

MINING CREATIVITY RESEARCH TO INFORM DESIGN RATIONALE IN OPEN SOURCE COMMUNITIES

Winslow Burleson

*School of Computing, Informatics, and
Decision Systems Engineering
Arizona State University
USA*

Priyamvada Tripathi

*School of Computing, Informatics, and
Decision Systems Engineering
Arizona State University
USA*

Abstract: *Design rationale can act as a creativity support tool. Recent findings from the field of creativity research present new opportunities that can guide the implementation and evaluation of design rationale's ability to foster creative processes and outcomes. By encouraging the exploration of failure through use of analogy, design rationale can foster creative transfer and enable progress in new directions. Open source communities offer an opportunity to observe a form of intrinsically motivated ad hoc design rationale, exhibiting formal and informal information transfer links within forums and allowing access to common tools, expertise, and mentorship. A discussion of a spectrum of implementations of design rationale informs strategies to mitigate conflicts and advance inherent synergies between design rationale and creativity.*

Keywords: *design rationale, creativity research, creativity support tools, open source communities.*

INTRODUCTION

The 2003 National Research Council's (NRC) *Beyond Productivity* report examined elements of creativity and the emergence of human–computer interaction in everyday life, emphasizing a need for information technology (IT) to support creativity across domains (Mitchell, Inouye, Blumenthal, & the NRC 2003). It was proposed that adoption of improved tools that support creativity using IT backbones would yield economic and cultural benefits. Since then, available technology has matured even further, making creativity support easier and more productive. The NRC report also acknowledged that today's IT user has needs that go beyond traditional requirements for productivity and efficiency. IT has enabled today's user to join and contribute to new communities with ease, and to engage in creative acts on an everyday basis. These users, collectively engaged in creation, production, and distribution, demand that technologies work

with them, engage them, and keep them motivated. They thrive on competition and feedback from peers. They are socially embedded through IT networks, seek community-level knowledge sources, and revel in collaborative work. These new generations of users and developers are, in short, mature users of technology.

A complementary report by the Computing Research Association (2002) outlined five grand challenges of computer science and information science, including the challenges of creating opportunities for personalized learning, “a teacher for every learner,” and using hybrid teams of humans and software/hardware system technologies to create a team of one’s own (p. 5). By interacting with individuals and teams to enhance design rationale, these systems act as support tools. When used to support creativity, these software agents are considered to be creativity research tools. Systems that allow teams of software agents and humans to collaborate present an opportunity to use design rationale as a creativity support tool, modulating and communicating process and outcome.

Open source communities present an environment in which users and developers can take advantage of design rationale to support creativity. Open source simply means software whose source code is freely available for modification and reuse, in contrast to the commercial model of software that restricts access to source code to a firm’s employees and contractors. The open source paradigm has given rise to new organizational structures and practices, allowed for distributed community management of software, and promoted collaboration among participants. Open source’s mass involvement of highly skilled, intrinsically motivated participants creates an optimal environment to examine techniques to support creativity and employ design rationale.

In this paper, we discuss the use of design rationale as a creativity support tool in the context of the open source paradigm. The goal of advancing creativity in software design environments faces at least two issues. First creativity and new opportunities for creativity must be appreciated within the ongoing cognitive activity that occurs among software developers. Second, a suitable framework that supports articulation of these creative processes must be developed and used to enhance the processes and products of programming. As Hanson, a principal research scientist at MIT cited in von Hippel (2005), stated,

Creative programming takes time, and careful attention to the details. Programming is all about expressing intent and in any large program there are many areas in which the programmer’s intent is unclear. Clarification requires insight, and acquiring insight is the primary creative act in programming. But insight takes time and often requires extensive conversation with one’s peers. (p. 124)

Understanding the ways in which the elements of time, attention to detail, expressing intent, clarity, insight, and deep and persistent conversation manifest within the structure, process, and success of open source communities can elucidate ways in which design rationale contributes to creative programming.

As the design community grapples with its understanding of the relationship between design rationale and creativity (Carroll, 2010), an exploration of the question, “How can design rationale support creative processes and outcomes?” might allow fuller realization of design rationale’s potential. To address this question we will turn to the many possible implementations of design rationale, and the effect that each of them may have on creativity. First, we must ask ourselves how well design rationale plays the role of a creativity support tool, and what strategies can be

employed to enhance its efficacy. To understand this in terms of the context of the open source paradigm and the role of creativity in open source communities, we must understand the functions and roles of creativity support tools. Likewise, we must understand what creativity research tells us about the best practices for supporting creativity. This will include wider discussion of the consensual assessment of creativity (Amabile, 1983) and the role of transdisciplinary collaboration and metacreativity (Buchanan, 2001). We ground this discussion in the evolving nature of human–computer interaction and the opportunities for design rationale to support creativity when design activity spans disciplines, as is seen in hybrid software/hardware development, and process and product development in open source and do-it-yourself communities (Kuznetsov & Paulos, 2010; von Hippel, 2005).

DESIGN RATIONALE

Moran and Carroll (1996) defined design as “the process of creating tangible artifacts to meet intangible human needs” (p. 2). They stated that design seeks to fill the stated and unstated needs of the end user by bridging the gap between requirements and end product. A typical lifecycle of a design project bridges this gap with six phases: the requirement phase, design phase, building phase, deployment phase, maintenance phase, and redesign phase. In some respects, this is similar to models of software development life cycles. In both, design rationale plays an important role because, in practice, these phases and cycles are usually not strictly delineated. The iteration and complexity of their interconnection adds richness and depth to design projects.

Several definitions of design rationale exist, but Moran and Carroll (1996) defined six broad ways in which the term is used. Fundamentally, it is an “expression of relationship between a designed artifact, its purpose, the designer’s conceptualization, and the contextual restraints on realizing the purpose” (p. 8). Design rationale could be (a) logical reasons given to justify a designed artifact, (b) a notation for the reasons, (c) a method by which reasons for the design are made explicit, (d) documentation of reasons for the design, (e) steps for the design, or (f) the history of the design and its context. It provides an explanation of why a designed artifact is manifest in the manner that it is. Overall, “design rationale is concerned with systemizing the design process—its tools, techniques, methods, and management—for artifacts and their specifications” (Moran & Carroll, 1996, p. 8).

Fischer, Lemke, McCall, and Morch (1991) stated that the benefits of using design rationale are support for the maintenance and redesign of an artifact, reuse of design knowledge, and critical reflection. MacLean, Young, Bellotti, and Moran (1996) stated that design rationale aids reasoning and communication. Moreover, design rationale can encompass several tasks (MacLean et al., 1996), including documentation, understanding, debugging, verification, analysis, explanation, modification, and automation (Moran & Carroll, 1996). From these perspectives it becomes apparent that design rationale is meant as an explicit effort to promote deeper understanding of the design process and decision making, and to transfer this knowledge within the design team and community. Multiple facets of design rationale have emerged to facilitate and enhance the design process. The question at hand, then, is whether design rationale can be implemented in ways that promote creativity. To address this question, we will explore different implementations of design rationale in terms of their granularity and formality, instantiation, and scale.

Granularity and Formality

Implementation techniques can vary across a range of granularities. Fine-grained approaches detail every step and decision in the process, while coarse-grained methods take a broader, macrolevel view of documentation. Fine-grained design rationale is frequently time-consuming and can be burdensome and disruptive to designers and programmers and their creative processes. Coarse-grained design rationale runs the risk of missing significant events.

