

Eirik Fatland

Democratic Interfaces

Decision-Making Tools for
Online Communities

MA thesis 2007 • Media Lab
University of Art & Design Helsinki



Cover illustration: «Torgny lagmann på Uppsalating», by Christian Krohg (1852-1925). Originally published as an illustration for the 1899 edition of Snorre Sturlasson's "Heimskringla". The picture is in the public domain.

Copyright © 2007 Eirik Fatland



This work is licenced under the Creative Commons Attribution-Non-Commercial-Share Alike 1.0 Finland License. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-sa/1.0/fi/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

This document was written as a Master of Arts thesis at Media Lab, the school of digital design at the University of Art & Design Helsinki (TaiK).

Contents

Acknowledgements	4
Abstract	5
Introduction	7
Project : What this is about.....	8
Citizensconstitution.org (CCEU)	9
WebTing.....	9
By-products	9
Status of the final work	10
1 The promise of e-democracy	11
1.1 Democracy and e-democracy	12
1.1.1 The paradox	12
1.1.2 Online communities	12
1.1.3 Democracy	13
1.1.4 E-Democracy.....	14
1.2 The case for democratic software.....	16
1.2.1 The case of the Reddit protests.....	16
1.2.2 Plebiscites and petitions	17
1.2.3 The plebiscites of LambdaMOO	18
1.2.4 The Second Life proposal system.....	18
1.2.5 The Downing Street e-petitions.....	19
1.2.6 Free Software Dictatorships.....	20
1.2.7 Debian and the Debian Constitution.....	21
1.2.8 Voting with your feet	21
1.2.9 E-democracy tools for Open Source software?	22
1.3 Deliberation	24
1.3.1 Discussion and deliberation	24
1.3.2 Deliberative democracy.....	24
1.3.3 Online deliberation.....	25
1.4 Towards a tool for online democracy.....	28
1.4.1 Scenarios.....	28
1.4.2 Design requirements for e-deliberation	28
2 software, community, conversation.....	31
2.1 Software, but how to design it?	32
2.1.1 Design methods.....	32
2.1.2 Contextualizing e-deliberation	33
2.2 HCI, Usability and Perception.....	34
2.2.1 HCI and Usability	34
2.2.2 Evaluator effects and user effects	34
2.2.3 Perception : Affordances and constraints	34
2.2.4 Perception in community software.....	35
2.3 Framing Conversations	37
2.3.1 Synchronous and asynchronous communication.....	37
2.3.2 Time, Sequence and reciprocity.....	37
2.4 Community as a design object	39
2.4.1 Software as design material.....	39
2.4.2 Code as Law / Design as Frame.....	39
2.4.3 Community and genre.....	40
2.4.4 From analysis to design	42

3 Process.....	43	5 WebTing	69
3.1 Problems and solutions.....	44	5.1 Design process and philosophy	70
3.1.1 The nature of design problems	44	5.1.1 Constraining conversation	70
3.1.2 Visual reasoning and the “Problem Map”	45	5.1.2 The nativist approach.....	71
3.2 The pattern language	47	5.2 Voting.....	72
3.3.1 Sample patterns	47	5.2.1 Fraud, coercion and bribes	72
3.3.2 Reflection and potential improvements.....	48	5.2.2 Voting standards	73
3.3 Scenarios: from disconnected patterns to complete design	51	5.2.3 The minimum consensus standard	74
3.3 Scenario.....	52	5.2.4 Direct democracy, elections and representation	74
4 citizensconstitution.org	55	5.3 The WebTing process.....	77
4.1 Political context	56	5.3.1 Issues	77
4.2 The WikiTing.....	58	5.3.2 The three phases	78
4.2.1 The WikiTing Model.....	58	5.3.3 Thresholds	78
4.2.2 Advantages.....	59	5.3.4 Motions.....	79
4.2.3 Risks.....	59	5.3.5 Configuration and constitution.....	80
4.2.4 Unresolved design problems.....	59	5.4 The WebTing User Interface	81
4.3 The CCEU process.....	61	Conclusions.....	89
4.3.1 The document finalization process	61	Sources	93
4.3.2 A flexible group structure.....	62	Appendix: a pattern language.....	97
4.4 Website design and usability	64		
4.4.1 The Paper Prototype.....	64		
4.4.2 Visual design	65		

Acknowledgements

I am indebted to five fellow travellers, who contributed ideas, comments and moral support: Ida Blekeli, Bjørn Saunes, Julie Bagger, Viki Ølgod and Li Xin. I am especially grateful for the contributions of Kari-Hans Kommonen, my MA thesis advisor, and of Andrea Botero Cabrera who, as teacher at the MDR Study Project and MDR Tutoring classes, has been something of a godmother for this project.

Valuable contributions came also from the Urban Mediator team (Joanna Saad-Sulonen, Andrea Botero Cabrera, Roman Suzi); from my collaborators at the paper prototype workshop (Priit Tammerts, Diana DeSouza, Eva Durall); and the teachers (Heidi Tikka and Rasmus Vuori), opponent (Richard von Kaufmann) and fellow students at the Final Thesis Seminar.

My former teachers at Designskolen Kolding's department of Interactive Media, Elle-Mie Ejdrup Hansen and Barnabas Wetton, deserve credit for insisting to young, impressionable minds that design, ethics and society can and should be thought of together.

Abstract

'Democratic interfaces' presents a design exploration into mobilizing the potential of the Internet for enabling new and more inclusive forms of democracy. Drawing on online deliberation research, the thesis argues that successful online democracy will need to facilitate open and informed discussion (deliberation) as a prerequisite for democratic decision-making.

The potential for deliberative democracy on the Internet is explored through proposed user interface designs for online deliberation software: WebTing, a tool to facilitate democratic assemblies for online communities; and the citizensconstitution.org website, a campaign for a more inclusive constitutional process in the European Union. Further proposals are annotated as a pattern language and documented as they appear in the design process.

The outcomes of this thesis work are relevant for the design and study of community-enabling software, and in particular online deliberation and discussion software.

Methods used are characteristic of interaction design, including information-gathering, sketching, prototyping and usability evaluation. Particular attention is paid to the challenges of designing community-enabling software, and to the normative influence of user interface design on user behaviour. These considerations suggest a need for new design methods independent of the HCI tradition, focused on user-to-user rather than user-to-system interaction, and on a prescriptive rather than reactive design practice.

(This page unintentionally left blank.)

Introduction

Project : What this is about

My objective, from the onset of this project until its conclusion, has been to design software that enables groups of people - any group of people, whether a sewing club or a national parliament - to reach decisions, in a democratic manner, over the Web. “Democratic” is understood in the sense of deliberative democracy - the philosophy that the transformation of preferences through open and informed discussion is the central component of democratic legitimacy, more important than the aggregation of preference through votes or opinion polls. (Elster 1998, p.2) These objectives place my design work in the context of “online deliberation”, the practice of deliberative democracy on the Internet.

Such software would meet a preexisting need, described in chapter 1, for online communities to reach democratic decisions without having to meet face-to-face. It might additionally, and ideally, enable new kinds of organizations and citizen engagement in the same way that wiki software has enabled new kinds of authorial collaboration, or weblogs have enabled new forms of self-mediation, political discourse and journalism. This latter goal is central to my design process: rather than producing a digital version of the procedures of offline democratic assemblies, I have been trying to envision the nature and toolset of a future online democracy. This ideal toolset, the goal of the design process, I refer to as a “Web Parliament” - or, to use the Norwegian term and avoid the connotations of real-life parliaments, a “Web *Ting*”.

While these objectives have been clear almost from the onset, the path to reach them has not. As drafts were made, literature surveyed, and ideas put to the test it became increasingly clear that it was not a matter of simply “designing” a piece of software, in the sense that one might

design a corporate web site, a car, or a sports shoe. In such cases, the designer may usually proceed from the general and well-established properties of the thing - e.g. cars have wheels, engines and doors - to the specific nuances of an individual design: *this* car has thick low wheels, red angular doors and a small engine in the back. Rather, it was necessary to *invent* a new kind of software; to formulate a hypothesis of what the general properties of such software might be, and then test that hypothesis by implementation in specific designs. To return to the car analogy: I have not been designing a “car”, but a “transportation vehicle”, and the first order of design problems was to determine whether a “transportation vehicle” might work by walking, running, dancing, rolling, flying or swimming.

How does one design the unknown? A typical process of interaction design might include scenarios, personae, task analysis, usability testing and prototyping. In this case, such methods seemed woefully specific: assuming a number of “knowns” that in this case were “known unknowns”. The problem is especially acute when we consider the nature of deliberation software as both supporting and enabling communities. The literature on Human-Computer Interaction (HCI) and Interaction Design (IXD) is rich with methods and examples of interaction between a user and a system. Not so when it comes to interactions between users and other users, for whom the “system” is merely an intermediary. As will be discussed in chapter 2, the main design problems of community software are found in predicting the consequences design will have on user-to-user interaction, rather than in framing the experience of individual users.

Grappling with these challenges led me to adopt a meandering, multi-faceted process, exploring the same problem through different projects and methods. As a result, this thesis documents a number of sketches

and concepts, and two complete user interface designs: WebTing and Citizensconstitution.org (CCEU). Process and methods are discussed specifically as they relate to each project, and generally in chapters 2 and 3.

These products have been developed in parallel, each one influencing and being influenced by the others. The order of their presentation in this thesis is therefore topical rather than chronological.

Citizensconstitution.org (CCEU)

Intended as a constructive form of political protest, the website citizensconstitution.org - documented in chapter 4 - invites European citizens to participate in an open, online process to draft an alternative constitution for the European Union, authored through an inclusive and deliberative process rather than back-room negotiation. CCEU was the “spark”, the initial idea that set of this investigation into e-democracy, but also - following many revisions - the last design to be finalized. At the time of writing, the first design and implementation phases have concluded while the campaign itself is in its start-up phase.

WebTing

The WebTing software design, presented in chapter 5, is the product that most closely matches my initial goal. In its current iteration, it is based on a three-phased procedure - with each topic/issue moving through an exploration and a discussion phase before reaching the decision phase where votes are cast. Each phase affords a certain kind of speech: collaborative authoring (exploration phase), individually signed comments (discussion phase) and vote aggregation (decision phase). An issue will move between the phases dependent on it reaching a pre-defined threshold, such as a certain amount of time having passed or an

amount of votes being cast. The application is designed to be scalable and highly configurable, while simpler to understand and use than most offline democratic procedures.

By-products

In addition to these two design projects, my design process has produced several by-products in the form of novel theory and smaller (or less completely explored) design concepts. The most generally

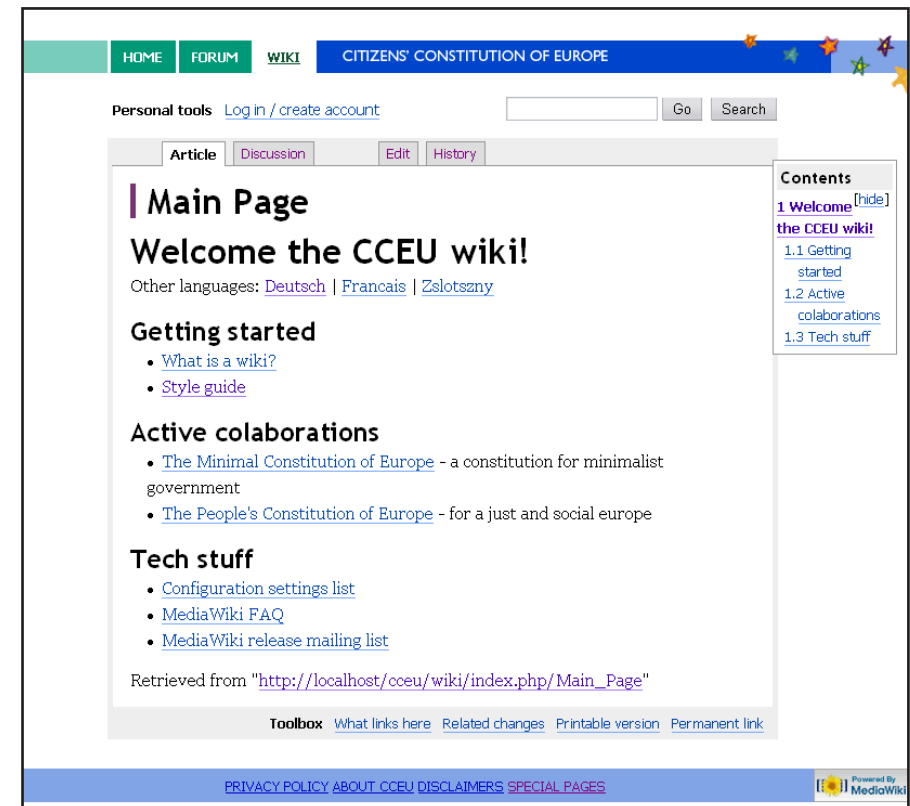


Fig. a: Screenshot of a development version of citizensconstitution.org, an activist project inviting citizens to author a constitution of Europe through wiki deliberation.

applicable of these by-products is the pattern language presented in the appendix, an attempt to map out the design patterns - both well-established and hypothetical - of social software as they might be used for the purpose of online deliberation. While it is presented here as a work in progress, the pattern language might have further use in mapping out other potential deliberative applications, and might be extended to become a pattern language of social software design in general.

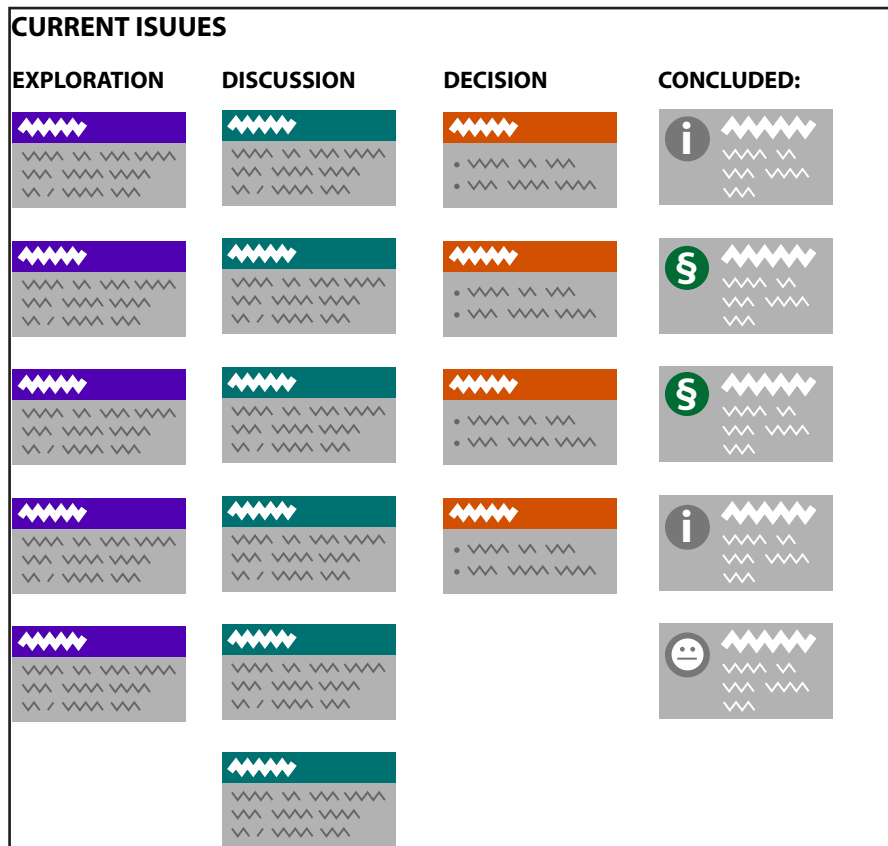


Fig. c: Sketch of the WebTing user interface. A tool for democratic deliberation and decision-making.

Status of the final work

Community software is not a “product” of the kind that can be designed, tested, produced, packaged and then forgotten about. Both the software and the communities that it enables are subject to cycles of improvement, feature requests, and evolving social practices. Even if the software has reached a stable and satisfactory form, the Web is subject to continuous evolution and software design must respond to both new challenges and to new opportunities. To validate these designs, they will need not only to be implemented as working software, but also to undergo the tests of being put to use by real communities. No design can expect to escape such a process unscathed.

The projects described here are in different stages of design and evolution. At the time of writing, it is difficult to predict whether the www.citizensconstitution.org project will succeed in recruiting a viable community, let alone whether the campaign will get anywhere close to its’ utopian goal of re-designing the European Union. WebTing, the most radical of these designs, exists only as blueprints and mock-ups. Turning those blueprints into viable software will be a considerable programming challenge, one that falls outside of the scope and time frame of an MA Thesis in New Media Design.

The promise of e-democracy

Summary

Despite sustained governmental and research interest into “e-democracy”, and despite the plethora of virtual communities and organisations that are currently found on the Internet, there are surprisingly few examples of successful online democracies. The lack of democracy in online communities also raises questions about the ability of ICT to support participatory politics in general.

In this chapter, I argue that the problem is one of tools: that current community-enabling software is poorly suited for democratic decision-making, and that a tool designed specifically for this purpose might enable strong democracy in online communities. In particular, such a tool should support democratic deliberation: procedure that enables open, informed and rational discussions amongst stakeholders as a prerequisite for decisions.

Cases from contemporary online communities indicate a need for such deliberation tool. Based on the explicit and implicit needs of these cases, a set of design requirements is presented.

1.1 Democracy and e-democracy

1.1.1 The paradox

For better or worse, “democracy” - that abstract, ambiguous and not always well understood ideal - has emerged onto the centre stage of history. In the span of a mere 18 years, the number of democratic states doubled from 66 in 1987 to more than 120 in 2005 (Freedom House 2005). Francis Fukuyama called it “the end of history”: the culmination of the 20th century’s ideological struggles (between communism, fascism, nationalism, imperialism etc.) with the enthronement of the Western Liberal Democracy, the sole survivor, as the One True Way of every nation. (Fukuyama 1989)

Manuell Castells’ notion of the “network society” describes global developments concurrent with this spread of liberal democracy. In his trilogy on the Information Age, Castells documents the many convergent changes in private enterprise, scientific production, media, economy, telecommunications, organization and entertainment that have occurred since the 1980s. The broad outline of these changes is summarized in the appearance of what Castells calls “the space of flows”, the synchronous and geographically displaced space of near-instantaneous communication, to challenge the “space of places” that the body traditionally inhabits. (Castells 2000, pp.440-459)

Our paradox, then, is that these two developments do not seem to connect. The formal relationships between democratic governments and their citizens have remained fundamentally the same. The space of flows, whatever other qualities it might have, does not seem to afford democratic decision-making. This is particularly evident when we search for democracy on the Internet, that most visible manifestation of

the “space of flows”. At first glance, even the organizations native to the Internet - such as the communities that inhabit virtual worlds, or open source software projects¹ - do not elect representatives, do not hold referenda, and are the objects of lively debates that have only incidental influence on their actual governance. Such organizations might appear as loose networks or strict autocracies. They might resemble monarchy, meritocracy, oligarchy or anarchy, cathedrals or bazaars - but they do not resemble democracy.

1.1.2 Online communities

We should take particular note of those “native organisations”, the associations formed by Internet users who first met online, and who might never otherwise have talked, worked or played together. For better or worse, they are the vanguards of Internet culture and Internet tools. Cultural conventions, such as the emoticon or the verbal style of e-mail or the name “spam” for e-mailed advertisements, turned up first in such communities and then spread outwards - to the point where they are used in the SMSes of people who have never touched a computer. The tools these communities invent and use, though they often begin their lives as quick hacks, tend to be more versatile and resilient than tools designed by armies of software engineers and interaction designers to meet the specified needs of off-line organisations. Consider, for example, blogs, web fora, wikis, comment fields, the e-mail, the issue tracker and version control software. These software tools can all track their lineage back to such online communities, where they were either invented or made popular, even though they are now found on corporate intranets, on the websites of major news organisations, busily at work producing supercomputer operating systems or free multilingual encyclopaedia.

¹ Exceptions to this rule will be discussed in section 1.2

A lot has been written about one kind of “virtual community”, the kind where participants meet to play or socialize freely. (e.g. Kollock and Smith 1999). But virtual communities also include communities that are primarily utilitarian or goal-oriented: software development communities, mailing lists for members of particular professions, groups of writers sharing and commenting on drafts. Such communities, often referred to as “Communities of Practice” (Wenger 2001) use online tools not just to socialize, but also to share knowledge, work and collaborate.

