

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/236132597>

Incomplete by Design and Designing for Incompleteness

Article in *Organization Studies* · April 2008

DOI: 10.1177/0170840607088018

CITATIONS

385

READS

2,593

3 authors:



Raghu Garud

Pennsylvania State University

145 PUBLICATIONS 19,307 CITATIONS

[SEE PROFILE](#)



Sanjay Jain

California State University, Northridge

31 PUBLICATIONS 2,889 CITATIONS

[SEE PROFILE](#)



Philipp Tuertscher

Vrije Universiteit Amsterdam

43 PUBLICATIONS 1,610 CITATIONS

[SEE PROFILE](#)

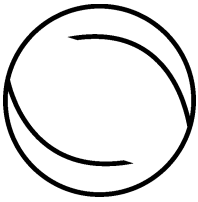
Some of the authors of this publication are also working on these related projects:



The Impact Lab: Co-creating the Design and Implementation of Active Blended Learning [View project](#)



Tackling Grand Challenges Through Open Organizing [View project](#)



Incomplete by Design and Designing for Incompleteness

Raghu Garud, Sanjay Jain and Philipp Tuertscher

Raghu Garud
Pennsylvania State
University, USA

Sanjay Jain
University of
Wisconsin at
Madison, USA

Philipp Tuertscher
Vienna University of
Economics and
Business
Administration,
Austria

Abstract

The traditional scientific approach to design extols the virtues of completeness. However, in environments characterized by continual change, there are challenges in adopting such an approach. We examine Linux and Wikipedia as two exemplary cases to explore the nature of design in such a protean world. Our observations highlight a pragmatic approach to design in which incompleteness is harnessed in a generative manner. This suggests a change in the meaning of the word 'design' itself — from one that separates the process of design from its outcome, to one that considers design as both the medium and outcome of action.

Keywords: platform, emergence, design, innovation community

Historically, much of the discourse on design has extolled the virtues of completeness. Completeness allows for the pre-specification of a problem, the identification of pre-existing alternatives and the choice of the most optimal solution. Such a scientific approach to design pervades much of management thinking, education and research (Romme 2003: 24).¹

For instance, this approach is evident in the design of traditional organizations at the turn of the 20th century. Organizations enhanced the efficiency of their operations by systematically applying principles of scientific management to discover 'the one best way' to organize (Kanigel 1997). Interchangeable parts, division of labor, routinization — each of these were features of an organizational design capable of mass producing 'any color car as long as it was black' (Ford and Crowther 1922: 72).

For such an approach to work, however, there needs to be a clear and stable boundary between the entity being designed and the context for which it is being designed. Such a boundary makes it possible to fix the purpose of a design based on a stable set of user preferences and performance expectations. Clear boundaries, stable preferences and fixed goals: these form the cornerstones of the scientific approach to design as articulated by Simon (1996).

But how does such an approach to design hold up in environments characterized by continual change? What if there are multiple designers, each with their own representation of the problem? What if users of a design are also its designers? To further complicate matters, what if the process of discovering new and potentially better states only takes place through a process of participation, and the unfolding of the process itself changes the problem?

Organization
Studies
29(03): 351–371
ISSN 0170–8406
Copyright © 2008
SAGE Publications
(Los Angeles,
London, New Delhi
and Singapore)

This is the new frontier in which we find ourselves. There is no clear separation between the inside and the outside, text and context. Rather, there is only an evolving and emerging network of associations (Barry and Rerup 2006: 267). Problems are ill-defined, preferences are fluid and solutions emerge in action. In such situations, an emphasis on completeness is likely to result in the creation of designs that foreclose future options.

It is useful to consider the dual meaning of the word 'design' within this context. As a verb, 'to design' refers to the process of developing a plan for a product, structure or component. As a noun, 'a design' is used to connote the outcome of the process.² In traditional settings, these two meanings of design have been separated from one another. One would engage in a process of design (the verb) so as to emerge with a design (the noun) for a specific context. In contemporary settings, however, designs are more appropriately viewed as being simultaneously noun and verb, with every outcome marking the beginning of a new process.³ Put differently, designs are like dynamic jigsaw puzzles in which multiple actors assemble pieces within templates that change as a result of the actors' engagement.

It is this proposition that we develop in the paper. We suggest that, rather than a scientific approach that tends to separate the two meanings of design, we must embrace a pragmatic approach to design that simultaneously embraces both process and outcome. Given this dual connotation, designs, by definition, have to deal with incompleteness. However, rather than pose a threat, incompleteness acts as a trigger for action. Even as actors try to complete what has been left incomplete, they generate new problems as well as new possibilities that continually drive the design. In this way, incompleteness is both a cause and consequence of the dynamics of organizing in contemporary environments.

We begin by providing a brief overview of the scientific approach to design and then highlight the challenges that one confronts in applying this within contemporary environments characterized by continual change. To empirically locate our observations, we examine two exemplary designs that appear to be always in-the-making: the Linux operating system and the Wikipedia online encyclopedia. We find that, rather than one group designing for another's consumption, designs emerge through situated use as actors co-theorize across multiple settings, and, in the process, create new options. These dynamics produce self-perpetuating processes that further drive continual change.

To Design or Not to Design?

In his book, *The Sciences of the Artificial* (1996), Simon suggested that the design of artificial systems — meaning man-made as opposed to natural — is contingent upon the goals of the designer and the purposes for which the system is designed. A key initial task for designers involves the specification of system boundaries. As Simon pointed out,

'An artifact can be thought of as a meeting point — an "interface" in today's terms — between an "inner" environment, the substance and organization of the artifact itself, and an "outer" environment, the surroundings in which it operates. If the inner environment

is appropriate to the outer environment, or vice versa, the artifact will serve its intended purposes.' (Simon 1996: 6)⁴

Once an interface has been specified and the problem has been defined in terms of its context, form and goals, it is possible to proceed using the organizing principles to be found in the natural world. For instance, Simon (1962) offered the principle of 'near decomposability' as a concept possessing clear evolutionary advantages for both natural and artificial systems (see also Langlois 2002). Decomposability refers to the partitioning of a system in such a way that the interactions of elements within a subassembly are greater than the interactions between them. Such decomposition reduces the complexities confronted by boundedly rational human beings in their efforts to design artifacts.

To illustrate this point, Simon offered a parable of two watchmakers, Tempus and Hora. Tempus organized his work in a manner that if he had 'one (watch) partly assembled and had to put it down — to answer the phone, say — it immediately fell to pieces and had to be reassembled from the elements'. Consequently, every time Tempus was interrupted and forced to set aside his work, the entire unfinished assembly fell to pieces. In contrast, Hora first built stable subassemblies that he then put together in a hierarchic fashion into a larger stable assembly. Thus, when Hora was interrupted, only the last unfinished subassembly fell apart, preserving most of his earlier work.

According to Simon, the differential cost of incompleteness that the watchmakers confronted is a specific case of a more general challenge that individuals confront when addressing complex problems. The differential cost can be explained by the interplay between the short- and long-term memories. When individuals address complex problems, transactions are carried out in their short-term memory. Given the limits to short-term memory, any interruption to a task can exact a toll. This is because any intermediate outcome that might have been accomplished before the interruption is lost. While it is possible to store intermediary outcomes in long-term memory, this too exacts a toll as the transfer between long- and short-term memory often requires considerable effort.