Design rationale can also be implemented across a range of formalities. Formal techniques usually track a prespecified set of concepts and categories, but may not fully represent every aspect of design processes. These communication tools can act as accessories to design process deliberation; however it is generally understood that even the most advanced capture tools will fail to completely record all underlying decisions and meanings (Gruber & Russell, 1996). This is particularly true when a new audience accesses the recorded design rationale, since the new users' backgrounds, assumptions, and even working vocabulary may have shifted significantly. More positively, formal processes can encourage deliberative processes, reflection, self-explanation, and incubation. The constraints of formal processes make design rationale easy to encode and enhance compliance, which in turn can aid subsequent decision making. Including explicit instruments to formally document creativity within design rationale may help designers to value it within their process.

Informal methods allow considerable freedom but also demand discipline if they are to be a useful compendium. Informal design rationale can be encoded freestyle and later transcribed, or it can be encoded in parallel, where programmers code decisions as they are made. Informal methods can be passive, such as videotaping, or active and explicit in capturing design processes and materials. In design studios, for example, informal compendiums of the design process can include recorded logbooks, Post-it wall and smart board images, concept maps, brainstorming sessions and conclusions, discussions and choices of methodologies, scenario design, and refinement or prototyping processes. In open source software and do-it-yourself communities, informal design rationale compendiums can include information transfer links (von Hippel, 2005), for example, logged communication channels and forums, e-mail, chat, and community websites such as www.linuxforums.org, instructables.com, [sparkfun](http://sparkfun.com), newgrounds.com, or e-how.

To fully understand design rationale's potential to support creativity, a system of evaluation is required. Amabile's (1983) consensual assessment technique can comparatively evaluate diverse implementations of design rationale. Take for example the assessment of design rationale across four conditions of a 2 x 2 experiment in which individuals or teams engage in a creative task with informal/formal x fine/coarse design rationale. Programmers who have been selected as judges based on their interrater reliability (the degree to which judges' ratings of creative products correlate with each of the others' ratings) can evaluate the creativity of the processes and products of these teams. The findings would inform the development of best practices for supporting creativity through design rationale.

Instantiation

Implementations of design rationale within a design process can occur at multiple levels. For example, design rationale practices can be structured to occur as a philosophy, protocol, schedule, tool, interface, or system. Different individuals and organizations will place different

values on design rationale as a philosophy, from considering it a core attribute to simply ignoring it. Similarly, compliance with the practice of design rationale may be enforced strictly, or followed only on an as-needed basis. With or without a philosophy, design rationale can be implemented as a protocol with guidelines and rules regarding its implementation. As a schedule (which may or may not be an element of a protocol), design rationale can be regulated, either on a time or event (i.e., new idea or change in course) basis. As a tool, interface, or system, design rationale can take a simple form or be mediated through multiple points of view (e.g., from the perspectives of various stakeholders). It can take place through e-mail, on-line activity, or even multiperson interfaces with virtual team members acting as design facilitators and design rationale elicitors. In the context of our considerations of design rationale as a creativity support tool, it is worth noting that creativity and support for creativity can also be considered at multiple levels (granularity, formality, instantiation, organizational, etc.).

Organizational

Regardless of the level of formality and granularity and its instantiation, the essence of the value of design rationale is that it provides a record of the reasons for a particular choice and preserves relevant consideration of alternatives, which in turn enables discussion, revision, reflection, and community building. Implementation of design rationale to facilitate creativity inherently takes place within the context of an organization and/or culture; that context has consequent impact on creativity and design rationale. Recent theory in creativity research has been generated both from large-scale cultural and organizational contexts and from individuals and small teams, while design rationale, at least at the point of its generation, has by and large focused on individuals and small teams. The environments' impact on design rationale can be considered in terms of its adoption and usage, and measured in terms of its benefits and efficacy in supporting creativity and the community. A full discussion of the relationship between design rationale and the context of its implementation is beyond the scope of this paper. Nonetheless, our exploration of creativity research and its appreciation for organizational influences on creativity can improve understanding of how design rationale can foster creativity.

In terms of organizational scale or complexity, applications of design rationale and creativity research can focus on relatively small organizations, ranging from individuals to small teams, and can extend to the broader issues of larger organizations, such as divisions, companies, and large-scale distributed communities, as are found in open source. Most environments in which design activity and creativity take place have their own forms of ad hoc design rationale that, at least in part, foster creativity. To the extent this is true, there exists the potential to apply lessons from both creativity research and design rationale to further enhance communication, creativity, and design outcomes. Design projects, ranging from architectural or landscape planning to writing a novel to software development, involve sketches (i.e., drafts, prototypes, or templates) that allow for the exchange and development of initial ideas and project requirements. When individuals and small teams work closely together, design requirements and possible solutions are frequently less complicated and more easily communicated across the team than when such projects are developed by larger organizations. Larger organizations must negotiate common agendas and effective strategies to build design rationale into their organizational structures and practices. Take, for example, individual software developers employing class and Unified Modeling Language diagrams to establish system architecture features and objectives as they manage initial

requirements. These sketches can be challenging, ambiguous, and cumbersome for small-scale organizations. However, in the context of a large-scale open source design community, project requirements and solutions are often difficult to articulate and agree upon. In order for these larger organizations to make progress, communication and the exchange of ideas and assumptions between peers is necessary. The communal development of this common understanding encoded by information transfer links (von Hippel, 2005) is a form of large-scale organizational design rationale, which, in turn, can support organizational progress and creativity.

Beyond the realm of software organizations, emergent hacker spaces and their commensurate on-line communities are an exciting contemporary open source phenomenon, the next generation of physical/digital design and human-computer interaction (Buxton, 2007). Increasingly, successful software is the product of integrated development, with well-defined software design attributes that are consonant with the hardware. For example, Apple's Multi-Touch trackpad technology, which allows users to navigate their electronics using various motions and gestures, makes use of both hardware and software affordances.¹ These synergies and designers' understandings between and across hardware and software systems are rapidly evolving within open source and do-it-yourself design communities. These communities develop not only highly creative hybrid physical/digital artifacts, but also expertise and social engagement among their members.

As described above, design rationale covers the spectrum from fine-grained descriptions of all reasoning processes to an organizational structure that provides guidelines to share creativity in the commons. In many contexts, the implementation of design rationale advances the principles and purposes of a creativity support tool.

CREATIVITY RESEARCH

Developing a design rationale that serves as a creativity support tool first requires an understanding of the field of creativity research. Creativity researchers suggest that the quality of creativity can be evaluated based on the value and level of meaning of a new idea product. In this vein, Csikszentmihalyi (1996) claimed that different people and groups experience creativity in multiple ways. Researchers also may distinguish ordinary creativity—small departures, insights, and innovations in everyday life—from the creativity of the few known geniuses, such as Einstein and Van Gogh. Gardner (1994) expressed the former as “little c” creativity as opposed to “big C” creativity. Within creativity research, creativity is recognized as a natural part of ordinary human existence. Shneiderman (2003) described these forms of creativity as everyday, evolutionary, and revolutionary. Since any advancement in society requires some new idea or process, creativity researchers also appreciate that creativity is an important process for societal transformation. Moreover, creativity is an integral cognitive process that is a fundamental part of human makeup.

In creativity research, creativity is considered to be present within any product or process that is novel and appropriate (Sternberg & Lubart, 2007), and is a part of everyday life and work activity (Certeau, 1984), in contrast to the more popular conception that creativity primarily occurs through “Eureka!” moments. French philosopher Michel de Certeau (1984) was one of the first theorists to propose the concept of everyday anonymous creativity by ordinary people. The many innovative ways that people recycle, adapt, or transform everyday

objects for their own benefit demonstrate everyday acts of creativity and design. This idea has taken hold in interaction and design research as well (Wakkary & Maestri, 2007). Photographer Richard Wentworth (1978), for example, created a photo series aimed at reframing our conception of everyday creative acts—using a bottle cap as a makeshift ashtray, or jamming an alarm clock with a half-eaten candy bar. Wakkary and Maestri (2007) reported similar everyday creative acts within families. Amabile (1983) developed the consensual assessment method as a way to rank creative acts. Studying activities such as writing poetry, building block towers, and making collages, she conceptualized a continuous creativity spectrum that ranks degrees of creativity. Her research demonstrated that judges, selected for their high degree of interrater reliability can be used to perform consensual evaluation of the creativity of most processes or products. In a range of studies, this method has been shown to have strong validity and is one of the most widely used and accepted evaluation methods in creativity research. The consensual assessment method could also be used to evaluate creativity in the process of design and compiling design rationale.

