© 2014 by Roshanak Zilouchian Moghaddam. All rights reserved.

CONSENSUS BUILDING IN DISTRIBUTED DESIGN DISCUSSIONS

BY

ROSHANAK ZILOUCHIAN MOGHADDAM

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Professor Brian Bailey, Chair
Professor Wai-Tat Fu
Professor Karrie Karahalios
Professor Andrew Ko, University of Washington
Professor Michael Twidale

Abstract

With the rapid growth of open source and other geographically distributed software projects, more User Interface (UI) design discussions are occurring online. These discussions typically unfold through interactive web forums where multiple stakeholders generate and debate different design proposals. One important challenge is bridging the different perspectives and needs of stakeholders and deciding on a design proposal that will be implemented and integrated into the product distribution. Failure to reach agreement can cause discussion participants to loose interest and therefore the product would not be improved. However, the existing discussion platforms do not provide any mechanisms to aid the process of reaching a decision.

Prior work has analyzed distributed design discussions from many perspectives, but my PhD dissertation brings consensus building as a new theoretical lens to study and support decision making in design discussions. This has allowed my work to make three contributions. First I contribute empirical knowledge showing that nearly half of the discussions do not reach consensus and knowledge of the content, process, and social factors that affect consensus building in distributed design discussions. Second, to address the inability to reach consensus, I developed a novel browser extension called Procid that provides interaction and visualization features for bringing consensus strategies to distributed design discussions. Key features include the ability to organize discussions around ideas, register and visualize support for or against ideas, and define criteria for evaluating ideas. It also applies interaction constraints promoting best practices of consensus. Finally, I present the results of two evaluations of Procid. The first collected perceptions of the tool from members of a large and successful open source community (Drupal.org) for their own discussions. The second compared how Procid affects a distributed design discussion relative to the current discussion platforms. Results of both studies showed that users perceived Procid as more effective for consensus building than the existing platforms.

To my parents, Mahboobeh, Mehdi, the love of my life, Mohsen, and my brothers Emad, and Omid for their love and support.

Acknowledgments

I would like to thank my adviser, Brian Bailey, for his guidance and support throughout my PhD career. He has taught me to have high standards and strive for the best. His inspiration and support has helped me to develop critical thinking skills and become an independent researcher. Brian, thank you for the time and effort that you have spent for me.

I have received excellent advice and guidance from many different mentors. Wai-Tat Fu, Andrew Ko, Karrie Karahalios, and Michael Twidale graciously agreed to be on my thesis committee and provided valuable feedback to improve my work.

I had the pleasure of working with Cinda Hereen as a teaching assistant for three semesters. Cinda's work ethic, enthusiasm, and leadership has left a lasting impression on me. She has been a true friend to me. She has always received me with open arms and a nice smile on her face.

When I interned at Google, I interacted with and learned from many researchers including my supportive mentor David Huffaker, Ed. Chi, and Gueorgi Kossinets.

At Illinois, I have worked with several undergraduates who were interested in conducting HCI research. I would like to thank all of them for their contributions to my research. My special thanks goes to Zane Nicholson for his significant contributions to the development of Procid and Christina Poon for being my peer in content analysis.

I thank all graduate students who supported me and shared their experiences with me. In particular, I would like to thank Moushumi Sharmin, Brett Jones, Raj Sodhi, Motahhare Eslami, Vera Liao, Anbang Xu, Mary Pietrowicz, Chris Cervantes, Yu (Wayne) Wu, and John Lee.

I would also like to thank my undergraduate adviser, Marjan Sirjani, who has been my role model and a great source of motivation.

My time in graduate school was much more enjoyable and interesting

through the company of my friends. There was never a dull moment with any of them. They have my deepest thanks and appreciation.

I am mostly thankful to my family. I am thankful to my mom, Mahboobeh, for believing in me, for encouraging me, and for being there for me whenever I needed her. I am grateful to my dad, Mehdi, for his unconditional love and support, for giving me great advice whenever I needed it, and for having short and fun conversations with me to cheer me up. I am thankful to my brothers, Emad and Omid, for motivating me to both succeed in my academic career and become a better person.

Lastly, I would like to thank my husband and best friend, Mohsen Vakilian, who has been my companion throughout this journey. He was the person who kept me motivated and helped me overcome my fears. Mohsen, thanks for always being there for me.

This dissertation would not have been possible without the help and support of many people. To everyone who was involved, both directly and indirectly, I thank you sincerely.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

UI design is a communication-driven process where stakeholders with different interests generate and debate design proposals to solve usability problems and integrate new features. Due to the rise in distributed teams and, in particular, Open Source Software (OSS) [24] many design discussions now unfold through interactive platforms such as Web forums. For example, to initiate a community discussion of a usability issue using a typical web forum, a community member creates a discussion thread and describes the issue and others join the discussion to propose and debate design proposals (Figure 1.1). It is imperative for discussion participants to eventually agree on one design proposal that will be implemented and integrated into the product distribution [68].

Despite the criticality of reaching agreement, the existing discussion platforms only offer basic commenting and sometimes voting mechanisms. These platforms are heavily text-centric and linear without any mechanisms for tracking design proposals, identifying comments related to the proposals, defining evaluation criteria, or handling tension and conflict. Therefore, participants may loose track of bright ideas in the midst of the discussion, engage in unnecessarily emotional or heated discussions, refrain from providing solid arguments against or for an idea, post irrelevant comments due to failure to keep up with the discussion, or even fail to reach a decision. In fact, my analysis of distributed design discussions in one large open source software community showed that 42% of these discussions do not reach a decision. This can cause loss of community investment, missed opportunities for enhancing the product's user interface, and experienced members to leave. Others have also identified the frustration caused by the inability to reach agreement as a major challenge facing online peer production communities [5]. To better illustrate the challenges of participating in distributed design discussions I describe an authentic scenario next.

Add colouring (and description) to password checker

Posted by webchick on November 8, 2008 at 8:06pm

The new password checker in Drupal 7 is very slick. It would be cool if it were just a *little* bit slicker. ;)

I really like the functionality of https://www.google.com/accounts/NewAccount:

Required information for Google account
Your current email address:
e.g. myname@example.com. This will be used to sign-in to your account.
Choose a password:
Password strength: Weak
Minimum of 8 characters in length.
Re-enter password:
Remember me on this computer.
Creating a Google Account will enable Web History. Web History is a feature that will provide you with a more personalized experience on Google that includes more relevant search results and recommendations. Learn More
Enable Web History.

Closed (fixed)

Project:	Drupal core
Version:	8.x-dev
Component:	user.module
Priority:	Normal
Category:	Task
Assigned:	nigel
Issue tags:	

JavaScript, Usability, accessibility

Log in or register to update this issue
Last updated on Oct 8, 2013 at 3:54pm

Comments

webchick commented 6 years ago #1

Title:	Add colouring to password checker	Add colouring (and description) to password checker

Correcting title.

Bojhan commented 6 years ago #2

Aww, almost got away with it! This has been a highly requested feature in the previous ticket around this. We initially said no to this request for the following reasons :

1. The visual emphasis would be put too strongly on the password, that it would again scream at you - scaring people off.
2. The way we structure the password validator now, should make it clear that as you're filling in the password checker the green bar will full up, assuming people don't use "a" as a password some of the suggestions would disappear and you would be left with a couple - which should trigger the behaviour of filling it up (as filling it in removed some recommendations).
3. We wanted to lay the strong visual emphasis on password matching, which is for the user the more important information (as otherwise he cant complete the form)

To many people it sounded as an obvious solution, so it might be weird that we initially decided against it. But you have to consider that sometimes what big other sites do, might not work for people using Drupal.

webchick commented 6 years ago #3

Here's why I filed this issue, and a summary of my arguments "pro" this:
* I got hung up here for a good 30 seconds or so thinking there was a bug. Since this was confusing for me -- I'm a user of Drupal too :P -- it's likely going to be confusing for others as well. alpritt pointed out though that I'm not exactly a "typical" user of Drupal, which might be true, but people who use Drupal every day are *also* our target market, remember. ;)
* yoroy pointed out that red is wrong because red means "stop" "error condition" etc. However, green is normally a colour that means "Yay! everything is great!" and in NO way is "a" a great password. :P

alpritt commented 6 years ago #4

Component:	javascript	» usability

@webchick:

My point was not that you are a highly experienced user of Drupal, but more that you have been involved in this specific issue. So you fall on the developer rather than user side, and so are more likely to be looking for problems. So, yes, there may be an issue here; but there is not necessarily one. We need to test against naive users to get a proper understanding.

I took a look at a couple of dozen password checkers before designing the current implementation. The current design is a lot closer to the norm, than what we had before. But as for getting closer to it, I'm not sure we can because there is really not a consistent design between them. So, at this more subtle stage, I again think we really need to test. I could argue the merits for the change either way.

Another option may be to go from grey to green.

Figure 1.1: A sample design discussion regarding redesigning Drupal.org's password checker. The user profile images were blurred to protect privacy while the rendering of the comments was condensed for the purpose of presentation.

1.1 An Illustrative Example

Figure 1.1 shows a sample interface design discussion from Drupal.org. Drupal is a large and highly successful open source content management system. This discussion started on November 2008 and closed on April 2012. During this time 26 developers and designers participated in this discussion and posted a total of 107 comments. From all the comments, 13 of them contained idea proposals, 21 of them were posted by experienced community members including Drupal's founder and two well-known usability experts in the community, and around 10 comments shared strong opinions. However, as shown in Figure 1.1 all posted comments are presented in a similar way, regardless of who wrote them, the importance of the content, or the strength of the message.

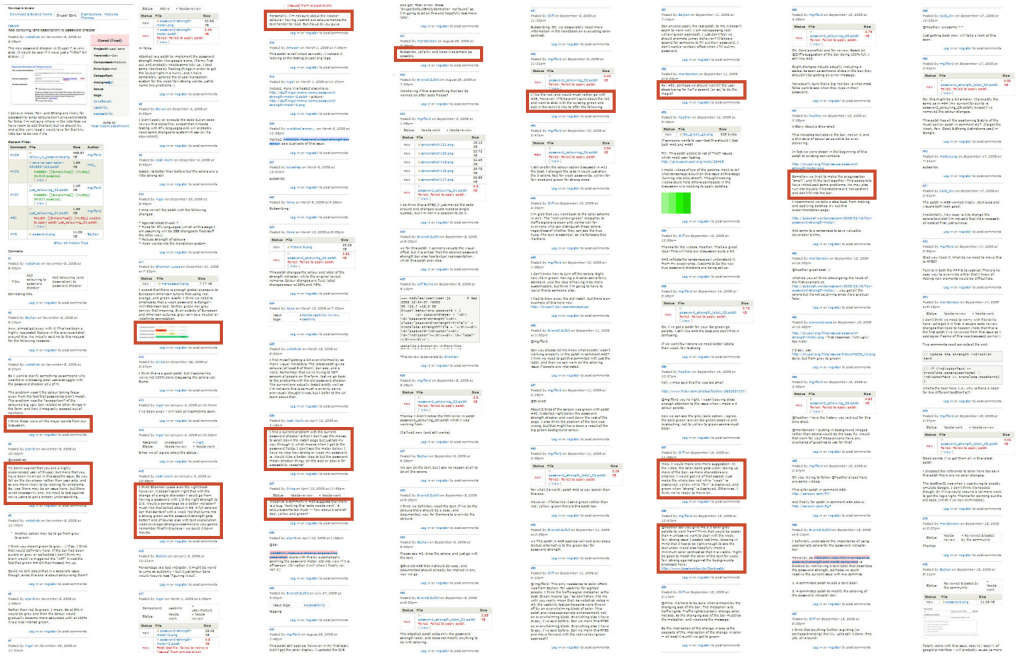

Figure 1.2: The screen-shots of first 80 comments posted to the password checker redesign issue, arranged in order from top to bottom and left to right. The thirteen highlighted posts are proposing ideas. The current interface does not provide any mechanism to track the ideas, therefore some of the ideas maybe overlooked and never be evaluated. This problem is exacerbated due to fact that the proposed ideas are distributed throughout the discussion, rather than batched at the very beginning.

In the current Drupal discussion platform, the only way a participant can maintain awareness of the proposed ideas is to rely on their memory, main-

tain notes in a separate tool, or scan the entire discussion thread. However, these methods are unreliable and cumbersome, therefore many of the proposed ideas may be overlooked or fade from participants' memory and therefore will never be evaluated. This problem is exacerbated due to fact that the proposed ideas are distributed throughout the discussion, rather than batched at the very beginning. Figure 1.2 highlights the ideas proposed in the first 80 comments of the password checker discussion. As the discussion grows, those who join later or otherwise do not keep up must review all the messages to track the ideas, the arguments for and against them, and the current direction of the discussion. However, since this is time consuming, some participants may post irrelevant comments that hinder the flow of the discussion. To partially address the lack of organization in the discussion, sometimes a participant writes a summary of the discussion, including the proposed ideas and opinions of those ideas. Figure 1.3 shows an example of such a summary written for the password checker discussion.

1.2 Existing Approaches

The problem of text-centric and linear distributed discussion platforms that do not support participants in reaching agreement is not recent. Prior research has taken different approaches to tackle this problem. One thread of research has taken an empirical approach by analyzing distributed design discussions from multiple perspectives. For exmaple, Twidale and Nichols [71] explored how usability issues are reported, discussed, and resolved in several OSS bug repositories with a goal of understanding how to improve the discussion interface. Ko and Chilana [45] also analyzed discussions in OSS repositories with the goal of understanding the structure of the discussions. Thematic coherence and argumentation in OSS design discussions were analyzed in [13]. Though the results of these studies illuminate important elements of the design discussions in OSS, they say little about the factors that affect decision making in authentic design discussions.

Another thread of research has taken a technical approach by building decision making support tools. For example, gIBIS and Compendium support design decision making by tracking issues, positions, and arguments [21, 63]. Other tools support structured argumentation and fact collection from doc-

heather *commented 5 years ago* #61

- #1 - webchick suggests that password checker "green from start" is confusing. (all green start to finish)
- #4 says go from grey to green
- #7 does it exactly like google's, with the metre to the site
- #8 points out needs testing with RTL
- #9, #10 it gets tested it's OK
- #11 shows microsoft's convention (what was wrong with the google convention??)
- #12 dries said "close, but work on colors"
- #13-19 - discussion of colour, now washes out colours. still google.
- #23 - suddenly it moves to "under" the password checker? no explanation for this change
- #25 webchick overwhelmed by colours. but she is talking about a big, like in #26 (to do with tabbing & colour change when password is deleted/changed, it doesn't go back to red)
- #33- 36 fixes bugs
- #37-39 bojhan doesn't like "all the colours". suggests going back to drop the colours, and just go with text.
- #40, ok, remove colours, just try text.
- #42-43 no, i like the red, but we'll try green again. trying just a change of green. (for readers, this means we are back to the situation at #1, which is why the thread started)
- #48 try solid green. (no progress bar)
- #51 we've gone back, red-yellow-green is good... but... feels "jumpy"
- #53-58 ok, this is my fault, i suggest the light green to green option. (discussed in #4)
- #60 based on #59 - Bar 100% full

Figure 1.3: A summary of the discussion regarding redesigning Drupal's password checker, including the proposed ideas and opinions of those ideas. In large discussions (more than 50 comments), one of the participants usually writes these summaries to partially address the lack of organization in the discussion. However, writing a through summary that covers all major points of the discussion takes a lot of time and effort.

uments [62] or provide tags and link structure to aid data exploration during collaborative sense-making [76]. However, these systems do not support design discussions, cannot integrate with an existing online community, and have solely focused on visualizing and tracking discussions rather than promoting the social process of decision making.

A related thread of research has utilized graph visualizations and interaction techniques to foster collaborative problems solving activities. For instance, Balakrishnan et al. showed that using graph visualization can help in collaborative problem solving [11]. Similarly, Alonso et al. proposed a graph visualization that represents the closeness of opinions to help small

groups reach agreement [4]. While such graph visualizations may help decision making in a small group, distributed design discussions usually involve a large group of participants who are debating design alternatives, their strengths and weaknesses, and implementation challenges using an established workflow. It is therefore essential to create a visualization that scales well, captures the unique content characteristics of these discussions (e.g. design proposals, implementation discussions, etc.), and integrates well with the current community workflow in these discussions.

1.3 My Solution

This dissertation brings the theory of consensus building to the practice of distributed User Interface (UI) design discussions by conceptualizing, developing, and evaluating PROCID (**PRO**moting **C**onsensus in **D**istributed design discussions). Procid is a novel browser extension that enables consensus building strategies to be realized in un-moderated distributed design discussions (Figure 1.4). Key features include the ability to organize discussions around ideas, register and visualize support for or against ideas, define criteria for evaluating ideas, and identify candidates to invite to active discussions. It also applies interaction constraints promoting best practices of consensus. The design of Procid was shaped by a subset of procedural, rhetorical, and social consensus building strategies derived from theory, an iterative design process, and general usability guidelines.

I built Procid as an add-on to an existing distributed discussion interface used in a mature open source community (Drupal.org), rather than as a stand-alone system. The benefits are ease of deployment and testing in an existing large community, recruiting participants who care about the discussion topic, integrating with an existing workflow, and leveraging the community's social structure to build novel features (e.g. participant invitations).

Procid promotes the social process of decision making in addition to visualizing and tracking the discussion. This is achieved by leveraging consensus theory in its design. The system provides a path way to consensus through imposing a number of process constraints, performing content analysis, and employing visual design techniques. For example to ensure that the interface is not a landscape of negative comments, Procid enforces a constraint

where the first comment posted about an idea should be positive. It also performs content analysis on the comments to inhibit overly negative comments and flame wars. Finally, the system refrains from using colors and icons commonly associated with negative affect (e.g. red). Empirical studies have shown that users found the features of Procid beneficial and perceived it as more effective for consensus building than the existing platforms. Procid should be considered as a system design pattern that can be mimicked to bring the benefits of consensus strategies to peer production and online deliberation platforms.

1.4 Consensus Building

Consensus means willingness to commit to a proposal despite any remaining objections. *Consensus building* is the social process of reaching consensus. In the context of distributed design discussions, consensus building is the social process through which all discussion participants agree on a design proposal that will be implemented and integrated into the product distribution. In an ideal consensus building scenario, the process is engaging and satisfactory and the outcome is agreement on an optimal design proposal. In this case the discussion usually follows a "U" shape pattern. It starts when the first participant posts a design proposal and the level of agreement is 100%. Other participants then join the discussion, generate alternative proposals, argue for or against them, and define criteria to evaluate the proposals. As a result, the level of agreement drops to a lower level. Eventually all the participants agree on (or don't disagree with) one design proposal and the level of agreement moves back to 100%.

While reaching consensus through this scenario is ideal, it is possible that the participants employ an ineffective process to reach agreement or even fail to agree. For instance, participants may avoid raising controversial issues or proposing alternative solutions and instead immediately agree on the first design proposal. This may happen due to groupthink [42] or because one of the participants tries to suppress other's ideas and opinions. Either way, employing an ineffective process can cause low quality or defective decisions [42], participants to feel undervalued, and even cause community members to leave because they are less engaged in the community decisions. It

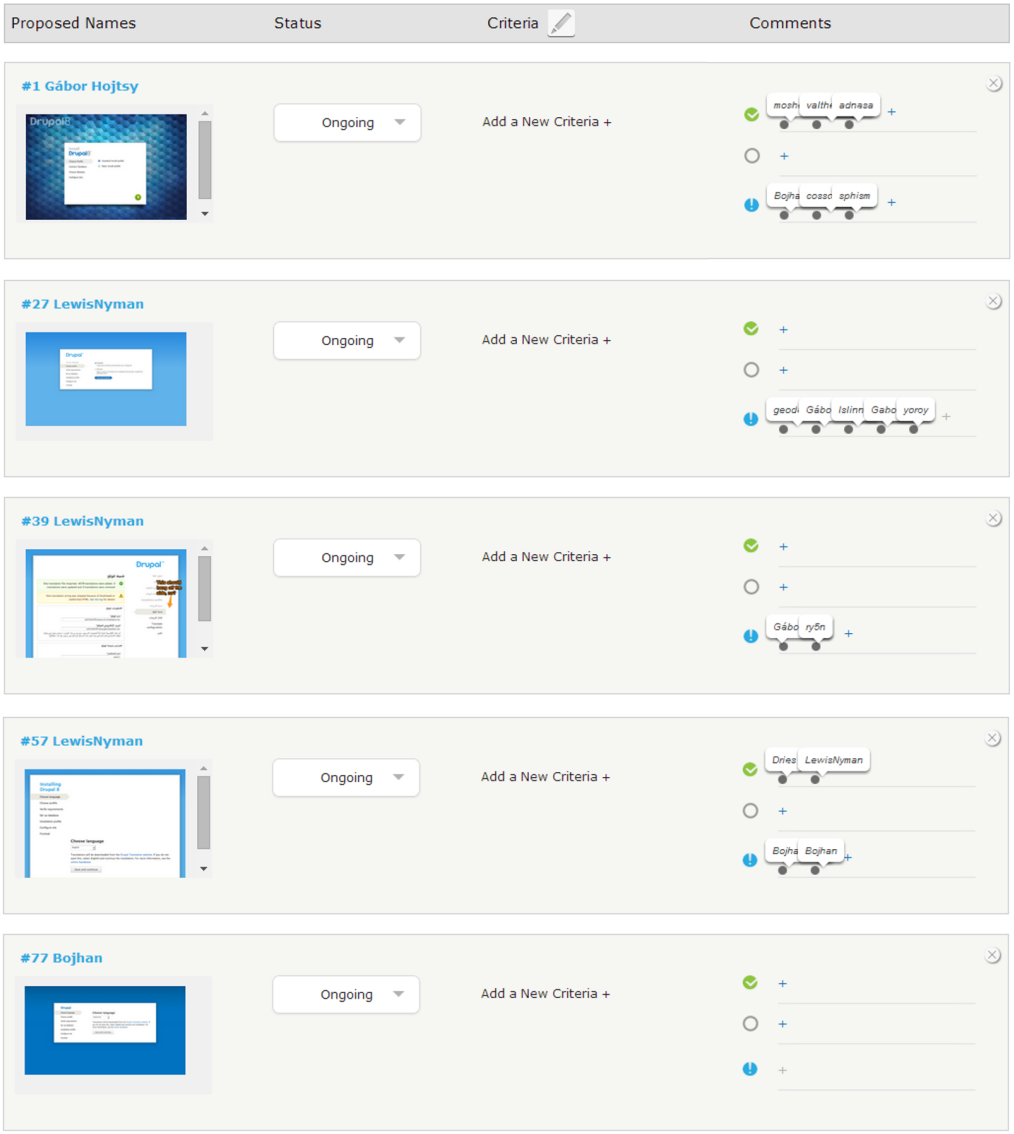

Figure 1.4: A sample Drupal discussion rendered in Procid's idea-centric view. This view summarizes the proposed ideas, their status (Dropped, Ongoing, or Implemented), and the criteria for evaluating the ideas. It also spatially organizes the comments for each idea into supportive, neutral, or constructive categories.

is also possible that participants engage in an effective process by proposing many design proposals, defining criteria, and providing arguments for and against the proposals, but never reach an agreement. This scenario is especially problematic due to the effort invested that did not lead to a beneficial outcome. For example, failure to reach agreement in this case can cause discussion participants to lose interest and therefore the product would not be

improved.

The overarching goal of this thesis is to design and develop a system that can support discussion participants to engage in an effective consensus building process. This system supports the social process of reaching consensus by embodying consensus building strategies, however it cannot guarantee consensus. It is up to the participants to decide how to use the tool features during the discussion to form consensus.

1.5 Scope

Consensus building involves a broad set of procedural, rhetorical, and social strategies [7, 45, 47, 64, 68]. I selected only a subset of the known strategies to design Procid. I prioritized strategies that are feasible to implement, would be difficult to realize without tool support or a moderator (e.g. organizing a lengthy discussion around the ideas), and would form a coherent user experience. The selected strategies provide a minimal but reasonable starting point for exploring how to design a system that brings consensus theory to the practice of un-moderated distributed design discussions. As experience is gained, additional strategies could be implemented and/or made accessible to community members via onboarding or other newcomer socialization techniques [19].

Procid was built to support consensus building in distributed UI design discussions. However, the system and principles should generalize outside design discussions to any situation in which multiple stakeholders need to propose and debate various possible solutions for solving an ill-defined problem where no optimal solution exists, e.g. deliberating business strategies and political disputes. Situations that do not deal with ill-defined problems or multiple stakeholders may not benefit from this tool. In addition to the task type, prior work has shown that group size may have an affect on decision making [12, 66, 40]. For example, a small group (2-5 members) who are meeting face-to-face may not benefit from Procid as much as a larger group (more than 5 members) in distributed or even face-to-face settings.

1.6 Contributions

Taken together, this thesis will be a big step forward for bringing the theory of consensus building to the practice of distributed design discussions. This thesis makes the following contributions:

Quantifying the problem. Consensus building is one of the most important challenges facing online communities. The first contribution of my dissertation was quantifying the fraction of distributed design discussions that failed to reach consensus. My data analysis of distributed design discussions in one popular open source community showed that 42% of these discussions do not reach consensus. Frequently failing to reach consensus causes significant loss of the community's time and resource investment and missed opportunities to improve the software product. It also results in frustration among community members that can make them abandon the community. This result raises awareness of how important consensus building is and what are the consequences of not reaching consensus (Chapter 4).

Empirical knowledge of factors affecting consensus. Researchers have examined how various group factors such as size, task, and anonymity relate to consensus building. However, little is known about how the consensus building process translates from small face-to-face management meetings to large distributed discussions in online communities. Through running regression analysis, conducting interviews, and inspecting several discussion threads I identified factors that affect the consensus building process in distributed discussions. For example, my analysis showed that discussions that have participants with more experience and prior interaction history are more likely to reach consensus. During this analysis I reported user perspectives on the challenges of reaching consensus in UI design discussions, the techniques utilized for addressing the challenges, and the consequences of not reaching consensus. (Chapter 4).

A novel browser extension designed to support consensus building in distributed design discussions. "Procid" is designed based on a subset of consensus building strategies and the lessons learned from the qualitative analysis and the statistical analysis showing the relation

of different factors to consensus. Key features of the tool include (i) the ability to visually track idea proposals and organize the discussion around ideas; (ii) the ability to define persistent criteria and rate the ideas against the criteria; (iii) the ability to register strong support for or against ideas, empowering individuals with a stronger voice in the discussion; (iv) interaction constraints that promote best practices of consensus efforts, e.g., encouraging the first comment on an idea to be supportive of it; and (v) the ability to identify candidates to invite to active discussions using attributes important for consensus in the domain. Procid extends the Web-based discussion platform used for distributed design discussions in one mature open source software community (Drupal.org). However, the tool should be considered as a system design pattern that can be mimicked to bring the benefits of consensus strategies to peer production and online deliberation platforms (Chapter 5).

Results from two evaluations to understand perceptions of and experiences with Procid. The first evaluation gauged initial reactions to the tool and collected feedback from the Drupal community. The second evaluation compared how the use of Procid affects a distributed design discussion relative to the existing platform used in the community. Results of both studies showed that users found the features of Procid beneficial for consensus and perceived the tool as more effective for consensus building than the existing platform (Chapter 7).

Chapter 2

Related Work

This chapter surveys related work in three broad areas. First, I will describe the concept of consensus and provide an overview of the traditional consensus building process. Second, I discuss prior studies on group decision making and consensus building in organizations. Third, I will cover studies in online group decision making. Fourth, I will discuss visualizations or tools built to support group decision making. Finally, I will briefly discuss evaluations of group decision support systems.

2.1 Consensus Building and Group Decision Making

In group decision making, consensus refers to when all participants are willing to commit to a proposal despite the fact that objections may remain [16]. In an ideal situation, consensus is achieved through a process of argumentation where the different viewpoints are equally valued. Thus, good arguments are recognized and the participants come to agree on the most rational solution [38, 50]. The process of building consensus is a distinct form of face-to-face decision making because decisions are reached through mutual consent with each participant given the power of veto, everyone with an interest participates in the process, and the process is constructed based on openness, fairness, and trust [64]. Following a consensus approach can increase participants' understanding of the issues involved, allow the participants to explore various solution proposals, build trust in between participants, and encourage participants to be more committed [64]. Above all, participants should feel good about the outcome and the process through which it was achieved. Due to these benefits, consensus building is an increasingly utilized technique for group decision making, especially in complex task domains such as UI de-

sign where no one person has all the required expertise or resources needed to solve a given problem [68].

A traditional consensus building process consists of two phases: (I) Convening phase and (II) The actual negotiation or consensus building phase. Convening usually involves defining the problem, gathering interested parties together, locating the necessary resources, and planning and organizing the process. The actual consensus building phase involves discussing interests and needs, seeking expert advice, brainstorming different alternative solutions, and deciding on a solution proposal that maximizes joint gains. Participants play a set of four roles during the consensus building phase: a facilitator, a recorder, a leader, and group members. The facilitator is a neutral process guide, the recorder captures the ideas of group members, the leader helps keep a group focused on its task, and group members contribute ideas and listen to other's ideas [68]. All of these roles are essential for a face-to-face consensus building meeting. However, when participants are engaged in a consensus building process through distributed discussions these roles may be eliminated, combined together, or practiced differently. For instance, in distributed UI design discussions, unlike face-to-face meetings, due to the large amount of discussions and the difficulty of acting as a facilitator, most of the discussions lack a dedicated facilitator. This thesis sheds more light on these differences by analyzing distributed UI design discussions through the theoretical lens of consensus building.

2.2 Studies of Group Decision Making in Organizations

Many researchers have examined how various group factors such as size, task, and anonymity relate to group decision making performance in organizations [12]. In these studies, researchers have relied primarily on the use of controlled studies and either considered only face-to-face (FTF) group work or compared technology-mediated and FTF group work. Controlled studies examining the effect of group composition factors in FTF condition, have shown that it is more difficult to reach consensus as group size increases [36]. They have also suggested that social interaction among group members can promote consensus by inducing divergent viewpoints to con-

verge over time [30] and groups whose members had considerable experience working together are superior in decision making than ones with a brief history [35]. Similar studies have also shown that groups that delay any show of affect until after reaching a decision outperform groups in which members readily express their feelings [34]. Controlled studies comparing FTF and technology mediated conditions have shown it is more difficult and takes more time to reach consensus when groups use synchronous or asynchronous communication technology than when working FTF [40, 66]. Studies have also reported fewer remarks from group members [65], less argumentation in the discussion [57], and a reduction in interpersonal communication [75] in technology mediated group work comparing to FTF. Similarly when consensus is required, group members experience less satisfaction when using communication technology than when working FTF [66]. While the results of these studies illuminate important elements of group decision making in organizational settings, they say little about decision making in online communities where communication is asynchronous and a larger group of participants are involved. Part of this thesis fills this gap by providing a comprehensive analysis of real world decisions made in the UI design discussions in a mature OSS community from the perspective of consensus building.

2.3 Studies of Group Decision Making in Online Communities

Researchers have recognized the need for and studied decision making in many distributed communities. For example, Viégas et al. noticed that the Wikipedia's interface better supports consensus building by enabling users who disagree with a statement to easily delete it. The interface also separates conflicts from the articles by offering discussion pages as a separate space to discuss changes to the article pages and build consensus [73]. Burke and Kraut also modeled the promotion decisions in Wikipedia to identify the criteria that affects promotion decisions in practice. They found that editors with a strong edit history and those who frequently submit edit summaries are more likely to receive promotion [17]. Similarly, Kriplean et al. examined how Wikipedia policies are employed through the consensus building process in talk pages [47]. The authors recommended building tools that summarize

behavior to help identify different aspects of a conflict over time and annotating prior instances of consensus and the participants involved. However, none of these recommendations have been implemented or tested. My research extends prior work on consensus building by developing a system that implements many of the prior recommendations and by evaluating the utility of the system with users in real distributed design discussions.