Scientific Design in Practice

The concept of decomposability that emerges from this parable has generated a lot of attention from scholars who study modularity (Baldwin and Clark 2000). Specifically, complex systems can be decomposed in a way similar to that suggested by Simon, in other words, into 'modules' (Langlois and Robertson 1992). Each module only interacts with another through standardized interfaces. As a result, each module becomes a 'black box' (Rosenberg 1982), possessing the detail required for its functioning, but hiding such detail from other interdependent modules (Parnas 1972).

An implicit assumption in this literature is that the overall system architecture needs to be completely specified a priori (Baldwin and Clark 2000). In terms of Simon's parable, this means that both Tempus and Hora work with designs that have clearly defined boundaries and pre-set user preferences (accurate time-keeping, for example). In such cases, the key design decisions revolve around issues such as detailing the elements that comprise the architecture,

establishing stable interface specifications to ensure smooth functioning between modules, and 'black-boxing' the modules to mask their complexity.

These decisions are integral to a design approach that values completeness. The form and function of a system must be clearly specified. Only with such a complete representation it is possible to identify clear and stable boundaries between self-contained components that mask much of the complexity. These facets are evident in much of the engineering literature with its focus on detailed definition of requirements (Petroski 1996).

In organizational studies, such a scientific approach to design is most evident in the work of Fredrick Taylor and his colleagues (Kanigel 1997). Once the purpose of a corporation was fixed (for example, to produce widgets with certain predetermined features in as efficient a manner as possible) this approach could be used to identify the 'one best way' to organize. The design process was driven by the need to completely understand and optimize all the cause-and-effect relationships that could influence the outcome of an organization's activities.⁵ Theorizing was done by specific individuals such as industrial engineers but not by those engaged in ongoing operations whose job it was to 'do and not to think'. Decomposability was manifest in division of labor and the presence of a hierarchy, as well as the use of interchangeable parts that were tied together through stable interface specifications within an overall organizational architecture. Costs associated with incompleteness that arose from interruptions were to be minimized at all costs by keeping the assembly line running even if there were defects in the system (Kanigel 1997). The result was the design of the quintessential lean and efficient manufacturing process that could mass produce goods for pre-set user preferences.

Pragmatic Approach to Design

A scientific approach to design — one that requires complete representation of the problem and identifies the optimal solution — is based on the assumption that the environment is stable. For decades, this assumption held. Contexts within which such designs were deployed changed infrequently, if at all. Consequently, boundaries and preferences could be specified and stabilized, and a design with enduring qualities established, to be tweaked as changes in the environment took place.

However, such an approach is likely to run into problems in environments characterized by continual change (Barry and Rerup 2006). In such contexts, system boundaries are often unclear and user preferences are both heterogeneous and evolving. As a result, the goals and purpose of the design are likely to remain a continually moving target (Rindova and Kotha 2001).

The advent of new information technologies has made such fluidity possible. Different material artifacts and social groups can now be easily associated in a dynamic network. Rendering the functionality of any product or service through software enables real-time changes to a design. Indeed, these technologies make it possible for customers to explore their preferences in use and for different social groups to engage with each other in an emergent fashion.

Simon appreciated the significance of such dynamic situations. For instance, in a section titled 'Designing without final goals', Simon explored a

paradoxical but pragmatic view of design — to motivate activity that in turn generates new goals. Offering the example of the rebuilding of Pittsburgh, where new goals emerged after initial goals had been accomplished, Simon (1996: 163) concluded:

‘Making complex designs that are implemented over a long period of time and continually modified in the course of implementation has much in common with painting in oil. In oil painting, every new spot of pigment laid on the canvas creates some kind of pattern that provides a continuing source of new ideas to the painter. The painting process is a process of cyclical interaction between the painter and canvas in which current goals lead to the new application of paint, while the gradually changing pattern suggests new goals.’

In these observations we see how a design approach need not be a static representation of a problem, but can involve a ‘theory-design-fly-test and start-all-over-again’ methodology (Boland et al. 2006). Despite this acknowledgment, there has been relatively little work that explores the nature and implications of such a design approach. Indeed, as Boland (2004: 109) states, ‘Much of Simon’s concern centers on the local and immediate experience of an individual who faces an environment that is essentially unchangeable. It is a given to which the managers must mold the organization.’

The challenges that arise in applying a scientific approach to design in dynamic environments become all the more apparent when we consider the nature of change that is upon us. If we accept Woodward’s (1965) powerful insight that technologies of work shape the way in which we work with technologies, new information technologies not only link islands of unconnected activities into an action net (Czarniawska 2004), but, in enabling such connections, change the very meaning of the term ‘design’ to connote continual evolution via interaction. Jelinek (2004: 115) fully understood the implications of this change when she stated:

‘A genuine revolution of possibility flowed from the move between prior flat file systems into relational databases and, in computer systems, the hyperlinked nodes of the internet. Similarly, the virtual organization — often pro tem, frequently voluntary, and broadly distributed — is the iconic organization of our times. But how does one design it? Or should one perhaps instead invite it to emerge? Is the issue deliberate design, or is it design of a process of interactions that allows a collaborative process to emerge? ... How shall we theorize about such organizations? ... Should we embrace the ephemeral organization as a new norm?’

An image of an organization that is not a series of nested black boxes (March and Simon 1958) operating in an immutable environment but, rather, a hyper-text organization that continues to emerge is a radical shift indeed. The design problem, then, is not one of developing a static interface (an edge in network terms) that connects the inside and the outside; rather, it involves the creation of multiple edges between many nodes in a dynamic network. In such an action net, each node can potentially act as a boundary object, ‘remaining between different realms, belonging to all of them simultaneously, and seen from different points of view’ (Czarniawska 2004: 104). Given this, ‘When such an organization does emerge, it may be both transient and protean’ (Jelinek 2004: 115).

These observations further highlight the difficulties associated with designing for completeness in a world that is continually changing. But, what does it mean

to design for incompleteness? Is this an oxymoron? We think not. There are now a number of new organizational forms that suggest that incompleteness, rather than posing a threat, can instead be a virtue. We examine two such cases in this paper: the Linux operating system and the Wikipedia online encyclopedia. These cases provide us with an appreciation of the generative nature of incompleteness. They amplify what Weick (2004: 43) astutely pointed out: 'Life persists when designs are underspecified, left incomplete, and retain tension.'

Research Design

Our objective is to offer a set of observations that form the basis for an ongoing conversation among those interested in understanding the nature of design in continually changing environments. By no means do we claim to offer a full-fledged theory: to do so would defeat the very premise of our argument. Here, we subscribe to the notions proposed by Boland and Collopy (2004), Romme (2003) and others who suggest that the value of theorizing lies in the options that are generated rather than the uncertainties that are resolved. Along these lines, our intent is to sensitize readers to a pragmatic approach to design that harnesses the generative forces of incompleteness.

