

# Redistributing Leadership in Online Creative Collaboration

Kurt Luther,<sup>1</sup> Casey Fiesler,<sup>2</sup> Amy Bruckman<sup>2</sup>

<sup>1</sup> HCI Institute

<sup>2</sup> GVU Center

Carnegie Mellon University  
5000 Forbes Ave., Pittsburgh, PA 15213 USA  
kluther@cs.cmu.edu

Georgia Institute of Technology  
85 5<sup>th</sup> St. NW, Atlanta, GA 30332 USA  
casey.fiesler@gatech.edu, asb@cc.gatech.edu

## ABSTRACT

In this paper, we integrate theories of *distributed leadership* and *distributed cognition* to account for the roles of people and technology in online leadership. When leadership is distributed effectively, the result can be success stories like Wikipedia and Linux. However, finding a successful distribution is challenging. In the online community Newgrounds, hundreds of collaborative animation projects called “collabs” are started each year, but less than 20% are completed. We suggest that many collabs fail because leaders are overburdened and lack adequate technological support. We introduce Pipeline, a collaboration tool designed to support and transform leadership, with the goal of easing the burden on leaders of online creative projects. Through a case study of a six-week, 30-artist collaboration called Holiday Flood, we show how Pipeline supported redistributed leadership. We conclude with implications for theory and the design of social computing systems.

## Author Keywords

Social computing; online creative collaboration; creativity support tools; distributed leadership; distributed cognition.

## ACM Classification Keywords

H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces --- *collaborative computing; computer-supported cooperative work; theory and models; web-based interaction.*

## INTRODUCTION

Effective leadership is a key ingredient in many forms of successful online creative collaboration. Many open-source software (OSS) projects are headed by “benevolent dictators” such as the Linux kernel’s Linus Torvalds [33]. Polymath, the online collaboration that solved a significant open problem in mathematics, was led by two prominent mathematicians [7]. Collaborative art projects such as the *Virtual Choirs* [1], *Star Wars Uncut* [31], or *The Johnny Cash Project* [11] are as often credited to their auteurs as the crowds who authored the content.

These projects are well known because they were successful. Their creators were effective at recruiting and motivating volunteers, providing a creative vision, dividing up work, and other essential leadership behaviors. But what happens when leaders aren’t as effective? In the Newgrounds community of Flash animation enthusiasts, hundreds of collaborative projects called “collabs” are started each year, but leaders of these projects frequently become overburdened, and less than 20% result in a finished animation [21,22].

We might look to leadership theory to explain why such failures occur, but theories of leadership in online creative collaboration are not yet well developed, particularly with respect to designing technological support. One promising option is *distributed leadership* (DL), a theory originating in education research that construes leadership as a set of behaviors that can be distributed across all members of a group, not just formal leaders [13,32]. While DL is well suited for analyzing leadership structures in traditional organizations, its ability to account for the role of technology or provide design implications is limited. To address this limitation, we integrate DL with *distributed cognition* (DCog), a theory embraced by the CSCW community for analyzing how cognitive processes can be distributed across people, artifacts, and time [17,19].

Using this theoretical framework, we argue that many Newgrounds collabs failed because leaders were overburdened and unable to effectively redistribute their responsibilities among other members or technologies. We propose that technology designed for *redistributing leadership* could ease the burden on leaders and allow for more complex creative projects. To explore this idea, we created Pipeline, a web-based tool for supporting and transforming leadership in online creative projects. We present a detailed case study of one Pipeline project, a collaborative artwork called Holiday Flood, to show how Pipeline supported and transformed leadership. We conclude with implications for theory and design of social computing systems.

This paper makes three main contributions to CSCW. First, it synthesizes the relative strengths of DL and DCog to provide a more complete theoretical perspective on online leadership that accounts for both the role of technology and the experiences of people. Second, it presents a new system and design implications for redistributing leadership in online creative collaboration when existing leadership

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CSCW '13, February 23–27, 2013, San Antonio, Texas, USA.  
Copyright 2013 ACM 978-1-4503-1331-5/13/02...\$15.00.

configurations are ineffective. Third, it provides a detailed case study of how such a system is used in a real-world context to illustrate its strengths and areas for improvement.

## BACKGROUND

### Distributed Leadership

A growing body of research has challenged the premise of a “heroic” individual leader and instead argued that leadership is often distributed across members of a group. Distributed leadership (DL) decouples leadership roles from leadership behaviors, proposing that group members without formal or “legitimate” leadership status can and do act as leaders [32]. The focus then shifts to understanding how, why, when, etc. certain members perform certain leadership behaviors. Most research under the DL banner has been conducted by United Kingdom-based scholars and focuses on traditional organizations in education and healthcare [5]. Computer-mediated contexts such as online creative collaboration have not yet received much attention from DL scholars. A notable exception is Zhu et al.’s investigation of shared leadership in Wikipedia [35,36]. They found that while “legitimate” (formally appointed) leaders like administrators are more influential overall, regular members also substantially influence others via leadership behaviors.

While not discussed in terms of DL, other research has shown evidence of distributed or decentralized leadership in online creative collaboration. Wikipedia has a “benevolent dictator,” co-founder Jimmy Wales, but governance on the site has become decentralized over time, with sub-communities adopting their own, sometimes conflicting, policies [10]. Like Wikipedia, many popular open-source software projects, such as the Linux kernel, embrace more decentralized governance over time [33]. Projects like the Apache web server embrace council-style leadership structures, where authority rotates among a small group [24], and research suggests that OSS projects with more utility-oriented goals lend themselves to decentralized leadership [26]. Finally, a study of the online mathematics collaboration Polymath found that though the project’s two formal leaders were essential to its success, regular members also made significant contributions [7]. Our work extends this prior research by exploring how it could inform the design of technological support for online leaders.

### Distributed Cognition

Distributed cognition (DCog) is a theory about cognition “in the wild”; i.e., how information is gathered and processed in the real world [19]. It asserts that cognition is a social, embodied process, and cannot be divorced from the culture in which it is situated. The theory considers systems of functional relationships between components, which can include individuals, groups, tools, computers, and other elements, and is chiefly concerned with how information moves between them, and over time.

Distributed cognition’s origins lie in cognitive science, as a response to a prevailing school of thought holding that cognition takes place only inside the human brain. Hollan et al. [17] argued that distributed cognition could also serve as a new foundation for HCI research and design. Halverson [16] analyzed the utility of distributed cognition in the context of CSCW, comparing it to activity theory, and identified some key advantages of distributed cognition. Distributed cognition is flexible, allowing the unit of analysis to be determined by the phenomenon, and including both people and artifacts in analyses without privileging either. Distributed cognition is also highly data-driven and presents data in a way that is convenient for informing system design.

These attributes make distributed cognition a valuable theory for understanding collaborative processes when a large proportion of communication is mediated, such as in online communities. Recently, scholars have begun to think about online creative collaboration, such as Wikipedia, in terms of distributed cognition. For example, Geiger and Ribes [12] analyzed vandalism detection on Wikipedia—a system of vandals, vandal fighters, and software tools—and argued that non-human agents such as vandalism-detection bots occupy significant social roles.

### Integrating DL and DCog

DL and DCog originated in different disciplines and have generally been considered separately. In this paper, we argue that combining elements from both theories can offer new insights into the design and study of social computing systems.

DL and DCog share some core concepts but there are notable differences as well. One similarity is that both theories assume a unit of analysis that is dependent on context, dynamically reconfigurable, and informed by empirical data. For DCog, the unit of analysis is a cognitive process and it is “delimited by the functional relationships among the elements that participate in it” [17]. For DL, the unit of analysis is the activity of leadership, which is composed of leadership behaviors that can be performed by any member of a group. DCog and DL are also similar in that both seek to broaden the unit of analysis beyond the individual. DL “does not privilege the work of particular individuals or categories of persons, nor is there a presumption about which individual’s behavior carries more weight with colleagues” [13]. It also “allows for the possibility that all organization members may be leaders at some stage” [13]. Similarly, DCog “does not privilege the individual” [16] and holds that cognitive processes can be “distributed across members of a social group” [17].