Amabile (1996; Amabile & Mueller, 2002) has studied the structure of creativity within individuals and developed the componential model of creativity that identifies three components within the individual that have an effect on creativity: an individual's intrinsic motivation; his/her thinking style; and his/her domain-relevant skills. Intrinsic motivation includes attitudes, perceptions of personal motivation toward a task. Domain-relevant skills include knowledge about the domain, technical skills, and talent. Thinking style includes convergent/divergent processes and implicit/explicit knowledge and appropriate use of heuristics for generating, evaluating, and implementing ideas.

In addition to the componential model's three elements within the individual, Amabile (1983) also identified environmental influences as a fourth component that affects team processes. Her studies of high-tech team collaborations "in the wild" have demonstrated important factors, including the role of affect; time pressure; focused attention; sequestered and prioritized creative activity; motivation; feedback; and actualizing rewards (Amabile, Hadley, & Kramer 2002; Amabile & Kramer, 2003). Actualizing resources and rewards are those that empower individuals to achievements they would otherwise not be capable of. For example, mentorship, tutorials, and on-line forums in open source software and do-it-yourself communities are actualizing resources for their members. Actualizing rewards might be an invitation or resources to attend a conference or lead a team that could assist in furthering already successful endeavors. In a series of studies creativity researchers have found that positive affect promotes creativity both in the moment and from one day to the next. Positive affect may come in the form of a small gift or joke, or as a positive event, breakthrough idea, or actualizing reward for progress in recognition of the creative activity. Open source communities often display elements of creativity support that are consistent with the framework developed by creativity researchers; existing frameworks for identifying and evaluating creativity can inform our discussion of creativity in these communities.

OPEN SOURCE COMMUNITIES

Open source software is characterized by its free availability to be modified and used by anyone, under a few sets of restrictions. The open source agreement restrictions generally

prohibit the use of code in commercial ventures. Open source on-line communities have their origins in the hacker culture, which is an example of what von Hippel (2005) called communities of “lead users.” Lead users’ adoption of toolkits and development of creative projects are core attributes of von Hippel’s notions of advancing democratic innovation. In the early 1990s, research laboratories, such as MIT’s Artificial Intelligence Laboratory, started licensing their software, restricting access to source code. Some lead programmers of the original source codes were upset by this control over what they felt was community property. In addition, several felt that this level of commercialization severely inhibited the growth of the field. This is an example of what Amabile & Mueller (2002) would call an environment that failed to provide “actualizing rewards.”

Stallman proposed the GNU General Public License (GPL) in the 1980s.² GPL grants the right to use software, to study and modify the source code, and to distribute or redistribute modified or unmodified versions at no cost. Furthermore, GPL restricts the right to use or incorporate the code into proprietary commercial software. In 1998, the open source software movement was formalized by Perens and Raymond. This movement emphasized benefits of sharing source code as we see today (DiBona, Ockman & Stone, 1999; Corbet, Kroah-Hartman, & McPherson, 2010; von Hippel, 2005). The open source movement resonated with individuals who were motivated to be part of the anticorporate culture that was emerging in response to big corporations such as Microsoft and their emphasis on closed system software. In contrast to the restrictive environment (described above), the environment created by GPL could be described as an actualizing one, in which the success of one’s creative products were shared and adopted widely and had visible impacts.

The scale of the open source community is significant. As of July 2011, more than 300,000 software projects have been registered on the Website sourceforge.net, a database of open-source software projects. The success of open source communities is largely explained by the intrinsic motivation of its contributors, who code and share information based on their intrinsic interests and domain expertise and act socially to engage in creative activity.

The rights granted by GPL have enabled the open source community to grow; this growth has required concerted conversation about the community’s agenda and decision making at the macro (within the structure of the community) and micro levels (within the code). This conversation and the deliberations now present a robust ad hoc design rationale corpus of the community’s endeavors. This corpus and the practices of the community around decision making and strategies for progress therefore form a basis for discussion of macrolevel design rationale.

CREATIVITY IN OPEN SOURCE COMMUNITIES

Open source communities are made up of programmers who are engaged in collaborative group activity, making the capture and effective communication of individuals’ design process and decision making particularly important. Open source communities use diverse modes of communication, forming unique structures that foster widespread everyday creativity. O’Mahony (Stark, 2003) investigated several characteristics of open source communities, finding that open source software developers are intrinsically motivated, value informality, and have distaste for “administrivia” (Stark, 2003). These characteristics are in line with a preference for informal and coarse-grained design rationale. However, as

O'Mahony (Stark, 2003) pointed out, some open source communities have formalized their organizational structures by creating formal boards and designated management roles. Design rationale can be seen to sustain the communal and creative goals of open source communities through a range of granularities, formalities, and organizational elements.

There are parallels between the existing functions of creativity support tools (see below) and the organization of open source communities. To more fully understand the interplay of creativity in open source communities, we now review the roles users play in the process of software coding in open source paradigms. A project is introduced by owners (also referred to as “maintainers” or even “gatekeepers”) who are responsible for project management. These project managers also set up an infrastructure for the project that those interested can use to seek help, provide information, or provide new open source code to test and discuss. People download the projects that attract interest and “play” with the code. Some of these people go on to create new and modified code. New code, deemed to be of interest and value by the project maintainers is authorized (von Krogh, Spaeth, & Lakhani, 2003), and posted back into the infrastructure. In essence this parallels the four-stage process of advancing creativity within information technology contexts, described by Shneiderman (2003) in his book *Leonardo's Laptop*, as a process comprising collecting (domain expertise, information collection), relating (analogical transfer, comparison among works), creating (development, testing), and donating (dissemination, diffusion). In open source communities, the credibility of members is determined through status, experience, and expertise. The roles can also overlap. Typically none of the roles are strictly enforced and most work is voluntary and intrinsically motivated, key parts of Amabile's (1983) componential model of creativity.

O'Mahony (Stark, 2003) identified three great challenges within the open source software paradigm that both inform our discussion and present opportunities for design rationale to support creativity. The first of these challenges is resources. The effective use of resources as actualizing rewards can be an opportunity for fostering a commitment to design rationale and as a tool for promoting intrinsic motivation and creativity. Second, the tension between creative freedom and need for structure and management is at the crux of the debate about the compatibility of design rationale and creativity (Carroll, 2010); the appropriate balance of formality and granularity can be difficult to find. Third, the need for sustaining pluralism in governance presents the classic challenge of individual and shared voices and shared language (O'Mahony & Ferraro, 2007). These are common challenges present in what Rittel (1972) termed “wicked problems.” Werner and Rittel's (1970/1979) participatory approach to the development of the issue-based information system (IBIS) and its use as a tool for design rationale have advanced approaches to shared dialogue; recent work on computational support and analysis of shared dialogue systems have further advanced these strategies (Conklin, 2003). Effective journalism (a record and forum for the communication and debate of multiple perspectives) can lead to effective policy and ultimately toward effective governance of a community, providing an actualizing environment within which to advance creativity. Beschastnikh, Kriplean, and McDonald's (2008) studies of organization and governance within Wikipedia support this; they characterized Wikipedia's “policy environment [as]—user editable, reflective of [best] practice, and easily citable” (p. 34). Further, they highlighted the potential for effective policy to foster public deliberation (Beschastnikh et al., 2008). At its core, this is what design rationale, and in turn advancing creativity (especially in open source communities), is about—creating an environment in which shared understanding, decision making, collaboration, and transdisciplinary creativity can occur.