The research approach that we adopt in this paper involves a detailed exploration of two exemplary cases. Research on analogical thinking (Loewenstein et al. 1999) suggests that individuals who are presented with multiple cases exhibit greater ease in their abilities to identify underlying patterns. Based on this finding, we decided to provide not one but two in-depth cases. We deliberately chose to examine the Linux operating system and the Wikipedia online encyclopedia as each represents a design that is continually evolving. We tracked available information from a wide variety of online sources as a means to collect raw data for our case narratives. The multiple data sources helped us 'triangulate' (Jick 1979) in that there were very few disagreements among the data sources on the factual details involved. Two of the authors separately developed the case studies. The individual inferences drawn from each of the cases were discussed and verified with the other authors. Our aim is not to reach a state of theoretical saturation (Eisenhardt 1989; Glaser and Strauss 1967). Rather, much like the phenomena that we are studying, our aim is to offer a set of observations that will hopefully generate 'theoretical tension' and form the basis for ongoing debate and understanding of this phenomenon.

Linux: Perpetually in the Making

Linux originated as a hobby project of Linus Torvalds, a computer science student at the University of Helsinki. Initially, he wrote programs to understand how the Minix operating system could use features of the 386 processor. Soon, he had written task-switching programs, a disk driver and a small file system. Torvalds realized that he was actually working on a new operating system, and, as a result Linux 0.01 was born. At this stage, Linux lacked

many of the functionalities that users expected from an operating system. Even the numbering of the version (0.01) signified its unfinished status (Diedrich 2001).

Torvalds' initial development efforts can be described as 'discovering design goals in action'. Upon releasing Linux version 0.02, he described the initiative as follows:

'This is a program for hackers by a hacker. I've enjoyed doing it, and somebody might enjoy looking at it and even modifying it for their own needs. It is still small enough to understand, use and modify, and I'm looking forward to any comments you might have ... If your efforts are freely distributable (under copyright or even public domain), I'd like to hear from you, so I can add them to the system ... Drop me a line if you are willing to let me use your code.' (Torvalds, 1991)

His announcement to the newsgroup suggests that he was unsure about what others might want in the emergent system. Although such lack of closure could be a problem from a traditional design perspective, it is interesting to observe how Torvalds turned this into a virtue. His decision to allow others to adapt the emergent system to their needs enabled him to harness the energies of the larger programming community. Within two months of his announcement, about 30 people had contributed close to 200 reports of errors and problems using Linux. In addition, these individuals developed new features for the nascent operating system that, in turn, became the basis for future directions along which the system evolved (Moon and Sproull 2000).

Traditionally, there has been a clear distinction between the designer and the user. However, Torvalds' actions blurred the boundaries between these two actors: the user could also be the designer (von Hippel and von Krogh 2003). By catalyzing such a process, Torvalds ensured that the purpose and functionalities of Linux would now emerge from the efforts of multiple contributors. Providing users with an opportunity to inscribe their local contexts into the design enabled the development and linkage of components in diverse and sometimes non-obvious ways. As these inputs were incorporated, the very purpose and functionality of the platform itself changed.

Technology is society made durable, suggested Latour (1991), and in the case of Linux we can recount many instances of how technology made engagement among contributors possible. Clearly, tools such as mailing lists and news groups made it possible for actors to engage with the emergent platform and with one another. Similarly, 'installers' that facilitated reuse and distributed modifications contributed to the growing functionality of the system. For example, the Linux community experienced a large growth in 1992 when the Yggdrasil distribution made it possible for users to install Linux from a CD-ROM (Diedrich 2001). Finally, Torvalds redesigned the Linux kernel to have one common code base that could simultaneously support a separate tree for different hardware architectures, thereby greatly improving its portability (Torvalds 1999). Taken together, these tools made it possible for contributors to continuously extend Linux in a decentralized manner.

Moreover, social rules built into the technology (Lessig 1999) further fostered generative engagement by actors. A key development was Torvalds' decision to release the kernel under General Public License (GPL), which mandated

that any user modifying or adding to the source code would have to make their own contributions available to everyone else (Stallman 1999). This decision signaled to the community that ownership and control of Linux was not Torvalds' alone. Rather, it facilitated the establishment of a stable 'generalized exchange system' (Kollock 1999) in which people both contributed to and benefitted from the assistance of others. Establishing mechanisms that recognized the contributions of individuals reinforced the feeling of common ownership. A 'credits' file made available with every released version of Linux listed various contributors and their roles in developing and maintaining the kernel.

The availability of the source code played a critical part in the actors' generative engagement with the platform. Through documentation in repositories such as the CVS (Concurrent Version System), the source code served as a design trace that provided a memory of how the platform had evolved until a specific point in time. While creating options for the future, contributors could go back to the past to find out how solutions to related problems had been developed. Moreover, the design trace provided the interpretive flexibility (Pinch and Bijker 1984) required to recontextualize the existing platform and make it work in new application areas without losing coherence with the original platform. Finally, enabling developers to leave their footprint in the source code ensured that their identities became intertwined with the platform and served as a strong attractor for them to contribute.

Overall, these social and technical mechanisms made it easier for contributors to tinker with the system and incorporate their own notions of how it should be further developed. In doing so, they extended Linux in ways that collectively covered a far greater domain than any single individual could have imagined. This manifested itself in the development of a variety of utilities and device drivers that supported specific hardware and peripherals (Thiel 1991). In an interview, Torvalds said about the community's engagement with the platform:

'After I had published Linux on the web, other users requested features I had never thought of and more and more new ideas emerged. Instead of a Unix for my own desktop, Linux should now become the best operating system ever. The requests by other people and their patches and help later on made the whole thing more interesting.' (Diedrich 2000)

Our observations from Linux, then, suggest that incompleteness is generative in two different ways. At one level, incompleteness serves as a trigger for the creation of many diverse ideas on how a design can be extended and further developed. At another level, engagement with such a system both transforms the design as well as creates new avenues for ongoing engagement which, in turn, attracts a new set of contributors who bring into the fold their own contextualized needs, purposes and goals. The contribution of a graphical user interface, for example, spurred a whole new application area called 'Linux on the desktop', a central aspect of current Linux versions (Diedrich 2000). This observation highlights a key benefit of designing for incompleteness: generative engagement with a platform is self-reinforcing in nature, with the boundaries of both its technical and social architecture co-evolving with one another (see also Neff and Stark 2003). While generative engagement enables developers to cope with incompleteness in the present, it is also the source of incompleteness in the future.

Such generative engagement by a multitude of users, however, needs to be channeled to maintain coherence. For instance, it is possible for individuals to introduce security holes or malicious code intended to cause damage (Singer 2005). But even contributors with the best of intentions can sometimes inadvertently bring about damage to the system. More fundamentally, the lack of control can lead to fragmentation of the platform as factions pursue different avenues of development. As an illustration, the many different distributions of Linux (e.g. RedHat Linux, SUSE, or Debian) all come with proprietary patches when system updates are released. In some cases, there have been so many changes made to the original package that the distributions have become incompatible (Wolf 2001). Given these challenges, what governance mechanisms can be established to enable incompleteness to be harnessed beneficially?