Researchers have also used the open nature of discussion in open source software to study the decision making process. For example Ko and Chilana have shown that the types of arguments and rationale applied in distributed discussions need to be better oriented toward consensus [45]. Similarly, in [39], the authors studied how participation relates to code-related design decisions. One finding was that in the more effective projects studied, the number of participants increases over time and shifts from administrators to other community members. Fielding explained the decision making process in the Apache project where the team was using a simple minimal quorum voting system via email. Anyone on the Apache mailing list could vote but only votes casted by the members of the core group were considered binding [29]. Yamauchi et al. studied two open source projects and found a culture of rational decision making where community members tend to always choose technologically superior options. They identified rational and criteria-based decision making as the only way through which community members can agree when computers limit their communication channel [78]. These studies have analyzed decision making in distributed design discussions from many perspectives. However, my PhD dissertation brings consensus building as a new theoretical lens to study and support decision making in design discussions.

2.4 Tools for Collaborative Group Work

The HCI community has a rich history of building tools that support group work. The design of these tools typically varies based on the proximity of group members, group size, and the task [23]. For instance, in co-located settings, prior work has studied how collaborative use of large or multiple displays affect negotiation [15], software development [14], and design [6, 59]. In distributed settings, researchers have developed Web-based systems to

support tasks such as information security planning [25].

Other research has built tools to support decision making by enabling the decision process to be tracked, archived, and reviewed. For example, gIBIS and Compendium support design decision making by tracking issues, positions, and arguments [21, 63]. Pathfinder is a tool that provides an explicit argumentation model and visualization of that model for engaging citizens in distributed discussions of research questions [55]. Polestar aids intelligence analysts by supporting structured argumentation and fact collection from documents [62] while CommentSpace integrates a small set of tags and link structure with comments to aid evidence gathering and data exploration during collaborative sense-making [76]. The system I developed as part of this thesis is original relative to this corpus of prior work because it promotes the social process of decision making in addition to visualizing and tracking the discussion. This is achieved by leveraging consensus theory in its design.

Another thread of research has created visualizations to support large-scale deliberation. For instance, ConsiderIt is a visualization tool that promotes public deliberation by guiding people to reflect on the perspectives of others on topics of civic interest (e.g. ballot initiatives) [48]. Opinion Space is a Web interface for collecting and visualizing users opinions on topics such as politics, parenting, and art [27]. While these tools focus on supporting the deliberation process, my system focuses on the consensus process, which includes debating multiple proposals, agreeing on one, and fostering a positive experience with the process.

A related thread of research has utilized graph visualizations and interaction techniques to foster collaborative problems solving activities [49, 69, 11, 4]. For instance, Balakrishnan et al. showed that providing a graph visualization of relevant information and shared interaction promotes topical focus and higher solution rates [11]. Alonso et al. proposed a similar graph visualization that represents the closeness of opinions to support consensus building in small groups [4]. While such graph visualizations may help decision making in a small group, distributed design discussions usually involve a large group of participants who are debating design alternatives, their strengths and weaknesses, and implementation challenges using an established workflow. Therefore to support consensus building in distributed discussions, I designed a system that scales well, captures the unique content characteristics of these discussions (e.g. design proposals, implementation discussions,

etc.), and well integrates with the current workflow in these discussions.

2.5 Evaluations of Group Decision Making Support Tools

Evaluation has been identified as one of the major challenges of developing groupware [33]. Evaluating a groupware system usually involves dealing with users with different backgrounds and preferences and it takes longer as the group interactions evolve over time. It is best to evaluate groupware in authentic settings, as lab studies often fail to capture social and motivational dynamics of the group. However, due to the complications of real world evaluations, lab studies are commonly used to evaluate groupware systems.

Decision support systems are typically evaluated based on two categories of measures: decision performance and group members' attitudes. Decision performance usually includes measures such as decision quality, decision speed, and choice shift, while group members' attitudes include decision confidence, satisfaction with the decision making process, satisfaction with the tool, and deindividualization. For example, Gallupe et. al. conducted a lab experiment to compare the effect of a Group Decision Support system called "DECAID1" on face-to-face versus remote settings. They measured decision quality, decision speed, choice shift, decision confidence, and satisfaction with the decision making process [32]. Similarly, when comparing computer mediated and FTF decision making, Hiltz et. al. measured equality of participation, decision quality, amount of communication, and degree of agreement [40]. Lam et al. also conducted a lab experiment comparing decision quality and patterns of group communication in presence and absence of a group decision support system called "Computer Aided Helper" [52]. The Lead Line Chat Interface was evaluated in terms of decision quality, degree of consensus, and user satisfaction with both the process and the tool [28]. In CONFER, the authors investigated how the system affects the quality of outcome, time to completion, and satisfaction with the tool and the process [77]. The evaluation studies explained in this thesis are built upon the above evaluations. However, while the evaluations conducted on prior work mostly focused on measuring decision performance, I focused on users perceptions of and experiences with the consensus-based features of my

tool.

Chapter 3

Quantifying and Supporting Behavior in User Interface Design Discussions

As the workforce in the software industry is becoming more distributed, more interface design discussions are occurring online. Participation in such discussions typically occurs via issue management systems or similar interactive discussion forums. While such systems have a low learning curve, they do not support key elements of design discussion such as comparing alternatives, maintaining awareness of the arguments for and against the alternatives, or building consensus. To better understand these and other challenges, I conducted a study of online interface design discussions. The study consisted of analyzing a large corpus of online discussion content and conducting interviews with designer and developer participants. The results of this study informed the design and implementation of a new interactive visualization tool, called IdeaTracker, that I built to address the identified challenges. In this chapter, I first discuss the methodology and findings of this study. Then I will explain the implementation and evaluation of IdeaTracker that was designed based on the findings.

3.1 An Exploratory Study of Distributed Design Discussions

This study was centered on answering these research questions:

R1 . How many ideas are typically shared in online design discussions? How extensive is the debate? How do designers maintain awareness of the ideas?

R2 . What are the strengths and weaknesses of the issue management systems used for discussing and managing ideas?

R3 . What are other challenges that designers encounter participating in

online design discussions and what strategies are used for addressing them?

3.1.1 Methodology

To answer the above research questions, I studied the evolution of interface design discussions in two well-known open source projects: Ubuntu and Drupal. Ubuntu is a popular Linux distribution and Drupal is a widely used content management system. I chose these two projects because they have an active interface design team, the design discussions are publicly accessible, and they contain a considerable amount of lively design discussion.

Within both of these projects participants post interface design problems, solution proposals, and related comments to the respective issue management system. These issue management systems are Web-based interactive forums dedicated to discussing software related issues including interface design issues. These forums are similar to the interactive forums used for discussing topics of interest in many other online communities or social media sites.

As shown in Figure 3.1, each interface design discussion (a new thread) starts when a participant posts the description of a design problem. Other participants can then contribute to the discussion by proposing design alternatives, arguing for or against the proposed alternatives, attaching an implementation of an alternative (called a patch), reporting the results of a patch review, or raising other concerns such as clarifying the scope of the problem or how it relates to other ongoing design efforts.

I analyzed the content of these online interface design discussions and conducted interviews with participants. For the quantitative analysis, I examined 1560 messages that spanned thirty discussion threads. Fifteen of the threads came from a pool of 300 in Ubuntu (average number of messages in each thread=35.4 σ=12.9). The other fifteen come from a pool of 500 threads in Drupal (average number of messages in each thread=68.6 σ=23.0). In both cases, I selected the fifteen threads from the most active threads. From inspecting a sample of the threads; I defined two characteristics for activity; (i) the number of images, where I considered an image to be a proxy for a proposed design alternative and (ii) the number of messages in that discussion. I rank ordered the pool of discussion threads based on these criteria and selected fifteen threads from the top fifty threads in each project.

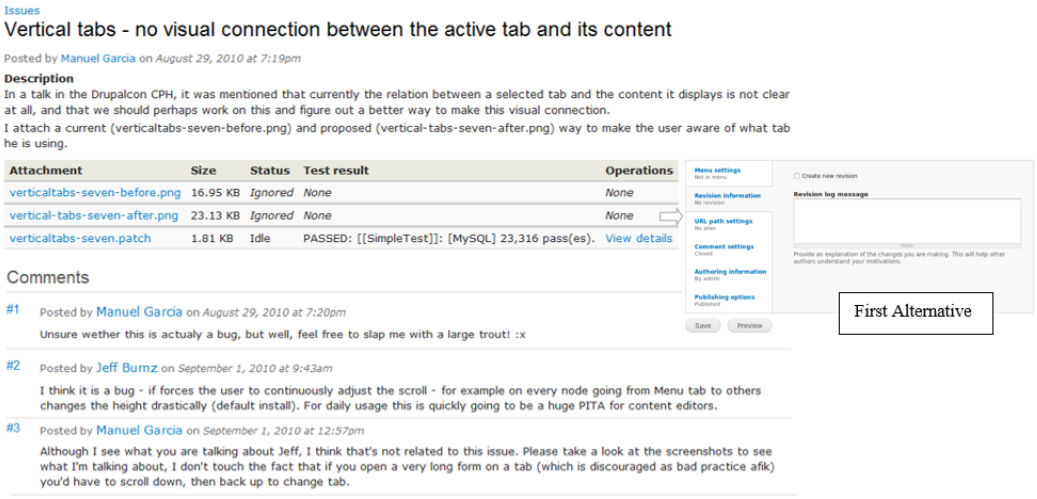

Figure 3.1: A sample interface design discussion thread in Drupal. The interface design problem discussed in this thread is the lack of visual connection between a selected tab and its content. An alternative has been proposed along with the problem description. In the first few comments, participants are discussing the scope of this problem and whether it is a valid issue.

I then analyzed the content of the messages. First, I divided the message into a smaller set of topical chunks. A new chunk was created when there was a transition from one topic to another. Then, another HCI researcher and I manually coded the chunks into eight categories: Issue, Alternative, Criterion, Clarification, Project Management, Implementation, Digression, and Other. We adapted these categories from the coding scheme developed by Olsen et al. [60] for capturing and analyzing the core elements of design discussions in collocated settings. Though the original coding schema had eleven categories; our adaptation only used seven. When testing the schema and resolving inconsistencies (based on five discussion threads from each project), we agreed that the other four categories were not applicable. An implementation category was added to capture the technical messages in the discussions.

The data analysis was complemented by a set of semi-structured interviews with twelve participants from both Ubuntu and Drupal projects. I interviewed six active designers, three from Drupal with an average of 7 years of experience ($\sigma = 1$) and three from Ubuntu with an average of 7 years of experience ($\sigma = 3.26$). I will refer to Drupal designers and Ubuntu designers as DD# and DU# accordingly. I also interviewed six developers

participating in resolving UI design and usability issues: three from Drupal with an average of 6.5 years of experience ($\sigma = 3.51$) and three from Ubuntu with an average of 7.5 years of experience ($\sigma = 4.5$). Each interview lasted about an hour and was conducted via phone and instant messaging. The latter was used to share Web links, images, and other data artifacts during the interview. The subjects were compensated with a $30 Amazon gift card.

The interview questions reflected the main research questions of the study and were informed by prior work (e.g. [9]). For example, the questions included: what are the main challenges in discussing design ideas online? How do you maintain awareness of the ideas? What techniques are used to promote consensus? What are the strengths and weaknesses of the issue management systems for discussing ideas?

3.1.2 Results

I discuss key challenges that designers encounter while participating in interface design discussions online and the methods used for addressing those challenges. Although this study focused on open source projects, the challenges identified were mostly due to the intersection of the interaction design of the forums and the topic of the discussion. As a result, I believe these observations apply to any interactive forum which serves as the primary means for discussing interface design issues.

Designers Struggle to Track Design Alternatives

A total of 299 alternatives (ideas) were submitted in the 1560 messages analyzed. On average, nine alternatives were proposed in each thread of discussion (σ=5.88, max=27, min=0), indicating that multiple alternatives were welcomed and considered for each design problem. Of all the alternatives proposed, 63% were described solely in narrative form, 13% included only a patch (a file that updates the implementation to provide a working preview of the idea), and 18% included only a screenshot of a proposed solution. Figure 3.2 shows a distribution of alternatives for each combination of modality.

Although I chose part of the data set from the discussions with the most images, more than half of the alternatives were presented in narrative form. This is partly due to the lower cost of expressing alternatives in narrative form. Other reasons for having fewer visual representations may be the

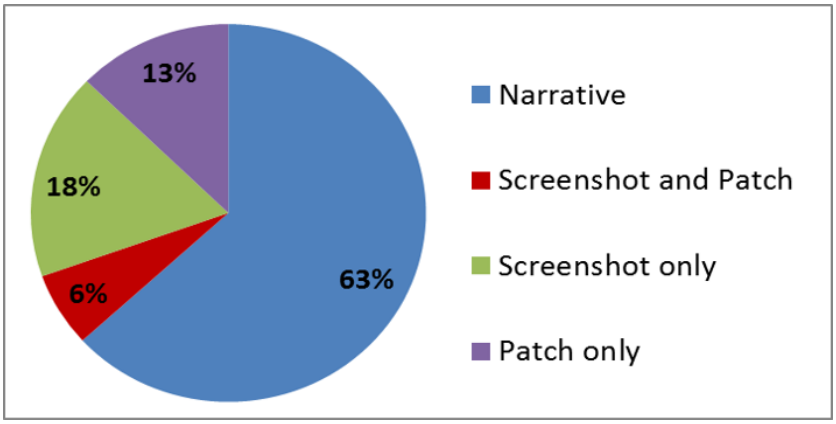

Figure 3.2: Distribution of alternatives over each combination of modality, aggregated across all messages (N = 1560).

text-based structure of the discussion threads and lack of emphasis on the importance of visually demonstrating alternatives by the discussion interface. However, all of the designers were aware of the importance of screenshots and other visual representations in explaining a design alternative. They mentioned that alternatives with visual representations have a better chance of receiving comments from community members. For instance DU2 said:

> *wireframes or other visual stuff legitimize ideas. Because they're very memorable. [...] People comment on images more, or codes more than paragraphs.*

I was able to confirm this claim by analyzing all usability discussions in Drupal.org. I found that the probability that a comment with an image or a patch receives a reply is 0.29, while the probability that a comment without an image or patch receives a reply is 0.13. I also found that the comments that include patches and the comments that include images on average receive 1.6 and 1.5 replies respectively. However, the comments without images or patches on average receive 1.3 replies. One way to improve the discussion interface is to better emphasize the importance of alternatives and the significance of providing visual representations for them by adopting a more visual structure.

Confirming the importance of alternatives, my quantitative analysis revealed that 48% of the conversation is spent discussing alternatives (Figure 3.3). Designers also mentioned alternatives as a vital piece of information when contributing to a design discussion. As DD1 said:

[When participating in a new discussion] I would want to know, in its current state, what is the exact problem the issue is trying to deal with, what are the proposed solutions so far, and what direction have people been taking on each of the proposed solutions.

However the current systems do not support tracking the proposed alternatives. As DU3 said:

They (bug reports and mailing lists) arent good for keeping track of all of the ideas. [...] There usually isnt anybody keeping track of these are all the possible options and these are some [discussions] about each option.

Today, the only way a participant can maintain awareness of the proposed alternatives is to rely on their memory, maintain notes in a separate tool, or scan the entire discussion thread to review the alternatives and rationale for and against each of them. The first method is unreliable as human memory is fragile and has a limited capacity [10]. The latter two methods are cumbersome and do not promote shared awareness among participants.

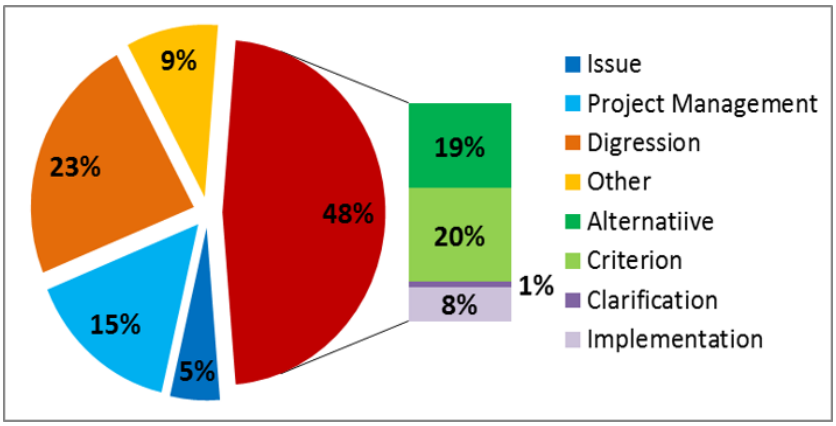

Figure 3.3: Distribution of coding categories. Summing the categories for alternative, criterion, implementation, and clarification indicates that 48% of the discussion was devoted to alternatives.

One of the consequences of using an interactive forum for managing a design discussion is that many of the proposed alternatives get buried in the midst of the discussion. As DU4 said:

Some ideas that people have are actually really good, but then they kind of get lost in the thread...

This means that not all alternatives will be evaluated thoroughly or may be forgotten during a lengthy discussion.

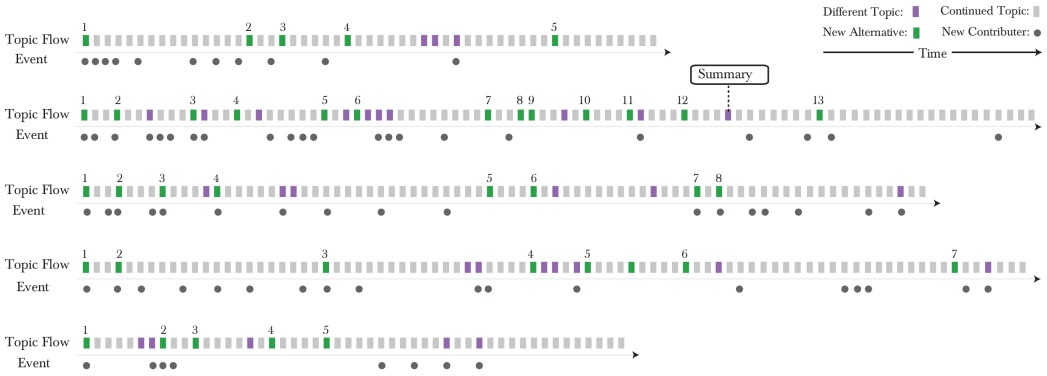

Figure 3.4: An event timeline for five complete threads of interface design discussion from the Drupal project. The timeline shows the fluidity of topics along with visual indicators of when new alternatives were posted and when new community members joined the discussion thread.

Figure 3.4 demonstrates some of these challenges by showing an event timeline for five complete threads of interface design discussion from the Drupal project. Though I only show five, the other threads analyzed exhibited similar patterns. The timeline shows the fluidity of topics (i.e. whether each message continues the topic of the previous message or changes it) along with visual indicators of when new alternatives were posted and when new participants joined the discussion. This figure offers interesting insights about the flow of conversation in the interface design discussions.

For example, within these threads, the proposal of design alternatives is distributed throughout the discussion, rather than batched at the very beginning reminiscent of commonly applied face-to-face brainstorming techniques. This may be a consequence of the distributed nature of the discussion in time and space as well as its integration with an issue management system. It may also reflect the fact that participants are able to generate and submit new alternatives as a function of the ongoing discussion. However, an important consequence of having the alternatives distributed throughout the discussion is that some may be overlooked or even fade from community memory. As DU3 explained:

> *Even if 20 or 30 ideas get generated during the mailing list discussion, a few days on people will be discussing [only] one or two*

which might be the worst ones because they might be the most controversial.

A related pattern is that the topic of the design discussion changes frequently. For example, in the first thread, there is a topic change after A3 (Alternative 3), A6, A8, and A11. Because of these topic changes there may be a reduced chance for these alternatives to be evaluated. In this case, A8 and A11 didnt receive any comments, and A3 only received one. Indeed, the lack of structure in the discussion sometimes prompts a participant to write a summary of the discussion to date, including the alternatives and opinions of those alternatives. One such summary is called out in the first thread in Figure 3.4. In this case, the participant wrote the summary mainly to compensate for the lack of awareness in the system. Twelve ideas were proposed and people were struggling with deciding which one works best. The summary reminded them of the goals and the description of each alternative.

Another interesting pattern is that the majority of the participants joined during the first half of the discussions, but continued to join throughout. As the discussion grows, those who join later or otherwise do not keep up must review the messages to track the alternatives, the arguments for and against them, and the current consensus of the other participants. The common method for acquiring this information is to (re-)read the discussion to date. But, since this is time consuming, some participants will post irrelevant comments that hinder the flow of the discussion. As DD1 said:

...one thing that gets very frustrating in this, it gets very frustrating when I'm involved in a long discussion and have been for the whole time, someone will often come in and just kind of jump in to the discussion and either drill it and say "Oh, this is such a great discussion I also noticed this other problem with this other issue or this other thing" and people will go off on a tangent for two weeks talking about this other thing and we've gotten away from what the core issue is which is can be frustrating.

To avoid losing bright ideas and to have a more organized discussion, the current systems should support better tracking of ideas. Highlighting the alternatives and connecting the messages that reference them can greatly aid participants. It could also reduce the time required to identify, compare, and

consider the alternatives without having to sift through all of the textual comments in the discussion thread.

Integration of UI Design and Development Activities is Essential

Designers and developers currently participate in the discussions through a centralized issue management system. This centralized venue helps designers in building trust and gaining merit by enabling them to interact with developers and exhibit their skills. Once they gain respect as a designer, they can more easily convince developers to implement their suggested improvements [70, 9, 83].

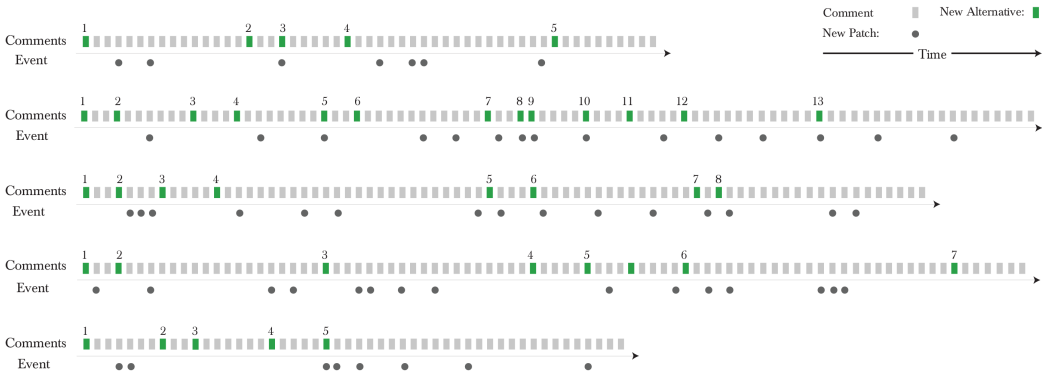

Figure 3.5: The timeline from Figure 4 showing only the introduction of alternatives and patches.

Integrating design and development activities also helps designers collaborate with developers. This collaboration is necessary for designers to receive feedback on the feasibility of their desired improvements [58] and for developers to be advised on the interaction design of their implementation. This iterative process of interface design and development is visible in current discussion threads. As shown in Figure 3.5 the proposed alternatives and submitted patches are distributed throughout the discussion threads, where each alternative is usually followed by a number of patches.

Any new interface for supporting interface design activities online should be fully integrated into the respective issue management systems. This will allow designers and developers to build mutual trust and collaborate more effectively.

Participants Need to Be Aware of Others Opinions Regarding Alternatives

Discussion participants typically demonstrate agreement by writing "+1" for a favored alternative or they simply state that they like the idea. In order to determine the current direction and the favored ideas, participants must read through the messages. Another option would be to ask others to clarify the current direction. However, participants may have inconsistent perceptions about the direction of the issue. As DD1 said:

> *...Its often hard to figure out what is the current direction. Thats definitely hard to do. Often it takes getting someone to clarify it. And not everyone would clarify it the same way. If there are two people, and theyre each kind of pushing their own ideas, within a Drupal issue ... and you were to go on IRC and ask each of them individually, "So whats the current direction?" youll get two very different answers.*

The current issue management systems lack a formal way of expressing ones preferred idea and visualizing others preferences. The absence of a mechanism to share opinions can hinder the consensus building process. Today, the consensus building process can be lengthy and it can be difficult to determine whether consensus has been reached at all. As DU3 said:

> *People can keep on arguing the point, long after the decision was made [...] The nature of the way that many online discussions work is that they let the discussion continue [indefinitely]. Thats the main difficulty.*

Implementing a mechanism to share preferences and formalize the consensus building technique (e.g. a voting system) may help facilitate the decision-making process. Also, it will be effective to highlight the alternative that has the consensus so far. Bringing the consensus to light can help developers determine which alternatives need to be implemented to improve the project or further inform the discussion.

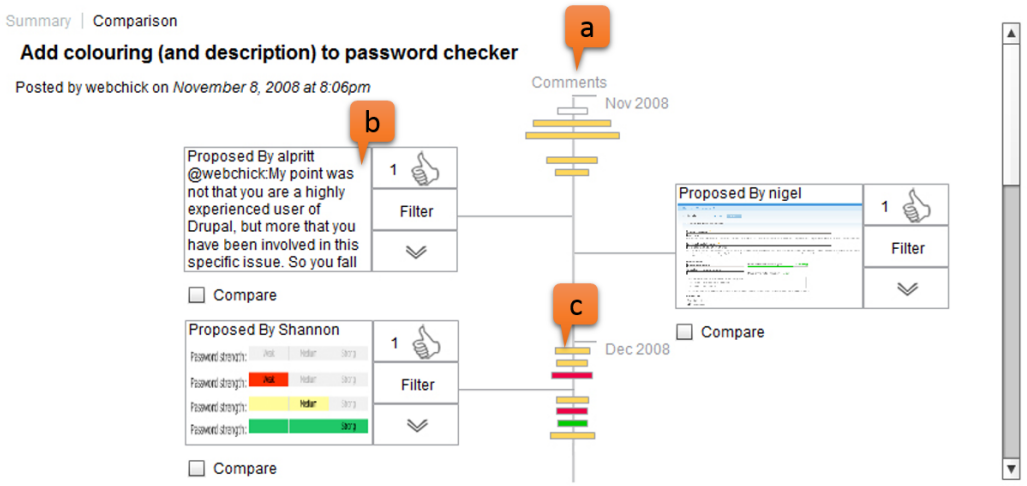

Figure 3.6: The main screen consists of an interactive visual timeline, highlighting alternatives and offering an abstract summary of the comments. (a) The timeline shows the chronological order of comments and alternatives. (b) The alternatives are shown in callouts so designers can easily track them. (c) The comments are represented by a rectangle whose width corresponds to the length of the comment. The rectangles are colored based upon their affective tone.

3.2 IdeaTracker: An Interactive Visualization Supporting Collaboration

In this section, I describe how I translated the above implications into the implementation of an interactive visualization tool for reviewing online interface design discussions - IdeaTracker. The tool was developed through an iterative design process, starting with four different prototypes that addressed the challenges identified in the study. An informal user study was conducted on these prototypes. For the study, each prototype was seeded with data from an actual design discussion. Four users representative of the target audience were recruited and asked to perform similar tasks (e.g., identify the idea that reflects community consensus) with each prototype and the existing interactive forum interface. The users were then asked to explain the strengths and weaknesses of each prototype. From the results, I implemented the final prototype of IdeaTracker.

3.2.1 User Interface

I first discuss the main interface components of the system and then illustrate its value through a user scenario. All of the figures illustrating the use of the system are based on data imported from an actual design discussion in Drupal where participants are proposing and debating alternatives for a revised password checker. To facilitate use and learning of the interface, all interactive controls in IdeaTracker have a tooltip which explains their functionality. For the visual elements, the user can access a short description of each element via a context menu.

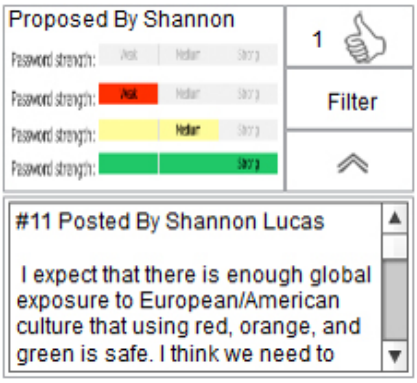

Figure 3.7: The user can press the expand button to reveal the entire post explaining the proposed alternative.

Tracking Alternatives

IdeaTrackers main screen consists of an interactive visual timeline, highlighting the alternatives proposed in the current thread of discussion and offering an abstract summary of the posted comments (Figure 3.6). The timeline illustrates when a specific comment or alternative has been posted (Figure 3.6a). This timeline allows participants to gauge the amount of activity that has occurred within a specific timeframe as well as the overall progress and pace of the discussion. The alternatives are shown in separate callouts so designers can easily identify and track them (Figure 3.6b). If an alternative has an attached screenshot, the screenshot is shown in the callout; otherwise, the first few sentences describing the alternative are shown. All other comments are represented by a thin rectangle, with the width of the rectangle corresponding to the length of the comment (Figure 3.6c). The comments are colored based upon their affective tone. If a comment has

a negative tone, the rectangle is colored red. If it has a positive tone, the rectangle is colored green. If the comment has both positive and negative words, then it is colored yellow. The number of negative and positive words is computed by looking up each word in a commonly used dictionary. This color coding allows designers to quickly assess the community opinion of a certain alternative. Designers can easily skim through the comments related to a specific alternative without having to read the text of each message. To aid in exploring alternatives and the comments regarding each alternative, three interaction mechanisms have been implemented:

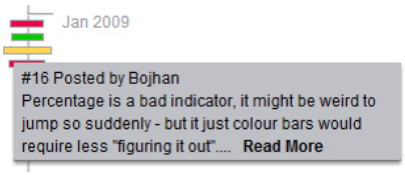

Figure 3.8: Placing the cursor over a comments representation will open a window containing its first few sentences.

Expand Alternatives and Comments: The user can select the expand/collapse button next to each alternative and read the entire post explaining the proposed alternative (Figure 3.7). Also, hovering over each comment representation will open a window containing the first few sentences of that comment (Figure 3.8).

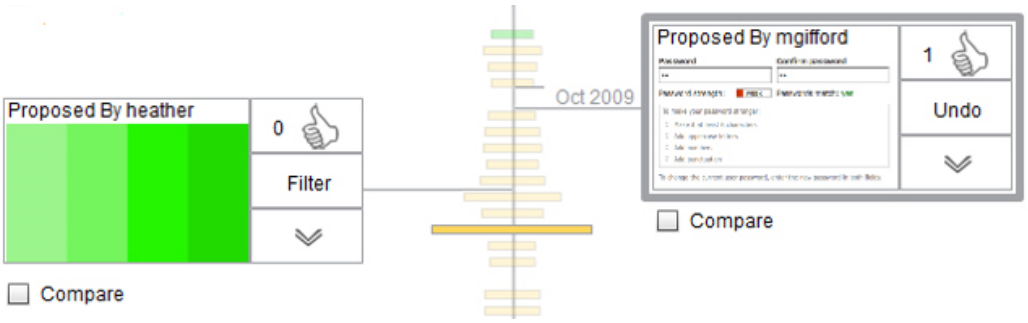

Figure 3.9: Users select 'filter' to dim comments unrelated to the alternative. Here selecting filter for the alternative in the top left shows one message referencing it in context of the discussion.