Von Hippel (2005) argued that open source communities are innovation niches that foster widespread group creativity. He defined innovation communities as “nodes consisting of individuals or firms interconnected by information transfer links which may involve face-to-face, electronic, or other communication” (von Hippel, 2005, p. 96). These information transfer links are key features where understanding and fostering effective forms of design rationale can be most productive in enhancing creativity. According to von Hippel, innovation in these communities is a distributed process that occurs through both informal (user-to-user) and organized (users interacting within communities) cooperation. The community supplies users with useful tools and infrastructure that are employed to develop, test, and diffuse their innovations. In some important respects, von Hippel’s distinction between informal and organized information transfer links recalls the distinction, discussed earlier, of formal and informal approaches to design rationale. In the informal conditions of von Hippel’s arguments, and those advanced by this paper, the ad hoc nature is both useful, since it is spontaneous and responsive to opportunity (e.g., to discuss an important issue or capture, in the moment, salient design rationale), and problematic, since the lack of structure can lead to omission. In the organized or formal condition, the rigid context is productive, in that it allows for organizational progress and the capture of widely agreed upon relevant information; yet the formal condition also runs the risk of missing key information that falls outside of its categorization, such as divergent or minority viewpoints. Luther, Kelly, Ziegler, and Bruckman (2010) suggest that the success of creative collaboration in open source communities relies on leaders with solid reputations and respect from their communities, and that Weber’s (2004) principle, “Talk a lot,” should be encouraged.

Within open source communities, diverse levels of domain expertise and transdisciplinary collaboration are leveraged to foster what creativity research would describe as fluency and flexibility. Because these “low-cost innovation niches” (von Hippel, 2005, p. 79) consist of novel combinations of and within preexisting elements and contexts (the members and organizations that exist within the community and the ongoing development of the source code), members typically draw on their own expertise to advance creative solutions. For example, in the realm of open source and do-it-yourself communities, an individual with a background in mountain biking (a hobby) and orthopedic surgery (a profession) may create a seat suspension that reduces shock to a biker’s spine upon landing. Thus, collectively, the community’s members are capable of generating a wide range of ideas (fluency) owing to their diverse backgrounds and a broad diversity in the type of ideas (flexibility). Open source communities, therefore, attract people from various backgrounds who are motivated by the same ideals to create or develop upon existing platforms, whether these are source code or electro-mechanical systems, knitting communities, or so on. The different combinations of backgrounds of the participants in the open source communities and their interconnections are aimed at enhancing the potential quality of the final products. This presents opportunities and challenges for design rationale to act as a cross-pollinator, to bridge the expertise and domain gaps, and to foster transdisciplinary communication.

In open source software development, where the transdisciplinary nature of the community is somewhat less diverse than in the open source or do-it-yourself communities, participants still take on a range of roles that allow them to apply their expertise to a wide range of shared interests and problems that serve the communal goals. To support the sharing of their expertise and advancement of their projects, these communities typically utilize

various forms of version-control tools that facilitate beta testing and revising processes. In open source software developer communities, the volume of individuals who can test and debug code increases the chances that a bug will be found; as Raymond put it, in a phrase that has come to be known as the Linus Law, “given enough eyeballs, the bugs are shallow” (2001, p. 30). The current paradigms for idea exchange in open source software developer communities are through forums or e-mail lists, forms of asynchronous informal communication that aim to focus on single issues in each thread. The main limitations to the effectiveness of current forms of ad hoc open source design rationale within these communities are the duplication and repetition of ideas. For a new developer, therefore, the onus lies in searching and rediscovering whether or not a certain idea or problem has occurred before and what the possible resolution of that idea might have been. New forms of design rationale or strategies for motivating more constructive implementations of design rationale in open source communities may be able to mitigate these limitations. For example, the Wikipedia community uses “barnstars” to reward their members’ effective contributions to articles and commentary (Beschastnikh, McDonald, Zachry, Kriplean, & Borning, 2009). One way to interpret the success of the debugging process in open source communities is that users’ diverse roles, backgrounds, and expertise allow them to view and discuss issues from multiple perspectives. Collective perspectives and implicit analogies enable them to overcome errors and recover from failure.

It is significant to note that design rational systems should include not only the rejected alternatives, but also, and more importantly, the failed implementations that constitute the valuable experiences needed to form the basis of analogies from which further work, progress, and breakthroughs can be advanced. As Dunbar’s (1994) creativity research studies of expert and nonexpert scientific teams show, individuals and teams with more domain-relevant knowledge to draw on can more readily draw parallels between failures and new, more productive domains through analogical reasoning, thus empowering them to overcome setbacks and realize solutions more effectively. In open source communities, users who come from a range of disciplinary backgrounds can use analogies implicitly, both for recognition of bugs as well as for creation of new applications. Advanced design rationale systems have the potential to go beyond their current roles to encourage users to record not just successes and rejected alternatives, but also experiences and reflections of failures as well. This type of design rationale might serve as supporting scaffolding for the development of appropriate analogies and access to generative tools (Gero, 1996; Ishikawa & Terano, 1996). For example, people who download programs could submit a failed version explaining what problems they encountered, how they tried to solve it, if they failed, and why. This record of failures would allow others developers to either not go the same route or take this up as a challenge and introduce improvements and insightful alternatives.

Individuals engaging in transdisciplinary knowledge sharing in open source communities can discover options that were originally neglected. Members play varying roles in groups that may be different from their own personal background and in turn can influence open source successes in creation and rediscovery. The challenge in realizing improved creativity is perhaps in the realization of these benefits; here, design rationale has a significant role to play. Design rationale can empower members of the group to take on, evaluate, and rationalize decisions from new, diverse, and informed perspectives that challenge and provide the community with paths to move their common agenda forward. One of the biggest challenges is the sharing of

unique information that each member possesses in the context of pooling common resources to balance the roles of team decision making. This balance serves both as a means to engender acceptance of, satisfaction with, and commitment to decisions, and to combine disparate points of view, knowledge, and ideas towards better decisions. In contrast to many other group settings, the opportunity to share and capitalize on the benefits of unique information increases with an increase in the number of members in open source communities.

The discussion of creativity and von Hippel's (2005) thesis on democratic innovation within open source communities echoes several of the models of creative processes proposed in decision sciences and creativity literature. For example, according to Cashman and Stroll (1989, p. 136), information technology-based decision processes can be expressed as a "create, communicate, review, and react" action cycle (including awareness management, autonomy, information gathering and dissemination, structuring, modeling options, and execution). Similarly, Shneiderman's (2003) collect, relate, create, and donate stages of creative processes within IT environments and von Hippel's (2005) analysis of open source communities, provide us with insights into the creative processes inherent to these open source communities. As such, these environments provide fertile ground to advance the interplay between creativity research and new forms of design rationale that are fundamentally creativity research tools.

