# Iterative Proposals in Online Decision-Making

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

**Björn Ebbinghaus**

geboren in

Remscheid

Düsseldorf, September 2025

aus dem Institut für Informatik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Martin Mauve
2. Prof. Dorothea Baumeister

Tag der mündlichen Prüfung: 12.02.2026

# Abstract

This work addresses the question of how large groups of people can use today's technologies to improve their collective decision-making by enhancing their ability to refine proposals.

Making collective decisions online is challenging when large and diverse groups of people are involved. Although traditional tools such as comment sections and polls help to gather opinions, they often fail to support the refinement and development of ideas. This dissertation examines ways in which online decision-making processes can be enhanced by systems that enable participants to collaboratively develop and evolve proposals.

First, the work presents a system in which users can build on each other's proposals, inspired by version control systems. This structure enables improvements, alternatives and coalitions to be expressed as new proposals with explicit relationships. A user study was conducted to observe the effect of different voting schemes on participation. The results showed that live feedback can influence strategic behaviour and satisfaction with outcomes.

Next, this work introduces a second system that incorporates structured mechanisms to support scalability and manage proposal complexity. Here, techniques adapted from software development allow participants to freely explore ideas while ensuring coherence and ease of navigation throughout the process. A merit-based model enables collaboration to occur without the need for central moderation.

Finally, a metric for measuring group progress during deliberation is presented. This metric can provide participants with feedback and highlights areas of convergence or divergence.

# Zusammenfassung

Diese Arbeit befasst sich mit der Frage, wie große Gruppen von Menschen die heutigen Technologien nutzen können, um ihre kollektive Entscheidungsfindung zu verbessern, indem sie ihre Fähigkeit zur Verfeinerung von Vorschlägen verbessern.

Die Online-Entscheidungsfindung stellt eine besondere Herausforderung dar, wenn große und heterogene Gruppen von Menschen beteiligt sind. Zwar helfen traditionelle Tools wie Kommentarbereiche und Umfragen dabei, Meinungen zu sammeln, doch unterstützen sie oft nicht die Verfeinerung und Weiterentwicklung von Ideen. In dieser Dissertation werden Möglichkeiten untersucht, wie Online-Entscheidungsprozesse durch Systeme verbessert werden können, die es den Teilnehmern ermöglichen, Vorschläge gemeinsam zu entwickeln und weiterzuentwickeln.

Zunächst wird ein System vorgestellt, in dem Nutzer auf den Vorschlägen anderer aufbauen können. Es ist von Versionskontrollsystemen inspiriert. Diese Struktur ermöglicht es, Verbesserungen, Alternativen und Koalitionen als neue Vorschläge mit expliziten Beziehungen auszudrücken. In einer Nutzerstudie wurde der Einfluss verschiedener Abstimmungsschemata auf die Beteiligung untersucht. Die Ergebnisse zeigten, dass Live-Feedback das strategische Verhalten und die Zufriedenheit mit den Ergebnissen beeinflussen kann.

Anschließend wird ein zweites System präsentiert, das strukturierte Mechanismen zur Unterstützung der Skalierbarkeit und Verwaltung der Komplexität von Vorschlägen beinhaltet. Hier ermöglichen Techniken, die aus der Softwareentwicklung adaptiert wurden, den Teilnehmern, Ideen frei zu erkunden, während gleichzeitig die Kohärenz und eine einfache Navigation während des gesamten Prozesses gewährleistet werden. Ein leistungsorientiertes Modell ermöglicht die Zusammenarbeit, ohne dass eine zentrale Moderation notwendig ist.

Abschließend wird eine Metrik zur Messung des Gruppenfortschritts während der Beratung präsentiert. Mithilfe dieser Metrik können die Teilnehmer Feedback erhalten und Bereiche der Konvergenz oder Divergenz identifizieren.

# Acknowledgements

This would not have been possible without the help of many people. I could not have done it without them. I am also aware of the privileges I have been given, and I pledge to use them responsibly to shape my environment.

Firstly, I would like to express my deepest gratitude to Martin Mauve for this opportunity, as well as for his continued guidance and support. Without the structure, focus and direction that our regular meetings provided, this work would not have been possible.

I am also very grateful to Christina van Arkel for her unwavering support, which gave me the strength and stability I needed to complete this dissertation.

I thank Christian Meter for his invaluable personal, academic and professional advice, both as a friend and as a mentor. He has arguably had the greatest influence on my academic development.

Special thanks go to Markus Brenneis, who embarked on this journey with me, for his patience and for engaging with my crazy ideas. Thanks also to Janine Golov, who survived the hordes of students with me. I would also like to thank Thomas Spitzlei and Sabine Freese for their administrative efforts which made the work of the department possible, as well as Tobias Schröder and Alexander Schneider for facilitating my start in the research group. Thanks also to my colleagues and office partners, Jan Steimann and Marc Feger. I acknowledge the Forschungskolleg Online-Partizipation for its financial support.

Finally, I wish to thank my friends and family for their encouragement and understanding throughout this time.

# Contents


**1 Introduction** **1**
- 1.1 Research Questions . . . . 2
- 1.2 Contributions . . . . 3
- 1.3 Outline of the Thesis . . . . 4

**2 Related Work** **5**
- 2.1 Tools for Online Deliberation . . . . 5
- 2.2 Computer-Supported Cooperative Work using Version Control Systems . . . . 9

**3 Preliminaries** **12**
- 3.1 Formal Argumentation with D-BAS . . . . 12
- 3.2 Decision Making with Argumentation Graphs . . . . 13

**4 Better Decision-Making through Collaborative Development of Proposals** **17**
- 4.1 Contributions & Results . . . . 18
- 4.2 Impact on the Thesis . . . . 19

**5 Version Control for Structuring Collaborative Decision-Making** **33**
- 5.1 Contributions . . . . 34
- 5.2 Impact on the Thesis & Future Work . . . . 35

**6 Measuring Progress to Support Cooperative Decision-Making Processes** **45**
- 6.1 Contribution . . . . 46

**7 Discussion** **60**
- 7.1 Proposal Proliferation . . . . 60
- 7.2 Complex Preferences . . . . 62
- 7.3 Negotiation / Compromises . . . . 63

**8 Conclusion** **64**
- 8.1 Main Results . . . . 64
- 8.2 Future Work . . . . 65

**Bibliography** **67**

# Chapter 1

# Introduction

Although the internet has made it easier than ever for people to connect, it has also brought its own set of challenges. The relatively low barrier to connectivity enables unimaginable numbers of people to communicate in unprecedented ways. The challenge now is to communicate efficiently. It is not enough for people to be connected; they must also be interconnected.

Making decisions in groups has always been a complex task, and the desire to include more voices only adds to this complexity. However, it is essential to involve a wider range of participants if decisions are to be fair, representative, and accepted by those affected by them. The resulting diversity of opinion and experience can lead to better outcomes—but only if it is managed well.

Online group decision-making processes aim to meet this challenge by enabling large numbers of people to contribute, debate and reach shared conclusions. However, the tools commonly used in these processes—comment sections, voting interfaces and polls—often fall short. While they can collect input, they do not necessarily facilitate understanding between participants, resolve disagreements or achieve consensus.

In order to facilitate meaningful group decision-making, systems are required that not only collect opinions, but also help to shape them. These systems must provide participants with ways to engage with each other's ideas and refine and combine proposals without becoming overwhelmed. At the same time, they must be transparent and intuitive enough to give participants a sense of control and make them trust the results.

This thesis explores how decision-making systems can be designed to support these goals. It examines how such systems can guide users without dominating the process, and how design choices—such as the presentation of proposals or the provision of feedback—can influence both the quality of outcomes and user satisfaction. Based on real participation processes and the development of new tools, this work aims to identify the factors that facilitate effective online collaboration and propose scalable solutions for real-world systems involving numerous participants.

## 1.1 Research Questions

**How can online decision-making and deliberation processes be improved?**
The overarching theme of this thesis. Today's technology offers unprecedented speed, scale, and opportunities for participation in online processes. However, while this can be empowering, it is crucial that these processes are accessible to all; otherwise, there is a risk of creating or widening the digital divide (NTIA, 1999). Conversely, we are still in the early stages of understanding how an always-connected world can influence society.

**How can iterative proposal development be enabled or supported?**
This is the specific focus of this work. Decisions are only as good as the alternatives from which they are made. However, current systems often treat proposals as static inputs rather than evolving ideas. Encouraging participants to collaborate in developing and refining proposals can lead to better outcomes and foster a stronger sense of ownership of the results (Lachapelle, 2008).

**What design requirements are there for iterative decision-making systems?**
How can a system be both open and reliable? How can guidance be provided without limiting creativity or participation? Key design questions include ensuring transparency, fairness, accountability and participant autonomy without introducing unnecessary complexity.

**How can iterative decision-making systems be kept manageable and scalable?**
How can participation be kept productive when many users are involved? What can be done to prevent decision-making system from becoming overwhelming or inefficient with larger numbers of participants? This work explores how structural mechanisms, such as organizing contributions, managing conflict, and reducing redundancy, can help maintain usability as participation increases.

## 1.2 Contributions

This work's central contribution is the development of systems that support structured, iterative decision-making in online participation processes. Rather than limiting participants to static voting or open-ended commenting, the focus is on enabling them to actively iterate and refine proposals over time.

To this end, a system was designed, implemented and evaluated that enabled proposals to be derived from each other. The evaluation included two trials with paid participants under slightly different conditions. One of the lessons learned was that participants should be given enough freedom to experiment with new proposals, while also ensuring that the process is not too lightly regulated, as this can lead to the system quickly becoming overwhelmed with short-term contributions. Participants should be given as many options as possible, but the resulting complexity should be hidden as much as possible without active moderation or censorship.

Based on experience gained from practical user testing of the first application, a second system was designed. This system adapted well-known *Version Control System* (VCS) techniques in a novel way to form the basis of scalable deliberation processes. Improvements over the first system mainly concerned the ability to provide users with a powerful yet user-friendly mechanism for managing collaborative work on proposals. The tools provided can function in the same way as traditional deliberation processes in the base case, with incremental enhancements for users willing and able to commit to the process. This new system addresses the need for regulation through a meritocratic system, whereby other users always have the option of creating their own proposal to avoid being locked in.

A further major contribution is a metric that measures the progress of the decision-making process. This provides participants with immediate feedback and guidance, indicating whether the group is moving towards or away from consensus. The metric also enables the system to suggest potential next steps or highlight areas where collaboration could be beneficial. To this end, a number of desirable characteristics were developed and rationalized, and it was explained why these characteristics should influence the score. This was deemed necessary to ensure that the solution reflects the intuition of participants in the decision-making process.

## 1.3 Outline of the Thesis

The next two chapters present related work. The first of these, Chapter 2, provides a general overview of the subject area, including related software and techniques. This is followed by a presentation of work closely related to this thesis in Chapter 3, which leads directly into its developments.

After that, Chapters 4 and 5 are about the development and exploration of software applications to support decision-making and online deliberation scenarios. Chapter 4 is the first publication and presents the initial prototype of an iterative proposal system. This was developed in response to an observation made during the previous participation process (Section 3.2), namely that, while participants often build on each other's ideas, improved versions of proposals tend to get lost. The system enables participants to generate new proposals explicitly from existing ones.

The work in Chapter 5 is a direct continuation of the work in Chapter 4. It introduces a new system that addresses the original application's scalability problem. The new system uses techniques from VCSs to keep the number of proposals manageable.

Chapter 6 presents a metric for quantifying the coherence in parties of participants. This is intended as a tool to measure progress towards consensus and to guide participants in forming effective coalitions.

Chapter 7 discusses key design choices and issues, structured around themes shaped by the work's objectives and constraints.

Lastly, Chapter 8 draws a conclusion and provides suggestions for future work.

# Chapter 2
# Related Work

This chapter provides an overview of the related work that forms the basis of this dissertation. It is organized into two main sections. Online Deliberation tools and *Computer-Supported Cooperative Work* (CSCW) in the context of *Version Control System* (VCS).

## 2.1 Tools for Online Deliberation

Online deliberation is the use of digital platforms to facilitate organized discussion, decision-making, and group problem-solving. It is commonly employed in politics, policy formulation, civic engagement and workplace decision-making. Unlike casual online chat, online deliberation focuses on logical arguments and fact-based discussions. Many platforms include moderation, structured formats, and voting features to improve the quality of discussions (Davies and Gangadharan, 2009).

With the growing reach of the web and the increasing digitization of government functions around the world (United Nations Department of Economic and Social Affairs, 2024), there is a growing interest in using the web to improve democracy by involving citizens more directly. Groups often make better decisions than individuals (Surowiecki, 2004). However, coordinating these groups to make these decisions can be quite difficult, especially when the web allows these groups to be physically and temporally separated. Online deliberation has been successfully applied in participatory budgeting, citizen assemblies, and collaborative policymaking.

### 2.1.1 LiquidFeedback

LiquidFeedback[1] (Interaktive Demokratie e.V, 2014; Kling et al., 2015) is an open-source, deliberative democracy tool designed to facilitate collective decision-making within organizations, political parties, and communities. It provides a platform where participants can propose, discuss, and vote on issues across four phases:

**Admission Phase** The first phase is used to gauge interest in the issue at hand. At least one proposed *alternative initiative*—a proposed solution to the issue—must have a quorum for the issue to proceed; otherwise, it will be closed.

**Discussion Phase** The alternative initiative is discussed in the second phase. During this phase, initiators of these initiatives can make changes to them. Supporters of an initiative can suggest changes to it. Notably, these suggestions can be used as a conditional support for the initiative, where the supporters can mark suggestions as mandatory or undesirable to be implemented in order for the support to count.

**Verification Phase** During the penultimate phase, initiatives are frozen and must achieve a quorum to proceed to the final voting phase. This ensures that only relevant initiatives are presented to the other participants, who can then vote on them.

**Voting Phase** In the final phase, everyone in the system can vote for or against the remaining initiatives using a preferential voting system.

The second and third phases are very similar to the aims of two of the papers presented in Chapters 4 and 5. In the second phase, there is also an emphasis on continuing to work on the proposals rather than merely discussing them. The procedure by which the initiator of an initiative incorporates suggestions for improvement is similar to that presented in Chapter 5, except that in LiquidFeedback, participants only contribute short ideas, whereas in Chapter 5, participants propose a new version of the aforementioned initiative, which can in principle be regarded as their own initiative.

LiquidFeedback also places a strong emphasis on its voting scheme: liquid democracy (Paulin, 2020). This is a system of delegated voting, whereas the voting phase in this work received less attention.

[1] https://liquidfeedback.com/

### 2.1.2 Loomio

Loomio[2] is also an open-source decision-making platform for groups. It is aimed more at smaller groups, such as councils or working groups. At its heart is a traditional forum-like discussion format organized into *threads*. In addition to this discussion environment, participants can create *proposals*, which represent a decision-making instance. These proposals can be used with different templates, which mainly differ in terms of the response options available to participants.

Unlike LiquidFeedback and the proposals of this work, decisions are not the central focus of Loomio, nor are they organized into processes aimed at decision-making. In Loomio, decisions are based on alternating discussion and feedback/decision stages in the form of proposals, which are repeated until the desired group outcome is achieved.

### 2.1.3 Decidim

Decidim[3] is a participatory tool designed for large-scale policy deliberations. Like many similar tools, it consists of modules that can be activated or deactivated according to the organizer's requirements.

The *Proposals* module offers several methods for supporting collaborative work on proposals, which can be used simultaneously.

**Amendments** Amendments allow participants to request changes to an existing proposal. Depending on the settings, these requests are either public or only visible to the author of the original proposal, who must review and accept them.

**Collaborative Drafts** This feature enables participants to request editing access to a new draft proposal, which can then be refined by a small group. All participants are eventually listed as authors. As the name suggests, this feature is intended for use before the proposal is published.

**Participatory Texts** When using *Participatory Texts*, the entire process revolves around a shared text, typically a legislative document, with sections that can be edited independently. In this mode, individual sentences or paragraphs function as standalone proposals.

[2] https://www.loomio.com/
[3] https://decidim.org/

Although these options are designed to improve proposals, participants have little support from the system's mechanics and there is a lot of external moderation control, the organizer of the participation process must explicitly enable these features.

### 2.1.4 Deliberatorium

Deliberatorium (Klein, 2007, 2011; Klein and Iandoli, 2008) is an application of argumentation systems for deliberation in large-scale scenarios. It uses structured arguments to break complex issues down into smaller parts that can be solved individually. Participants can suggest ways to address the main issue, which can then be debated for and against. Most importantly, new issues can be added specifically to that proposal, which can then be addressed with its own proposals and arguments. An example of this is shown in Figure 2.1.

Deliberatorium aims to solve issues through pure deliberation, rather than 'mechanical' help, such as voting. Its goal is similar to that of Chapters 4 and 5 which explore and iterate on many proposals to improve the outcome of the process.

- **Issue** How should we fight the climate crisis?
  - **Proposal** By reducing our $CO_2$ emissions.
    - **Pro** Reducing $CO_2$ is a great first step.
    - **Issue** How do we reduce our $CO_2$ emissions?
      - **Proposal** By reducing the number of flights.
        - **Con** Flights are a fast and convenient way of travel.

Figure 2.1: Example of an argumentation that gets broken down in the Deliberatorium.

### 2.1.5 Pol.is

Pol.is[4] (Megill, 2016; Small et al., 2021) is an online platform used to collect and analyse short statements from participants. Users can vote statements up or down, but cannot respond directly to others. This structure is intended to keep discussions focused on individual contributions rather than back-and-forth replies.

[4] https://pol.is

The platform groups participants based on similar voting patterns. These groupings are visualized and used to highlight areas of agreement and disagreement within the participant pool. Unlike tools such as LiquidFeedback or Loomio, Pol.is does not structure the process around phases or editable proposals. Its focus is on mapping opinions rather than guiding participants through a decision-making workflow.

Pol.is has been used in various civic and governmental contexts, including in Taiwan, to collect input on public policy topics. The platform's features are designed to support large-scale participation and quantitative analysis of group sentiment.