Linux's governance approach is best described as 'centrally facilitated yet organizationally distributed' (Garud and Kumaraswamy 2005). A combination of technical and social rules keeps the platform from falling apart. Ongoing documentation of the development process, inscribed into the source code and CVS, serve as a form of cohesive tissue. Chat rooms and bulletin boards act as threads that order interactions among contributors across space and time (Lanzara and Morner 2005). These mechanisms are buttressed by an overarching meritocracy within the community in which developers focus on providing new code and modules that add features and functionality to the platform while others, based on their reputation and prior contributions, assume maintenance tasks such as the evaluation of code submitted by developers for inclusion into new releases. Sitting atop this meritocracy is Torvalds, who centrally facilitates decisions on strategic issues related to Linux. Decisions to deliberately change the platform are entrusted to a much smaller group of individuals when compared to the total number of contributors. Finally, the provisions of the GPL provide the legal and cultural basis for interaction among community members.

These governance mechanisms facilitate access to knowledge, encourage constant tinkering, and curb private appropriation and free riding. Together, these mechanisms operate with an almost invisible touch vis-à-vis ongoing developmental activity on the platform. This enables extensive experimentation to take place on the system even as it maintains a coherent core. By contrast, a tightly controlled design would bear the risk of falling apart, given the difficulties of accommodating the contradictory requirements of a heterogeneous community.

A number of Linux's governance mechanisms have themselves emerged over time in a relatively unstructured fashion. One such convention pertains to the successive release of Linux versions even while distributed development progresses. An even-numbered release (for instance, version 2.4) denotes that the release is stable and ready for use, whereas an odd-numbered release (for instance, version 2.5) signifies that the release is still being built and under evaluation. The announcement of an odd-numbered release serves as the call for new code and modules that drives the current stable release forward. After a period of development, testing, discussion and reviews, no new additions are allowed so that the entire operating system can be tested for performance and reliability. Once testing has ended and modules of code that compromise reliable performance are removed, the operating system is ready to be released as

a stable even-numbered version for widespread use. Such a convention has enabled distributed development.

Overall, our description of Linux highlights what it means to be incomplete by design. To the extent that the system has the capacity to incorporate functionality relatively easily, it is possible to harness its generative properties. Blurring the distinction between users and producers, assuming that preferences are heterogeneous and evolving, and maintaining abstractness in goal definition often facilitates this generative process. These facets lie in contrast to scientific approaches to design that favor clear boundary definitions, fixed user preferences, design closure and platform stability. While systems that are designed for incompleteness may appear messy by conventional design evaluation metrics, they can often outperform traditional designs by being extremely adaptable to continually changing contexts. Adopting such a design approach, then, involves appreciating design as the interplay between intermediary outcomes and processes, with one influencing the other on an ongoing basis.

Wikipedia: A Lumpy Work in Progress

Initiated in 2001 by Jimmy Wales and Larry Sanger, Wikipedia's ambitious mission has been to 'create and distribute a free encyclopedia of the highest possible quality to every single person on the planet in their own language' (Wales 2005). As of 2006, it had become one of the most visited websites in the world, with 5 billion page views monthly. Currently, it is growing at around 6,000 edits a day. What is truly remarkable is that Wikipedia has run as a non-profit organization since 2003 and has five employees in addition to Wales. It meets most of its budget through donations, the bulk of these being contributions of \$20 or less (Schiff 2006).

The typical encyclopedia, as represented by the *Britannica*, is the epitome of a product that is designed to be complete. Its production process involves assembling a large group of experts, working under the direction of a manager, each performing a task on a detailed work chart to produce a work of enormous breadth. This product is then packaged and bound in a set of volumes as an authoritative and accurate source of knowledge. The encyclopedia, then, subscribes to a scientific approach to design, with a clearly defined purpose, a production process that is neatly modularized, clearly delineated boundaries between producers and users, and an emphasis on stability and reliability.

Contrast this with the inner workings of Wikipedia. While the overarching goals of this initiative are similar to that of the *Britannica*, participation in its development is deliberately inclusive, blurring the boundaries between user and producer. Any registered user can write an article that others subsequently modify and refine. This conceptual shift — along with mechanisms that enable generative engagement — has contributed to a system that has now begun to fundamentally question the ontological basis of a traditional encyclopedia.

Technically, entries are made using a wiki, a tool that allows multiple users to create, edit and hyperlink pages. The word 'wiki' comes from the Hawaiian word for 'quick', but also stands for 'what I know is ...', and these definitions, taken

together, provide an essence of the design approach underlying Wikipedia. Anyone can initiate a new article by making a basic entry, a 'stub' in Wikipedia parlance. If a topic that users are searching for is not covered on the site, then they are encouraged to write it themselves. Knowledge, from this perspective, is assumed to reside in a distributed fashion within the entire online human network.

Why might someone contribute to Wikipedia? For many, its goals — a free encyclopedia that is a fun experience to create, and the open-source principles on which it is based — are an important draw. This translates into a strong level of commitment for a subset of individuals who end up spending a considerable number of hours tending to the project. As a Wikipedian described his attachment to the website, 'You can create life in there. If you don't know about something, you can start an article, and other people can come and feed it, nurture it' (Pink 2005).

Given the minimal restrictions on membership, the project has turned fundamental assumptions about the design of knowledge systems on their head. The system is truly democratic in that it does not favor experts over the well-read amateur. As Wales puts it, 'To me, the key is getting it right. I don't care if they're a high-school kid or a Harvard professor' (Schiff 2006). Over 100,000 individuals have made contributions to the website since its inception, with a 24-year-old University of Toronto graduate being the site's premier contributor, having edited more than 72,000 articles since 2001.

How can a system that is based on the participation of literally any individual become a useful knowledge source? On Wikipedia, every article has an open invitation for edits, and the expectation is that most edits will be improvements. Wikipedia operates from the presumption that any individual's knowledge is by definition incomplete, and that ongoing revisions enabled by mass collaboration tools and involving a large group of 'eyeballs' will produce a reliable yet continually evolving knowledge repository. More fundamentally, it reflects an appreciation of the inherently accretive nature of knowledge, one in which the content of any article provides the generative basis for the next set of changes. As Earl (2005) commented, 'The philosophy of Wikipedia is that an article gains validity and maintains currency by continuous widespread updating. Thus, hitherto it has not declared any item finished.'

The wiki keeps track of all changes made by users and allows them to compare multiple versions of an article.⁶ This information provides a design trace which allows actors to understand how and why an article has emerged over time. Possessing such an overarching temporal perspective is critical to repairing damaged elements in an entry as well as reverting to an older version if the current one is inaccurate. Moreover, this trace often forms the basis for new directions in which the article is developed. The design trace, then, both chronicles and initiates generative engagement with an article. In doing so, it serves as a locus of coordination as well as a point of departure, allowing an article to remain in a state of perpetual change.

In addition, social rules embedded in Wikipedia's website enable further generative engagement. For example, tags such as 'The neutrality of this article is disputed' represent calls for action to improve an article. Disambiguation notices placed at the bottom of certain articles help readers accurately locate information

that they are looking for. Finally, the creation of an ontological scheme via categories aids contributors in organizing information available on the site. The categorization of entries itself facilitates a process of adaptive structuration (DeSanctis and Poole 1994) in which the contribution of articles with different content change the meaning of a category that subsequently gets relabeled. These dynamics highlights how the community structures Wikipedia in use.