It is not surprising that these communities should be the “wine-tasters” of online tools and practices, since an online community is not merely *supported* but entirely *enabled* by such tools: without functioning tools and social practices, the community cannot exist. This raises the bar for both usability and utility, making obvious problems that might otherwise be hidden. The apparent absence of democracy in online communities, therefore, is significant.

1.1.3 Democracy

Have we been searching for the right thing? Would online democracy look like offline democracy, with elections, representatives and assemblies? After all, the established forms of democratic governance - majority votes, separation of powers, elected representatives in parliaments or city councils - occupy a fairly narrow niche of what historically has been called “democracy”. According to Aristotle², the first classifier of governments, our present systems of governance would not even be considered democratic - an elected elite was too much like “oligarchy” for his taste, proper democracies selected their representatives by lottery. The Scandinavians of the Viking era gathered

for their “Ting” (parliament/council) not just to determine laws or coordinate action, but also to pass judgement: the legislative and judiciary bodies were one and the same. And when representatives were needed - as they were for larger Tings in distant locations - the law-sayer, the secretary whose job it was to memorize laws, would do the job of representing the local Ting. U.S. citizens were actually never meant to elect their president directly – rather, they would vote on an assembly of “electors” who in turn would deliberate on the best choice of president.³ (Fishkin 2000) These were all democracies, in the sense of allowing a population (a “demos”, in the ancient Greek) to rule themselves. But their manner of self-rule was starkly different from the present forms of democracy.

Nor can we limit the notion of “democracy” to the governance of states. As a decision-making process, democratic governance can be found in membership organisations of all kinds, and even in otherwise non-democratic institutions such as corporations (shareholders electing board members, employees electing union representatives) or universities (faculty electing deans and principals). Outside the formal electoral systems of states, democracy plays an important role for the internal governance of political parties, issue-oriented campaigns and non-partisan interest groups. It might not be reasonable to expect a *state* to be governed through the Web, but it is not too much to ask why (or if) there are so few democratic *organisations* on-line.

So what should we be looking for? What are the tell-tale signs of democratic governance, independent of their scale and implementation? There are many, many definitions and theories of democracy. Without delving too far into political science or philosophy, I will present my

³ With the rise of political parties, electors began voting for their party’s presidential candidate, giving rise to the present system.

² In *Politics*, Book III, Chapter 9

own definition here, focusing on two recurring criteria: **consent** and **legitimacy**, concepts that are abstract enough to take many different forms depending on the material and social circumstances.

Definitions based on the consent criterion (e.g. Buchanan and Tullock 1962) state that for something to be called “a democracy” it requires that those who are governed - the *demos* - consent to the decisions taken on their behalf. Consent is something less than “approval”, a citizen need not approve of the decisions being taken but needs to agree that the decision-maker has a right to take them. As a consequence, the *demos* must also have an ability to overturn decisions or decision-makers they do not consent to - Karl Popper famously defined “democracy” simply as a way to remove those in power without bloodshed. (Popper 1963)

How do we separate a decision we disagree with, but consent to, from one that merits overturning? It will need to be seen as legitimate. Legitimacy is not only a component of democracy, but of nearly all types of government: the Pope derives his legitimacy not from the *demos* of believing Catholics, but from Christ’s words to Peter and the notion of “apostolic succession”. For a democracy, “legitimacy” may mean that a decision is taken according to predetermined rules or procedures - and that these rules and procedures themselves have the consent of the *demos*. Such rules are usually found in constitutions and by-laws, but they may also have evolved through custom.

A functional democracy will need to ensure, through its procedures, that consent and legitimacy can actually be expressed and tested. One of the more obvious procedures needed to ensure measurable consent is freedom of expression - if the members of the *demos* are not free to

speak their mind without fear of reprisal, it is impossible to determine whether they consent.

To summarize: when talking about “democracy”, I am talking about a decision-making procedure that (a) has the consent of its’ stakeholders, (b) produces decisions that are seen as legitimate by its’ stakeholders, and (c) provides the stakeholders with a realistic mechanism for challenging and changing the procedure. This definition is broad enough that we can assess procedures that do not involve electoral politics in the traditional sense as democracies.

1.1.4 E-Democracy

Already in the pre-Web days of computer-mediated communication, claims were made about how such communication might transform democracy by enabling the direct participation of citizens in government. The 1992 US presidential campaign of Pat Buchanan, for example, had as its’ cornerstone a promise to introduce a system of government based on “Electronic Town Hall Meetings”. (Clift 2000) The early years of the Internet certainly gave reason to see it as a democratizing medium.

The Internet Engineering Task Force (IETF) which, since 1986 has defined the standards that make the Internet possible, does so through an open and inclusive process, accepting input from all stakeholders and determining standards through a consensus-like process. (IETF 2006) However, while the IETF is still operational, the layers of governance added to the Internet in later years - such as the ICANN, which manages the domain name system, and the W3C, which sets

standards for World Wide Web technologies, have been increasingly opaque and decreasingly consensus-oriented.⁴

The words “teledemocracy” and “E-democracy” have been used interchangeably to describe the notion that digital communication might somehow transform democracy, or that democracy might somehow be implemented through digital communication technologies. (Clift 2000) There is a large body of research that addresses, or tries to realize, these claims. Some of this research is produced by academic communities and guided purely by research interest (e.g. Dahlberg 2001, Witschge 2002). A lot of it is either conducted or funded by political actors : governments, the public sector, transnational organisations (e.g. EU Comission 2003), political parties and think-tanks (e.g. Rushkoff 2003). Such research most often addresses the potential of ICT to improve some aspect of the functioning of democratic *states*: accessibility, public consultations, communication between citizens and representatives. The field of “e-government” is a close relative of such research.

Little of this research, though, deals with the question of *democracy itself*. Most often, the “e-” in “e-democracy” is seen as something ancillary to “democracy”, and reduced to a supporting function. That “democracy” should retain it’s current (19th century) form, one of face-to-face debates and elected representatives, is taken for granted. And a further given is that “democracy” is the “democracy” of states, governments, the public sector - not the “democracy” of the Red Cross or the parents council. It is not surprising that politicians or civil servants should favour this approach - their main interest, we can assume, is to discover how ICT might help them do better what

they are already doing. But e-democracy research answers few of the questions I am asking.

⁴ The fragmentation of W3C consensus can be seen, for example, in the creation of the parallel WHAT working group: <http://www.whatwg.org/>

I.2 The case for democratic software

Democratic decision-making incurs a costly overhead: research, discussion and voting take up time and energy. A significant body of literature documents how citizens are often averse to political discussion and conflict. (Witschge 2002, Regezci 2004) We need to ask the question: does the internet need democracy? Are there examples of communities that would benefit from e-democracy? Of places where the seeds of democratic rule are already sown?

Through the following cases, I hope to show that the answer to these questions is a qualified “yes”.

I.2.1 The case of the Reddit protests

The website reddit.com belongs to a new class of services known as “social aggregators”. In a typical social aggregator users submit links to web sites they find interesting, and then vote on each others submitted links. Links that accumulate a certain amount of points in a certain time-span are presented higher up in the list of links, where they might attract more votes. The front page of the site contains the most popular links at any given time, and is the page visited by the majority of users. In addition to submitting links and voting such submissions up or down, reddit allows users to post comments discussing the links, giving rise to an active user community. The user community is, for the most part, self-moderating: comments, as well as submissions, can be given a positive vote (“upvote”) or negative vote (“downvote”). Comments with a negative score (more downvotes than upvotes) are hidden, while those with a high positive score are given prominence. The users call

themselves “redditors”, a pun on the word “editor” and a statement of communal identity.

During 2007, reddit.com became increasingly politicized. Links to articles criticizing the Bush administration and the war in Iraq rose quickly to the front page, and were the subject of long discussions. On the 16th of July, a redditor by name of 325i tried to channel this political passion into action, by submitting a link with the following title:

“Reddit: Let’s organize a massive, nationwide protest against the policies of the current administration. We want to end the war, we want the whole truth about 9/11, and we want the executive branch to be held accountable to the law. It’s time to take back the country our ancestors gave their lives for.”⁵

Instead of any external site, the link pointed to it’s own comment page, a common way of presenting a question to the community or opening for a discussion about reddit itself. The first comment was authored by 325i, elaborating on his/her proposal:

“We bitch and complain every day on Reddit, Digg, and other sites about the direction our country is headed. Though we disagree on many things, I think we can agree on at least one thing - Without our freedom we have nothing. It’s being taken away bit by bit, day by day. The nature of the 9/11 attacks is in question, and we need to put politics aside and find some real answers once and for all. We’re engaged in a worldwide struggle for global domination, and global domination is not in the interest of everyday Americans, or everyday citizens of the

⁵ <http://reddit.com/info/26sg7/comments>

world. The will of the people is clear. We want to live our lives in happiness and prosperity. It's time to make it clear that we outnumber the super-elite members of society, and our will can and will be done.”

The submission received a large amount of “upvotes” (1125 as a final count), bringing it to the very top of the reddit front page and indicating that at least 1125 readers (not counting “lurkers” without user accounts) thought the protests were a good idea and would presumably participate in them.

But there agreement ended. Another redditor, by the name of “OMouse”, replied to 325i’s original comment, quoting from his text:

“>The nature of the 9/11 attacks is in question

No, no it isn't.”

OMouse’s comment received 82 points, more than 325i’s original comment (79). The discussion about the proposed protests then derailed into a long exchange about whether or not the terrorist attacks of Sep. 11th 2001 might have been the result of a US government conspiracy of some kind, and whether slogans about the 9/11 issue might detract from the credibility of the protest as a whole. The reddit community, while fairly united around the idea of protesting “the policies of the current administration”, split over the 9/11 question.

A good deal further down on the comment page, other redditors had begun discussing locations and dates for the protests. But those discussions were occluded by the discussions over paroles. Later on, further submissions about the proposed protests appeared on the front

page, including an idea that all protesters wear formal clothing.⁶ The submissions and discussions continued for a while, but (to make a long story short) the final outcome of this process consisted of four (4) redditors appearing at an anti-war demonstration dressed in suits and neckties.

What went wrong? How did 1125 interested individuals become four protesters? The challenge facing the reddit community was one of electronic decision-making. The incentives to reach a decision were clear: reach agreement or protests will not happen. In a regular, off-line process, such a conflict would normally be resolved by participants focusing on the paroles they had in common rather than those that split the community. But reddit’s structure prevented such a resolution: controversial statements were awarded prominence, while statements that might attract consensus were not. Comments were organized by time and votes rather than by topic, making it difficult to get an overview. Finally: reddit had no obvious feature for concluding discussions. No visible way, except for the submission voting mechanism, to reach agreement on an answer to 325i’s question other than simply “yes” (upvote) or “no” (downvote), and no way to force users to reach a conclusion within a given time frame. As the discussion proceeded, more and more users presumably stopped following the discussion. The failure of the reddit protests was largely a failure of tools.

1.2.2 Plebiscites and petitions

Perhaps the simplest form of “direct democracy” is the mechanism of “plebiscitary” democracy. In a plebiscitary democracy, any and all decisions can be brought up for a general referendum - a “plebiscite”. A

⁶ <http://reddit.com/info/26tys/comments>

“proposition”, a law or decision, is presented to voters who may either accept or reject it. Most states have used plebiscites on occasion, but in a typical plebiscitary democracy, such as Switzerland or California, voters themselves have the power to bring a proposition to the ballot, e.g. by gathering the signatures of a certain percentage of fellow citizens.

Plebiscites are formal institutions: elected representatives are, by law or custom, expected to yield to the decision of the electorate. A related but less formalized institution, that of the “petition”, has no such power. A petition can be started by anyone - politicians, NGOs, concerned citizens - who proceed to gather as many signatures as possible in the hope that legislators or government will take note of the petition as an expression of “the will of the people”.

1.2.3 The plebiscites of LambdaMOO

While plebiscites and petitions are clearly separate entities in ordinary political life, their functions have been blurred on the Internet. The virtual world LambdaMOO, after a failed experiment in entirely hands-off governance, (see Smith 1999, Reid 1999) instituted a form of plebiscitary democracy: users could make proposals about changes to the technical features of LambdaMOO. If the proposal was approved by a system administrator, it would pass to a vote by all users - thereby transforming it from a status akin to a petition into one akin to a plebiscite. While LambdaMOO’s democracy is often cited as a model of self-governing virtual worlds, its inventor - Pavel Curtis - regards the petition system as a failure:

It has, by and large, failed to be the jumping off point I hoped for; we have not seen it used successfully to move LambdaMOO to a working, stable form of self government. There were long

periods, indeed, where many petitions reached ballot stage and none of them passed; it seems to me now that the voting population could never agree on anything of real substance. I think that this is the real lesson of LambdaMOO’s experiment with direct democracy. (Curtis 2002)

1.2.4 The Second Life proposal system

Several years later Second Life, another virtual world, would introduce a similar system, albeit on Second Life’s website rather than inside the virtual world itself. In the Second Life system, users could propose new features or changes to existing ones. Any user could vote on each proposal, either positively or negatively, and the list of proposals was displayed ordered by the most popular requests first.⁷

While Second Life features did not need to be “vetted” by the system administrators, the implicit agreement was that if Second Life’s owners saw no good reasons not to implement a popular feature, it would be implemented. Once again, we are dealing with a system that blurs the line between “petitions” and “plebiscites”.

The Second Life system, however, failed in some important respects. Since the most popular features were displayed first, users would focus their attention on these issues rather than on those further down the list. In practice, this made it almost impossible for new proposals to attract votes - the front page was dominated by issues that had the twin advantages of having been proposed early and of being popular. While a discussion was associated with each issue, these discussions were not linked together: two contradictory proposals might become the stages of two entirely separate discussions about the exact same question.

⁷ Observations made in 2004-2005. The petition system was accessible only to Second Life customers, and no publicly archived version exists.

Arguments and information relevant to both proposals would usually be posted only in the discussion of one of them. The high volume of proposals furthermore meant that the discussions attracted few readers and debaters. Most users, apparently, looked through the list and clicked on proposals they immediately felt they agreed with, without following the link to the discussion. This led to technically infeasible, contradictory or outright foolish proposals reaching high levels of popularity.

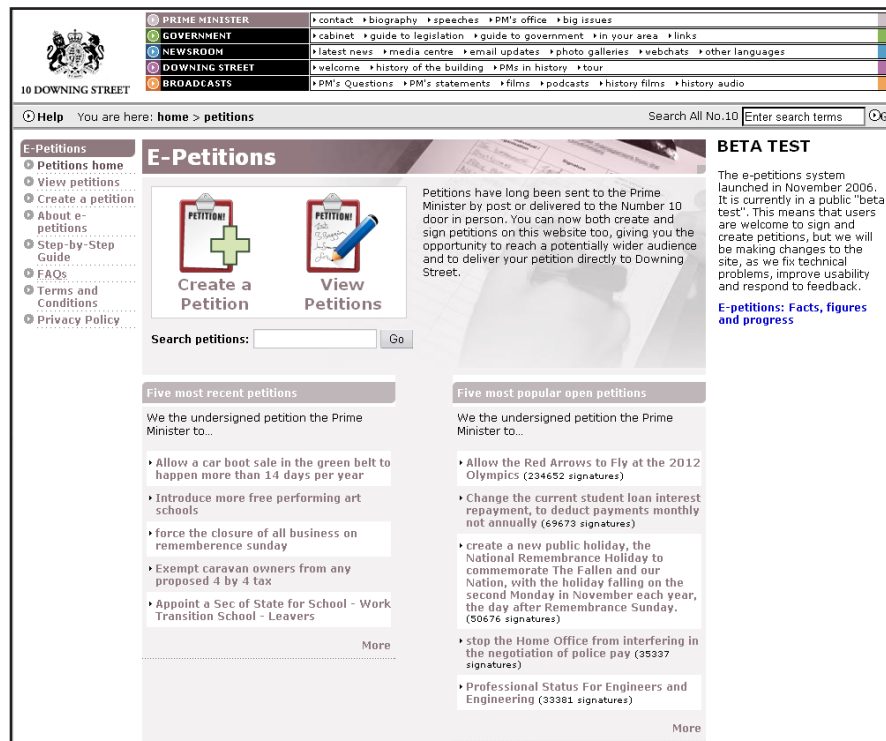


Fig. 1a: Screenshot of 10 Downing Street E-petitions.

1.2.5 The Downing Street e-petitions

Second Life's experiment in petitionary democracy was eventually discontinued. While an exact reason was not given, it is not hard to imagine reasons for shutting it down. Apparently, though, no-one in the office of UK prime minister Tony Blair were players of Second Life, as in November 2006 a very similar system was implemented under the Prime Minister's website.⁸ At the time of writing, the Downing Street petitions lists its five most popular petitions as:

1. Allow the Red Arrows to Fly at the 2012 Olympics
2. Change the current student loan interest repayment, to deduct payments monthly not annually
3. Create a new public holiday, the National Remembrance Holiday to commemorate The Fallen and our Nation, with the holiday falling on the second Monday in November each year, the day after Remembrance Sunday.
4. Stop the Home Office from interfering in the negotiation of police pay
5. Professional Status For Engineers and Engineering

The first petition is directed towards the wrong recipient: the opening ceremony of the 2012 Olympics is the decision of the Olympic Organizing Committee, not the Prime Minister. The second petition is based on a misunderstanding: Student loan interests are calculated on a daily basis, but presented annually. The fourth petition is an attempt to influence the adversary (government) in an ongoing negotiation between labour unions and their employers. It is doubtful whether this petition makes the case any more persuasive than the union negotiators already are. The fifth petition demands that the term "engineer" be made a protected title. It is doubtlessly popular with its signatories, who

⁸ <http://petitions.pm.gov.uk/>

would remain “engineers” under the proposal, but says nothing of what the government can expect from those who would lose the title of “engineer” if this proposal were to become policy. In other words: of the five top petitions, only one - the proposal for a remembrance holiday - is neither a misunderstanding, nor an impossibility, nor directed towards the wrong recipient.

The Downing Street petition system has many of the same flaws as we found in the proposal system for Second Life. Displaying petitions ranked by their popularity biases the viewer towards a small number of already-popular petitions. An interface that affords simple voting/signing without reading up on the issue or relevant counter-arguments leads to citizens voting their “gut feeling” rather than their informed opinion. Poor support for discussions means that relevant information and counter-arguments cannot be expressed in the voting context. But the Downing Street petition system is enormously popular with its constituents: during its first year, the system has recorded 5.8 million signatures from 3.9 million individual e-mail addresses⁹. Is there perhaps a need here that might be served better by a different user interface? And might we apply the same solutions to e-government as to purely online communities? The similar issues faced by Second Life and the United Kingdom suggest that we might.

1.2.6 Free Software Dictatorships

The Linux Kernel is perhaps the most resounding success of the Free/Open Source Software (FOSS) movement¹⁰. When combined with other components, such as the GNU command-line tools and the GNOME or KDE graphical user interfaces, the Linux Kernel forms the

⁹ <http://www.pm.gov.uk/output/Page11051.asp>

¹⁰ For more information on the FOSS movement, see www.opensource.org or www.fsf.org

basis of a family of operating systems (usually called “Linux distributions”) such as Ubuntu, Debian and Fedora.

The origin myth of Linux places it firmly in the realm of the online communities discussed in 1.2: In 1991 Linus Torvalds, a Finnish hacker, sends out a message¹¹ to a USENET¹² group, talking about his hobby programming project - an operating system kernel - and offering to send the source code to anyone who might be interested. Several group members respond, make additions and improvements to Torvalds’ original code, and send the improved versions back to Torvalds. Mr. Torvalds reviews the improvements, includes some of them in the next version of his kernel, and the rest is history.

At the time of writing, sixteen years after that USENET message, the Linux kernel is growing at a speed of 2000 lines of code per day. It involves thousands of individual contributors, and a complex hierarchy of officers¹³, who manage submissions of new code in a given domain. Some of these are salaried professionals - with contributing to the Linux kernel as part of their job description. Most are volunteers, people who might be professional programmers but who are contributing to the kernel on their own time. At the top of the hierarchy, we find Linus Torvalds, with the title of “Benevolent Dictator For Life”, a title that is both tongue-in-cheek, but also fairly accurate: it is unclear what might happen were Linus Torvalds to disappear from the Linux project. It is unclear how a serious dispute between Torvalds and the developer community might be resolved.

¹¹ <http://groups.google.com/group/comp.os.minix/msg/b813d52cbc5a044b>

¹² USENET is a system for the asynchronous exchange of messages, organised by “newsgroups”, that are comparable to e-mail lists. USENET was the frame for many of the early online communities of the Internet. E-mail lists and web discussion have now largely supplanted Usenet, thanks to their relative accessibility and due to the inability of many of the old USENET groups to handle the problem of spam.