Filter Unrelated Comments: To examine the comments related to a particular alternative, designers can press the filter button next to the alternative. This dims all of the representations that do not reference this alternative (Figure 3.9). This interaction isolates the pros and cons of an alternative

pointed out by other designers. It also aids in detecting the alternatives that have received insufficient or controversial discussion.

Link to the Original Post: The user may want to read the original post corresponding to a comment or an alternative. To make this interaction possible, a link is provided in both expanded versions which redirect the user to the original post corresponding to that particular alternative or comment. Also, the title of the issue at the top of the main screen links to the original discussion thread.

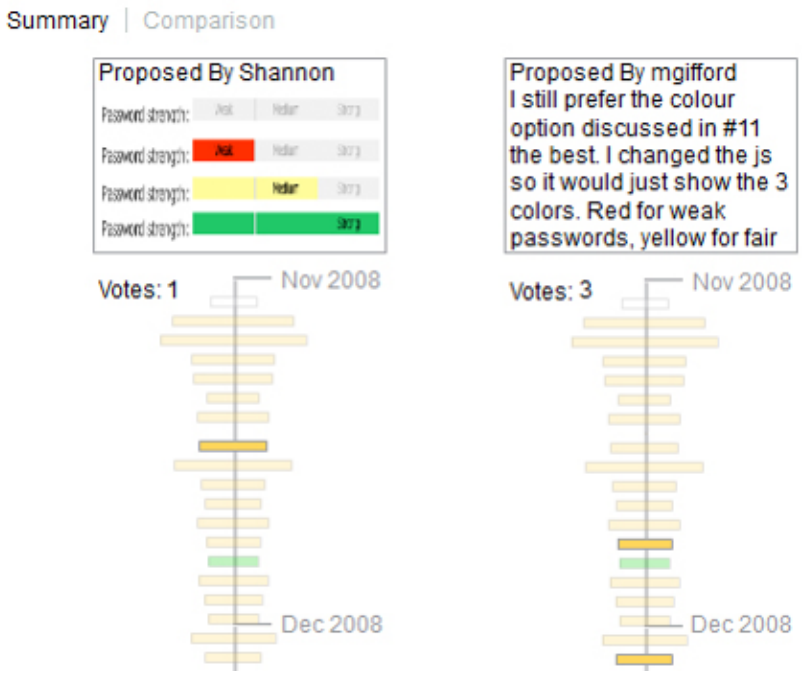

Figure 3.10: The compare view shows a timeline for each selected alternative and the comments referencing them. All of the comments are shown under each alternative, but the comments unrelated to the particular alternative are dimmed.

Comparing Alternatives

To compare different ideas, IdeaTracker offers a comparison view. Users select the ideas they would like to compare by clicking on the check-box at the bottom of each idea. Then, selecting the compare link will redirect them to the compare view (Figure 3.10). This view shows a timeline for each idea and the representations of comments referencing those ideas are available on the timelines for comparison. To provide users with a reference point

for comparison, all of the comments are shown under each idea. But, the comments that are not related to a particular idea are dimmed.

Voting System

A voting system has been implemented in IdeaTracker to aid designers in promoting and reaching consensus. The number of votes for each alternative is shown on the vote button next to the alternative. Hovering over the vote button will show the list of people who voted for the idea. The user can vote for an alternative by clicking on the vote button. If the user clicks the vote button, the number of votes for that idea will increase by one and the vote button will be highlighted to indicate which idea the user has voted for. To synchronize IdeaTracker with the original issue, a comment will be automatically posted to the original issue on behalf of the user stating that the user favors that particular alternative. Conversely, if a user posts a comment using the common notion of "+1" for an alternative in the original discussion thread, the number of votes for that alternative will be updated in the IdeaTracker.

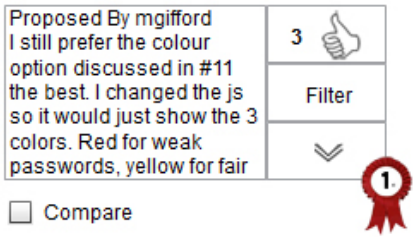

Figure 3.11: To promote awareness, the alternative currently with the most votes is highlighted.

Through IdeaTracker each user can only vote for one idea. If a user votes for a different idea, her initial vote will be re-assigned to the new idea. This feature enables users to retract their votes if a better idea is proposed or an existing idea is refined. This way IdeaTracker reflects the participants recent views about the proposed alternatives. To promote awareness of the current consensus, the idea with the highest number of votes is highlighted in the system (Figure 3.11).

User Scenario

Bob is a UI designer who contributes to open source projects in his spare time. Looking through the usability issues in Drupal, he finds an issue about improving the usability of Drupals password checker. He launches IdeaTracker and enters the URL corresponding to the discussion thread for that design issue.

He first wants gain awareness of the proposed alternatives and their pros and cons as pointed out by other participants. He quickly scans the list of alternatives highlighted on the main screen and notices the idea posted by Lisa that suggests borrowing the design of Google's password checker. Bob then expands the idea to read it in detail and selects the filter button to identify the comments that reference that idea. He immediately notices a negative comment colored in red and hovers the cursor over the comment to read it. The comment has been posted by Mark who thinks that copying and pasting a design from Google will spoil Drupal's trade mark. Bob agrees, so he continues to scan the alternatives to determine if someone else has proposed a better solution. At the end of the discussion, he finds Anne's idea. Anne's idea is a tweak of Lisas idea. She suggests that instead of using red, orange, and green to indicate a weak, medium, and strong password, they can use different shades of green. Bob decides to compare Anne's and Lisa's ideas. He selects the compare checkbox next to these ideas, selects the compare link, and is redirected to the compare view. He reads the comments posted regarding each idea and compares them. He decides that he likes Lisas idea more than Annes. He returns to the main screen and votes for Lisa's idea. A comment is generated on Bob's behalf and posted to the original discussion thread indicating that Bob is in favor of Lisa's idea.

In contrast, using only the current interface, Bob needs to review each comment to identify the alternatives and their pros and cons. He then either needs to use another tool to create a summary of the discussion or rely on his memory. Comparing the two alternatives would also be challenging because it is difficult to isolate the comments that specifically address only the desired alternatives. Finally, since there is no running tally of "+1" votes, it is difficult to identify the currently favored idea.

3.2.2 Implementation

IdeaTracker is fully implemented and its interface was written in ActionScript using Adobe Flex 3 interface framework. The software consists of two layers: the data and visualization layers. The data layer parses the collected data and translates it into an internal format understood by the visualization layer. The data layer receives the data in XML format. The XML data consists of a set of $\langle comment \rangle$ tags, and each $\langle comment \rangle$ tag should have $\langle author \rangle$, $\langle content \rangle$, $\langle date \rangle$, and $\langle image \rangle$ tags.

When the user launches IdeaTracker to view a particular discussion thread, an adaptor component parses the html source of the thread and converts it to the XML format readable by the data layer. The data layer is independent of the html format and only depends on the XML format. In order to apply IdeaTracker to design discussions on other interactive forums, an adaptor component needs to be written that translates the html source of that forum to the XML format readable by our tool.

The data layer processes the XML file to find all the posts. In order to find the alternatives the system uses two heuristics: (i) it considers posts with image attachments as alternatives and (ii) it considers posts that have been referred to by other posts as alternatives. Initial testing indicates that these heuristics accurately detect most alternatives in a thread of discussion.

After detecting the alternatives, the system uses natural language processing techniques to infer the comments related to each alternative and the tone of each comment. To find the comments related to an alternative, the system looks for certain key phrases participants commonly use to reference a comment, for instance: "#34" to refer to comment number 34 or "@Lisa" to refer to the latest comment posted by Lisa. In order to determine the affective tone of a comment, the system finds the number of negative and positive words in the comment using standard online dictionaries. Based on the percent of negative and positive words, it assigns respective values to that comment, which is used by the visualization layer for color coding.

3.2.3 Evaluation

I conducted a qualitative evaluation to assess design choices and gauge initial user reactions to IdeaTracker. The evaluation was performed using the implementation of the tool as described in the previous section. It involved

eight designers and six developers who actively contribute to interface discussions in Drupal and Ubuntu. The evaluation started with an introduction to IdeaTracker and a demonstration of its main features. Afterward I asked participants about their perceptions of the overall direction and different features of the tool (e.g. what do you think about showing the ideas in separate callouts, providing an abstract visualization of the comments, and using color codes for the affective tone of the comments?) and encouraged them to respond openly. Each session lasted about thirty minutes.

Overall, the participants reacted positively to IdeaTracker. All of the participants appreciated the visual separation of ideas from the other comments and being able to filter comments related to a particular idea. For example, one Drupal designer said:

> *The most useful feature to me is the callout of the major ideas that cuts through all lot of the crufty comments*

While another said:

> *I like this sort of compressing it... here is some big comments, here is a bunch of small comments, and if they are generally in favor or not, sort of at a glance as an overview is very cool.*

Most of the participants appreciated having access to an abstract visualization of the comments and felt I was using reasonable decision rules for identifying ideas and filtering comments. Participants also appreciated the fact that IdeaTracker was seeking to complement the existing issue management systems rather than trying to replace them.

The evaluation also highlighted several opportunities for improvement. For example, some of the participants were unsure of the utility of the idea-centric comparison view. Instead they preferred the ability to filter comments based on user id, thereby allowing them to see the comments that one user made across all of the ideas. Participants also expressed that the content of the negative and positive arguments for an idea was more important than the number of votes. For example, as one Drupal designer said:

> *Often there are issues though where an idea has lots and lots of "likes" until one person discovers why it shouldnt be done*

It may therefore be useful to extract the arguments for and against an idea and represent them within the main visualization. Most of the designers were concerned about accuracy of coloring comments based on affective tone and suggested to color comments based on their type (e.g. code review or patch). Participants also asked for more information to be included in the visualization of each comment (e.g. who posted the comment).

3.3 Discussion

This study was designed to quantitatively and qualitatively understand the challenges of participation in the issue management systems. One of the major challenges that emerged both from interviews and evaluation results was *consensus building*. During the interviews participants explained the consensus building process, what is hard about this process, and the techniques that they use to foster consensus. Analyzing the interviews, my initial thought was that the problem of consensus building can be resolved by a structured decision making mechanism such as a voting system which was implemented in IdeaTracker. However, during the evaluation I observed a resistance towards such a voting mechanism. Participants told me that it's not about choosing an idea that has the most number of votes, but it's about choosing an idea that works and everyone can live with. Because of the criticality and lack of support of consensus building for distributed UI design discussions, the rest of my thesis is dedicated to first understanding this process and then it describes the design, implementation, and evaluation of a new interface for distributed discussions that can support consensus building.

Chapter 4

Quantifying Consensus Building in Distributed Design Discussions

From my previous studies I found out that consensus building is a common and critical task in distributed OSS discussions. Therefore, it is important to understand how often consensus is (not) reached, what techniques are utilized to foster consensus, and which elements of a design discussion affect consensus, among many other interesting questions. To answer these research questions, I conducted a mixed-methods study where I relied on my previous 12 interviews, 5 new extended interviews with more focus on consensus building, and a detailed quantitative analysis of interaction data gathered from OSS discussions. The interviews captured user perspectives on the challenges of reaching consensus, techniques employed for building consensus, and the consequences of not reaching consensus. The data analysis was performed to determine how different elements of the content, process, and user relationships in the design discussions affect consensus.

In this chapter, I report the results of this study. I first explain the interview results in a section called "User Perspectives" and then the quantitative data analysis results in a section called "Consensus Building Model".

4.1 User Perspectives

The interviews aimed to better understand the nature of consensus building from the users' perspective and centered on answering the following research questions.

R1. How important is consensus building in these types of UI design discussions, what are the key challenges of reaching consensus, and what are the consequences of not reaching consensus from the user's perspective?

R2. What techniques are used to promote consensus around specific alternatives and how effective are these techniques?

4.1.1 Methodology

I conducted 17 semi-structured interviews with designers and developers participating in either of two open source projects, Drupal and Ubuntu. Eight designers were interviewed, five from Drupal and three from Ubuntu, with an average of 4.5 years of experience in the community (σ=2.6). Nine developers were interviewed, six from Drupal and three from Ubuntu, with an average of 5 years of community experience (σ=2.6). Each interview lasted about an hour and was conducted via phone (n=14) or IM (n=3), whichever a participant preferred, and remuneration was either a $25 or $30 gift card depending on the duration of the interview. I will refer to Drupal and Ubuntu designers as DD# and DU# and Drupal and Ubuntu developers as DevD# and DevU#, respectively.

I first asked a participant to describe one or two recent or memorable discussions s/he participated in. In context of these discussions, I asked the participant to describe the consensus building process, what is hard about this process, the techniques utilized to foster consensus, the factors affecting consensus, and the consequences of not reaching consensus. Twelve interviews were conducted prior to the data analysis and five were performed afterward. For the latter interviews, a few questions were added to probe further about specific results of the analysis.

Interviews were coded to derive common themes using a Grounded Theory approach [67]. The results were used to gain insight into the consensus building process, identify features to include in our interaction analysis, and help interpret the results.

4.1.2 Results

Drawing from the interviews, I report user perspectives on the benefits of consensus and the consequences of not reaching consensus, the challenges of consensus building, and techniques used for promoting consensus.

Benefits of Reaching Consensus and Consequences of Not Reaching Consensus

Participants stated that reaching consensus in the UI design discussions was critical for building a better product (n=4) and for strengthening the com-

munity (n=4). To both of these points respectively, one participant (DD4) explained:

> *...when we reach consensus we are taking our strengths [to] make the world together, we have something that is at least as good as what the two of us could bring to separately, and probably is better because our strengths tend to reinforce each other.*
>
> *...The more we can reach consensus in itself, the fact that we reach consensus in itself, foster stronger feeling in community, so the little instances of that build on one another and help us become stronger as a community, and therefore more likely to invest in reaching consensus on other projects, on other issues.*

On the other hand, the inability to reach consensus can result in an unimproved product, build resentment in the community, and demotivate community members to the point of leaving the discussion or the community altogether.

> *[Consequences of not reaching consensus are] stupid interfaces surviving yet another version in Drupal, known issues not being fixed, frustrated contributors. Consequences can be that people disappear for a couple of weeks or entirely because they get burned out on a too long discussion that didn't reach consensus... [DD5]*

The inability to reach consensus also causes the loss of significant community effort. For example, of the 577 UI design discussions we analyzed, 241 (42%) did not reach consensus. These discussions contained 4968 messages and 460 patches, contributed by 1934 participants. This outcome highlights the need for techniques for enhancing consensus building within the UI design discussions.

Challenges of Consensus Building

Despite recognizing the importance of consensus building participants identified key challenges that make consensus difficult to achieve. For example, one challenge is bridging the different perspectives and needs of the community members engaged in a UI design discussion (n=8):

> *There are many different use cases for Drupal, what is optimal for one use case may be suboptimal for another, and there are strong differences of opinion within the Drupal community about which use cases, if any, should be given preference... Some people build for small sites, some people work on large sites, some people are designers, others are developers [and] others are end users...[DevD2]*

Another challenge is overcoming a strong sense of ownership over one's contributions (n=4). For example, one reason that members contribute is because they can adapt the software to their own needs [37]. However, building consensus requires members to detach themselves from their own contributions and consider alternatives:

> *People have egos and they have a lack of human contact with the people that they are talking to and trying to discuss with and a lot of time because these ideas are our own creations and our own feelings it's very difficult to separate ourselves from our own egos.[DevU1]*

Similarly, expressing strong emotions during a discussion can also hinder consensus building (n=2). As DD4 said:

> *certainly, there've been some that people just got so frustrated that their emotions, me included, to some extent let our emotions lead the way of communicating, rather than communicating based on the facts that matter...*

Other challenges for building consensus identified by interviewees included not having enough participants who are interested in a discussion (n=2), having too many participants lacking necessary background or general design knowledge (n=2), the absence of evidence supporting various claims (n=1), and lack of time and resources (n=1). All the challenges for building consensus identified by interviewees are listed in Table 4.1.

Techniques for Promoting Consensus

From the interviews, I identified different techniques that designers and developers use to promote consensus. The interviewees said that providing

Challenges	(1) Discussion participants have different needs and perspectives.	7
	(2) Due to a sense of ownership, participants may fail to consider others' perspectives	4
	(3) Participants cannot communicate clearly due to a low bandwidth.	3
	(4) Participants let their emotions lead the way of communication.	2
	(5) Discussions become one-sided due to the lack of variety in viewpoints.	2
	(6) Participants lack background knowledge, especially in design.	2
	(7) There is not a defined product vision.	1
	(8) Discussions lack convincing evidence.	1
	(9) There is not enough time for implementation.	1
	(10) Participants who do more have a louder voice than others.	1
	(11) It's hard to undo a decision and go backwards.	1
	(12) There are not enough time and resources.	1
Techniques	(1) Participants provide evidence (e.g. usability tests, design argument, and outside resources)	7
	(2) Participants write patches or provide screenshots for their ideas.	5
	(3) Branch maintainers make a call.	5
	(4) Participants write a summary of the discussion.	3
	(5) Participants with experience and reputation lead the discussion.	3
	(6) Participants spend time to understand other people's viewpoints and experiences.	3
	(7) Participants discuss over IRC.	3
	(8) Participants vote to decide between contentious alternatives.	2
	(9) Participants will call attention to a discussion.	2
	(10) Participants spend a lot of time on a discussion to meet a release deadline.	1
	(11) Participants propose a new idea.	1
	(12) Participants explain to others how a proposal is in their best interest.	1
	(13) Participants make others focus on the discussion.	1

Table 4.1: User perspectives on key challenges of reaching consensus in the design discussions and the techniques utilized for promoting consensus.

evidence in support of a design proposal can better convince opposing parties in a discussion and accelerate consensus building (n=7). For instance, sharing the results of usability testing on a design proposal or showing how the proposal worked in a similar situation can convince other participants.

> *what we do is look to what other projects and other web standards exist along the problem, say the problem of where to locate the help link, on the admin bar, we look at a bunch of different web applications, Facebook, Google docs, also desktop applications things like that, basically starts to conform a consensus around that ok it got to be on the right hand side of the menu and so we tend to go with that ...[DevD6]*

Another technique for promoting consensus was presenting screenshots or writing a patch for a design proposal (n=5). As DevD2 said participants in a discussion are more likely to comment on a proposal that has a patch attachment:

> *I present arguments in favor of it and then post a patch. People are typically more inclined to go with a solution that has a patch than another solution that does not have a patch, unless they have a major reason for liking the other solution better.*

Participants also noted endorsing experienced members of the community in the discussion (n=3), writing a summary of the discussion (n=3), communicating via synchronized channels (n=3), having an administrator make the final decision (n=5), spending time to understand others' perspectives (n=3), voting for different design proposals (n=2), and advertising a stalled discussion (n=2) as techniques for promoting consensus. However, it is unclear how effective these techniques are given that 42% of the UI design discussions we analyzed did not reach consensus. All of the techniques noted by participants for promoting consensus are listed in Table 4.1.

4.2 Consensus Building Model

Enabled by open access to peer production communities such as Wikipedia and OSS, researchers have begun to study elements of group decision making

in real world data sets. For example, Lam et al. studied how group size, experience, and group formation influences decision quality in Wikipedia and found that larger groups make better decisions [51]. Similarly, Burke et al found that extensive and diverse contributions in Wikipedia can predict promotion decisions [17]. Analogous to these studies, the goal of this section is to test how different factors relate to reaching consensus on UI design issues in a peer production community. In particular this section is centered on answering the following research questions:

R1. What factors affect the consensus building process in distributed UI design discussions?

R2. How well can comments be classified to consensus and non-consensus comments?

4.2.1 Methodology

To test how different discussion elements relate to consensus, I analyzed a large corpus of interaction data. The interaction data was extracted from the discussion threads (discussions) in the issue management system of Drupal, an open source content management system initiated in 2001. Drupal is a mature community with an established workflow and social organization. At the time of data collection, for example, the software was being used in about 490,000 websites to manage content and about 440,000 people had registered to contribute to the project.

Changes to the user interface and system software of Drupal are requested, discussed, and implemented (or not) through its issue management system (see Figure 4.1). Any community member can create an issue in the issue management system describing a design problem or feature request, which establishes a separate discussion. Others may participate in the discussion by proposing design alternatives, critiquing the alternatives, implementing an alternative (writing a patch), reviewing a patch, clarifying the problem, or offering other insights. To indicate the current progress of a discussion, participants can set its status to *'active', 'needs work', 'needs review', 'reviewed and tested by the community', 'fixed'*, or *'closed'*.

There are four categories of discussions in the issue management system: bug reports, feature requests, tasks, and support requests. According to dru-

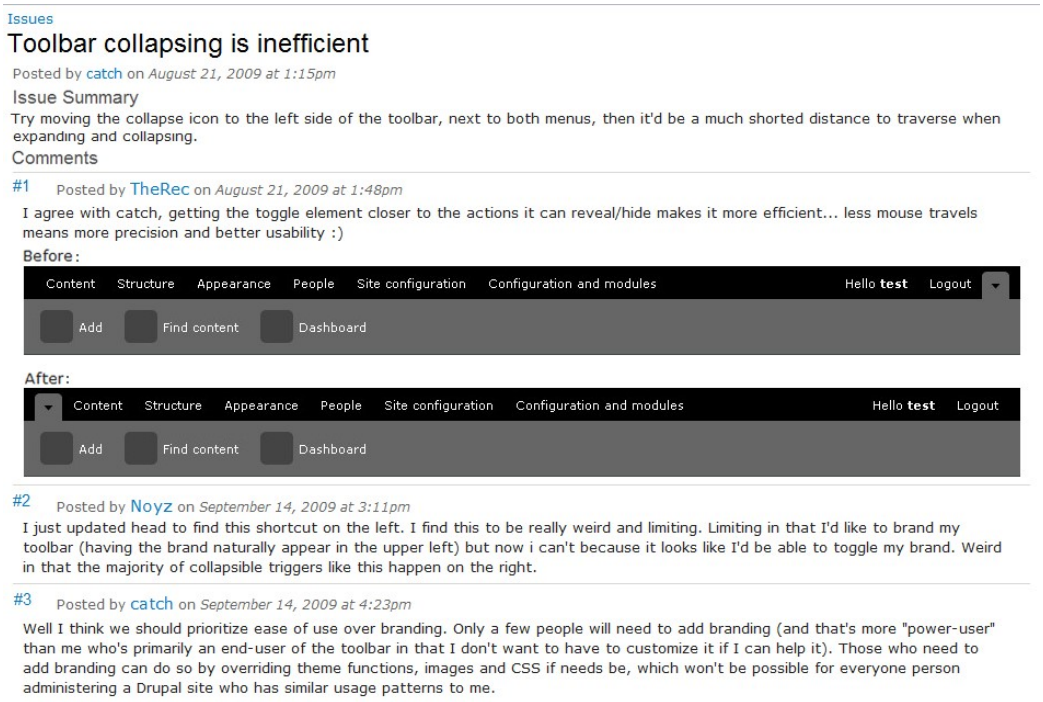

Figure 4.1: A UI design discussion occurring in the issue management system of Drupal. Participants are proposing and debating ideas for where to locate a shortcut on the toolbar. The image was condensed for the purpose of explanation.

pal.org, bug reports aim to resolve functionality and usability problems while feature requests are for adding new functionality. Tasks are non-functional things that *'need to be done'* while support requests are for technical support. I only examined bug reports and feature requests as they contained the majority of the UI design discussions I wanted to study.

There were 285,008 discussions tagged as bug reports and feature requests in the issue management system at the time of data collection. This set was filtered to include only the discussions tagged with "Usability" or "d7ux" (usability in Drupal 7), which left 577 UI design discussions. These discussions occurred between March 2004 and September 2011. The usability issues ranged from significant redesigns to design details. For instance, an issue titled "Initial D7UX admin overlay" aimed to revamp the interaction design of admin pages in Drupal by providing themed admin pages as an overlay on top of the actual website while another issue only requested changing the location of a shortcut on a toolbar (see Figure 4.1).

I used the status to categorize discussions as consensus, non-consensus, or

Characteristic	Type	Mean	SD
Discussion duration (weeks)	C	41.76	50.71
	NC	101.70	83.17
Number of comments	C	48.34	61.02
	NC	33.21	36.95
Number of participants	C	12.82	10.42
	NC	11.82	8.77

Table 4.2: Summary statistics for the consensus (N=200) and non-consensus (N=141) UI design discussions. 'C' indicates consensus and NC' indicates non-consensus discussions

ongoing (unclear if consensus has been reached). I considered discussions marked as closed as having reached consensus. This typically means there was collective support for a decision such as implementing a specific proposal or concluding the issue was unnecessary or not a problem after all. This status is a reasonable proxy for consensus because if any participant strongly objected to the proposal, s/he could have reverted the status (e.g. back to 'active') and caused the discussion to continue.

Differentiating non-consensus and ongoing discussions was more challenging. I calculated the idle duration, the time from when the last comment was posted to the time of data collection and considered a discussion to be non-consensus if it's idle duration was more than 90% of the idle durations between comments in the consensus discussions. The remaining discussions were considered ongoing. This categorization yielded 284 consensus and 241 non-consensus discussions. The 52 ongoing discussions were discarded.

Finally, I filtered this data set to include only those discussions that were non-trivial. By reading a large fraction of the discussions and experimenting with different thresholds, I found that a threshold of having at least seven comments filtered almost all of the non-trivial discussions. After this filtering, I had 200 consensus threads and 141 non-consensus threads for the analysis. Table 4.2 reports summary statistics for the consensus and non-consensus UI design discussions after this filtering.

4.2.2 Data Analysis

Based on the interview results, consensus building literature [36, 16], and prior analyses of online communities [1, 2], I identified 23 metrics that may relate to consensus building. Table 4.3 lists these metrics grouped into three categories: content, process, and user relationships. Though not exhaustive, these metrics provide a useful starting point for understanding which factors affect consensus building in UI design discussions.

For example, for Content metrics, I counted the number of messages with screenshots attached as a proxy for the number of design alternatives proposed. More alternatives may create more opportunities for consensus. The number of question marks was counted as a proxy for attempts at building shared understanding [30]. From the interviews, I found that synchronous chats can promote consensus and therefore included how often "IRC" was mentioned in a discussion. Similarly, occurrences of "usability testing", "code review", and "summary" were counted in each discussion. The number of non-Drupal links was included to capture use of external evidence in the design arguments.

For the Process metrics, I counted patches, comments, and contributors as a proxy for the level of activity in each discussion. A discussion with increased activity may have a better chance of reaching consensus. The duration of a discussion was also included as allocating more time to a discussion may indicate stronger commitment to identifying an agreeable solution.

For User Relationships, I calculated the number of triads contributing to a discussion to estimate prior interaction history [72]. Triads were determined from the social graph created from the users, discussions, and relationships [72, 74, 18, 46]. In a social graph, the nodes represent users and discussions while edges represent their relationships. An edge between a user and discussion is established when a user contributes to that discussion. An edge between two users is established when one user responds to the other. An edge is weighted based on the length of the comment. I also computed a page rank score [61] for each participant to estimate influence' within the community. The page rank score was also calculated from the social graph. The duration of community participation was used to estimate the experience of participants as interviewees felt having more experienced members of the community participate in a design discussion promotes consensus.

Category		Metrics
Content	Occurrence of keywords	(1)# of "usability testing"s
		(2) # of "summarys" ∗
		(3) # of "code review"s ∗
	Content Qualities	(4) Avg. # of words ∗
		(5) Total # of words
		(6) Priority of the issue
		(7) # of non-Drupal links
Process	Activity	(8) # of comments ∗
		(9) # of patches ∗
		(10) # of authors/participants ∗
		(11) # of screenshots ∗
		(12) # of comments by creator ∗
	Duration	(13) Duration of the thread ∗
		(14) Avg. duration b/t comments
	Quality	(15) # of "IRC"s ∗
		(16) # of questions marks ∗
User Relationship	Social Graph	(17) # of triads in graph
		(18) Avg. page rank score of participants
		(19) # of alternate replies ∗
	Prior Contributions	(20) Avg. # of total participation duration
		(21) Avg. # of participants' prev. comment ∗
		(22) # of participation weeks of creator ∗
		(23) # of creator's prev. comments ∗

Table 4.3: Categories of metrics affecting consensus in distributed UI design discussions. In searching for the keywords I considered all common alternatives for the keywords (e.g. usability test, Usability Testing, User testing, etc.). The metrics marked with (∗) were later removed from the analysis to avoid problems with collinearity.

	B	Df	Sig.	Exp(B)
Average # of participation weeks	.01	1	.00	1.01
# of triads in social graph	.10	1	.03	1.10
IRC	.24	1	.05	1.26
Constant	-1.06	1	.02	.35

Table 4.4: Results of the binary logistic regression. The Hosemer-Lemoshow test confirmed the validity of our regression model (Chi-square = 7.91, p = 0.44).

To calculate values for these metrics, I incorporated information from the discussion content (e.g. length of comments), metadata of the discussions (e.g. duration of the thread), and contributor's Drupal profile.

Logistic Regression

To investigate how these metrics relate to consensus, I performed a binary logistic regression. Binary logistic regression is a type of regression used to model the relationship between independent variables and a binary response variable. For this analysis, the metrics from Table 3 served as the independent variables and were computed for each discussion in the data set while the dependent variable was whether the discussion reached consensus. To avoid problems with collinearity in the regression analysis, I removed fourteen variables that demonstrated strong correlations ($r>0.4$). These variables are marked in Table 4.4.

I performed binary logistic regression as implemented in SPSS and used step-down regression to identify a partial model. I first entered all variables and removed each variable that did not show significance and repeated until a set of variables was reached that were all significant. Three of the nine metrics included in the analysis showed significance ($p<0.05$): average number of participation weeks, number of triads, and mentions of IRC. Table 3 summarizes the results. To assess the goodness of fit of the model, I performed the Hosmer-Lemoshow test (Chi-square=7.91.56, p=0.44). In this test, the model is valid if the p-value is greater than 0.05.

4.2.3 Results

To aid interpretation of the results, I conducted five follow-up interviews as described in the "User Perspectives" section. These interviews followed the original script, but probed further about the factors found to be significant. In addition, I analyzed thirty of the discussions that reached consensus in our data set. The discussions were sorted based on the three factors found to be significant and ten threads from the top of each of these three lists were reviewed.

Experience with Drupal

The regression analysis showed that having people in a discussion who have participated in Drupal longer promotes consensus. Research studies confirm that including experienced people can positively influence group decision making performance [44]. My interview results and review of the discussion threads illustrated how experience can facilitate consensus building. First, I learned that members who have been in the community for a long time facilitate consensus by helping other members, especially new ones, understand the norms of communication and the process of participation in the community.

> *...what's important for reaching consensus is having common ground rules or communication and process and you know working those out and to that extent more experienced in Drupal community might help people be better at reaching consensus, because they'll understand that you don't say things this way or you do, or these are the options for contacting somebody if you have a problem or that kind of thing...[DD4]*

Second, it was reported that experienced members are more skilled in unblocking a discussion. For example, comments and opinions posted by experienced members are valued more than those posted by other participants.