## 2.2 Computer-Supported Cooperative Work using Version Control Systems

VCSs are the de facto way of collaborating on software development. They manage different versions of a file throughout its life cycle. The entire change history is retained and navigable. This makes it possible to revert to an earlier version at any time.

**Git**[5] is a distributed Version Control System designed to help teams collaborate on projects by tracking changes and managing different versions of files. Unlike centralized systems, Git allows each team member to have their own local copy of the project, enabling them to work independently and subsequently merge changes. This approach helps avoid conflicts and ensures that everyone's contributions are preserved. GitHub[6], built on top of Git, adds a layer of collaboration by providing a platform for hosting and sharing Git repositories. It provides project management features such as pull requests, issue tracking and project boards to help teams communicate and organize their work. GitHub's interface and collaborative tools facilitate code review, improvement suggestions and consensus building on changes, fostering a collaborative environment for small and large teams working on complex projects.

**Collaborative Text Editors** These editors allow multiple users to write and revise content together, often in real time. They combine live editing features with mechanisms to review and

[5] https://git-scm.com/
[6] https://github.com

restore earlier versions of documents. Google Docs[7] supports real-time editing and automatically tracks all changes, offering the option to name and restore previous document states. Overleaf[8], a collaborative LaTeX editor, builds on Git-like versioning principles, enabling researchers to collaborate in real time while maintaining a detailed revision history of the document. Etherpad[9] is an open-source editor that enables real-time collaboration and includes a built-in revision history, allowing users to easily review and navigate past edits.

**Design and Visual Collaboration Tools** Design and visual collaboration platforms enable teams to work together on graphical content. These tools often include versioning features to facilitate experimentation, review, and coordination. Figma[10] is a widely used design platform that allows multiple users to edit shared design files concurrently. It supports branching, enabling users to explore design changes in isolated versions before merging them back into the main file. Miro[11] provides a collaborative whiteboard environment tailored for brainstorming and planning activities. Its board history features allow teams to revisit earlier stages of their work. Excalidraw[12] offers a lightweight collaborative whiteboard experience focused on sketching diagrams and informal visuals with support for real-time editing and version history.

**Project Management Tools** These tools facilitate structured collaboration by organizing tasks, tracking progress and maintaining change histories. Their versioning mechanisms help teams to coordinate effectively and reflect on how their processes evolve. Jira[13] maintains detailed issue histories and logs changes to tasks, priorities, and assignments, helping teams to coordinate their work and review project progress collaboratively. Trello[14] facilitates collaboration through a shared task board that records changes to cards and workflows, enabling users to track the evolution of projects over time.

Making version control more accessible to non-technical users is an ongoing challenge. Potential improvements include features such as automatic conflict resolution, AI-assisted change tracking and context-aware versioning. The goal is to refine Version Control Systems to fit more real-time

[7] https://docs.google.com/
[8] https://www.overleaf.com/
[9] https://etherpad.org/
[10] https://figma.com/
[11] https://miro.com/
[12] https://excalidraw.com/
[13] https://www.atlassian.com/software/jira
[14] https://trello.com/

collaboration models more effectively, while retaining the benefits of a structured history and rollback capabilities.

In addition to sequential versioning, many systems support simultaneous work on multiple versions in parallel, as well as providing tools for merging these versions. This enables efficient collaboration on a large scale. There is ongoing research into how advanced features can be utilized more extensively in collaborative work, such as writing (McKelvey et al., 2023; Weidner, 2023) or research, like Litt et al. did in their work on Patchwork.

**Patchwork** Patchwork (Litt et al., 2024) is a recent research project that explores the general application of Version Control Systems in everyday software, such as text editors and drawing applications. Particular emphasis is placed on providing an easy user experience and enabling collaborative use of the features. Context-based grouping of changes is introduced to simplify the edit history by collapsing edits based on information such as author, time, or manually defined milestones.

The authors also present an initial proposal for a commenting system that uses the time and location of an edit to anchor comments. This allows the comment to be displayed only in later versions that contain the relevant text. The challenge of discussing potentially different versions of the same entity is discussed in more detail in Chapter 5.

# Chapter 3

# Preliminaries

This chapter provides an overview of the background and motivation behind the topic of this thesis. It stems from a history of research aimed at developing software systems that enable large-scale, asynchronous communication, particularly in the context of structured discussions and collaborative decision-making. While it does not attempt to cover the entire field, the chapter only outlines the progression of ideas and systems that have directly influenced the design decisions and research questions explored in this work.

## 3.1 Formal Argumentation with D-BAS

Argumentation systems structure discussions by structuring arguments and statements as a graph, where edges define logical relationships, such as attacks or support for arguments. While this gives arguments a formal structure that is easy for machines to process and reason about, it can be difficult for untrained users to grasp due to its comparatively novel, non-sequential nature.

The *Dialog-Based Argumentation System* (D-BAS) (Krauthoff, 2018; Krauthoff et al., 2016, 2018) was developed to address this by integrating argumentation graphs with a dialog-based interface. Rather than displaying the unfamiliar graph, D-BAS guides users with the use of chat like interface. This interface enables D-BAS to function as a virtual argumentation partner. Users are guided through a series of contextually relevant questions and answers, simulating a cooperative debate. Each turn in the dialogue corresponds to traversing the argument graph, making the underlying structure accessible while reducing cognitive load. The interaction cycle the users are in is shown in Figure 3.1. Using D-BAS as a proxy also enables discussions to scale efficiently without the need for direct user-to-user interaction.

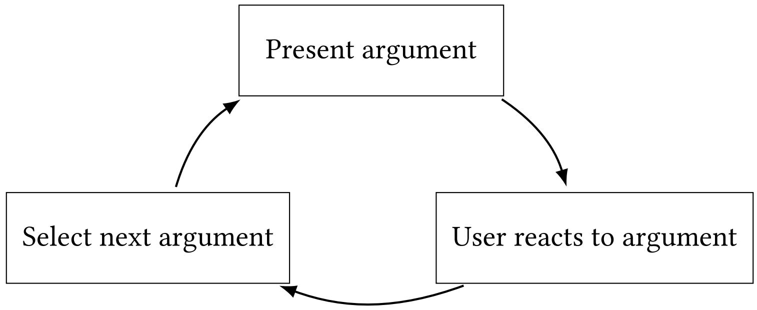



Figure 3.1: Circular process of argument presentation and user interaction in D-BAS. Graphic adapted from “Rethinking Online Discussions” (Meter, 2020).

## 3.2 Decision Making with Argumentation Graphs

During and after the argumentations in D-BAS, questions often arose about the outcome of the discussion, such as “Who won the argument?”. Although not every argument requires a definitive conclusion, we recognized the need to explore whether and how structured argumentation could inform decision-making processes.

This was the topic of a subsequent master’s thesis (Ebbinghaus, 2019). In this work, a large-scale experiment was conducted in which students at the Department of Computer Science at Heinrich Heine University Düsseldorf participated in a participatory budgeting process. The aim was to democratically allocate institutional funds to improve teaching conditions by collecting, discussing and prioritizing competing proposals with the help of structured argumentation.

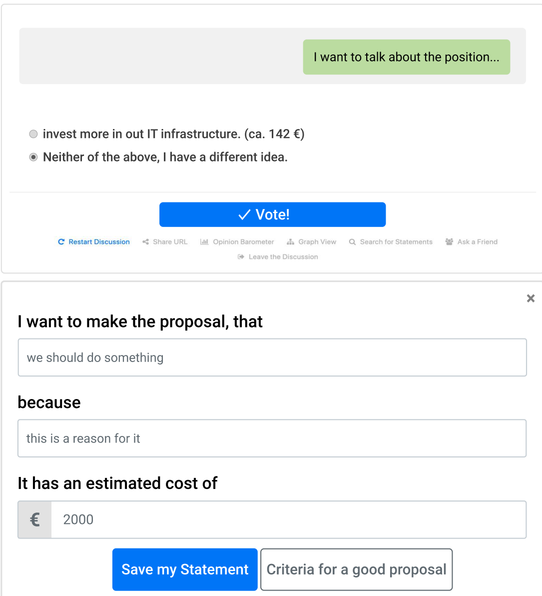
For this purpose, D-BAS was modified and extended to accommodate not only argumentation, but also proposal formulation, favourite selection and prioritization. The resulting platform was called *decide* and implemented a three-step approach to the decision-making process.

**Step 1** First, participants submit proposals, including an estimated cost (shown in Figure 3.2a). All participants then use dialog-based argumentation to argue about the validity and priority of the proposals. This phase accounted for a significant proportion of the overall process time. Submitting new proposals is possible at any time.

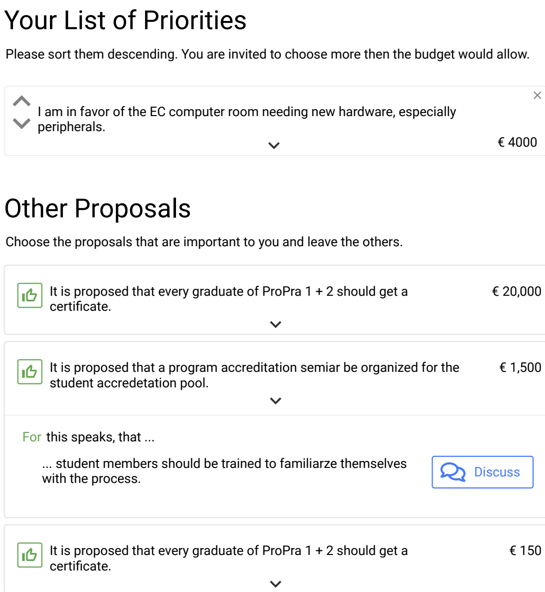
**Step 2** The second step is to validate the proposals. This means that in our specific experiment we checked whether there were any reasons why one of the proposals could not be implemented, even if the proposed resources could be allocated to it. For example, one proposal required significant construction work, which was not feasible. The remaining proposals, together with their supporting arguments, then proceeded to the next stage.

**Step 3** Finally, participants prioritize the proposals. First, they select the proposals they wish to support. They then rank these proposals according to their own priority (see Figure 3.2b). At this stage, the arguments attached to the proposals can be viewed and extended, but no new proposals can be created. The final result is then computed using a truncated Borda count, followed by a greedy collection of proposals that fit the budget.

Ultimately, 165 students participated in the process, of whom only 52 took part in the argumentation phase. We demonstrated the system at Comma 2020[1]. The three steps and figures in this subsection are taken from the accompanying abstract (Ebbinghaus and Mauve, 2020).




(a) Interface for entering new proposals into D-BAS. A dedicated page explained to participants what constitutes as an acceptable proposal.




(b) Participants can accept proposals at the bottom and then prioritize the selected proposals at the top. The proposals can also be expanded to show arguments for and against them, together with a direct link into the argumentation.

Figure 3.2: Examples from *decide*'s user interface.

[1]Recording of the demo: https://link.cs.hhu.de/decide-demo-video

### 3.2.1 Observations and Findings

During the argumentation phase of the process, participants argued for and against the various proposals. As the argumentation system tried to find counter-arguments automatically, tending towards confrontation and always demanding a new counter-argument, the discussions about some proposals often took the form of "I disagree because..." or "Yes, but...". Participants were alerted when an argument was made against their proposal or argument, and could enter the argument at that point, again providing a counter-argument to defend their original claim. Participants argued back and forth as follows:

**Proposal** We should do X.

- ⇒ That won't work for reason Y.
  - ⇒ This could be solved by changing Z.

As with similar participatory processes, comments (or, in this case, arguments) made after the initial proposal did not have the same level of importance or influence on the process. Arguments could be made for and against, but ultimately it was the original proposal that was voted on, while the argumentation only served to inform and form opinions.

In some cases, proposals were moderated between the argumentation and voting phase to incorporate improvements suggested during the argumentation stage. This manual process was not suitable for larger participatory processes and would have added a considerable amount of work. Furthermore, this manual intervention failed to address certain issues and introduced new ones.

Firstly, the introduction of an external moderator to oversee the argumentation process and determine how the proposal was drafted raises questions of independence and impartiality. The task was not only to incorporate the necessary improvements from the argumentation, but also to decide how the changes should be formulated or even interpreted. This creates the risk of unintentionally or intentionally changing the original proposal.

Secondly, it is not unusual for a proposal to lead to multiple counter-arguments with different and sometimes mutually exclusive proposals for resolution. Example:

**Proposal** We should buy a car.

- ⇒ A red car would work best!
- ⇒ Blue is a good colour for a car.

Depending on the circumstances, this can be resolved in different ways.

1. One argument wins over the other.
2. A compromise can be found.

The second option can be considered an extension of the first, whereby the compromise (a third argument) would take precedence over the original arguments. However, it would be even more difficult for the participants to make this compromise as an argument because of the capabilities of the argumentation system. In any case, a moderator would need to judge which argument is the most convincing, which is a highly subjective task.

Finally, there is the case where the conflict cannot be resolved. In this situation, rather than choosing, a moderator must split the proposal and create an additional candidate for the final ballot. This raises a number of other issues. If the proposal is split after the argumentation stage, participants do not have a chance to make adjustments. This may result in proposals where the participants would have preferred to make more substantial changes if they had been able to do so during the process. Any compromises that one of the participants has made would be irrelevant after the split, as they would not have had to compromise at all if the proposal had been split earlier. Additionally, since the two new proposals are likely to be very similar in content, this could lead to vote-splitting problems or incompatibility constraints, depending on the voting system.

These observations and findings during and after the argumentation lead directly to the development of techniques for managing proposals and their evolution in large-scale online participation processes. This was the motivation for the first publication in Chapter 4.

# Chapter 4

# Better Decision-Making through Collaborative Development of Proposals

Björn Ebbinghaus and Martin Mauve

"Better Decision-Making Through Collaborative Development of Proposals"

DOI: 10.1007/978-3-031-05544-7_2

In: *HCI International 2022 Proceedings*. Vol. 26: *HCI in Business, Government and Organizations*, pp. 11–23. 9th International Conference, HCIBGO 2022. Held as Part of the 24th HCI International Conference, HCII 2022 (Virtual Event, June 26–July 1, 2022)

Acceptance Rate: ~ 57.8%

This chapter discusses collaborative online participation processes where participants can iterate on proposals. Often the means of collaboration are voting and, more importantly, comments. Comments are a basic feature of the Internet—and even predate it. In a process comments, at best, clarify outstanding questions that some participants may have about a proposal. At their worst, they are filled with hatred and useless partisanship.

Sometimes comments highlight real issues with a proposal, whether they are fundamental flaws or just matters of opinion. This type of comment is extremely valuable. However, there is usually a gap in functionality: on one hand, participants can make proposals; on the other, they can leave comments. Changes to or concerns with proposals remain as simple text snippets and do not enter the decision-making process as actionable entities. What is missing is an easy way for participants to independently iterate on proposals, and to take ideas from the comments and turn them into new proposals that are actually considered in the process.

This opens up participation processes to the risk that valuable ideas are lost in long threads or dismissed as mere feedback rather than as a basis for further refinement, as happened in our participation process (Ebbinghaus, 2019).

## 4.1 Contributions & Results

To address these issues, we have developed a proof-of-concept web platform[1], where users can derive new proposals from existing ones. This allows participants to build on each other's ideas and turn discussions into actionable proposals. The user interface is specifically designed for proposal iteration, with features such as a step-by-step merging interface and situational call-to-actions to help participants navigate the unfamiliar system.

Participants can also vote on proposals. These votes are counted live and used by the systems to suggest areas where collaboration or compromise between proposals might be fruitful.

Both the idea and the software were evaluated using experiments with different voting configurations involving groups of students. One group used limited (plurality) votes and the other used unlimited (approval) votes. During the experiment, the participants' actions were recorded and analysed. This made it possible to track voting behaviour and the creation of proposals. In addition, the students were asked to fill in a questionnaire afterwards, asking about their subjective experience with the system, especially regarding the perceived fairness of the outcome and their influence on it.

We found that live voting during the iterative proposal development process had a significant impact on the way proposals were made. During the experiment, participants in the group with limited votes were more reserved in their voting and in creating derivative proposals. New iterations of proposals became a risky endeavour because of vote splitting. Because the vote was live, participants could see the effect immediately, which hindered the development of proposals and ultimately resulted in significantly lower user satisfaction with the outcome.

#### Personal Contributions

Björn Ebbinghaus was responsible for the central ideas, implementation and execution of the experiment and the surveys. Martin Mauve acted as an advisor during the conception and writing stages.

[1] The *decide3* application is open source and the code can be found here: https://link.cs.hhu.de/decide3-repository.

## 4.2 Impact on the Thesis

The new system achieved the goal of generating new, improved proposals, but unfortunately this quickly became a usability issue. Because the decision platform was designed to be egalitarian, with little need for moderation, it was not possible to determine whether a new proposal was genuinely an improvement. The original idea behind limited plurality voting was to force participants to regularly change their vote to indicate outdated proposals. Instead, votes could not be used to indicate the relevance of proposals, forcing the system to display multiple related—and therefore very similar—proposals to participants. This resulted in the system being difficult to scale to larger amounts of participants, making it practical only in disciplined expert rounds or short-lived processes.

Further research is needed into scalable decision-making systems that allow public online participation processes and have the necessary features for advanced collaboration, while protecting participants from becoming overwhelmed. This is the main motivation behind the forthcoming Chapter 5, which describes a system that appears more meritocratic and familiar to participants at first glance, but which allows users greater flexibility in developing and voting on proposals.

# Better Decision-Making through Collaborative Development of Proposals

Björn Ebbinghaus[1][0000−0002−5586−3327] and Martin Mauve[1]

Heinrich Heine University Düsseldorf, Universitätsstraße 1, 40225 Düsseldorf, Germany {ebbinghaus,mauve}@hhu.de https://cn.hhu.de

**Abstract.** In a traditional decision-making process, proposals are made and usually commented on by the participants, and finally a vote is taken. We have found in past public decision-making processes that there are also discussions about negative sides of proposals together with possible improvements of them. We have taken this as an opportunity to model the improvement of proposals within the decision-making process.
Our new model for this is similar to version control systems like git work. Proposals in a decision process can have none, one or several predecessors. This structure allows different constructs of the real world to be modelled. Where previously only proposals could be made without reference and structure among each other, the system presented in this work allows modelling of further developments of proposals or even coalitions and compromises, in a real-time collaborative decision-making process.
To validate our ideas, we tested them in two controlled experiments and concluded that our modifications are useful.

