups of processors, defining and using high-level processor connection topologies and user-specified derived data types for message creation. Clearly, to become a "black belt" MPI programmer as a scientist-developer is a challenging task. This is on top of understanding the hardware or "computer architecture" as previously discussed, programming language C, etc. These elements make cyberinfrastructure and infrastructures for big data a meta/complex innovation.

The analysis continues to build on the finding that the ability to fully optimize cyberinfrastructure resources requires a very high level of programming skills (the "black belt" skills). Computing programming on a commercial personal computer is a difficult skill to master. It is an even more difficult on a supercomputer for data-intensive collaboration, computational simulation, and interdisciplinary projects. Due to the complexity of supercomputing systems, the difficulty is exponential. If the programming algorithm is not written efficiently by a master programmer, the speed and performance can go down. A big challenge is for scientist-developers to think at the computational scale to program for cyberinfrastructure. A center administrator in Texas explains what is required to program on supercomputing or high-performance computing systems:

> Well, using high-performance computing systems is much more complex than using an individual workstation… You've got to rethink your application. You have to come up with a parallel algorithm that leverages the concurrent power of all the processors that you want to use and it does it efficiently – which is also very difficult. Even if you have a parallel algorithm, it's challenging to implement it efficiently… When it becomes inefficient is if you aren't fully utilizing the memory that comes with all those processors so you do something not very smart, like every processor contains the entire problem in it. Well, then you're not really leveraging all that memory. If your performance doesn't scale, it begins to turn over and that can actually happen. Your speed up can actually start going down if you write a parallel code, but you write it inefficient so there's lots of communications going on. Communications are not computations, so you're performance can actually go down if you write it particularly badly.

For scientist-developers to become familiar with the hardware aspects of supercomputers and to become master programmers for parallel computing for their science, they really have to play two roles or take on a dual identity effectively. They have to remain fully knowledgeable of their own field, which is very difficult given the rapid advancements in science. In addition, they have to take on the second identity of becoming a computational technologist as they acquire the knowledge and skills needed to program efficiently and effectively for their own science. Otherwise, when a programming code is badly written, not only can the performance suffer, but valuable resources will be wasted. This center administrator continues:

> The real barrier is learning enough about high-performance computing in addition to the field that they're in – biology or astronomy or chemistry or whatever – is learning enough about high-performance

> computing to be effective with it, such that it really does provide an advantage for your research. It's not too challenging to write a parallel code that stinks, but you don't get any real research advantage if it's not speeding up your research or solving bigger problems. So you have to write it well to use multiple compute nodes at once… If they aren't efficient and effective, they're kind of wasting a valuable resource.

Scientist-developers (or self-developers) face two challenges in adopting the meta/complex cyberinfrastructure and infrastructures for big data through developing their own tools for data-intensive collaboration, computational simulation, and interdisciplinary projects. First, they have to know the nitty gritty details of CI hardware. Second, they need to become highly skilled in computational science as master programmers. Given these challenges faced by scientist-developers, cyberinfrastructure implementation is hampered.

Partly due to a high degree of temporal flexibility, defined as "the degree of rigidity in time structuring and task completion plans" [1 p. 5], in how academics choose to perform work, scientists are able to expand their work from simply doing discovery research to time-intensive co-production and self-development of CI tools. However, this expansion also leads to an increased workload and perhaps internal/external expectations because there is not a clearly defined boundary on how they should structure their time for work. For those who are not able to attend to such a demand on their time, some simply adopted existing CI tools built by the pioneering scientists who came before them. In the mean time, the time intensively continues to persist in this second group, as Lee and Bietz argue, "[m]ost scientists [are] reluctant to invest more than a very small amount of time to learn to use new technologies unless the benefits [are] substantial and related directly to their research" [6, p. 3].

ADOPTERS OF EXISTING CI TOOLS

In addition to the scientist-technologists who co-produce and self-develop CI tools, an important and growing group of adopters are the scientists who simply use existing tools. A physicist in Indiana states, "Not all scientists want to be technologists or technology developers". Science Gateways are existing CI tools these scientists can simply adopt. According to TeraGrid's website [during the time of data collection],

> A Science Gateway is a community-developed set of tools, applications, and data that is integrated via a portal or a suite of applications, usually in a graphical user interface, that is further customized to meet the needs of a targeted community.

Due to a graphical user interface, Science Gateways adopters do not have to face the challenges of scientist-developers.

Using pre-existing tools makes CI adoption and implementation easier for scientists who are not computationally savvy. However, in order for Science Gateways to exist, there will always be a need for a group of scientists willing to work with technologists to produce new tools. Therefore, while Science Gateways increases the simplicity for infrastructures for big data, there will also be a need for scientists to be computational experts or to work with technologists in co-production. An administrator from Indiana talks about how a Science Gateway can increase CI adoption and implementation, "People who are discipline science experts [can] use computing at scale without first having to become computational experts… [But] there's another group of scientists some place that is made up of computational experts who are making these Gateways". Although existing CI tools exist, adoption and implementation of them for data intensive collaboration in science and engineering is not without challenges.