DCog differs from DL in that it allows for cognitive processes to be distributed across artifacts and time as well as people. Consequently, it has proven especially applicable for analyzing systems that include a substantial technological component [16,17]. DL remains focused on

people, and though it acknowledges that leadership can change over time [13], it does not account for the role of non-human systems in leadership. Researchers have also established DCog as a valuable source of design considerations for new systems (e.g. [4]). While DL may also make valuable contributions to system design, we were unable to find any examples of such attempts.

On the other hand, because DL is more human-centered than DCog, it is better suited for providing insight into *why* the human elements of the system behave the ways they do. DL studies typically involve in-depth, phenomenological interviews with leaders that explore both the rationale behind decisions as well as their practical effects. This includes, for example, the role of power and influence, though this is still a nascent area of research [14]. DL studies have also sought to identify different patterns of leadership and understand the sociocultural forces that created them (e.g. [5,8,13]). Because DCog is concerned mainly with functional relationships among elements, it is only able to account for the externally observable consequences of emotions, motivations, etc. experienced by individuals within the system [16].

In summary, both DL and DCog assume a unit of analysis that is dependent on context, informed by empirical data, changing over time, and broader than the individual. They differ in that DL is more human-centered, providing greater insight into the “whys” of human behavior in a system, while DCog, which includes non-human artifacts in its analysis, is better equipped to provide design implications. We propose that an integrated theory combining the strengths of DCog and DL can bring a valuable new perspective to social computing research and design.

### Leadership Behaviors

As mentioned above, DL fundamentally argues that leadership roles can be separated from leadership behaviors. Therefore, in order to analyze DL in a particular context, it is necessary to first identify which leadership behaviors are the focus of the analysis. Throughout this paper, we discuss leadership using the five behaviors for managing work identified by Yukl, in his review of leadership research in organizational behavior [34]. These task-oriented behaviors—planning, problem solving, clarifying, informing, and monitoring—reflect the broader theoretical view of leadership as a process of initiating structure for followers. In the rest of the section, we briefly define each of the behaviors according to Yukl.

#### Planning

According to Yukl, planning is “deciding what to do, how to do it, who will do it, and when it will be done” [34]. The activity of planning is generally a cognitive one, though the leader may involve others in the planning process. Thus, it is easiest to identify planning when the leader is actually implementing the plans. Planning takes many forms, including strategic and operational planning. Two sub-types

with particular relevance to this paper are action planning, defined as “determining how to implement a strategy or carry out a project in an effective manner,” and contingency planning, which is “the development of procedures for reacting...to unavoidable problems and crises.” In contrast to problem solving (discussed below), planning is not reactive and has a long-term orientation [34].

#### Problem Solving

The leadership behavior of problem solving “involves identifying work-related problems, analyzing them in a systematic but timely manner, and acting decisively to implement solutions and deal with crises” [34]. Unlike planning, which has a long-term orientation, problem solving is about dealing with emergencies and other situations that occur unexpectedly and require immediate attention. Leaders who are effective problem solvers are especially apt at being decisive and taking responsibility for a problem rather than avoiding it or excessively “buck-passing” (making somebody else responsible) [34].

#### Clarifying Roles and Objectives

Clarifying, according to Yukl, “is the communication of plans, policies, and role expectations” [34]. He identifies four sub-types of clarifying behaviors: (1) defining job responsibilities, (2) setting performance goals, (3) assigning tasks, and (4) providing instructions for doing a task. This latter sub-type can involve behaviors such as providing feedback on members’ work and offering demonstrations of best practices [34].

#### Informing

The leadership behavior of informing “involves the communication of task-relevant information needed by subordinates, peers, superiors, or outsiders” [34]. A leader who is an effective informer is able to expose project members to the information they need and avoid wasting their time with information that isn’t relevant to them. The leader is essentially a curator, processing all the information that passes through the project, pulling out the bits that are meaningful to certain members, and sharing it with them in a timely fashion [34].

#### Monitoring

Monitoring “involves gathering information about the operations of the manager’s organizational unit,” including elements such as progress, individual performance, and the success of various endeavors [34]. Monitoring provides the information to be used for the planning and problem solving behaviors described above.

In the following section, we describe the context of our study, Newgrounds collabs, and explain how leaders of these projects struggle to perform these behaviors.

### STUDY CONTEXT: COLLABS

The context of our research into distributed leadership is the “collab,” a collaborative project organized over the Internet

with an artistic, expressive, or entertainment-oriented goal, such as a movie, game, or artwork. Previous work focused on the challenges for collab leaders on Newgrounds,<sup>1</sup> a popular online community of Flash animation enthusiasts [21,23]. A follow-up study examined success factors in collabs and found that more than 80% of collabs fail to produce a completed work [22]. This percentage suggests significant opportunities for improvement, as most collab members define success as completing the project.

Why do so few collabs succeed? First, collabs bring with them a number of challenges common to artistic and expressive collaboration in general. These include *completion* (work can only be made public if it is finished and polished); *originality* (members strive for originality, which is necessarily riskier because it eschews familiar conventions [2]); *subjectivity* (fundamentally subjective media like visual art and storytelling rarely have one clear best approach); and *ownership* (members must reconcile their own creative visions with the group's best interest) [21].

These general challenges of creative collaboration are complicated by the challenges of online, distributed collaboration and other characteristics of the Newgrounds community. Distance matters, and collab work, like other types of collaborative design, is often tightly coupled [27]. Collabs also require diverse groups to find common ground and work together: amateurs and professionals, novices and experts, artists and programmers. Furthermore, collab members are volunteers, so they frequently drop out when work or “real life” take priority.

Finally, the technology used by Newgrounds members to organize collabs is poorly suited to the task. Most collabs are organized in the Newgrounds forums—one thread per collab—which were originally designed for conversation but have since been appropriated for collaboration. Threads quickly balloon to dozens or hundreds of replies, becoming almost impenetrable to newcomers. The chronological ordering of replies causes individual posts to be quickly buried beneath a flurry of unrelated replies, and the forums support minimal file sharing (one image upload per post).

In this environment, leaders struggle to perform leadership behaviors. *Monitoring* members' progress is already difficult because geographic distribution makes direct observation impossible. Furthermore, as Yukl notes, monitoring is “most difficult when work involves unstructured, unique tasks for which results can be determined only after a long time interval” [34], a description that applies to the animation and visual art requirements of many collabs. Since monitoring supplies the information for *planning* and *problem solving*, a lack of information makes these leadership behaviors difficult to perform. Leaders consume large amounts of time *informing*

and *clarifying* because members miss answers to their own questions or other relevant posts, and the forum software prevents leaders from editing their own posts to add clarification. Consequently, leaders must frequently repost the same information or deal with the resulting confusion.

In addition to these leadership behaviors, collab leaders also assume additional responsibilities not normally associated with leadership (e.g. managing project files). Thus, many leaders propose relatively simple projects and adopt centralized, top-down leadership styles, simply to minimize coordination and communication. For example, one of the most popular collab structures requires members to submit short animations based around a theme, which the leader edits into a single movie. This simple structure allows members to work almost entirely independently and in parallel.

These challenges weigh on all collab members, but most intensely on the leader, who, in a vicious cycle, typically invests the most time and effort in the collab and therefore has the most to lose. It follows that the leader is also the most likely member of the collab to become overburdened. When leaders become overwhelmed, they often quit, and their collabs fail shortly afterward. We suggest that one of the most promising avenues for improving collab success is to focus on the leader, who is both the most influential member of the collab and the most likely to benefit from a design intervention. *Our hypothesis is that new technology can be designed to ease the burden on collab leaders by supporting and transforming the work they do as leaders.* Specifically, the technology could *support* leadership by helping leaders perform their roles more easily, and *transform* leadership by redistributing some leader responsibilities more broadly across the group.