¹³ <http://www.itnews.com.au/News/63459,how-linux-is-testing-the-limits-of-open-source-development.aspx>

For the Linux kernel, these are mostly non-issues: Torvalds, as the founder, enjoys legitimacy. His skills in diplomacy and management are undoubtedly factors behind the success of the Linux project. But if there's any lesson to learn from history, it is that benevolent dictators - whether Julius Cæsar or Robert Mugabe - cannot be relied on to remain benevolent throughout their reign, and furthermore that the institution of the benevolent dictator is open to abuse by the successors of the initiator. Open source software development is not a tea party: ideological fractures inside the community include different perspectives on development methodology, differing attitudes to development tools (especially differing preferences regarding programming languages) and the split between ideological adherents of "Free Software" on the one hand and pragmatic, business-minded "Open Source" on the other. Some of Torvalds' decisions, such as using a closed-source program to host the kernel source code and his conflict with Richard Stallman over the GPL v3 license, are far from uncontroversial. A "benevolent dictator" with less stature or legitimacy might have significant difficulties mediating such contentious issues.

1.2.7 Debian and the Debian Constitution

By way of contrast, consider the case of Debian GNU/Linux - one of the most prominent "distributions", or operating systems using the Linux kernel. The Debian project has an elected "Project Leader", who sits for a one-year term, and a "technical committee" whose members are appointed by the committee or by the project leader. The powers of these institutions are clearly defined and limited through the "Debian Constitution". (debian.org 2006) Elections are held at regular intervals, and Debian developers (whose status and membership criteria are also defined in the constitution) additionally enjoy the power to initiate plebiscites, with votes held through the Debian e-mail lists. Were

Linus Torvalds to resign without an obvious and equally benevolent successor, it might send the Kernel community into turmoil. But replacement of Debian's Project Leader is a matter of constitutional routine. Here, at least, we are faced with an online organization that is clearly a democracy. It also happens to be an exception in a world where "benevolent dictators for life" are the rule.

1.2.8 Voting with your feet

There is a simple, and common, reply to my criticism of benevolent dictators: a group of programmers volunteering their time does not a *demos* make. On the pluralistic, unregulated Internet people always enjoy the ultimate power to "vote with their feet": unhappy with Linus Torvalds? There are plenty of other Open Source Software projects to join. And since the code is open source, available to anyone, developers always have the freedom to "fork" the code - to make a copy of the code, and continue working on the copy independently of the community that originally developed it. Fig. 1b¹⁴, a timeline of Linux distributions, illustrates the forking mechanism in practice - showing how a handful of original distributions have become the hundred or so distributions available today. Not all of these are "forks" - Ubuntu Linux, for example, is built "on top of" Debian GNU/Linux and contributes source code to Debian - but many are. Fig. 1b also aptly illustrates the problem of forking: it diffuses the energy of developer communities, and creates multiple alternative solutions to problems that might be better solved by a single one. While some of these distributions are not competitors - but rather experimental or specialized operating systems, the sheer number of them is a considerable hindrance for new adapters and for new developers wondering where best to contribute their time.

¹⁴ Source: <http://futurist.se/gldt/>

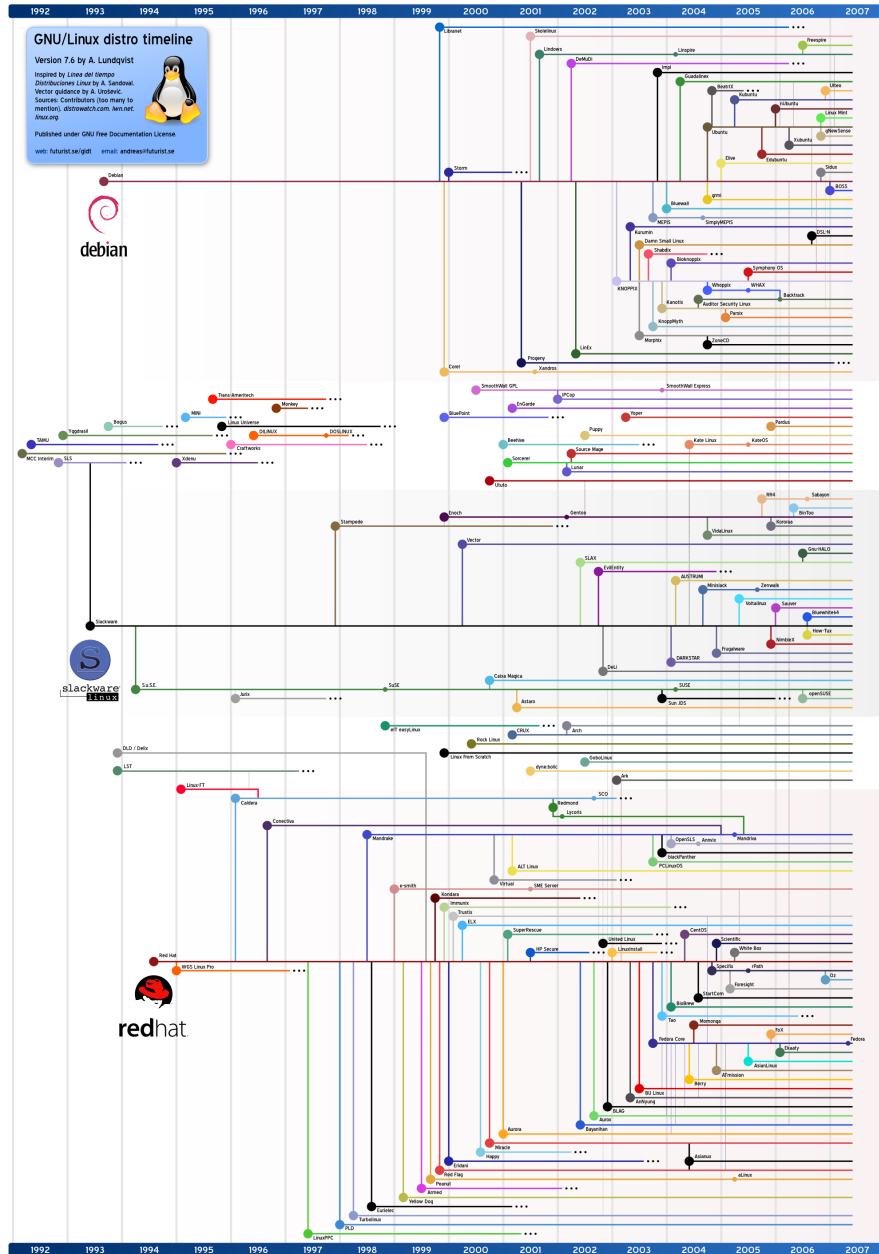


Fig. 1b: Timeline of Linux Distributions

The “vote with your feet” approach is appropriate if we construe the members of online organizations as consumers - my having no influence on the taste of Coca Cola does not infringe on my democratic rights, nor does it cost me much to buy Pepsi or drink tap water instead. But members of online organizations may invest significantly, in terms of time or work done, in their community - thereby behaving more like political subjects. The programmer who has devoted his life to improving the Linux kernel cannot simply fork or switch projects after losing a quarrel with Linus Torvalds, the cost of doing so is high.

1.2.9 E-democracy tools for Open Source software?

Here, then is the case for democracy in open source projects: to reduce dependence on a founder, to ensure continuity if the founder(s) depart, to resolve differences of opinion in a manner that avoids unnecessary forks and departures. On the face of it, procedures such as those enshrined in the Debian Constitution might achieve all this. Why, then, is Debian exceptional? The Debian Constitution speaks in a language that reflects its particular community, uses procedures that reflect its particular needs, and is clearly an entity that has been evolved over time under the influence of uniquely skilled community members. It is at once too particular for general use, and too complex for easy imitation.

Websites such as sourceforge.net have contributed to a standardization of open source procedures. The founder of a new open source project need only open an account at sourceforge.net in order to create project-specific discussion forums, code management tools, a support and bug system, mailing lists etc. ¹⁵ Such software packages easily become normative for new open source projects, who can be expected to

15 <http://sourceforge.net/support/getsupport.php>

care more about *getting work done* than about long-term questions concerning management, governance and legitimacy. If a blueprint for democratic governance was encoded, not as a particular and context-specific text such as the Debian constitution, but as a general-purpose tool alongside the other standard tools at sourceforge.net, it might lead to more open source projects adopting democratic governance at an early stage.

I.3 Deliberation

One of the earliest decisions I took was to base my design on the philosophy of deliberative democracy. I did not know, at the time, that such a thing as “deliberative democracy” existed. I had noticed that there already existed tools for online voting, such as the petition/plebiscite systems mentioned above, without this resulting in a functioning online democracy. On the contrary - the most “democratic” organisations on the web, such as the IETF and Wikipedia, were characterised not by voting but by large volumes of discussion. If a well-designed tool might enable strong online democracies, the design challenge lays somewhere else than voting - presumably in the phase that comes before voting, where the question of “what do we vote on?” is resolved.

I.3.1 Discussion and deliberation

Jon Elster (1998, p.5) distinguishes between three modes of collective decision-making: arguing, bargaining and voting. All three are frequently found in democracies, or indeed any situation where groups of equals take decisions, but it is the mechanism of “arguing” that holds the main interest for deliberative democracy.

“Deliberation” describes the kind of arguing where participants, in addition to persuading others about the validity of their own views, are equally open to being persuaded by their peers. The process of deliberation serves to share knowledge and information between participants, to weigh arguments and counter-arguments, to uncover underlying values and prejudices, and to determine the most relevant points of contention. Consensus might be the ideal outcome of deliberation, the goal participants are working towards. But if consensus

is not achievable the decision should at least be well understood, also by those who disagree with it. There are several advantages to such conversations: they tend to produce better informed decisions, they tend to uncover and compensate for flaws in initial positions, and they tend to reduce conflicts over final decisions since even disagreeing participants will have an understanding of *why* the decision was taken - of the motivation and values of their peers.

There is nothing exotic or unusual about deliberation. Most discussions are, in theory, deliberations. In practice not all participants might be open to changing their opinion and might resort to bargaining, manipulation or coercion in order to achieve their preferred conclusion. Two of the questions we shall assess later on are: which online conversations constitute deliberation, and how might we design to encourage deliberation?

I.3.2 Deliberative democracy

“Deliberative democracy” describes the view that for a decision to be legitimate, it must (a) be reached through deliberation, involving (b) all stake-holders affected by the decision.¹⁶ The contemporary discourse on deliberative democracy takes as its starting point the works of Jürgen Habermas. But the basic tenets of deliberative democracy were clearly articulated already by Edmund Burke in 1774 and exhortations

¹⁶ Almost every theorist of deliberative democracy has their own definition or explanation of “deliberative democracy”. Elster 1998 (p.8) summarizes the points of agreement as follows: ““All agree, I think, that [deliberative democracy] includes collective decision making with the participation of all who will be affected by the decision of their representatives: this is the democratic part. Also, all agree that it includes decision making by means of arguments offered by and to participants who are committed to the values of rationality and impartiality: this is the deliberative part”. My explanation above is based on this summary, with emphasis on the aspects that are most relevant to this thesis.

of the advantages of open and informed discussion, as opposed to crowd-pleasing sophistry, can be found as far back as Socrates. As such, deliberative democracy evades classification as a “right-wing” or “left-wing” political position - it is both, and neither, being more concerned with the way that politics are made than with their outcomes.

Contemporary research on deliberative democracy reaches broadly, consisting of interlinked discussions that are under continuous revision, interpretation and expansion. Theorists of deliberative democracy typically study the behaviour of “deliberative bodies”, such as parliaments or juries, or the effects of deliberation amongst citizens outside of decision-making institutions. Perhaps the most widely discussed aspect of deliberative democracy is Jürgen Habermas’ notion of the “public sphere”, a “network for communicating information and points of view (i.e., opinions expressing affirmative or negative attitudes)” - with nodes such as cafes and the news media - whose ultimate outcome is public opinion. (Habermas 1996, p.360). The public sphere is to society as a whole what debate is to a parliament or legal deliberations are to a court.

Unlike many other models of democracy, deliberative democracy is rarely prescriptive, in the sense of offering clear methodologies for how political institutions should be designed. It is more often analytical, a general perspective on governance and political life that might result in fairly different recommendations for different cases at different times. Insofar as this project is concerned, that is both an advantage and a problem: it is an advantage since it allows us to easily transfer arguments originally concocted with 19th century Europe in mind to the domain of virtual communities. It is a problem since there is no clear-cut prescription that merely awaits a software implementation.

The sole exception to this is James Fishkin’s protocol for “deliberative polling”. In a deliberative poll, the opinions of a representative group of citizens is sampled. They are then invited to deliberate on the issues, backed up by expert testimony, and changes in opinion as a result of deliberation are measured. The result is a method of measuring the “informed opinion” of citizens, the opinion they would state if they were confronted with all relative facts and arguments. (Iyengar et al 2003)

A deliberative poll, in other words, is not a decision-making instrument, but an alternative to ordinary polling. If we are seeking a decision-making tool, we will need to figure out by ourselves what the consequences of deliberative democracy are for software design.

1.3.3 Online deliberation

In recent years, the Internet has come to the attention of researchers in deliberative democracy. As a result, two multi-disciplinary conferences have been held - in 2003 and 2005 - on the topic of “online deliberation”, and a number books and research articles have been published or are currently pending publication. Some of these researchers compare online to offline deliberative consultations, in an effort to determine whether online deliberation is possible, and the conditions that might favour it. The empirical evidence, to date, seems inconclusive, but there are indications that better designed tools or protocols may improve the quality of online deliberation. (Witschge 2002, Regeczi 2004). Simultaneously, some of the most successful experiments of e-democracy in general, most notably the “Minnesota E-democracy” project, have been exercises in carefully moderated deliberation. (Clift 2000, Dahlberg 2001)

Democratic Interfaces

Fig 1c shows 2003 buying patterns of books on US politics, mined from the online bookstore amazon.com. Books favoured by democratic voters (blue) are clearly separated from those favoured by republican voters (red) - the overlap in preference (gray) is so small as to be insignificant (Krebs 2003). Fig 1d is a map of the US political blogosphere around the time of the 2004 presidential elections. Red arrows represent links from a conservative blog to another conservative blog, blue represent links between liberal blogs, and orange arrows represent links between liberal and conservative blogs.¹⁷ (Adamic and Glance 2005) The size of a circle represents its prominence, as measured by the number of links it receives.

These graphs tell a story of *polarization*, and of the Internet public sphere. Cass Sunstein¹⁸ worries that the Internet principally allows citizens to hear ‘echoes of their own voices and to wall themselves off from others’. This problem is not lost on the bloggers themselves and the term “echo chamber” is used not just in scholarly worries about the Internet but also in political discussion on the Internet. (e.g. Leonard 2004) The net effect of such echo-chambers are polarization: a tendency for citizens to seek the extremes of political opinion rather than compromising on the middle ground. Group polarization inside deliberating groups, where the group as a whole ends up adopting the most extreme position, is a fairly well-documented mechanism. (Sunstein 2001, pp.22-27).

Contrast fig. 1c and fig. 1d with our third graph: Kelly, Fisher and Smiths visualization of USENET discussion in the newsgroups talk.
abortion and alt.politics.bush (fig. 1e). The left circle shows messages

17 The terms “liberal” and “conservative” are used here in the American sense, as approximate synonyms for the positions of the Democratic Party and the Republican Party.
18 In “Republic.com”, as quoted in Witschge 2002.

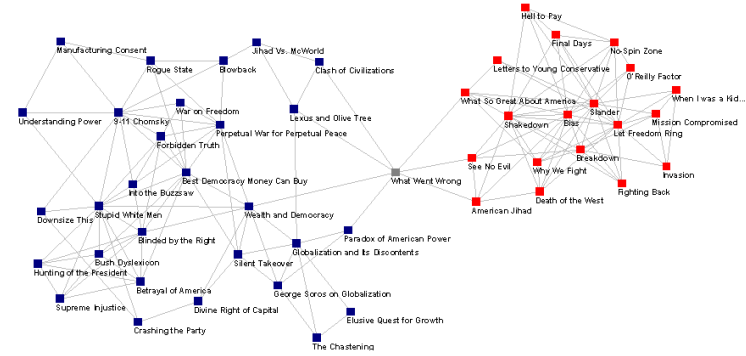


Fig. 1c: Polarization in amazon.com buyer preferences. (Krebs 2003)

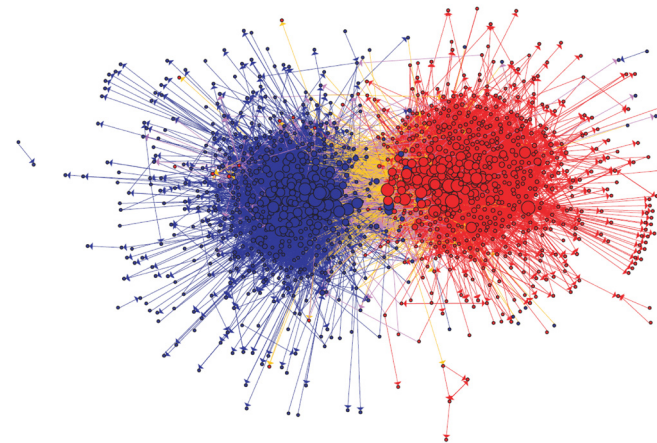


Fig. 1d: A map of the political blogosphere. (Adamic and Glance 2005).

expressing pro-life (red) and pro-choice (blue) positions, while the right circle illustrates messages that endorse pro-Bush (red) and anti-bush (blue) positions. In this visualization, we are seeing the opposite of the echo chamber: authors of widely differing opinions exposing themselves and their opponents to new facts and arguments. While we cannot say for sure, without reading the original messages, whether this is evidence of “deliberation” - it is certainly one of the prerequisites for deliberative conversation. If the kind of conversation observed in fig. 1e could be combined with a decision-making mechanism, we might be one step further towards an e-democracy tool.

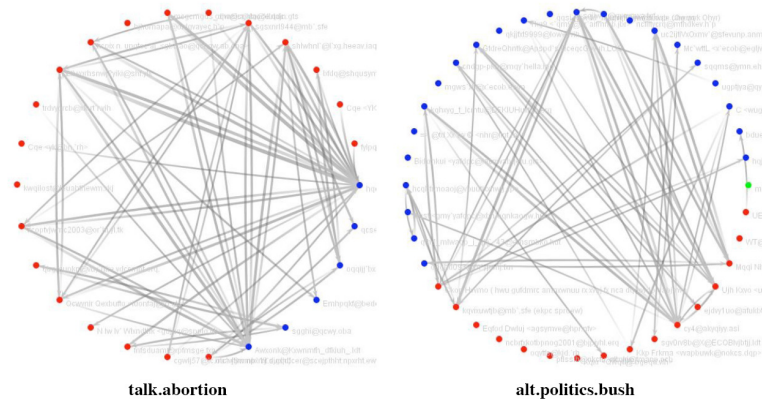


Fig. 1e: *Political discourse in USENET newsgroups . (Kelly et al 2005).*

1.4 Towards a tool for online democracy

Through this chapter, I hope to have demonstrated that:

1. To find viable forms of e-democracy, we should first study the communities that are native to the Internet.
2. Online communities are rarely democratically governed, lacking appropriate tools for self-governance.
3. Such a tool would need to facilitate structured deliberation as well as decision-making.

1.4.1 Scenarios

To further illustrate these points, and to begin estimating the future shape of our tool - the “Web Ting” - consider how the cases discussed in chapter 1.2 might have been different if the right tool had existed:

Following a religious epiphany, Linus Torvalds retires to a monastery, leaving the Linux project without a benevolent dictator and without an uncontroversial heir. The remaining community of Linux Kernel hackers and distribution authors discuss the project’s future on mailing lists. One of them sets up a Ting on his web server, and invites the rest to continue discussion there. They trash out arguments for a while, consider alternatives, and vote. At the end of the one-month deliberation, depending on the outcome of their Ting, they have appointed a new Dictator, elected a Board as their supreme authority, or decided on a process for handling all future issues of government through the Ting.

In July 2008, the redditor who calls himself 325i makes another attempt at unifying the Reddit masses around political action. He

submits his call-to-arms (“Redditors! Enough talk - let us join together for a protest!”) as a link. The link points not to the comment page but to a Ting, where redditors can propose, discuss and then decide upon the date, paroles and location of the protests. After the decision is taken, another redditor submits the results to Reddit - where it quickly proceeds to the front page - except, this time, the proposal is one most of the community can identify with and feel engaged by.