An interesting by-product of such an evolving knowledge project is its ability to instantaneously respond to current events. When the Indian Ocean tsunami erupted in late 2004, contributors produced several entries on the topic within hours. By contrast, the *World Book*, whose CD-ROM allows owners to download regular updates, had not updated its tsunami or Indian Ocean entries a full month after the devastation occurred (Pink 2005). This ability to provide instant information is a metric on which a knowledge repository organized to change perpetually outperforms a traditional encyclopedia.

The flip side of this design approach, however, is that it produces entries that are often amateurish. While the facts may be sturdy, clarity and concision is sometimes lacking. The initial contributor to an article can set its tone and may not necessarily be highly knowledgeable in the area (Wattenberg and Viegas 2006). Disagreements on an article can lead to repeated back-and-forth editing, with the user who spends the most time on the site prevailing. Given the obsession of many users to rack up edits, simple fixes often take priority over more complex edits. Moreover, the open access to the system has produced incidences of vandalism on the site that involve individuals inserting obscenities within entries. Given these shortcomings, what governance mechanisms ensure that articles do not lose their integrity?

Here again, a mix of technical and social elements work together to maintain the integrity of the website. Five webbots continually troll the site, searching for obscenities and evidence of mass deletions and reverting text as they go. Any editing on an article is automatically logged on a 'recent changes' page that participants can monitor. Anyone who sees something false, biased or otherwise defective can quickly and easily change the text. Moreover, every article also carries with it a separate discussion section on which debates about what to include on the page are encouraged. Besides this, users employ internet relay chat to discuss ongoing issues, from article details to general policy.

An underlying community of volunteers holds these distributed contributions together. This includes anonymous contributors, people who make a few edits. Next, there are registered users who make edits using their by-line. Administrators are individuals who can delete articles, protect pages and block IP addresses. Finally, there are the super elites who make direct changes to the Wikipedia software and database. There is also a mediation committee and an arbitration committee that rule on disputes. On this front, the inner circle of Wikipedians know each other and value their reputations, which are themselves built bottom-up through their past activity on the site. The consequence of such reputation-building has been the creation of a meritocracy with individuals occupying more central positions because of their ongoing involvement when compared to others.

In terms of policies, the founders of Wikipedia have instituted a NPOV (neutral point of view) rule, which urges contributors to present conventionally

acknowledged 'facts' in an unbiased way as well as accord space to both sides when arguments occur (Poe 2006). Over time, Wikipedia has instituted additional rules, reflecting the complexities involved in managing a growing decentralized community. For example, a policy preventing certain contentious subject matters from being openly edited by users was ratified recently (FinancialWire 2006).

This has prompted concern that Wikipedia is becoming a regulatory thicket complete with an elaborate hierarchy of users and policies. Wales, while ambivalent about the growing number of rules and procedures, recognizes them as necessary. According to him, 'Things work well when a group of people know each other, and things break down when its a bunch of random people interacting' (Schiff 2006). And Earl (2005) captured Wikipedia's dilemma as follows:

'This is a classic knowledge-creation conundrum. On the one hand there is real advantage in assembling collective wisdom; on the other hand there are concerns about validity and about incentives or, more particularly, disincentives for content contributors.'

However, taking these steps raises an even more fundamental issue: should Wikipedia abandon its current design approach in order to emulate a traditional encyclopedia? For many, the incomplete nature of an article is valuable in its own right. First, such an article is better than nothing at all. Second, the articles might actually trigger more experienced participants to make contributions that they would otherwise not have made. Instituting governance mechanisms that make it appear more like an encyclopedia could potentially rob Wikipedia of its unique identity and impede generative engagement by contributors. As Carr (2005) elaborated:

'Wikipedia is not an authoritative encyclopedia, and it should stop trying to be one. It's a free-for-all, a rumble-tumble forum where interested people can get together in never-ending, circular conversations and debates about what things mean. Maybe those discussions will resolve themselves into something like the truth. Maybe they won't. Who cares? As soon as you strip away the need to be like an encyclopedia and to be judged like an encyclopedia — as soon as you stop posing as an encyclopedia — you get your freedom back.'

This observation underscores the value of incompleteness. Incompleteness allows participants the freedom to engage with the design in a way that is meaningful to them. As the quote suggests, to the extent that the goals and metrics of traditional approaches that emphasize completeness are adopted in evaluating designs that are inherently incomplete, there exists a danger of losing the unique value that such designs can provide.

Discussion

The Linux and Wikipedia cases affirm what Boisvert (who built upon Deweys' (1934) pragmatic approach) pointed out: 'Affairs are never frozen, finished, or complete. They form a world characterized by genuine contingency and continual process. A world of affairs is a world of actualities open to a variety of possibilities' (Boisvert 1998: 24). Indeed, these cases provide a deeper appreciation of the title of the paper: incomplete by design and designing for incompleteness. In an environment that is continually changing, designs that have

been completed at a point in time are likely to become incomplete over time. On the other hand, designs that anticipate their incompleteness are likely to remain more up to date over time.

Design Participation within the Pragmatic Approach

The traditional scientific approach employed principles from the natural world to design an artifact with enduring qualities that fulfilled a specific purpose in an unchanging world (Simon 1996). From this perspective, a design was fixed in time and space; it was opened and modified only to accommodate exogenous environmental changes. Moreover, the locus of a design, i.e. the demarcation between designer and user, was clear and unambiguous.

In contemporary environments, however, the distinction between designers and users has blurred, resulting in the formation of a community of co-designers who inscribe their own contexts into the emergent design, thereby extending it on an ongoing basis in diverse and non-obvious ways. Such generative engagement by multiple co-designers is facilitated by numerous sociotechnical mechanisms. Tools such as the wiki, licenses such as the GPL, forums such as bulletin boards and the infrastructure provided by the internet work with one another to facilitate participation and enable distributed development. This dynamic action net (Czarniawska 2004), then, contributes to the design remaining in a fluid state.

We can further contrast design participation within the scientific and pragmatic approaches by returning to the parable of the watchmakers offered by Simon. In its original version, user engagement with the design was considered an interruption resulting in watches that remained incomplete and therefore of little value. By contrast, the Linux and Wikipedia cases demonstrate that incompleteness acts as a trigger for generative engagement by co-designers. They are the ones who complete what they perceive is incomplete. They discover the purpose of a design in use. They create avenues for future development that, in turn, attract new groups of co-designers. From a pragmatic design approach, what was considered to be an interruption then, now becomes the basis for ongoing change.