Better technological support could ease the burden on leaders, increase communication between members, open up new leadership styles, and result in more successful, complex, creative projects. To explore these ideas, we built and launched Pipeline, a web-based collaboration tool for supporting and transforming leadership. In the next section, we describe the Pipeline system and explain how its major features are designed with redistributed leadership in mind.

## PIPELINE

### Designing for Decentralization and Automation

Our goal with Pipeline was to provide a tool that helps online leaders redistribute some of their responsibilities in order to avoid becoming overwhelmed. The literature on DCog and DL provides some theoretical insights into how to accomplish this. From DL and DCog, we borrow the idea that leadership and cognitive processes can be distributed across members of a social group. This suggests *decentralization* as a design principle: Pipeline should allow leaders to easily share their responsibilities with other members of the project. From DCog, we borrow the idea that cognitive processes can also involve coordination with

<sup>1</sup> <http://www.newgrounds.com/>

The screenshot shows the Pipeline tasks system interface. At the top, there's a navigation bar with 'Pipeline' and links to 'Start a Project', 'Find Projects', 'Help', and 'Admin'. On the right, user info 'johndoe' and links to 'Inbox', 'Settings', and 'Log Out' are visible. Below the navigation bar, the main header shows 'Holiday Flood' with a 'FINISHED' status and a message 'You are a member of this project.' with a 'Leave Project' button. A tabbed interface below the header includes 'Basics', 'Tasks', 'People', 'Discussions', 'Files', and 'Activity'. The 'Tasks' tab is active, showing a breadcrumb 'Tasks » WIPs » Partridge in a Pear Tree WIP'. The main content area is split into two columns. The left column, titled 'CONTRIBUTION', shows the 'Partridge in a Pear Tree WIP' contribution by 'Spagneti', posted 180 days ago. It includes a text description, a preview of the image 'partridge wip.jpg' (351 KB), and a comment from 'RAMATSU' from 179 days ago. The right column contains three sections: 'TASK INFO' showing the task is 'closed' and 'due 183 days ago' with '3 people needed'; 'MORE CONTRIBUTIONS' showing a table with one entry 'Partridge WIP 2' by 'Spagneti' in 'seeking feedback' status, posted 177 days ago; and 'RECENT ACTIVITY' showing three recent actions: a comment by 'RAMATSU', an edit by 'Spagneti', and the creation of the contribution by 'Spagneti'. An 'Older Activity »' link is at the bottom right of the activity section.

**Figure 1. The Pipeline tasks system.** This screenshot depicts a task contribution. The left column displays (from top to bottom) the metadata, contents, and comments for this version of the contribution. The right column lists (from top to bottom) the contribution’s parent task, other versions of this contribution, and recent activity for the contribution.

artifacts, and extend this notion beyond cognitive processes to include leadership. This suggests the design principle of *automation*, meaning that Pipeline itself should be able to assist with leadership as a non-human agent, much as, for example, bots and scripts perform social roles in combating Wikipedia vandalism [12]. Finally, since DL and DCo emphasize that distributions can happen over time, Pipeline should allow leaders to reconfigure decentralization and automation in real time, in response to changing circumstances.

### System Overview

Pipeline<sup>2</sup> is a web-based collaboration tool built with PHP, MySQL, and jQuery. It was not designed to be a new online creative community nor a new host for finished projects; these needs are already well served by existing sites like YouTube, DeviantArt, and Newgrounds. Instead, Pipeline is meant to provide collaboration support that integrates with existing online communities. It is free, open source software released under the GPL and available on GitHub. A moderately technical person can install a new Pipeline

instance for any online community in a few minutes, similar to the process used by the popular blogging software Wordpress. Alternatively, our research group’s web server hosts Pipelines for several popular creative communities, including Newgrounds.

Pipeline’s basic structure borrows many familiar concepts from other social software, such as user profiles, private messages, and a dashboard. Projects are the cornerstone of Pipeline. Any registered user can start a new project or find one to join by browsing a list of public projects. Users can also create private projects that are visible only to invited members. The key features of Pipeline projects that enable decentralization and automation are the trusted members system, the tasks system, and the activity feeds, which we describe below.

### Trusted Member System

Pipeline uses the concept of “trust” to distribute leadership within a project. Content creation, e.g. joining open tasks, sharing work in progress, leaving feedback on others’ work, and participating in discussions, is open to all project members. However, *articulation work* [20,29], e.g. creating

<sup>2</sup> <http://pipeline.cc.gatech.edu/>

tasks or editing high-level project settings like deadlines, specs, and rules, is limited to trusted members.

When a user creates a new Pipeline project, she can choose which members to trust, if any. On one extreme, the creator could trust only herself, potentially replicating the top-down, centralized approach to leadership seen in most collabs. On the other extreme, the creator could trust all project members, granting full technical permissions to everyone. In this case, the project could resemble a wiki, relying on social rather than technical controls to influence member behavior. Beyond these extreme options, project creators could also trust any subset of members, from a small group of close friends to all but a handful of novices, for example. Trusted status can be granted or revoked at any time, allowing for dynamic adjustments of leadership distribution throughout the project's lifespan.

### Tasks System

Pipeline provides a lightweight tasks system to help project members organize their work (Figure 1). Only trusted members can create tasks, and each task has a “task leader” responsible for ensuring the task gets done. This feature is meant to encourage decentralization, but not at the expense of members' free choice. Most Pipeline users are volunteers, and being able to self-select for tasks is both an important motivator and a highly efficient mechanism for dividing up work in online collaboration [3]. Hence, the design of Pipeline encourages members to claim work for themselves rather than assigning work to others. For example, there is no explicit software support for assigning work to project members; instead, members find work by browsing a list of open tasks and joining those that appeal to them. When a trusted member creates a new task, the default option is to lead the task herself.

Any project member can join a task and make contributions to it. Each contribution can include attached files in many popular media formats, including Flash animation, video, audio, images, and text documents, some of which Pipeline compresses or streams for quick previewing in the web browser. Members can iterate on their contributions by posting new versions, which Pipeline links together visually. Members can also leave feedback on any version of a contribution in the form of threaded comments.

### Activity Feeds

Pipeline keeps comprehensive, fine-grained histories of user behavior. More than 40 types of events are stored, including users creating or editing projects, tasks, contributions, file uploads, discussions, and comments; users joining or quitting projects and tasks; and users inviting, banning, or trusting other members. Pipeline's database also stores sufficient metadata for each event to allow a project's complete history to be reconstructed.

One benefit of these histories is to aid our research efforts, but more importantly, Pipeline shares this information with

users in order to increase the visibility, awareness, and accountability of their actions—i.e., to facilitate *social translucence* among users [9]. Group awareness is critical to successful collaboration, but it can be difficult and time-consuming to maintain, particularly when members are geographically distributed, as in online collaboration [15]. Pipeline seeks to automate some of the more tedious aspects of group awareness previously handled by leaders by making member behavior more socially translucent, i.e., displaying it as “recent activity” feeds. These feeds appear as a sidebar on most pages and are context-sensitive, showing only events related to the current page. Each project also includes a dedicated activity page where members can filter and explore the project's complete history. For most event types, users can zoom in on an event's details, e.g. the complete text of a comment or the “diff” of an edit to a task's description. Users can also set up customized email notifications for when certain event types occur, such as new comments on their contributions, or changes to a task they joined.

It is worth noting that while most user actions are logged, some events are hidden from activity feeds to protect users' privacy (e.g. events related to private messages). The goal with social translucence is not transparency but rather effective use of constraints, such as the balance between privacy and visibility, and helping users understand these constraints [9]. To this end, Pipeline uses a simple privacy model: all project members (creator, trusted, regular) see exactly the same set of events.

### Comparisons to Similar Systems

Pipeline differs from other web-based collaboration tools, such as Asana, Basecamp, Podio, and Teambox, in that Pipeline's features encourage users to work with people they may not already know. It supports public projects and its trusted member system (described below) offers flexible permissions to help such projects succeed. Additionally, while most collaboration tools are proprietary and charge monthly fees, Pipeline is free, open-source, and designed to integrate with existing online communities.