**Keywords:** Online-Participation · Decision-Making · Applications

## 1 Introduction

Our goal is to understand how a potentially large group of participants can be supported to conduct collaborative online decision-making processes with a minimum amount of external governing factors. Since there exists a multitude of systems that are regularly used for tasks such as online participation, much of the required functionality is well known. This includes mechanisms for proposing an item, discussing it and voting on alternatives. However, there is one aspect of collaborative online decision-making that is not yet well understood: how can a large group of participants work *together* to come up with and decide on proposals? In real-world politics, there is a vast array of instruments to refine proposals and form coalitions in order to reach a final decision.

Our idea is to take two of the key underlying elements of these instruments and translate them to an online setting. Namely, improving an existing proposal to gather more support for it and forming a coalition to join the support of two or more proposals. We believe that these elements can be translated to mechanisms that are similar to collaborative software development through distributed version control systems.

Based on this idea, we have developed *decide*, an application for collaborative online decision-making that supports the participants to work together to come up with the best possible solution for a given issue. This does not require the participants to have homogenous interests and agree on a solution. It gives them all the tools they need to come up with good proposals, improve them, form coalitions and decide on the result.

We have conducted several experiments using *decide*. The results of those experiments are encouraging (see Section 4). Participants understand and use the new mechanisms provided for improving and merging proposals. Our experiments also suggest that the specific voting system employed may have a significant impact on how the participants use the option to improve and merge proposals.

The overall idea is inspired by observations of shortcomings in a recent participatory-budgeting process of ours [2,3] where we asked participants to make proposals on how to improve the study course of computer science at Heinrich Heine University (HHU). For that process, we used D-BAS [4], a dialog-based argumentation system, to collect proposals and let participants argue about them. They were able to present pro- and contra-arguments to each other and to help inform newcomers to the process. The proposals made were used in a final vote. In this process, we noticed two behaviours:

During the process, participants made proposals that were very similar. Some proposals were slight amendments of other proposals, or they overlapped in their intentions. This poses a problem in a later vote as it can lead to vote splitting — i.e. votes are split between two candidates where there would have been a majority with only one candidate — or, on the contrary, if both overlapping proposals win at the same time, they cannot be implemented as they represent mutually exclusive issues.

Secondly, we found that when discussing proposals, participants often defended (or attacked) the proposal in question by providing possible implementation details or alternative solutions. They lacked the option of modifying or deleting an existing proposal, so their only option to add modifying details would be to make a completely new and separate proposal, which means more friction in the participation process.

### 1.1 Related Work

The field of collaborative decision-making has a long history. Most of it, however, deals with formal processes and (semi-)automatic decisions, often in the context of business.

This paper deals with a mainly social domain. It does not present a new decision-making framework, but an extension of traditional participatory processes in which proposals are made by participants, accompanied by a software prototype.

With the ever-increasing availability and use of the internet, online participation processes have become increasingly popular, sometimes involving large segments of the population. From informal opinion polls to legally binding participatory budgets.

In a traditional participation process, where proposals can be made by the population, these proposals are submitted by the participants and, often after a review by moderators, are presented to the public. Afterwards, there is the possibility to comment on these proposals and possibly already give one's approval. This can be possible for everyone who visits the respective website in order to keep the hurdle to participation low, or it is secured via postal invitation codes or e-passports; this usually only happens when legally binding decisions are involved.

A similar real world process is called *consensus decision-making* or *proposal-based decision-making*, which is similar to our system in that it focuses on the development of proposals, but since it is offline, it is synchronous, meaning all participants collaborate on the same step in the decision-making process at once. This is simply not feasibly for large, online processes.

### 1.2 Structure

In this paper we will first explain the core idea of this paper and how it has been integrated into a complete decision-making process.

We will then briefly look at voting methods, as these are relevant to the way participants interact with the system and are then used to make the final decision. Among other things, we have learned how much influence the voting method has on the motivation to cooperate and the perception of fairness.

We then present how we have experimentally tested our idea for feasibility. In doing so, we consider our idea as part of the complete system. We also explain and discuss the observations, conclusions and results.

Finally, we will draw a conclusion and show what further steps can be taken with this work.

## 2 The *decide* Decision-Making Platform

As a proof of concept for our idea, we developed decide. A web application, tailored for groups of non-expert participants to develop proposals together, and eventually make a decision.[1]

A decision-making process starts with someone putting forward a topic for decision and defining certain parameters, such as the desired time frame, restrictions on certain participants and visibility. Care should be taken here to describe the topic as precisely as possible and to do justice to the issue to be addressed. A decision-making process with a handful of friends will have different characteristics than a decision-making process of governmental and administrative institutions with an indefinite, large number of participants. Although our

[1] The *decide* application is open-source, and the code can be found here: `https://link.cs.hhu.de/decide3-repository`. We also have an in-development live instance here: `https://link.cs.hhu.de/decide3-repository/decide3`

idea itself is independent of the number of participants, it is intended for use in larger groups of individual participants. Even though we use terms like *coalition* and *compromise*, it does not mean a coalition between two fixed, defined groups such as parties or other interest groups. We aim at grassroots democratic and self-regulated processes.

After a process has started, participants interact with the system by making proposals to solve the issue, arguing about them and at the same time giving their agreement to the proposals made. These approvals serve both as indications during the process of which proposals are popular and where it is worth working together (see Section 3), and as votes for the final decision, if it is desired that the process has a fixed end at all.

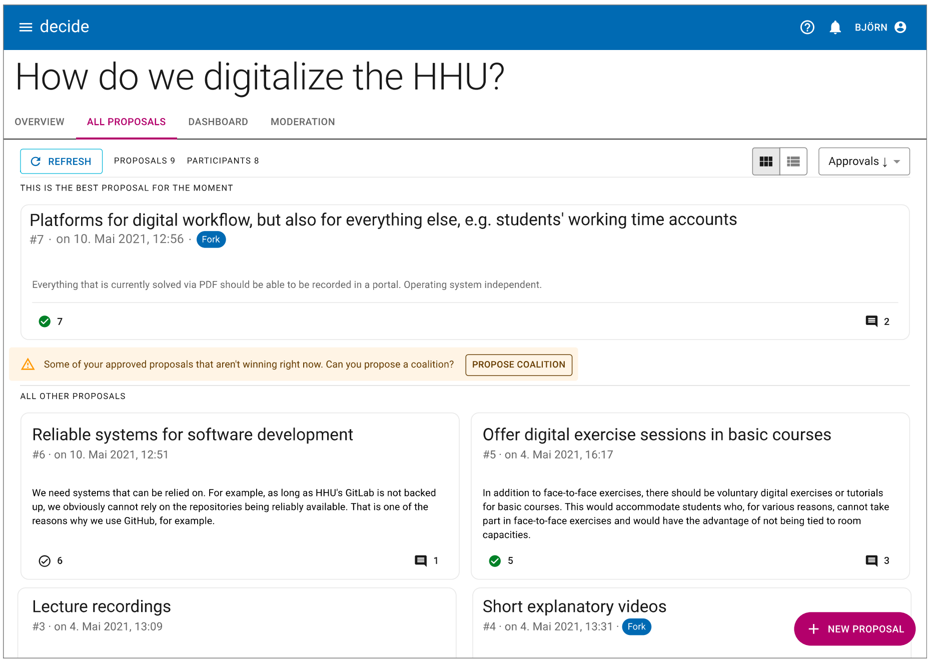



Fig. 1: The main view presented to participants in a process. The current best proposal is displayed on top. Proposals are sorted by approval by default.

### 2.1 The Inner Workings

We base the system of the notion of a *proposal*. A *proposal* is a text entry made by a participant. Every participant can make a *proposal* at any time during a process.

Important to note is, that proposals are immutable, no one, not even the authors can modify or even remove them. In this sense the author of a proposal hosts no special power over this proposal, nor is there a notion of stewardship. We even hid the author of a proposal, after observations of internal tests in our group, as not to associate a proposal with a specific person. This way, we foster a sense of belonging to the idea of a proposal rather than to the author. Having immutable proposals is therefore not just a convenience, but a requirement.

Another reason why proposals cannot change is our live voting system. In order to develop proposals, we need a way to communicate interest and approval of proposals between the participants. Therefore, we decided that having a traditional vote at the end of a participation process is not sufficient. Until the end of such a process it would not be visible how much interest and approval there is in a proposal. Approvals in our application are therefore possible and publicly visible to every other participant immediately after the creation of a proposal. This allows participants to assess the current climate of a process and make better decisions about which proposals are worth to develop further. We explain our voting system in more detail in Section 3.

### 2.2 Processes

Every *proposal* belongs to exactly one (decision-)*process*. A *process* is a topic that frames a decision. It has a title and a description. Optionally there can be a start and/or an end time in which a process is active and participants can participate. Usually there is at least an end time for when a decision process has to come to an end, but nothing speaks against a process that continues forever with ever evolving proposals.

Processes can be public or private and only be visible for invited participants. They can also modify rules about the process, like how much of the approvals are visible. Either the number of approvals is not visible, shown as the sum (this is the default) or with total transparency where everyone can see exactly who approves what. The last setting was popular for more informal decision-processes in small groups.

### 2.3 Development of Proposals

As we explained, we want to support decision-making by developing a set of proposals to a greater, better set of proposals. For this we allow and encourage multiple ways of adding new proposals.

First, every participant can submit a new proposal with an informative and unique title and a detailed description. In the best case a proposal is solely identifiable by the title, with the description only serving as implementation details or clarification.

In addition to this there are multiple ways of creating new proposals from existing proposals. They are all based on the idea that we want to model the relationship of a new proposal and the proposals that have led to it. Figure 2 shows one way, how this is presented to participants.

We call proposals with a single parent *fork* and proposals with multiple parents *merge*. These are terms from version control software, where the state of a project is tracked by a chain of annotated changes. This allows to track every development that ever occurred in the said project. Figure 3 portrays these relationships.

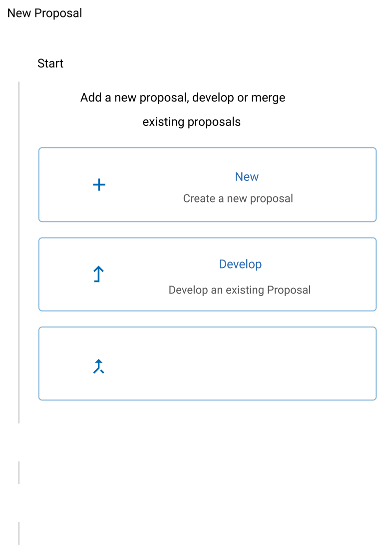



Fig. 2: Step by step interface to add a new proposal

Semantically, a *fork* can represent a variety of relationships. It would represent an enhancement of a proposal, an alternative or really any related semantic modification of a proposal. This is the intended way of modifying a proposal. Modifying it in place would not be possible as every proposal is a candidate in an ongoing vote. For the moment, we leave the exact meaning to the author of the new proposal. Of course, a more specific annotation would be conceivable, but for now we want to focus on different groups, according to the underlying system.

*Merges*, proposals with two or more parents, are the construct with the most potential in our system for collaboration. By enabling participants to link multiple proposals into a new one, real-world constructs are made possible. These can be, for example, compromises and coalitions. It should be noted that the content of a new *merge* proposal is not only a technical linkage of existing proposals. In terms of content, an independent proposal is created here, with its own title, content and arguments. This is a necessary freedom, as participants should not be restricted by the system as to how their *merges* should be. The application should only play an advisory role and not restrict the participants.

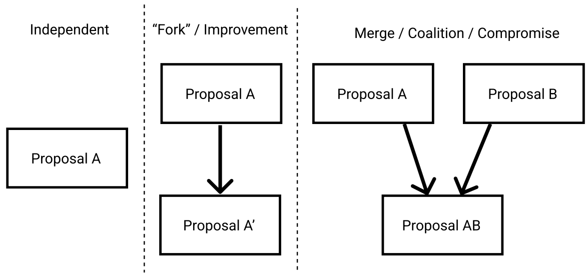



Fig. 3: Different kinds of relationships between proposals. Note, that *merges* are not limited to only two parents.

## 3 Voting

To come to a final decision, we use votes. So far, we already have the advantage that our system is designed to put up more candidates for a voting, which allows participants to better reflect their opinion. However, we have some special requirements in our system. In order for the participants to be able to work effectively with each other, it is necessary for them to be able to communicate amongst themselves in one way or another. We offer three different ways to do this in our system:

1. The creation of proposals itself (portrayed in Section 2)
2. Giving approval to one (or more) proposals
3. Nested comments on each proposal

This section deals exclusively with Point 2. It is necessary for the participants to recognize during the ongoing process which proposals are liked and which are disliked. Only then will participants be able to suggest useful enhancements and coalitions.

We have looked at different voting methods and weighed up which is suitable for the processes. For this, we have established the following requirements.

1. It must be easy to cast a vote.
2. It must be possible to see how popular a proposal is compared to another proposal.
3. The current ranking of the proposals (and thus also the current winner) must be transparent.

We first used approval voting [8]. It is a good match regarding our requirements. By today most people are familiar with a notion of a *like*-Button. Just a single click is necessary to cast a vote, fulfilling the first requirement. It is easy to aggregate and display the votes by summing them and thus communicate the current popularity of a proposal. Fulfilling the second requirement, it allows sorting the proposal by this sum, which fulfils the third requirement. This leaves the possibility of a draw, especially in processes with fewer participants. We have decided not to consider this problem further, as draws are an anomaly that can — and often should — be solved in the context of the situation. For our experiments we reliably broke the tie with the creation time of the proposals, letting more recent proposals win over older proposals. We decided on the basis of a gut feeling that newer proposals had less time to convince and woo away potential voters.

We also considered more complex voting methods. In particular, a preference-based voting method, such as Instant-runoff Voting (IRV) [1]. In such a method participants have to enter more precise votes. They have to rank proposals against each other. The winner is determined by repeatedly eliminating the worst candidate and recounting the remaining votes, until only the winner remains. While this may result in a vote that better matches their real intentions, it also requires more effort from the participants, especially if the process involves many proposals. Furthermore, with preference voting, it is no longer possible to display a simple aggregate of votes of the other participants. This makes it difficult to compare two proposals and thus the overall state of the process is no longer easily tractable.

The fact, that voting is iterative [6] in our system creates unique situations. While the effect of one's own vote in a traditional election is only visible after the counting, the participants in our system see the effects immediately.

This means that, for example, strategic voting can be experienced first-hand by participants. Participants can experiment with how they cast their votes and immediately see the impact this has on the current interim result of the process. In a process that uses plurality voting, i.e. a process where only one proposal can be approved, participants immediately see that giving their vote to a proposal has a direct negative effect on their previous favourite. We will see in Section 4 that this quickly leads participants to perceive the voting method as unfair, a perception that is in line with reality but usually only noticeable to people who have studied the voting method more closely.

Although in this case the weaknesses of the voting method are made more tangible by the immediate feedback, there are also possible advantages. Participants experience the direct influence of their vote, which may motivate a more detailed, further engagement with the candidates.

## 4 Experiments

To evaluate our platform, we tested our idea in two experiments with untrained students. After the first run, we made modifications based on our observations and feedback from the participants and then reviewed them in the second run. This led to a significant improvement in some parts of the surveys following the experiments.

## 4.1 Environment

We conducted the two decision-making processes with a duration of five days each, with a subsequent survey via a questionnaire. Both processes had the same theme namely: *How can we improve digitalization at the university?*. This is a current topic, which affects students from different study programmes and even staff. It lends itself well as a topic for our experiments, as there is still much need for development here. Large-scale digitalization of various aspects of university life is still uncharted territory and therefore lends itself well to input and ideas from students and staff.

In addition to payment for participation, as a special incentive, it was promised that we would forward the most approved proposal to the university's Pro-Rector for Digitalization, who would then comment on the proposal and the arguments relating to it. We then recruited the participants through advertising in lectures, the student council and personal letters. The majority of participants came from the Faculty of Natural Sciences at our university.

## 4.2 Setup and Execution

Both groups had the same topic and the same length for their run. Before starting the experiments, the participants were informed about what features the application provides and what the special features are. However, the task was deliberately kept simple. The participants were asked to visit the application daily, learn about new developments and participate in the process. No guidelines were given on how or when to participate. In particular, it was not required that the novel features of the application had to be used.

The processes took place completely remotely, partly due to the current pandemic. Although it is more difficult to observe the process and the participants, the process is closer to reality.

The processes started on Monday at noon. On Wednesday, there was another appeal and a brief overview of the current progress. On Friday at noon, the process ended and we released the questionnaire. At least 15 participants took part in each of the processes, although in both cases only 10 participants were actually active during the entire week like planned.

During the processes we fixed software and minor quality-of-life issues, like improved loading times and better contrast for people with impaired eyesight, based on participant feedback and observations, but the rules and content of the processes were not touched.

In the first experiment, participants had only one approval they could give to a proposal. If they agreed to another proposal, the approval to the former proposal was removed. We decided to do this because we noticed in internal, technical tests that participants tended to just agree to everything. The following working hypothesis was formulated for this:

**Hypothesis 1** *Some pressure exerted on participants by the decision-making mechanism motivates collaboration with other participants.*

Participants of the second experiment could approve as many proposals as they liked. In both cases the winner with the most votes won.

### 4.3 Observations in the First Experiment

This made for a more active process, as participants had to think more about which proposal they wanted to vote for and which proposals they did not. They could not just approve to everything, a behaviour we noticed in earlier, internal tests.

It is possible for us to track the changes in votes in order to understand the voting behaviour of the participants. This allows us to observe and answer questions such as: *Do participants stay in one strand of proposals or do they switch back and forth?*, *Are participants more likely to shift their approvals to newer, developed proposals?*.