Design Task within the Pragmatic Approach

For co-designers, the design task is very different from the one faced by designers adopting a scientific approach. For the latter, optimization of an objective function given constraints (as in linear programming), represented the dominant approach to designing. By contrast, contemporary designs such as Linux and Wikipedia can be conceptualized as an interlinked set of subjective functions, where one person's subjective function serves as another person's constraints. Complicating matters, the set of interlinked subjective functions is itself underspecified as it emerges in use over time. Under these conditions, the design remains incomplete as the solution to the optimization problem corresponds to more than one point in an n -dimensional space of design parameters. It is for this reason that co-designers must learn to theorize on the fly: i.e. they become reflective practitioners (Schön 1983) who generate

provisional workable solutions to their immediate problems, knowing fully well that the platform that they draw upon and the context to which they apply their solutions will inevitably change.

What is the role of modularity in such an emergent system? Modular elements allow for platform extension as is apparent in both the cases we studied. Under these conditions, extension of the platform takes place through an accretive process where modules are tacked on as 'plug-ins' (Garud et al. 1998). But the two cases also demonstrate a less understood, generative facet of incompleteness that is fostered by 'partial partitioning'. On this front, we suggest that interdependence between partitioned tasks results in situations where changes in one task have a cascading effect on other tasks. Consequently, the system never comes to a rest.

Modularity scholars have suggested that partial partitioning can be a problem (Ethiraj and Levinthal 2004). The Linux and Wikipedia cases, however, indicate that there may be benefits to be harnessed from partial partitioning. The 'shared division of labor' (Nonaka and Takeuchi 1995) that emerges allows for redundancy of functions, i.e. certain functions can be fulfilled by more than one component. This enables the system to work even if some components are damaged or left incomplete. More significantly, such redundancy of functions can be generative in that components can be deployed for a different purpose, thereby facilitating reconfiguration of the design in response to changes in the environment (Garud and Kotha 1994). It also allows the system to more easily assimilate emergent contributions from co-designers into a continually evolving platform.

Design Governance within the Pragmatic Approach

Given the distributed, emergent and protean nature of designs, the challenge now becomes one of establishing rules that provide some stability. While an absence of rules is likely to lead to design fragmentation, too many rules can potentially stifle the design. It is this paradox that needs to be managed to preserve the value proposition that such designs offer.

An appropriate form of governance is required to coordinate real-time distributed contributions in a way that preserves the design's dynamic qualities, i.e. one which allows elements of a system to inform but not determine one another (Barry and Rerup 2006: 267). Governance mechanisms need to be underspecified (Weick et al. 1999) or semi-structured (Brown and Eisenhardt 1997); that is, they possess minimum critical specifications (Emery 1980) to keep the design in a state that is neither too fluid nor too crystallized (see Gehry 2004 for this distinction).

The design trace is a key element that enables such governance. By providing widespread access to knowledge on who contributed what, when and why, the trace makes it easier for actors to understand how a design has emerged over time. Possessing such an overarching temporal perspective is often critical in mitigating design fragmentation. For instance, if a new contribution turns out to be damaged or incompatible, then the trace makes it possible to simply use an older version. Equally important, the trace can be viewed as a boundary architecture (Star and Griesemer 1989) that co-designers draw on to

develop extensions to the design. In sum, the ongoing design trace serves as a locus of coordination as well as a point of departure for such designs.

To further elaborate, consider two different logics of engagement described by Bruner (1986). A causal logic, following the role model of the natural sciences, operates when the set of variables and relationships that define the functioning of a design are fully specified (see Romme 2003 for a more detailed explication). This logic, associated with a scientific approach to design, leads to the articulation of design rules (Baldwin and Clark 2000; Romme 2003). A narrative logic, on the other hand, operates on the basis of a narrative's internal coherence and its external coherence with the listener's existing knowledge. In providing designers with the interpretive flexibility required to generate contextualized solutions and to imagine what might be, the narrative can help coordinate distributed activities across time and space. Such an epistemology connects designs with the emergent purposes of the social groups involved.⁷

The design trace allows for these two different logics to operate simultaneously. The trace possesses a scientific logic that offers co-designers with the *raison d'être* of the design and the necessary details for it to be functional in real time. From this perspective, each node of a trace is but a module with tags that can be opened up and reused. At the same time, the trace also allows for a narrative logic to operate. Co-designers are motivated to participate with a design because of the flexibility that it offers, and the design in use that emerges is convincing to these participants as the narratives recorded have verisimilitude (Bruner 1986). A design trace, then, possesses the equivalents of both an ostensive and a performative dimension (Feldman and Pentland 2003). By providing connections to assets across time and space, a design trace makes it possible for co-designers to engage in the present by building upon the past in anticipation of a new future (Ricoeur 1984).

In enabling the two logics to operate simultaneously, the trace is able to alleviate some of the problems associated with human memory and bounded rationality that Simon considered in his parable. *Tempus* and *Hora* were boundedly rational individuals, relying on their own short- and long-term memories. A mechanism such as the design trace could potentially have extended their memories by serving as a collective mind (Weick and Roberts 1993). As a narrative, the trace makes it easier for individuals to recontextualize the design to their own situations. This makes it possible for interruptions not only to be welcomed, but also to serve as the basis for design extension.

In offering these observations we see how future options on a design are generated even as current contributions are incorporated. Moreover, recording changes in the design trace signals to co-designers that their contributions will be honored in the future. If a trace was not to exist, part of the motivation to participate might be lost: why contribute to a commons when the contributions would not be used in the future? At the same time, if the co-designers were over-dependent on the extant trace, then the future design would largely be based on past experiences and become path-dependent. For a design (and its trace) to promote diachronic processes, then, anticipations of the future and memories of the past must jointly inform contributions in the present.

Conclusion

We have explored the Linux and Wikipedia cases to sketch out the elements of a pragmatic approach to design. In continually changing environments, adopting a design approach that attempts to fix boundaries, goals and purposes is potentially counterproductive. While such an approach may produce a system that is optimal at a point in time, given continual change the system is likely to rapidly become obsolete over time. Under these conditions, a pragmatic approach (one that views design as continually evolving and essentially incomplete) may be more appropriate. Within such an approach, boundaries between designers and users become blurred, heterogeneous user preferences emerge in use, tasks remain partially partitioned and the goals of the design emerge through interaction. Indeed, such an approach harnesses the benefits of incompleteness in comparison to the scientific approach that views incompleteness as a threat.

Eventually, a pragmatic approach involves the fusing together of two meanings of design, that is, as both process and as outcome. Any outcome is but an intermediate step in an ongoing journey, representing both the completion of a process as well as its beginning. Whereas the scientific approach emphasizes the need to crystallize designs, the pragmatic approach highlights the value of retaining fluidity. The essence of this approach is well captured by Hedberg et al. (1976: 43) who noted, ‘Designs can themselves be conceived as processes — as generators of dynamic sequences of solutions, in which attempted solutions induce new solutions and attempted designs trigger new designs.’

Notes

Some of the ideas in this paper are based on prior discussions with Arun Kumaraswamy. We thank him. We also thank the three reviewers of this paper and the editors of this Special Issue for their valuable feedback.