Pipeline also bears similarities to peer production tools. Its purpose is similar to software development host Github, but Pipeline aims for breadth rather than depth, supporting a variety of creative domains. Pipeline also draws inspiration from the wiki platform MediaWiki, but even in its most wiki-like configurations, Pipeline provides more structure than MediaWiki's defaults. Finally, Pipeline's task system resembles those of crowdsourcing marketplaces like Amazon Mechanical Turk, but unlike these systems, Pipeline assumes users will directly interact with each other and are driven by intrinsic motivations rather than payment.

### Development Process and Launch

The design of Pipeline began in earnest in the fall of 2009. We created low-fidelity HTML mockups and solicited feedback via email from a subset of interviewees from our

initial studies of collabs. After receiving this feedback and modifying our designs, we spent most of 2010 developing a working prototype. In early 2011 we released our prototype, Pipeline 1.0, to a small group of beta testers. We spent the summer developing a major revision based on feedback from the beta period. We launched Pipeline 2.0 in October 2011, advertising in creative communities such as Newgrounds, Kongregate, DeviantArt, and Vimeo. In November 2011, we released the Pipeline source code on GitHub. Since then, the Pipelines on our web server have attracted more than 350 registered users and 90 projects, including Flash movies and games, films and videos, transmedia story development, and even Minecraft texture packs. In the rest of the paper, we focus on a case study of one Pipeline project, Holiday Flood, a six-week visual art collaboration between 30 artists from around the world.

## METHODS

Because DCoG is a theory for understanding cognition in the wild, much DCoG-based research takes the form of detailed case studies of real-world phenomena (e.g. [12,16,17,18]). We take a similar approach, using primarily qualitative methods to provide a rich description of the process by which leadership was distributed in Holiday Flood. For this case study, we collected and analyzed three sources of data: activity logs from the Pipeline project where Holiday Flood was organized, discussion about Holiday Flood in the Newgrounds Art Forum, and in-depth interviews with active members of Holiday Flood.

### Pipeline Server Logs

The main source of data for this case study was the set of activity logs for Holiday Flood's Pipeline project. Pipeline logged a total of 1,155 events for Holiday Flood between November 9, 2011 (project start) and December 25, 2011 (project end). This count omits non-project-specific events, such as users registering for Pipeline, sending private messages, or updating their profiles. It also omits 15 events generated by the researchers (mostly related to Pipeline bug reports and feature requests).

### Newgrounds Art Forum

The Newgrounds forums provided a secondary source of data for this research. The creator of Holiday Flood announced the Pipeline project, recruited members, and posted occasional updates in a thread in the Newgrounds Art Forum. From November 9 to December 27, 2011, this thread accumulated 144 replies and, as of May 2012, more than 5,100 views.

### In-depth Interviews

A final source of data for this study was a series of in-depth, semi-structured interviews [30] with the five most active Holiday Flood members. We interviewed the project creator (Renae) twice, first when the project was halfway done, and again when the project was finished. We also

interviewed four other members after the project was finished. Participants were interviewed either via Skype or instant message, whichever they preferred, and were not compensated for their time. Skype interviews were audio-recorded with participants' permission and fully transcribed. Interviews lasted an average of 105 minutes each (min: 35, max: 167), for a total of 627 minutes (10.5 hours). Our questions focused on participants' experiences with the Holiday Flood project, including the role of leadership, division of labor, feedback and communication patterns, attitudes towards the finished product, and benefits and drawbacks of using Pipeline.

### Using Real Names

Since many of our participants view themselves as artists, it is potentially unethical to deny them credit for their work by anonymizing them [6]. Throughout this paper, we refer to participants by their Pipeline usernames, with two exceptions. We use interviewees' real names instead of their usernames if they requested it. We also do not name participants who dropped out of Holiday Flood, to avoid embarrassing them.

### CASE STUDY: HOLIDAY FLOOD

The story of Holiday Flood's success is worth telling per se, but it also sheds light on how Pipeline supported and transformed leadership in a real-world example of online creative collaboration. In the following sections, we first describe the result of Holiday Flood, and then present a rich description of the process by which it was created.

### The Holiday Flood

On the morning of December 12, 2011, a 19-year-old artist with the username Ashman submitted an illustration of a drummer to the Newgrounds Art Portal. Later that day, another drummer illustration appeared, authored by a different young artist, ZaneZanesorrow. At the time, few visitors to the Portal made note of these artworks among the dozens of others submitted that day.

By Christmas Eve, 28 artists from around the world had submitted 24 artworks (two per day) illustrating the lyrics of "The Twelve Days of Christmas," a traditional carol. Each artwork incorporated a special thumbnail image and was tagged with the same obscure keyword: "ohf11". Viewers who noticed and clicked on this tag discovered that the thumbnails were actually puzzle pieces. This action aligned all the puzzle pieces in the proper sequence, revealing a hidden message: a large, festively illustrated "Season's Greetings" poster, complete with the signatures of all 28 artists (Figure 2). Thus, the Holiday Flood artworks comprised not only a visual interpretation of the "Twelve Days of Christmas" song, but also a kind of digital Advent calendar. Traditionally, these calendars anticipate the arrival of Christmas with a collection of doors on a card, one opened each day, revealing parts of an image or story.

Finally, on Christmas Day, “Operation Holiday Flood,” an interactive art gallery, appeared in the Newgrounds Flash Portal.<sup>3</sup> It was a compilation of all 24 artworks, complete with a custom arrangement of the classic song.

In many ways, Holiday Flood was a success. Artists completed and submitted their works within their 12-hour windows, the thumbnails lined up correctly, and the interactive gallery was finished by Christmas Day, all according to plan. The Newgrounds community responded well to their efforts: most artworks received high ratings and positive reviews, and the gallery won an award for being one of the top-rated submissions on December 25.

### Previous Floods

Since October 2010, two moderators of the Newgrounds Art forum, Renae (username: ReNaeNae) and Robert Westgate (username: TurkeyOnASTick), had organized a series of events called “Flood the Portal Fridays.” In a typical flood, Renae or Robert would announce a theme (e.g. frogs, facial hair), a specially created thumbnail image, and a time frame, usually the first Friday of each month at 6 o’clock EST. Artists were invited to create an artwork based on the theme, submit it within the specified time frame, and include the provided thumbnail image. The result was a “flood” of new artwork to the Art Portal, unified by the theme, thumbnail, and a custom tag (e.g. “ftp12” for “Flood the Portal Fridays 12,” the twelfth such flood). The floods were a lightweight form of collaboration intended to build a sense of community among Newgrounds artists and encourage submissions to the Art Portal. As Renae observed:

*With regular floods, it’s just for fun...nothing more comes of it than covering the page with the same thumbnail, so...it’s not that big of a deal if you skip out.*

As the end of 2011 approached, Renae wanted to organize a new, more ambitious type of flood. Meanwhile, we had just released Pipeline to the public, and Renae had been an early beta tester. She decided (without prompting from us) to use Pipeline to organize the new flood, which would be called “Holiday Flood” and based on a Christmas theme.

### Designing the Holiday Flood

Renae’s plan for Holiday Flood was more complex than previous floods in several ways. Previous floods were openly publicized on the Art Forum, but Holiday Flood would be, as much as possible, organized in secret. Holiday Flood would occur in a particular sequence, not just at a particular date and time. Rather than being identical, the artworks’ thumbnails would be unique puzzle pieces that, when fitted together, revealed a hidden holiday greeting poster. Finally, Holiday Flood would not only be released



**Figure 2. The result of Holiday Flood. Each of the 24 artworks contained a thumbnail with the artist's signature. Searching for the tag "ohf11" displayed all the thumbnails in order of submission, revealing the hidden "Season's Greetings" poster.**

as individual submissions to the Art Portal, but also collected together in a submission to the Flash Portal.