CREATIVITY SUPPORT TOOLS

In order to advance an understanding of the potential of design rationale to act fundamentally as a creativity support tool for open source communities, it is necessary to review recent developments in the realm of creativity research tools. Over the past decade or so, the goals of fostering creativity at the individual and group levels within the context of computing have evolved, furthering the domain of creativity research tools. The Association for Computing Machinery's Conference on Human Factors in Computing Creativity and Interface Workshop in 2002 focused on opportunities to use interface tools in fostering end-user creativity. Common themes of the discussion included interface elements that offered "exploration, parallel experimentation, generative ideation, media and content pliability, iteration, support for creative mistakes and insights and process assistance" (Burlison & Selker, 2002, p. 89). A subsequent National Science Foundation (NSF) Workshop on creativity research tools (Shneiderman et al., 2006) highlighted the potential of creativity research tools as offering "more effective searching of intellectual resources, improved collaboration among teams, and more rapid discovery processes, ... potent support in hypothesis formation, speedier evaluation of alternatives, improved understanding through visualization, and better dissemination of results, ... [to] facilitate exploration of alternatives, prevent unproductive choices, and enable easy backtracking" (p. 62). A set of guidelines for creativity research tools were developed, encouraging a "low threshold, high ceilings, and wide walls" (Shneiderman et al., 2006, p. 70); in other words, easy entry to usage for novices, powerful facilities for sophisticated users, and a small, well-chosen set of features that support a wide range of possibilities, easy exploration of multiple alternatives and powerful history-keeping.

Creativity research tools can take many different forms to support these objectives, encompassing a variety of activities. Shneiderman (2003) demonstrated this in the context of

multiple professional domains and organizations, including architects, lawyers, doctors, and the Compumentor and TechSoup communities. NSF's Creative IT (NSF, 2009), a 3-year funding initiative and the community it fostered, further advanced creativity research tools to explore their role in assessing creativity in everyday activities. Following up on Amabile and Kramer's (2003) study of the creative practices of high tech researchers, creativity investigations have advanced multimodal real-time tools that computationally track affect, voice, and motion "in the wild" and relate these to self- and peer-report measures (Burleson & Pentland, 2008). Similar tools have been advanced to detect diverse affective states, including frustration at 79% accuracy (Kapoor, Burleson, & Picard, 2007). States of frustration and failure in turn present opportunities to promote affective self-awareness and/or algorithmic thinking that have been shown to be instrumental in fostering creative solutions to challenging problems and setbacks (Dunbar, 1994).

Just as open source community programmers and do-it-yourself hackers can be encouraged to use design rationale tools to communicate their reasoning process to others who build upon their code and artifacts, design rationale can also act as a creativity support tool to foster reflection. Empowering users at appropriate times, such as times of frustration and failure, to learn from analogies (Dunbar, 1994) can be conducted by encouraging them to describe their design rationale choices, both for decisions and practices that were eventually implemented and those more exploratory approaches that were not. Further coupling these with the underlying reasoning as to why they were chosen will allow for design rationale tools to provide the creativity support tool features of a low threshold entry for beginners and a high ceiling for experts to encourage a broader engagement by the community. Although this may sound onerous and fine-grained, we are seeing some initial elements of these types of explanations and design rationale emerging within open source communities, such as Linux developers and the do-it-yourself instructables.com hacker postings and their responsive feedback and discussion groups.

Open source and do-it-yourself communities foster broad participation, expertise development, and communication for novice and experts alike. Within the instructables.com community, we see exciting examples of information transfer links that serve as ad hoc design rationale. Take, for example, robonerd's³ "Do It Yourself Arduino or 'The DIY-Duino'" (robonerd, 2010) that describes how to make a version of a popular microcontroller from scratch. In response to a community member's interest in minimizing the board size, robonerd augmented the original tutorial's description to provide additional ad hoc design rationale, explaining his appreciation of the anthropomorphic qualities of the spatial layout of the circuit, "... when you look at it vertically, it kinda looks like a face.... I just couldn't change the look on that face!" (March 7, 2011). There is also evidence that the Linus Law (Raymond, 2001) helped robonerd with debugging, "I see it, SHOOT! I thought you were talking about the elec caps not the ceramics. Crud, I gotta fix that! Thank you very much for the catch! I appreciate it" (March 7, 2011). The multiple perspectives of the community also offer suggestions for new directions. David97 said, "I want to remote control my arduino using my xbox controler (bluetooth). how can I do this?" (March 7, 2011). Motivation for expression of rich design rationale can be sustained through positive feedback and extended mentorship: jpr3 said, "This was a GREAT tutorial. Your web site had each and every step documented! Great work!!!!!" (March 9, 2011). Through a detailed dialogue involving 10 posts, robonerd mentored angelovalorreed until his/her microcontroller worked; robonerd

then agreed to update the instructable to include further detailed suggestions: “I’m going to keep the iable as it is though, because if you use the exact components listed, it works like a charm. Though I will add a suggestion to try the caps you used when in use with that crystal” (March 16, 2011). Throughout this and other such communities, elements of design rationale are affected by the organizational structures (tutorials and comments) and by governance policies, which support a range of formality and granularity. There are typically higher levels of formality and granularity in the tutorials than in the comments. Instructables’ policy that allows individuals to remove posts (erniehatt said, “I removed the comments because I found a couple of errors”; March 17, 2011) allows community members to alter the history of the design rationale. On the one hand, this allows users to correct errors; on the other hand, it may ultimately inhibit the community from learning from failures. Within these forums, design rationale is not always supportive of creativity; as noted above, there are redundancies, nonsequiturs, and even occasional detractors.

Providing design rationale guidelines that encourage developers to encode design rationale for not only the choices they pursue but also for nonelected choices is likely to encourage others to explore a wide range of alternatives (e.g., through analogy), ultimately leading to more successful and creative processes. Such a process of open source community development and appreciation for broader forms of design rationale is likely in turn to foster the wide walls that are necessary for creative exploration. Studying how design rationale can not only foster its own recording and reviewing processes but also how these can be better understood as synergistic with, and indeed as key elements of, creativity support tools within open source communities will help to advance better choices for the implementation of new design rationale strategies. It also will foster stronger understandings of best practices for encoding and disseminating expert (and nonexpert) deliberation and insight, from which a broader community can learn, and support the development and advancement of their creative endeavors. This can be achieved through the development of a deeper understanding of both the processes and consequences (beneficial and detrimental) of design rationale implementation informed by the perspective of creativity research.

The Creativity and Cognition (C&C) conference⁴ community focuses on the nexus of creativity, cognition, design, and emerging technologies. As open source and do-it-yourself communities have emerged, C&C’s interests have included understanding design processes and design rationale related to hybrid physical/digital tools for fabrication and collaboration. A study of the design of “egg drop challenge” devices⁵ showed that providing designers with tools and resources that foster fluency (lots of ideas) and flexibility (a range of diverse ideas) affected their design rationale practices. Individuals who were provided fewer eggs engaged in fewer opportunities to test their designs. Their explanations of their designs and process (their design rationale) revealed that they were less fluent and flexible in their design process and reflection than were their counterparts who were given many eggs (Dow, Heddleston, & Klemmer, 2009). Likewise, their products were less effective and less creative. An explicit example of this can be seen where the mental frame in which designers approach problems, and hence expressed design rationale, is affected by the environment, tools, and resources, in ways that impact elements of their creative process and outcome. This demonstrates that broad access to actualizing resources can directly impact design rationale and the creativity of processes and products (Amabile & Kramer, 2003; Dow et al., 2009). Related work shows that prototyping in parallel is more creative than prototyping sequentially (Dow et al., 2010). Open source software development

environments and gaming worlds also address issues of fluency, flexibility, and prototyping by offering sandbox opportunities that act as “playgrounds” and rapid prototyping environments with which to explore and develop ideas and relationships. Examples of this in the realm of educational gaming include Shute and Becker’s (2010) advancement of 21st century assessment that places an emphasis on the importance of learning to think creatively through data mining of learners’ activities and collaborations in educational gaming environments. Furthering this agenda, Wegerif et al. (2010) have demonstrated the ability to automatically recognize creative reasoning in student e-discussions within in situ dialogue analysis of intelligent tutoring learning environments and their data streams. These examples present opportunities that can inform the development of design rationale implementations as creativity support tools.