After moving into 10 Downing Street, the next Prime Minister of Britain takes a look at the website she has inherited from Blair and Brown. One of her first decisions is to remove the petition system, which right now lists as it’s #1 a petition: “J.K. Rowling should write another book about Harry Potter”. Her IT advisor has a better idea - why not let users discuss the issues, between themselves and with Government staff, so that the petitioners receive sufficient information before they begin placing their votes?

1.4.2 Design requirements for e-deliberation

So, what must such a tool be capable of doing? Our cases, and discussion of the nature of democracy, point to some design requirements:

1. Process and constitution.

The system must support a process that has the consent of a community, and can provide legitimacy for decisions taken through the system. The definition of this process is called a constitution.

- 1.1 For legitimacy to be ensured, the system must provide the community with a “process for changing the process”, a constitutional mechanism. This need not be easy to invoke, or invoked often, but it needs to be there. As long as the community has control of the process, decisions taken through the process will be legitimate.
- 1.2 Only if users have grounds for comparison are they capable of designing their own process. Despite being open to change, the tool should come with a default constitution that provides a “best practice” process.

2. Conclusion mechanism

The tool must provide a mechanism that ensures that issues raised will lead to a conclusion, even if that conclusion is abandonment of the issue.

3. Decision mechanism

The system must provide mechanisms for ending deliberation and reaching decisions. This must be designed in relation to the conclusion mechanism. Decisions might take the form of voting, polling or consensus/veto rights.

4. Discussion mechanism.

- 4.1 The system must provide mechanisms for identifying points of contention, and arguing about their right outcome.

- 4.2 The discussion mechanism should be designed in such a way as to encourage deliberation.

5. Transparency

- 5.1 The process must be transparent to users. The powers of the system owner, and the exercise of such powers, must be visible to all users.
- 5.2 User actions in the system must be transparent to other users. As a consequence, no user-created data may be permanently deleted from the system.

6. Freedom of speech

- 6.1 To ensure legitimacy of decisions, freedom of speech for the system’s users must be guaranteed.
- 6.2 Any user shall have the possibility to raise issues requiring decisions
- 6.3 To prevent abuse, the powers granted users under 6.1 and 6.2 might be constrained - made difficult, but not impossible, to exercise.

7. Security

- 7.1 The community must be able to distinguish between members and non-members, and reserve system powers for the exclusive use of members.
- 7.2 Requirements 6 and 5.2. point towards particular challenges in the cases of spam or vandalism.

2 *software, community, conversation*

Summary

In addition to facilitating a form of democratic process, e-deliberation software may be understood as a species of (social) software, enabling (virtual) community and (online) conversation.

Software, as a designed object, can be analysed through the relationship between user and tool - through the user's perception of the system's affordances, constraints, sequence and expected input. But as a medium for community and conversation, community software must also be studied and designed as a framework for inter-human communication.

In this chapter, I argue that the unifying component between software-as-tool and software-as-medium is one of genre, the expectable form of content. Users can be shown to perceive both the software frame and the community-created content when determining the genre, and the designer can therefore influence the creation of community by designing with specific genres in mind.

2.1 Software, but how to design it?

In Chapter 1, I laid out the case for a democratic decision-making tool, and defined the requirements - or design problems - that such a tool would need to fulfil. I will refer to this cluster of problems as “e-deliberation”, in order to distinguish them from the more general discourse on “online deliberation” (see chapter 1.3) as well as the specific solutions found in the CCEU and WebTing interfaces (chapters 4 and 5).

Knowing what a tool ought be capable of doing does not tell us much about what the tool should look like, or how to design it, any more than identifying a need for “time travel” would allow us to travel in time. Design methods should offer a tentative path from the known problems to potential solution. But which methods?

2.1.1 Design methods

The “toolbox” of design is a large one. It includes methods for idea generation, visualization, classification, user studies, user engagement, analysis, prototypes, mock-ups and market research.¹ Jones (2002) has distinguished between 8 types of design methods, according the manner in which they work, while Buxton (2003) distinguishes between “ideation” and “validation”. I will use a four-fold classification of the methods considered for this thesis, according to the purpose a method has in a design process:

¹ I here use the term “design methods” to lump together such different things as Lego prototypes and ethnographic field studies. Children and ethnographers might protest at this. But for design-in-the-making, both “Lego” and “Ethnography” are equally valid answers to the question: “What do we do next?”

- **Production methods** help us approximate or create the final artifact. “Sketching” is a production method, as are “paper prototypes”, “3D models” and “programming”. Idea-generating methods, such as brainstorming, also belong to this category - their approximation is less complete, further removed from the final artifact, but their function is still to approximate.
- **Validation methods** help us prove that our artifact works according to its’ objectives. Examples include usability evaluation (user tests, cognitive walk-throughs etc.), legibility testing and focus group studies.
- **Information-gathering methods** seek to uncover information or knowledge about the *context of use* for the artifact. Such methods include benchmarking (investigating prior art), consulting relevant literature, participatory observation, ethnographic studies or “contextual inquiry”, surveys and questionnaires and (again) focus group studies.
- **Analytical methods** take for granted that something has already been designed, validated, and deployed - they study the design artifact as it appears, and ask questions about it’s relationship to a model. The model might describe how such artifacts work or should work, or might offer ways of interpreting the artifact in terms of its *meaning*. I consider Jacob Nielsen’s usability heuristics² and Donald Norman’s notions of affordances, constraints and conceptual models (Norman 2002) - to be examples of analytical methods.

Analytical and information-gathering methods, apart from being of use to anthropologists or art historians, also have a role to play in the design process. An industrial designer given the task of designing a better

² http://www.useit.com/papers/heuristic/heuristic_list.html

mobile phone might first try to study how people use their current phones. Such a study serves not only to identify user needs that are not met by current designs, but also increases the designer's insight into the world of "mobile phones".

These four categories of method do not represent different stages of a design process. A process might begin (as I have in the previous chapter) by determining objectives through information gathering, proceed to produce a couple of prototypes and try to validate them, and then return to idea-generation or information gathering when validation fails. In my own design work, I typically alternate between using production and analytical methods - producing a range of sketches, and then studying those sketches through analytical methods in order to make qualitative judgments about them. Validation, usually the most expensive part of the process, is done only at larger projects - but not necessarily at the end.³

2.1.2 Contextualizing e-deliberation

Not many of the aforementioned methods can be applied consistently across design disciplines and materials. This is especially, but not uniquely, the case with validation methods: A legibility test might help the typographer determine how easily read his typeface is, but will not do much for a sports shoe. Cardboard might be all that is needed to prototype a chair, but not a car. Usability testing makes little sense for a fashion designer, unless their design includes a particularly bizarre zipper.

Before we can hope to find a method, our first question must be: what is being designed? We do not yet know what, exactly, our design will

look like, but we know its material (code) and its primary context: the world of Internet communities. This tells us what it might be similar to - to what classes of artifacts it would belong. For e-deliberation software, we can view our design as a form of **software**, as a tool supporting (virtual) **community**, and as a tool supporting (computer mediated) **conversation**. Each of these contexts merits its own discussion and set of methods.

This is hardly the first project to deal with the design of software, communication or community, nor is it the first to aim for the design of software for online deliberation. But, in the following I will argue that there are unresolved methodical problems specific to this kind of design, and outline the tentative solutions I have applied to my own design process.

³ An early usability test of the CCEU design is discussed in section 4.4.1

2.2 HCI, Usability and Perception

2.2.1 HCI and Usability

Human-Computer Interaction (HCI or CHI) is the field of scientific inquiry that studies the behaviour of humans using computers. The practice of “interaction design”, or “user interface design”, borrows many of its methods and theoretical frameworks, such as task analysis and usability tests, from HCI research.

Perhaps the best known expression of HCI is “usability”. The term is used to describe a set of methods (“usability evaluation”), the quality that those methods measure (“the usability of hotmail.com”), guidelines or heuristics arrived at through the methods (“usability heuristics”), and the community of advocates and practitioners (“the usability movement”). In a typical usability test, the yardstick of all usability evaluation methods, a user is given a set of tasks to accomplish with a given application or interface. The user is carefully observed, by an interviewer, a camera, screen capture software - perhaps even eye tracking software. Factors such as task completion rate, response speed and “User Satisfaction” (normally determined by asking the user) are measured. The resulting data indicates whether the product is “usable” or not.

Here, we encounter a problem: the user and the computer, the human and the interface. This relationship is certainly important, but for a designer of community software it pales in comparison to another relationship: that between a user and other users. The interface is of interest to us only insofar as it enables communication between members of the community. The important question is not “can they use it?”, but: “what will happen if they do?”

2.2.2 Evaluator effects and user effects

Usability testing has been shown to be prone to both a “user effect” (Law & Hvannberg 2004), a problem of the representativeness of the sampled user behaviour, and an “evaluator effect” : when different usability professionals evaluate the same product using the same method, they will often reach remarkably different conclusions. (Jacobsen et al 1998, Hertzum & Jacobsen 2001) In my own design practice, I have experienced the evaluator effect as persistent disagreements between evaluators. If we are several observers of a usability test, each observer will usually note different problems, and offer different interpretations of the usability problems discovered. The “evaluator effect” casts serious doubts on HCI theory that claims the support of empirical data from usability tests. But for the interaction designer, the relevance of a usability test might be a different one: one of observing differences between the designers ideas of how an interface will work, and the users reactions to the design. As an educational experience, a way of building design intuition, usability tests remain valuable.

Finally, the user interfaces of community software tend to be far simpler than classical objects of HCI and usability studies, such as operating systems, word processors, or 3D modelling applications. While usability of such applications should not be ignored, I suggest that it is the problem of least concern. We must look elsewhere for solutions to the problems that really matter.

2.2.3 Perception : Affordances and constraints

But we cannot expel all of HCI for dealing only with individual users and their interface. James J. Gibson’s notion of “affordances” and Donald Norman’s complementary notion of “constraints”, while

originating elsewhere, have been highly influential on HCI. These notions describe how users perceive their environment, including designed objects. In the following, I will try to show how these perspectives apply to the design of community software.

An affordance is the “perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used” (Norman 2002), or a “perceivable possibility for action” (Gibson 1986). In other words, an affordance is a possibility to interact with an object - as deduced from observing the appearance of the thing. Chairs afford sitting and keyboards afford typing, while instant messaging applications afford the rapid and impulsive exchange of text. Users interacting with an object, whether it is a doorknob or a 3D animation application, discern the possible uses of the object by observing its affordances. Such deductions may be wrong, as was the case of the mythical old seamstress who tried to use a computer mouse as a foot pedal, which is why the clear communication of affordances is a necessity in user interface design. (Norman 2002)

The converse of an affordance is a “constraint”, properties of designed objects that are perceived as hindering a particular use. Constraints are as important as affordances in communicating how an object might be used. A chair, due to the low elevation of its seat, constrains eating from it. Instant messaging applications - with their small and low-featured text input fields - constrain being used as rich-text editors, and a keyboard constrains painting or drawing. Constraints may be classified as being logical, cultural, semantic or physical. Physical constraints are built into the three-dimensional properties of an object. The size of a keyboard key - for example - constrains it being pressed with anything but a finger, and the shape of a mouse constrains it being held with any other limb than the hand, which is why that old seamstress is a

rare example. Semantic constraints rely on the meaning of the situation where the object appears, as determined by the user’s knowledge of the situation and the world. Cultural constraints are derived from signs with a culturally determined meaning, such as the convention that an “X” symbol is used to close an application. Logical constraints, finally, rely on the users ability to reason. The presence of two large buttons in a dialogue, one called “OK” and the other called “Cancel”, indicate their logically mutually exclusive use. Unlike the multiple small icons of a text processor, the user may press one or the other, but not both.

2.2.4 Perception in community software

There are significant advantages to studying interactive artifacts, under a general theory such as Norman’s, rather than the domain-specific findings and heuristics of usability. Norman’s model, as demonstrated by his application of it to such different artifacts as doorknobs and database management systems, can presumably be applied to any new design. James J. Gibson’s original theory goes even further in general application. Gibson, a perceptionist, does not limit his discussion to human beings. In his “ecological theory of perception”, affordances are properties of nature as perceived by animals. To the wolf, Gibson argues, the forest affords the hunting of prey, while to the human it might additionally afford shelter from storms and the gathering of forages. (Gibson 1986)

As with most HCI research, Norman’s analysis is framed by scenarios where a single human interacts with a single artifact. What happens when the design object is an intermediate - when its primary mode of interaction is to enable communication between communities of users? We may still talk of “affordances” and “constraints”. But the thing being perceived is no longer just the designed frame - the user interface - but

Democratic Interfaces

messages, artifacts produced by other users. A human perceiving a web forum is perhaps more likely to notice how other humans are explicitly communicating than to notice the implicit communication of the design.

Other users interacting through the object also modify its perceivable affordances and constraints. A wiki contributor adding a hyperlink to a document indicates to other editors that the wiki interface affords linking, while a forum moderator deleting a spam message indicates that the forum constrains advertising. The next question, then, is if such interactions - the chains of perception-action-perception - might take predictable and classifiable forms.

2.3 Framing Conversations

2.3.1 Synchronous and asynchronous communication

Digital conversations, as with telecommunication in general, can be grouped into four modes according to the time and location of users:

1. Same place - same time
2. Same place - different times
3. Different places - same time (synchronous communication)
4. Different places - different times (asynchronous communication)

Modes 1) and 2), those where people are at the same place but still use computer-mediated communication, are less common than modes 3) and 4). They might include a team in the same office collaboratively editing a document through an application such as SubEthaEdit, or a chat channel opened in parallel with a meeting. Of these modes, the second - same place, different times - is probably the rarest, and will in most cases resemble the fourth mode: there is little to distinguish between a person reading an e-mail from a colleague at the office from where it was sent versus at home. These modes are not necessarily a feature of technology: an exchange of SMSes might occur rapidly (synchronous communication) or be spread out over days. A “chat” application, while nominally designed to support synchronous communication, might also be used asynchronously.

All four modes are of interest to online deliberation: PICOLA uses synchronous communication to implement Fishkin’s protocols of deliberative polling (Cavalier 2005), while a related application

- Delibera⁴ - also has a role as a rapid aggregator of opinion in face-to-face meetings. The emphasis of this design exploration, however, is on tools that enable democratic decision-making in asynchronous communication. This follows from the nature of online communities: members are distributed not just in different places, but also in different time zones, and have different patterns of computer use. Even if the online community itself is based on synchronous communication - as in a chat room, or a virtual world - gathering all stakeholders for a democratic assembly at the same time would be nearly impossible.

2.3.2 Time, Sequence and reciprocity

Kurvinen (2007) has described sequence and reciprocity as important features of computer-mediated social action. By reciprocity I mean the expectation of return - the assumption that a question asked will be answered, an opinion voiced will be discussed. Sequence is the order of actions, or messages, in a conversation. Sequence can be understood along two different dimensions:

1. The actual sequence of messages, when users did what.
2. The displayed sequence of messages, the order that messages are read in, and the relationships displayed in the system.

In web discussions, the second dimension is rarely identical with the first. The displayed sequence often encodes not just the actual sequence, but also the reciprocal relationships between messages.

Display forms are commonly classified as being either “star-shaped” (fig. 2a) or “tree-shaped” (fig.2b). In a star-shaped discussion, messages are ordered in a linear manner, with either the oldest or the most recent message displayed first. Several such “threads” or “topics”, unified by a

⁴ delibera.net

Democratic Interfaces

title, can be active at the same time - giving rise to the metaphor of a star shape.

In a tree-shaped discussion, messages are ordered as sequences of replies. Each message is displayed below and to the right of the message that it is a response to. Thereby, the properties of sequence and reciprocity become entangled.

The combination of display form and reciprocity can be seen to play an important role in online discussions. An argument that is not gainsaid easily appears to be the “winning argument”. This is especially the case if the most recently posted message is given the highest priority in the sequence, which often is the case in star-shaped discussions. An expectable byproduct of such discussion interfaces is the “never-ending discussion”, where users defend their position and honour by repeating the same arguments endlessly. As time passes, and no resolution is found, the discussion might degenerate into a “flame-war” - discussion characterised by highly aggressive emotional content but little factual content.

The tree-shaped discussion strikes a balance. Originating messages are given the highest visibility in the sequence - that of being to the left, and oriented towards the top. Time is still present: frequent users will notice the changes since their last visit, and the software might highlight newer messages through other visual cues than sequence. But as the discussion proceeds, replies move towards the right edge of the screen, losing visibility and importance. It is perhaps not coincidental that some of the higher-quality discussions on the Web, such as those on Wikipedia’s discussion pages, follow the tree-shaped pattern.

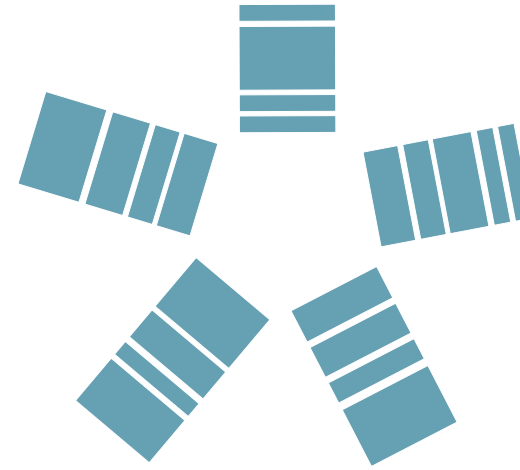


Fig. 2a: “Star shaped” discussion threading.



Fig. 2b: “Tree-shaped” discussion threading

2.4 Community as a design object

2.4.1 Software as design material

One of the ways we can see “design”, is as engagement with the properties of a given material. A graphic designer works with paper, ink, and printing methods while an architect might work with concrete and glass, or wood and stone. In order to design effectively, the designer needs to take the properties of the material into account, alongside the available technology for manipulating it: wood is bendable in a way concrete is not, and a digital printer will make the serifs of the 10pt Garamond typeface appear differently than an offset printer. A designer adds such intangibles as “form”, “function”, “beauty” and “meaning” by working with, and by initially choosing or accepting, materials. The design of software represents an exceptional case, since its primary material is an intangible: *code*.

Code has no properties in and of itself, other than those given it directly by the programmer and indirectly by the tools used to develop and run the code. While no human intention guides the design of a tree, software consists of nothing but human intention.

What do we know about code? We know that it is meant to be executed by computers, but also to be understood by humans - which is why it is usually written in a programming language, and then translated into the language of digital circuits. We know that a program is almost never executed alone, but is entirely dependent upon an *ecosystem* - such as the kernel, APIs and libraries of an operating system, or the servers and protocols of the Internet. We know that the creation of code involves design, but a different *kind* of design - according to different criteria (elegance, speed, extensibility, etc.) than those of interaction design. We

know that the design of code constrains the design of the user interface; and that user interfaces are sometimes designed because a particular new kind of code enables it. We know that the design of new code represents a considerable challenge, perhaps more than any other kind of design, because unwritten code consists entirely of design problems and because computers, unlike humans, will react to incomplete or ambiguous reasoning by refusing to do anything at all.

2.4.2 Code as Law / Design as Frame

We know, by now, that the design of code, and the shape it gives to user interfaces, affects the shape of the things that humans produce with digital tools. Lev Manovich (2002) points out that a currently dominant style of animation has been influenced by Macromedia’s Flash tool. Some animation techniques, such as zooming, have become far more common in the Flash age than they were in the age of hand-made animation, while others - such as detailed textures - have become more rare. The popularity (or lack of popularity) of these animation techniques - the Flash Aesthetic - follow from the simplicity (or lack of simplicity) of applying them in Flash. Clay Shirky describes the unintended consequences of electronic conversations as largely an issue of code: “If you assume a piece of software is for what it does, rather than what its designer’s stated goals were, then mailing list software is, among other things, a tool for creating and sustaining heated argument.” (Shirky 2002). Lawrence Lessig, describes code - or rather, the consequences of code - as an analogy of legislation. Lessig describes four “regulating agencies” (Lessig 1997) or “distinct modalities of regulation” (Lessig 1999, p.88):

Democratic Interfaces

- a) law, the legal system
- b) the market
- c) social norms
- d) architecture (the built or physical environment)

In Lessig's analysis, these four kinds of regulation combine to prohibit or encourage human behaviour - and their effects may counteract each other. The legal system, might severely punish the use of drugs. But the market provides a ready source of drugs, and drawn curtains allows for the privacy of using them. Social norms might in some communities (e.g. the Hippie movement) encourage the use of illegal drugs, while in others (e.g. the school or family) punish their use.