One of the first issues Renae encountered was how to recruit members for a secret project. She wanted to recruit from the Newgrounds Art Forum without spoiling the surprise for those who opted out. The simplest option might have been to privately invite a group of familiar artists, but Renae wanted Holiday Flood to be open to the entire community, at least initially. Her solution was to frame Holiday Flood as a kind of secret mission. On November 9, she created a thread in the Art Forum titled, “Operation Holiday Flood 2011,” describing a “*top secret*” project for which she needed to recruit a large number of “agents.” Anyone who betrayed the mission's secrets would be punished severely, she warned, adding, “*This is Christmas dammit! We're not messing around!*”

<sup>3</sup> <http://www.newgrounds.com/portal/view/586537>

Simultaneously, Renae created a private project on Pipeline as a counterpart to the public thread on Newgrounds. Private projects on Pipeline are visible only to members, providing a space for Holiday Flood artists to work in secret. Renae instructed interested users to create Pipeline accounts and post their usernames in the public thread. She then sent invitations to the Pipeline accounts listed in the thread, on a first-come, first-served basis. Within a few days, Renae had recruited almost 30 artists, and she had to turn away several others. According to the information self-reported in their profiles, the recruits were mostly male and mostly in their 20s. They represented at least 12 countries on 5 continents, with the majority concentrated in North America and Europe.

Renae gave “trusted” status to two members of Holiday Flood beyond herself. The first was Robert, her fellow Art Forum moderator and frequent flood organizer. Renae also trusted BlueJay, a 25-year-old artist from the Netherlands.

### Managing the Artists

For Holiday Flood to succeed, four major tasks needed to be completed. The 24 artworks, two for each of the 12 Days of Christmas, had to be claimed by artists and created. The hidden holiday greeting poster had to be created, divided into 24 “puzzle piece” thumbnails, and reintegrated into the finished artworks. The finished artworks needed to be submitted in a particular sequence at particular dates and times. Finally, all of the artworks, as well as original music, needed to be collected and integrated for a Flash gallery.

### Creating the Artworks

Renae created a task on Pipeline titled “CLAIM YOUR PART!” For the task’s instructions, Renae had divided up the 12 Days of Christmas into 24 slots, two for each day. She asked the artists to each claim one of the slots, on a first-come, first-served basis, and create an artwork based on that slot’s theme.

Renae took a straightforward divide-and-conquer approach to this task, intending each slot to be claimed by one artist. Yet three slots were each claimed by pairs of artists working in a partnership. In “Buurds” and “Three Calling Birds,” one artist drew the line art, and the other colored it in. For “Buurds” (Figure 3, top) one of the Four Calling Birds artworks, Fifty-50 provided the Angry Birds-inspired line art and Larkatosa colored it. “Three French Hens” by CosmicDeath and Kashi, which shows a hen bearing the three colors of the French flag and the word “trois,” was also a line art-coloring partnership.

The third partnership, “Captain Sauron” (Figure 3, bottom) by MajesticBob and Kinsei, was more complex than the others. A collage of tangible and digital artwork, the final product combined a digital illustration and a custom-built picture frame. The illustration depicted characters from *Lord of the Rings* wearing rings from the *Captain Planet* television show, while the picture frame included



**Figure 3.** Two examples of partnerships. The top pair depicts *Four Calling Birds* by Fifty-50 (left, line art) and Larkatosa (right, coloring). The bottom pair depicts *Five Golden Rings* by MajesticBob (left, physical art) and Kinsei (right, digital art).

handcrafted copies of the rings shown in the illustration. While MajesticBob was responsible for the tangible portion of the artwork, both partners contributed to the digital portion. He explained:

*I hand made the rings and the frame. Kinsei did the Picture. I cropped and fitted the pic into the frame and I did the color streams and [Photo]shopped the wind ring... Kinsei cleaned up the background and [Photo]shopped the rest of the rings.*

After all the artwork slots were claimed, Renae created another task called “WIPs”, a.k.a. works-in-progress, with a deadline of December 12 (12 days before Christmas Eve). In this task, she encouraged artists to share drafts of their artworks and critique other members’ work. This task served multiple purposes. It gave artists an opportunity to improve their work prior to submission, but it also provided progress updates, signaling which artists were more or less likely to make the deadline. About half of the artists shared a WIP before posting their completed artworks.

### Creating the Poster and Thumbnails

In addition to creating the 24 artworks mentioned above, Holiday Flood also required a Season’s Greetings poster (Figure 4), which would be divided up into puzzle pieces, one for each artwork. Renae created a task for the poster at the start of Holiday Flood, around the same time as the task for creating the artworks.



**Figure 4.** Evolution of the "Season's Greetings" poster. From left to right, Gabriel created the first two versions. Robert continued the development with the next two versions. Renae elaborated on Robert's concept to create the final version on the far right.

Initially, the poster task was ill-defined. No one had decided what the poster was supposed to look like, aside from Renae's vague instructions: "a greeting card type thing... with all of our signatures on it." Also, no one had volunteered to work on it. After some discussion within the task, Gabriel (username: ZombieMonkey), a 19-year-old Canadian artist, made the first rough drafts. He posted several versions on Pipeline, receiving detailed feedback from Renae and Robert after each.

After about a week, Robert felt that Gabriel's work had progressed as far as it could, and asked him to share his source files on Pipeline. Robert then used those source files to create a series of revisions that were generally agreed to be improvements. Gabriel himself admitted that "[Robert] took my image and just made it so much better looking. I mean, it just looks so good now. Mine was a little bit cartoony and he just made it look so beautiful." Robert spent nearly three weeks working on the poster, but as the Holiday Flood deadline approached, he found himself lacking the time to complete it. He recalled:

*I polished it further and further until I couldn't really work on the image any more because I had my own stuff to do. All of those files were on Pipeline and then [Renae], because she wanted to polish up the image further, she took those files from Pipeline and edited that image further to be the finalized design.*

The last step was to collect the artists' custom signatures so they could be superimposed onto the poster. Robert created a new task requesting every member to submit an image containing his or her signature. Once all signatures were submitted, Renae used Photoshop to divide up the poster into 24 equal-sized puzzle pieces, or "thumbnails," and overlay a signature onto each. She then attached all the thumbnail files to a new task, so that artists could incorporate them into their finished artworks. As artists confirmed downloading their thumbnails, Renae manually checked them off the list by editing the task description.

#### *Deploying the Artworks*

With the artworks and their corresponding thumbnails completed, the next task was to release them on a particular schedule. Each of the 24 artworks needed to be submitted, or "deployed," within a 12-hour window. There were several possible failure modes. If an artist submitted late, the artwork wouldn't appear on the correct Day of Christmas. If an artist mixed up the order or didn't submit at all, the thumbnails wouldn't line up properly. The geographic distribution of artists, living in numerous time zones across a dozen countries, added to the challenge.

To assist with deployment, Renae created a task called "Deployment Schedule." For each of the 12 days, it listed who was responsible for submitting. As artists submitted, Renae or Robert updated the task, writing "CHECK" next to the artists' names and linking to their submissions on Newgrounds. This process gave artists a clear, updated reference as to who was done and who was up next.

#### **Completing the Flash Gallery**

With all of the artworks complete and deployment underway, the interactive art gallery could now be created. Renae took the lead developing the gallery, and in the last few days before Christmas, she posted frequent updates in a task called "Flash Submission," and artists responded with rapid feedback and bug reports. Renae had invited bakeroflongo, a 30-year-old musician from Boston, to provide a custom arrangement of the "12 Days of Christmas" song. After adding his music to the Flash gallery, Renae submitted the finished product to Newgrounds on Christmas Day.

#### **DISCUSSION**

Holiday Flood was a successful project, but how much of its success can be attributed to Pipeline? In the following section, we discuss ways that Pipeline supported and transformed (redistributed) leadership in Holiday Flood, focusing on Yukl's leadership behaviors reviewed above:

planning, problem solving, clarifying, informing, and monitoring.