> *...there are those people in the community that are recognized people who have been in the community a long time, or who are respected because they have written a lot of code, or they have written a lot of patches, or they are the maintainer of a certain*

bit of code and when those people chime in, it tends to hold a little bit more weight when someone unknown chimes in...[DD1]

As a result, experienced members can help direct the discussion toward a specific design proposal. For instance, in a discussion about adding edit and delete operations to a page in Drupal, when two of the participants (X and Y) proposed different solutions and were not able to come to an agreement, a community member with design experience was invited to review and decide between the proposals:

I am inclined to agree with X here, following the logic of menus and taxonomies this should make more sense...

Finally, experienced members can promote consensus by understanding the need for proposing solutions that accommodate competing alternatives. Satisfying opposing views allows stalled discussions to move forward. For instance, in a discussion about placement of a shortcut for collapsing the Drupal toolbar, X thinks that the icon for the shortcut should be placed on the left side of the menu to prevent accidental clicks on the "logout" icon while Y thinks it should remain on the right side because the space on the left is needed for branding. They cannot come to an agreement until Z who has been in the community for six years comes in and proposes a new solution:

Thought: Move /help over to the right of "log out", move the shortcut collapsing back to the right, then you'd at least accidentally click a "safe" link.

Prior Interactions

My analysis showed that having more triads participate in a discussion increases the likelihood of consensus. Triads represent three people who have previously interacted and produce closed social structures that promote trust [72].

Interview results and review of the discussions confirm trust as an important factor in consensus building. First, I found that participants are more likely to read, learn from, and evaluate comments posted by members whom they trust. This exchange of knowledge can create mutual understanding and consequently promotes consensus.

I think I'm less likely to dismiss something if it's from somebody I know and I respect. It's a little more likely to read carefully what they say and believe that they have something meaningful to say...[DevD4]

This finding reflects findings in other research studies that indicate a high degree of trust existing within dense parts of a social network facilitates the exchange of complex knowledge [80]. Second, prior interaction and increased trust promotes agreement among participants.

... it's sort of like a trust matrix type of thing, because if I don't know you and you are suggesting this thing that sounds like a bad idea to me, I probably fight against it, but if you are proposing something and I don't know you but three other people that I do know are saying yeah actually that's a great idea and this is why, then I'll be far more likely to be like alright let's go with it then.[DevD6]

Finally, I found that trust in other participants' technical abilities can save time in the process. For instance, knowing that the person who wrote a patch usually conforms to coding standards can accelerate code review.

... if people know who somebody else is, it saves a heck of a lot of time, at all levels, like, for example, if I know the person who wrote the patch and I know that traditionally they write pretty good patches that conform to coding standard and stuff like that and then I see the person that reviewed it is the person I associate with being the smart person about that thing and the person who marked it as reviewed and tested by community [...], and that person was also someone I recognize as if they say something is RTBC it's actually good to go. Then it saves all kinds of time.[DevD6]

Similarly, recognizing the person who wrote the patch as a skillful programmer can accelerate implementation. For instance, in a discussion where the proposal was to add an edit link to all Drupal pages, one member (Y) was able to build upon another's (X's) patch and save time. Y says:

So, yeeha, X's last changes contained some really good ones that allowed me to proceed further.

Use of Synchronized Communication Media

Based on the regression analysis, threads containing more mentions of "IRC" are more likely to reach consensus. A group of two to five people usually participate in the synchronized discussions and are expected to report their conclusions back to the corresponding discussion for the benefit of all. Failure to report may cause the other participants to loose context.

> *The danger in IRC becomes when and this happens sometimes when there are huge discussions that go on IRC, big community impacting discussions and only the people who happen to be on IRC at that time, know about them and if those don't make their way back to the issue queue or groups or some other mean of more permanent storage that's really dangerous because a lot of people loose context in these discussions that way...[DevD6]*

From the interviews and careful inspection of discussions, I identified three ways that IRC can help build consensus. First, I found having discussions in IRC can accelerate agreement between opposing viewpoints.

> *... it [IRC] can help if there is one or two people who are disagreeing about something, if those people go to IRC they can chat it out much faster than the issue queue.[DD4]*

Second, in IRC people can come up with an initial design proposal for solving the usability problem that may not be possible in context of the larger discussion.

> *IRC is great for say a small group of people going off and coming up with an initial proposal that they all agree on and then proposing that to the community.[DevD6]*

Reporting this proposal back to the discussion advances consensus building because participants can argue for or against the proposal as opposed to developing their own. As DevD6 said:

> *...then it becomes let's argue against this position as opposed to try to come to a position to argue against...*

Finally, I learned that participants use synchronized communication to hasten collaborative design review, programming, and debugging sessions.

This finding corroborates observations reported in [26]. For example, when discussing the design and implementation of an overlay for the Drupal interface, one of the developers (Y) asked another developer (X) to join him in a synchronized chat for a collaborative debugging session:

> *X, some of your files are being cut off, such as overlay-parent.css(?) Please come onto IRC so we can help you debug.*

4.3 Discussion

The regression analysis showed that three of the factors tested are predictive of consensus in a UI design discussion: the experience of participants, number of triads, and mentions of synchronous communication. Interestingly, none of the content metrics were significant. One interpretation of this result is that who participates in a UI design discussion is more important than how many design alternatives are proposed or what arguments are made for the purpose of building consensus. For example, this may be due to not having a facilitator in the discussion skilled at steering the group toward consensus [20]. Participation of experienced members may therefore compensate for the absence of trained facilitators, i.e., they have a better understanding of how to guide the discussion toward consensus. Another possibility is that the content metrics used in the analysis were incomplete. Future work should therefore examine additional metrics such as the use of different argument types [53] and rhetorical devices [45] to further test how content attributes may relate to consensus.

A number of factors perceived by the interviewees to relate to consensus did not show significance in the regression analysis. For instance, interviewees mentioned contributing concrete evidence to an ongoing discussion such as usability tests of the design proposals and external links to interface examples positively affect consensus building.

One reason these factors did not correlate with consensus is that they were seldom performed. For example, in the data set, mentions of usability appeared in only 0.06% of the consensus discussions and in 9% of the non-consensus discussions. One way to foster the inclusion of concrete evidence is to establish specific community guidelines for discussing UI design issues. Another method would be to configure a testing platform where participants

can easily try a patch and provide feedback in the discussion without having to worry about applying the patch to their locally installed version of the product.

A second possible reason some of the factors did not show significance in the analysis is that I did not consider their context. For example, in the data set, the number of links to external sources was similar in consensus ($\mu = 5.5, \sigma = 7.5$) and non-consensus ($\mu = 4.9, \sigma = 6.9$) discussions. This may be due to not considering the helpfulness of the link targets. For example, future work may consider weighing the link counts based on the helpfulness of the link targets to the discussion (e.g. did the links reference sketches of design proposals, interaction examples within well-known Web sites, or community design standards).

4.4 Design Implications

This work has several design implications for discussion interfaces w.r.t. promoting consensus. One implication is to enable discussion participants to quickly identify others with whom they have had prior interactions. These community members could then be invited to join the discussion, thereby increasing the number of triads. For example, for each discussion, the community software could maintain a list of members whose participation would form triads by analyzing the social graph [41] or history of participants' contributions [22]. Options could be offered for filtering the list, e.g., requiring a minimum number of prior interactions or specifying that only the interactions within specific types of design discussions be considered.

A related implication is to allow discussion participants to identify experienced members who may be willing to join the discussion. Inviting appropriate people to join a discussion may not only aid consensus building, but may also assist community members in identifying discussions of interest. For example, analogous to [22], the system could recommend experienced members appropriate for the discussion by considering the duration of their community membership, interest profiles, and recent activity within the community (to prevent core members from receiving too many invitations). As before, options could be provided for modifying these search parameters. This analysis showed that participants value the comments contributed by experienced

members or members with whom they have had prior interaction. The discussion interface could therefore allow participants to filter comments within the current discussion contributed by others meeting these criteria or by including appropriate visual cues for these criteria within the comments.

Results of the interviews and inspection of discussions revealed that certain types of comments aid consensus building more than others. For example, comments that strongly argue for or against design alternatives can build agreement, comments that summarize the discussion to date can help participants make sense of the thread, and comments that report the conclusions from synchronized discussions can help participants maintain context. The discussion interface could therefore employ color codes or other visual cues to highlight these types of comments [55, 76]. To classify comments, the author or other participants could be allowed to assign pre-defined community tags. To reduce or eliminate the costs of tagging, an alternative would be to automatically infer the comment types, which could be modified by participants to correct any errors.

To further aid the consensus building process, recent key contributions to the discussion could also be highlighted. For instance, comments that include key contributions such as the most recent design proposal, implementing or reviewing a recent patch, or changing the status of the discussion could be highlighted. It is important to note that not all of the filtering, searching, and highlighting mechanisms described need to be included in a discussion at the same time. For example, end users could configure which of the features are applied in their local interface.

Consensus building is a critical component of UI design discussions in OSS as it promotes a better product and a stronger community. The work presented in this chapter had three main contributions. One contribution was reporting user perspectives on the challenges of reaching consensus in UI design discussions, the techniques utilized for addressing the challenges, and the consequences of not reaching consensus. A second contribution was analyzing how various metrics related to the content, process, and user relationships of the discussions correlate with reaching consensus. The main result from this analysis shows that discussions having participants with more experience and prior interaction history are more likely to reach consensus. Finally, I offered design implications for promoting consensus in distributed discussions of UI design issues. In the next chapter, I use these design implication along with

other consensus building strategies to design a new interface for distributed discussion that better supports consensus building.

Chapter 5

PROCID: Bringing Consensus Theory to Distributed Design Discussions

In this section I will describe Procid, a novel browser extension that enables consensus building strategies to be realized in un-moderated distributed design discussions. Key features of the tool include (i) the ability to visually track idea proposals and organize the discussion around ideas; (ii) the ability to define persistent criteria and rate the ideas against the criteria; (iii) the ability to register strong support for or against ideas, empowering individuals with a stronger voice in the discussion; (iv) interaction constraints that promote best practices of consensus efforts, e.g., encouraging the first comment on an idea to be supportive of it; and (v) the ability to identify candidates to invite to active discussions using attributes important for consensus in the domain.

I have designed Procid based on a the design implications learned from Chapter 4 and a subset of procedural, rhetorical, and social consensus building strategies gathered from the theory and practice of consensus building.

5.1 Strategies for Consensus Building

Developing a system that embodies consensus building within distributed design discussions is a major challenge. To address this challenge, I enumerated many consensus building strategies recommended for moderated face-to-face discussions [7, 45, 47, 64, 68, 81]. See Table 5.1. From this set, I prioritized strategies that would be feasible to implement in software, would be (possibly) difficult to realize without tool support, and would form a coherent user experience. This led me to focus on six strategies:

S1: Promote perception of valued contribution and sense of empowerment. All participants should feel welcome to contribute to a discussion, that their comments are valued, and that they have the power to affect the decision outcome. Enhancing these perceptions promotes understanding of the deci-

sion and increased commitment to its implementation [7]. Existing discussion interfaces cannot directly support this strategy because they process and represent all comments the same, regardless of who wrote them or if the content conveys strong support for or objection to an idea.

S2: Express concerns in a constructive manner. A discussion needs to be constructive, even when there are strong disagreements [68]. Similarly, when ideas are proposed, the strengths of the ideas should be discussed first and then the weaknesses. In absence of a facilitator, however, distributed discussions can be plagued by destructive comments and flame wars, causing participants to leave the discussion, or discouraging participation.

S3: Build on prior relationships and seek expert advice. Some participants can promote consensus more than others. For instance, experienced and/or trusted participants can promote consensus by helping opposing parties develop options or by identifying barriers to effective negotiation [68, 81]. However, existing discussion platforms do not provide mechanisms for including potential participants based on their ability to promote consensus in a discussion.

S4: Evaluate and decide according to established criteria. Solution alternatives need to be evaluated to find one that best satisfies all stakeholders interests [68]. The criteria help participants weigh trade-offs and identify the solution that best matches the majoritys interest. But in existing discussion forums, criteria must be proposed as part of the text in a comment, thereby becoming fragmented across the discussion. Also, there is no explicit means for evaluating the idea proposals against the set of criteria defined.

S5: Maintain a visual summary of key points of agreement and disagreement. In moderated face-to-face discussions, a recorder tracks the proposed options and organizes the key points of agreement and disagreement [68], which can be shared during the meeting. This helps the group see the direction of the discussion and maintain task focus [7]. In distributed discussions, the participants carry the burden of tracking ideas and related arguments, making it easy to lose them in the discussion.

S6: Mark and revisit influential points of the discussion. Distributed discussions often contain many comments posted over a wide window of time. It is therefore easy for important discussion points to become lost or forgotten. This is especially problematic for participants who join late, or who enter and leave throughout and only read subsets of comments as time allows. In cur-

rent discussion platforms, there is no easy way to mark and revisit important discussion points and share these with others.

As shown in Table 5.1, I selected only a subset of the known strategies for our design. These strategies provide a minimal but reasonable starting point for exploring how to design a system that brings consensus theory to the practice of un-moderated online design discussions. As experience is gained, additional strategies could be implemented and/or made accessible to community members via onboarding or other newcomer socialization techniques [19].

Figure 5.1: Procid is a browser script that extends the existing discussion interface in Drupal with a navigation panel on the top and a lens panel on the left. (a) The three icons on the top panel enable navigation between the main discussion page, the idea-centric view, and the invite page. (b) Five filters are available at the top of the lens panel that toggle the highlighting of comments with the corresponding property, e.g., the must read (leftmost) lens highlights the comments endorsed as must read by participants.

5.2 PROCID

Procid is a browser extension that supports consensus building strategies for distributed design discussions. See (Figure 5.1). I built Procid as an add-on to an existing discussion platform, rather than as a stand-alone system. The benefits are that I can deploy and test the interface in an existing community, recruit participants who care about the discussion topic, and integrate an existing workflow. It also allows me to leverage the community's social

Social Strategies	*Promote perception of valued contribution (S1).*
	Build on prior relationships and seek expert opinions (S3).
	Promote community ownership of ideas.
	Help new members become part of the group.
Rhetorical Strategies	*Express concerns in a constructive manner (S2).*
	Value feelings and show empathy towards strong emotions.
	Avoid answering to all the objections to your viewpoint.
	Provide descriptive, specific, and tentative feedback.
	Disagree with ideas, not people.
	Keep the discussion on topic.
	Listen for the aspects of an idea that you find attractive and acknowledge the positive when responding.
Procedural Strategies	*Evaluate and decide according to established criteria (S4).*
	Maintain a visual summary of key points of agreement and disagreement (S5).
	Mark and revisit influential points of the discussion (S6).
	Respond to disruptive behavior.
	Generate a wide variety of proposals.
	Try to equalize power and balance participation.
	Blend ideas together and try to maximize joint gains.
	Dont quit after the first good idea.
	When all viewpoints have been expressed, estate the conclusion toward which the group appears to be moving.
	Test for the agreement.
	Before proposing solutions, understand the problem.

Table 5.1: Consensus building strategies derived from prior studies of moderated face-to-face discussions. The strategies were grouped into three categories, but are not intended to be exclusive. The grouping did not affect our design and was done merely for the purpose of presentation. Highlighted strategies are the focus of our design.

structure to build novel features (e.g. participant invitations).

To provide a testbed, I built Procid to augment design discussions in Drupal, an open source content management system initiated in 2001. Drupal is a mature open source community with an established workflow and organization. Drupal's issue management system includes thousands of discussions requesting usability fixes, improved designs or new features for the Drupal product. Members participate in the discussions by proposing design alternatives, critiquing the alternatives, implementing an alternative (writing a patch), reviewing a patch, or offering other insights.

As the Drupal product expands and grows a large user base, adding features or fixing usability issues is getting harder. For example, proposed changes must now often consider customer preferences, product roadmaps, and consistency issues; along with the (strong) opinions of current members. By interacting with the Drupal community for the past four years, I have become aware of the need for employing consensus building strategies in the community. This interaction has included participating in and analyzing online discussions in the community, discussing consensus building with members through interviews and existing threads, and presenting at a regional Drupal conference.

Procid organizes its features into three groups: an *idea-centric view* that summarizes the proposed ideas, criteria, and comments; an *invite page* that lists potential members to invite to a discussion, and a *lens panel* that wraps the existing discussion and enables filtering of the comments.

5.2.1 Idea-Centric View: Organize the Discussion around Ideas

Proposed ideas are the focal point of any design discussion. Keeping track of the ideas enables participants to evaluate or build upon each others ideas and revisit ideas not yet discussed. For tracking ideas and visualizing agreement and tension, Procid offers an idea-centric view of a discussion. This type of view prevents ideas from becoming lost in a discussion [82]. This view summarizes the proposed ideas, their status (Dropped, Ongoing, or Implemented), and the criteria for evaluating the ideas. It also spatially organizes the comments for each idea into supportive, neutral, or constructive categories (Figure 5.2). Before detailing the features of this view, I will explain

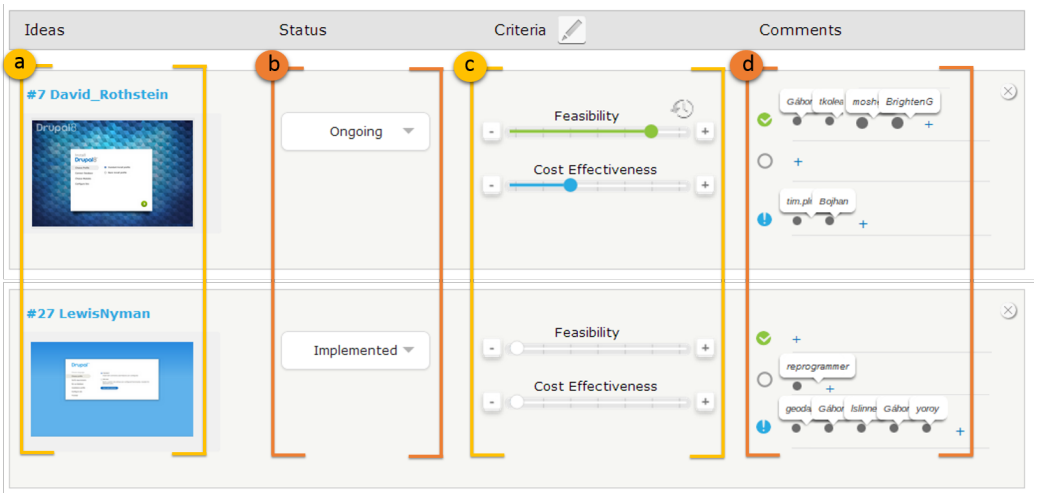

Figure 5.2: Selecting the light bulb icon in Figure 5.1 opens the idea-centric view which contains four columns of information. (a) In this view ideas are represented with the image that was attached to the corresponding comment or the initial text if no image was attached. (b) A menu for setting the status of the idea and (c) a list of user-defined criteria. Each criterion has a shared slider for rating the idea which any user in the discussion can manipulate. (d) The comments referring to an idea are spatially organized into three rows: supportive (top row), neutral (middle) or constructive (bottom). Comments registered as strong support or objections are rendered with larger circles, which are intended to attract more attention than the other comments.

how the system knows which comments contain ideas or refer to ideas.

Identifying Ideas and participants opinions about ideas

Procid inserts an interaction panel within the existing dialog for authoring a comment. In this panel, a participant can toggle if the comment proposes an idea or if it refers to an existing idea. For the latter, the participant can also indicate her disposition toward the idea (Figure 5.3). Whether a comment contains an 'idea' can be later changed by selecting the icon inserted by our tool in the bottom right of the rendered view of the comment

Asking for user input to render alternative views of a discussion follows a long thread of CSCW research (e.g. [21, 55, 62]). A unique aspect of my approach, however, is that I am injecting the request into an existing community interface. This imposes a constraint on how much input I can realistically request for fear of community backlash. It also prevents me from expanding the input to the point of making the system undesirable to use

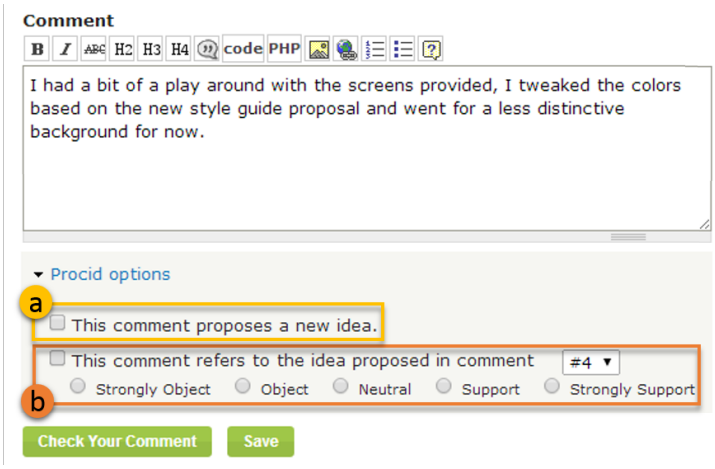

Figure 5.3: When authoring a comment, Procid allows the user to mark if the comment (a) proposes an idea or (b) refers to an existing idea and, if so, to indicate their disposition toward it.

(an important lesson learned from the class of design rationale systems).

Procid therefore only requires this minimal input, but in return can render the view shown in Figure 5.2. From this view, users can quickly determine which ideas are favored or opposed, which need more attention, patterns in the comments, and how the ideas compare based on the criteria

Establishing concrete criteria to evaluate ideas

In design discussions, different solution alternatives need to be evaluated in order to find the alternative that best satisfies all participants interests. In the idea-centric view, participants can define the criteria for evaluating ideas. For example, if a solution must have *low implementation cost*, a participant can add *cost effectiveness* as a criterion. This is achieved using the criteria editor, accessed by selecting the pencil icon (Figure 5.4).

For each newly added criterion, a slider is generated and replicated in the row of each idea (Figure 5.2c). Using the shared slider participants can specify how much each idea satisfies each criterion on a seven-point scale (from unsatisfactory to satisfactory). Upon changing the rating, the participant is prompted to enter rationale for the change which will be posted in the discussion (Figure 5.5a). Requiring justification is meant to reduce superficial back-and-forth with the ratings. The scales allow participants to visualize, compare, and discuss trade-offs of ideas and eventually determine the one that best satisfies the criteria.

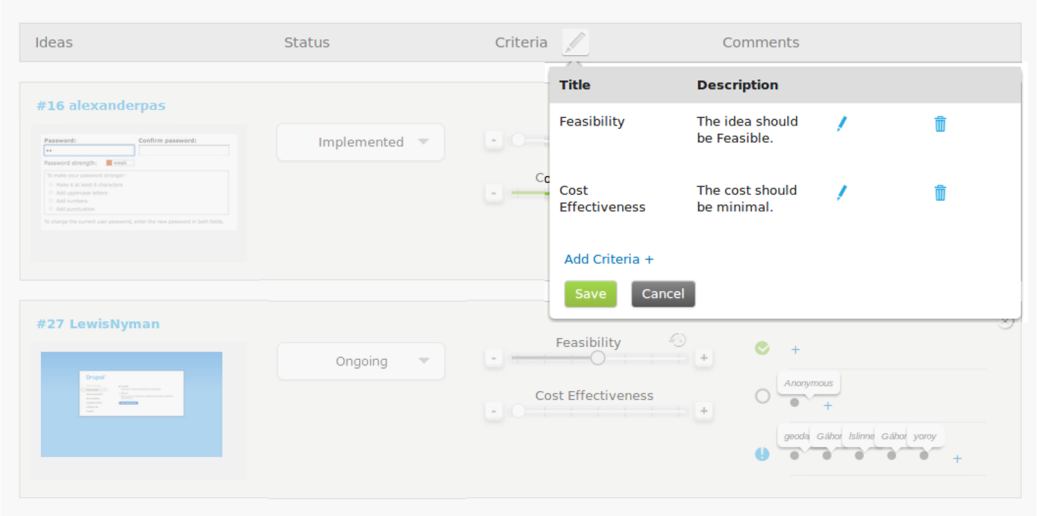

Figure 5.4: Users can establish new criteria for evaluating the ideas (Add Criteria +) and edit or delete the existing criteria.

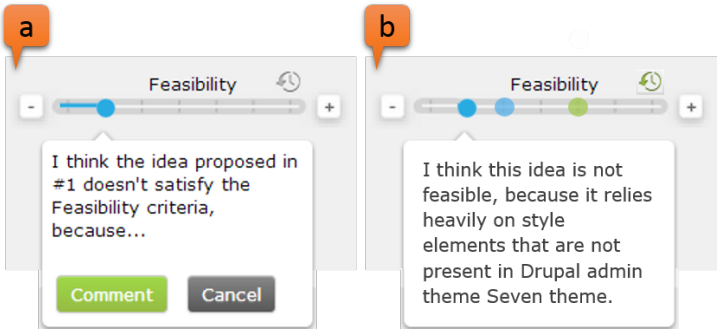

Figure 5.5: (a) Users can specify how much each idea satisfies each user-defined criterion. (b) Users can click on the history button to see previous ratings and their associated comments.

The criteria scale shows only the most recent rating, but the tool maintains the prior ratings and associated comments. This history can be accessed by selecting the button next to each scale (Figure 5.5b) and is intended to prevent users from feeling their opinions have been supplanted.

Spatially organizing comments around ideas

The idea-centric view organizes the comments referring to an idea into supportive, neutral, or constructive categories. The category of a comment is determined by the disposition rating provided when it was authored (Figure 5.3b). A participant can also add a comment from this view by selecting the "+" button corresponding to the desired disposition and idea. The ad-

vantage is that this opens the comment authoring dialog with the input parameters already set. When an idea is first posted, the "+" button for constructive (critical) comments is disabled until a neutral or supportive comment is added. The purpose of this constraint is to encourage discussing the strengths of an idea before discussing its weaknesses [43].

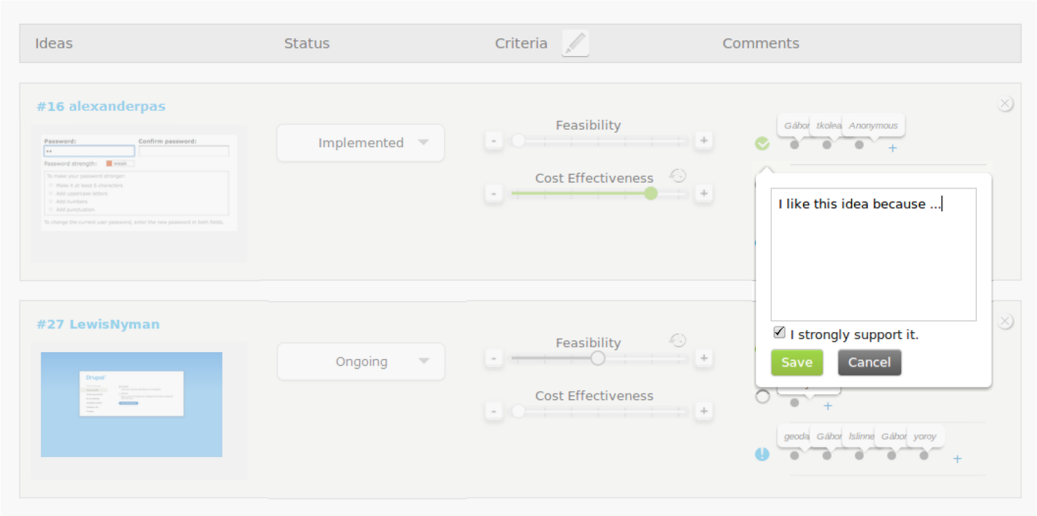

Figure 5.6: A user may register strong support when posting a supportive comment for an idea.

When a user is adding a supportive comment for an idea, s/he can register it as strong support for the idea (Figure 5.6). Alternatively, when users are adding a constructive comment, they can register it as a strong objection to the idea. Comments registered as objections or strong support are rendered with larger circles to indicate unusually strong opinions about an idea (Figure 5.7). The purpose is to provide participants with a stronger sense of empowerment over the proposed ideas and decision process and make these opinions easily noticeable in the visualization.

If an idea receives several objections, the person who proposed the idea can set its status to "Dropped". This status informs others that the idea need no longer be considered and the ability to add comments is disabled.

5.2.2 Invite Page: Seeking Potential Contributors

A discussion can become stalled due to a controversy over implementation or strong disagreements between parties. Seeking additional expert advice or inviting trusted colleagues to join the discussion can help it move forward.

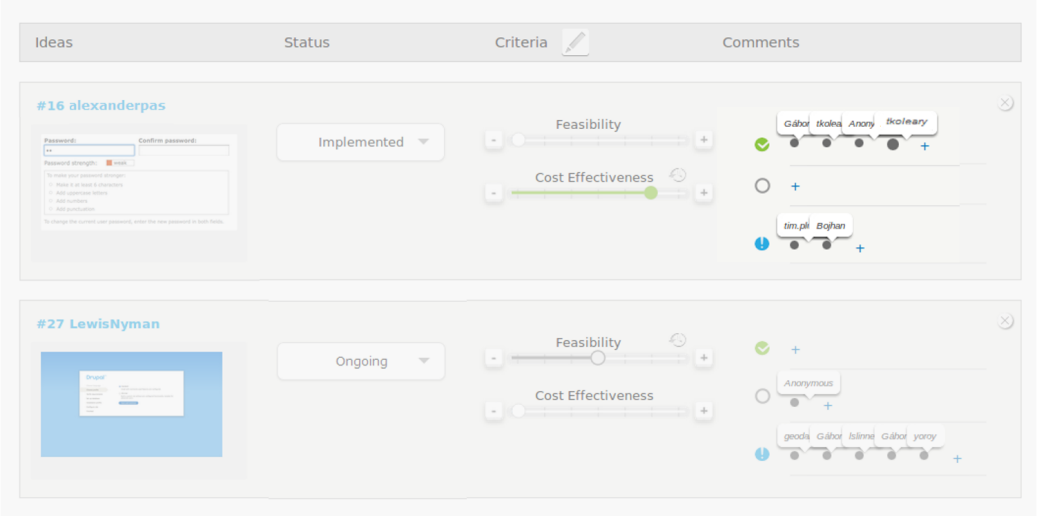

Figure 5.7: Comments registered as strong support or objections are displayed with larger circles, which attract attention when the page is viewed.

However, there is no convenient mechanism for finding people to include in the discussion who could contribute to the consensus effort.

To explore such a mechanism, Procid's Invite page dynamically creates a list of community members sortable by attributes important for consensus in the domain (Figure 5.8). The attributes include the duration of membership, number of patches submitted, recent participation, prior interaction with participants in the current discussion, and number of prior discussions involved in that reached consensus (i.e. the discussions were closed). For example, experienced members may bring historical perspectives on the product and past decisions that can influence the direction of the discussion and those with connections to participants in the current discussion may be more likely to accept an invitation to participate. The set of attributes could be expanded in future work, such as including a relevance score and the number of active discussions to gauge workloads prior to invitation.

Discussion participants can sort the list to seek potential members to invite, and not have to guess, rely on memory, or blindly search for possible contributors. The list can be especially helpful in situations where the user is less familiar with the topic or current participants. To compute these attributes, Procid analyzes the social graph [41], interaction history, and profiles of community members as an offline process and repeats it periodically.