Based on the final survey and feedback during the process, we noted that some participants have found the voting procedure unfair or dissatisfactory, as they have had to take a risk with their vote to support a new proposal they like better. Hurting their current favourite proposal by giving their only approval to the new proposal with fewer votes.

Although vote splitting is not a new problem, in our public, live voting system it is directly experienced by the participants and thus poses a bigger problem. An open, live voting system makes strategic voting more visible and thus also the advantages and the disadvantages it entails.

We have thus made the following hypothesis for the next iteration of our system:

**Hypothesis 2** *The voting mechanism is highly influential to the satisfaction with the decision result in our system.*

Based on this knowledge we ran the second experiment with simple approval voting. Each participant can give as many or as few approvals as they want. Approval voting, which is a cardinal voting method, is immune against vote splitting [7]. This removes the risk for participants to vote for a new proposal that has not yet received much attention.

### 4.4 Observed Differences in the Use of Our New Features

Figure 4 shows the proposals and relations between parents and children that emerged during each experiment. It can be seen that in the first experiment much more use of our tools to develop proposals was made, with the same number of participants and proposals.

We therefore assume that our original **??** 1, that more collaboration takes place with a little pressure, is pointing in the right direction. Although this behaviour can be attributed to chance with the low numbers of participants, we believe that the difference in behaviours is significant enough to support our hypothesis.

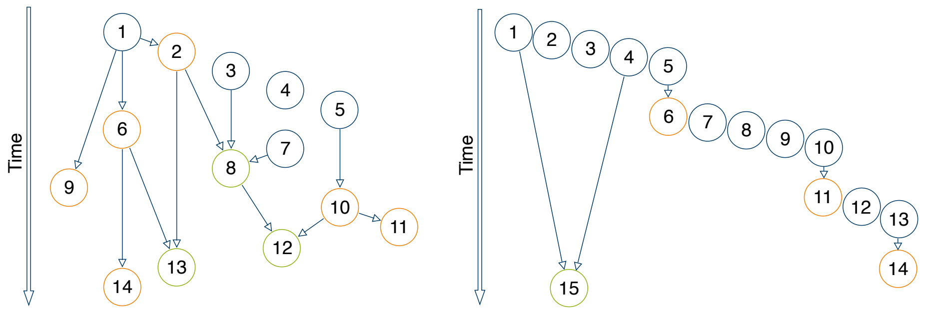



Fig. 4: Proposals and their relations made by participants in the two experiments.
**Left**: First experiment with plurality voting. (3 *merges*, 6 *forks*, 5 independent)
**Right**: Second experiment with approval voting. (1 *merge*, 3 *forks*, 11 independent)

### 4.5 Survey Results

The final survey that both groups completed after their process, consists of several sections with different statements. Participants should indicate how much they agree with these statements. We used a Likert scale [5] from 1 (strongly disagree) to 5 (strongly agree).[2]

The topics of the sections are:

S5. First encounter with the application.
S5. Feelings regarding the outcome.
S5. Statements about the process in general.
S5. Statements about argumentation.
S5. Statements about *forks*.
S5. Statements about *merges*.

Both groups do not differ significantly in their answers except for statements in one section. Overall the reception to the experiment, the application and the features it comes with was positive. The biggest criticism of the application by the participants was the confusing nature given a larger number of proposals or arguments. There was a wish for clearer indications of which proposals came from which other proposals.

Questionnaire-Section 4.5, about the participants feelings regarding the outcome, had significantly more positive grades with the second experiment, where participants had unlimited approvals. We compared the statements of both groups with a two-sided t-test for different variances [9], due to our low $n$ of 10 and 12 respectively. The values are presented in Table 1.

[2] The full, translated data, can be found here: `https://link.cs.hhu.de/data-ebbinghaus22_collabproposal`.

Table 1: Comparison the statements in question-section Section 4.5 of the first experiment with a single approval and second experiment with multiple approvals. 1 = strongly disagree, 5 = strongly agree

| | Plurality | | | Approval | | | |
|---|---|---|---|---|---|---|---|
| Statement | n | Avg. | Var. | n | Avg. | Var. | p(TT) |
| I am satisfied with the outcome of the decision. | 12 | 3,75 | 0,75 | 10 | 4,70 | 0,23 | 0,005 |
| I think my opinion contributed to the final decision. | 12 | 3,50 | 0,64 | 10 | 4,60 | 0,49 | 0,003 |
| I find my opinion reflected in the decision. | 12 | 3,67 | 1,15 | 10 | 4,70 | 0,23 | 0,009 |
| I think my voice was heard. | 12 | 4,00 | 0,73 | 10 | 4,80 | 0,18 | 0,011 |
| I felt the process was fair. | 12 | 4,00 | 1,64 | 10 | 4,80 | 0,18 | 0,061 |
| I think I was correctly informed about the process. | 12 | 4,50 | 0,45 | 10 | 4,70 | 0,90 | 0,584 |
| I think the decision is a joint product of everyone. | 12 | 3,75 | 1,48 | 10 | 4,80 | 0,18 | 0,014 |

The clear difference in the results of the two experimental groups reinforces our belief that the perceived fairness of a voting system used is a key determinant of satisfaction with the process and outcome of a process in our application. We accept **??** 2. This should therefore be considered and investigated in further research.

## 5 Conclusion and Future Work

In this paper, we have shown a system with features that provide participants in a decision-making process with tools for developing proposals together.

We have shown that these features are understood and used by participants. However, we have also learned through the second experiment that such a system, in its current state, needs to encourage participants to collaborate.

In our first experiment, this happened involuntarily due to constraints in the voting method. Participants were motivated by the characteristics of the voting method not to deviate further from the currently strong proposals and therefore tend to stick with the existing ones. By being limited to one approval, exploring new proposals became a risk for participants. Although effective, we believe this is not good for future processes because it goes against our overarching goal of providing the tools to make more and better proposals.

A suitable methodology to encourage participants to cooperate and at the same time provide a voting process that is perceived as fair is thus indispensable for further research on this system.

We believe that the system itself could provide more guidance and advice to participants in a process. Currently, the system itself is not *smart*, it does not actively support participants, e.g. by recommending suitable proposals or worthwhile coalitions. We have already made first attempts in this direction. By using a public, iterative voting method, the system already has information about which voter groups support which proposals during a decision-making process. This can be used for collaborative filtering, among other things, in the future.

**Acknowledgements**

Björn Ebbinghaus is a member of the PhD-programme *Online Participation*, supported by the North Rhine-Westphalian funding scheme *Forschungskollegs*.

## References

1. Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A.D.: Handbook of computational social choice. Cambridge University Press (2016)
2. Ebbinghaus, B.: Decision Making with Argumentation Graphs. Master's thesis, Department of Computer Science, Heinrich-Heine-University Düsseldorf (May 2019). https://doi.org/10.13140/RG.2.2.12515.09760/1
3. Ebbinghaus, B., Mauve, M.: decide: Supporting Participatory Budgeting with Online Argumentation. In: Prakken, H., Bistarelli, S., Francesco, S., Taticchi, C. (eds.) Computational Models of Argument. Proceedings of COMMA 2020. Frontiers in Artificial Intelligence and Applications, vol. 326, p. 463–464. IOS Press (September 2020). https://doi.org/10.3233/FAIA200535
4. Krauthoff, T., Meter, C., Betz, G., Baurmann, M., Mauve, M.: D-BAS – A Dialog-Based Online Argumentation System. In: Computational Models of Argument. pp. 325–336 (September 2018). https://doi.org/10.3233/978-1-61499-906-5-325
5. Likert, R.: A technique for the measurement of attitudes. Archives of psychology (1932)
6. Meir, R.: Iterative voting. In: Endriss, U. (ed.) Trends in computational social choice, chap. 4, pp. 69–86. AI Access (2017)
7. Poundstone, W.: Gaming the vote: Why elections aren't fair (and what we can do about it). Macmillan (2009)
8. Weber, R.J.: Approval voting. Journal of Economic Perspectives **9**(1), 39–49 (1995)
9. Welch, B.L.: The generalization of 'student's'problem when several different population varlances are involved. Biometrika **34**(1-2), 28–35 (1947)

# Chapter 5

# Version Control for Structuring Collaborative Decision-Making

Björn Ebbinghaus and Martin Mauve

"Version Control for Structuring Collaborative Decision-Making"

URN: urn:nbn:de:hbz:061-20250715-102007-0

In: p. 9

This chapter presents improvements to the work presented in Chapter 4. The main concern is the inevitable proliferation of proposals. Small, incremental improvements could not realistically be supported, as each change would generate a new proposal, and there was no automatic way of deciding which proposal should take precedence when displayed. The previous approach attempted to provide tools for iterative collaboration, but failed to consider accessibility and scalability for the real world. As a result, it functioned in theory, but not in practice with real users.

As a consequence, the requirements have been adapted in this contribution. There should still be no absolute hierarchy in the process, i.e. no external moderation deciding what is important and what content should be hidden, but the new system recognizes the advantage of having a controlling instance with (limited) control over part of the collaborative process. Consequently, the new approach has shifted from a purely egalitarian model to a more meritocratic one, where participants willing to coordinate the development of proposals implicitly gain control over them.

The new approach follows the idea of *progressive enhancement*, whereby novice or uninterested participants can experience the process as they would any other traditional process, while those willing to delve deeper can make use of increasingly complex yet powerful features.

## 5.1 Contributions

The newly proposed system adopted techniques from field-tested distributed collaboration systems, namely *Version Control Systems* (VCSs) such as Git[1], which the original system described in Chapter 4 was already close to. The mechanics and features of VCSs were adapted to the semantics of collaborative decision-making processes.

The most notable change is probably the idea of separating the content of proposals from their public façade, which enables the advanced features while hiding the complexity from novice participants. The *decide3* system already stored proposals as immutable entities, including their relationships to other proposals. This data structure is now hidden—but still accessible—to participants and contains the content, called the *version*, while the public-facing *proposal* is merely a reference to one of these versions. In this sense, it resembles a directed acyclic graph of commits in a VCS, where versions are commits and proposals are branches. While this may seem very technical, especially to untrained users, the system can be used by most, if not all, users without knowledge of the underlying principles. At its simplest, the process can be thought of as a traditional decision-making process with editable proposals.

Using this underlying data structure, the work demonstrates how features of VCSs and VCS platforms such as GitHub[2] can be used to enhance the collaborative process by enabling asynchronous work on proposals. It demonstrates how participants can request or propose changes to existing proposals, and how these changes can be managed in a way that is flexible enough to ensure that these mechanics do not add too much overhead to the process itself.

### Personal Contributions

Björn Ebbinghaus developed the central ideas of the work. Martin Mauve acted as an advisor during the conception and writing stages.

[1] https://git-scm.com/
[2] https://github.com/

## 5.2 Impact on the Thesis & Future Work

This is the most recent work in this thesis and remains untested as of now. Nevertheless, it builds on the lessons of at least previous works (Ebbinghaus, 2019; Ebbinghaus and Mauve, 2022) and concepts of VCSs that have been battle-tested in collaborative software development.

Although moving away from a strictly egalitarian model makes the system more pragmatic, it remains to be seen whether the provided mechanics and implicit power dynamics will ensure the system remains inclusive. If so, the system would be more open than most current systems, which do not address these issues.

Future work should focus on testing the ideas in this paper and refining the mechanisms that guide participation without restricting it. This would ensure that advanced features remain accessible to motivated non-experts while not hindering more casual users.

# Version Control for Structuring Collaborative Decision-Making

BJÖRN EBBINGHAUS and MARTIN MAUVE, Heinrich Heine University Düsseldorf, Germany

The collaborative development of decision alternatives is a complex challenge. Our approach is to apply mechanisms from a domain that faces similar problems—version control systems used in software development. These mechanisms, designed to manage contributions and avoid conflicts, offer valuable solutions for structuring collaborative decision-making. By adapting these techniques, we propose ways to ensure effective participation while keeping the process manageable. This paper demonstrates their application and the necessary modifications for decision-making contexts.

Additional Key Words and Phrases: group decision-making, social computing, online-participation, version control systems

## 1 Introduction

The widespread availability of Internet access has given rise to a variety of collaborative decision-making processes. Entire communities, cities and even countries are trying to involve their citizens in decisions that would affect them. What was previously only possible in small groups of experts is now, at least theoretically, possible on a large scale.

One aspect of collaborative decision-making processes is the ability of participants to contribute their own ideas and proposals. In previous collaborative decision-making processes, this was limited to participants posting their ideas and others commenting or voting on them. While there have been some ideas about making the generation of proposals itself collaborative[1], there has been no widely accepted approach to providing this functionality in collaborative decision-making systems. What is traditionally done by committees of a few experts now poses a major communication challenge when done by participants themselves. Proposals generated by untrained participants need to be vetted and structured for refinement, not by said experts, but by a crowd of co-participants.

Supporting collaborative proposal generation requires some rules and support for what participants can do. How can a large group of untrained participants systematically create options and develop them iteratively without the whole process descending into chaos? While it is desired that *all* participants have the opportunity to participate, they must not get in each other's way. If a participant wants to improve an option, it must not simply be overwritten. At the same time, many participants may have many ideas for improving a proposal, so how can these be coordinated? No version of a proposal should be lost, but at the same time there may be many similar proposals. Participants in an online process interact asynchronously with the system, effectively forming a distributed system in which it is not certain that all participants are on the same page at all times.

This work explores the usability of version control system (VCS) techniques that have been successfully used for collaborative software development. The approach of incremental improvement through iterative steps, combined with a structure that allows almost anyone to participate with minimal risk, seems a good fit, even if some improvements are needed.

### 1.1 Related Work

There has been work on collaborative writing tools using VCS, such as *Upwelling*[4] and *Patchwork*[3]. These tools combine real-time collaboration with version control, allowing multiple authors to work on the same document at the same time. While popular software such as Google Docs also

Authors' Contact Information: Björn Ebbinghaus, ebbinghaus@hhu.de; Martin Mauve, mauve@hhu.de, Heinrich Heine University Düsseldorf, Group Computer Networks, Computer Science Department, Faculty of Mathematics and Natural Sciences, Düsseldorf, Germany.

uses versions, which can be described as other VCS features, this is kept simple and is not the main focus of their design. Most closely related to this work is the proposal by Weidner[5], which outlines a design for versioned collaborative documents. The proposal combines Google Docs-style real-time collaboration with Git-style fork-merge collaboration. The author outlines the architecture of a local-first platform for versioned collaborative documents, outlines the programming model that programmers would use to support new document types on top of the platform, and discusses the tools needed to implement the proposal. It focuses more on the technical aspects of the local-first nature of the system, while this work explores the usability implications of such a system for use in large-scale collaborative decision-making processes.

### 2 Version Control Systems

Version control systems are widely used by software developers to manage and track changes to the files in their code base. It allows developers to collaborate on a project simultaneously while keeping track of every change made. The most widely used VCS for code today is git[1].

Version control systems such as git track changes to files in a project. Developers decide when it is time to save the latest changes to a set of files, usually when a coherent change to the code base has been completed. Once these changes have been recorded, developers share them with their collaborators by publishing the changes to a central platform.

Once a change has been recorded, it is uniquely identifiable and becomes a permanent part of the history, i.e. a *version*, of the project. The entire history of all files can be replayed or rewound by applying or removing these changes from the files one by one. This allows people to follow the evolution of the code, for example to see when a bug crept in, or perhaps to see in what context those changes were needed at the time. Both of these features are important not only for software development, but for many other collaborative projects.

VCSs allow multiple developers to work on the same files at the same time. These systems minimise conflicts by automatically merging changes when edits are made to separate parts of a file or to different files. This feature ensures a scalable workflow, allowing projects to involve large teams working simultaneously without constant manual coordination.

However, these mechanisms are not foolproof when changes overlap in the same part of a file. In such cases, conflicts arise, and automatic merging becomes impossible. To overcome this, these systems provide advanced conflict resolution tools.[2] Developers can review conflicting changes, understand their context, and manually reconcile differences to maintain a consistent project history.

These tools have evolved from working with known collaborators to tools that facilitate external collaboration, inviting people from all over the Internet to contribute their changes and move a project forward together. And while this depends on a few developers in privileged positions reviewing and accepting those changes, with the entire history of changes visible to everyone, it is trivial to make a private copy of a project and continue working on it, while getting updates from the main project when needed.

Using these features, VCSs like git have proven to be powerful tools for collaboration at scale, allowing thousands of developers to work together on developing complex software.[3]

[1] https://git-scm.com/
[2] https://git-scm.com/docs/merge-strategies
[3] https://github.com/torvalds/linux

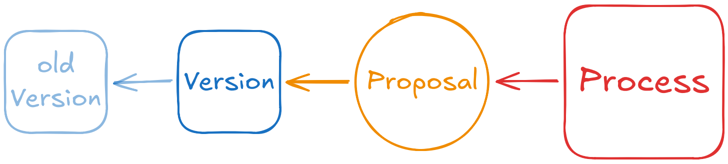



Fig. 1. Iconography. Note that the real relation from child to parent is shown here, while in later figures, the relations between versions are reversed, i.e. they display them chronologically.

## 3 Basic Mechanics

Collaborative decision-making processes require a structure that accommodates multiple stakeholders with different interests and perspectives. For example, in a public participation process, a local government may propose a new zoning code. The initial proposal outlines changes to land use in a district, which stakeholders—including residents, business owners and urban planners—review and discuss. As they make suggestions, such as adjusting building heights or adding green space, each idea is recorded as a new version, providing a clear history of revisions. If a group of residents proposes an alternative zoning plan, a new branch is created so that their input can be considered without changing the original proposal.

During this process, several versions may develop in parallel, each reflecting a different perspective. For example, while one branch may focus on adjustments to business districts, another branch may prioritise green spaces in residential areas. Later, if there is agreement, these branches can be merged into a single proposal that combines both perspectives.

### 3.1 Proposals and Versions

*Proposals* are mutable entities in the system and the main entry point for user participation. As they change, their history should be traceable. For this purpose, proposals consist of a chain of *versions* leading to the current *version*, called the proposal's *head*.