### Planning

Holiday Flood's leaders had extensive experience planning previous floods and collabs, and they brought this experience to bear on Holiday Flood. For example, Robert noted that contingency planning (in his words, "risk management,") was an important practice he had developed over the years:

*To be honest, whenever I handle projects, I tend to think of these problems early on, anyway, and try and come up with, well, risk management really. Try and work out how to control and mitigate those problems rather than encounter them and cry because we didn't think it would happen.*

Some leadership behaviors like this one are likely more attributable to Robert and Renae's experiences than any affordances of Pipeline. Yet, there is evidence that Pipeline had a major impact on the planning process. Both leaders emphasized that Pipeline's technological support emboldened them to plan a much more complex, elaborate flood. Robert explained:

*Renae wanted to do something for Christmas, and...because this type of project needs a bit more management on the site, because you've also got to look after the order of the thumbnails coming into the Art Portal, as well as trying to make sure people actually come up with work, rather than doing absolutely nothing. So basically that project needs a lot more management, and that's why we brought Pipeline into it.*

Renae agreed, "Holiday Flood was all planned and plotted...it's the reason we needed Pipeline. [I] doubt it would have worked out any other way."

These quotes suggest that Holiday Flood's leaders felt that Pipeline made it possible to organize a new type of flood, one that was too difficult or ambitious to attempt previously. Much of the collaboration in Holiday Flood was highly interdependent, with the added complication that everything had to be completed in secret by a certain deadline. The entire process was filled with potential bottlenecks due to dependencies. If any of these steps had been omitted or seriously delayed, Gabriel explained, "the whole thing would've been ruined," a sentiment shared by all of our interviewees.

Several features of Pipeline may have supported these more elaborate plans. The "Start a Project" interface was designed to use scaffolding to help leaders include key structural/planning elements in their new projects. This tool, coupled with each project's Basics tab, communicated the most important information about the project to members. Robert recalled, "At the start of the project, [Renae] outlined how she was going to run the project, which was a simple idea which needed to be fleshed out."

Following this, the leaders used Pipeline's task system to implement these "actions plans," in Yukl's words, and share them with members. The general consensus was that implementing similar plans on the Newgrounds forums would have been very difficult or impossible. In Luwano's words:

*To be honest [Holiday Flood] would simply not have been possible without [Pipeline]. Getting 24 people to draw their pics would have been possible on NG, but arranging all the thumbnails and setting up the schedule would not have been possible there.*

Some of Pipeline's benefits are real, but others could have been imagined. The perceived advantages of Pipeline may have catalyzed a high level of enthusiasm among members and a desire to attempt something elaborate. More projects must be studied to tease apart the perceived and actual benefits provided by the software.

### Problem Solving

Overall, Holiday Flood's leaders encountered few emergencies and minimized the need to engage in problem solving. Potentially, this was the result of the leaders drawing on their past experience and engaging in effective contingency planning. However, some problems did arise and the leaders dealt with them successfully. The most frequent and recurring emergencies were drop-outs. When artists claimed a slot for one of the Twelve Days of Christmas and then failed to deliver, Renae and Robert worked to quickly find replacements. Early in the project, they responded to drop-outs by inviting new members recruited on the Newgrounds forums. However, when the artist responsible for Twelve Drummers Drumming dropped out four days before deployment was scheduled to begin, the impending deadline created additional pressure. Renae took responsibility by posting a breathless plea for help on Newgrounds: "The project has taken a critical hit... the entire operation is now in danger." But Holiday Flood had been shrouded in secrecy for weeks, so finding an external replacement on such short notice proved to be challenging. Rather than a new recruit, it was an existing member, ZaneZanesorrow, who quickly drew up an "Emergency Drummers" artwork. "It saved the whole project, basically," recalled Gabriel. Thus, both a leader and a regular member ultimately shared responsibility for solving the problem.

In the above examples, Renae responded by working to find others to claim open slots. Other times, Holiday Flood's leaders solved the problem by doing the work themselves. One of the best examples of leaders solving problems directly involves the Season's Greetings poster. As described above, when Gabriel became too busy to complete the poster, Robert and Renae stepped in and made the necessary improvements themselves, in a process of collaborative iteration. Pipeline's support for file sharing, commenting, and multiple versions facilitated this form of

direct problem solving, as the leaders could download and modify Gabriel's work-in-progress without even contacting him. If Gabriel was too busy to finish the artwork, he was likely also too busy to share his work materials. Without Pipeline, it may have been more expedient for the leaders, unable to gauge his progress, to start over with someone else.

### Clarifying Roles and Objectives

#### *Defining Job Responsibilities*

Pipeline helped leaders define job responsibilities most significantly through the trusted member system. The software gave trusted members access to tools for supporting articulation work, while regular members were limited by the system to content creation. The evidence suggests that overall, this division of labor worked well for Holiday Flood. Regular members enjoyed the creative freedom they sought while trusted members retained the authority to keep the project focused and organized. In BlueJay's opinion, "[S]tuff might have gotten cluttered if everyone could start tasks etc."

Leadership theory emphasizes the importance of clarifying roles, but Holiday Flood provides an example of a context where it was sometimes useful to allow for multiple interpretations. The notion of "trust" provided a valuable degree of ambiguity about the role of trusted members that might be harder to achieve with more traditionally power-laden concepts like "moderator" or "leader." Particularly online, where participation is often voluntary, leaders have a strong incentive to avoid flaunting their authority. For example, Reagle's [28] theory of "authorial leadership" describes the somewhat paradoxical relationship between "benevolent dictators," such as Wikipedia's Jimmy Wales and Linux's Linus Torvalds, and the egalitarian-minded communities they lead. Similarly, Holiday Flood's trusted members did not view themselves as authoritarians despite their extra permissions, authority, and influence. Robert reflected:

*Even though I had a bit more of a major role, as a guy looking after the thumbnails or the other guy [BlueJay] who was meant to look after the Flash project, I didn't really see that role as a really significant one in the overall project. I still thought of that as on the same level as a contributor making a picture and sticking a thumbnail over top of it.*

The non-leaders we interviewed concurred with this view. They found the subjective experience of participating in the project to be egalitarian and empowering and saw the leaders primarily as organizers whose goal was to facilitate progress. While they agreed that Renae and Robert had authority, they found it directed mainly at preserving the project's basic structure rather than influencing artists' creative decisions. BlueJay felt that the leaders were "mostly making sure everyone stuck to the schedule...kept the Pipeline a bit organized, making threads/tasks for stuff

*etc. [whereas] generating ideas and deciding on them was mostly done in consultation with everyone."*

#### *Setting Performance Goals*

Pipeline helped leaders set performance goals primarily through the Basics tab and task management system. Early in the project, Renae described the project's overall goals in the Pitch section of the Basics tab. As the project progressed, she created tasks for all major components of the project. For eight of the nine tasks, she also supplied a deadline, which further clarified the expectations for those tasks.

#### *Assigning Tasks*

As mentioned above, Pipeline encourages self-selection for tasks by preventing leaders from directly assigning work to members. This design decision appeared to be a good match for the Newgrounds community and Holiday Flood in particular. For example, one of the first tasks Renae created was titled, "CLAIM YOUR PART!," in which she encouraged members to sign up for artwork slots on a first come, first served basis. Members were pleased with their selections, as evidenced by excited comments on the task and the fact that all artworks were completed. Meanwhile, leaders were free to focus on responsibilities other than matching artists with tasks and dealing with those who were unsatisfied with their assignments.

#### *Providing Instructions for Tasks*

The leaders of Holiday Flood provided instructions to members throughout the project. Pipeline supported this practice in several ways. Tasks were designed with several fields for including instructions, such as the number of people needed for a task, and a plain-text "Instructions" area for more detailed information.

Yukl notes that providing feedback is another important component of the leadership behavior of providing instruction [34]. Traditionally, formal leaders provided most or all of the feedback to collab members because the limitations of the Newgrounds forum made it difficult for members to share their work or exchange feedback among themselves. Most had to wait for audience reviews of the finished product, or they could email their files to the project's leader and request comments.