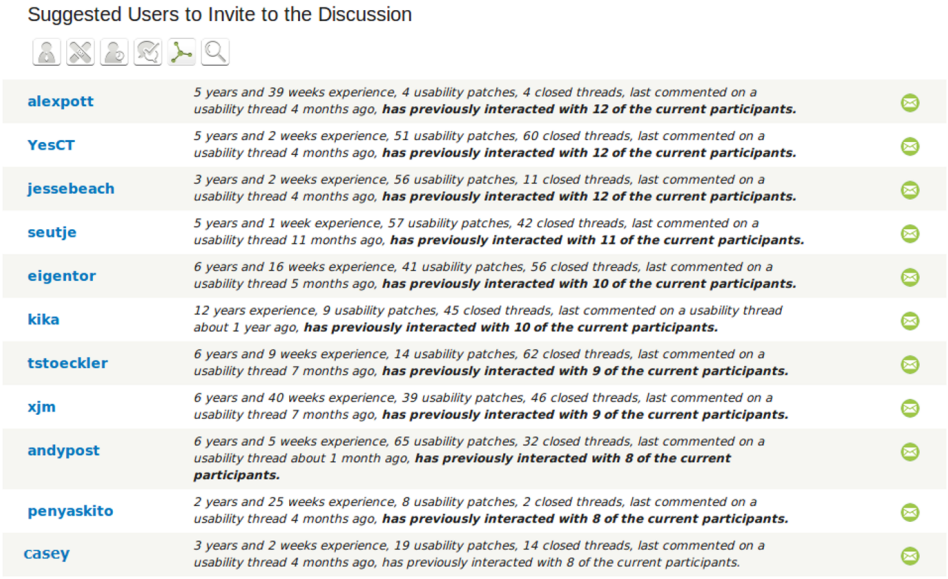

Figure 5.8: The invite page generates a list of users to potentially invite to the discussion. It considers criteria important for consensus such as the number of patches contributed and prior interaction history with participants in the discussion.

5.2.3 Lens Panel: Gaining a Rapid Overview of Discussion

Procid extracts significant discussion bits from multiple perspectives including community endorsement, the content, and the conversation behavior of the posters. These perspectives are captured in five lenses available in the lens panel wrapped around the main discussion page: *must read, idea, conversation, experienced,* and *patch* (Figure 5.1).

Through these lenses, Procid allows new comers to gain rapid awareness and existing participants to mark and revisit key points of the discussion (Figure 5.10). For example the must read lens highlights comments that are critical for understanding the discussions trajectory. Examples include comments that summarize the discussion to date, plot the direction of the discussion, or report conclusions from synchronized discussions. Any participant can endorse a comment as must read by toggling an icon that our tool inserts with the comment (Figure 5.9). It can be toggled off if a participant feels the comment was incorrectly marked or is no longer deserving of this property by selecting the icon inserted in the bottom right of the rendered

Figure 5.9: (a) Procid allows the user to toggle the must read property of a comment and (b) if this comment offers an idea.

view of the comment.

Figure 5.10: The idea lens (in blue) highlights all of the comments categorized as an idea either by the system or user input.

5.2.4 Promoting a Positive Tone

Strong disagreements may result in destructive comments, hindering consensus. Procid promotes a positive tone by detecting when a comment has an overly negative tone relative to prior comments in the community and alerts the author just prior to posting it (Figure 5.11). The goal is not to censor the user, but to prompt a revision that is more constructive. However, once alerted, the user chooses whether to revise or post it as is.

To determine if a comment is overly negative, I computed the sentiment of 9000 randomly chosen comments from Drupal using AlchemyAPI [3]. I sorted

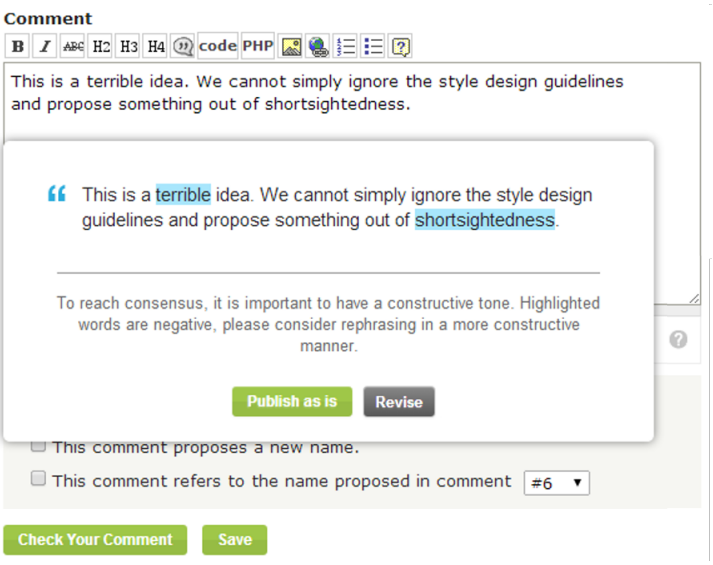

Figure 5.11: Procid analyzes the sentiment of the comment just before it is posted and, if an overly negative tone is detected, highlights the negative words and encourages revision.

the comments based on their negative sentiment and defined the threshold as the average of the top 1% of the comments. Just before a new comment is posted, Procid compares the sentiment of the comment to the threshold. If exceeded, the negative words in the comment are highlighted and the user is encouraged (but not forced) to revise it [31]. The threshold can be easily tuned or more advanced sentiment analysis techniques can be applied in future iterations. Alternatively, the user can choose to have the system check the tone of the comment and highlight the negative words, regardless of how it compares to prior comments.

5.2.5 Rendering Ongoing Discussions

It is very likely that community members would use Procid to participate in ongoing discussions with existing content. For its features to be effective, the tool needs to know which comments propose ideas, which comments refer to ideas, the comment dispositions, etc. One approach is to prompt users to provide this information at first use, but this upfront cost could be large. Instead, Procid applies heuristics to determine the ideas, comments referring to the ideas, and dispositions of comments toward the ideas and offer interactions for revising the assigned values. For example, Procid detects if a

comment contains an idea by calculating a score based on whether the comment contains an image or patch, the number of references to the comment, and the sentiment of the references. If the score exceeds a certain threshold, it marks the comment as an idea.

For each marked idea, Procid parses the discussion to find the comments referring to it. The tool then uses an internal sentiment analysis tool to categorize the comments. Similar to other tools (e.g. SentiWordnet [8] and LIWC [54]), Procid uses a lexicon that consists of two categories of negative (e.g. disappointment, criticism) and positive (e.g. gratitude, pride) words derived from a commonly used online dictionary. A comment is categorized as supportive if it has more positive words, constructive if it has more negative words, and neutral if it has none or nearly equal negative and positive words. This classification provides a default value, but users can override these values using toggles at the bottom of the comment box (Figure 5.12).

Figure 5.12: Users can change the classification of comments by selecting the relevant icon at the bottom of the comment box.

5.3 Revisiting Strategies

I summarize how Procid realizes the six selected strategies. To promote perception of valued contribution (S1), Procid prominently displays the proposed ideas to ensure that discussion participants do not ignore those contributions and organizes the comments referring to those ideas. This helps participants track the ideas and their status and notice patterns of comments for and against the ideas. Also, to help users consider opposing views, Procid refrains from using colors commonly associated with negative affect (e.g. red) [79]. To cultivate a sense of empowerment (S1), the tool enables registering strong support or objection for ideas and visually emphasizes these dispositions in the idea-centric view (Figure 5.7).

To promote constructive tone (S2), Procid analyzes each comment just before it is posted and, if an overly negative tone is detected, cautions the user about posting it and reminds him or her why this is important. For building on prior relationships (S3), Procid generates a list of potential participants for including in a discussion by analyzing their prior contributions, social graphs, and public profiles.

Procids idea-centric view visually summarizes key points of agreement and disagreement (S5). It lists the ideas along with their supportive and constructive comments (Figure 5.2). It also allows participants to create concrete criteria and rate the ideas against the criteria (S4). Procid offers five lenses on the main page for marking and revisiting influential points of a discussion (S6) based on community endorsement, reply behavior, and content of the comments (Figure 5.1).

5.4 User Scenario

Below I explain how a group of users utilize the key features of Procid through a hypothetical usage scenario. This scenario is derived from an authentic usability discussion posted to Drupal issue queue in November 2008 and was finally resolved in April 2012. In this scenario, I assume that all of the users have installed Procid and are using the tool to design a new password checker for Drupal users.

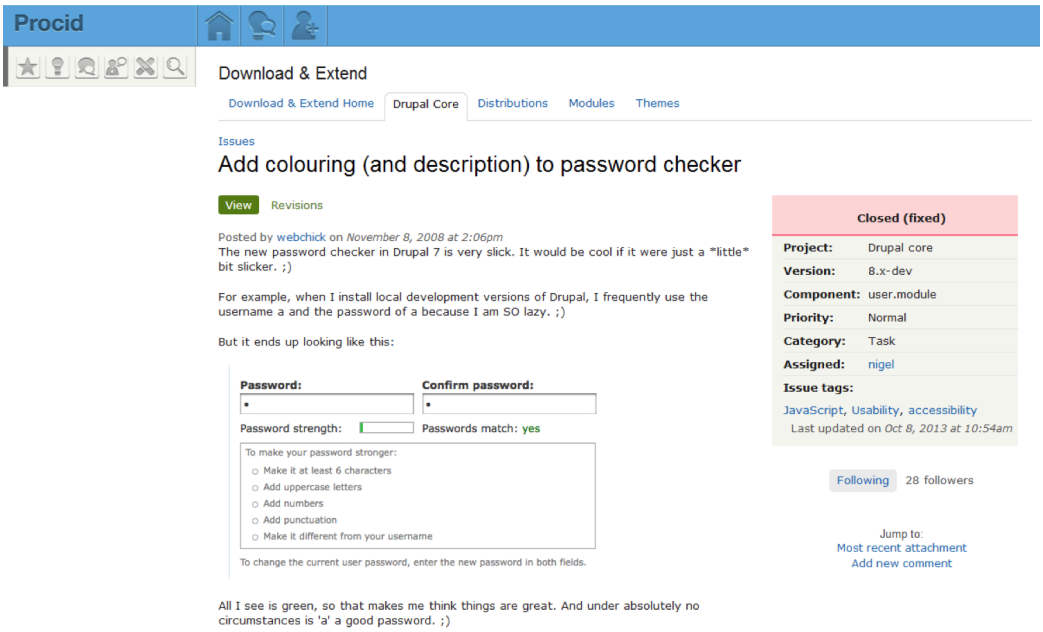

Figure 5.13: Webchick is an experienced developer who contributes to Drupal core. She adds an issue to Drupal's issue management system and explains that the current design of Drupal password checker is ineffective.

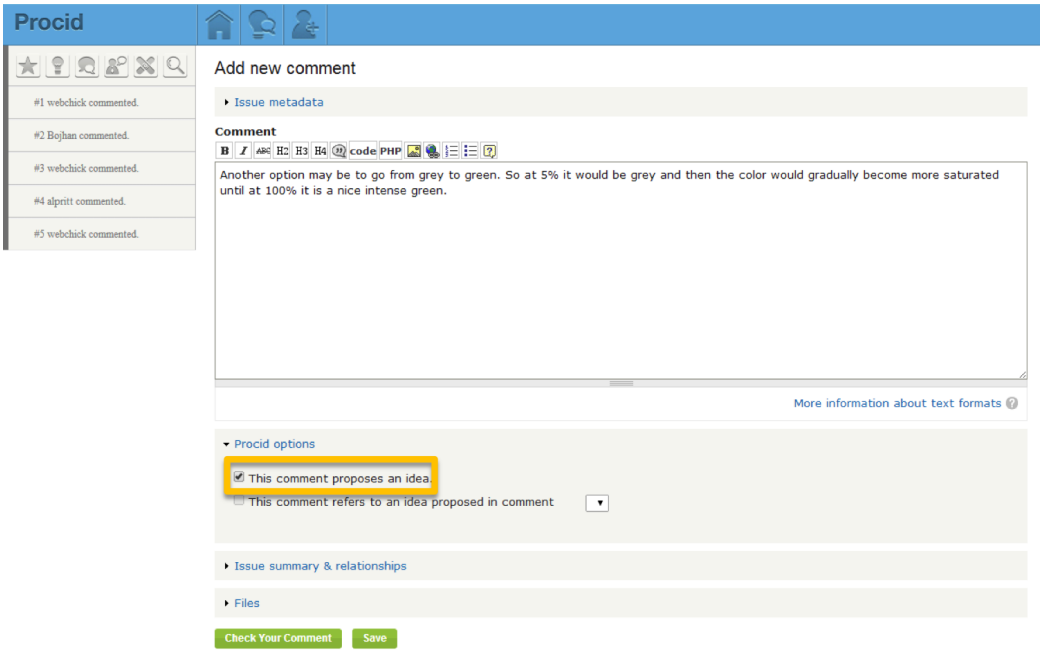

Figure 5.14: alpritt proposes a new idea. He uses Drupal's comment authoring dialog to write his idea. He then uses the interaction panel provided by Procid within the authoring dialog to mark the comment as an idea (this interaction panel is highlighted by an orange border). Procid later uses this information to render the idea-centric view.

Figure 5.15: Procid adds a bulb shape icon on the bottom right corner of alpritt's comment indicating that the comment is proposing an idea.

Figure 5.16: A few other people add comments and propose ideas. Jeff joins the discussion and he clicks on the idea lens on Procid's lens panel to see the list of proposed ideas.

Figure 5.17: Jeff reads both alpritt and Shannon's ideas and he disagrees with Shannon's approach. Jeff thinks Shannon is using too many visual cues.

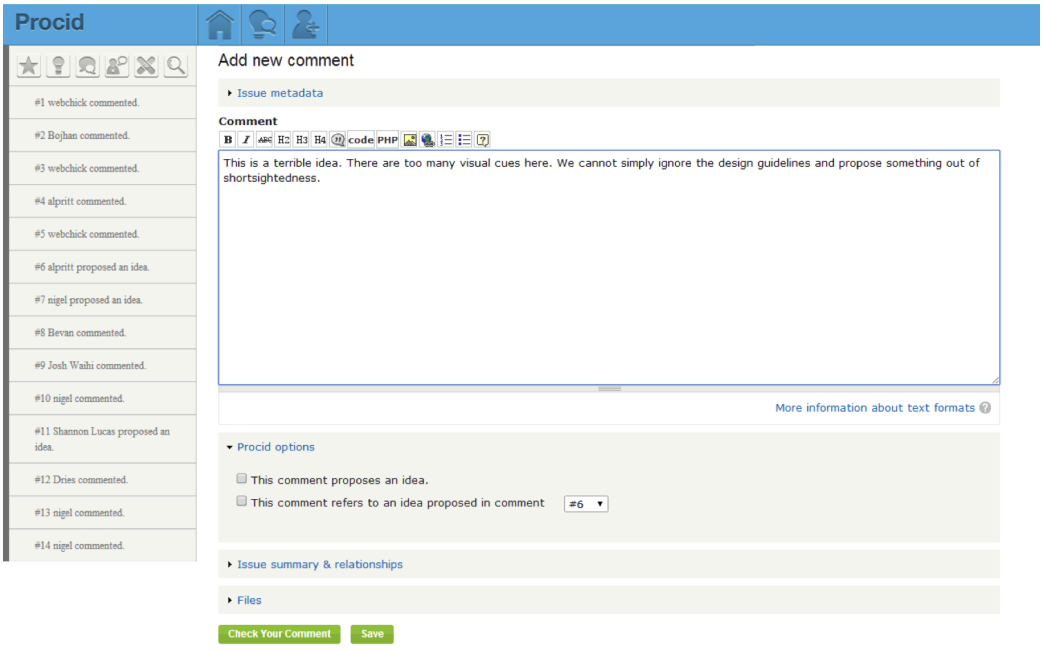

Figure 5.18: Jeff uses the Drupal's comment authoring dialog to add a comment regarding Shannon's idea. His comment has a highly negative tone, but before saving the comment, he clicks on the "Check Your Comment" button provided by Procid next to the "Save" button.

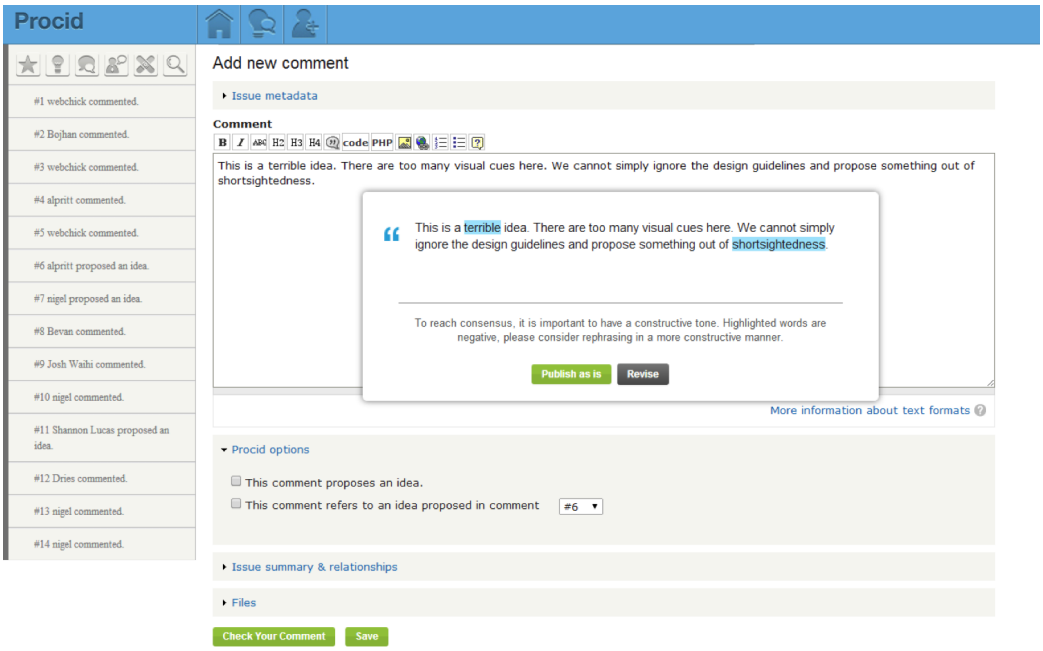

Figure 5.19: Procid analyzes Jeff's comment and highlights the negative words. The tool also shows a message that encourages Jeff to consider revising his comment.

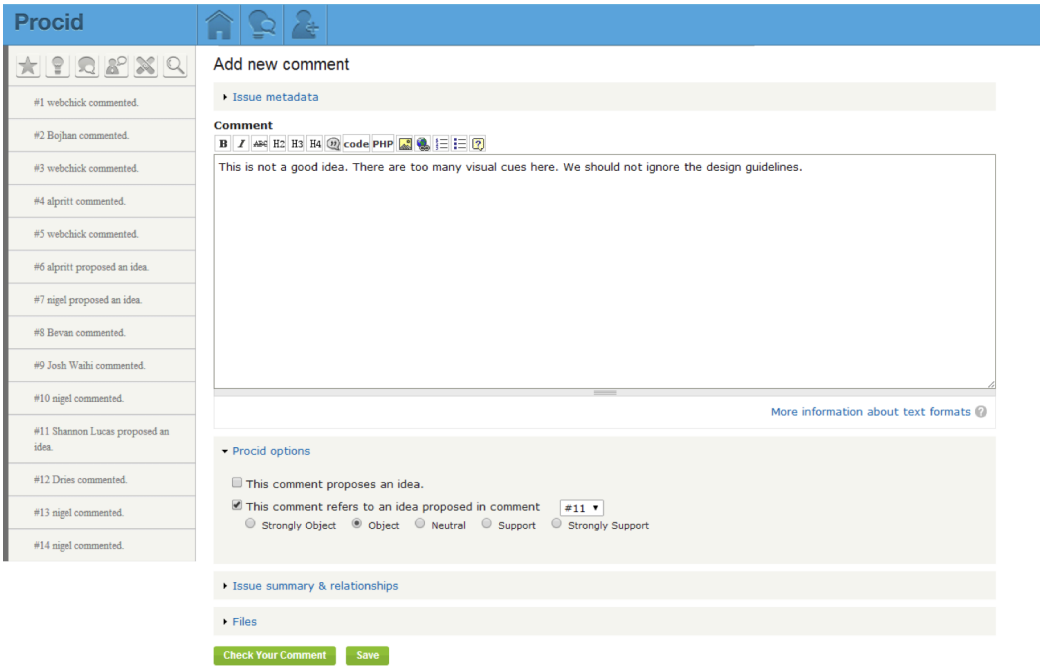

Figure 5.20: Jeff decides to revise his comment, before saving his new comment he uses the interaction panel provided by Procid within the authoring dialog to indicate the comment is related to Shannon's idea. Procid later uses this information to render the idea-centric view.

Figure 5.21: Bojhan joins the discussion. He clicks on the bulb shape icon and opens the idea-centric view to see the current status of the discussion.

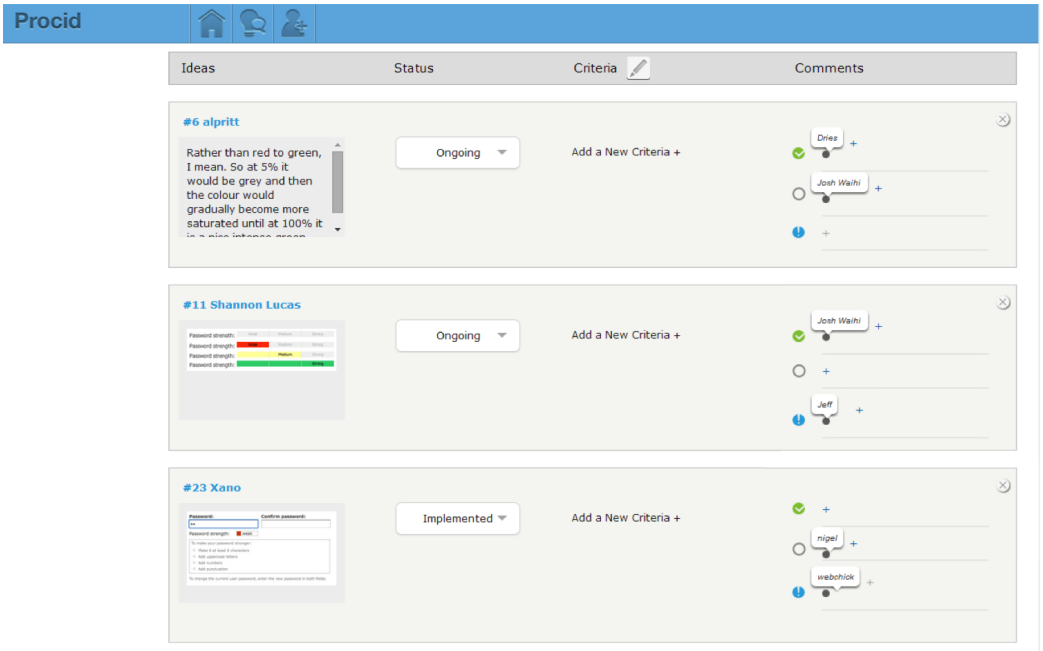

Figure 5.22: Bojhan strongly agrees with alpritt's idea. He clicks on the plus button in the supportive comments row to add a supportive comment.

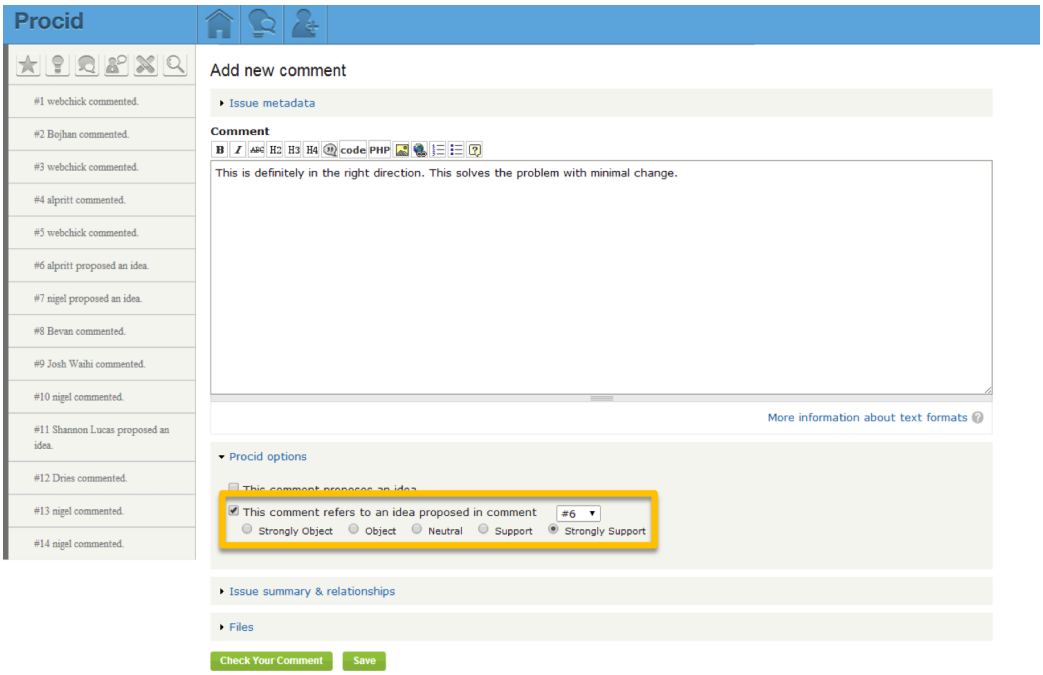

Figure 5.23: He is redirected to Drupal's comment authoring dialog to add his comment. He then uses Procid interaction panel provided inside the authoring dialog to indicate that he strongly agrees with the idea.

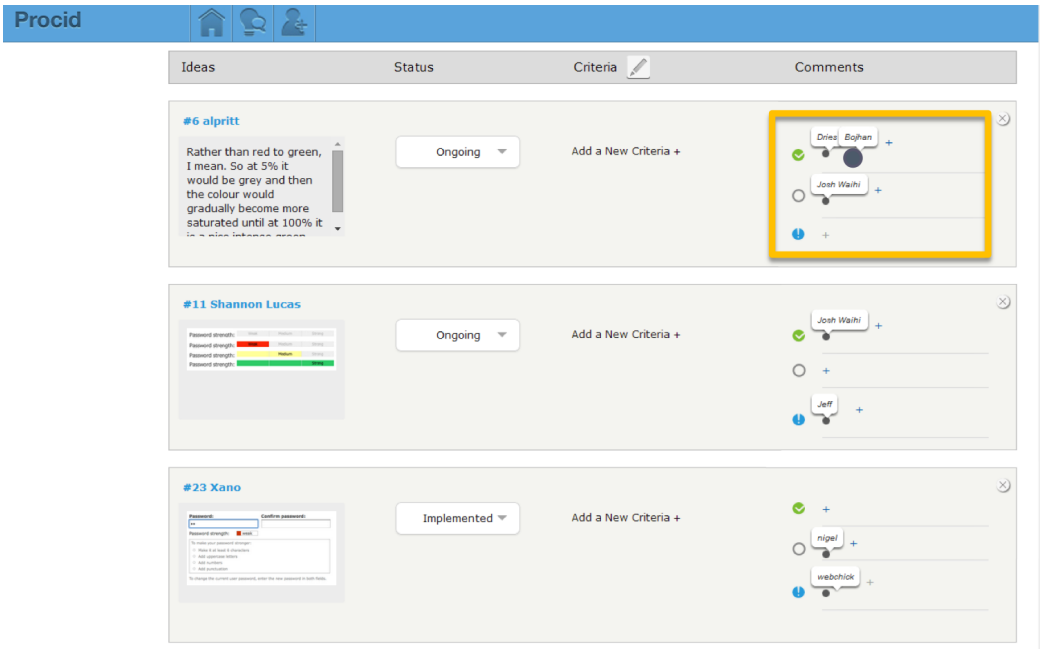

Figure 5.24: Bojhan's comment is rendered with a larger circle to indicate his strong opinion.

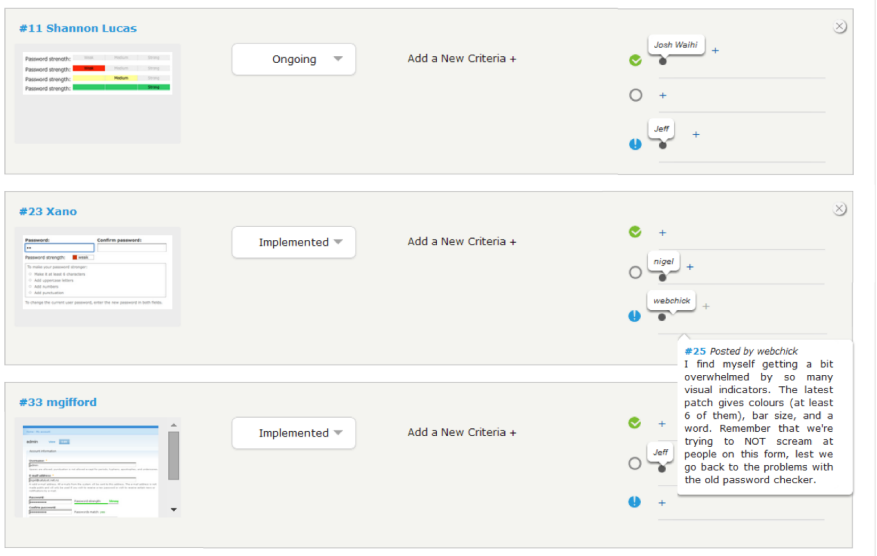

Figure 5.25: Bojhan scrolls down the idea-centric view and clicks on webchick's comment on Xano's idea. Webchick doesn't agree with Xano's idea, because Xano is using too many colors in his design.

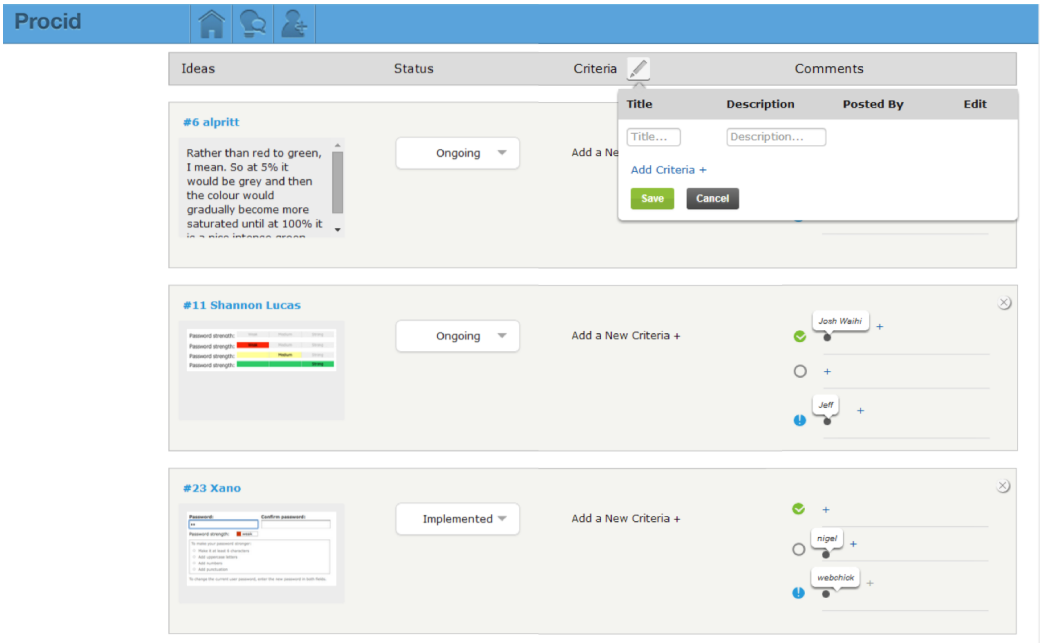

Figure 5.26: Bojhan agrees with webchick's comment. He also thinks that Xano's design is very similar to the design of Google's password checker. Bojhan decides to add two criteria to the discussion. He clicks on the pencil icon on the top of the page and adds "Proper use of color" and "Respect Drupal brand" as new criteria.