A *version* of a proposal contains its content as plain text or really any appropriate data type and a set of $0 - n$ parents. While proposals are mutable, versions are immutable, meaning that once created nothing about them can change. Neither the contents nor the references. This implies that these references can only be to existing versions, resulting in a causally directed acyclic graph.

Proposals always explicitly refer to exactly one version, and implicitly contain their history through the ancestors of that version. A new proposal does not have to be created with a new version; it can refer to an existing version. Several proposals can refer to the same version at the same time. To keep proposal development reasonably simple, proposals are only allowed to move to a version that is a descendant of the current version, i.e. a previous version can never be removed from a proposal's history.

The creation of a new proposal, while transparent to the user, creates a new version containing the content and a new proposal referencing that version. The proposal is then in the set of proposals for the process.

Later, if the user wants to change the proposal, a new version is created with the new content, referencing the previous version. The proposal is then updated to the new version. The resulting state is shown in Fig. 1.

### 3.2 Branching and Change Requests

The concepts introduced so far work for single users iterating sequentially on proposals. But the aim of the system is for multiple users to collaborate simultaneously.

Once one user has created a proposal, another user can come in and request a change to move it forward. To do this, they create a new version as before, but instead of updating the proposal, they create a new type similar to a proposal called a branch.

Branches and proposals are exactly the same, except that proposals are added to the process's set of proposals, and branches are not. Everything that applies to proposals also applies to branches. A *branch* can be elevated to a proposal at any time.

After the new version and branch have been created, the user can make a request to the original proposal to accept this change (see Fig. 2). If this request is accepted, the proposal will update its head to the head of the requesting branch. The requesting branch can then be deleted.

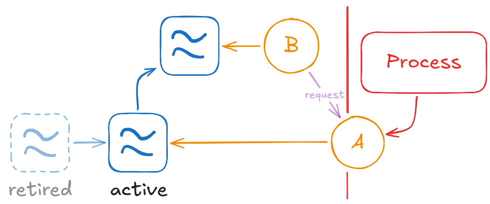



Fig. 2. Branch $B$ requests a change from proposal $A$. If $A$ accepts, it would update its reference to the head of $B$.

Although there are no predetermined roles in this system, someone has to be in control to drive the proposals forward. Control is somewhat meritocratic. The user who originally created a branch/proposal becomes its *maintainer*, and is responsible for the development of that branch/proposal. The maintainers are therefore the ones who open, accept or reject change requests to and from other branches.

If a change request is permanently rejected, the maintainer of the requesting branch may decide to elevate that branch to an independent proposal by adding it to the process. This means that there may be several proposals with a common history. If the situation allows, the two proposals may be merged in the future.

### 3.3 Blockage of Change Request / Divergence of Branches

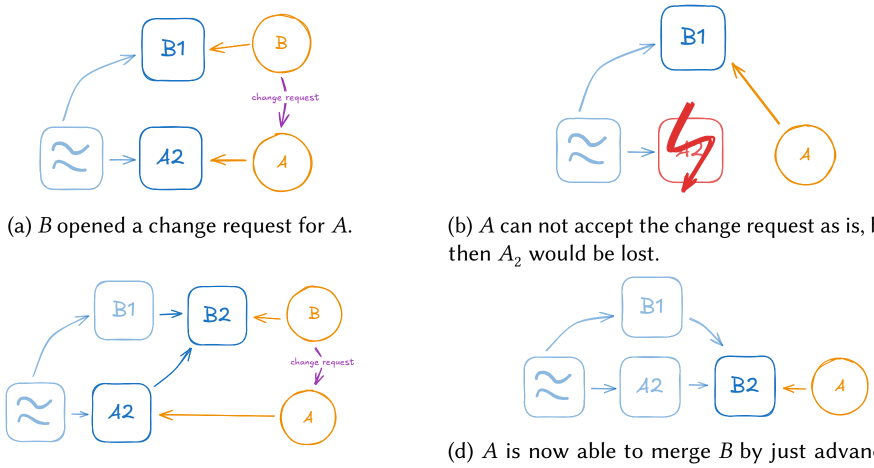
Consider the scenario in Fig. 3a. $B$ created a change request for $A$. Subsequently, $A$ progressed to version $A_2$. In that case, the history of the two branches $A$ and $B$ is no longer linear, but *divergent*. One has versions in its history that the other does not, and vice versa. This is a potential conflict, and it cannot be trivially merged with a click, it is *blocked*, because changing the head of the proposal to the head of the requesting branch would abandon the current head (Fig. 3b). Someone needs to decide how to add the missing versions to the respective branches.

Normally it is the responsibility of the requesting maintainer to resolve the situation, but technically both maintainers are able to merge the content of the two versions. There are ways of automatically merging texts[2], but either way the maintainer needs to check that the semantics of the new version matches the intent.

The requesting maintainer provides a trivially mergeable version again, by including the missing, divergent history in a new merged version (Fig. 3c). After this, the proposal's history is a strict subgraph of the change request, and the receiving maintainer simply has to accept it (Fig. 3d).

### 3.4 Discussion

The mechanisms presented so far give rise to different ways of reflecting real decision-making processes. It allows all participants to participate equally, without fixed roles. The role of *maintainer*




(a) $B$ opened a change request for $A$.

(b) $A$ can not accept the change request as is, because then $A_2$ would be lost.

(c) $B$ creates a new merge version $B_2$, that includes $A_2$.

(d) $A$ is now able to merge $B$ by just advancing its head to $B_2$.

Fig. 3. Scenario in which a change request gets blocked and needs to be resolved before $A$ can easily accept the change request.

is not assigned globally during the process, but is earned when the proposal is created and is then restricted to that role. Responsibility for this proposal is taken on in addition to control.

It is important to note that maintainership is not the same as ownership. Created versions of proposals are publicly available to all participants and can be reused by them at any time, even retaining their full history. If there is a problem with a maintainer, for example because they are no longer involved in the process or are unwilling to accept the desired changes, other participants can easily create a new proposal with the same version. This allows participants to moderate the process themselves. However, the underlying immutable data structure does not allow anyone in the process to perform destructive actions such as deleting versions.

However, the underlying complexity of the system has natural drawbacks. Forks in closely related proposals can be confusing for contributors. Duplication and possibly ambiguous navigation can quickly lead to contributor fatigue, especially if they are unfamiliar with the data schema used. This, combined with the nature of the system where content changes regularly, can lead to disorientation. A mechanism for prioritising proposals is therefore needed to keep the focus on the relevant proposals. More on this in the next section.

### 3.5 Voting

The system can facilitate the creation and iteration of proposals, as well as basic collaboration between users. As a quick and easy way to participate, users can express their opinions as votes, just like in other participation systems.

Primarily votes are cast on versions, as they are immutable. Each vote has the same weight, and it is possible to cast or withdraw votes at any time during the process. The value of a vote is not relevant to the general mechanics, the system supports multi-value votes, e.g. *approve* and *reject*.

An explicit vote cast by a user for a version remains with that version until it is withdrawn, even if the version is obsolete. A vote for the leader of a branch or proposal implies a vote for every branch or proposal that refers to it. Thus, a vote for a single version may imply a vote for several branches/proposals at once.

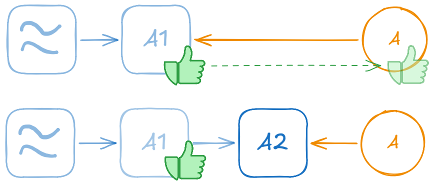



Fig. 4. A vote for $A_1$ implies a vote for $A$, as long as it is the head of the branch. If the branch advances to $A_2$, the implication is lost, and the user needs to agree to $A$s new head again.

By associating votes with versions rather than proposals, the system allows votes to be cast on potential change requests. Users can vote on the version of a change request before it is accepted. Once a change request has been accepted, the version in question becomes the head of the proposal, and thus a vote for the proposal. This approach can also be thought of as a *conditional vote*. This concept is illustrated in Fig. 5.

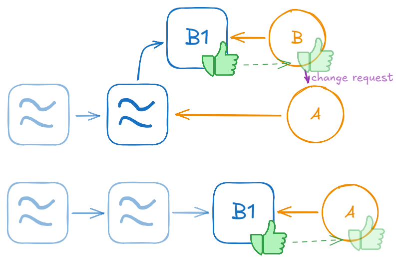



Fig. 5. Branch $B$ has an open change request towards $A$ with version $B_1$, which has a vote. Once $A$ accepts the change request the vote automatically implies a vote to $A$.

*3.5.1 Automatic Following of Votes.* As well as explicitly voting for immutable versions, users can also allow their vote to follow a branch or proposal. This is a usability enhancement to voting. A branch that has a follow vote sets an explicit vote for the current head and any future heads until the vote for the branch is withdrawn. These votes are not just implicit, but explicit votes for versions as explained above. They can be changed individually, and versions that existed before the follow vote was enabled are not affected. Existing votes are not overwritten either.

This automatic following of votes is branch-specific, so it will not follow diverging branches of change requests. After a merge, the history of the requesting branch will not receive any votes, as they were never the head of the branch, although they are now in its history

*3.5.2 Discussion.* The ability to vote in the system is not primarily intended as a means of making a final decision, but as a means of enabling rapid communication. Without it, users would have no clear indication of how many other users are involved in a proposal, other than perhaps the number of change requests.

The ability to vote for a version in a change requesting branch can be important in several ways:

Firstly, for the maintainers of the receiving branch, as they can use it to prioritise their efforts towards more popular requests. They can also see how many votes they could immediately gain with a merge.

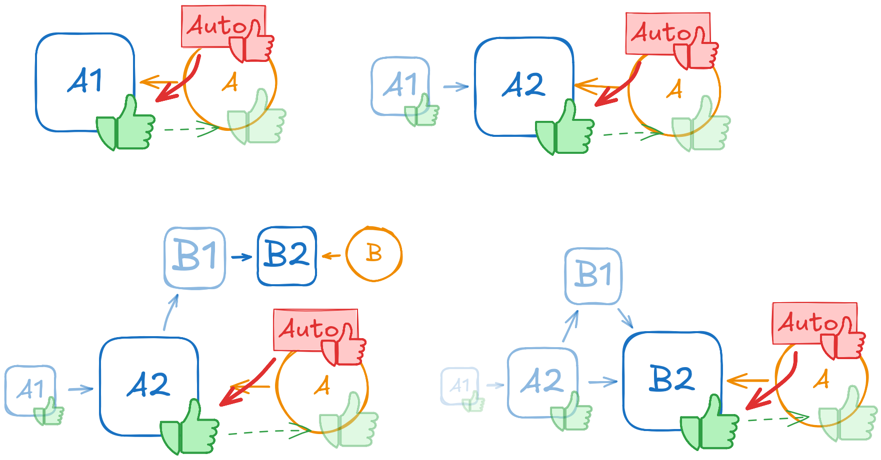



Fig. 6. Auto votes follow the head branch $A$, but not version $B_1$ of branch $B$. Once $B$ is merged into $A$, does $A$ point to $B_2$, and it gets the vote.

Second, for the maintainers of the requesting branch: they have leverage over the requested branch, and if the request is not accepted, and they add their branch as a new proposal, they keep the votes, giving them a head start and reducing the risk involved.

Third for the users themselves. They can shift their vote to something that *might* happen, reducing the time they have to be actively have to keep up with the development of the process.

As participants interact with the system and express their preferences, the system can provide feedback on the popularity of proposals, and the direction of the community. This can help to guide the development of the process, and to identify areas that need more attention. Users can be notified of changes in the status of proposals they are interested in, and can be encouraged to participate further in the process.

## 4 Comments & Issues

Participants can express themselves in three ways: proposals, change requests and votes. Proposals and change requests require more time and effort, while voting is quick, requiring only a single click to express support or opposition.

A common form of communication in the system is comments. Comments are short pieces of text attached to any entity, including other comments. This allows discussions to develop naturally into threads or trees.

A challenge arises when discussions occur in branches that evolve or are reused in forks. A comment made on a version of a branch may no longer be relevant to a fork, but it may still be relevant to an active change request based on that version. To solve this problem, comments in the system are contextualised by the version and branch they were made on. This creates an overlay of comments on top of the version graph (see Fig. 7), allowing users to track discussions across different versions and branches. Each comment is linked to a specific version within a branch, so it is possible to filter and group comments based on when and where they were added.

Filtering allows users to hide comments that are not relevant to their current branch or version, while grouping makes it easier to see comments from specific branches or drafts, depending on what the user wants to see. For example, in a fork that is branching away from the source, users

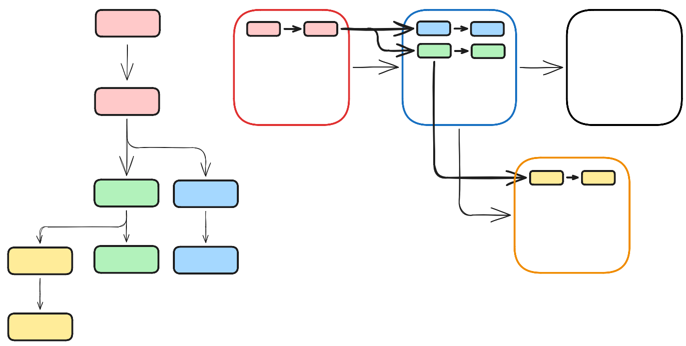


Fig. 7.  A comment tree forms an overlaying graph of comments on top of the version graph.

could hide all comments that are not part of the fork, or that are not predecessors of a comment within the fork.

### 4.1 Issues

There is a more formal variation of the comment, the issue, which draws inspiration from the issue systems found in version control systems such as GitHub or GitLab, but with modifications to fit the context of participation system, such as a tighter relationship with a specific version and branch. Issues allow participants to raise concerns, suggest improvements, or flag areas that need attention.

Issues are like an actionable, top-level comment. In addition to their text content, they can have a status, such as open, closed, or resolved. Participants can use those if they have a specific issue, that they need to track or resolve, but can not or do not want to create a full change request for it, yet. For example, an issue might highlight an unclear section of a proposal, or suggest an area for further research.

Since issues are more restricted in their position, and have more formal semantics, they act as anchor points for further discussions, utilizing common comments. When a new branch is created, the maintainer can decide whether to carry over existing issues, or discard them if they are no longer relevant.

As work on the proposal progresses, issues can be resolved by new versions, containing changes that address the issue. Once an issue has been addressed, it can be marked as closed. Closed issues, along with their comment threads, can be hidden from the current branch view, keeping the focus on the active areas of the proposal, while being archived for future reference.

Issues fill the gap between lightweight comments and more heavyweight change requests. The spread of options for users to participate (see Fig. 8) aims at providing tools for collaboration for different situations.

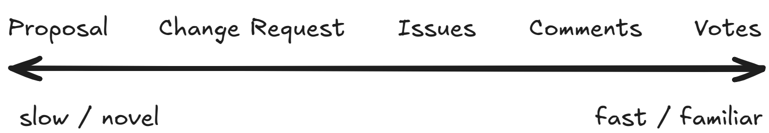



Fig. 8.  Scale of ways for participants to interact.

## 5 Conclusion

We explored how version control systems (VCS) can support collaborative decision-making. Using techniques commonly used in software development, we explored how similar mechanisms could be adapted to support collaborative proposal generation.

By introducing concepts such as immutable versions, branching and structured change requests, the system provides a scalable approach to collaborative proposal development. It ensures that all participants' contributions are preserved, while maintaining a coherent and traceable history. By providing structure to the process, it gives participants tools and guidance on how to contribute effectively. This structure also helps to manage the complexity of large-scale collaboration, making it easier to navigate and understand each other's contributions.

A major challenge in using the features presented here for collaborative decision-making is to ensure that they are easy to use for everyone, not just tech-savvy users. Tools such as Git, while well known in software development, can be confusing to people who are not familiar with them. This complexity can be a barrier to participation for less experienced participants. Therefore, features have been designed so that novice participants do not have to rely on them to participate. A novice participant does not need to know about versions in order to browse, vote and comment on proposals.

The proposed system will now need to be tested in user trials. This will help to identify usability issues and assess the effectiveness of the system in real-world scenarios. The system should be tested in different scenarios, from small expert groups to large public participation, to see how well it scales and is used in different contexts.

## References

[1] Jan Behrens, Axel Kistner, Andreas Nitsche, Björn Swierczek, and Interaktive Demokratie e.V (Eds.). 2014. *The Principles of LiquidFeedback* (1st edition ed.). Interaktive Demokratie e.V, Berlin.

[2] Geoffrey Litt, Sarah Lim, Martin Kleppmann, and Peter Van Hardenberg. 2022. Peritext: A CRDT for Collaborative Rich Text Editing. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–36.

[3] Geoffrey Litt, Paul Sonnentag, Max Schöning, Adam Wiggins, Peter van Hardenberg, and Orion Henry. 2024. Patchwork lab notebook: Version control for everything. https://www.inkandswitch.com/patchwork/notebook/. [Online; accessed 26-June-2024].

[4] Karissa Rae McKelvey, Scott Jenson, Eileen Wagner, Blaine Cook, and Martin Kleppmann. 2023. https://www.inkandswitch.com/upwelling/

[5] Matthew Weidner. 2023. Proposal: Versioned Collaborative Documents. In *Programming Local-first Software Workshop*. https://mattweidner.com/assets/pdf/versioned_collaborative_documents.pdf

# Chapter 6

# Measuring Progress to Support Cooperative Decision-Making Processes

Björn Ebbinghaus and Martin Mauve

“Measuring Progress to Support Cooperative Decision-Making Processes”

In: *HCI International 2026 Proceedings*. Vol. 15: *Social Computing and Social Media – Part I*. 18th International Conference, SCSM 2026. Held as Part of the 28th HCI International Conference, HCII 2026 (Montreal, Canada, July 26–31, 2026). Forthcoming

This chapter is about a metric that was developed to measure progress and coherence in collaborative decision-making processes. The motivation for this work arose from the need to evaluate the impact of iterative proposal-making and guide participants towards collaboration. The software platform, decide, presented in Chapter 4 aims to facilitate the development of better proposals by enabling users to merge proposals, i.e. form alliances or reach compromises. The platform already provides hints and call-to-actions to propose these merges between proposals with overlapping followers. Our aim is to provide feedback on the potential benefits of a merge, thereby motivating participants to move towards consensus instead of pursuing their individual goals.