- 1 In this paper, we make a distinction between a ‘scientific approach’ to design that applies analytic thinking to address clearly defined problems to discover an optimal solution (following the natural sciences as a role model) and a ‘pragmatic approach’ that applies synthetic thinking to address ill-structured problems. Others have used the terms ‘science’ vs ‘design’ (Romme 2003) and ‘decision science’ vs ‘design science’ (Boland and Collopy 2004) to make such a distinction.
- 2 Dewey’s (1934) approach to pragmatism informed our understanding of design as noun and verb, an understanding reinforced by the entry on design in Wikipedia (<http://en.wikipedia.org/wiki/Design>).
- 3 This is derived from a structural perspective (Giddens 1979) where structure is both medium and outcome of action.
- 4 Specifying the boundaries is by no means a trivial task. Alexander’s insightful analysis (1964) of the redesign of a tea kettle illustrates this point. At first blush, such a redesign seems simple, given that the kettle is a clearly defined object and the boundaries between the kettle and its environment are obvious. However, Alexander goes on to demonstrate that by changing the nature of the problem to be addressed — for instance, by asking if it is the method of heating kettles rather than the kettle itself that needs redesigning — the solution obtained can change drastically. In reframing the question, the kettle becomes part of the outer environment and the stove becomes the inner environment.
- 5 See Boland and Collopy (2004) for the use of linear programming methods in inventory control, which is based on the maximization of a clearly specified objective function given a set of pre-specified constraints.
- 6 We invite the reader to experiment with the design trace for the Wikipedia entry on ‘Design’ <http://en.wikipedia.org/w/index.php?title=Design&action=history>.
- 7 We thank Georges Romme for this insight.

References

- Alexander, C.
1964 *Notes on the synthesis of form*. Harvard University Press, Cambridge, MA.
- Baldwin, C. Y., and K. B. Clark
2000 *Design rules: The power of modularity*. Cambridge, MA: MIT Press.
- Barry, D., and C. Rerup
2006 'Going mobile: Aesthetic design considerations from Calder and the constructivists'. *Organization Science* 17/2: 262–276.
- Boisvert, R. D.
1998 *John Dewey: Rethinking our time*. Albany, NY: State University of New York Press.
- Boland, R. J.
2004 'Design in the punctuation of management action' in *Managing as designing*. R. J. Boland and F. Collopy (eds), 106–120. Stanford, CA: Stanford University Press.
- Boland, R. J. and F. Collopy, editors
2004 *Managing as designing*. Stanford, CA: Stanford University Press.
- Boland, R. J., M. Jelinek, and A. G. L. Romme
2006 Call for Papers: Organization Studies as a Science of Design. *Organization Studies*.
- Brown, S. L., and K. M. Eisenhardt
1997 'The art of continuous change: Linking complexity theory and time-paced evolution in relentlessly shifting organizations'. *Administrative Science Quarterly* 42: 1–34.
- Bruner, J. S.
1986 *Actual minds, possible words*. Cambridge, MA: Harvard University Press.
- Carr, N.
2005 'Let Wikipedia be Wikipedia' (http://www.routhtype.com/archives/2005/12/let_wikipedia_b.php).
- Czarniawska, B.
2004 'Management as designing for an action net' in *Managing as designing*. R. J. Boland and F. Collopy (eds), 102–105. Stanford, CA: Stanford University Press.
- DeSanctis, G., and M. S. Poole
1994 'Capturing the complexity in advanced technology in use: Adaptive structuration theory'. *Organization Science* 5/2: 121–147.
- Dewey, J.
1934 *Art as experience*. New York: Minton Balch & Company.
- Diedrich, O.
2000 'Was Anwender tun, ist niemals falsch' c't, 90.
- Diedrich, O.
2001 'Happy Birthday, Tux!' c't, 162.
- Earl, M.
2005 'Wikipedia's struggle to govern a knowledge democracy'. *Financial Times*, London.
- Eisenhardt, K. M.
1989 'Building theories from case study research'. *Academy of Management Review* 14/4: 532–550.
- Emery, F.
1980 'Designing socio-technical systems for "greenfield" sites'. *Journal of Occupational Behaviour* 1: 19–27.
- Ethiraj, S. K., and D. Levinthal
2004 'Modularity and innovation in complex systems'. *Management Science* 50/2: 159–173.
- Feldman, M. S., and B. T. Pentland
2003 'Reconceptualizing organizational routines as a source of flexibility and change'. *Administrative Science Quarterly* 48/1: 94–118.
- FinancialWire
2006 'Wikipedia to change revision policy'. June 19 2006. Forest Hills, MD: FinancialWire.
- Ford, H., and S. Crowther
1922 *My life and work*. Garden City, NY: Doubleday, Page & Company.
- Garud, R., S. Jain, and C. Phelps
1998 'Technological linkages and transience in network fields: New competitive realities' in *Advances in strategic management*, Vol. 14, J. Baum (ed.), 205–237. Greenwich, CT: JAI Press.
- Garud, R., and S. Kotha
1994 'Using the brain as a metaphor to model flexible production systems'. *Academy of Management Review* 19/4: 671–698.
- Garud, R., and A. Kumaraswamy
2005 'Vicious and virtuous circles in the management of knowledge: The case

- of Infosys Technologies'. *MIS Quarterly* 29/1: 9–33.
- Gehry, F. O.
2004 'Reflections on designing and architectural practice' in *Managing as designing*. R. J. Boland and F. Collopy (eds), 19–35. Stanford, CA: Stanford University Press.
- Giddens, A.
1979 *Central problems in social theory: Action, structure, and contradiction in social analysis*. Berkeley, CA: University of California Press.
- Glaser, B. G. and A. L. Strauss
1967 *The discovery of grounded theory: strategies for qualitative research*. Hawthorne, NY: Aldine de Gruyter.
- Hedberg, B. L., P. C. Nystrom, and W. H. Starbuck
1976 'Camping on seesaws: Prescriptions for a self-designing organization'. *Administrative Science Quarterly* 21: 41–65.
- Jelinek, M.
2004 'Managing design, designing management' in *Managing as designing*. R. J. Boland and F. Collopy (eds), 113–120. Stanford, CA: Stanford University Press.
- Jick, T. D.
1979 'Mixing qualitative and quantitative methods: Triangulation in action'. *Administrative Science Quarterly* 24/4: 602–611.
- Kanigel, R.
1997 *The one best way: Frederick Winslow Taylor and the enigma of efficiency*. New York: Viking.
- Kollock, P.
1999 'The economies of online cooperation: Gifts and public goods in cyberspace' in *Communities in cyberspace*. M. A. Smith and P. Kollock (eds), 220–239. London: Routledge.
- Langlois, R. N.
2002 'Modularity in technology and organization'. *Journal of Economic Behavior & Organization* 49: 19–37.
- Langlois, R. N., and P. L. Robertson
1992 'Networks and innovation in a modular system: Lessons from the microcomputer and stereo component industries'. *Research Policy* 21/4: 297–313.
- Lanzara, G. F., and M. Morner
2005 'Artefacts rule! How organizing happens in open source software projects' in *Actor-Network-Theory and Organizing*. B. Czarniawska, and T. Hernes (eds), 67–90. Copenhagen: Copenhagen Business School Press.
- Latour, B.
1991 'Technology is society made durable' in *A sociology of monsters: Essays on power, technology and domination*. J. Law (ed.), 103–131. London: Routledge.
- Lessig, L.
1999 *Code and other laws of cyberspace*. New York: Basic Books.
- Loewenstein, J., L. Thompson, and D. Gentner
1999 'Analogical encoding facilitates knowledge transfer in negotiation'. *Psychonomic Bulletin and Review* 6/4: 586–597.
- March, J. G., and H. A. Simon
1958 *Organizations*. New York: Wiley.
- Moon, J. Y., and L. Sproull
2000 'Essence of distributed work: The case of the Linux kernel'. *First Monday* 5/11.
- Neff, G., and D. Stark
2003 'Permanently beta: Responsive organization in the internet era' in *The internet and American life*. P. E. N. Howard and S. Jones (eds). Thousand Oaks, CA: Sage.
- Nonaka, I., and H. Takeuchi
1995 *The Knowledge-creating company*. New York: Oxford University Press.
- Parnas, D.
1972 'On the criteria to be used in decomposing systems into modules'. *Communications of the ACM* 15/12: 1053–1058.
- Petroski, H.
1996 *Invention by design: How engineers get from thought to thing*. Cambridge, MA: Harvard University Press.
- Pinch, T. J., and W. E. Bijker
1984 'The social construction of facts and artefacts: or How the sociology of science and the sociology of technology might benefit from each other'. *Social Studies of Science* 14/3: 399–441.