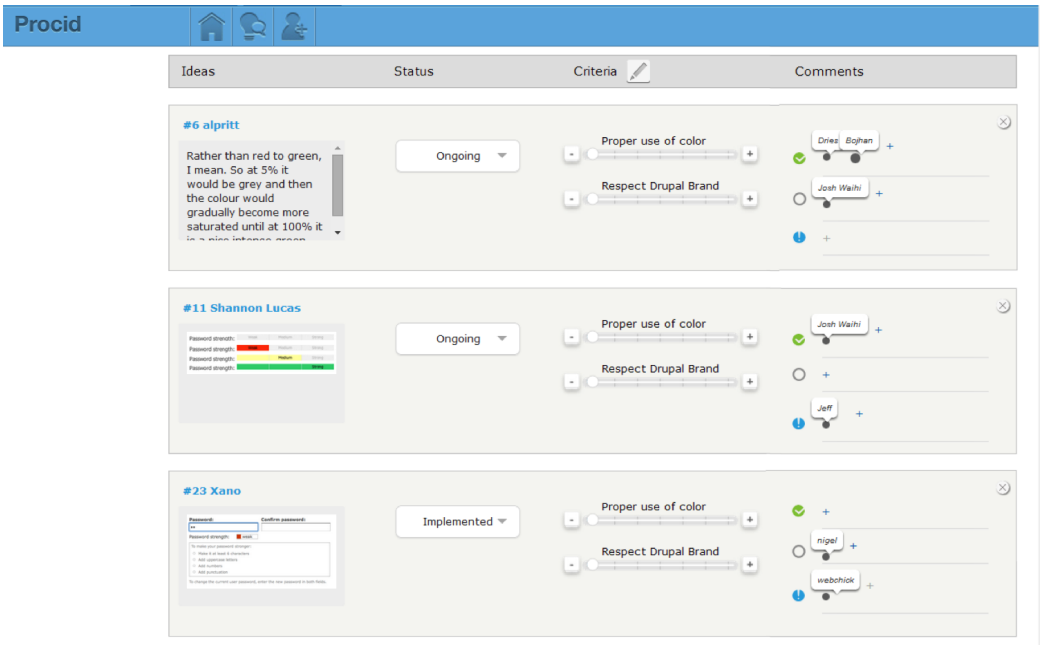

Figure 5.27: The newly added criteria is shown in front of each proposed idea.

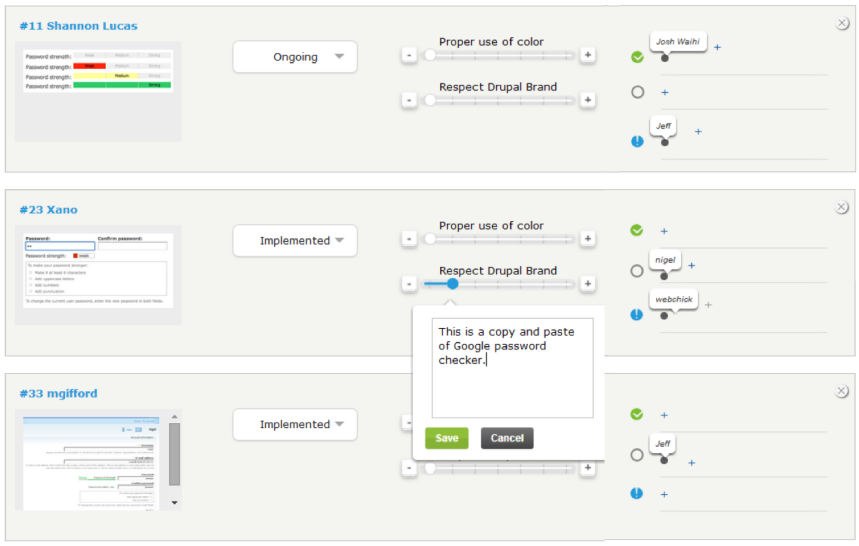

Figure 5.28: Bojhan then uses the slider to rate Xano's idea based on the "Respect Drupal brand" criterion. He also provides a rationale for his rating.

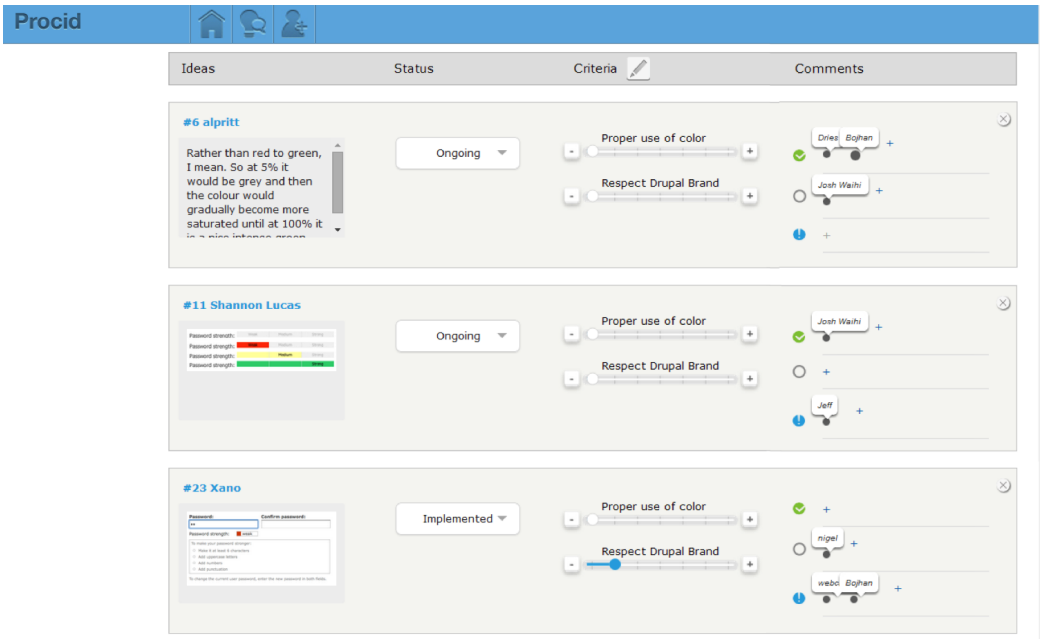

Figure 5.29: The new rating is shown in front of Xano's idea.

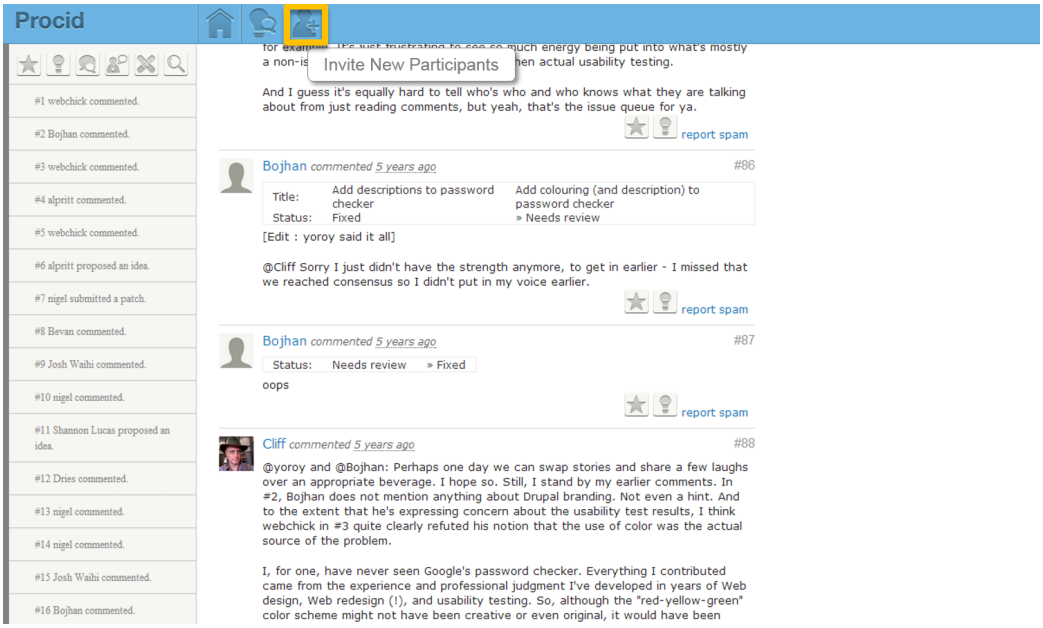

Figure 5.30: The discussion continues. After a while, Michael observes a conflict between Bojhan and Cliff. While Bojhan thinks that they should refrain from using too many colors, Cliff insists that the use of at least three colors is appropriate. Michael decides to invite someone to the discussion to help resolve the conflict. He therefore opens the invite page.

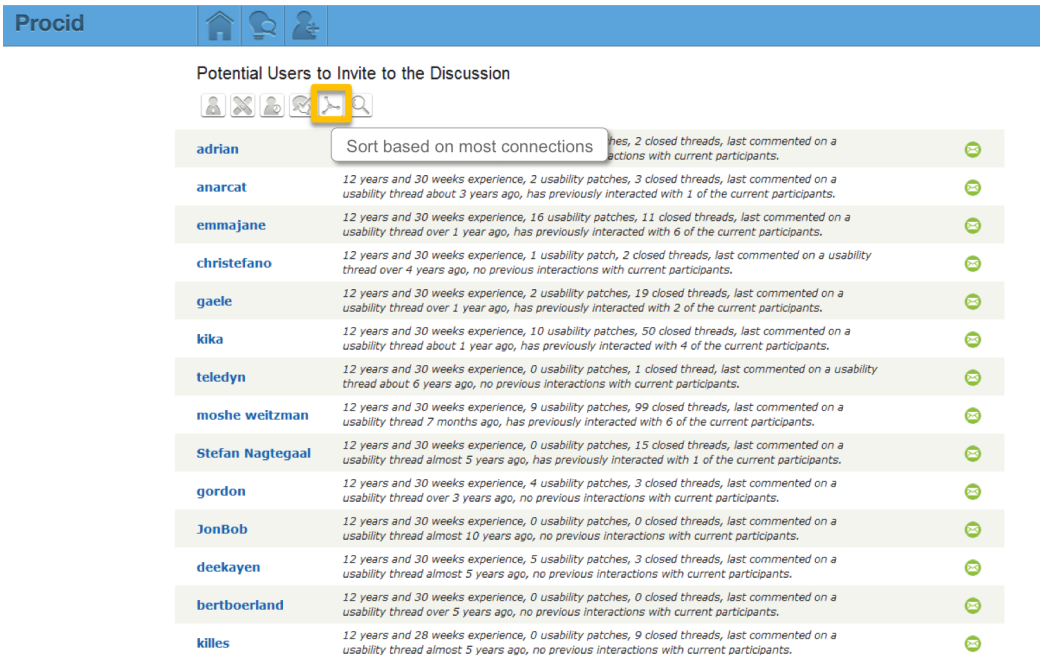

Figure 5.31: Michael sorts the suggested list based on their connection with the current participants.

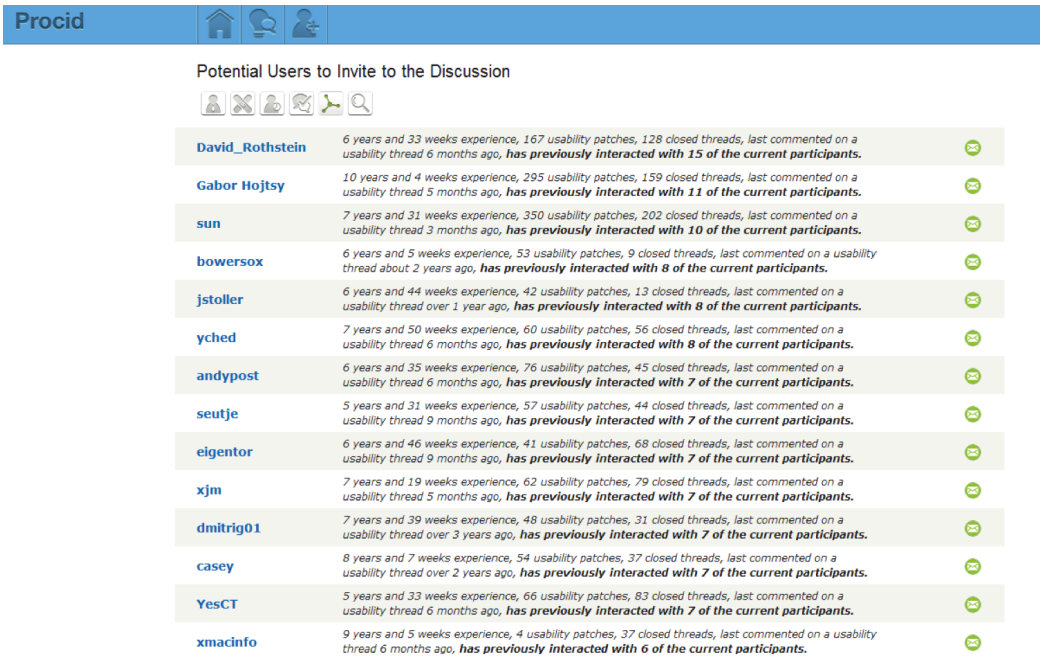

Figure 5.32: He decides to invite David to help resolve the conflict.

5.5 Discussion

Procid helps participants to track and participate in the discussion without imposing a staged process. Anyone can add idea proposals, add new criteria, or rate the ideas based on the criteria at anytime in a fluid manner. However, if the community decides to employ a staged process, Procid can be easily modified to support that.

Procid was inspired by and targets discussions about design issues, particularly in the UI domain. However, the design of the system should generalize to any situation in which distributed participants need to propose and debate concrete proposals for solving problems, e.g. to name a startup. The system is not tied to the type or content of the discussion.

Procid encourages recommended practices of consensus building, e.g., the first comment on a proposed idea should be positive and even critical points should be written in a constructive tone. I believe, in time, the behaviors encouraged by the constraints introduced in the tool would become integrated into the practice of the community. In addition, for practices that cannot be easily integrated into the tool, communities could include training for consensus building as part of the onboarding process [19].

The system provides features useful for consensus building, but it is up to the participants to decide how to use these features during the discussion. Even if fully utilized, there is no guarantee that consensus will be reached or that users are satisfied with the experience of the discussion. Finally, there may be situations in which the consensus building approach is not desirable, e.g., participants may be reluctant to concede individual influence to the collective good. In such cases, the use of a tool such as Procid can still be useful for tracking the status of the discussion.

Chapter 6

Implementation

I built Procid as an add-on to an existing distributed discussion interface, rather than as a stand-alone system. The benefits were that I could discover test principles of consensus building through authentic use, recruit participants who care about the discussion topic, and integrate an existing workflow. It also allowed me to leverage the community's social structure to build novel features (e.g. participant recommendations).

Procid was built using a client/server architecture. The client is a user script (JavaScript and JQuery code) that executes on a browser and customizes the appearance and functionality of an existing discussion interface. To track user data and perform data analysis the script connects to the server built with Ruby on Rails and PostgreSQL. Both client side and server side are open source software under Apache license 2.0 and freely available on Github: `https://github.com/albaloo/procid/`. In this chapter I provide a brief overview of Procid's overall architecture and describe the implementation of Procid-client and Procid-server in details.

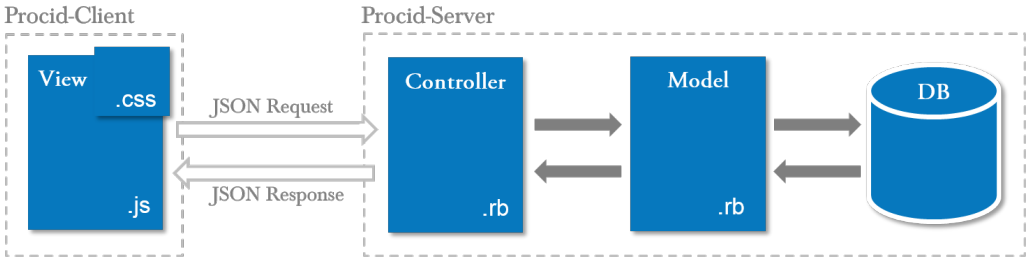

Figure 6.1: An overview of Procid's architecture.

6.1 Architecture

Procid code-base follows the Model2-MVC (Model-View-Controller) pattern. The View is handled by the client side and the Models and Controllers are

on the server side. Figure 6.1 shows the overall architecture of Procid. When the user performs an action on Procid-client (e.g. adding a new constructive comment), the client sends a JSON request to the server. As a workaround to Javascript same-origin-policy, the client uses CORS (Cross-Origin Resource Sharing) to communicate with the server and exchanges data using JSON. CORS is a mechanism that enables a JavaScript installed on one domain (e.g. Drupal.org) to send requests to another domain (e.g. Procid-server). Due to the same-origin security policy, web browsers do not allow"cross-domain" requests by default. Through CORS the browser and the server can interact to determine whether or not to allow the cross-domain request. Listing 6.1 shows how CORS is enabled on the server side in Procid's code base.

The request sent by the client will be received by a controller on the Procid-server. The server then starts processing which usually involves finding a record in the database, adding a new record, or deleting a record. The server then sends back the necessary information (e.g. summary of the newly added comment) back to the client in form of a JSON response.

```
1 class Application < Rails::Application
2    config.middleware.use Rack::Cors do
3       allow do
4          origins '*'
5          resource '*', :headers => :any,
6                                 :methods => [:get, :post, :options]
7     end
8  end
```

Listing 6.1: CORS is enabled by adding this code snippet to the application.rb file in Procid-server. CORS is a mechanism that allows a JavaScript installed on one domain (e.g. Drupal.org) to send requests to another domain (e.g. Procid-server).

6.2 Server

Procid-server is written using Rails 3.2 and PostgreSQL and uses Datamapper for object-relational mapping (instead of Active Record). Table 6.1 shows the main files in Procid-server's codebase. The server has two interconnected parts: models and controllers. Procid-server implements three con-

Controllers	homepageController.rb	553
	ideapageController.rb	272
	invitepageController.rb	79
Models	idea.rb	63
	comment.rb	94
	criteria.rb	20
	criteria-status.rb	10
	issue.rb	672
	participant.rb	19
	tag.rb	9
	pp-connection.rb	9
	ip-connection.rb	9

Table 6.1: Lines of Code for the main files in Procid-server

View	procid.user.js	3935
	style.css	1659

Table 6.2: Lines of Code for the main files in Procid-client

trollers: *HomepageController*, *IdeapageController*, and *InvitepageController* which handle the requests coming from the client's homepage, ideapage, and invitepage respectively. The server also contains eight major models: *comment*, *criteria*, *criteria-status*, *idea*, *issue*, *pp-connection*, *ip-connection*, *participant*, and *tag*. There is a table associated with each of these models in the database. Figure 6.2 presents the database schema.

6.3 Client

Procid-client is written in Javascript and jQuery, making use of d3 for creating complex interactions. Table 6.2 shows the main files in Procid-client's codebase. The javascript code has a modular style and consists of 28 different methods. Once executed the code modifies the HTML code of a Drupal discussion page and attaches a number of new elements to the existing HTML elements. The styling of the new components are loaded from a separate CSS file. Procid-client first adds the homepage (see Listing 6.2). To speedup

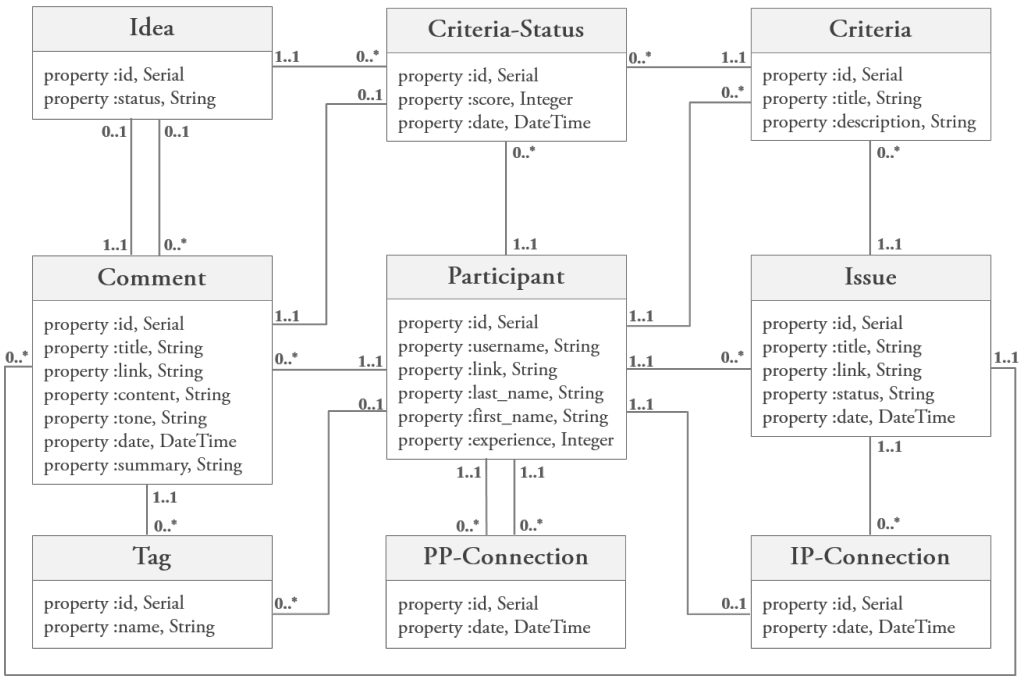

Figure 6.2: Procid-server's database schema.

the loading process, the ideapage and invitepage will be created lazily. Figure 6.3 shows the HTML code of a sample discussion from Drupal before Procid-client was installed and after it was installed.

6.4 Working Together

Procid's codebase (both the server and client code) has been divided in to three modules: homepage, ideapage, and invitepage. For each of these modules, the view is handled by Procid-client and the controller and the models are in Procid-server. In this section I will explain how the client and server work together to make each module work.

```
1  var pageWrapper = document.createElement('div');
2  pageWrapper.setAttribute('id', 'procid-page-wrapper');
3
4  //Wrap the <div id="page"> from Drupal code with a new div
5  $("#page").wrap(pageWrapper);
6
7  //Creating a div element to hold homepageBody
8  var homePageBody = document.createElement('div');
9  homePageBody.setAttribute('id', 'procid-home-page-body');
10
11 //Attaching the homepagebody
12 $("#procid-page-wrapper").wrap(homePageBody);
13
14 //Adding the left panel to homepage
15 var leftPanel = createHomePageLeftPanel();
16 homePageBody.appendChild(leftPanel);
```

Listing 6.2: Procid-client wraps the $< divid = "page" >$ element in the HTML code of a Drupal discussion with a new div element and then adds the homepage body to the wrapper.

6.4.1 Homepage

In Procid homepage users can filter comments from multiple perspectives through five lenses: must read, idea, conversation, experienced, and patch. Each comment may have one or more tags and each tag is associated with a separate lens. Conversation, experienced, and patch tags are determined on the server side and must read and idea tags are assigned by the user at the time of posting. When a user writes a new comment, he/she can use a form provided by Procid-client to mark the comment as an idea or as a mustread comment. The user can also use the form to specify whether the comment is referring to an idea in a supportive, constructive, or neutral way. When the users clicks on save to post the comment, one request will be sent to Drupal server to save the comment and another request will be sent to Procid-server along with the tags provided by the user. Procid-server processes the comment and assigns the patch, conversation, or experienced tags as necessary and computes a summary for the comment. The server then saves the new comment and the tags and sends the comments' tags and

a

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head profile="http://www.w3.org/1999/xhtml/vocab">…</head>
  <body class="html not-front not-logged-in one-sidebar...">
    <div id="skip-link">…</div>
    <div id="header" class="clearfix">…</div>
    <!-- /#header -->
    <div id="page" class="container-12 clearfix">
      <div id="page-inner">…</div>
      <!-- /#page-inner -->
      ::after
    </div>
    <!-- /#page -->
    <div id="footer" role="contentinfo">…</div>
    <!-- /#footer -->
```

b

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <script>…</script>
  <head profile="http://www.w3.org/1999/xhtml/vocab">…</head>
  <body class="html not-front logged-in one-sidebar...">
    <div id="skip-link">…</div>
    <div id="header" class="clearfix">…</div>
    <!-- /#header -->
    <div id="procid-home-page-body">
      <div id="procid-left-panel-header">…</div>
      <div id="procid-left-panel-body">…</div>
      <div id="procid-page-wrapper">
        <div id="page" class="container-12 clearfix">
          <div id="page-inner">…</div>
          <!-- /#page-inner -->
          ::after
        </div>
      </div>
    <div id="footer" role="contentinfo">…</div>
    <!-- /#footer -->
```

Figure 6.3: (a)The HTML code of a sample discussion from Drupal before Procid was installed. (b)The HTML code of a sample discussion from Drupal after Procid was installed.

summary information back to Procid-client in form of a response.

In some cases users may want to use Procid on an ongoing discussion. To be compatible with these discussions, the tool runs a set of preprocessing modules on the comments. When an ongoing discussion is opened with Procid, Procid-client parses the discussion's HTML code and identifies the content, author, and date of each comment. The client then sends the dis-

cussion identifier along with the comments' information to the server (see Listing 6.3).

```
$.ajaxSetup({
    'async' : false
});

$.post(serverURL + "postcomments", {
    "issue" : JSON.stringify(issue),
    "commentInfos" : JSON.stringify(commentInfos)
}, function(data) {
  $.each(data.issueComments, function(i, comment) {
      commentInfos[i].tags = comment.tags;
      commentInfos[i].tone = comment.tone;
      commentInfos[i].comments = comment.comments;
      commentInfos[i].summary = comment.summary;
    });
    criteria = data.criteria;
});
```

Listing 6.3: This code snippet shows how the client sends a JSON request to the server and receives a JSON response with comments' tags, relations, tone, and summary information from the server.

When the server receives the information, it checks to see whether any records for the discussion exist in the database. It then processes the comments. For each comment, if there is a record in the database, the stored tags (e.g. idea, patch, conversation, etc.) are assigned to the comment. Otherwise the server computes values for as many tags as possible for the comment and stores them in the database. For example, the server detects if a comment contains an idea by calculating a score based on whether the comment contains an image or patch, the number of references to the comment, and the tone of the references. If the score is above a certain threshold, it tags the comment as an idea.

Procid-server also organizes the comments referring to an idea as supportive, neutral, or constructive. For each idea, the server parses the discussion to find the comments referring to that idea. It then uses an internal sentiment analysis tool to organize the referring comments. Similar to other sentiment analysis tools (e.g. SentiWordnet [8] and LIWC [54]), Procids analyzer uses

word lists to detect sentiment. The system uses its own lexicon that consists of three categories of negative (e.g. disappointment, criticism), positive (e.g. gratitude, pride), and stop (e.g. is, the) words from a commonly used online dictionary. Removing stop words from calculation, a comment is categorized as supportive if it has more positive words, constructive if it has more negative words, and neutral if it has none or nearly equal numbers of negative and positive words. This classification provides a default value, but users can later override these values using toggles provided at the bottom of the comment box in Procid's idea-centric view.

When the preprocessing of comments is completed, the server sends the comments' tags, tone, relation, and summary information back to Procid-client in form of a response. Procid-client then shows the summary of the comments on the homepage and enables searching and filtering of the comments based on the tags.

6.4.2 Ideapage

To help participants track ideas and key points of agreement and tension, Procid's ideapage provides an idea-centric, spatial organization of comments. One of the key actions that a user can perform in this page is adding a new criteria. This action will both add a new criteria to the ideapage and posts a new comment explaining the newly added criteria to the discussion. The latter is done for two reasons. First, it will make users accountable for the change they make. Second, it enables users without Procid to follow the discussion. When the user clicks on the save button to add a new criteria, first a request will be sent to Drupal server to save the criteria as a comment. Listing 6.4 shows how the request is sent to Drupal.

```
1  var saveCommentToDrupal = function(commentText, issueLink) {
2     var title, link = "";
3     $.ajaxSetup({
4        'async' : false
5     });
6
7     //fill out the comment form with the text
8     $("#edit-nodechanges-comment-body-value").val(commentText
9           + "<i>Powered by Procid</i>");
10
11    //send a request to save the comment form to Drupal server
12    $.post('https://drupal.org/' + issueLink,
13       $("#project-issue-node-form").serialize(), function(data) {
14          var result = $(data).find("div[class^='comment ']").last();
15          title = $(result).find(".permalink").text();
16          link = $(result).find(".permalink").attr("href");
17       });
18
19    return [title, link];
20
```

Listing 6.4: The saveCommentToDrupal function in Procid-Client first fills out the comment form with the given text. It then sends a request to Drupal server to save the comment.

Then another request will be sent to Procid-Server with information about the user, newly added criteria, newly added comment, and the discussion identifier. This request will be received by the addCriteria method in the *IdeapageController*. This method handles the request by saving the new criteria (See Listing 6.5 Lines 11-15) as well as the newly added comment about the criteria (See Listing 6.5 Lines 17-25).

```ruby
def addCriteria
    issueLink = params[:issueLink]
    userName = params[:userName]
    criteriaTitle = params[:title]
    criteriaDescription = params[:description]

    #loading the issue and participant
    currentIssue = Issue.first(:link => issueLink)
    currentParticipant = Participant.first_or_create({
        :user_name => userName})

    #saving the new criteria
    currentCriteria = Criteria.first_or_create({:issue => currentIssue},
       {:title=>criteriaTitle, :description=>criteriaDescription,
        :participant => currentParticipant})
    currentCriteria.save

    #saving the comment added to the discussion about the new criteria
    time = Time.now
    newCommentTitle = params[:newCommentTitle]
    newCommentLink = params[:newCommentLink]
    newCommentContent = params[:newCommentContent]
    newComment = Comment.first_or_create({:issue => currentIssue,
       :participant => currentParticipant, :title => newCommentTitle},{
       :content =>newCommentContent, :link => newCommentLink,
       :commented_at=>time, :tone => "neutral"})

    render :json => {}
end
```

Listing 6.5: The IdeapageController handles the add criteria request that comes from Procid-client.

6.4.3 Invitepage

The invitepage shows a list of users to potentially invite to the discussion. When the page is loaded, a request will be send to the findPotentialParticipants method in the *invitepageController* to get the list. Procid-server compiles the list based on criteria important for consensus such as the number of patches contributed, recency of participation, experience, and prior interaction history with participants in the discussion. For example, to get

a list of users with prior interaction history, the server finds the list of users who replied to any of the discussion participants or received a reply from any of the discussion participants by querying the pp-connection and ip-connection tables(see Listing 6.6). The pp-connection table holds a record for all instances where a participant replied to another participant in a prior discussion. The ip-connection table holds a record for all instances where a participant posted a comment to an issue. The information in these two tables has been gathered by a separate Java code and has been pre-loaded to the database with a ruby script. For the purpose of this project, the information transfer was done manually. However, it can be automated by adding a script to the server.

```
#Finding users who previously replied to current participants
resL = adapter.select("SELECT t1.source_id, COUNT(t1.target_id) AS tr
   FROM (pp-connections AS t1 INNER JOIN ip-connections AS t2 ON
      t2.participant_id=t1.target_id)
   WHERE (t2.issue_id=#{issueid}) AND t1.source_id IN
   (SELECT id FROM participants WHERE NOT EXISTS
      (SELECT participant_id, issue_id FROM ip-connections
         WHERE ip-connections.participant_id=participants.id AND
            ip-connections.issue_id=#{issueid}))
       GROUP BY t1.source_id ORDER BY tr DESC LIMIT 10;")

#Finding users who received a reply from current participants before
resR = adapter.select("SELECT t1.target_id, COUNT(t1.source_id) AS tr
   FROM (pp-connections AS t1 INNER JOIN ip-connections AS t2 ON
      t2.participant_id=t1.source_id)
   WHERE (t2.issue_id=#{issueid}) AND t1.target_id IN
   (SELECT id FROM participants WHERE NOT EXISTS
      (SELECT participant_id, issue_id FROM ip-connections
         WHERE ip-connections.participant_id=participants.id AND
            ip-connections.issue_id=#{issueid}))
      GROUP BY t1.target_id ORDER BY tr DESC LIMIT 10;")

resL.concat(resR)
```

Listing 6.6: The IdeapageController handles the add criteria request that comes from Procid-client.