The Internet analogy of "architecture", according to Lessig, is "code". And it's function is similar: copyright law might constrain the sharing of copyrighted works - but the market, social norms and especially the design of peer-2-peer networks all conspire to encourage it.

Lessig uses the term "code" in a different way than I do above: not to describe the material aspect of software, but as a descriptive term for the regulating modality of the Internet, the built environment. Lessig's original text should be read as an argument against the position that the nature of the Internet is something inherent and unchangeable (and hence impossible to regulate), by pointing out that "code" is not a law of nature but the result of deliberate and malleable decisions taken by its architects and programmers. (Lessig 2006)

As I will talk about it here, Lessig's "code" is the same as "design" - both of the user interface/end user kind and of the design of code. If the argument holds, we will find that various forms of behaviour are enabled or prohibited, encouraged or discouraged by the design of the

software in use. And the conclusion we must make based on Manovich, Shirky and Lessig's observations is that **the design of software regulates the behaviour of it's users** - not just in the sense of permitting or prohibiting, affording or constraining, but in the sense that users of software are expressing themselves in ways that would not have occurred to them if they had not used these particular digital tools.

Altogether, these considerations (perception, regulation) tells us something about how virtual communities work, how design affects them: it places interface design in a position roughly analogous to that occupied by political culture or meeting leaders in analogue life. But the perspective of "code as law" is even more central to the task of designing for digital democracy, since it's implication is that we must see the design and capabilities of the user interface as being akin to the constitutions and rules of order that frame the democratic conversation.

2.4.3 Community and genre

Philip Agre, in the paper "Designing Genres for New Media", defines a "community" as "a set of people who occupy analogous locations in social or institutional structures, and a "genre" as "an expectable form that materials in a given medium might take."

These definitions are wide, in the sense that there are a whole lot of things that fit Agre's definitions that are not normally called "communities" and "genres", but also narrow, in the sense of being unambiguous. A family may be considered a "community", and a form of conversation between parents and children in that particular family may be considered a "genre". But also all Norwegian men with the role of "father" may be considered a "community", and books giving parenting advice to such men may be considered a "genre". At

the centre of Agre’s model we find the idea that genres are utilized by communities in order to support the activities of that community, and the accompanying prescription that new media designers focus their attention on the design of genres. (Agre 1998)

We may detect genres amongst the comments of Flickr users, amongst colleagues or friends utilizing instant messaging, in the duration and editing style of YouTube videos, or in the voice chats of players in “World of Warcraft” Looking at online behaviour in terms of “genre” and “community” does not prevent looking at them through any other lense, for example as “actors” in “networks”. But Agre’s model is particularly interesting, because it gives us a way of classifying the aforementioned chains of perception and actions undertaken by users in an online community: they form genres, and genres are created and used by communities. Thereby, we can make some tentative statements about the framing of online genres:

1. A genre is an expectable form: users will recognize and imitate a genre.
2. The perceivable affordances and constraints of the user interface frame the likelihood of genres arising.
3. Genres are subject to the regulatory power, the “code-as-law” of the user interface. A user interface may make some genres possible, others impossible.

To illustrate, consider a simple example from my own life: a circle of friends who met while studying in Denmark, but currently live in different countries. In order to stay in touch, we use Skype - an instant messenger and VOIP application. Some of our conversations have purely factual content - “Are you coming to Oslo for Christmas?”. Some serve the looser purpose of “staying in touch”: “What’s up?”

“Working hard on my MA thesis. And you?”. Both of these are genres, the first characterised by complete sentences and to-the-point communication, the second by “chatting”. In some cases, a third genre, that of the “emoticon exchange” intervenes. Here’s an example:

A: Coming to Copenhagen for New Years?

B: Yep!

A: 😊 🎉

B: 😎 🍺 🍷

A: 🤢

B: 😁

In this example, we know the community: a circle of friends who studied together. We also know that this circle is spread, and that most of the people are fairly busy - especially during their “online time”. Members of this community recognize the beginning of an emoticon exchange by the use of multiple emoticons in a message. This is the “expectable form”, the genre. The “emoticon exchange” serves as a form of entertainment, and as a simple way of expressing connection, appreciation, of maintaining friendship - without taking up too much time. An emoticon exchange can be maintained while working, while feeding a baby, while writing an MA thesis. This is the function that the genre has for this community.

The emoticon exchange is made possible by the user interface in Skype (see fig. 2c), and made desirable by the brevity of messages in Skype. This illustrates affordances, constraints, and architecture respectively.

Skype not only enables animated emoticons - a question of architecture - but also clearly affords them by placing the “emoticon palette” directly above the new message box.

While the emoticon exchange developed a specific purpose in the aforementioned community, identical genres may arise in other communities of Skype users, as a result of the affordances of the Skype user interface. Emoticon exchanges might be used for flirting, as a way of “getting to know each other” (rather than “maintaining contact”), as a substitute for words by communities uncomfortable with textual conversation.

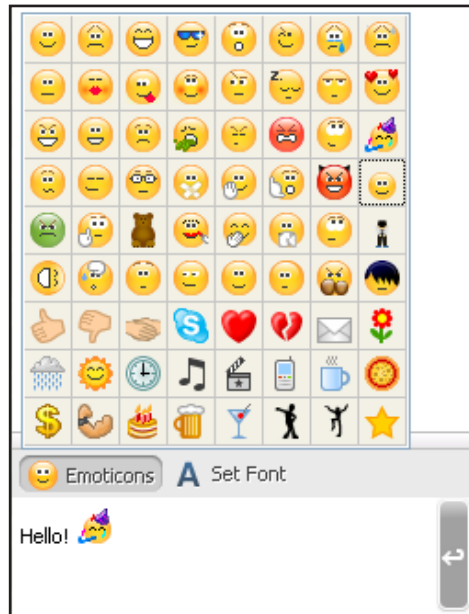


Fig. 2c: Skype's emoticon palette

2.4.4 From analysis to design

In the preceding, I have outlined an analytical model for community software design, taking into account the combined influences of affordances and constraints, the regulatory nature of code, the sequence and reciprocity of conversations, and the genres formed inside these frames. To conclude, I will explore some of the design implications of this model:

1. The central consideration for a designer seeking to control these factors lays in the design of an interface in its initial, empty state - the state of *tabula rasa* - and in considering how the tabula rasa will be perceived. The Skype UI (fig.2c), in its tabula rasa state, clearly affords the use of emoticons. The small size of the message box indicates a constraint, a prevention of long messages. All genres formed by Skype users originate from the affordances and constraints of this empty interface.
2. Additionally, we must consider the sequence and expected reciprocity of messages. Which kind of responses does the user interface afford, and how are they valorized in the display of sequence?
3. Finally, but on a case-by-case basis, we should consider the community: which communication needs does this community have? Which genres might facilitate them? How does the interface support those genres?

3 *Process*

Summary

This chapter discusses the initial steps of the design process, the tentative movement from design problems to multiple design solutions, and their consolidation towards “final designs”: sketches, scenarios and visual reasoning.

In particular, the pattern language in appendix A, a set of annotated design patterns and feature proposals, is discussed. The ideas presented in the pattern language became the foundations for the CCEU and WebTing designs.

3.1 Problems and solutions

Let us recap: we began this design exploration with an idea that tools for online democracy might be a good thing, and formulated some design requirements that such a tool should fulfil. Immediately we encountered a problem: there was no obvious theory or methodology for how such tools should be designed. Having discussed some available theories, and proposed some ideas of our own, we arrive now at the phase where actual design is being made. This chapter documents some of the early, and most critical, stages of the design exploration: The steps taken to overcome the almost staggering complexity of the task, and begin searching for *design solutions* to the *design problems* - in order to approach a “final design”.

3.1.1 The nature of design problems

Design is often described as a problem-solving activity, but neither design problems nor design solutions are straightforward entities. Design solutions tend towards the holistic, the complete, but also the satisfactory and sufficient (Lawson 2004, p.12). Cross (2006, p.7) has described design problems as being characteristically ill-defined: despite the design process often being initiated by a requirements document, a client briefing or similar - designers almost invariably end up identifying, or re-defining, the problems only by solving them.

The chapters subsequent to this will document the CCEU and WebTing designs, two projects so conceptually different that it might be difficult to imagine that they had the same origins. They are both design solutions arising from the same cluster of problems, but they have ended up defining their problems differently.

Bruno Latour has described scientific fact through the metaphor of the “black box”: once a point is sufficiently strong, sufficiently accepted in the scientific community - it passes from being an item of debate, of rivalry, of creativity or exploration and becomes a simple, straightforward fact: we are left with the elegant exterior of the black box, everything else is hidden inside. (Latour 1987) Design is not science, but what I here call “final designs” are similar, in many ways, to Latours black boxes: the goal of the design process is to produce them, and - once they are produced - sweep away all the alternatives uncovered by the process as if the design object was a single eternal artifact that never could have been any other way. None of the designs presented here have reached such a stage of finality, of black-boxing.

As I touched on in the opening of chapter 2, some of the principal methods of design deal with production (or “ideation”) the approximation of a final design. Even a simple, hand-drawn sketch of a user interface is in some way an approximation of a black-boxable final design. It is by exposing our approximation to various tests - is it beautiful? usable? does it achieve what we want? - that we can determine whether it is a candidate for black-boxing or not.

The “holistic” aspect of design solutions, mentioned above, means more than that a design must “hang together”, form a whole. It also means that the whole must solve several different problems, simultaneously, in a way where the solutions not only avoid contradicting each other, but merge together into something more. When faced by the complexity of the problems I encountered, my initial approach was almost the opposite: one of breaking down problems and possible solutions (or solutions, and possible problems) into smaller units, and then seeing how those smaller units might

be assembled to form designs that were holistic, functional, black-boxable.

The “breaking down” part of this process took the form of a number of different sketches, including the “problem map”(3.1.2), and the e-deliberation pattern language (3.3). The reassembling took the form of scenarios, wireframes and blueprints.

3.1.2 Visual reasoning and the “Problem Map”

The “problem map”, fig. 3a was produced in the early stages of the project, at a point where I faced great difficulties in finding a focus and structure for my thesis work, by writing down design problems and issues that needed illumination on individual notes (figure 3b). These notes were compared, duplicate or overlapping issues were merged and the language was harmonized. The notes were reorganized topically into different groups, until a natural grouping emerged.

The “problem map” has two dimensions: field or topicality is indicated by colour and proximity, while the position on the vertical axis sorts problems according to their degree of being underlying and abstract (top/periphery) or specific and concrete (centre-bottom). Where applicable, the problems are organized into threads showing how the answers to more specific and concrete questions may follow from the more general and abstract ones.

Edward Tufte (1997) has documented how visualization may be used for presenting explanations and arguments. Donald Norman (1993, p.43) calls these “external aids”, and ascribes them a dual function: that of communicating between senders and receivers, but also that of externalizing thought, increasing the designer’s own comprehension of the shape and structure of the subject matter. John Chris Jones

(1992, p.45) describes the function of several new design methods as one of “making public the hitherto private thinking of designers: to externalize the design process”.

The problem map is a special kind of design: it is an “external aid”, it “externalizes the design process”, but it is not production or ideation - it does not approximate a final design, instead it attempts to externalize the ideas and argument - the visual reasoning - behind a design. This practice overlaps with “visual communication”, in the sense that visual reasoning may also be a prerequisite for efficient visual communication, but I have found it to be a useful design practice on its own terms.

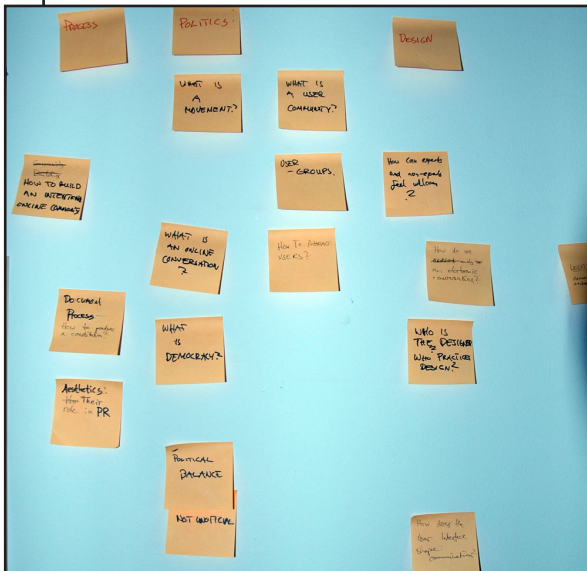
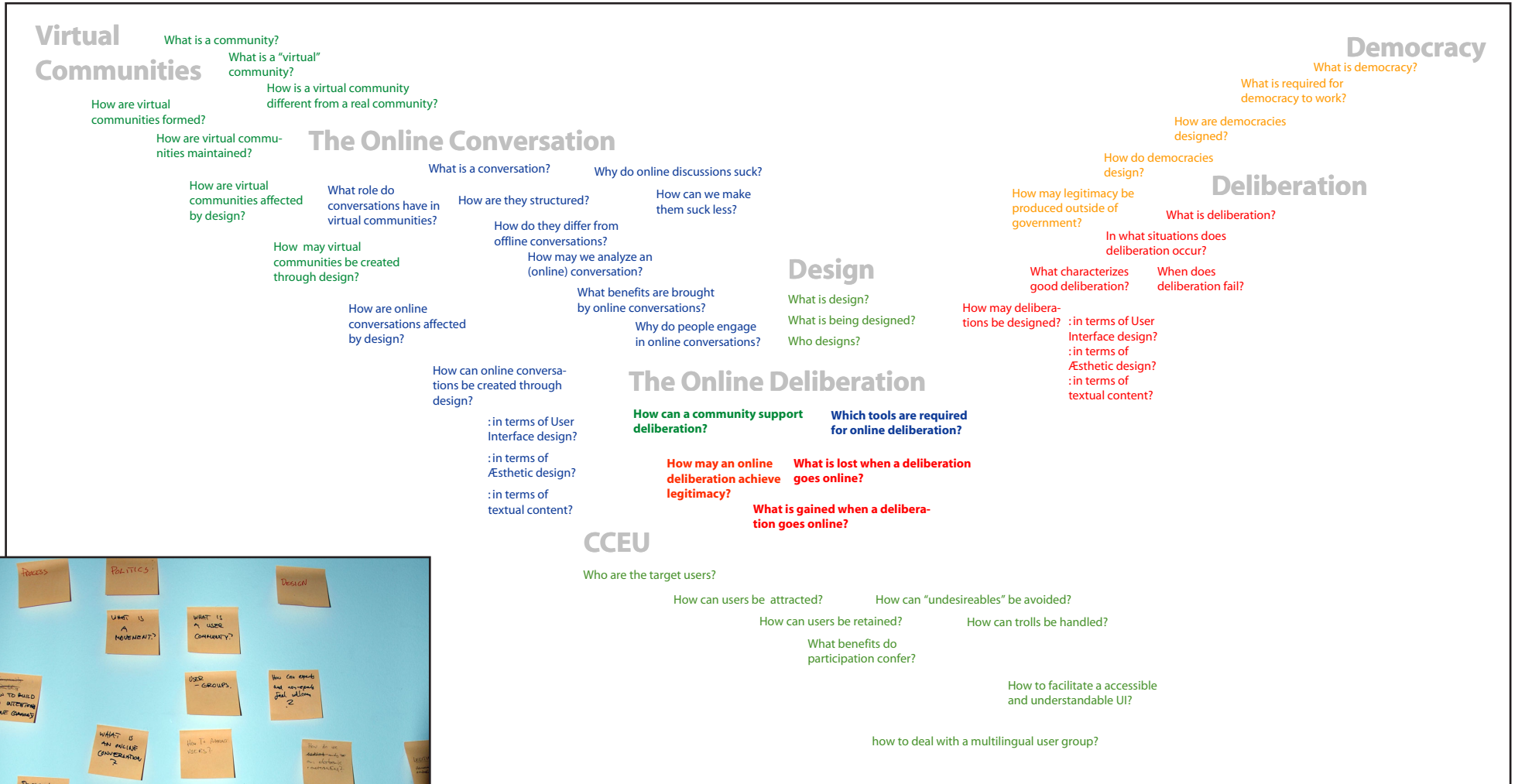


Fig. 3a (left) and 3b (above): The "problem map", a visual representation of the structure of problems faced in the design process. The "problem map" was made for purposes of visual reasoning, to help me comprehend the subject matter. The diagram above was based on the note-sorting exercise shown to the left.

3.2 The pattern language

In the early process, the ideas came in bits and pieces: there were sketches on paper napkins, a nifty feature found on some obscure site, a sudden insight reached while reading a paper on a political organisation and pondering what the organisation might look like as a user interface. Each of these were design solutions, but they were not solutions to any of the larger problems.

The challenge became one of connecting these bits of pieces into something more, something holistic and black-boxable. But the number of new ideas, alongside the features of established community software that needed consideration, was disorienting. Not because there were many of them, but because it was hard to see any common shape, difficult to see how they might fit.

In an attempt to solve this problem, I began writing down these ideas, looking for some standard format to describe them. After some searching, I found that the best fit was the format known as a “pattern language”.

A design pattern consists of a design problem described together with its best known solution. A pattern language is a collection of design patterns belonging to a given domain, written in a way clarifies their inter-dependencies. The idea of pattern languages originates with the architect Christopher Alexander, and the first pattern language described architectural patterns, but pattern languages have been adopted by several other disciplines - including engineering, computer programming, and interaction design. (Borchers 2000, Winn and Calder 2003)

Pattern languages have traditionally been used to document best practices and the established knowledge of experts, rather than novel and hypothetical design patterns. The pattern language I developed for this design exploration does the opposite: it describes some best practices and established knowledge, but only insofar as they are needed to contextualize and support new ideas. This transforms the genre of pattern languages from a teaching tool to a tool for concept development and ideation.

By freely mixing new ideas with well-known features, it was possible to get a sense of the shape of future designs - of the neighbourhoods where they might fit. The pattern language serves the purposes of providing an overview of the domain, of isolating individual patterns from their conventional context. Most importantly: it acts as a palette from which new concepts are created.

3.3.1 Sample patterns

The full pattern language is included in this thesis as an appendix (Appendix A) rather than in the thesis text, as I consider it to be a work in progress. Some of the ideas described in the pattern language are worth mentioning here, since we will soon meet them again:

The Editable Text (Fig. 3c): A document that can be edited by anyone, accompanied by a version history accessible through a “history” button. In other words: it is a wiki interface, and an example of how features of pre-existing software forms are included in the pattern language.

The Proposition (Fig.3d): This pattern is built on top of the editable text, and describes a document that users can vote to finalize.

Collaborative Discussion (Fig.3e): Using a structure similar to propositions, the collaborative discussion occurs between groups who author their messages collaboratively as editable texts.

Dispute freeze (Fig. 3f): What happens if two users continue reverting each others edits - an “edit war”? In normal wiki systems, this problem lacks clear solutions. The dispute freeze suggests a solution: if an edit war is detected, the document “freezes”: no further edits can be made until the conflict is resolved. Other patterns describe potential conflict resolution mechanisms.

Iconic reply (Fig. 3g): Might there be a way to rapidly communicate the strength an ambiguity of opinion? To encourage less active users to express their position - albeit briefly? The “iconic reply” uses emoticon-style symbols to do just that.

3.3.2 Reflection and potential improvements

While the pattern language was integral to my design process, that process also revealed flaws and unexplored possibilities in the design and execution of the pattern language:

- **Incompleteness:** the pattern language does not provide a full overview of community software design patterns. It is biased towards those features that I found most interesting at the time. A more complete pattern language should also challenge the bias of both reader and writer.

- **Poor granularity:** The pattern language does not explore design patterns down to their smallest components. It contains patterns such as “editable text”/”wiki”, but not “edit button”. Many of the insights to arise from the pattern language came from imagining smaller patterns, such as the “editable text”, applied to a different context than where it is originally found.
- **Difficult to visualize:** While plain text works well as a way of describing and annotating patterns, it does not provide overview-at-a-glance or a possibility to rapidly explore patterns. In this sense, paper prototypes or user interface wireframes offer more opportunities for rapid design exploration.
- **Does not capture user-driven patterns:** The pattern language is biased towards the influence of the user interface. But this is only one half of the equation that makes online community - the other half consists of the genres and practices of the users. It would be interesting to describe both socially and technically constructed patterns in the same language, showing how some problems might have either a social or a technical solution.