Whether a situation in a procedure is good or bad, and how good or bad it is, is highly subjective. The goal is to provide a metric that aligns closely enough with participants' intuitions to earn their trust as a guide. While some situations in a decision-making process are easy to compare, others are not.

**Impact on the Thesis** This paper makes a contribution to the broader topic of collaborative decision-making processes. By introducing a measurable progress indicator, it aligns with the dissertation's goal of improving the practicality of decision-making. In this respect, it stands alone, as it focuses more on psychological influence than practical procedures.

## 6.1 Contribution

The paper provides a step-by-step explanation of the properties that make a suitable metric. To achieve this, a couple of assumptions are made about what constitutes a sensible metric for collaborative group decision-making processes. The initial assumptions are:

1. The metric is used live during the process.
2. The number of participants in the process is variable.
3. Each participant belongs to exactly one group of one or more participants who agree with each other.
4. There are two fixed points of known value.
   a) Having exactly one group in the process means that everyone agrees, which is the best possible situation and therefore receives the maximum score of 1.
   b) Having no groups with more than one participant means that everyone disagrees with everyone else, which is the worst possible situation and therefore has the minimum score of 0.

Based on these assumptions, further observations can be made. For example, if the best possible situation is when everyone agrees with everyone else, then it would make sense that the second-best situation would be when everyone except one participant agrees. This reasoning is used to derive a partial order of situations within a process, which is then used to inform the final metric.

However, the partial order does not indicate how much better one situation is compared to another. We compare different approaches, outlining their respective advantages and disadvantages, as well as the conditions under which they may struggle to represent intuition.

Ultimately, we arrive at a metric that is easy to calculate, meets expectations and can be adapted to different usage scenarios. For example, consensus may not be the goal of the process as a whole, but it may be sufficient for a limited number of groups.

**Personal Contributions** Björn Ebbinghaus developed the central ideas of the work. Martin Mauve acted as an advisor during the conception and writing stages.

# Measuring Progress to Support Cooperative Decision-Making Processes

Björn Ebbinghaus[1][0000−0002−5586−3327] and Martin Mauve[1]

Heinrich-Heine-Universität, Universitätsstraße 1, 40225 Düsseldorf, Germany
{ebbinghaus,mauve}@hhu.de https://cn.hhu.de

**Abstract.** In this paper, we argue that the participants in a cooperative decision-making process should be supported to work together towards a common goal, rather than to prioritise their individual goals. To achieve this, we propose using a metric to measure progress towards a common goal that participants can use for guidance. We derive the metric through various approaches of practical constraints and show problems that arise of each approach.

**Keywords:** group decision-making, consensus measure, social computing, online-participation

## 1 Introduction

Making decisions in large groups can be a significant challenge, requiring a high degree of coordination and cooperation between the participants. In many decision-making scenarios, individuals may unknowingly refrain from working together due to the lack of a common goal to rally behind. Therefore, we aim to improve decision-making processes by providing a common goal that is readily available through a measurable metric of progress.

We consider group decision-making (GDM) processes where an arbitrary number of participants cooperate to identify potential solutions to a given issue and reach a decision on it. Just the issue is specified; the participants themselves propose *decision alternatives* and decide upon them in the course of the process. Example for such a process is a community who utilize an online participation tool to brainstorm and decide on what to spend their funds on. In such a process, the participants signal their positions to each other constantly by changing their votes. These votes make it clear which proposals should receive attention during the process, and also determine its outcome. However, by communicating mainly in this way, it could unintentionally discourage participants from cooperating.

In decision-making processes, egocentric goals, such as maximizing the number of supporters for one's own alternative, can slow overall progress. Simple votes on alternatives prioritise individual victory and condition participants to pursue goals that are exclusively aligned with their own interests. This ignores the potential benefits of collaboration and can hinder overall progress. Rather than thinking about how to collaborate with supporters of competing alternatives, communication stalls and participants focus on how to improve their own alternative's *score*.

We believe that this problem lies with the design of the process rather than with the participants themselves. This follows from the recognition that the ultimate goal of a decision-making process is *not* just to reach a decision at any cost, regardless of quality. In the context of a public decision-making process, we assume that a good decision is one in which the maximum number of participants agree and have been involved in. The obvious optimum of such a process is a decision that all participants accept, i.e. when they reached consensus. We assume that the participants have an interest in finding a mutual solution, although we do not presuppose that a consensus always is possible.

Our aim is to improve the decision-making process by promoting a shared sense of responsibility for achieving the best possible outcome for all participants. To achieve this, we propose a metric of progress that is easy for participants to interpret. This metric can be used as a tool for informative deliberate decisional guidance for structuring and executing by the definitions of Silver [11].

## 2 Related Work

The ability to quantify and assess the progress or quality of a decision-making process has long been a challenge that researchers and practitioners have grappled with [7]. In the context of practical decision-making processes, e.g. in a business setting, measures exist to score the outcomes of the process and assess the extent of completion of predetermined goals, like with key performance indicators (KPIs)[5]. For example, one can measure the success of a decision by the increase in sales over a given quarter. In addition, progress within the decision-making process itself can be evaluated by the achievement of pre-defined milestones, such as the elimination of certain alternatives, because these milestones represent concrete goals for humans to strife to.

There exist approaches in academia, specifically in the area of fuzzy relations[3]. Spillman et al.[12, 1] presented a similar approach to the measure we will present. They measure the progress over the distance between the current state of a decision process and the respective best possible state (the consensus). Unlike our approach, they take into account fuzzy relations[13] to represent preferences between pairs of alternatives, whereas we omit preferences for the moment and have focused solely on the agreement to individual alternatives.

Our approach is compatible with theirs in that we consider a subset of their state model of the decision processes. While their measure is partially consistent with our approach, it does not meet all the requirements we set. As a consequence, our approach provides favourable results, and we will explore the reasons for this in Section 6.1. Whether we can apply our approach to determine progress to their fuzzy preference relations remains an open question for future research.

Previous research has taught us that the methodology used to determine progress affects how fast participants reach consensus[4, 8], or even how fast they approach it. Therefore, we believe it is important to develop more accurate measures of progress and benchmark them against each other to ultimately improve decision-making outcomes and promote effective collaboration.

## 3 Basics

To track the progress of a decision-making process, we need a metric that is easy for non-experts to understand. We propose a range of values from 0 to 1, where 0 represents the worst possible state of the decision-making process and 1 represents the best possible state. This fixed range allows participants to easily see how close they are to the best or worst possible state.

Since the goal is for participants to agree with each other, we will use the number of agreements as the basis for our first metric.

1. If no participants have agreed on anything, as it may be the case at the start of the process, it will be a score of 0, indicating the worst possible case.
2. If all participants have agreed and reached consensus, it will be a score of 1, the best possible case. That the process reaches consensus is not a requirement.

We score the state of a process at a given point in time, and we will call that a *situation*. A situation is a set of *decision alternatives*, which are disjunct, non-empty sets of participants who agreed with them. Note that we require a participant agree to exactly one alternative at a time.

For example, in a process with three participants $(p_1, p_2, p_3)$, the worst case scenario is where each participant supports a separate decision alternative, resulting in no agreement between the participants. We represent this as a set of three sets with cardinality 1: $\{\{p_1\}, \{p_2\}, \{p_3\}\}$. The best case would be $\{\{p_1, p_2, p_3\}\}$ where all three participants agree on one alternative. Note that we omit decision-alternatives without supporters.

Since we do not care about specific participants who agree to alternatives, we use just the cardinality of the alternatives and represent them in a descending vector of the number of supporters as shown in Table 1. For brevity, we may write $[4, 1^3] = [4, 1, 1, 1]$.

Table 1: Representations of all possible situations with $n = 3$ distinct participants.

| Set Representation | Vector Representation |
|---|---|
| $\{\{p_1\}, \{p_2\}, \{p_3\}\}$ | $[1, 1, 1]$ |
| $\{\{p_1, p_2\}, \{p_3\}\}$ | $[2, 1]$ |
| $\{\{p_1\}, \{p_2, p_3\}\}$ | $[2, 1]$ |
| $\{\{p_1, p_3\}, \{p_2\}\}$ | $[2, 1]$ |
| $\{\{p_1, p_2, p_3\}\}$ | $[3]$ |

## 4 Ranking of Situations

Consider a decision-process with $n = 4$ participants. It has five possible situations that need to be scored. We start by defining some constraints on how to score them. In the best case, where everyone agrees on one alternative ([4]), we assign the value 1. In the worst case, where everyone agrees to a different alternative ($[1, 1, 1, 1]$), we assign the value 0. The remaining three possibilities are $[2, 1, 1]$, $[2, 2]$, and $[3, 1]$, and we need to decide how to score these cases.

We start by creating an order of situations where we can say *situation X is definitely preferable to situation Y*.

Let us look at the example again, and think about how the situations could have come to be. We say that $[2, 1, 1]$ is one *step* away from $[1, 1, 1, 1]$. It is the only situation that can arise if a single participant changes their opinion. Therefore, we can say with certainty that $[2, 1, 1]$ is the next better situation after $[1, 1, 1, 1]$. It must therefore rank higher than $[1, 1, 1, 1]$, but lower than any other situation. Accordingly, we say that $[2, 2]$ and $[3, 1]$ are both one more step away from $[2, 1, 1]$, since these are the only situations that progress the process and can arise from changing a single opinion. They will have to rank higher than $[2, 1, 1]$, but less than [4].

Now we need to decide which of the remaining situations has progressed further: $[2, 2]$ or $[3, 1]$? In both cases, two participants changed their opinion. They would rank the same, if we just consider how much agreement has already been reached or, to phrase it differently, how much energy was invested into the process to reach the situation.

If we not just take into account the participants who have already changed their opinion but also consider how many opinions still need to change to reach consensus, we can deduce that situation $[3, 1]$ is better than $[2, 2]$. This is because $[3, 1]$ is just one step away from the optimal situation [4], implying that it should rank higher than all other situations except [4]. Figure 1 shows these relations between the situations.

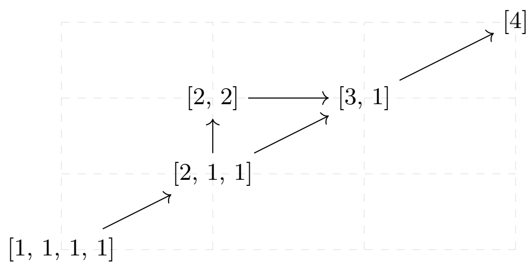



Fig. 1: All possible situations of a decision-process with four participants. The arrows show how a situation can change towards consensus when a single participant changes their opinion.

With this ranking, through made and missing agreements, we get the so-called *dominance order*[2] of situations, a partial order of the set of partitions for a given natural number.

**Definition 1.** *For two situations with the same number of participants, a situation* dominates *another if the sum of its best $i$ alternatives is always greater or equal than the corresponding sum of the other situation.*

*Example 1.* $[3,1]$ dominates $[2,2]$ because $3 \geq 2$ and $3+1 \geq 2+2$.

For situations that dominate other situations, and thus transitively dominate all previous situations, determining the order is trivial. A dominant situation is more advanced in the process than the situation it dominates.

However, with six or more participants it can happen that one situation does not dominate the other and vice versa. Consider for example $[3,3]$ and $[4,1,1]$.

*Example 2.*

$$\begin{array}{cl} [3,3] \text{ vs } [4,1,1] & \\ 3 < 4 & [4,1,1] \text{ dominates} \\ 3+3 > 4+1 & [3,3,0] \text{ dominates} \\ 3+3+0 = 4+1+1 & \\ \Rightarrow \textit{Neither situation dominates the other.} & \end{array}$$

In this case, the order is not as easy to determine and may be subjective or depends on the nuances of the process in question. One could argue that $[4,1,1]$ is better than $[3,3]$ based on the number of remaining participants who need to reconsider in order to reach consensus. On the other hand, $[3,3]$ has fewer alternatives overall, and there has been more agreement among participants in general, which can be more desirable.

An inflated example $[500,500]$ vs $[501,1,1,\dots]$ illustrates the uncertainty even more. The situation with 501 participants is technically closer to the goal, but few would deny the effort made for the total number of agreements in the situation $[500,500]$.

We will see later that our final metric does not favour the number of alternatives or the strength of the leading alternative. For example, the score of situation $[4,1,1,1]$ is halfway between $[3,2,2,1]$ and $[3,3,1,1]$. We think this is sufficient.

The following Figure 2 shows the graph of situations with seven participants and the relations to their neighbours, the arrows point to the dominating situation. Dominance is transitive. Different situations that can not reach each other through a directed path do not dominate each other, e.g. $[4,3]$ and $[5,1,1]$. But in some situations a participant can change their opinion such that the process stays in the same situation. For example: In situation $[4,3]$, a participant could decide to switch from the leading alternative to the weaker alternative, resulting in $[4,3]$ again.

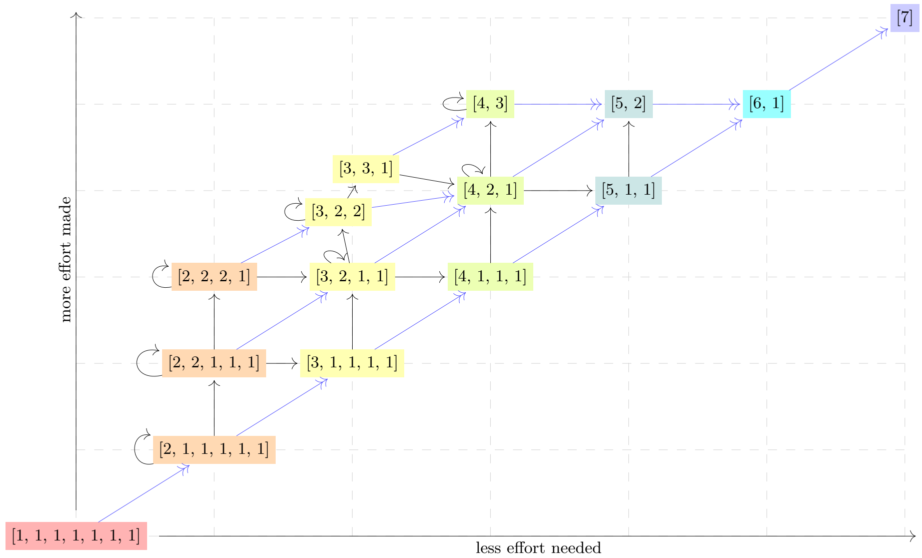



Fig. 2: Situations in a decision-making process with seven participants. Blue, double-headed arrows are the optimal next step a situation can take. Note that situations $[3, 2, 2]$ and $[3, 3, 1]$ are technically in the same place.

## 5 Scoring

We have identified the best and worst case scenarios and a partial order of the situations. The next question is how to assign displayable values to the alternatives. If $score([1, 1, 1, 1]) = 0$ and $score([4]) = 1$, what score should we assign to $[3, 1]$?

For a naive approach, we can take the order of the situations and enumerate them through the interval $[0, 1]$. With three participants the values would be $[1, 1, 1] = 0$, $[2, 1] = 0.5$ and $[3] = 1$.

Table 2: All situations with four participants together with their score by enumeration.

| Situation | Rank | Normalised Score |
|---|---|---|
| $[1, 1, 1, 1]$ | 0 | 0.00 (0/4) |
| $[2, 1, 1]$ | 1 | 0.25 (1/4) |
| $[2, 2]$ | 2 | 0.50 (2/4) |
| $[3, 1]$ | 3 | 0.75 (3/4) |
| $[4]$ | 4 | 1.00 (4/4) |

As Table 2 shows, each situation has a unique score that reflects its position in the ranking. The scores increase as the situation improves and decrease as it worsens, but the method has significant drawbacks.

Firstly, in order to determine the rank of a situation, it is necessary to know how the number of situations before and after it. This is not a trivial problem. In fact, it is not even easy to find the number of possible situations for a given number of participants. That number grows exponentially. With 50 participants there are 204 226 possible situations, with 100 there are already 190 569 292[1]. In the worst case all situations would have to be determined and counted. This is not feasible.

Secondly, the distance between two situations depends on the number of situations between them. In Table 2 we see that the situations $[2,1,1]$ and $[3,1]$ are each 0.25 away from their predecessor and successor. The distance between these two situations, however, is twice as large at 0.5. In principle, however, we would expect the situations on the *direct path* between the worst and the best case ($[1,1,1,1] \rightarrow [2,1,1] \rightarrow [3,1] \rightarrow [4]$) to be equidistant from each other, or at least to change steadily in one direction. This problem gets worse as the number of participants increases. With 100 participants, the distance from $[1^{100}]$ to $[2,1^{98}]$ and $[99,1]$ to $[100]$ is tiny, while the distance from $[50,1^{50}]$ to $[51,1^{49}]$ is huge.

### 5.1 Score Based on Geometric Distance

Another family of metrics that is easy to calculate comes from the situation transition graphs we used in the previous section (Figures 1 and 2). One way to measure progress is to determine the length of the path in the graph between a situation $v$ with $n$ alternatives to consensus. This is trivial, as it is simply the sum of all participants who have not yet agreed with the leader:

$$d_{consensus}(v) = \sum_{i=2}^{n} v_i$$

This corresponds to the number of participants who have agreed on the best alternative:

$$d'_{consensus}(v) = n - d_{consensus}(v) - 1 = v_1 - 1$$

This can be a valid metric by itself if we are just interested in the winning alternative. But we would be ignoring any progress made in the other alternatives, and even ignoring the total number of remaining alternatives. By ignoring progress in other alternatives, the score will only change if the leading alternative improves or worsens. Thus, especially in large processes, there will often be a situation where the actions of an individual or even whole groups have no impact at all on the progress towards more agreement.

We can improve on this by considering not just the steps needed to reach consensus, but also the steps already taken from the start. Again, this is trivial

[1] https://oeis.org/A000041/list

to determine; any alternative in a situation that has more than one participant adds one step away from complete disagreement.

$$d_{dissent}(v) = \sum_{i=1}^{n}(v_i - 1)$$

Figure 3 illustrates this, using the two distances as x and y coordinates for the placement of the situations.