Low Usability

The first challenge for adoption and implementation of existing tools is low usability. Science Gateways are tools developed by pioneering scientists. Their tools often become public goods due to the open source nature of the tools. In other words, everyone can contribute to and benefit from open source tools. Yet many of the tools lack usability, which can negatively impact CI implementation for data intensive collaboration, computational simulation, and interdisciplinary projects. The cyberinfrastructure vision is to build a system that allows a community of scientists to share data with each other and collaborate virtually. However, the interface with low usability is difficult for scientists to adopt and use because the tools were built within a particular project. A CI project manager in California talks about a CI tool that NSF was trying to get researchers to use to put data into the system, and he recounts how the navigation required multiple screens, "A user had to drill down and navigate multiple screens – might have to click through maybe 5 to 10 screens to get to the screen they wanted to get to… There's more work that needs to be done".

Furthermore, the design of a CI tool reflects the preference and/or workflow of the original scientists (and/or technologists). The outcome is a tool that is complicated for other scientists to adopt. Furthermore, many of these are the scientists who set the standards for data interoperability that guide future CI development and data collection/integration. The interface and design may not make sense for scientists with no experience with the particular domain. This can severely impact the adoption and spread of existing tools. In the same interview, the project manager in California also comments, "The system

was complicated. It had a complicated database schema for somebody who works in that domain as scientific researcher. It would make no sense to somebody who's never worked in that area". As this interview was one of the very first conducted for this project, the usability of this tool might be better today.

Long-Term Availability/Sustainability Concerns

The second challenge adopters of existing tools face is the uncertain long-term availability/sustainability. Kee and Browning explain that CI projects are funded on a short-term basis, and there is limited or no long-term funding specifically for software development on these projects [4]. Because scientists are aware of this pattern of NSF and other science funders, they hesitate to adopt a tool for fear that they may become dependent on it and that then the funding for continuing development might stop or the developing scientist might leave. A physicist in Louisiana shares, "I was trying to understand why the fluid dynamics groups are reluctant to embrace our cyberinfrastructure that we're trying to develop for them. They said – Well, how do we know that it's going to be available in two years?". Scientists are careful about their CI adoption decision, even in the case of adopting an existing tool built by pioneering scientists and technologists. The key concern is long-term availability/sustainability if they become dependent on it, thus reducing its long-term relative advantages.

Similar to adopters in co-production and self-developing software, scientists who simply adopt existing tools face two challenges as a result of the meta/complex natures of cyberinfrastructure. First, existing tools often have low usability. Second, because CI projects are funded on short-term grants, their long-term availability/sustainability is uncertain. A retired senior administrator from California shares, "There is the feeling that cyberinfrastructure is not ready for prime time, would be one way of putting it – the software isn't fully stable and it isn't fully featured and it's hard to use".

CONCLUSION

While some scientists engage in co-production of CI and computational tools with technologists for data-intensive collaboration, computational simulation, and interdisciplinary projects, this paper focuses on two other groups of users: (a) scientists who develop their own CI tools, and (b) scientists who use existing tools developed by pioneering scientists for similar large-scale research activities. However, both of these groups experience challenges that stemmed from the meta and complexity natures of cyberinfrastructure and infrastructures for big data.

Scientist-developers (or self-developers) face two challenges in adopting the meta/complex cyberinfrastructure and infrastructures for big data through developing their own tools for data-intensive collaboration. First, they have to know the nitty gritty details of CI hardware. Second, they need to become highly skilled in computational science as master programmers. On the other hand, scientists who simply adopt existing tools face two challenges as a result of the meta/complex natures of cyberinfrastructure. First, existing tools often have low usability. Second, because CI projects are funded on short-term grants, their long-term sustainability is uncertain.

We argue that while examining data intensive collaboration, computational simulation, and interdisciplinary projects in science and engineering, it is critical to pay attention to the knowledge requirements and funding patterns that affect the usability and sustainability of infrastructures for big data. These key factors can impede the adoption and diffusion of computational approaches and data-intensive collaboration that are believed to be able to lead to big scale scientific discoveries in the future.

ACKNOWLEDGMENTS

We thank all the interview participants for their insights.

REFERENCES

1. Ballard, D. I., & Seibold, D. R. Communication-related organizational structures and work group temporal experiences: The effects of coordination method, technology type, and feedback cycle on members' construals and enactments of time. *Communication Monographs, 71*, 1 (2004), 1-27.
2. Bird, I., Jones, B., & Kee, K. F. The organization and management of grid infrastructures. *Computer, 42*, 1 (2009), 36-46.
3. Corbin, J., & Strauss, A. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology, 13*, 1 (1990), 3-21.
4. Kee, K. F., & Browning, L. D. The dialectical tensions in the funding infrastructure of cyberinfrastructure. *Computer Supported Cooperative Work, 19*, 3-4 (2010), 283-308.
5. Kee, K. F., Cradduck, L., Blodgett, B., & Olwan, R. Cyberinfrastructure inside out: Definitions and influencing forces shaping its emergence, development, and implementation. In D. Araya, Y. Breindl & T. Houghton (Eds.), *Nexus: New intersections in Internet research* (pp. 157-189). New York: Peter Lang. 2011.
6. Lee, C. P., & Bietz, M. *Barriers to the adoption of new collaboration technologies for scientists*. Paper presented at the ACM Conference on Computer-Human Interaction (CHI), 2009.