- Pink, D. H.
2005 'The Book Stops Here'. *Wired* 13.03.2006 (<http://www.wired.com/wired/archive/13.03/wiki.html>).
- Poe, M.
2006 'The Hive'. *The Atlantic Monthly* September 2006: 86–96.
- Ricoeur, P.
1984 *Time and narrative*. Chicago: University of Chicago Press.
- Rindova, V. P., and S. Kotha
2001 Continuous 'morphing': Competing through dynamic capabilities, form, and function. *Academy of Management Journal* 44/6: 1263–1280.
- Romme, A. G. L.
2003 'Making a difference: Organization as design'. *Organization Science* 14/5: 558–573.
- Rosenberg, N.
1982 *Inside the Black Box: Technology and economics*. Cambridge: Cambridge University Press.
- Schiff, S.
2006 'Know It All'. *The New Yorker* 31 July 2006 (http://www.newyorker.com/archive/2006/07/31/060731fa_fact).
- Schön, D. A.
1983 *The reflective practitioner: How professionals think in action*. New York: Basic Books.
- Simon, H. A.
1962 'The architecture of complexity'. *Proceedings of the American Philosophical Society* 106/6: 467–482.
- Simon, H. A.
1996 *The sciences of the artificial*, 3rd edn. Cambridge, MA: MIT Press.
- Singer, M.
2005 'Linux creator: A little fragmentation is good'. *Internetnews.com* (<http://www.internetnews.com/dev-news/article.php/3467241>).
- Stallman, R.
1999 'The GNU Operating System and the free software movement' in *Open sources: Voices from the open source revolution*. C. DiBona, S. Ockman, and M. Stone (eds), 53–70. Sebastopol, CA: O'Reilly & Associates.
- Star, S. L., and J. R. Griesemer
1989 'Institutional ecology, "translations" and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907–39'. *Social Studies of Science* 19/3: 387–420.
- Thiel, W.
1991 'Keyboard.S with German Keyboard Linux-Activists' (<ftp://tsx-11.mit.edu/pub/linux>).
- Torvalds, L.
1991 'Free Minix-like Kernel Sources for 386-AT'. *comp.os.minix* (<http://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind9110a&L=minix-l&T=0&P=8303>).
- Torvalds, L.
1999 'The Linux edge' in *Open sources: Voices from the open source revolution*. C. DiBona, S. Ockman, and M. Stone (eds), 101–111. Sebastopol, CA: O'Reilly & Associates.
- von Hippel, E., and G. von Krogh
2003 'Open source software and the "private-collective" innovation model: Issues for organization science'. *Organization Science* 14/2: 209–223.
- Wales, J.
2005 'Wikipedia is an encyclopedia' (<http://mail.wikipedia.org/pipermail/wikipedia-l/2005-March/038102.html>).
- Wattenberg, M., and F. B. Viegas
2006 'The hive mind ain't what it used to be'. *Edge* (http://www.edge.org/discourse/digital_maoism.html).
- Weick, K. E.
2004 'Rethinking organizational design' in *Managing as designing*. R. J. Boland and F. Collopy (eds), 36–53. Stanford, CA: Stanford University Press.
- Weick, K. E., and K. H. Roberts
1993 'Collective mind in organizations: Heedful interrelating on flight decks'. *Administrative Science Quarterly* 38: 357–381.
- Weick, K. E., K. M. Sutcliffe, and D. Obstfeld
1999 'Organizing for high reliability: Processes of collective mindfulness' in *Research in organizational behavior*. R. I. Sutton, B. M. Staw (eds), 81–123. Stamford, CT: JAI Press.

Wolf, C.
2001 The ROCK Linux Philosophy
(www.linuxdevcenter.com)

Woodward, J.
1965 *Industrial organization: Theory and practice*. New York: Oxford University Press.

Raghu Garud

Raghu Garud is Alvin H. Clemens Professor of Management and Organization and the Research Director of the Farrell Center for Corporate Innovation and Entrepreneurship, Pennsylvania State University. Before joining the Smeal College, Raghu Garud was at the Stern School of Business, New York University. Raghu earned a PhD degree in strategic management and organization from the University of Minnesota. Raghu's research currently explores the emergence of novelty. Specifically, he is interested in understanding how new ideas emerge, are valued, and become commercialized. He is a PI on an NSF-funded project titled 'The architecture of collaboration in scientific communities'.

Address: Farrell Center for Corporate Innovation and Entrepreneurship, Smeal College of Business, The Pennsylvania State University, 451 Business Building, University Park, PA 16802, USA.

Email: rug14@psu.edu

Sanjay Jain

Sanjay Jain is an assistant professor of management and human resources and a principal faculty member of INSITE (Initiative for Studies in Technology Entrepreneurship) at the University of Wisconsin–Madison. He received his PhD from the Stern School of Business at New York University. His current research focusses on technology strategy, institutional entrepreneurship, innovation management and university technology commercialization. His work has been published in the *Academy of Management Journal* and *Industrial and Corporate Change* and covers such technologies as Java and embryonic stem cells.

Address: 4271 Grainger Hall, 975 University Avenue, Madison, WI 53706, USA.

Email: sjain@bus.wisc.edu

Philipp Tuertscher

Philipp Tuertscher is a researcher at the E&I Institute for Entrepreneurship and Innovation at the Vienna University of Economics and Business Administration, Austria, and PhD candidate at University of St Gallen, Switzerland. Before joining E&I, Philipp was a visiting researcher at the Farrell Center for Corporate Innovation and Entrepreneurship, Pennsylvania State University. His current research explores how technological architectures emerge in complex innovation projects. He is particularly interested in the role of technological controversies and how these get resolved as different groups in a project interact with each other.

Address: Vienna University of Economics and Business Administration, Augasse 2-6, A-1090 Vienna, Austria.

Email: philipp.tuertscher@wu-wien.ac.at