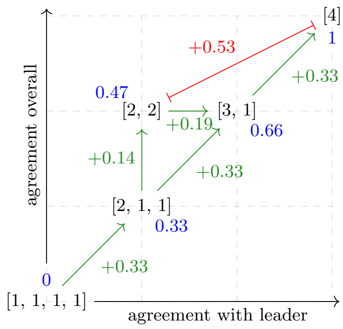



Fig. 3: All situations (black) with four participants and the change in Euclidean distance/score (green/red: change, blue: score) between them. The scores are normalised, such that the distance to the consensus is 1.

Finally, since we want a single score, we can calculate the distance between two situations using any distance metric we like. For example, we can use the Euclidean metric to calculate the (normalised) distance between the situations $[2, 2]$ and $[4]$ to be 0.47 (see Table 3).

$$score(v) = 1 - \sqrt{d'_{dissent}(v)^2 + d_{consensus}(v)^2}$$

Table 3: Scores for situations with four participants.

| Situation $(v)$ | $d'_{consensus}(v)$ | $d_{dissent}(v)$ | $score(v)$ |
|---|---|---|---|
| $[1, 1, 1, 1]$ | 0 | 0 | 0 |
| $[2, 1, 1]$ | 1 | 1 | 0.33 |
| $[2, 2]$ | 1 | 2 | 0.47 |
| $[3, 1]$ | 2 | 2 | 0.66 |
| $[4]$ | 3 | 3 | 1 |

Although we have improved the metric, the problem remains that, as the number of participants increases, the score for several situations could be the same. An example would be the situations $[4, 3, 1]$ and $[4, 2, 2]$. If a participant changes their opinion and moves the process from one situation to another, neither the total number of alternatives nor the number of agreements in the leader changes, so the score remains the same. But $[4, 3, 1]$ dominates $[4, 2, 2]$ and should therefore have a higher score.

## 6 Score by Top-N-Encoding

Based on the idea that the number of participants needed to reach a consensus is important and can be used to measure progress, we can iterate. Rather than considering only the number of participants needed for consensus—a situation with only one alternative—we also consider the number of participants needed to reach situations with fewer alternatives.

Consider the two situations $[3, 3, 1]$ and $[3, 2, 2]$. Both cases need four participants to reach consensus, but they differ in their remainders $[..., 3, 1]$ and $[..., 2, 2]$. If we now consider these remainders, we can make the same distinction as before. Thus, we consider the number of participants needed to reach a situation with two alternatives. We continue in the same way for situations with 3, 4, ..., $n - 1$ alternatives, by considering the components from the 3rd, 4th, $\ldots$, $(n - 1)$th place on. We can encode the number of missing participants in a new vector of size $n - 1$. The encoding for $[3, 3, 1]$ is therefore $[4, 1, 0, \ldots]$. Similarly, the encoding for $[4, 2, 2]$ is $[4, 2, 0, \ldots]$ (see Table 4).

$$encoding(v) = \left[ \sum_{i=2}^{n} v_i, \sum_{i=3}^{n} v_i, \ldots, v_n \right]$$

Table 4: Selected situations of seven participants with the number of participants needed to reach a situation with at most $n$ alternatives.

| | Missing agreements for situation with $\leq n$ alternatives ($encoding(v)$) | | | | | | |
|---|---|---|---|---|---|---|---|
| Situation ($v$) | $n = 1$ | 2 | 3 | 4 | 5 | 6 | |
| $[1, 1, 1, 1, 1, 1, 1]$ | [ 6 | 5 | 4 | 3 | 2 | 1 | ] |
| | … | | | | | | |
| $[3, 3, 1]$ | [ 4 | 1 | 0 | 0 | 0 | 0 | ] |
| $[3, 2, 2]$ | [ 4 | 2 | 0 | 0 | 0 | 0 | ] |
| | … | | | | | | |
| $[7]$ | [ 0 | 0 | 0 | 0 | 0 | 0 | ] |

With this encoding we can determine the geometric distance to consensus as before. In this way, the effort put into agreeing on non-winning alternatives helps to reduce the distance to consensus, albeit to a lesser extent. The consensus always has an encoding of $[0, \ldots, 0]$, so we can just take the length of the encoded vector as the distance to the consensus.

$$d_{consensus}(v) = \|encoding(v)\|_2$$

We normalise the score to $[0, 1]$ and reverse the distance, because so far we have determined the distance to the consensus, but we want the consensus to have a score of 1. This gives us the following function, where $d_{max}(n)$ is the distance from the worst case to consensus for $n$ participants.

$$score(v) = 1 - \frac{d_{consensus}(v)}{d_{max}(n)}$$

This method of scoring maintains the *dominance order* and is easy to calculate. It is still possible for two situations to receive the same score (e.g. $[4, 1, 1, 1, 1]$ and $[3, 3, 1, 1]$), but in that case the dominance of them will always be undetermined. This means it will never happen that the situation of a process changes, but the score does not, which would be contrary to intuition.

An additional advantage of this approach is that it is trivial to set other goals for a process. For example, instead of a consensus, a Top-3 could suffice. In this case, the total includes just the alternatives from the fourth position onwards. Otherwise, the calculation of the score would remain unchanged, with the score for all situations with three or fewer alternatives now being 1. In this way it is possible to fine-tune the desired outcome of a decision-making process.

In summary, we have explored different methods for assigning values to alternatives to determine their scores. The naive approach of normalising the rank order had drawbacks, such as difficulty in determining the rank and inconsistent distance between situations. We then introduced a scoring method based on Euclidean distance, which takes into account both progress towards consensus and distance from dissent. Although it improved the metric, it had limitations such as situations with the same score and the inability to capture progress in non-winning alternatives.

To conclude, we presented a scoring method using Top-N encoding, which took into account the number of participants needed to reach situations with fewer alternatives. This method allows fine-tuning of the desired outcome and maintains dominance order while being easy to compute. Overall, the Top-N encoding scoring method provides a valuable approach to determining progress in decision-making processes.

## 6.1 Comparison to Spillman et al.

Earlier in Section 2 we said that Spillman et al.[12, 1] have a similar approach to ours. Our vector representation of a situation can easily be translated into their fuzzy preference matrix and their distance calculated. In fact, in the end, the score is the Euclidean norm of our situation vector.

*Example 3.*

$$\begin{bmatrix}2 & 1 & 0\end{bmatrix} \Rightarrow \left(2 \cdot \begin{bmatrix}0 & 0 & 0\\ 1 & 0 & \frac{1}{2}\\ 1 & \frac{1}{2} & 0\end{bmatrix} + 1 \cdot \begin{bmatrix}0 & 1 & \frac{1}{2}\\ 0 & 0 & 0\\ \frac{1}{2} & 1 & 0\end{bmatrix} + 0 \cdot \begin{bmatrix}0 & \frac{1}{2} & 1\\ \frac{1}{2} & 0 & 1\\ 0 & 0 & 0\end{bmatrix}\right) \div 3 = \begin{bmatrix}0 & \frac{1}{3} & \frac{1}{6}\\ \frac{2}{3} & 0 & \frac{1}{3}\\ \frac{5}{6} & \frac{2}{3} & 0\end{bmatrix} = R$$

$$D(R^*, R) = \frac{1}{3} = \textit{normalized}\left(\|[2,1,0]\|_2\right)$$

We argue that our measure has some advantages over this Euclidean norm. Figure 4 shows that the simple Euclidean distance does not change between the situations $[2,2,2]$ and $[3,1,1,1]$ or $[3,3]$ and $[4,1,1]$, although in both situations the leading alternative has become stronger and the competitor weaker. Preferring a larger majority to weaker competitors also comes closer to the idea of the *soft consensus*[9, 6, 10], according to which *hard* consensus is unrealistic in practice and a situation where *most of the participants agree* will have to suffice.

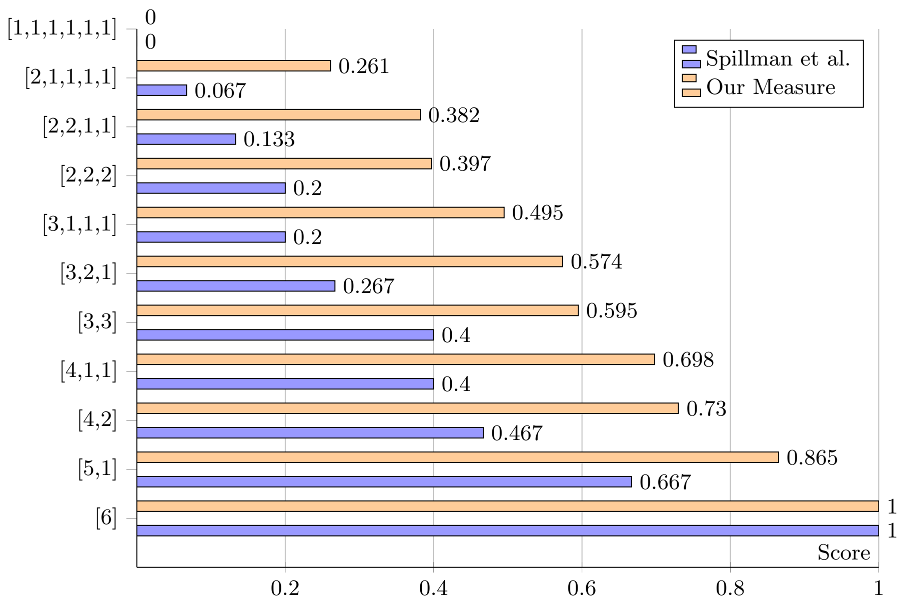



Fig. 4: Comparison of our measure with Spillman et al. in a decision-making process with six participants.

## 7 Conclusion & Future Work

In this paper we have presented a possible metric to measure progress in a decision-making process. We have discussed its properties and proposed its applications. The metric can be used on its own or as an extension of other metrics.

Further research can focus on the evolving nature of decision-making processes by examining the changing number of participants and their agreement over time. For example, based on experience of past processes, we expect that a decision-making process would start with a high score of progress, which would decline as more and more participants entered the process and contributed different ideas. Progress would increase further when participants start to agree again. An exploratory phase without scoring may be necessary until a certain number of participants are present.

Another worthwhile direction would be to explore how the progress of the process behaves if we consider that certain group dynamics develop in such processes and whole groups/parties change their opinions together, rather than looking at individual participants. This has the potential to bring the metric closer to real world scenarios.

We recognize that such a metric can be highly opinionated and may be more or less appropriate depending on the characteristics of the decision-making process. But the metric provides room for extension and fine-tuning. We have already touched on some of these in the previous section.

Furthermore, it remains to be seen whether the metric aligns with the participants' intuition and whether a process with it as a common goal leads to more cooperation and a better outcome.

# Bibliography

[1] Bezdek, J.C., Spillman, B., Spillman, R.: A fuzzy relation space for group decision theory. Fuzzy Sets and Systems **1**(4), 255–268 (Oct 1978), ISSN 01650114, https://doi.org/10.1016/0165-0114(78)90017-9

[2] Brylawski, T.: The lattice of integer partitions. Discrete Mathematics **6**(3), 201–219 (1973), https://doi.org/10.1016/0012-365X(73)90094-0

[3] Cabrerizo, F.J., Pérez, I.J., Chiclana, F., Herrera-Viedma, E.: Group Decision Making: Consensus Approaches Based on Soft Consensus Measures. In: Torra, V., Dahlbom, A., Narukawa, Y. (eds.) Fuzzy Sets, Rough Sets, Multisets and Clustering, vol. 671, pp. 307–321, Springer International Publishing, Cham (2017), ISBN 978-3-319-47556-1 978-3-319-47557-8, https://doi.org/10.1007/978-3-319-47557-8_18

[4] Chiclana, F., García, J.T., Del Moral, M., Herrera-Viedma, E.: Analyzing Consensus Measures in Group Decision Making. Procedia Computer Science **55**, 1000–1008 (2015), https://doi.org/10.1016/j.procs.2015.07.103

[5] Fitz-Gibbon, C.T. (ed.): Performance Indicators. No. 2 in BERA Dialogues, Multilingual Matters, Clevedon, Avon, England ; Philadelphia (1990), ISBN 978-1-85359-093-1 978-1-85359-092-4

[6] Kacprzyk, J., Fedrizzi, M., Nurmi, H.: "Soft" Degrees of Consensus Under Fuzzy Preferences and Fuzzy Majorities. In: Zimmermann, H.J., Kacprzyk, J., Nurmi, H., Fedrizzi, M. (eds.) Consensus Under Fuzziness, vol. 10, pp. 55–81, Springer US, Boston, MA (1997), ISBN 978-1-4613-7908-9 978-1-4615-6333-4, https://doi.org/10.1007/978-1-4615-6333-4_4

[7] Kline, J.A.: Orientation and group consensus. Central States Speech Journal **23**(1), 44–47 (Mar 1972), https://doi.org/10.1080/10510977209363091

[8] Labella, Á., Liu, Y., Rodríguez, R., Martínez, L.: Analyzing the performance of classical consensus models in large scale group decision making: A comparative study. Applied Soft Computing **67**, 677–690 (Jun 2018), ISSN 15684946, https://doi.org/10.1016/j.asoc.2017.05.045

[9] Loewer, B., Laddaga, R.: Destroying the consensus. Synthese **62**(1), 79–95 (Jan 1985), ISSN 0039-7857, 1573-0964, https://doi.org/10.1007/BF00485388

[10] Perez, I.J., Cabrerizo, F.J., Alonso, S., Chiclana, F., Herrera-Viedma, E.: Soft Consensus Models in Group Decision Making. In: Calvo Sánchez, T., Torrens Sastre, J. (eds.) Fuzzy Logic and Information Fusion, vol. 339, pp. 135–153, Springer International Publishing, Cham (2016), ISBN 978-3-319-30419-9 978-3-319-30421-2, https://doi.org/10.1007/978-3-319-30421-2_10

[11] Silver, M.S.: Decisional Guidance for Computer-Based Decision Support. MIS Quarterly **15**(1), 105 (Mar 1991), https://doi.org/10.2307/249441

[12] Spillman, B., Bezdek, J., Spillman, R.: Development of an instrument for the dynamic measurement of consensus. Communications Monographs **46**(1), 1–12 (1979), https://doi.org/10.1080/03637757909375986

[13] Zadeh, L.: Fuzzy sets. Information and Control **8**(3), 338–353 (Jun 1965), ISSN 00199958, https://doi.org/10.1016/S0019-9958(65)90241-X

# Chapter 7
# Discussion

This chapter presents a collection of issues and design choices that need to be considered in relation to the general theme. It provides a broader view of the insights and considerations that have emerged and guided this work.

While the issues discussed are not necessarily novel and may have solutions in certain circumstances, they have arisen in the context of this work due to the chosen assumptions and objectives, or simply because of the constraints that large numbers of untrained users place on practical solutions.

## 7.1 Proposal Proliferation

As discussed in Chapter 4, providing decision-making procedures in which many people can participate equally and are encouraged to iterate on proposals naturally leads to numerous very similar proposals. Each iteration, even with a tiny change, is essentially a new proposal. This presents a problem when the aim is to treat each participant's voice (approximately) equally, as it results in the need to consider each proposal.

It is not possible to present an unfiltered view of all the proposals and their potentially numerous variations. Participants would simply be overwhelmed, and most of the energy they would be willing to put into a process would be wasted on the overhead of navigating the cumbersome process. This motivates some form of organization or facilitation that somehow decides which versions of which proposals get more attention than others.

This is not a problem specific to this scenario, but the number of small refinements exacerbates it. In a traditional large-scale participation process involving a lot of user content, it is expected that the proposals will differ enough that participants will be able to distinguish them clearly, and will be able to express clear preferences that can be used to decide which proposals to show to others and which to hide. Iteration of proposals can lead to an increased problem of *clones* (Tideman, 1987). This problem arose in Chapter 4 during one of the field experiments, where participants had only a single vote and were therefore reluctant to risk agreeing or forming new proposals. However, the problem with cloning is not only one of choice in a vote, but also of user behaviour during the process. Long before proposals are presented as candidates in any form of voting, participants need to decide where to focus their efforts.

*Example:* A proposal is improved by a new proposal. Participants who have an idea for a change, regardless of what was changed in the improvement, must decide whether to apply this new change to the original proposal, the new proposal, or both. However, applying the new change to both would only make the problem worse.

The system proposed in Chapter 5 deals with this problem in the usual way: people decide what is relevant and should be the focus of the process. Having maintainers guide proposal branches provides a more focused process, containing proposal sprawl to some extent. The maintainers effectively determine how the process appears to those who are unfamiliar with it or not interested in becoming more involved.

This is a compromise in that it elevates some participants over others based on merit, which Chapter 5 tries to achieve without locking participants in. However, it is still a compromise. However, the underlying problem of a solution space that is growing so rapidly that it is impossible for participants to overlook it is still present.

This is also similar to how real-world processes deal with this complexity. People form parties to consolidate their views and to divide and conquer large, complex issues. This is probably the best solution, even in modern times, where we do not need to send messengers on horseback to represent our interests. Nevertheless, recent developments such as Liquid Democracy (Interaktive Demokratie e.V, 2014; Paulin, 2020) show that today's enhanced communication capabilities are capable of supporting powerful tools that enhance the idea of the static party and allow for more flexible (i.e. liquid) solutions.

The (expected) explosion of proposals is a problem that cannot be solved easily. A compromise has to be found between the sovereignty of the individual participant and the practical problem of managing many opinions. A solution would have to be transparent and comprehensible, so that participants do not get the impression of an unfair or even censoring system.

## 7.2 Complex Preferences

Section 7.1 has already mentioned the issue of proposal clones as an issue of usability and clarity for users. In a voting context, this raises the familiar problems of potential vote splitting. However, the incremental modification of proposals adds another dimension to the problem. Proposals are not only very similar to each other, they are also causally related, i.e. some proposals are descendants of others, while others are siblings. This makes it difficult to cast meaningful votes in a live process.