COMPUTATIONAL CREATIVITY, EXPERTISE, AND TEAMS

Now that creativity support tools and their potential in design rationale for open source communities have been presented, we explore more recent findings in the area of team and computational creativity research that offer opportunities for future work and development. These should serve as related resources that offer opportunities for the community that is advancing design rationale as creativity support tools.

Team Brainstorming

Smith (2003) proposed that in order to achieve a new way of thinking in a team, members must ignore an existing “fixated” point of view (i.e., sticking with one perspective or idea to the detriment of the overall process) and arrive at a nondominant point of view. The dominant response tends to block minority responses. This characteristic is evident within team interactions when a big idea starts gaining more weight in spite of its possibility of failure or incompatibility with team objectives. This often occurs due to the familiarity (or safety) of the idea, tendencies of teams to want to agree (groupthink), or individuals’ production blocking. Therefore, maintaining and advancing divergent points of view during a large group discussion can be a difficult and daunting task.

In shared brainstorming activities, a high number of ideas often are generated, with one of them subsequently being selected based on discussion of merit with respect to context. A key role of team decision making is to engender acceptance, satisfaction, or commitment to decisions. Smith (2003) suggested that groups should go further, striving to play an important role in combining disparate points of view, knowledge, and ideas towards better decisions. As will be discussed below, open source software communities contend with the interplay of these two roles as they engage in building and generating acceptance for various versions of the open source software code.

One proposed solution is to maintain a log of ideas, avoiding discussion until every idea has been enumerated, such as in electronic brainstorming via individual contributions (Sutton & Hargadon, 1996). However, the simple log that this form of brainstorming creates still requires revisiting each idea, thinking it through at both the individual and group levels, and then deciding on the merits of all of this with respect to the group agenda. This latter process

typically still requires group-level communication, which again is often confounded by fixation and the effects of interpersonal hierarchy.

Recent work on feedback in group settings has shown that higher individual and/or group self-awareness leads to self-directed adaptation of behavior (DiMicco & Bender, 2007). DiMicco and Bender showed that public visualization of group members' verbalization leads to subsequent moderation or improvement of participation, resulting in stronger group outcomes. For example, participants who talk too much will tend to talk less when a bar graph publicly portrays them as an outlier; group members who do not talk enough likewise tend to talk more. For low participating members, talking more fulfills their normative needs, but more importantly provides them with sufficient conversational bandwidth to contribute beneficial information that would otherwise not be available to the group. Such strategies of group self-awareness may also allow higher attention to group processes, goals, and strategies (West, 1996). Gersick and Hackman (1990) found that work groups can break dysfunctional habitual routines by self-reflection. Farooq, Carroll, and Ganoë (2007) found that group self-monitoring can enhance the understanding of breakdowns of creativity and lead to prevention of breakdowns.

Understanding team creativity research with respect to the nature of the work a group shares can provide insights into the functioning of open source communities. Tjosvold (1998) showed that creating a common task that requires collective action among members of a team can induce cooperative orientation, thereby promoting resource and information exchanges as well as openness to each other's ideas. Similarly, Wageman (1995) found that teams employing task interdependence increased the need for collaboration and mutual adjustment among members by raising the collective sense of responsibility among team members. Thus, we may conclude that the overall success of team creativity can be ensured by creating conditions of common fate with rewards and/or task interdependence. While the specifics are not well understood, interesting insights about the importance of leadership and active participation are emerging (Luther et al., 2010). These conditions of common fate along with cooperative orientation help to drive today's open source culture.

Thus, a significant opportunity for creativity support in open source communities is to sufficiently reduce sources of inhibition such that each member may engage in adequate expression of ideas. Through the customization and promotion of environments that encourage design rationale expressed through information transfer links, users can be encouraged to share their unique contributions. Tools that support various means to enhance the discussion of individuals' ideas without exacerbating team members' inhibitions are likely to lead to increased team creativity.

Influence of Expertise and Computational Systems

Building on the context of creativity support in teams, we will now discuss expertise with respect to team and computational creativity. In addition, we address how improved understandings of dynamics within their processes might inform implementation of design rationale as a creativity support tool in groups.

Atman et al.'s (2007) research on individual problem solving demonstrated that experts engage in iterative processes—ranging across information gathering, problem definition, modeling, evaluation, reflection, and so forth—that are richer than those of their novice counterparts. Experts engage in activities that allow them to accumulate experience, reflect on

them, and transfer their experience and knowledge between diverse stages and activities within design processes; they also engage in significantly more reflection than novices. Here we see that level of expertise impacts both design rationale practices within design process and the outcomes of these open-ended creative activities. Similarly, organizational approaches to design (e.g., organizational processes that pursue iterative design vs. sequential design, sometimes referred to as the waterfall model) impact design rationale and creative outcomes. We have conducted pilot studies applying Atman and colleagues' approach to groups engaged in design processes, exploring the question: If a weaker designer joins a team, does the team become weaker or does the weaker designer rise to the occasion and improve his/her performance (Burleson, 2007)? Due to the complexities of conducting team studies, this remains an open question. We also currently are exploring the potential for an embodied agent to participate as a virtual facilitator to prompt shifts in individual or team activities.

Similar processes and questions arise in the realm of efforts to advance computational creativity and its interactions with individuals and integration with teams, for instance in systems aimed at demonstrating expertise and fostering effective team collaboration and creativity. Buchanan (2001) approached computational creativity in terms of metacreativity. He argued, as summarized in Burleson (2005, p. 443), that programs should "provide the ability for the AI [artificial intelligence] to accumulate past experiences and information, reflect on them, and transfer this information throughout the system, as a means for enhancing creative collaboration between machine and user." Even in computational systems that do not yet operate with Buchanan's (2001) metacapacity, there is strong evidence for the potential of creative systems to play a significant role within hybrid teams. Goldenberg, Mazursky, and Solomon's (1999). "Creative Sparks" research, published in *Science*, demonstrated that a computerized routine (one easily algorithmically implemented by humans) "produces solutions consensually judged to be more creative than those achieved by humans" (Burleson, 2005, p. 443). Yet when this algorithm was made available to human teams, they failed to recognize or realize its benefits, opting instead to pursue their task without the aid of the computational creativity system. This example shows that even when computational creativity is highly capable, there is still significant work to be done on the social elements of human-computer interaction to encourage its acceptance by a human team. This example also shows that the team was not sufficiently appreciative of the algorithm as one of its actualizing resources to advance creativity. The creativity process and its outcome could be improved by a design rationale implementation that had sufficient formality and granularity to encourage effectively and persuade the team to record and reflect on the creative resources and concepts at its disposal, and with decision-making support that helped individuals and teams select the most creative ideas (regardless of their origin).

Ultimately, given the advances in computational creativity, humans and computers could work on hybrid teams to foster creativity. Facilitating a frame of collaboration between the creativity support tool and humans is one of the ongoing opportunities for design rationale and creativity support tools, and advances the Computing Research Association's (2002) Grand Challenges. The range of attributes that we find in creativity support tool approaches, if applied to design rationale at the individual and group levels, would arguably have the potential to enable design rationale to serve as a creativity support tool to foster higher levels of creativity, in both processes and outcomes. Since many of the approaches discussed above lay the foundation for enhancing creativity through design rationale at the individual and team levels, they also

hold important strategies for implementation of design rationale that supports creativity and acts as creativity support tool within open source communities. Specifically, the creativity support tool guidelines (low thresholds, wide walls, powerful history-keeping, etc.) could be applied throughout open source community information transfer links, within the user or system interface, as a community-level guideline, and as a design rationale philosophy promoted by the community. Merging Shneiderman's (2003) collect, relate, create, and donate approach to systems that foster creativity with Amabile et al.'s (2002) and Amabile & Kramer's (2003) findings from organizational behavior studies—specifically by sensing elements of organizational behavior and understanding positive affect and frustration, time pressure and interruption—could guide, tailor, and refine open source communities' implementation of design rationale as creativity support tool. Open source and do-it-yourself communities that employ actualizing resources and rewards consistent with lessons from creativity research and from individual and team expertise can more effectively use design rationale to enhance the fluency and flexibility elements of creativity. Design rationale can play an important role in understanding better ways to incorporate technological support (computational creativity and creativity support tools) in hybrid teams.