6.5 Discussion

Building Procid as an add-on rather than a stand-alone application had several benefits (e.g. enabling me to test the interface through authentic use). However, it also imposed a number of limitations. First, the design of Procid-client was bounded by the design of Drupal discussion interface. This means that I had to work with the theme and the existing elements in Drupal interface and I could not change the whole format of the discussion. Second, some of the users may not be willing to install a Javascript add-on on their browsers due to security concerns. This is a valid concern given that a malicious Javascript code may introduce a security vulnerability in their system. Finally, when loading large discussions for the first time, there may be a noticeable performance degradation in Procid-client due to the data preprocessing performed on the Procid-server. It may take up to 10 seconds for the client to receive the processed data and load all of its content.

The implementation of the client is customized to work with the Drupal issue management system. It parses Drupals HTML code and attaches its components to the HTML elements. However, the client uses a modular architecture. Therefore, by updating the parser module (around 15 lines of code) and the HTML elements that the client will attach to (around 10 lines of code), it can work with other issue management systems. The server has its own independent API which is accessible by any client.

Chapter 7

Evaluating Procid

In this chapter I will explain two evaluations that I have conducted on Procid. The first study gauged initial reactions to the tool and collected feedback from the community. The second study was a controlled experiment aimed at empirically comparing the use of Procid to the existing community interface for design discussions.

7.1 Study 1: Initial Reactions from the Community

The purpose of this study was to gauge user reactions to the tool and learn how well it addressed its design goals. The approach was to have members of the Drupal community interact with key features of our tool in context of their discussions. This allowed me to evaluate features for most (G3-G6) but not all (G1, G2) of the goals.

7.1.1 Participants

We recruited nine participants who actively contribute to Drupal. Three participants (male) were designers with an average of 5.6 years of experience, one participant (male) was a site builder with 4 years of experience building websites with Drupal, and one was both a designer and a developer (female) with 2 years of experience in both fields. The four remaining participants (male) were developers with an average of 5.25 years of experience.

7.1.2 Procedure

We chose five usability discussions from the Drupal issue management system. These discussions were active (not closed) and had recent activity and

many comments (μ=124 comments). We then tailored an interface walkthrough for each of these discussions and asked the participants in these discussions to complete the walkthrough. Figure 7.1 and Figure 7.2 show one of the discussions used in the walkthrough before and after Procid was installed respectively. The key point is that a participant could see the discussion s/he was actively participating in rendered in our system and could interact with it. Participants were also informed that none of their interactions with Procid would affect the actual discussion because the content had been mirrored on our server.

The walkthrough began by having a participant install Procid in her Web browser. The participant then navigated to a page that mirrored his or her own Drupal discussion along with Procid. The participant then performed specific tasks relating to each main feature of the tool and freely explored that feature of the tool. After each task and exploration, participants rated their perception of the feature and commented on its strengths and weaknesses. Finally, participants rated and commented on the overall utility of the system and the discussion interface currently used in the community. Participants received $15 for remuneration.

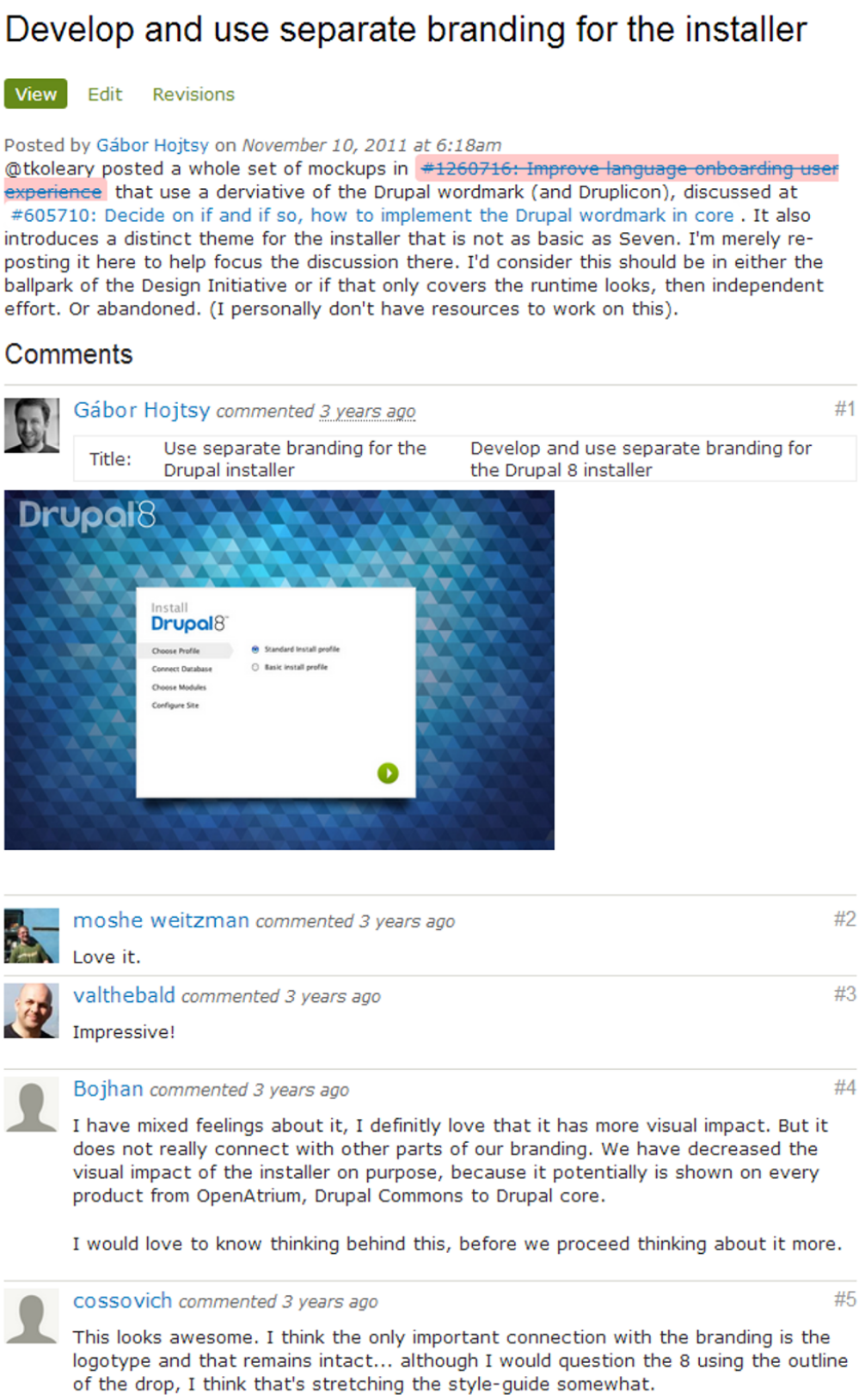

Figure 7.1: A sample Drupal discussion used in the walkthrough without Procid.

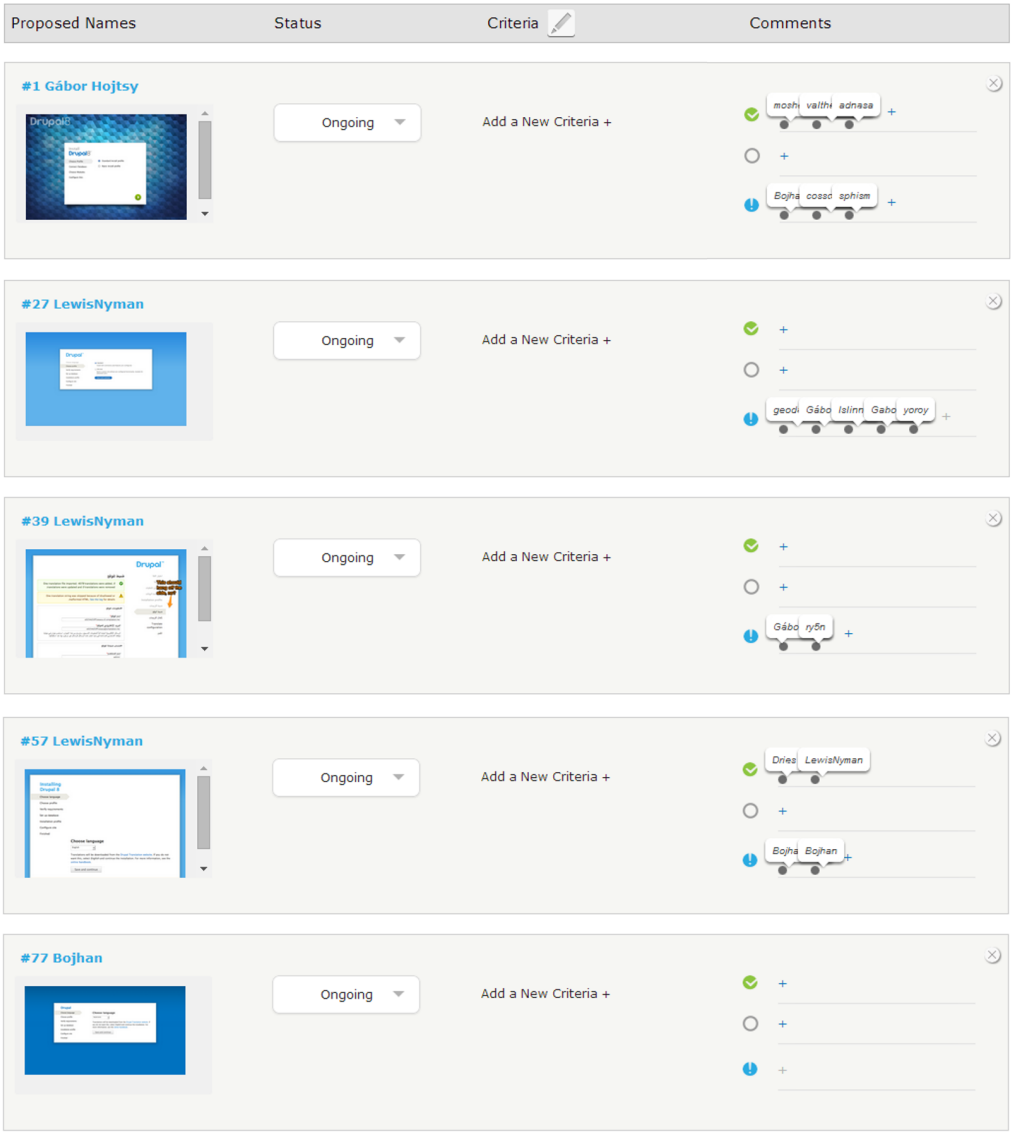

Figure 7.2: A sample Drupal discussion used in the walkthrough with Procid.

7.1.3 Measures

We collected survey responses and logged interactions. In the survey, participants rated the perceived usefulness of different features of the system, the system overall, and the discussion interface currently used. All ratings were made on a 7-point Likert scale ranging from strongly disagree (1) to strongly agree (7). We also recorded participants screen interactions to understand how Procid was used by them.

7.1.4 Results

A summary of survey results is presented in Table 7.1. Participants rated Procid to be more useful for consensus building (μ=4.7, s=1.2) than the communication interface currently used (μ=3.7, σ=1.5; (t(8)=4.2, p=0.003 using paired samples t-test). Two representative statements are:

> I think this could help people see the current status of an issue in a way that our current 'tool' for this (manually-written issue summaries) can't, and in a more in-depth and balanced way than that tool allows. [P9]

> This tool could help focusing on ideas and not so much on people. Furthermore this tool could help to invite feedback even for long running issues with a lot of comments. Such issues normally are very hard to get into but would profit most from additional input. [P5]

Establishing Concrete Criteria to Evaluate Ideas

The feature that participants found most useful in Procid was the ability to establish criteria (μ=5.9, σ=0.6). P7:

> [Criteria related features are] useful for reviewing: you know what to pay attention to. Useful for establishing what should be in scope of this current issue and what could be a follow-up issue.

Participants also agreed that the ability to read the rationale behind idea ratings is useful (μ=5.7, σ=0.7). For example, P9 said:

> I don't think the ranking itself is as important as being able to actually read the rationale behind it (e.g., who gave it the ranking and whether they had something useful and constructive to say about the particular criterion when they did so).

However, three participants were concerned about the risk of establishing a large list of criteria. P7:

> Risk is getting to detailed and specific. Too many of these and the discussion could easily get into a deadlock.

Questions	μ	σ
Q1. The communication tools currently used in Drupal (e.g. issue queue) are useful for building consensus.	3.7	1.5
Q2. The navigation panel is useful for identifying the important comments in the discussion.	5.0	0.9
Q3. The navigation panel helped me in navigating to different parts of the discussion.	5.3	0.9
Q4. Using the navigation panel, I was able to gain an overview of the discussion in an efficient manner.	5.1	1.6
Q5. The Idea Page was useful for tracking ideas and their status.	5.1	1.3
Q6. The summary of supportive, neutral, and constructive comments related to each idea was useful for evaluating ideas.	5.3	1.0
Q7. The Idea Page is helpful for comparing and contrasting the proposed ideas.	5.0	0.9
Q8. Being able to establish concrete criteria is useful.	5.9	0.6
Q9. Seeing the ratings of ideas on different criteria is useful for evaluating ideas.	4.9	1.0
Q10. Being able to read the rationale behind the idea ratings on a criteria is useful.	5.7	0.7
Q11. Seeing the recommended people and the summary of their attributes is useful for identifying who to invite to the discussion.	5.6	0.7
Q12. The sorting options provided in the Invite Page are useful.	4.3	1.4
Q13. I would use this interface for future discussions.	4.8	1.4
Q14. The interface was easy to use and learn.	4.6	1.2
Q15. The interface would help with reaching consensus.	4.7	1.2

Table 7.1: Summary of the survey results. From left to right, the table contains the questions, mean and standard deviation for the interface elements tested. Responses used a 7-point Likert scale ranging from strongly disagree (1) to strongly agree (7).

One way of addressing this issue is to limit the number of criteria that people can establish. Future research is necessary to find the right balance and how it would affect the discussion.

From the interaction data, four of the participants established criteria such as "Scalability", "Accessibility", and "Performance". Two of the participants rated the ideas against the criteria established by earlier participants.

Seeking Potential Contributors

Participants found the invitation list and the summary of members attributes useful for identifying who to invite to the discussion (μ=5.6, σ=0.7). P1:

> This is a good idea, if one wants to get someone involved. Else there is hardly a way to do this if you do not want to use their contact form on d.o. [drupal.org].

The majority of participants stated that in the proposed list they actually found a member they would like to invite to the discussion (78%). P4 said:

> Yes, I immediately saw people that I should invite to the discussion. The algorithm used for this is interesting, its slightly geared towards existing users and not really 'current' users. As in active currently within the UX issues. [current users refers to those who are currently active]

Two of the participants were concerned about overusing this feature. P1:

> It depends of how much it is used. If a lot of people invite the same users to different or the same discussion, they may feel spammed and get angry or not go to the discussion.

This potential issue may be addressed by removing the people who have been recently invited to this discussion from the invitation list.

Two participants wanted the selection algorithm to be more transparent. Participants also had suggestions for improving the algorithm. For example, one of the participants stated that selecting people based on experience will not select the people who are currently active in the usability discussions. Another participant recommended that we use topic similarity in our selection algorithm. P8:

> Facts that person has X patches, or participated in an issue which is now successfully closed, or was recently on the site, can't guarantee that person will be interested in this specific issue. [...] What is useful is relevance, e.g. if person participated in similar issues on the same topic.

One of the participants recommended that we add the invite feature on the comment level as well as the issue level. This would enable the participants to invite members to respond to a particular comment, rather than the whole issue.

An Idea-Centric View of the Discussion

Participants found the idea page useful for tracking the ideas (μ=5.1, s=1.3). P9 said:

> It reminded me of some of the earlier ideas proposed in the drupal.org issue (I'm already a follower of this particular drupal.org issue so I'm somewhat familiar with it, but I'd forgotten about some of the earlier discussion and this helped me remember it).

They also found the spatial visualization of supportive, neutral, and constructive comments to be useful for evaluating ideas (μ=5.3, σ=1). P5 said:

> Especially in an issue where multiple solutions are proposed like this one, the idea page helped me to get an overview about the different proposals and the support they already have.

Participants also found the status field useful, as it enabled them to determine the status of ideas without having to read through the thread. P6 said:

> It was good to see the actually status of each comment and whether or not it had been implemented. I would not have been able to find this without reading through the entire thread.

The interaction data showed that three of the participants actually updated the status of three different ideas. As for areas of improvement, two participants mentioned that they would prefer a larger display of ideas and three indicated that it took them a while to understand the different features of the idea page.

Marking and Revisiting Influential Points

Participants found the five lenses useful for navigating to different parts of the discussion (μ=5.3, σ=0.9) and for identifying important comments (μ=5, σ=0.9). For example, P5 said

> The panel absolutely helped to dive into a bigger, already evolved issue. Usually those issues are very hard to get into and it gets cumbersome to read similar comments over and over again especially if the issue goes on for a longer time period. I especially like the comment highlighting. This feature helps massively to find the comments that bring the meat to the issue so to speak.

Users also identified some areas for improving the lens panel. For example, three of the users remarked that they would prefer if the lenses filtered the comments, instead of highlighting them. For example, P1 said:

> All filters should not only highlight the selection, but hide all other comments from the panel. This would save scrolling and make it much more navigable.

Two users were concerned that highlighting the comments provided by experienced participants may discourage newcomers.

> The bigger importance given to 'experienced' user could block out valuable input from user lower on the ladder [P5].

From the interaction data, the most frequently applied lens on the main page was the idea lens (n=17), followed by conversation (n=14), experienced (n=13), patches (n=11), and must read (n=11).

Additional Insights

Participants found the lenses and idea-centric organization of Procid useful in terms of consensus building. For example, P4 said:

> Its easier to get a 'big picture' of what is going on, where its going and what is missing. Its actually quite a big tool

P7 also said:

> The ideas page is very useful, allows for easier comparison of different ideas and seeing who and how much feedback each idea gets

To make Procid more useful for consensus building, participants suggested ranking the ideas in descending order or adding a status field to communicate the status of the whole discussion. P7:

> Maybe the ideas page can be sorted in reverse, with the latest most current idea on top. Might help focus the discussion on the latest and greatest idea. [...]. Another thing that could be useful is to have some kind of status indication for the issue as a whole: are we still exploring options or are we looking for detailed reviews on an almost done solution.

The interaction data showed five of our participants tried the interface on issues other than the ones linked in the walkthrough. They mostly explored the lens panel for those issues (e.g. clicking on patch, idea, and must read lenses).

7.2 Study 2: Comparative Evaluation

The purpose of the second study was to compare how Procid affects the content, interactions, and user perceptions for distributed design discussions relative to the existing discussion platform used in Drupal. The latter platform is representative of most interactive Web forums used today. Given the limited experience participants would have with either interface, the focus of this study was on user perceptions of and experiences with the platforms rather than the decision quality.

7.2.1 Design Tasks and Participants

The design task was to propose and debate names for a startup and was selected for three reasons. First, the task of naming a startup is a hard design problem. The task is ill-structured, many creative solutions are possible, there are no defined evaluation criteria, and the outcomes matter. Second, many design discussions in the community target selecting effective labels for various interface elements. Finally, proposing and naming a startup does not

	Status quo		Procid	
	Week 1	*Week 2*	*Week 1*	*Week 2*
Startup A	G1	G4	G3	G2
Startup B	G2	G3	G4	G1

Table 7.2: The naming tasks and interfaces were assigned to the groups using a Latin-square design.

require user interface design and development expertise, therefore making the task accessible to a wider range of participants.

To make the discussion authentic, I interacted with two technology startups that were seeking effective names. One company is commercializing technology to verify network data flow security and correctness in real time (startup A). The other one is developing a content creation and camera hardware solution that enables users to create immersive experiences for advertising and entertainment (startup B).

I recruited 37 (15 female) participants with diverse disciplinary backgrounds from a large university. None of the participants had knowledge of this project. Ages ranged from 18-34. Based on self-reports, about half (18) of the participants frequently participated in Web discussion forums, while the other half did not. Participants received $15 for participating in the study.

7.2.2 Discussion Interfaces and Experimental Design

I created eight issues in the Drupal issue management system for the design tasks and participants created Drupal accounts. Permission to do this was requested and granted from the administrators of Drupal. This approach allowed participants to use the discussion interface in Drupal and to use Procid as described earlier. To disguise which interface was developed by me, I always referred to the existing discussion interface as Interface A and my tool as Interface B. Also, participants installed a script for each interface, though the script for Interface A performed no function. Figure 7.3 shows a sample discussion rendered using Interface B and Figure 7.4 shows a sample discussion rendered using Interface A.

The study was a within-subject design with Interface (our tool vs. existing

discussion interface) as the one factor.

7.2.3 Procedure

Participants went through an informed consent process and were given an overview of the study. Participants were then randomly assigned to one of four groups (G1, G2, G3, G4), each with 10-12 participants. The order of the naming tasks and interfaces used were balanced using a Latin-square design (see Table 7.2).

The study was conducted over two weeks. During the first week, two groups (G1 and G2) used the status quo interface to name startups A and B, respectively. The other groups (G3 and G4) used Procid to name startups A and B respectively. At the beginning of the week, participants received the installation instructions for the assigned Interface, a video demonstrating its main features, and details of the task in an email. Participants were requested to generate and debate as many names as possible, and to identify their favorite name by the end of the week. Also, participants completed a survey on their experience using the assigned Interface. During the second week, each group switched to use the other Interface for the other design task. The procedure was then the same as the first week.

Each participant was asked to post at least ten helpful comments to a discussion. To motivate participation, we framed the naming tasks as a competition for each week. Groups earned points using a simple point system (Table 7.3) intended to reward effort and quality. The group with the most points at the end of the week won the competition for that week and each participant received an additional $15.

To make the discussions engaging and realistic, the startup founders gave feedback on the direction of the discussion twice during each week. After the study, we conducted interviews with eleven of the participants asking about their perceptions of and experiences with each interface.

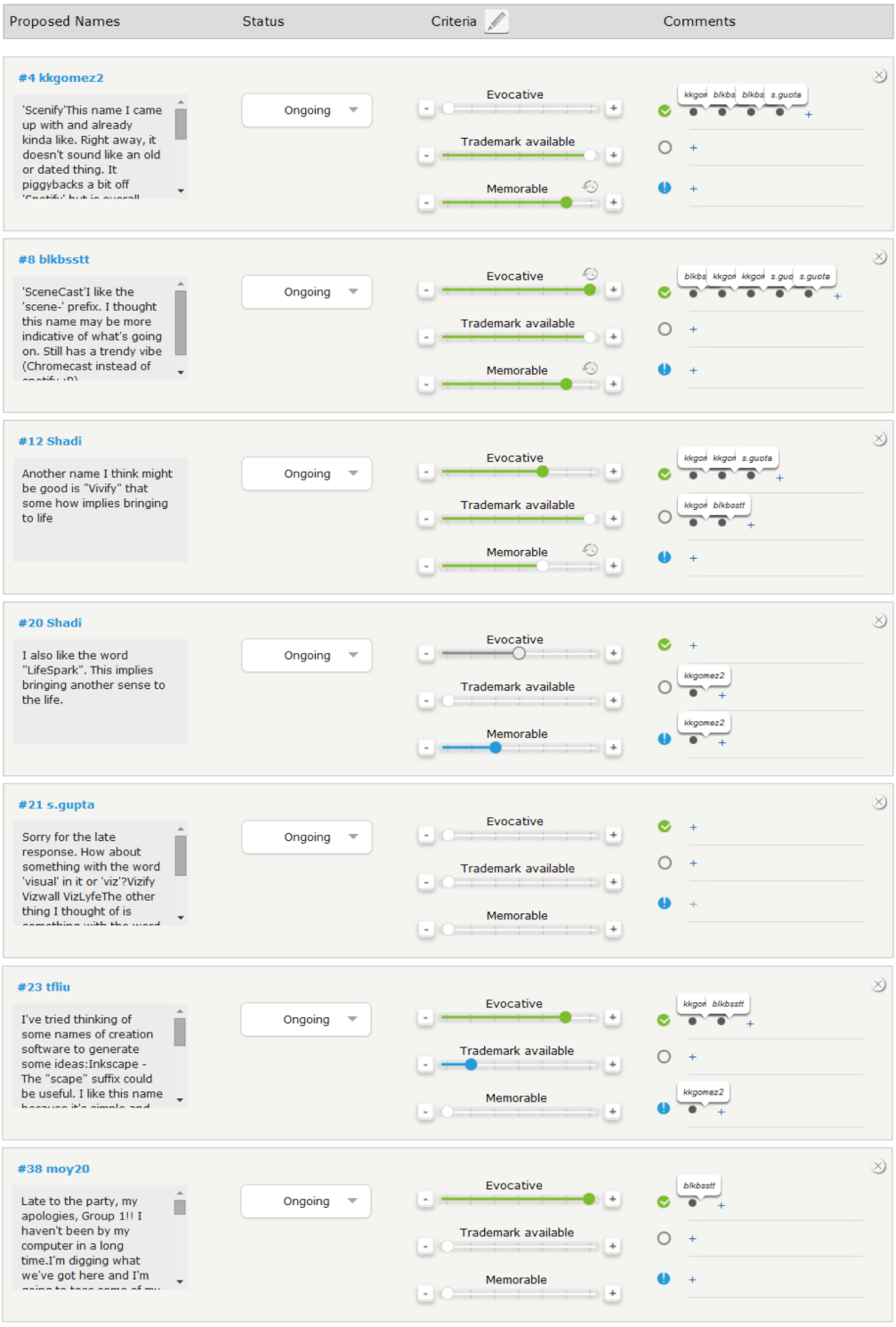

Figure 7.3: G1's discussion rendered using InterfaceB (Procid).

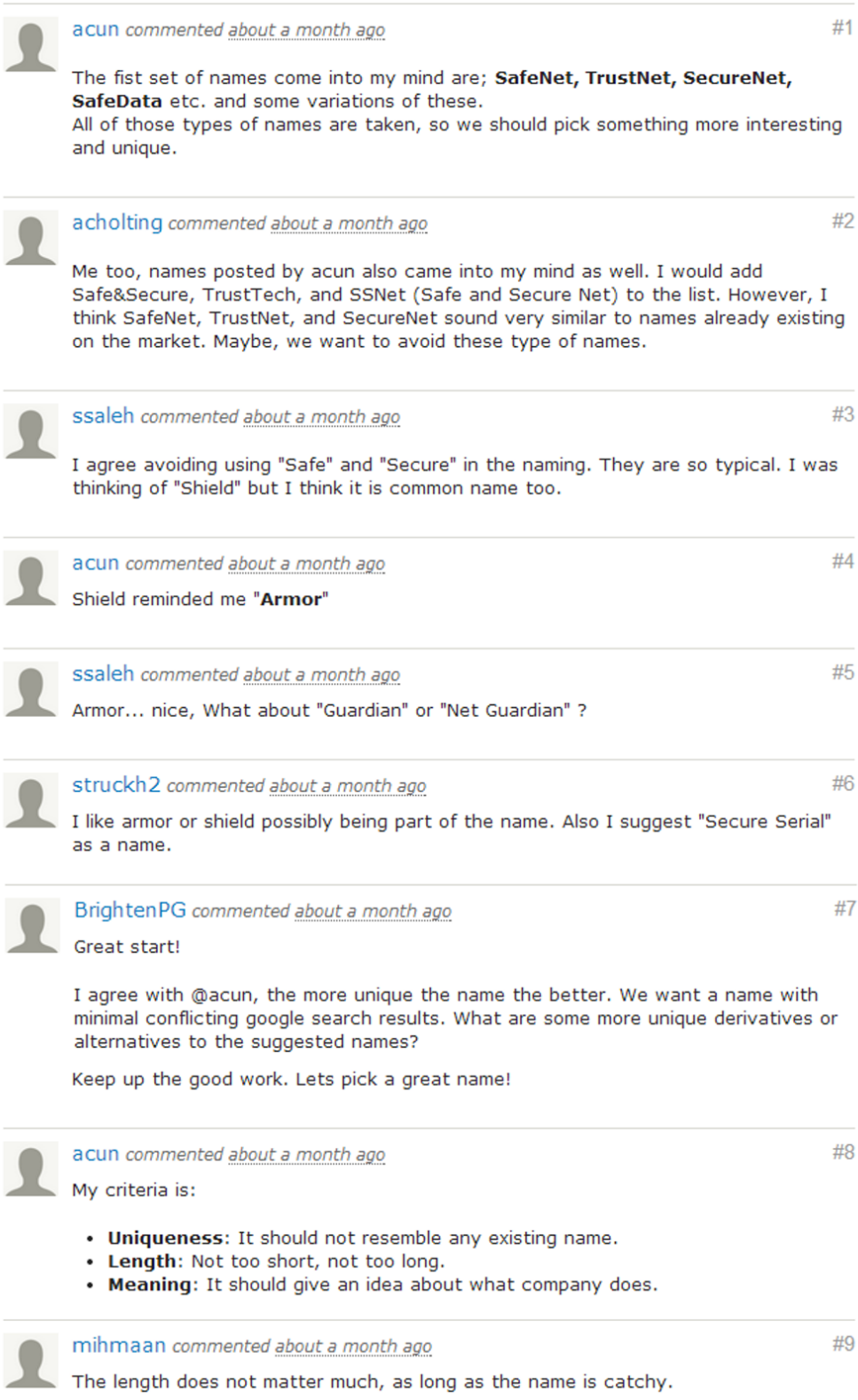

Figure 7.4: G4's discussion rendered using InterfaceA (status quo).

Contribution	Points
Propose a meaningful startup name.	2
Evaluate a proposed name.	3
Determine criteria to evaluate the names.	3
Post other meaningful comments.	1
Building consensus on a name that company founders chose as their favorite.	15

Table 7.3: A simple point system was used to motivate participation and determine the winning group.

7.2.4 Measures

I collected data from the discussion content, interaction logs, surveys, and interviews. For the content, I measured the number of comments, names proposed, criteria defined, and evaluative statements. The interaction logs recorded each participant's interactions with the features of Procid. For the surveys, participants rated the perceived usefulness of the assigned interface for the consensus-related tasks and its overall usefulness. Ratings were made on a 7-point Likert scale ranging from strongly disagree (1) to strongly agree (7). At the end of the second survey, participants identified what they believed were the important differences between the two interfaces, selected which interface they would use in the future and why, and explained which interface helped the most with consensus building and how. I will refer to survey participants as S#. At the end of the study, I conducted semi-structured interviews with 11 participants. Participants were asked about their overall perceptions of Procid and the usefulness of its main features. I will refer to interview participants as I#. Each interview lasted 15-20 minutes.