A final question we might ask of the pattern language is: what does it describe? Since it mixes “common knowledge” or “best practices” with hypotheticals, this is somewhat blurry. Some of the patterns, such as the idea of “iconic reply”, describe an element of a user interface - while others, such as “moderator role”, describe a structural pattern. The user interface expression of the “moderator role” might consist of patterns such as “delete button”, “user status panel” and “move message button”. Introducing a separation between “structural” (or underlying) patterns and their potential expressions might further enhance the value of the pattern language as a tool for design exploration.

Fig. 3c: Editable text



Fig. 3d: Proposition



Fig. 3e: Collaborative discussion

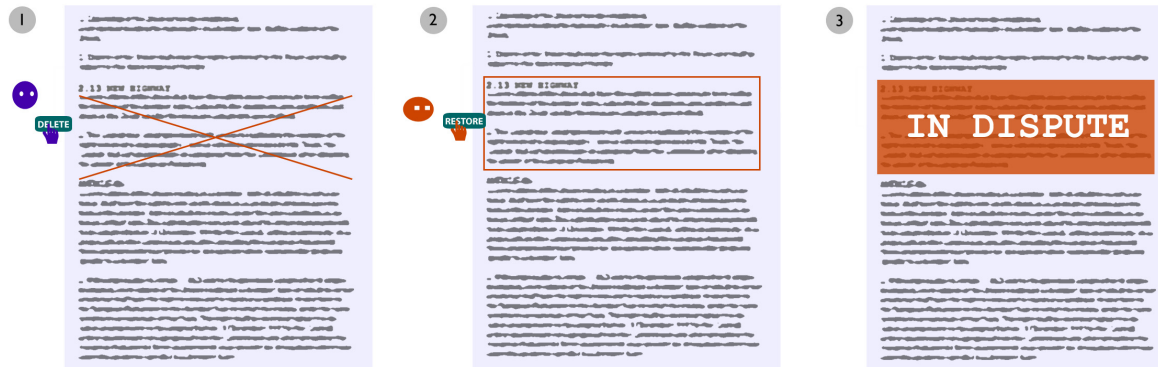
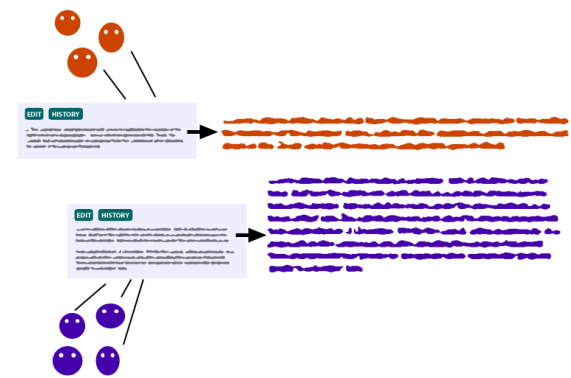


Fig. 3f: Dispute identification

? Should we support legalization of Khat?
 User **Ann Petersen** posted the following proposal on Mar.13.2006, 20:35
 Khat is an African herb that gives a mild, nicotine-like intoxication. It is currently listed as an illegal drug. This law is racist, as Khat is primarily imported by African immigrants. Should our organisation support legalisation of khat?

- Bob Phearson** Mar.13.2006, 20:45
- Jean-Francois Nabama** Mar.13.2006, 20:52
Khat prohibition is racist! We should take a clear stand against this.
- Abdullah Öztürk** Mar.13.2006, 21:01
- Trine Hansen** Mar.13.2006, 21:14
What are the medical consequences of khat?
- Lisa Simpson** Mar.13.2006, 21:42
- Gerhard Mackey** Mar.13.2006, 22:11
Drugs are bad. We should support prohibition, irrespective of race.

Fig. 3g: Iconic replies

Democratic Interfaces

An improved pattern language should:

- Map out both socially and systemically enforced patterns.
- Be accompanied by, or integrated into, a visual language that allows for visualization, rapid recognition and recombination of patterns.
- Further separate problems from their solutions, allowing the designer to consider alternate patterns for a given challenge.
- Separate structural patterns (e.g. the notion of responding to a message) from their various implementations (e.g. a reply button, a tree-shaped discussion).

3.3 Scenarios: from disconnected patterns to complete design

Scenarios may be used by a designer as a way of exploring the design, of mapping out its consequences, or as a way of communicating the design to clients or colleagues. I used scenarios primarily for the former, explorative purpose: imagining situations where actual users would be interacting with software based on subsets the pattern language. The scenarios served to move the process from disconnected ideas (patterns) towards “complete designs”.

Some of these scenarios were written down and sketchily illustrated, some were purely mental exercises. The scenario on the following pages is presented as a representative example of these. The scenario revealed both opportunities and problems: some of the ideas that appear in the scenario were invented as a result of writing it - “how does Bob get from A to B?”, while some of the ideas presented in the scenario are shown to be problematic: the “dispute mechanism”, for example, makes it inordinately easy for an individual citizen to block a process that affects many citizens.

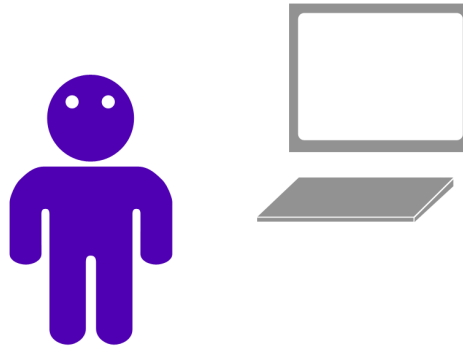
Furthermore, the case presented in the scenario, that of a conflict between urban planners and a group of neighbours being resolved through online deliberation, is not particularly plausible. Why would a group of neighbours, who can easily meet and deliberate face-to-face, need a website to conduct their discussion? The advantages of such mediation - rapid communication with urban planners, and easy access to previous discussions - are cancelled out by its disadvantages: the slower speed of writing over talking, and the fact that such a process

favours the digitally literate. This issue led me to narrow the focus of my design towards tools that primarily address the needs of online communities.

3.3 Scenario



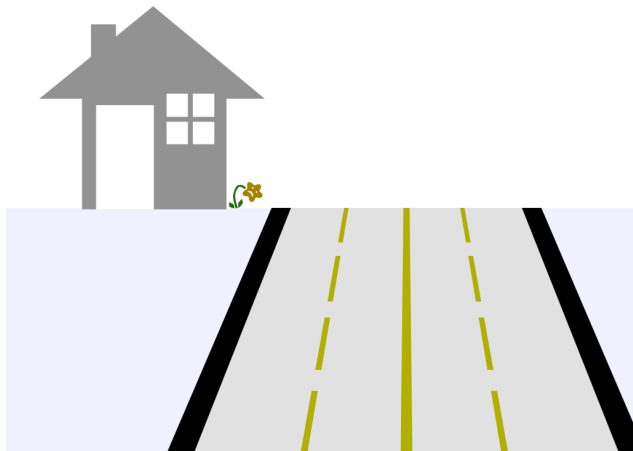
This is Bob, a resident of Evergreen Terrace.



Bob is an active Internet user, and frequently reads websites about local issues.



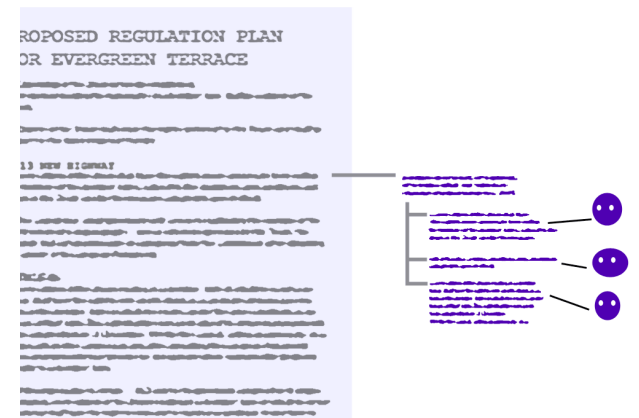
One day, on the local council's website, Bob discovers a new regulation plan for his area.



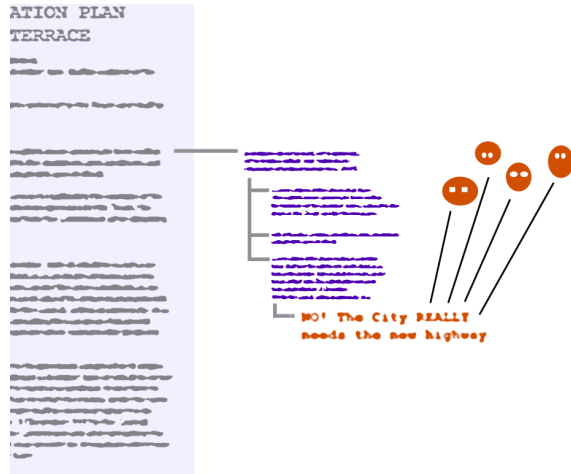
The plan calls for a large, new road through the heart of Evergreen Terrace, pass right by Bob's house.



Bob writes a comment to the plan, criticizing it.



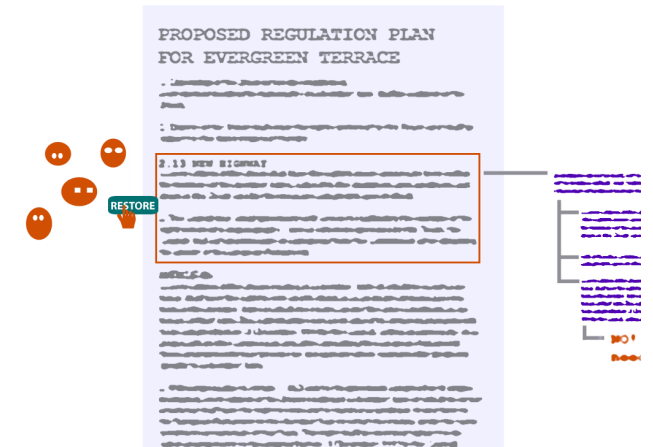
Word spreads through Evergreen Terrace, and several neighbours join in the criticism of the plan.



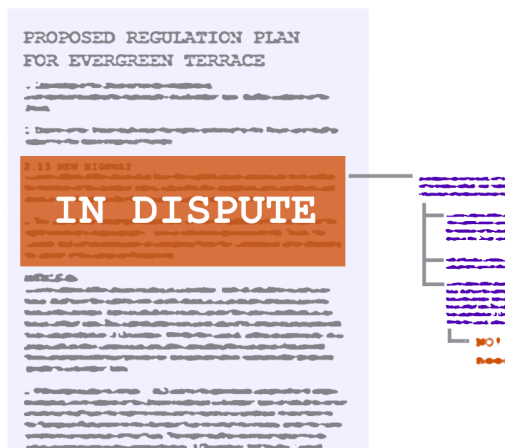
But the city council is not convinced.



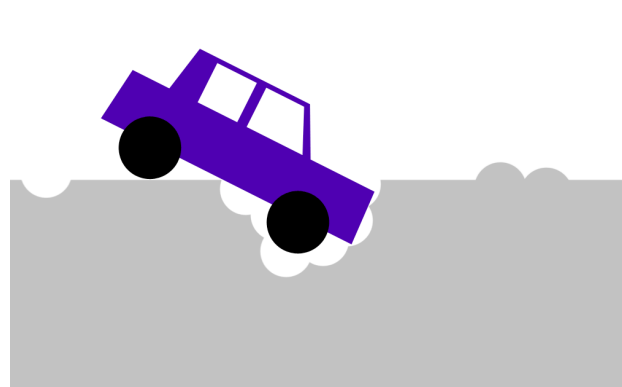
One of Bob's neighbours takes the dramatic step of removing the new road from the plan.



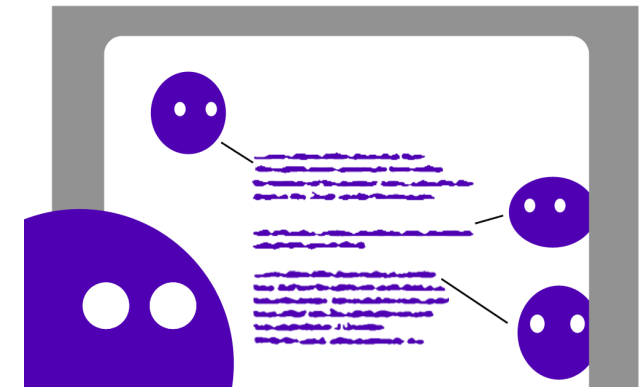
The council does not accept this change, and re-inserts the section about the road.



As a result, the section is marked as disputed, and the whole document is locked, unable to proceed until the dispute is resolved

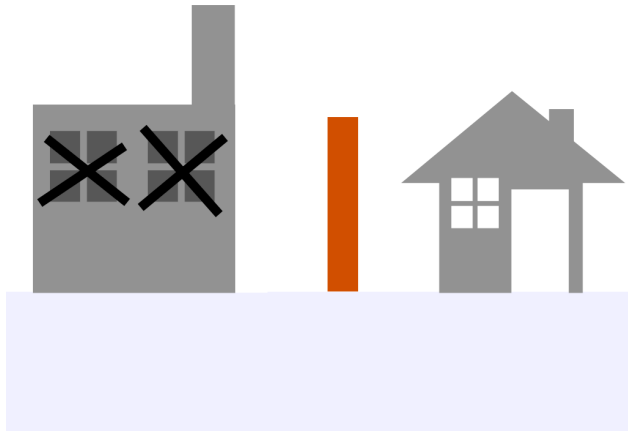


This situation is bad for all parties. The roads of evergreen terrace badly need repair, but nothing will happen while the regulation plan is locked.

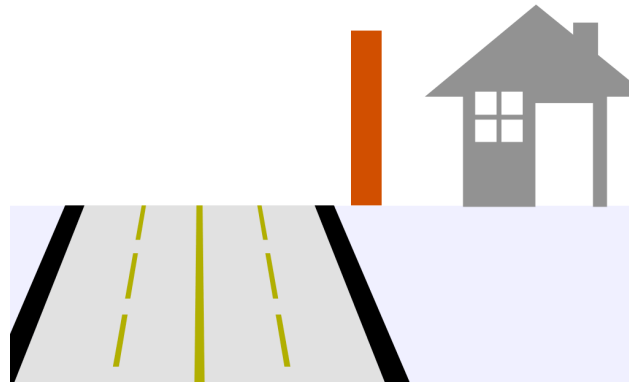


Bob and his neighbours discuss what to do about the situation.

Democratic Interfaces



Sam - another of Bob's neighbours - brings up the abandoned industrial area next to his house.



Perhaps the road could pass through there instead?



The neighbours post this proposal as a way to resolve the dispute.



The city council approve of the compromise...



... and all was good.

4 *citizensconstitution.org*

Summary

The Citizens Constitution of Europe (CCEU) is an experiment in online political activism, and in the use of community software to facilitate deliberative democracy. Intended as a constructive form of political protest, the website (www.citizensconstitution.org) invites European citizens to participate in an open, online process to draft an alternative constitutional text for the European Union.

The design of the CCEU website moved through several phases, based on an initial design concept - the “WikiTing”, modified and simplified in order to be easily implemented using open source software. As such, it is a test of my hypotheses on online deliberation and an ongoing experiment intended to map out the ways by which a user community might subvert the available software to support a process not encoded in it.

4.1 Political context

In the autumn of 1994, on the eve of the EU referendum - the most important plebiscite since Norway declared independence from Sweden in 1905 - I sat on a commuter train from Oslo to my home suburb of Opegård trying to make up my mind about which way to vote. While squeezed up in a corner of the carriage, I was given two different leaflets by two different activists - one encouraging the reader to vote Yes! - the other to vote No! I read both. The Yes! brochure mentioned the sorry plight of the Swiss, who a year earlier had rejected EU membership and now regretted it bitterly. The No! brochure also mentioned Switzerland, as a land of milk and honey prospering thanks to their sensible decision to stay outside of the European Union.

That evening, I voted blank. It was the first vote I ever cast. 52% of Norwegians voted no, 48% voted yes. Blank votes were not counted. I still do not know what difference it would make if it had been the other way around, if one out of twenty Norwegians had voted “yes” instead of “no”. No-one can really say for sure. The debate before that referendum had plenty of passion, the kind of political passion that splits families, and very few facts. How was one to acquire facts? The Maastricht treaty, the guiding document of the European Union, was some 40.000 pages long, written in terse legalese. Only lawyers could truly understand the text, only experts on government were able to form an opinion about its potential consequences. The influence of the EU on members and non-members alike was opaque - regulations were written by faceless committees in Brussels, and then “implemented” in national law. Voters are rarely aware of this - the national laws are framed as national issues, even if they have a European origin. To what degree is a controversial law authored in Brussels? Very few citizens of EU countries are able to answer such questions.

In 2003, Sweden and Denmark rejected the Euro by strong majorities. Why such an empathic rejection of a change in monetary policy? After the Swedish rejection, an anonymous EU commissioner admitted to the Financial Times: “When we ask voters a European question, the answer is either no, or yes by only the narrowest of margins... That should be telling us something.” But what should it be telling them, and were they listening?

The European Union has a severe democratic deficit, a lack of democratic influence. The “democratic deficit” is not a matter of euroskeptical rhetoric - it is a term used by researchers of the EU in reference to something they consider a well documented fact. It is being discussed in much the same way as one might discuss the budget deficit of a national economy. (Bellamy and Castiglione 2000)

In a widely published 2001 article “Why Europe needs a constitution”, Jürgen Habermas called for the creation of a constitution of Europe (Habermas 2001), to address the democratic deficit, to differentiate the “old world” from the “new world” and as a way to include more citizens in the European project. Whether due to Habermas’ influence or not, the EU leadership - the Commission and heads of state - began working on what was referred to as a “Constitutional Treaty”. The result of this process, the “Treaty establishing a constitution for Europe” was some 500 pages long¹. Compact, perhaps, in comparison to the Maastricht Treaty, but longer than any other constitution of any national or supranational entity in history. The Norwegian constitution of 1814, in the same print, would take some 10 pages. While Habermas encouraged voters to support the Constitutional Treaty, due to necessities of global politics, the text could hardly have been what he had in mind. Nor, apparently, was the debate:

¹ Depending on language.

“While in the normal case a people decides on its own constitution, the European constitution must result from the supporting votes of 25 peoples, and not from the common will of the citizens of Europe. For there is still no European public space, no trans-national bundling of themes, no common discussion. Each one of these votes takes place within the bounds of the individual country’s public sphere. This asymmetry is dangerous, because the primacy of national problems, for instance reservations about Chirac’s government, can obstruct the view of the problems actually posed by the acceptance or rejection of the European constitution.”
(Habermas 2005)

54% of French voters ended up rejecting the constitutional treaty. Shortly thereafter, 64% of Dutch voters did the same, effectively killing the treaty and leaving the EU leadership in disarray about how to proceed. At present (as of autumn 2007), a second attempt at introducing an EU constitution - the “Treaty of Lisbon”, is being prepared for ratification in 2008. It is not likely to be any more popular than the Constitutional Treaty was.

The idea behind CCEU was conceived not long after the Dutch referendum. It was not surprising that the constitutional treaty should be rejected. How can one have a reasonable public discussion about a document that consists of 500 pages intricate legalese? How can one call such a monstrosity for a “constitution”? Neither should one be surprised that voters decided on their vote based on “gut feeling” and on national issues: these were, after all, what voters could relate to. The constitutional treaty, like it’s predecessors, was the product not of deliberation or of any unified position on the future of Europe: it

had been arrived at largely by bargaining, by heads of state trading concessions in lengthy meetings hidden from the public view.

The idea was simple: open a website running Wiki software, and invite as many citizens as possible to use it to author an alternative constitution for Europe. In the best of all possible worlds, this constitution will be ratified as the future constitution of the European Union. In a more likely scenario, the project would serve as the unorthodox vehicle of a message, aimed at the EU leadership. This message would be both pro-European and anti-EU. It would reject the Constitutional Treaty (and the Treaty of Lisbon), but support the idea of a European Constitution. If anything, the message would be anti-bargaining and pro-deliberation: a message about means rather than ends.

The design challenges introduced by this idea were more complex. Wikipedia, the best known wiki site, is a tremendously effective aggregator of knowledge precisely because it deals with facts - things that can or should be beyond dispute. Wikis are an effective collaboration tool, but it can be difficult to identify and resolves disputes in them. A telltale sign is that a lot of the deliberation about Wikipedia governance occurs on mailing lists rather than inside the Wiki system. How might we collaboratively author a constitution, without risking fragmentation into several different texts - one for the euroskeptics, one for the federalists, one for the socialists etc.? If the project became popular (and that, after all, was the goal), how could deliberation on such a large scale be achieved? While CCEU was an idea for a campaign, rather than an “application”, it would end up inside the same domain of problems and solutions as the WebTing application.