CONCLUSION

Applying findings from creativity research and recent efforts that have advanced creativity support tools has the strong potential to realize significant advances to design rationale, both in terms of evaluating diverse implementations of design rationale for their ability to foster creativity and toward transforming existing design rationale tools into creativity support tools. These lessons can and should be used to guide efforts to transform existing design rationale tools into design rationale–creativity support tools. In this manner, a range of organizational practices and innovative interfaces that include appropriate levels of granularity and formality can foster creativity through novel design rationale implementations and enhancements. These might include features that encourage metacreativity and promote users' ability to engage in the expert practices and rich processes that emphasize reflection, transfer, learning, and recovering from failure through analogy. They might also identify times of failure and frustration, and might create sandboxes or equivalent features that foster fluency and flexibility, providing low thresholds, wide walls, and actualizing rewards for creativity. They might minimize time pressure and promote opportunities for positive affect and productive social interaction. While these strategies can be employed throughout programming, design practice, and indeed in any context in which design rationale may apply, they are particularly relevant to open source software and do-it-yourself communities that are highly active in advancing new forms of creative collaborations and creative IT endeavors.

ENDNOTES

1. Multitouch is a technology that allows Apple products to recognize when two or more points are in contact with its surface, enabling two-finger scrolling, rotation, zoom, etc.
2. The Free Software Foundation is a nonprofit that campaigns for free open source software and drives development of the GNU license. For more information, go to <http://www.fsf.org>

3. The quotes excerpted from the Do It Yourself Arduino (The DIY-Duino) have been quoted exactly as written. These forum comments took place during March 2010, and the dates of specific comments are included within parentheses.
4. Creativity & Cognition is an annual conference run by the Association of Computing Machinery, which brings together professionals from diverse fields to discuss the depth and breadth of human creativity.
5. The Egg Drop challenge is a popular engineering challenge in which participants are given a limited set of materials (often straws, paper, or toothpicks), and asked to create a device that will allow an egg to be dropped from a specified height without breaking.

REFERENCES

- Amabile, T. M. (1983). Social psychology of creativity. *Journal of Personality and Social Psychology*, 45, 357–376.
- Amabile, T. M. (1996). *Creativity in context: Update to the social psychology of creativity*. Boulder, CO, USA: Westview Press.
- Amabile, T. M., Hadley, C. N., & Kramer, S. J. (2002). Creativity under the gun. *Harvard Business Review*, 80(8), 52–61.
- Amabile, T. M., & Kramer, S. J. (2003, October). *The best (and worst) days in creative project teams: Some preliminary results*. Paper presented at the Society for Experimental Social Psychology, Boston, MA, USA.
- Amabile, T., & Mueller, J. (2002). Assessing creativity and its antecedents: An exploration of the componential theory of creativity. In C. Ford & N. Mahwah (Eds.), *Handbook of organizational creativity* (pp. 33–64). London: Lawrence Erlbaum Associates.
- Atman, C. J., Adams, R. S., Cardella, M. E., Turns, J., Mosborg, S., & Saleem, J. (2007). Engineering design processes: A comparison of students and expert practitioners. *Journal of Engineering Education*, 96, 359–379.
- Beschastnikh, I., Kriplean, T., & McDonald, D. (2008, March–April). *Wikipedian self-governance in action: Motivating the policy lens*. Paper presented at the Second International Conference on Weblogs and Social Media, Seattle, Washington, USA.
- Beschastnikh, I., McDonald, D. W., Zachry, M., Kriplean, T., & Borning, A. (2009, May). *Promoting quality in Wikipedia through enculturation*. Paper presented at the Supporting the Amateur Workshop at the ACM 2009 International Conference on Supporting Group Work, Sanibel Island, FL, USA.
- Buchanan, B. (2001). Creativity at the metalevel: AAAI-2000 presidential address. *AI Magazine*, 22(3), 13–28.
- Burleson, W. (2005). Developing creativity, motivation, and self-actualization with learning systems. *International Journal of Human-Computer Studies*, 63, 436–451.
- Burleson, W. (2007, May). *Opportunities for ubiquitous design environments in a flat world*. Paper presented at the Harvey Mudd Design Workshop, Claremont, CA, USA.
- Burleson, W., & Pentland, S. (2008). NSF Award 0846148: SGER: Human-Centered Computing: Creativity in IT Research Organizations. Award notification retrieved August 24, 2011, from <http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0846148>
- Burleson, W., & Selker, T. (2002). Introduction: Creativity and interface. *Communications of the ACM*, 45(10), 88–90.
- Buxton, B. (2007). *Sketching the user experience*. San Francisco: Morgan Kaufmann Publishers.
- Carroll, J. (2010). The essential tension of creativity and rationale in software design. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*, 6, 4–10.
- Cashman, P. M., & Stroll, D. (1989). Developing the management systems of the 1990s: The role of collaborative work. In M. H. Olson (Ed.), *Technological support for group collaboration* (pp. 129–146). Mahwah, NJ, USA: Lawrence Erlbaum Associates Inc.
- Certeau, M. D. (1984). *The practice of everyday life*. Berkeley, CA, USA: University of California Press.