In contrast, Pipeline provided each contribution with a dedicated area for file sharing, iteration, and discussion. Many members were able to share rough drafts and receive comments prior to the deadline, from both leaders and regular members. As a result, feedback was more evenly distributed among the group. BlueJay explained:

*In previous floods people usually provided feedback after everything had been submitted to the portal. With other art collabs the amount of feedback is usually pretty much limited to the collab organizer. For the Holiday Flood people uploaded and got more feedback on WIPs.*

BlueJay's impression is supported by the data. We analyzed the Pipeline logs and found that members of Holiday Flood had posted 114 comments on contributions that were not their own. Of these, less than half (50 comments, or 44%) were posted by Renae and Robert. In other words, the majority of feedback on works-in-progress was provided by regular members, not formal leaders.

### Informing

Pipeline was designed with the goal of helping to organize the information flow and supporting leaders as informers. Each project has a Discussion tab, which allows members to participate in forum-style discussions that are organized by topic, rather than as one chronological list. Renae used this feature several times to make general announcements to the Holiday Flood membership.

Other Pipeline features sought to help automate certain aspects of informing. Yukl notes that leaders can increase effective informing by "improv[ing] direct access by others to relevant information" [34]. Pipeline is intended to support this direct access by connecting members to the content that matters to them. For example, the Tasks tab displays the user's tasks first, and Recent Activity feeds are contextual, showing only the events that are relevant to the surrounding content. Our intention was that Pipeline would automate some of this tedious informing work, easing the burden on leaders and freeing them to focus on other tasks. Our inspiration for this idea was Hutchins' analysis of DCog in airplane cockpits, where he describes how small instrument markers called "speed bugs" do not actually help pilots remember significant airspeeds, but instead store the information so that pilots need not remember it at all, and can focus on other matters [18]. He writes, "The beauty of devices like speed bugs is that they permit these reconfigurations of functional systems in way that reduce the requirements for scarce cognitive resources." Similarly, we hoped that Pipeline could provide tools that, like an aircraft's speed bugs, reduced the requirements for scarce cognitive resources. Holiday Flood raised two issues with this line of thinking that prompted me to unpack the goal of "easing the burden on leaders," which may have been overly simplistic.

First, many of our design choices for automating leadership behaviors in Pipeline had the unexpected consequence of actually distributing small but significant bits of work to non-leaders. For example, instead of leaders collecting members' files and providing feedback, members uploaded their files directly to Pipeline so that anyone in the project could comment. Instead of leaders answering questions about the project's structure, members could explore the open tasks to answer their own questions. Instead of leaders manually updating member lists, members clicked "Join" or "Leave" buttons and Pipeline updated the member list accordingly. In general, Pipeline encouraged Holiday Flood's members to interact regularly with the site so that it could capture this activity and share it with other members

in the form of recent activity feeds. Instead of automating leadership behaviors (e.g. informing and monitoring), it may be more accurate to say that Pipeline was improving the social translucence of these leadership behaviors.

Second, there were unexpected benefits to having human leaders act as informers. For example, some leaders manually updated tasks when members made a contribution, even as Pipeline updated the task automatically. While the leaders' manual efforts and the task system both communicated the same progress information, they were not redundant because the manual updates contained additional information: the leaders were paying attention to the artists' progress. The automatic updates were quicker and more reliable, but the manual ones provided valuable motivation. Robert explained:

*[It was] very helpful for everybody really, just to make sure that people are contributing and that there is some active progression on the site. Not only because it's a clear sign that the leader is very much motivated with the project, but that other people also have that motivation as well.*

Hence, it is important to consider the tradeoffs of automating leadership behaviors, as sometimes even tedious work is more effectively done by human leaders. This finding bears similarities to attitudes towards authorship in online remix communities, where manual crediting is viewed as essential even when the system automatically attributes remixed content [25].

### Monitoring

Pipeline supports monitoring most significantly through its Recent Activity feeds, email notifications, and the underlying infrastructure that logs user behavior. As mentioned above, this infrastructure caused many member activities that previously happened off-site or went uncaptured to become socially translucent, and therefore available for other users to interpret. Additionally, the fact that Recent Activity feeds were visible to all project members, not just leaders, meant that monitoring was transformed into a more distributed leadership behavior. Leaders could more easily monitor the progress of members, but now members could also monitor the leaders and one another.

While the Recent Activity feeds helped redistribute some aspects of monitoring, other Pipeline features also supported more traditional, leader-driven monitoring. One example is the Discussion tab, which Renae used to organize two "roll calls" on November 18 and December 5 in order to smoke out drop-outs. In these discussions, Renae required every member of Holiday Flood to check in and post a brief update before a certain deadline—otherwise, they would be banned from the project. As described above, Renae followed through on this promise and banned a handful of unresponsive members, taking advantage of both

Pipeline's tools for communication and for cutting off communication.

### CONCLUSION AND IMPLICATIONS

Leadership plays a significant role in the success or failure of online creative collaboration. When leaders have proper technological support and assume a manageable workload, they can lead complex creative collaborations with significant impact. These success stories include the world's largest encyclopedia, some of its most popular and useful software, and compelling, award-winning art and entertainment. However, many other promising efforts fail because leaders are overburdened. In the Newgrounds community, the challenges of online creative collaboration and poor technological support contributed to a culture of simple projects and top-down, centralized leadership styles. Leaders who attempted more ambitious projects ended up dealing with many of the complications that arose themselves, with little help from technology or other members. In this paper, we developed a theoretical framework to understand how leadership could be redistributed; described the design of Pipeline, a web-based tool informed by this framework; and presented a case study showing how the tool was used in a real-world context, a successful online artistic collaboration.

The case study revealed that Holiday Flood's leaders organized a highly ambitious project and were rewarded by success. To understand Pipeline's role in this success, we analyzed leadership in the project through the theoretical lens of Yukl's five leadership behaviors. While the leaders, Renae and Robert, brought extensive leadership experience to the project and drew upon that experience, their efforts were also streamlined and emboldened by Pipeline.

For the most part, Pipeline served to support Renae and Robert's top-down, centralized leadership style. Its "Start a Project" tool and Basics tab assisted with early planning and structure for Holiday Flood, while the software's overall capabilities encouraged the leaders to strive for something more complex and interdependent. Pipeline helped leaders solve problems by providing quick access to members' works in progress. While Pipeline's trusted member system helped clarify roles by limiting permissions of regular members, it also provided a valuable degree of ambiguity that deemphasized power relationships. The software supported leaders' informing behaviors by helping them communicate important information to members via discussions, and leaders also used discussions to initiate "roll calls" that helped them monitor progress.

There is also evidence that Pipeline transformed some aspects of leadership by redistributing certain leadership behaviors across a broader set of Holiday Flood members. Assigning tasks is an important leadership behavior identified by Yukl [34], yet unlike most project management tools, Pipeline does not allow leaders to assign tasks. Rather, Pipeline's design draws inspiration from peer

production theory and encourages users to self-select for tasks. Holiday Flood's leaders created tasks and asked members to claim ones that appealed to them, reducing the leaders' coordination efforts and leveraging the motivations and self-knowledge of members. Another leadership behavior identified by Yukl, providing feedback, was redistributed via Pipeline's support for comments, file sharing, and iteration such that more than half of all Holiday Flood feedback was provided by non-leaders. Finally, the Recent Activity feeds in Pipeline helped redistribute the leadership behaviors of informing and monitoring by encouraging members to make their actions more socially translucent, and displaying this behavior in meaningful, context-sensitive ways.

Overall, the Holiday Flood case study suggested that Pipeline can be a powerful tool for both supporting and transforming leadership in online creative collaboration. However, more studies of additional projects are needed to understand its strengths and limitations, and tease apart the interrelationships between leader expertise, domain requirements, and technological support.

The major theoretical implication of this work is that online leadership can be redistributed to make it more effective. Previous work established that leadership can be distributed in online creative collaboration [35,36]. We extend this work by integrating theories of distributed leadership (DL) and distributed cognition (DCog) to show how technology can help redistribute leadership. This integration furthered the development of DL by introducing technology as a means of distributing leadership, and pushed DCog forward by showing how its constructs can be used for analyzing and designing for leadership, not just cognition.