4.2 The WikiTing

How might we achieve a trans-European online deliberation? My initial ideas for solving this problem owed much to a subset of patterns which initially appeared to resolve more design problems than any others. In retrospect, I have come to call this combination for the “WikiTing”.

WikiTing was never a “complete design”, a blueprint for working software, but rather a model, an abstraction of how such a system might work. When I tried to work out the implications of this model as a user interface design, the design problems I discovered led me to adopt revisions that would eventually depart entirely from the model. The revised model became the WebTing concept. The CCEU model, which will be discussed in the next sub-chapter, represents another path of thinking originating in the WikiTing - but one framed by different needs and realities than WebTing.

Both WebTing and CCEU address other design problems than those posed by the WikiTing model. This merits a closer look at WikiTing; Not just because of its importance in the design process for the final products, but also because it promises some solutions not found in the final products. Further design might resolve WikiTing’s problems without sacrificing these solutions.

4.2.1 The WikiTing Model

WikiTing has three entities: Users, Groups, and Chambers. A **user** is a member of one or more groups - without group membership, a user may only read but not contribute. **Groups** are assumed to consist of users with similar positions or interests, comparable (but not analogous) to political parties, parliamentary committees, or expert commissions.



Fig. 4a: Illustration of the WikiTing model. Users (bottom) contribute by collective authoring of messages in their group’s wiki interface. These messages are passed on to the chamber (top), where they are displayed as a discussion thread.

Individual discussions, between User A and User B, occur only inside groups. Although users might send each other private e-mail, the WikiTing has no features to support publicly visible communication between individual users who are not members of the same group. Publicly visible discussions instead occurs in chambers. A **chamber** is a discussion page, but one where messages are authored by groups, not individuals. If a group wishes to make a statement in a chamber, they must author this statement collectively through a wiki document. When the message expresses the consensus position of the group, it will pass on to the chamber.

4.2.2 Advantages

The WikiTing model promises several advantages over traditional online discussion:

High signal-to-noise ratio: presumably, potential misunderstandings and irrelevant arguments will be filtered out at the group level. Thereby messages posted to the chamber will contain more information in less space than messages posted to an ordinary person-to-person discussion forum.

Scalability: The high signal-to-noise ratio, combined with the time delays imposed by group authorship, would limit the size and speed of discussions - make the WikiTing particularly useful for very large deliberations, potentially involving tens of thousands of active participants. Each participant may both read the public discussion (the chambers) in full and contribute actively through the group mechanism.

Prevention of flaming, trolling and “heat-of-the-moment” authorship: Since flaming and trolling are characteristic behaviours of individuals, the WikiTing might eliminate them or at the very least contain them inside group discussions. By ensuring that a message sent to the chamber has passed through the review of a user’s peers, excessive or ambiguous rhetoric written in “the heat of the moment” may be prevented.

Consensus-building: The group authoring process provides a strong incentive for group members to reach consensus. Without the ability to reach acceptable consensus and compromise, the group cannot function.

4.2.3 Risks

However, there are also some significant risks specific to the WikiTing model:

Polarization: The strong incentive for consensus-building inside groups does not exist on the chamber level, where the positions of groups are pitted against each other. This may lead to a heightened risk of polarization in the discussion.

Slow pace: Reaching consensus inside a group may take significant amounts of time. At the very least, a group’s message cannot be published until group members have had a chance to review it. For some kinds of deliberation, such as deliberation on laws, this is all well and good. But a democratic assembly, even one that nominally deals only with decisions that should take a long time to reach, would still require mechanisms for brief discussions on various cases. This is especially the case with procedural issues, such as whether to open an issue for discussion by the whole assembly.

4.2.4 Unresolved design problems

Finalization mechanism: For a message to be posted to the chamber, group members need to approve it as “final”, as representative of the position of the group. But how is this approval to take place? I considered the idea of using a time constraint: a message is started by a single user, and is open for collaborative editing by all group members for a set period of time (e.g. a week) before it is published. The time constraint, however, distorts power towards the users who perform the final edits of the message. A manipulative user might wait until the last seconds before publication and submit his own version of the message,

Democratic Interfaces

which thereby appears to be the statement of the group but is in reality only the opinion of the last author.

I also considered the idea of using a voting mechanism - either by majority or by consensus: when a message is sufficiently mature, a user proposes publication. When publication is proposed, the message is closed for editing until all votes are cast. If the vote succeeds, the message is published - if it fails, the message is re-opened for editing. A voting mechanism might work, but it introduces additional time delays to what is already a slow-paced process. While voting mechanisms certainly have a role to play in democratic decision-making, a vote on every single message seems too cumbersome a process.

Potential for manipulation: Who may be a member of a group?

If this question is left to the individual user, it allows for a particular kind of manipulation: a user who disagrees strongly with the positions of another group might join that group and sabotage its ability to post messages by initiating edit wars or blocking consensus votes. Even if the user is well-behaved, and joins a dissenting group merely to confront its members directly, it will move significant discussion away from the chamber and into groups, thereby obliterating the scalability advantage.

If, on the other hand, the question of membership is left to the groups, who may invite or expel members, it gives inordinate power to majorities over minorities: a group has an incentive to expel dissenters to ensure a smoother process (and greater visibility in the chamber) - even though the dissent might be legitimate, genuine and conducive to deliberation. In a worst-case scenario groups become the domains of individual participants who enlist allies or clients as “yes-men” in

order to give individually authored messages the rubber stamp of group authorship. This scenario becomes highly likely if a policy of “only one username per person” is not enforced: an individual need only create some sock puppet accounts in order to appear as a “group”.

Conclusion mechanism: While WikiTing encourages decision-making within groups, the model does not provide a solution for chambers to reach final decisions. Conceivably, a conclusion mechanism such as that of WebTing might be added to the WikiTing design without altering the model. This issue was not explored in the design process, and it remains to be seen how various conclusion mechanisms might affect the WikiTing model.

4.3 The CCEU process

The idea for the CCEU process originated with the WikiTing idea, but with the important constraint that it needed to be implemented within a reasonably short time. From the first moment, therefore, it was clear that the CCEU website would need to run off-the-shelf open source software, modified to fit the needs of a wiki deliberation. This technical constraint would become the largest challenge, in terms of the time it consumed, of the entire thesis work: forcing me to think “inside the box” on solutions that might be implemented using standard software, and forcing me to re-think those solutions when attempts at implementing them revealed them to be infeasible.

The “final” product consists of a web site, www.citizensconstitution.org, running MediaWiki software² and the Vanilla web forums³. In the end, no significant modifications could be made to the software, but the default appearance of both has been altered, to form a consistent visual identity. In the case of MediaWiki, the alterations constitute a radical re-design of the software, while in the case of Vanilla, a mere cosmetic change. The project is currently in its starting phase, and it is possible that some of the desired modifications might be added later. Features that are described here as software features might alternately be principles for organising the wiki’s content.

4.3.1 The document finalization process

Fig. 4b shows the progress of a document through the CCEU process. The authors of the document are a group, a subset of the CCEU’s user population. A group might have been defined through a shared political

² The same software that powers Wikipedia. See www.mediawiki.org

³ A light-weight discussion forum software. See www.getvanilla.com

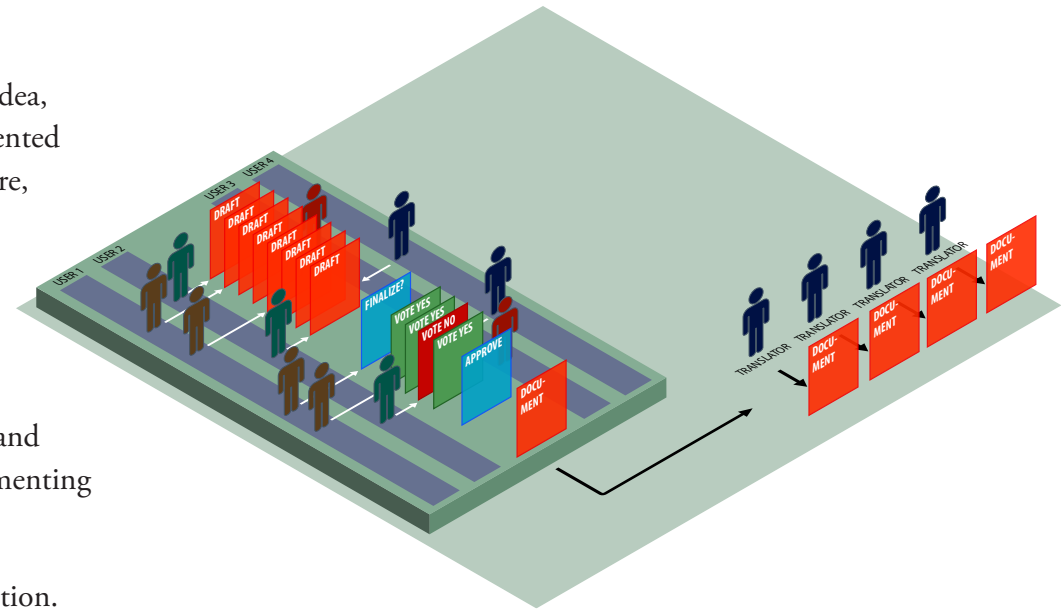


Fig. 4b: *The CCEU process. Illustrated with the example of a document requiring translations.*

position or interest, or (as illustrated) through sharing a common language. The document is produced collaboratively by the group, through normal wiki editing. The standard features of wiki systems - markup, a history page, internal hyperlinks, the possibility of reversion - are available here as well. But the editable document is considered a “draft”. A final “document” is locked, with no further changes possible. The text proceeds from “draft” to “document” when the group, according to its own standards, agree that it is final. After being finalized, the editing process cannot be re-opened, but the contents of the document can later be used as the basis for a new text.

In fig. 4b we see the most obvious advantage of finalizing wiki documents: they can be translated, and translations can refer back to the same document. This is not the case with Wikipedia, where the different language editions frame entirely different user communities. Many articles on Wikipedia exist in multiple languages, but they are not the same article - they are articles about the same topic. How could it be different? Wikipedia articles undergo constant revision, and if a translation is made it will not take long before both the original and the translation have been fundamentally altered. For a site like Wikipedia, this is not necessarily a problem. For a project like CCEU, where deliberations' in many languages needs to be coordinated, it is.

The same process can be used to handle several different kinds of documents: position statements, arguments, drafts of the constitution, fragments of the constitution, drafts of declarations to be made on behalf of the whole community, etc. This process is fundamentally similar to the WikiTing model (above), but more flexible. Since it lives inside a Wiki environment, accompanied by separate threaded discussions, the process needs not solve every conceivable use-prescription but rather supplement the ordinary features of wikis and discussion fora.

4.3.2 A flexible group structure

The idea of “groups” merits some closer examination. Groups in the CCEU process serve several purposes:

- Allowing collaboration in a monolingual forum
- Segmenting deliberation into special interest areas.
- Providing space for like-minded individuals to formulate common positions.

- Identifying experts.
- Administration.

The CCEU process of editing, finalizing and sharing documents is intended as the “bridge” between the groups, allowing the same discussion to take place in several languages. Ideally, groups should be supported by the software: individuals may click a button to apply for group membership. Usernames would appear together with texts or icons indicating the groups a given user was a member of.

The last feature would resolve another issue: identification of experts and politicians. A constitution is not a simple thing to author, and the knowledge and opinions of experts do matter in deliberation. Likewise, the voices of senior politicians - national and European parliamentarians, should be acknowledged as such. Both because they might have a partisan agenda, but also to make other users aware that they are speaking to the target audience: people who have the capability to change the direction of the European union. But how does one indicate that a given user is an expert - without appearing elitist? The group mechanism allows us to set up self-appointed groups, such as “Constitutional Scholars” or “EU parliamentarians”, with membership limited by profession. Since anyone can establish a group, there is nothing elitist about the group system. But since group membership is listed alongside the user’s name, the relative competence and power of members can be easily recognized by other users. This leaves us with the question: what does it mean that someone is a Constitutional Scholar, an EU parliamentarian or a member of the “EU Research Community”? Such questions cannot be answered by the designer or the software, but by the community.

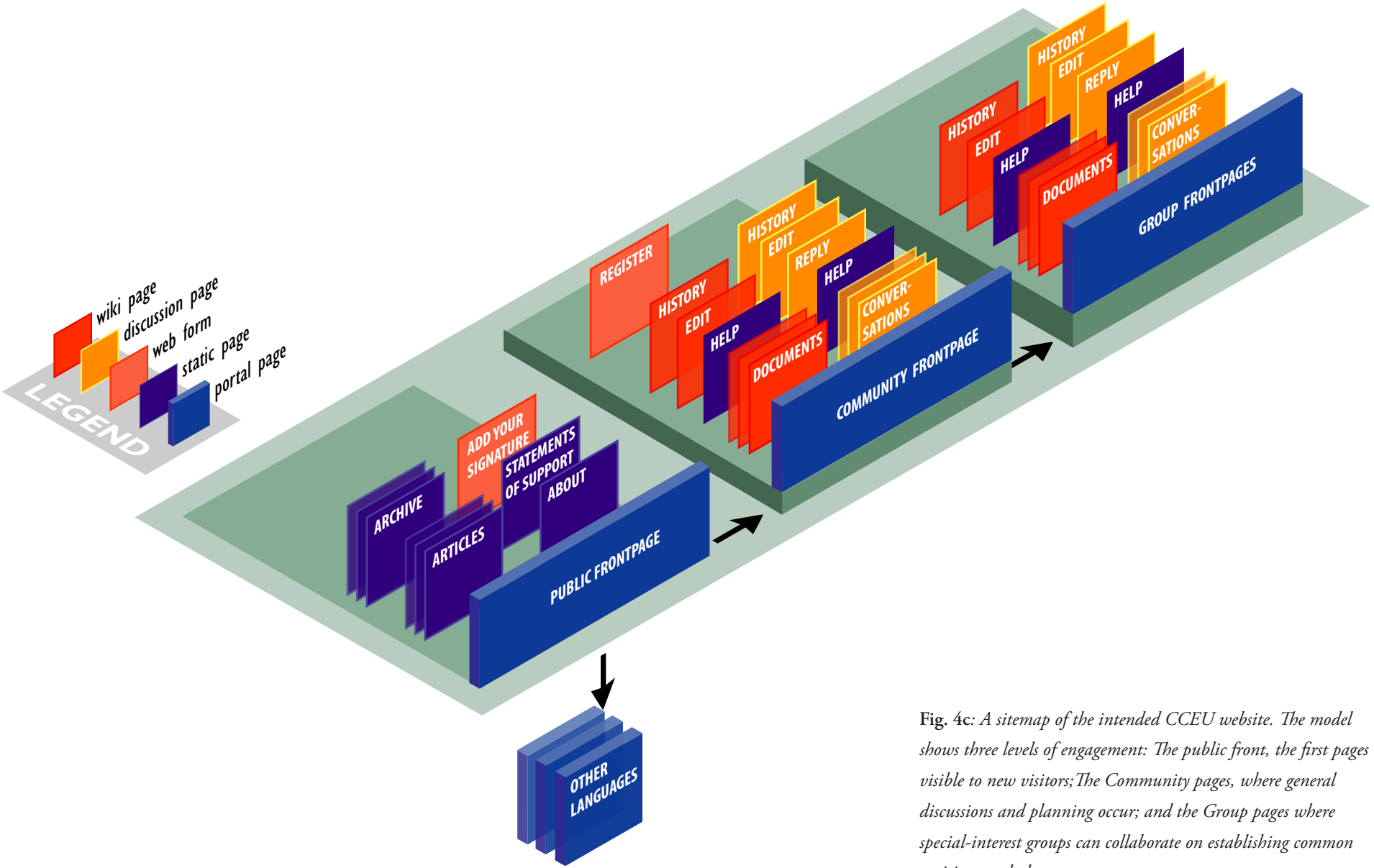


Fig. 4c: A sitemap of the intended CCEU website. The model shows three levels of engagement: The public front, the first pages visible to new visitors; The Community pages, where general discussions and planning occur; and the Group pages where special-interest groups can collaborate on establishing common positions and plans.

4.4 Website design and usability

One of the first steps I took towards the final site was to make simple blueprints, or “wireframes”, of the site’s functionality (fig. 4f). The purpose of the wireframes was to focus exclusively on functionality and usability, separating out visual design questions from the early phases of design.

4.4.1 The Paper Prototype

Based on the wireframes, and in collaboration with other students at Media Lab Helsinki, a paper prototype of the website was constructed. In connection to this, we also interviewed three potential users about their attitude to democracy and to the Web and conducted a simple usability test of the paper prototype with two other potential users as test subjects. Since it was difficult to find interviewees with an interest in/knowledge of EU matters, the CCEU design was re-cast as a campaign site for an issue, a proposed merger between TaiK and two other universities, that was intensely debated at the time.

Since the sample of users was small and the prototype was far removed from the final website, it is difficult to reach general conclusions on the basis of these interviews. One of the clearest conclusions to emerge from the test and interviews was that the notion of “wikis” was not widely known - all our interviewees expressed surprise at the idea that anyone could edit a text - and that a WYSIWYG editor was a necessity; none of the users found the notion of wiki markup easy to comprehend. When discussing online political discourse, several of the interviewees would - without being prompted - raise the issue of online anonymity. Anonymity was bad for online discussions, went the consensus, and they would not attend a site or mailing list where

anonymity or pseudonymity were permitted. From the usability test and interviews, I drew three conclusions for the further design of CCEU:

- WYSIWYG editing was necessary, and the general usability of wikis should be carefully considered.
- The idea of “wikis” and open editing needed to be explained and marketed to potential users.
- Use of real names should be encouraged

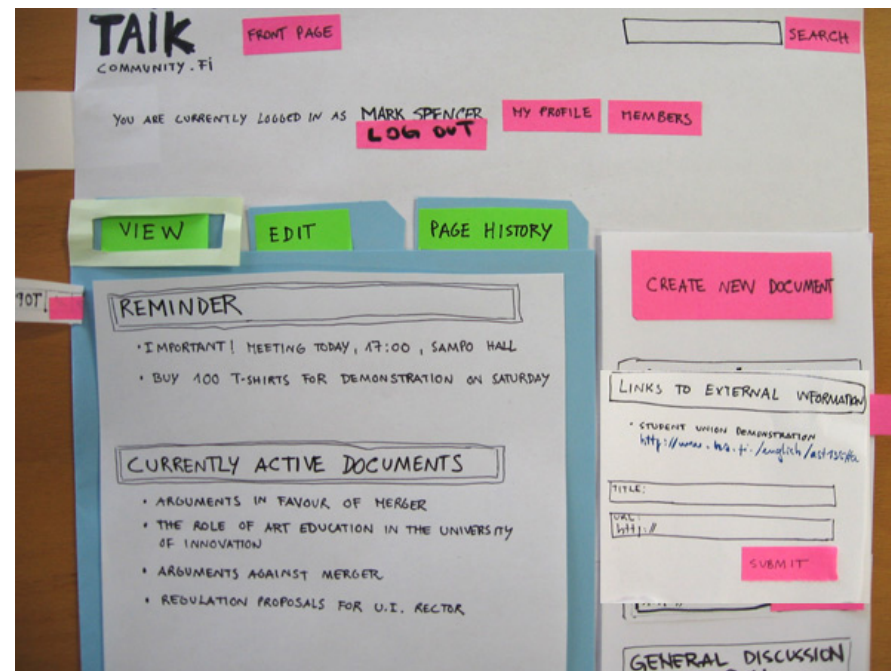


Fig. 4d: A snapshot of the paper prototype.



Fig. 4e: The paper prototype usability test. I am in the bottom-right corner, conducting the interview with the test user (centre). Two of the three other team members are respectively recording the test and simulating website interaction by switching between different layers of the prototype.

4.4.2 Visual design

The visual design of the CCEU website was based on the following criteria:

- Recognition. The visual identity should be unique enough that users could instantly recognize a CCEU page as such.
- “Roughness”. The Website should set the bar for participation as low as possible. It was important to avoid an “over-designed” look.

- Credibility. The site was designed with the research on captology, or “persuasive technology” (Fogg 1998) in mind, and especially according to captological guidelines on website credibility.⁴
- Content-centric. The site was expected to consist primarily of text, and the visual design is intended to emphasize this content, with particular attention to the aesthetics and legibility of text.
- Accessibility. Text sizes etc. were set with screen readers and low-sight readers in mind.