7.2.5 Results

Participants posted a total of 480 comments during the study. Of those, 254 contained proposed names (ideas) and 26 had criteria for evaluating the names. This shows that the participants were engaged in the study. Examples of some of the proposed ideas were VeriPut, DataRail, Raptor, GreenWall, and SecuraNet for startup A and Phantomizer, Mirage Maker,

		G1	G2	G3	G4
Status quo	*Comments*	36	62	66	40
	Criteria	3	1	2	3
	Ideas	41	36	41	21
	Evaluations	17	34	30	15
Procid	*Comments*	55	59	83	79
	Criteria	3	5	5	4
	Ideas	46	19	18	32
	Evaluations	31	27	53	39

Table 7.4: Number of comments, proposed ideas, user-defined criteria, and evaluation statements in the discussions.

ProMagix, Luminocity, and Morphology for startup B. The winning name for startup A was *NetGator* and the winning name for startup B was *Astral*, both of which the startup founders thought were creative and had not been considered beforehand. The sets of proposed names were shaped and filtered based on user-defined criteria that emerged during the discussions. Examples of criteria were "Reflective: The name should reflect the company's idea", "Trademark available: Whether trademark is available", "Catchiness: How catchy the name is", and "Uniqueness: The name should be different from the existing companies." For example, the name GreenWall was decided against because it wasn't reflective as S25 pointed out in the discussion:

> The name GreenWall suggests something related to the environment. But the company is focused on security in a network.

Table 7.4 summarizes the comments, ideas, criteria, and evaluative statements for both interfaces. Though the trends favored Procid, there were no statistical differences due to the relatively small number of groups.

I also analyzed how the content of the discussion evolved over time, see Figure 7.5. As shown in this timeline, in the discussions that happened on Procid the majority of the criteria emerged at the beginning of the discussion and overall participants posted more criteria compared to the status quo condition. This is not surprising given that Procid provides features for adding criteria and rating the proposals based on the criteria. Another interesting

pattern is that when using Procid participants proposed new names throughout the discussion, while the majority of proposals in the status quo interface were batched at the beginning. This may indicate that because of Procid's idea-centric view, participants notice and evaluate each other's ideas before proposing their own.

Figure 7.5: An event timeline showing when new proposals, evaluative statements, and criteria were posted. As shown in this timeline, in the discussions that happened on Procid the majority of the criteria emerged at the beginning of the discussion and overall participants posted more criteria compared to the status quo condition. Also when using Procid participants proposed new names throughout the discussion, while the majority of proposals in the status quo interface were batched at the beginning.

Overall Preference

From the survey, a large majority of participants (31 out of the 37) experienced Procid to be more helpful for consensus building than the status quo interface and preferred to use Procid for future design discussions. From the open-ended responses, the strong preference for Procid was due to its organization of comments (n=5), the ability to create and evaluate names based on the criteria (n=7), the ability to view the summary of agreements and disagreements with each name (n=7), and the ability to track the proposed names (n=5). Three representative statements were:

> Interface B's second screen; a lot more information is presented in a shorter amount of time. Even though I like to read the

> paragraphs for a poster's entire meaning/explanation behind his or her proposal, I think most forum users/readers do not have the patience to go through an entire thread. Interface B seems to solve that problem by sifting out the important bits and showing the highlights. Though scoring each post at the time may take longer, in the long run and for others on the forum it makes things a lot easier. [S4]

> It was easier to reach a group consensus on Interface B, because you could make a Positive, Neutral, or Critical vote/comment on someone's proposed name. This made it way easy to see the front-runners in public opinion of the board, rather than trying to keep track of which people agreed, which results in the last discussed names to be the main ones being considered. [S19]

> B because you can easily view suggested names, provide feedback and rate them in a separate view. In A you have to scan through posts and keep track of the thread. [S5]

Conversely, the majority of participants noted the lack of organization of comments and difficulty in tracking names as weaknesses of the status quo interface (Interface A). For instance, S6 wrote:

> It was really hard to keep track of all the posts. Some of them had proposed new criteria, some new names, and some where just supporting, or disagreeing with the suggested names. The main con was that all these posts were in one thread and there was no organization. For example, if you just wanted to know a list of all suggested names you had to read all the posts.

Experienced Differences

From the survey, participants reported experiencing three key advantages of Procid relative to the status quo interface. One advantage was the ability to evaluate ideas based on user-defined criteria (n=13). Three representative responses from the surveys and interviews were:

> Interface B allowed you to rate names based on criteria and it made it easy to move forward from there. Whereas Interface A was simply a forum discussion. [S3]

> Interface A was quite plain and not very interactive but got the job done. Interface B on the other hand was very useful and allowed posters to really evaluate other posts and facilitated a meaningful discussion. [S5]

> I feel its pretty much one of the only ways you can do that sort of thing. It's like you cannot really have an abstract goodness or badness you have to say why it's good why it's bad. I feel like the criteria, really do help you determine why it's good or why it's bad. [I5]

As shown in Table 7.5, the survey responses also showed significantly higher ratings for Procid than the status quo interface on the three questions relating to criteria use (Q4, Q5, Q6). And, from the content analysis, groups created 14 criteria when using Procid while they created only 9 criteria using the status quo interface. Participants also wrote a total of 150 evaluative statements based on criteria when using Procid, but only 96 statements when using the status quo interface.

A second key advantage was the ability to track the proposed names when using Procid (n=10):

> The option for viewing the already proposed names is super convenient because then you dont have to read the whole discussion to see them. [S26]

> Interface B was easy to follow different names and you can just focus on the specific parts. However, Interface A was simple and much easier to work with. You had to read most of the comments to be aware of discussion. [S35]

Procid was also rated as being significantly better for tracking ideas (Q7, Q8) than the status quo interface.

The third major difference was the better organization of comments provided by Procid (n=8):

> [Interface B] seems much more organized. Interface A is simpler, but it can be hard to determine what the key points in the conversation are. This would be particularly true if the discussion is very long. [S25]

Questions	**Status quo**		**Procid**		p
	μ	σ	μ	σ	
Q1. I felt my comments were considered by others in the discussion.	5.73	1.18	5.62	1.07	0.34
Q2. I felt that I could express strong support for or against a name.	5.24	1.44	5.95	1.11	0.00**
Q3. The proposed names received critical comments after posting.	5.11	1.16	4.95	1.01	0.24
Q4. It is important to have objective criteria for evaluating the names.	5.70	1.21	6.14	0.96	0.00**
Q5. It was easy to establish objective criteria to evaluate the names.	5.08	1.17	5.89	0.95	0.00**
Q6. Participants evaluated the names based on the criteria.	5.00	1.19	5.81	1.01	0.00**
Q7. It was easy to track the discussion and the status of the names.	3.78	1.71	4.97	1.65	0.00**
Q8. It was easy to find other participants opinions about a name.	4.03	1.75	4.81	1.61	0.030**
Q9. It was easy to find the important comments in the discussion.	3.65	1.76	4.81	1.56	0.00**
Q10. I felt that I was encouraged to add positive comments.	5.14	1.32	5.41	1.17	0.06*
Q11. In our group, we had a collaborative discussion.	5.46	1.20	5.65	0.85	0.23
Q12. This interface supports the consensus building process.	4.76	1.38	5.24	1.12	0.06*

Table 7.5: Summary of the survey results. From left to right, the table contains the questions, mean and standard deviation for the interfaces tested, and results of a paired-sample t-test. ** = p <0.05; *= p <0.10. Responses used a 7-point Likert scale ranging from strongly disagree (1) to strongly agree (7).

> Interface A was just a discussion that kind of just blended down so like people would just keep going and going which is good but then they forget the stuff that happened before and then we would have to bring that up again and so Interface A was just kind of like a flow of stuff where you could forget the past discussion whereas Interface B you could advance your discussion a lot more better because things in the past are more apparent. [I9]

In contrast, the main relative strengths of the status quo interface were openness (n=8) and ease of use (n=9).

Feature Use

From the interaction data, we found that the groups used most of the key features of Procid (see Table 7.6) with Groups 3 and 4 making the most extensive use of them. The most common interactions were with defining criteria, evaluating ideas based on the criteria, checking the tone of the comment, and applying lenses to filter comments. The interaction data also shows that all but one group spent at least 36 percent of their time using the idea-centric view. This usage was balanced toward the latter stages of the discussion when there was more content. One interesting pattern from Table 7.6 is that G1 and G2 used the features with the same relative proportions as G3 and G4, but less overall. One explanation is because G1 and G2 used the status quo interface first and, once assigned to use Procid the next week, did not discover its features as quickly as those who used it first.

Opportunities for Improvement

To improve Procid, participants identified wanting better support for comments that contain multiple names or criteria (n=5), better awareness and centrality of the provided features (n=3), and the ability to define an aggregate score for the proposed names. Future work should also address unexpected behaviors observed in the study. For example, one behavior identified by the interviewees was participants positively rating their own ideas, which obviously did not carry much weight. For example, I10 said:

	G1	G2	G3	G4
Rate ideas based on criteria	24	1	18	26
Add a new criteria	3	3	4	4
Check a comment's tone	5	5	26	18
Register strong opinion	0	0	1	4
Drop an idea	0	0	2	4
Tag comment as mustread	3	0	3	7
Click on the mustread lens	10	5	25	24
Tag comment as idea	16	16	11	10
Click on the idea lens	11	10	41	32

Table 7.6: Frequency of actions performed using Procid.

> I saw there was like three pluses for somebody and I was like oh somebody really likes this idea, and I looked and it was their idea. So, if you are doing such a thing, like three pluses is really good when we were doing it, that might confuse people that plus feature kinds of need some sifting, you need something to be able to turn that off so you can see what other people think of other people's ideas.

Another area for future improvements would be handling the implicit criteria or the criteria that are hard to articulate. The interviewees mentioned that in some cases it is hard to define objective and explicit criteria to evaluate ideas. For example, I7 said:

> As people would offer a new name, you realize that it meets all of the current criteria but it feels really bad, it sounds bad, it doesn't roll off your tongue well, and it's hard to make a criteria for roll of your tongue, because it doesn't really make sense or it feels like it doesn't make sense.

One possible way of dealing with this issue would be to add an "overall preference" criterion to partially compensate for the lack of support for implicit criteria.

7.3 Discussion

From both studies, I found support for the six consensus strategies implemented in Procid (See section 5.1). There was strong support for *S4 (evaluate and decide according to criteria)* and *S5 (maintain visual summary of agreement and disagreement).* The relevant features in Procid were frequently used (e.g. the idea-centric view and criteria interactions) and reported as a distinct advantage of the tool, and the ratings of Procid for the associated tasks were higher than the status quo interface. There was support for *S1 (promote perception of valued contribution)* as participants rated Procid higher for tracking each others contributions (ideas, criteria, comments). From Study 1, I found support for *S3 (build on relationships and seek expert advice)*, as community members were able to use the invitation feature of Procid to locate at least one member to include in a discussion who may not have been otherwise considered. *S6 (mark and revisit influential points)* was supported as the lenses were frequently applied and Procid was rated higher for filtering key comments. Finally, little support was found for *S2 (express concerns in a constructive manner).* The 'tone' dialog in the tool was accessed, but the comments did not contain an overly negative tone to begin with and therefore the comments were not revised. Despite promising results, these studies focused on perceptions of and experiences with the consensus-based features in Procid. Additional studies are needed to assess how these features are used to shape discussions and how their patterns of use may change over time, with larger groups or with different design problems.

The two studies conducted reported on measures that could be immediately collected (e.g. survey and interactions data). However, future studies should also consider measuring the longer-term impact of a tool such as Procid on a community. For example, longer-term measures could include changes in membership duration and turnover, changes in patterns of community discourse, and other community health metrics [56]. Another possibility is to create an event timeline (similar to Figure 7.5) for each discussion to see the possible longer-term effects of the tool on the discussion patterns. Future work may even extend Procid to automatically generate and configure such visualizations. As one goal of consensus building is to improve the social process of decision making, favorable outcomes may be found for these measures long term.

Chapter 8

Conclusions and Future Work

8.1 Summary of Contributions

Consensus building is a critical component of distributed design discussions as it promotes a better product and a stronger community. However, the discussion platforms commonly used do not provide mechanisms for building consensus. This thesis has made three contributions to close this gap.

First, I studied consensus building in UI design discussions from an established OSS community using qualitative and quantitative methods. From this study I reported user perspectives on the challenges of reaching consensus in UI design discussions, the techniques utilized for addressing the challenges, and the consequences of not reaching consensus. I also analyzed how various metrics related to the content, process, and user relationships of the discussions correlate with reaching consensus. The main result from this analysis shows that discussions having participants with more experience and prior interaction history are more likely to reach consensus. This result indicates that who participates in the discussion is at least as important than the quantity of discussion participants or proposed design alternatives. This has important implications for studies on computer mediated collaboration and in particular decision support systems. For example, while the majority of prior studies examined the effect of group size, task, and anonymity on decision making performance in lab settings, my study shows that expertise and social connections of participants can have a significant effect on reaching consensus and on the effectiveness of the collaboration. This invites future studies on decision support systems in authentic settings where the effect of social factors can be considered in addition to task and process related factors.

Second, I built Procid, a novel browser extension that enables consensus building strategies to be realized in un-moderated distributed design discus-

sions. Key features of the tool include (i) the ability to visually track idea proposals and organize the discussion around ideas; (ii) the ability to define persistent criteria throughout the discussion and rate the ideas against the criteria; (iii) the ability to register and visualize strong support for or against ideas, empowering individuals with a stronger voice in the discussion; (iv) interaction constraints that promote best practices of consensus efforts, e.g., encouraging the first comment on an idea to be supportive of it; and (v) the ability to identify candidates to invite to active discussions using attributes important for consensus in the domain. Procid was inspired by and targets distributed UI design discussions. However, the design of the system is not tied to the type or content of the discussion. It therefore should generalize to any distributed discussion where multiple stakeholders are collaborating to solve an ill-structured problem. If needed, the consensus strategies that informed the design of Procid can be used to explore alternative UI mechanisms for bringing the benefits of consensus building to other peer production and online deliberation platform. Procid's implementation strategies can also inform the design of future online discussion and deliberation platforms. For example, Procid was built as an add-on to an existing discussion platform rather than a standalone application. This approach imposed some constraints on the design of Procid both in terms of appearance and the amount of input required from the users. However, it had several benefits such as ease of deployment and testing in an existing large community, recruiting participants who care about the discussion, and leveraging the community's social structure to build novel features.

Finally, I reported results from two evaluations of Procid. For the first, I rendered five active discussions in the community in Procid and invited participants from those discussions to use and provide feedback on the features of the tool and the overall approach. The second evaluation compared how the use of Procid affects a distributed design discussion relative to the existing platform used in the community. Results of both studies showed that users found the features of Procid beneficial for consensus and perceived the tool as more effective for consensus building than the existing platform. Together, these two studies both captured community member's perceptions of Procid in authentic settings and assessed the effectiveness of Procid in comparison to an existing discussion platform. However, future longer term studies are needed to find out how a tool like Procid changes community's discourse or

affects patterns of behavior in the community.

8.2 Future Work

One immediate direction for future work is to conduct a similar mixed methods study on other types of discussions (e.g. performance and security discussions) to shed more light on the differences between UI design and other types of discussions for consensus building. Similar analysis should also be performed on UI design discussions in other distributed software projects to assess the generalizability of the results. The future studies can also include additional metrics in the regression analysis such as the language complexity of the messages, number of arguments for or against design proposals, the sentiment of those arguments, use of rhetorical devices, and advanced techniques for assessing expertise.

Another direction is to integrate more consensus building strategies into Procid and compare how the addition of those strategies affects the discussion experience and outcomes. Consensus building involves a broad set of procedural, rhetorical, and social strategies. Procid prioritizes a subset of these strategies that is feasible to implement and would be difficult to realize without tool support or a moderator (e.g. organizing a lengthy discussion around the ideas). Additional strategies can be implemented in the tool. In time, the behaviors encouraged by these strategies in Procid could become integrated into the practice of the community. In addition, for practices that cannot be easily integrated into the tool, consensus building training modules can be designed as part of the community's onboarding process [19].

Finally, one can conduct additional longer studies probing how the consensus-based features of Procid are used to shape discussions and how their patterns of use may change over time, with larger groups or with different design problems. Future studies should also consider measuring the longer-term impact of Procid on a community. For example, longer-term measures could include changes in membership duration and turnover, changes in patterns of community discourse, and other community health metrics [56]. These metrics can be integrated into a visual analytics dashboard for online communities that captures and presents the current status of a community at any given point in time.

References

[1] Adamic, L. A., Zhang, J., Bakshy, E., and Ackerman, M. S. Knowledge sharing and yahoo answers: Everyone knows something. In *Proceedings of the 17th International Conference on World Wide Web* (New York, NY, USA, 2008), WWW '08, ACM, pp. 665–674.

[2] Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining* (New York, NY, USA, 2008), WSDM '08, ACM, pp. 183–194.

[3] AlchemyAPI. Sentiment analysis. http://www.alchemyapi.com.

[4] Alonso, S., Herrera-Viedma, E., Cabrerizo, F. J., Porcel, C., and Lopez-Herrera., A. G. Using visualization tools to guide consensus in group decision making. In *WILF* (2007), pp. 77 – 85.

[5] Angwin, J., and Fowler, G. Volunteers log off as wikipedia ages. *The Wall Street Journal.* (Nov. 23 2009), A.1.

[6] Applegate, L. M., Konsynski, B. R., and Nunamaker, J. F. A group decision support system for idea generation and issue analysis in organization planning. In *CSCW* (1986), pp. 16–34.

[7] Avery, M., Auvine, B., Streibel, B., and Weiss, L. *Building united judgment: a handbook for consensus decision making.* The Center for Conflict Resolution, Madision, Wis., 1981.

[8] Baccianella, S., Esuli, A., and Sebastiani, F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC* (2010), pp. 2200–2204.

[9] Bach, P. M., DeLine, R., and Carroll, J. M. Designers wanted: Participation and the user experience in open source software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), CHI '09, ACM, pp. 985–994.

[10] Baddeley, A. Working memory. *Science 255*, 5044 (1992), 556–559.

[11] Balakrishnan, A., Fussell, S., and Kiesler, S. Do visualizations improve synchronous remote collaboration? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), CHI '08, pp. 1227–1236.

[12] Baltes, B., Dickson, M., Sherman, M., Bauer, C., and LaGanke, J. Computer-mediated communication and group decision making: A meta-analysis. *Organizational Behavior and Human Decision Processes 87*, 1 (2002), 156–179.

[13] Barcellini, F., Dtienne, F., Burkhardt, J.-M., and Sack, W. A study of online discussions in an open-source software community. In *Communities and Technologies 2005*, P. Besselaar, G. Michelis, J. Preece, and C. Simone, Eds. Springer Netherlands, 2005, pp. 301–320.

[14] Biehl, J. T., Baker, W. T., Bailey, B. P., Tan, D. S., Inkpen, K. M., and Czerwinski, M. Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), CHI '08, pp. 939–948.

[15] Birnholtz, J. P., Grossman, T., Mak, C., and Balakrishnan, R. An exploratory study of input configuration and group process in a negotiation task using a large display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2007), CHI '07, pp. 91–100.

[16] Briggs, R. O., Kolfschoten, G. L., and d., V. G.-J. Toward a theoretical model of consensus building. In *Americas Conference on Information Systems* (2005).

[17] Burke, M., and Kraut, R. Mopping up: Modeling wikipedia promotion decisions. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work* (New York, NY, USA, 2008), CSCW '08, ACM, pp. 27–36.

[18] Buskens, V. *Social Networks and Trust (Theory and Decision Library C)*. Kluwer Academic Publishers, Norwell, MA, 2002.

[19] Choi, B., Alexander, K., Kraut, R. E., and Levine, J. M. Socialization tactics in wikipedia and their effects. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work* (2010), CSCW '10, ACM, pp. 107–116.

[20] Clawson, V. K., Bostrom, R. P., and Anson, R. The role of the facilitator in computer-supported meetings. *Small Group Research 24*, 4 (1993), 547–565.

[21] Conklin, J., and Begeman, M. L. glbis : A hypertext tool for exploratory policy discussion. *ACM Transactions of Office Information Systems 6*, 4 (1988), 303–331.

[22] Cosley, D., Frankowski, D., Terveen, L., and Riedl, J. Suggestbot: Using intelligent task routing to help people find work in wikipedia. In *Proceedings of the 12th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2007), IUI '07, ACM, pp. 32–41.

[23] DeSanctis, G., and Gallupe, R. B. A foundation for the study of group decision support systems. *Management Science 33*, 5 (1987), 589 – 609.

[24] Deshpande, A., and Riehle, D. The total growth of open source. In *Open Source Development, Communities and Quality*, B. Russo, E. Damiani, S. Hissam, B. Lundell, and G. Succi, Eds., vol. 275 of *IFIP The International Federation for Information Processing*. Springer US, 2008, pp. 197–209.

[25] El-Gayar, O. F., and Fritz., B. D. A web-based multi-perspective decision support system for information security planning. *Decision Support Systems 50*, 1 (2010), 43–54.

[26] Erickson, T., Kellogg, W. A., Laff, M., Sussman, J., Wolf, T. V., Halverson, C. A., and Edwards, D. A persistent chat space for work groups: The design, evaluation and deployment of loops. In *Proceedings of the 6th Conference on Designing Interactive Systems* (New York, NY, USA, 2006), DIS '06, ACM, pp. 331–340.

[27] Faridani, S., Bitton, E., Ryokai, K., and Goldberg, K. Opinion space: a scalable tool for browsing online comments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), CHI '10, pp. 1175–1184.

[28] Farnham, S., Chesley, H. R., McGhee, D. E., Kawal, R., and Landau, J. Structured online interactions: Improving the decision-making of small discussion groups. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (New York, NY, USA, 2000), CSCW '00, ACM, pp. 299–308.

[29] Fielding, R. T. Shared leadership in the apache project. *Commun. ACM 42*, 4 (Apr. 1999), 42–43.

[30] Fiol, C. Consensus, diversity, and learning in organization. *Organization Science 5*, 3 (1994), 403–420.

[31] Friedman, R., Olekalns, M., Anderson, C., Brett, J., Goates, N., and Lisco, C. C. The positive and negative effects of anger on dispute resolution: Evidence from electronically mediated disputes. *Applied Psychology 89*, 2 (2004), 369–376.

[32] Gallupe, R. B., and McKeen, J. D. Enhancing computer-mediated communication: An experimental investigation into the use of a group decision support system for face-to-face versus remote meetings. *Inf. Manage. 18*, 1 (Jan. 1990), 1–13.

[33] Grudin, J. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM 37*, 1 (1994), 92–105.

[34] Guzzo, R., and Waters, J. The expression of affect and the performance of decision maknig groups. *Journal of Applied Psychology 67*, 1 (1982), 67–74.

[35] Hall, J., and Williams, M. A comparison of decision-making performances in established and ad hoc groups. *Journal of Personality and Social Psychology 3*, 2 (1966), 214–222.

[36] Hare, A. Study of interaction and consensus in different sized groups. *American Sociological Review 17*, 3 (1952), 261–267.

[37] Hars, A., and Ou, S. Working for free? motivations for participating in open-source projects. *International Journal of Electronic Commerce 6*, 3 (2002), 25–39.

[38] Healey, P. *Collaborative planning : shaping places in fragmented societies*. Macmillan, Houndmills, Basingstoke, Hampshire RG21 6XS, and London, 1997.

[39] Heckman, R., Crowston, K., Eseryel, Y. U., Howison, J., Allen, E., and Li, Q. Emergent decision-making practices in free/libre open source software (floss) development teams. In *IFIP International Conference on Open Source Software* (2007).

[40] Hiltz, S. R., Johnson, K., and Turoff, M. Experiments in group decision making communication process and outcome in face-to-face versus computerized conferences. *Human Communication Research 13*, 2 (1986), 225–252.

[41] Horowitz, D., and Kamvar, S. D. The anatomy of a large-scale social search engine. In *WWW* (2010), pp. 431–440.

[42] Janis, I. *Victims of Groupthink: A Psychological Study of Foreign-Policy Decisions and Fiascoes*. Houghton Mifflin Company, 1972.

[43] Kelley, T., and Littman, J. *The Art of Innovation: Lessons in creativity From IDEO, Americas Leading Design Firm.* Doubleday, New York, NY, 2001.

[44] Kerr, N. L., and Tindale, R. S. Group performance and decision making. *Annual Review Psychology 55* (2004), 623–655.

[45] Ko, A., and Chilana, P. Design, discussion, and dissent in open bug reports. In *iConference* (2011), pp. 106–113.

[46] Kossinets, G., and Watts, D. J. Empirical analysis of an evolving social network. *Science 311*, 5757 (2006), 88–90.

[47] Kriplean, T., Beschastnikh, I., McDonald, D. W., and Golder., S. A. Community, consensus, coercion, control: cs*w or how policy mediates mass participation. In *GROUP* (2007), pp. 167–176.

[48] Kriplean, T., Morgan, J., Freelon, D., Borning, A., and Bennett., L. Supporting reflective public thought with considerit. In *CSCW* (2012), pp. 265–274.

[49] Kriplean, T., Toomim, M., Morgan, J., Borning, A., and Ko, A. Is this what you meant?: promoting listening on the web with reflect. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), CHI '12, pp. 1559–1568.

[50] Kumar, A., and Paddison, R. Trust and collaborative planning theory: The case of the scottish planning system. *International Planning Studies 5*, 2 (2000), 205–223.

[51] Lam, S. K., Karim, J., and Riedl, J. The effects of group composition on decision quality in a social production community. In *GROUP* (2010), pp. 55–64.

[52] Lam, S. S. K. The effects of group decision support systems and task structures on group communication and decision quality. *J. Manage. Inf. Syst. 13*, 4 (Mar. 1997), 193–215.

[53] Lemus, D. R., Seibold, D. R., Flanagin, A. J., and Metzger, M. J. Argument and decision making in computer-mediated groups. *Journal of Communication 54*, 2 (2004), 302–320.

[54] LIWC. Linguistic inquiry and word count. http://www.liwc.net.

[55] Luther, K., Counts, S., Stecher, K. B., Hoff, A., and Johns, P. Pathfinder: an online collaboration environment for citizen scientists. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), CHI '09, pp. 239–248.

[56] Matthews, T., Whittaker, S., Badenes, H., Smith, B. A., Muller, M., Ehrlich, K., Zhou, M. X., and Lau, T. Community insights: Helping community leaders enhance the value of enterprise online communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), CHI '13, ACM, pp. 513–522.

[57] McGuire, T., Kiesler, S., and Siegel, J. Group and computer-mediated discussion effects in risk decision making. *Journal of Personality and Social Psychology 52*, 5 (1987), 917–930.

[58] Muller, M. J. Retrospective on a year of participatory design using the pictive technique. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1992), CHI '92, ACM, pp. 455–462.

[59] Oehlberg, L., Simm, K., Jones, J., Agogino, A. M., and Hartmann, B. Showing is sharing: building shared understanding in human-centered design teams with dazzle. In *DIS* (2012), pp. 669–678.

[60] Olson, G. M., Olson, J. S., Carter, M. R., and Storrøsten, M. Small group design meetings: An analysis of collaboration. *Hum.-Comput. Interact. 7*, 4 (Dec. 1992), 347–374.

[61] Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[62] Pioch, N. J., and Everett., J. O. Polestar: collaborative knowledge management and sensemaking tools for intelligence analysts. In *CIKM* (2006), pp. 513–521.

[63] Shum, S. J. B., Selvin, A. M., Sierhuis, M., Conklin, J., Haley, C. B., and Nuseibeh, B. *Hypermedia Support for Argumentation-Based Rationale*. 2006, pp. 111–132.

[64] Sidaway, R. Consensus building. Tech. rep., Scottish National Rural Partnership, 1998.

[65] Siegel, J., Dubrovsky, V., Kiesler, S., and McGuire, T. W. Group processes in computer-mediated communication. *Organizational Behavior and Human Decision Processes 37*, 2 (1986), 157–187.

[66] Straus, S., and McGrath, J. Does the medium matter? the interaction of task type and technology on group performance and member reactions. *Journal of Applied Psychology 79*, 1 (1994), 87–97.

[67] Strauss, A. L. *Qualitative Analysis for Social Scientists*. Cambridge University Press, Cambridge, UK, 1987.

[68] Susskind, L., McKearnan, S., and Thomas-Larmer, J. *The Consensus Building Handbook*. Sage Publications, Thousand Oaks, CA, 1999.

[69] Tegarden, D. P. Business information visualization. *Communications of the AIS 1*, 4 (1999).

[70] Terry, M., Kay, M., and Lafreniere, B. Perceptions and practices of usability in the free/open source software (foss) community. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), CHI '10, ACM, pp. 999–1008.

[71] Twidale, M. B., and Nichols, D. M. Exploring usability discussion in open source development. In *In: Proceedings of Thirty-Eighth Annual Hawaii International Conference on System Sciences, HICSS-38, Track 7. 198c* (2005), p. 198.

[72] Uzzi, B. Social Structure and Competition in Interfirm Networks: The Paradox of Embeddedness. *Administrative Science Quarterly 42*, 1 (1997), 35–67.

[73] Viégas, F. B., Wattenberg, M., and Dave, K. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2004), CHI '04, ACM, pp. 575–582.

[74] Walker, G., Kogut, B., and Shan, W. Social capital, structural holes and the formation of an industry network. *Organization Science 8*, 2 (1997), pp. 109–125.

[75] Watson, R. T., DeSanctis, G., and Poole, M. S. Using a gdss to facilitate group consensus: Some intended and unintended consequences. *MIS Q. 12*, 3 (1988), 463–478.

[76] Willett, W., Heer, J., Hellerstein, J., and Agrawala, M. Commentspace: Structured support for collaborative visual analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), CHI '11, pp. 3131–3140.

[77] Wong, F., Fernandez, G., and McGovern, J. Confer: Towards groupware for building consensus in collaborative software engineering. In *Proceedings of the Eight Australasian Conference on User Interface - Volume 64* (Darlinghurst, Australia, Australia, 2007), AUIC '07, Australian Computer Society, Inc., pp. 31–38.

[78] Yamauchi, Y., Yokozawa, M., Shinohara, T., and Ishida, T. Collaboration with lean media: How open-source software succeeds. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (New York, NY, USA, 2000), CSCW '00, ACM, pp. 329–338.

[79] Yasuda, M., Webster, S., and Webster, M. Color and facial expressions. *Journal of Vision 7*, 9 (2007), 946.

[80] Zaheer, A., and Bell, G. G. Benefiting from network position: firm capabilities, structural holes, and performance. *Strategic Management Journal 26*, 9 (2005), 809–825.

[81] Zilouchian Moghaddam, R., Bailey, B., and Fu., W.-T. Consensus building in open source user interface design discussions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), CHI '12, pp. 1491–1500.

[82] Zilouchian Moghaddam, R., Bailey, B. P., and Poon., C. Ideatracker: an interactive visualization supporting collaboration and consensus building in online interface design discussions. In *INTERACT* (2011), pp. 259–276.

[83] Zilouchian Moghaddam, R., Twidale, M., and Bongen, K. Open source interface politics: Identity, acceptance, trust, and lobbying. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (2011), CHI EA '11, ACM, pp. 1723–1728.