According to Halverson [16], theories that are useful for CSCW have four important attributes: inferential power, application power, descriptive power, and rhetorical power. Our proposed framework helped us *infer* that leaders could be more effective, and projects more complex, if leadership was redistributed in Newgrounds collabs. It helped us *apply* the theory of DL to online creative collaboration by informing our design considerations for Pipeline (e.g. decentralization and automation). It helped us *describe* the Holiday Flood creation process using a case study that focused on the interplay between people (formal leaders, other members) and technologies (Pipeline, Newgrounds). And finally, it provided *rhetorical* power by helping us map observed phenomena in Holiday Flood to theoretical constructs (e.g. leadership behaviors, leader legitimacy, and social translucence).

This work also suggests a major design implication. Systems supporting online creative collaboration should assume formal leaders exist, but also help them redistribute leadership across the entire group as needed. In the case of Holiday Flood, Pipeline helped Renae and Robert organize a complex artistic collaboration by helping them decentralize some tasks traditionally handled by leaders,

such as giving feedback and sharing progress reports, across nearly 30 other members. Other projects may have the opposite problem, where leadership is too distributed to be effective and needs to be concentrated among a smaller number of formal leaders. By providing tools like Pipeline's "trust" system that allow members to customize leadership models for each project, or even make real-time adjustments within one project, system designers will encourage a richer variety of creative processes and outcomes from users.

## ACKNOWLEDGMENTS

We wish to thank the participants in this study, Pipeline users and beta testers, and the Newgrounds staff and community. Boris de Souza, Chris Howse, and Kevin Ziegler made substantial contributions to the design and development of Pipeline. Scott Counts, Jill Dimond, Betsy DiSalvo, Ellen Do, Eric Gilbert, Wendy Kellogg, Sarita Yardi, and our anonymous reviewers provided valuable feedback on this paper. This research is supported by NSF award CreativeIT-0855952.

## REFERENCES

- Adair, M. Eric Whitacre's Virtual Choir will be unveiled Thursday. *The Los Angeles Times*, 2011. <http://latimesblogs.latimes.com/culturemonster/2011/04/eric-whitacre-virtual-choir-revealed.html>.
- Becker, H.S. *Art Worlds*. University of California Press, 1984.
- Benkler, Y. Coase's Penguin, or Linux and the Nature of the Firm. *Yale Law Journal* 112, 3 (2002), 369–446.
- Boland, R.J., Tenkasi, R.V., and Te'eni, D. Designing Information Technology to Support Distributed Cognition. *Organization Science* 5, 3 (1994), 456–475.
- Bolden, R. Distributed Leadership in Organizations: A Review of Theory and Research. *International Journal of Management Reviews* 13, 3 (2011), 251–269.
- Bruckman, A. Studying the amateur artist: A perspective on disguising data collected inhuman subjects research on the Internet. *Ethics and Inf. Technol.* 4, 3 (2002), 217–231.
- Cranshaw, J. and Kittur, A. The polymath project: lessons from a successful online collaboration in mathematics. *Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM (2011), 1865–1874.
- Edwards, G. Concepts of Community: A Framework for Contextualizing Distributed Leadership. *International Journal of Management Reviews* 13, 3 (2011), 301–312.
- Erickson, T. and Kellogg, W.A. Social translucence: an approach to designing systems that support social processes. *ACM Trans. Comput.-Hum. Interact.* 7, 1 (2000), 59–83.
- Forte, A., Larco, V., and Bruckman, A. Decentralization in Wikipedia Governance. *Journal of Management Information Systems* 26, 1 (2009), 49–72.
- Galant, R. Software brings Johnny Cash back to life. *CNN*, 2011. [http://articles.cnn.com/2011-04-21/opinion/koblin.data.art\\_1\\_images-compelling-data-users?\\_s=PM:OPINION](http://articles.cnn.com/2011-04-21/opinion/koblin.data.art_1_images-compelling-data-users?_s=PM:OPINION).
- Geiger, R.S. and Ribes, D. The work of sustaining order in wikipedia: the banning of a vandal. *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, ACM (2010), 117–126.
- Gronn, P. Distributed leadership as a unit of analysis. *The Leadership Quarterly* 13, 4 (2002), 423–451.
- Gronn, P. The future of distributed leadership. *Journal of Educational Administration* 46, 2 (2008), 141–158.
- Gutwin, C., Penner, R., and Schneider, K. Group awareness in distributed software development. *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, ACM (2004), 72–81.
- Halverson, C.A. Activity Theory and Distributed Cognition: Or What Does CSCW Need to DO with Theories? *Comput. Supported Coop. Work* 11, 1-2 (2002), 243–267.
- Hollan, J., Hutchins, E., and Kirsh, D. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Trans. Comput.-Hum. Interact.* 7, 2 (2000), 174–196.
- Hutchins, E. How a cockpit remembers its speeds. *Cognitive Science* 19, 3 (1995), 265–288.
- Hutchins, E. *Cognition in the Wild*. A Bradford Book, 1996.
- Kriplean, T., Beschastnikh, I., and McDonald, D.W. Articulations of wikiwork: uncovering valued work in wikipedia through barnstars. *Proceedings of the ACM 2008 conference on Computer supported cooperative work*, ACM (2008), 47–56.
- Luther, K. and Bruckman, A. Leadership in online creative collaboration. *Proceedings of the ACM 2008 conference on Computer supported cooperative work*, ACM (2008), 343–352.
- Luther, K., Caine, K., Ziegler, K., and Bruckman, A. Why it works (when it works): success factors in online creative collaboration. *Proceedings of the 16th ACM international conference on Supporting group work*, ACM (2010), 1–10.
- Luther, K., Diakopoulos, N., and Bruckman, A. Edits & credits: exploring integration and attribution in

- online creative collaboration. *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, ACM (2010), 2823–2832.
24. Mockus, A., Fielding, R.T., and Herbsleb, J.D. Two case studies of open source software development: Apache and Mozilla. *ACM Trans. Softw. Eng. Methodol.* 11, 3 (2002), 309–346.
  25. Monroy-Hernández, A., Hill, B.M., Gonzalez-Rivero, J., and boyd, danah. Computers can't give credit: how automatic attribution falls short in an online remixing community. *Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM (2011), 3421–3430.
  26. Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K., and Ye, Y. Evolution patterns of open-source software systems and communities. *Proceedings of the International Workshop on Principles of Software Evolution*, ACM (2002), 76–85.
  27. Olson, G.M. and Olson, J.S. Distance matters. *Hum.-Comput. Interact.* 15, 2 (2000), 139–178.
  28. Reagle, J.M. Do as I do: authorial leadership in wikipedia. *Proceedings of the 2007 international symposium on Wikis*, ACM (2007), 143–156.
  29. Schmidt, K. and Bannon, L. Taking CSCW seriously: Supporting articulation work. *Computer Supported Cooperative Work* 1, 1 (1992), 7–40.
  30. Seidman, I. *Interviewing As Qualitative Research: A Guide for Researchers in Education And the Social Sciences*. Teachers College Press, 2006.
  31. Stelter, B. An Emmy for Rebuilding a Galaxy. *The New York Times*, 2010.  
<http://www.nytimes.com/2010/08/28/arts/television/28uncut.html>.
  32. Thorpe, R., Gold, J., and Lawler, J. Locating Distributed Leadership. *International Journal of Management Reviews* 13, 3 (2011), 239–250.
  33. Weber, S. *The Success of Open Source*. Harvard University Press, 2004.
  34. Yukl, G.A. *Leadership in Organizations*. Prentice Hall College Div, 1997.
  35. Zhu, H., Kraut, R., and Kittur, A. Effectiveness of shared leadership in online communities. *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, ACM (2012), 407–416.
  36. Zhu, H., Kraut, R.E., Wang, Y.-C., and Kittur, A. Identifying shared leadership in Wikipedia. *Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM (2011), 3431–3434.