- Conklin, J. (2003). Dialog mapping: Reflections on an industrial strength case study. In P. Kirschner, S. J. B. Shum, & C. S. Carr (Eds.), *Visualizing argumentation: Software tools for collaborative and education sense making* (pp. 117–136). London: Springer-Verlag.
- Computing Research Association (2002). *Grand Research Challenges in Information Systems*. Retrieved August, 6, 2011, from <http://archive.cra.org/reports/gc.systems.pdf>
- Corbet, J., Kroah-Hartman, G., & McPherson, A. (2010, December). *Linux kernel development: How fast it is going, who is doing it, what they are doing, and who is sponsoring it* [White paper]. San Francisco: The Linux Foundation. Retrieved August 24, 2011, from http://www.linuxfoundation.org/docs/lf_linux_kernel_development_2010.pdf
- Csikszentmihalyi, M. (1996). *Creativity: Flow and the psychology of discovery and invention*. New York: Harper Perennial.
- DiBona, C., Ockman, S., & Stone, M. (1999). *Open sources: Voices from the open source revolution*. Sebastopol, CA, USA: O'Reilly Media.
- DiMicco, J. M., & Bender, W. (2007). Group reactions to visual feedback tools. In Y. de Kort, B. J. Fogg, W. IJsselsteijn, B. Eggen & C. Midden (Eds.), *Proceedings of The Second International Conference on Persuasive Technology* (pp. 132–143). Berlin, Germany: Springer.
- Dow, S. P., Glassco, A., Kass, J., Schwarz, M., Schwartz, D. L., & Klemmer, S. R. (2010). Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction*, 17(4), 18:1–18:24. doi: 10.1145/1879831.1879836
- Dow, S. P., Heddleston, K., & Klemmer, S. R. (2009). The efficacy of prototyping under time constraints. In N. Bryan-Kinns, M. D. Gross, H. Johnson, J. Ox, & R. Wakkary (Eds.), *Proceedings of the 7th Conference on Creativity & Cognition* (pp. 165–174). New York: ACM Press.
- Dunbar, K. (1994). How scientists really reason: Scientific reasoning in real-world laboratories. In R. J. Sternberg & J. Davidson (Eds.), *The nature of insight* (pp. 365–395). Cambridge, MA, USA: MIT Press.
- Farooq, U., Carroll, J. M., & Ganoë, C. H. (2007, November). *Supporting creativity with awareness in distributed collaboration*. Paper presented at the International ACM Conference on Supporting Group Work, Sanibel Island, FL, USA.
- Fischer, G., Lemke, A., McCall, R., & Morch, A. (1991). Making argumentation serve design. In T. H. Moran & J. Carroll (Eds.), *Design rationale: Concepts, techniques, and use* (pp. 267–294). Hillsdale, NJ, USA: Erlbaum.
- Gardner, H. (1994). *Creating minds: An anatomy of creativity seen through the lives of Freud, Einstein, Picasso, Stravinsky, Eliot, Graham, and Gandhi*. New York: Basic Books.
- Gero, J. S. (1996). Creativity, emergence and evolution in design: Concepts and framework. *Knowledge-Based Systems*, 9, 435–448.
- Gersick, C. J. G., & Hackman, J. R. (1990). Habitual routines in task-performing groups. *Organizational Behavior and Human Decision Processes*, 47, 65–97.
- Goldenberg, J., Mazursky, D., & Solomon, S. (1999). Creative sparks. *Science*, 258, 1495–1496.
- Gruber, T. R., & Russell, D. M. (1996). Generative design rationale: Beyond the record and replay paradigm. In T. P. Moran & J. Carroll (Eds.), *Design rationale: Concepts, techniques, and use* (pp. 323–349). Hillsdale, NJ, USA: Erlbaum.
- Ishikawa T., & Terano T. (1996). Analogy by abstraction: Case retrieval and adaptation for inventive design expert systems. *Expert Systems with Applications*, 10, 351–356.
- Kapoor, A., Bursleson, W., & Picard, R. (2007). Automatic prediction of frustration. *International Journal of Human-Computer Studies*, 65, 724–736.
- Kuznetsov, S., & Paulos, E. (2010, October). *Rise of the expert amateur: DIY projects, communities, and cultures*. Paper presented at the ACM NordiCHI conference, Reykjavík, Iceland.
- Luther, K., Kelly, C., Ziegler, K., & Bruckman, A. (2010, November). *Why it works (when it works): Success factors in online creative collaboration*. Paper presented at the Group 2010 conference, Sanibel, FL, USA.

- MacLean, A., Young, R. M., Bellotti, V. M. E., & Moran, T. P. (1996). Questions, options, and criteria: Elements of design space analysis. In T. P. Moran & J. Carroll (Eds.), *Design rationale: Concepts, techniques, and use* (pp. 53–106). Hillsdale, NJ, USA: Erlbaum.
- Mitchell, W. J., Inouye, A. S., Blumenthal, M. S., & the National Research Council. (2003). *Beyond productivity: Information technology, innovation and creativity*. Washington DC, USA: National Academics Press.
- Moran, T., & Carroll, J. (Eds.). (1996). *Design rationale: Concepts, techniques, and use*. Hillsdale, NJ, USA: Erlbaum.
- National Science Foundation (NSF). (2009). Creative IT Program Solicitation. Retrieved August 1, 2011, from <http://www.nsf.gov/pubs/2009/nsf09572/nsf09572.pdf>
- O'Mahony, S., & Ferraro, F. (2007). The emergence of governance in an open source community. *Academy of Management Journal*, 50, 1079–1106.
- Raymond, E. S. (2001). *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. Sebastopol, CA, USA: O'Reilly Media.
- Rittel, H. (1972). On the planning crisis: Systems analysis of the 'first and second generations.' *Bedriftskonomen*, 8, 390–396.
- robonerd (2010, March 7–March 17). Do it yourself Arduino or The DIY-Duino. [On-line forum comments]. Retrieved August 24, 2011, from <http://www.instructables.com/id/DIY-Arduino-or-The-DIY-Duino/>
- Shneiderman, B. (2003). *Leonardo's laptop: Human needs and the new computing technologies*. Cambridge, MA, USA: MIT Press.
- Shneiderman, B., Fischer, G., Czerwinski, M., Resnick, M., Myers, B., Candy, L., Edmonds, E., Elsenberg, M., Giaccardi, E., Hewett, T., Jennings, P., Kules, B., Nakakoji, K., Nunamaker, J., Pausch, R., Selker, T., Sylvan, E., & Terry, M. (2006). Creativity support tools: Report from a U.S. National Science Foundation sponsored workshop. *International Journal of Human-Computer Interaction*, 20, 61–77.
- Shute, V., & Becker, B. (2010). Prelude: Assessment for the 21st century. In V. Shute & B. Becker (Eds.), *Innovative assessment for the 21st century: Supporting educational needs* (pp. 1–12). New York: Springer Science+Business Media, LLC.
- Smith, S. M. (2003). The constraining effects of initial ideas. In P. B. Paulus & B. A. Nijstad (Eds.), *Group creativity* (pp. 15–31). Oxford, UK: Oxford University Press.
- Stark, M. (2003, July 7). The organizational model for open source: Q&A with Siobhán O'Mahony. Retrieved August, 6, 2011, from <http://hbswk.hbs.edu/item/3582.html>
- Sternberg, R. J., & Lubart, T. I. (2007). The concept of creativity: Prospects and paradigms. In R. J. Sternberg (Ed.), *Handbook of creativity* (pp. 3–15). Cambridge, MA, USA: Cambridge University Press.
- Sutton, R., & Hargadon, A. (1996). Brainstorming groups in context: Effectiveness at a product design firm. *Administrative Science Quarterly*, 41, 685–718.
- Tjosvold, D. (1998). Cooperative and competitive goal approach to conflict: Accomplishments and challenges. *Applied Psychology*, 47, 285–313.
- von Hippel, E. (2005). *Democratizing innovation*. Cambridge, MA, USA: MIT Press.
- von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: A case study. *Research Policy*, 34, 1217–1242.
- Wageman, R. (1995). Interdependence and group effectiveness. *Administrative Science Quarterly*, 40, 145–180.
- Wakkary, R., & Maestri, L. (2007). The resourcefulness of everyday design. In *Proceedings of Creativity and Cognition 2007* (pp. 163–172). New York: ACM Press.
- Weber, S. (2004). *The success of open source*. Cambridge, MA, USA: Harvard University Press.
- Wegerif, R., McLaren, B. M., Chamrada, M., Scheuer, O., Mansour, N., Mikšátko, J., & Williams, M. (2010). Exploring creative thinking in graphically mediated synchronous dialogues. *Computers & Education*, 54, 613–621.
- Wentworth, R. (1978). Making do and getting by. *Artscribe*, 14, 21–23.

- Werner, K., & Rittel, H. (1979). *Issues as elements of information systems*. (Working paper No. 131). Heidelberg, Germany: Studiengruppe für Systemforschung. (Original work published 1970)
- West, M. A. (1996). Reflexivity and work group effectiveness: A conceptual integration. In M. A. West (Ed.), *Handbook of work-group psychology* (pp. 555–579). Chichester, UK: Wiley.
-

Authors' Note

We thank John Carroll and Siobhan O'Mahony for their feedback and encouragement and Jamie Wernet, Sonia Shah, Margaret Pingolt, and Daniel Davis for their help in preparing this paper.

All correspondence should be addressed to:
Winslow Burleson
School of Computing, Informatics, and
Decision Systems Engineering
Arizona State University
699 S. Mill Avenue
Tempe AZ, 85281
Winslow.Burleson@asu.edu

Human Technology: An Interdisciplinary Journal on Humans in ICT Environments
ISSN 1795-6889
www.humantechnology.jyu.fi