Consider two proposals. A parent proposal, $A$, and an improvement (child) proposal, $B$. It is assumed that, for large public participation processes to be successful, any type of voting must be as simple as possible. In the simplest case, users can approve proposals. This might be fine, were it not for the relationship between proposals and the iterative nature of the processes, where voting is not only a means of making a decision, but also a useful signalling tool to help participants find and focus on the right proposals. If a participant voted for both $A$ and $B$ during the process, this would signal general interest in the family of proposals, but would not indicate the desired *movement* of the proposals. This requires the use of a preferential voting system to distinguish between a true improvement from $A$ to $B$ and a regression that the user does not prefer but is willing to accept.

The inability to determine whether a change to a proposal was an improvement for a user or not also leads to a major usability problem. When a new proposal is made, the new proposal cannot receive votes from existing users who voted for the original proposal. Users cannot be expected to keep up with every development of proposals, so the development of a proposal runs the risk of leaving early proposals with a disproportionate number of votes, which undermines the goal of using votes for signalling.

Since the idea of incremental proposals is about improvement, and voting in this context can be used for signalling, it is reasonable to assume that participants will feel the need to express disagreement with the status quo while agreeing with the general direction of the proposal. For example, a proposal may be on the right track, but needs some refinement to meet users' needs. This motivates conditional preferences as a means for users to express their desired direction for proposals (e.g. *I approve this proposal if x is changed.*). Chapter 5 offers a possible solution to this, in that versions of proposals are initially not really globally visible proposals, but more like drafts that can still be voted on. This allows votes to be cast without having a global impact on the process. Once these drafts are accepted and become full proposals, the votes are realized in the overarching process. This allows users to cast conditional votes and reduces the complexity of keeping up with development by enabling them to vote on upcoming proposals. The ability to express conditional preferences is also a prerequisite for another important aspect of group decision-making: Negotiation.

## 7.3 Negotiation / Compromises

In large-scale public decision-making processes, it is rare for there to be a single point of view. There are usually multiple stakeholders with interests that range from overlapping to mutually exclusive. Depending on the objectives of the decision-making process, it may be necessary to work with competing ideas, which is a complex task even with only two parties.

Extending this to the kind of processes examined in this work poses a significant challenge, given the lack of organization among the participants. Therefore, the system must natively support negotiations; otherwise, participants will organize themselves on a large scale through back channels.

Successful negotiations require participants to accept compromises, which may not come naturally in the more impersonal environment of the internet and may even conflict with a system designed for participants to improve proposals. Participants are expected to make worse proposals for themselves in order to achieve a higher goal.

As mentioned above, Chapter 5 provides a solution for rudimentary negotiation solution by enabling a system of offers and counter-offers that both parties can iterate until a solution is found. At no point are the parties required to commit to anything until the final solution is ready. The system even allows multiple offers to exist simultaneously, enabling in-depth negotiations on specific issues to take place in separate groups, thereby reducing the cognitive load on participants.

However, this solution is far from perfect and may only be theoretically feasible. A more concrete model of parties involving designated negotiators might be a better solution, even though it would undermine some of the goals set out in this work. This model would allow for more efficient synchronous communication, i.e. chatting.

# Chapter 8
# Conclusion

The goal of this dissertation was to explore how online participation systems can be designed to better support group decision-making processes, with a focus on enabling participants to collaboratively develop, refine and build upon proposals in a structured and scalable way. It aimed to address the limitations of existing tools, such as comment sections and voting interfaces, which often collect input without facilitating mutual understanding or consensus building. Through investigating system design choices, creating new tools and evaluating their effectiveness, this work sought to identify key requirements for developing transparent, intuitive and adaptable platforms that promote meaningful online deliberation and improve the quality and inclusivity of collective outcomes.

## 8.1 Main Results

This thesis examined ways in which online systems could better support collaborative decision-making. The focus was on enabling participants to express their opinions and develop and refine proposals together.

The first system (*decide3* in Chapter 4) allowed proposals to evolve through iterations. Participants could build on each other's ideas; however, as the number of proposals increased, the process became difficult to manage. Without a clear structure, the system became overwhelming and difficult to understand.

To address this issue, a second system was developed in Chapter 5, incorporating concepts from version control systems. This made proposals editable through a controlled, versioned structure, allowing more flexibility while keeping the process organized. Basic users could participate as before, while more experienced users had access to advanced tools. This helped to balance openness with ease of use.

Both systems demonstrate that the development of proposals can be made both collaborative and structured. However, effective support requires clear mechanisms for managing complexity. The design must help users stay oriented and prevent the process from becoming fragmented.

This work demonstrates that more effective group decision-making can be facilitated by better tools that enable participants to develop ideas rather than simply selecting from them. The results suggest that structured collaboration can improve clarity and engagement in group decision-making processes. However, they must consider the cognitive load imposed by increasing complexity in their design.

The real-world experiments from Chapter 4 and Section 3.2 showed how important the participants' perception of the process is. Originating from that and the need to quantify the results of the proposed techniques from this thesis the metric in Chapter 6 was introduced to measure a group's progress towards consensus. This metric can provide real-time feedback and actionable guidance to help users assess whether their contributions are converging or diverging. The metric was developed based on practical considerations and shaped by participants' intuitive expectations.

## 8.2 Future Work

The systems and concepts presented in this thesis have laid the groundwork for a more structured and scalable approach to online participation. However, there are still several areas that require further research and development.

Future work should focus on testing the new systems in real-world settings. It remains to be seen whether the structure of the system promotes or limits participation in large-scale environments. Do the enhancements provided work as intended in large-scale, high-stakes decision-making scenarios? Would participants be willing to invest effort and trust in such a novel system? Additional guidance tools, such as automatic, proactive suggestions or similarity detection by the system, could also enhance collaboration.

While decisions are inherently complex, they should be made as easy to understand as possible for participants. A distinctive feature of both iterative proposal systems is the traceability of proposal development. Transparency and comprehensibility are crucial for participants to accept the process and trust the collective decisions made.

The proposed metric for measuring progress in decision-making opens up several avenues for research. Its effectiveness in guiding users and reflecting group dynamics depends on its ability to capture meaningful aspects of participation. Future work could examine how the metric evolves as the process unfolds, particularly in response to changes in the number of participants or the emergence of group-level dynamics (e.g. shifts in party alignment). It is also necessary to test whether the metric aligns with participants' intuition and whether using it as a visible guide during deliberation improves outcomes.

## Closing Thoughts

Although today's technologies offer us powerful communication tools, they are far from being fully developed and only scratch the surface of their potential. As with many things, humans are not naturally capable of achieving this alone, but we have demonstrated our ability to invent creative solutions to almost any problem. This thesis is intended to be a small step towards a society in which everyone's wishes and needs are reflected in transparent decisions, allowing humanity to realize its full potential in a bright future.

# Bibliography


Davies, Todd and Seeta Peña Gangadharan, eds. (2009). *Online Deliberation: Design, Research, and Practice*. CSLI Lecture Notes no. 182. Stanford, Calif: Center for the Study of Language and Information. ISBN: 978-1-57586-554-6 (cit. on p. 5).

Ebbinghaus, Björn (May 2019). "Decision Making with Argumentation Graphs". Master's Thesis. Heinrich Heine University Düsseldorf. DOI: 10.13140/RG.2.2.12515.09760/1 (cit. on pp. 13, 17, 35).

Ebbinghaus, Björn and Martin Mauve (Sept. 2020). "Decide: Supporting Participatory Budgeting with Online Argumentation". In: *Computational Models of Argument. Proceedings of COMMA 2020*. Comma 2020 (Virtual Event). Ed. by Henry Prakken, Stefano Bistarelli, Santini Francesco, and Carlo Taticchi. Vol. 326. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 463–464. ISBN: 978-1-64368-107-8. DOI: 10.3233/FAIA200535 (cit. on pp. 14, 70).

Ebbinghaus, Björn and Martin Mauve (June 2022). "Better Decision-Making Through Collaborative Development of Proposals". In: *HCI International 2022 Proceedings*. Vol. 26: *HCI in Business, Government and Organizations*. 9th International Conference, HCIBGO 2022. Held as Part of the 24th HCI International Conference, HCII 2022 (Virtual Event, June 26–July 1, 2022). Ed. by Fiona Fui-Hoon Nah and Keng Siau. Lecture Notes in Computer Science 13327. Springer, pp. 11–23. ISBN: 978-3-031-05544-7. DOI: 10.1007/978-3-031-05544-7_2 (cit. on pp. 17, 35, 70).

Ebbinghaus, Björn and Martin Mauve (2025). *Version Control for Structuring Collaborative Decision-Making*. Technical Report: Heinrich Heine University Düsseldorf, p. 9. URN: urn:nbn:de:hbz:061-20250715-102007-0. URL: https://docserv.uni-duesseldorf.de/servlets/DocumentServlet?id=70154 (cit. on pp. 33, 70).

Ebbinghaus, Björn and Martin Mauve (July 2026). "Measuring Progress to Support Cooperative Decision-Making Processes". In: *HCI International 2026 Proceedings*. Vol. 15: *Social Computing and Social Media – Part I*. 18th International Conference, SCSM 2026. Held as Part of the 28th HCI International Conference, HCII 2026 (Montreal, Canada, July 26–31, 2026). Ed. by Adela Coman and Simona Vasilache. Lecture Notes in Computer Science 16715. Springer. Forthcoming (cit. on pp. 45, 70).

Interaktive Demokratie e.V, ed. (2014). *The Principles of LiquidFeedback*. eng. 1st edition. Berlin: Interaktive Demokratie e.V. ISBN: 978-3-00-044795-2. URL: https://liquidfeedback.com/en/book-the-principles-of-liquidfeedback.html (cit. on pp. 6, 61).

Klein, Mark (2007). *How to Harvest Collective Wisdom for Complex Problems: An Introduction to the MIT Deliberatorium.* Technical Report: DOI: 10.13140/RG.2.2.32743.24489 (cit. on p. 8).

Klein, Mark (May 2011). "The MIT Deliberatorium: Enabling Large-Scale Deliberation About Complex Systemic Problems". In: *2011 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 161–161. DOI: 10.1109/CTS.2011.5928678 (cit. on p. 8).

Klein, Mark and Luca Iandoli (2008). "Supporting Collaborative Deliberation Using a Large-Scale Argumentation System: The MIT Collaboratorium". In: *SSRN Electronic Journal.* ISSN: 1556-5068. DOI: 10.2139/ssrn.1099082 (cit. on p. 8).

Kling, Christoph Carl, Jerome Kunegis, Heinrich Hartmann, Markus Strohmaier, and Steffen Staab (2015). *Voting Behaviour and Power in Online Democracy: A Study of Liquidfeedback in Germany's Pirate Party.* DOI: 10.48550/ARXIV.1503.07723 (cit. on p. 6).

Krauthoff, Tobias (2018). "Dialog-Based Online Argumentation". PhD thesis. Heinrich Heine University Düsseldorf. URN: urn:nbn:de:hbz:061-20180628-094005-1. URL: https://docserv.hhu.de/servlets/DocumentServlet?id=46330 (cit. on p. 12).

Krauthoff, Tobias, Michael Baurmann, Gregor Betz, and Martin Mauve (2016). "Dialog-Based Online Argumentation". In: *Computational Models of Argument.* Comma 2016 (Potsdam, Germany). Vol. 287. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 33–40. DOI: 10.3233/978-1-61499-686-6-33 (cit. on p. 12).

Krauthoff, Tobias, Christian Meter, Gregor Betz, Michael Baurmann, and Martin Mauve (Sept. 2018). "D-BAS – A Dialog-Based Online Argumentation System". In: *Computational Models of Argument.* Comma 2018 (Warsaw, Poland). Vol. 305. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 325–336. ISBN: 978-1-61499-905-8. DOI: 10.3233/978-1-61499-906-5-325 (cit. on p. 12).

Lachapelle, Paul (Apr. 2008). "A Sense of Ownership in Community Development: Understanding the Potential for Participation in Community Planning Efforts". In: *Community Development* 39.2, pp. 52–59. ISSN: 1557-5330, 1944-7485. DOI: 10.1080/15575330809489730. (Visited on 06/23/2025) (cit. on p. 2).

Litt, Geoffrey, Paul Sonnentag, Max Schöning, Adam Wiggins, Peter van Hardenberg, and Orion Henry (Feb. 2024). *Patchwork Lab Notebook: Version Control for Everything. Ink & Switch.* Lab Notes. URL: https://www.inkandswitch.com/patchwork/ (visited on 06/26/2024) (cit. on p. 11).

McKelvey, Karissa Rae, Scott Jenson, Eileen Wagner, Blaine Cook, and Martin Kleppmann (Mar. 2023). *Upwelling: Combining Real-Time Collaboration with Version Control for Writers. Ink & Switch.* URL: https://www.inkandswitch.com/upwelling/ (cit. on p. 11).

Megill, Colin (May 2016). *Pol.Is in Taiwan.* URL: https://blog.pol.is/pol-is-in-taiwan-da7570d372b5 (cit. on p. 8).

Meter, Christian (2020). "Rethinking Online Discussions". PhD thesis. Düsseldorf, Germany: Heinrich Heine University. URN: urn:nbn:de:hbz:061-20200625-134846-5. URL: https://docserv.uni-duesseldorf.de/servlets/DocumentServlet?id=53471 (cit. on p. 13).

Meter, Christian, Björn Ebbinghaus, and Martin Mauve (Sept. 2018). "Jebediah – Arguing with a Social Bot". In: *Computational Models of Argument.* Comma 2018 (Warsaw, Poland). Vol. 305. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 467–468. DOI: 10.3233/978-1-61499-906-5-467 (cit. on p. 70).

NTIA (July 1999). *Falling Through the Net: Defining the Digital Divide.* URL: https://www.ntia.gov/report/1999/falling-through-net-defining-digital-divide (cit. on p. 2).

Paulin, Alois (2020). "Ten Years of Liquid Democracy Research: An Overview". In: *Proceedings of the Central and Eastern European e|Dem and e|Gov Days 2020*, pp. 455–464. DOI: 10.24989/ocg.338.36 (cit. on pp. 6, 61).

Small, Christopher, Michael Bjorkegren, Timo Erkkilä, Lynette Shaw, and Colin Megill (July 2021). "Polis: Scaling Deliberation by Mapping High Dimensional Opinion Spaces". In: *RECERCA. Revista de Pensament i Anàlisi.* ISSN: 2254-4135,1130-6149. DOI: 10.6035/recerca.5516 (cit. on p. 8).

Surowiecki, James (2004). *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies, and Nations.* eng. 1. ed. New York, NY: Doubleday. ISBN: 978-0-385-50386-0 (cit. on p. 5).

Tideman, T. N. (Sept. 1987). "Independence of Clones as a Criterion for Voting Rules". In: *Social Choice and Welfare* 4.3, pp. 185–206. ISSN: 0176-1714, 1432-217X. DOI: 10.1007/BF00433944 (cit. on p. 61).

United Nations Department of Economic and Social Affairs (2024). *United Nations E-Government Survey 2024: Accelerating Digital Transformation for Sustainable Development - with the Addendum on Artificial Intelligence.* eng. New York: United Nations. ISBN: 978-92-1-106728-6. URL: https://publicadministration.un.org/egovkb/en-us/Reports/UN-E-Government-Survey-2024 (cit. on p. 5).

Weidner, Matthew (2023). "Proposal: Versioned Collaborative Documents". In: *Programming Local-first Software Workshop.* PLF '23. URL: https://mattweidner.com/assets/pdf/versioned_collaborative_documents.pdf (cit. on p. 11).

# Personal Publications

Björn Ebbinghaus and Martin Mauve (June 2022). “Better Decision-Making Through Collaborative Development of Proposals”. In: *HCI International 2022 Proceedings*. Vol. 26: *HCI in Business, Government and Organizations*. 9th International Conference, HCIBGO 2022. Held as Part of the 24th HCI International Conference, HCII 2022 (Virtual Event, June 26–July 1, 2022). Ed. by Fiona Fui-Hoon Nah and Keng Siau. Lecture Notes in Computer Science 13327. Springer, pp. 11–23. ISBN: 978-3-031-05544-7. DOI: 10.1007/978-3-031-05544-7_2

Björn Ebbinghaus and Martin Mauve (2025). *Version Control for Structuring Collaborative Decision-Making*. Technical Report: Heinrich Heine University Düsseldorf, p. 9. URN: urn:nbn:de:hbz:061-20250715-102007-0. URL: https://docserv.uni-duesseldorf.de/servlets/DocumentServlet?id=70154

Björn Ebbinghaus and Martin Mauve (July 2026). “Measuring Progress to Support Cooperative Decision-Making Processes”. In: *HCI International 2026 Proceedings*. Vol. 15: *Social Computing and Social Media – Part I*. 18th International Conference, SCSM 2026. Held as Part of the 28th HCI International Conference, HCII 2026 (Montreal, Canada, July 26–31, 2026). Ed. by Adela Coman and Simona Vasilache. Lecture Notes in Computer Science 16715. Springer. Forthcoming

## Peer-Reviewed Demos

Christian Meter, Björn Ebbinghaus, and Martin Mauve (Sept. 2018). “Jebediah – Arguing with a Social Bot”. In: *Computational Models of Argument*. Comma 2018 (Warsaw, Poland). Vol. 305. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 467–468. DOI: 10.3233/978-1-61499-906-5-467

Björn Ebbinghaus and Martin Mauve (Sept. 2020). “Decide: Supporting Participatory Budgeting with Online Argumentation”. In: *Computational Models of Argument. Proceedings of COMMA 2020*. Comma 2020 (Virtual Event). Ed. by Henry Prakken, Stefano Bistarelli, Santini Francesco, and Carlo Taticchi. Vol. 326. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 463–464. ISBN: 978-1-64368-107-8. DOI: 10.3233/FAIA200535

Eidesstattliche Erklärung
laut §5 der Promotionsordnung vom 15.06.2018

Ich versichere an Eides statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf" erstellt worden ist.

| | |
|---|---|
| Ort, Datum | Björn Ebbinghaus |