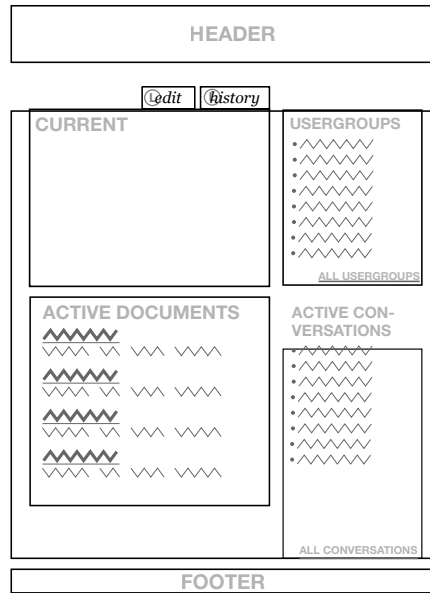
Two iterations of visual design were produced. The first (fig. 4i) appeared both over-designed and too cramped. The second iteration (fig. 4j) tried to resolve these problems by increasing the amount of white space in the design.

Significant attention was paid to re-organizing position and display of links and navigation, so that the context and location of a link or button would provide cues as to its function, and so that lists of navigation and functions would appear less intimidating. The differences between my design and the default Mediawiki design can be seen by comparing fig. 4g, which uses placeholder content from Wikipedia, with the same content in the Wikipedia user interface (fig. 4h).

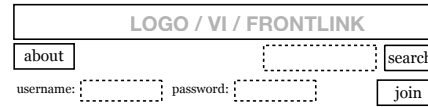
⁴ www.webcredibility.org/guidelines/

Democratic Interfaces

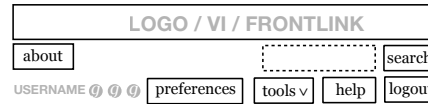
Community Frontpage



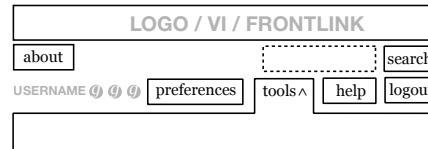
Header - logged out



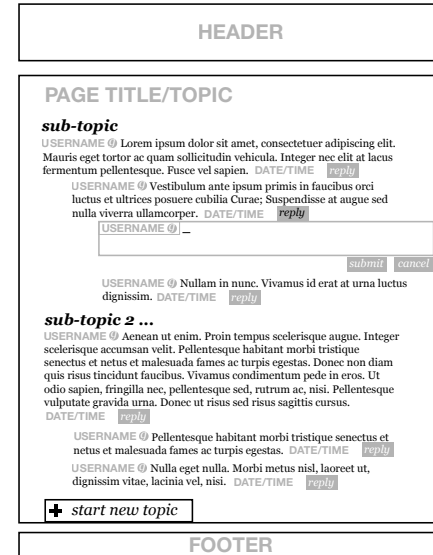
Header - logged in



Header - logged in with tools



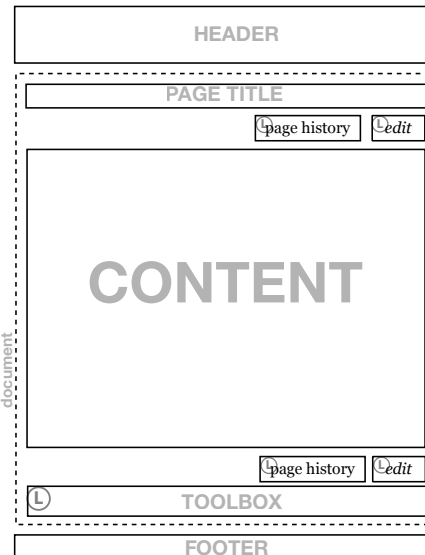
Conversation Page / Reply Interface



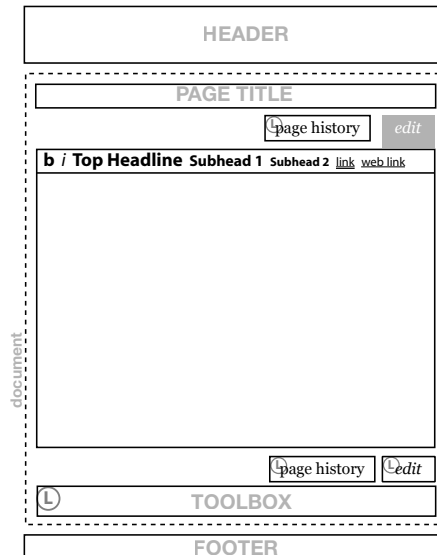
The **reply text-box** appears in the position where the actual reply will appear, with a left margin according to the tree position. Optional vertical scrollbar. (overflow:auto)

Fig. 4f: Wireframe designs for the CCEU website, reflecting the sitemap in fig. 4c.

Document Page



Document Page - edit interface



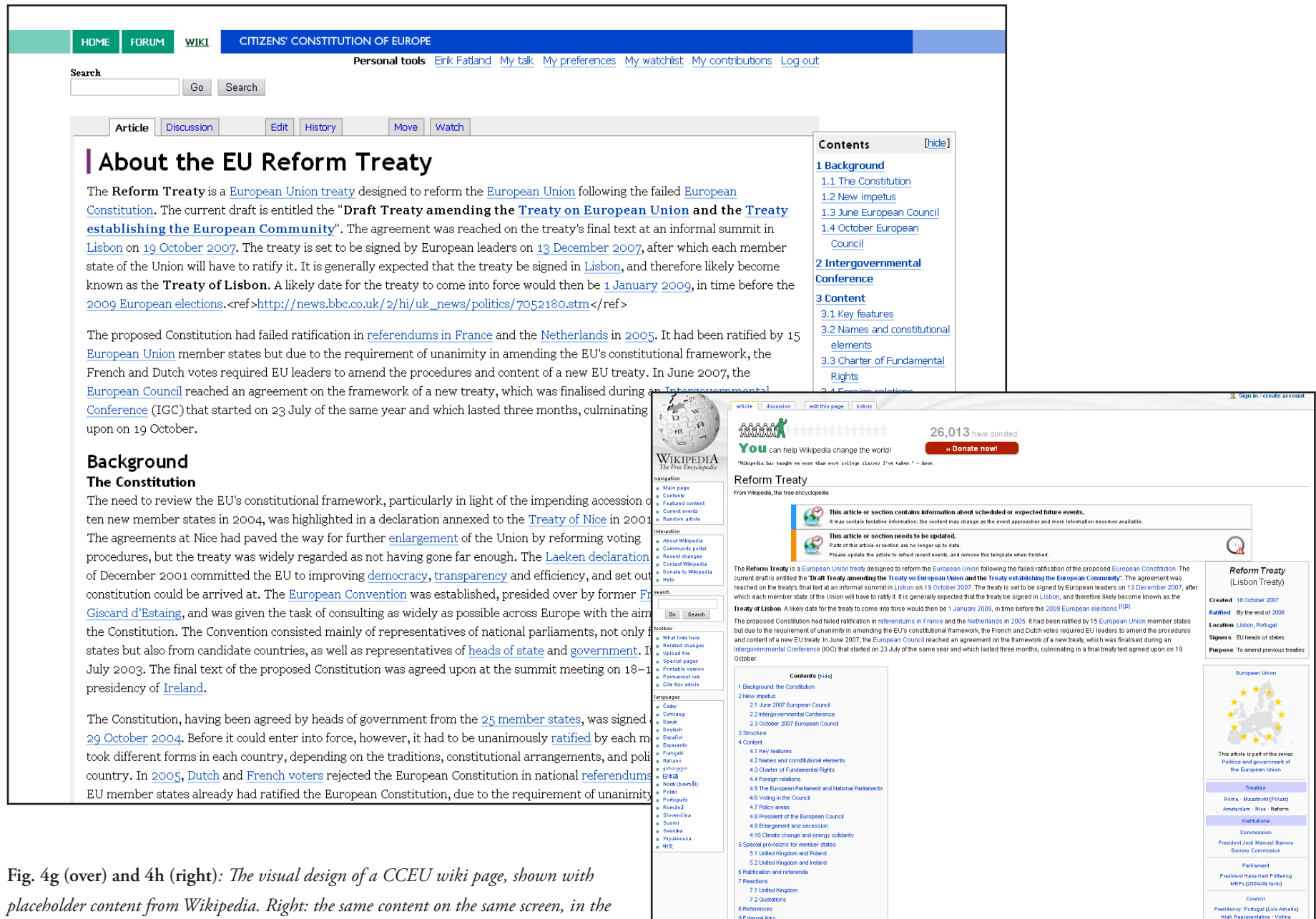


Fig. 4g (over) and 4h (right): The visual design of a CCEU wiki page, shown with placeholder content from Wikipedia. Right: the same content on the same screen, in the Wikipedia user interface.



Fig. 4i (over) and 4j (left): An earlier iteration (over) compared to the final visual design (left).

5 *WebTing*

Summary

The WebTing software design provides a sub-set of the pattern language discussed in chapter 4, elaborated as wireframes (or “blueprints”) and written specifications. These blue-prints simultaneously describe a specific application (the “WebTing software”), and attempt to illustrate, through example, how a general class of “e-deliberation software” might look.

WebTing encodes a three-phase process for each topic or issue: exploration, discussion and decision. Each of these phases has its own user interface, thereby affording certain types of discourse while constraining others. Exploration is handled through an editable, or wiki-style, interface encouraging users to build consensus about the framework for the discussion. Discussion is handled through a tree-shaped discussion forum, which additionally allows users to enter propositions for the final decision. An issue will pass from one phase to the next dependant on a predetermined threshold being reached. Thresholds might take the form of a vote, a specified time limit, a specified limit of activity, or a combination of these. Thresholds are configurable by the system administrator and anyone to whom such power has been delegated, and allow communities to classify and valorize the decisions reached through a WebTing installation.

5.1 Design process and philosophy

The WebTing software design constitutes a proposed solution to the problems and cases described in chapter 1, as well as problems encountered in the design of the WikiTing and CCEU concepts. Intended to be implemented as an open source web application, to be installed and run on any web server, it is presented here through descriptions and “blueprints” of the user interface. The WebTing application provides a frame for the democratic decision-making of a community - it is designed with virtual communities in mind, but might be used for any small to medium-sized community or organisation. The name “WebTing” refers both to the design itself (“the WebTing software design”) and to any hypothetically installed version of the software in use by a community (“a WebTing”, “the Linux administration WebTing”, “the Liberal Party WebTing” etc.). The latter use is analogous to that of “blog” or “forum” - “Joe Trippi’s Blog” or “The Media Lab Forum”.

In this chapter, I present the complete blueprints of the WebTing design along with the reasoning behind each feature. My general design process and philosophy has been described in the previous chapters, but two particular design principles - design constraints and the “nativist” approach - merit further discussion as they especially relate to the current design.

5.1.1 Constraining conversation

WebTing is built on the idea that the careful application of constraints (in Norman’s sense of the word) will contribute to the overall quality and functionality of the design. In each view I have been asking not

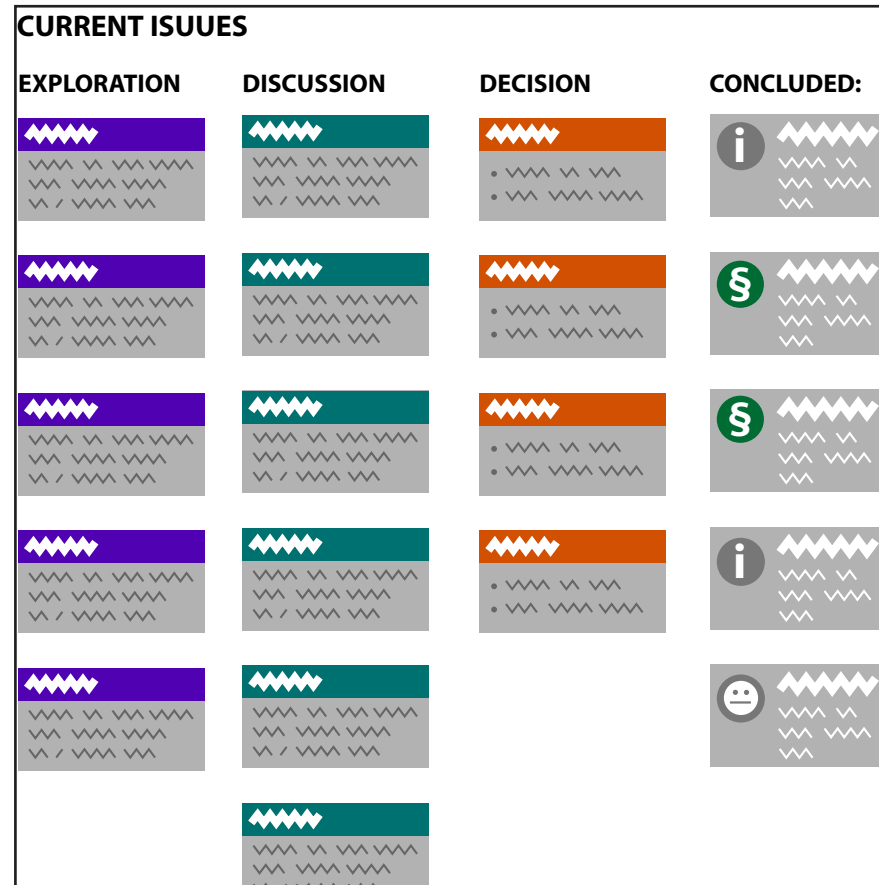


fig. 5a - An early draft of the WebTing front page.

just: “What might the user do here?” but also “what can or should the user **not** be able to do here?”. By sacrificing a modicum of power and flexibility, I hope to have gained clarity, usability and a quality of process. Most importantly, I hope the WebTing user interface will provide its users with a *model* of e-deliberation, a clear starting point for understanding and practicing online democracy.

DEME¹, while it has similar objectives and some of the same features as are proposed for WebTing, demonstrates the problems of the opposite philosophy: the DEME user interface affords, at any given time, an overwhelming amount of options and potential procedures. This makes it difficult to get an overview and a clear understanding of the affordances - and ends up reducing, rather than enhancing, the usability of the system. While DEME can do a great many things, it is unclear what - exactly - it is *meant* to do.

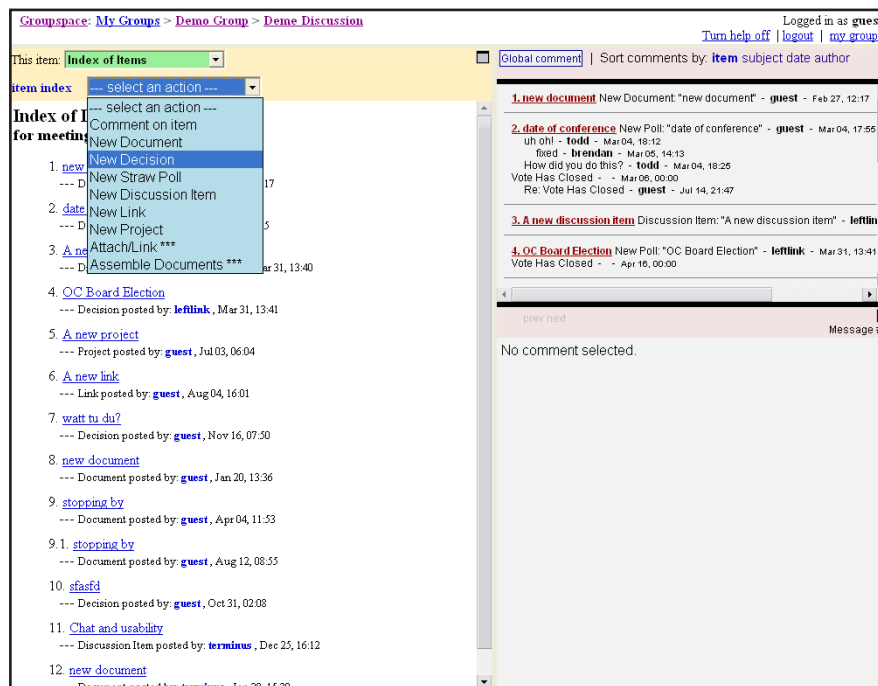


fig. 5b: Screenshot of DEME: a contrasting approach towards similar ends.

5.1.2 The nativist approach

Unlike most other approaches to online deliberation, WebTing is designed for communities native to the Internet, for people who have never and may never meet each other face to face. This does not mean that it could not or should not be used as a decision-making tool for other types of communities or organisations, but it reflects my conviction (argued in chapters 1 and 2) that:

1. Any new design of social software needs to pass the test of online communities before it can hope to reach widespread use.
2. An online democracy cannot and should not mirror the procedures of traditional democratic institutions, but must instead be designed to reflect the fundamental requirements of democracy (consent, legitimacy) in a form that is suited to the unique conditions and human behaviour of the Internet.

1 DEME can be tested at www.groupspace.org

5.2 Voting

Before we can discuss the specifics of the WebTing process, we need to take a closer look at the issue of voting. Votes cast online have traditionally been limited to the function of “straw polls”, consequence-free measurements of what a website’s readers might think about a given issue at a given time. But in recent years, social aggregators - such as reddit.com and especially slashdot.org - have demonstrated that voting can be an effective technique for communal filtering of content and alleviate some of the inherent problems of the online discourse.

Voting was not addressed in the CCEU process or in the WikiTing design. In the design of WikiTing, I intentionally avoided any kind of voting mechanisms in order to focus on the deliberative aspects of democracy, while the CCEU project aims to eventually include face-to-face meetings and real-world democratic assemblies. But for WikiTing, the lack of voting mechanisms led to several problems that I was not able to resolve without re-evaluating the whole design. As a result, WebTing uses voting to reach final decisions on issues and as a component of procedural mechanisms.

A perennial benefit of online voting mechanisms are their positive effect on the motivation of contributors. As a Redditor and Digg user I have experienced this first hand: enthusiastically hitting the “refresh” button to see whether my submission or comment has attracted any more votes. The user community of these sites is shaped by the voting process, as users learn to contribute in ways that are appreciated by the community. There is a genuine risk here of “echo chambers” - of users writing to please, rather than writing to inform. But there is also an incentive for the right kind of compromise: for users to state

their unequivocal opinion, but using language and rhetoric that are acceptable and understandable to other community members.

5.2.1 Fraud, coercion and bribes

To be clear: online voting is problematic, for several reasons.

Computer networks are notoriously difficult to secure, and a successful intruder could change not only vote counts but also erase all tracks of an intrusion. Since it is difficult to detect and even harder prove, the mere suspicion of electronic voting fraud is enough to de-legitimize the democratic process. Voting by computer also casts doubts on another important safeguard of democracy: the voter’s right to vote in secret, a protection against the democratic process being distorted by bribes or coercion.

By designing for the day-to-day decision-making of smaller communities and organisations, we partially sidestep the problems of security and privacy. Bribes, coercion and election fraud in local elections or organisations are uncommon, due to the potential rewards being so low, and far easier to detect in a small community than in national referenda or elections. The same should hold true for virtual communities. And by emphasizing deliberation before decisions are taken, potential fraud or coercion is easier to reveal: if there are significant discrepancies between the opinions expressed by members in discussion and opinions aggregated through votes, the community is likely to suspect fraud. A hacker might easily change vote counts (a simple matter of altering numbers in a database), but it is far more difficult to successfully impersonate a number of users in a discussion, especially without those being impersonated noticing.

If a future version of WebTing is scaled to handle more important decisions for larger groups of people, the issues of security and privacy will need to be resolved. In part, these are not interaction design challenges but challenges of technology: encryption, electronic user identities and the like. If no adequate solution is found, e-deliberation software could be used exclusively for the stages leading up to a vote, while the voting itself is handled the traditional way.

5.2.2 Voting standards

In the world of offline democracy, voting standards are a fairly well established matter: a majority vote (meaning more than 50% of voters) will, in most situations, suffice for a law to pass, an appointment be approved, or similar. We can consider three types of voting standards:

- **majority vote/low standard:** More than 50% of participants must agree.
- **supermajority vote/high standard:** A higher percentage of voters / members than 50% are required, e.g. 2/3 or 70%.
- **consensus vote/highest standard:** All participants must agree, giving every participant a veto right.

The term “participants” in this list is ambiguous - it might refer either to voters (members who actively cast a vote), or to the membership as a whole. I will refer to the former as a “weak” standard, and the latter as a “strong” one. There might be a significant difference between these two standards in cases where only a minority of the membership actively contribute to decision-making.

There are good reasons to avoid strong standards for WebTing thresholds and decisions. Votes requiring a certain percentage of the membership may have difficulty passing in an online situation², since membership in a virtual community is somewhat less committing than serving in a democratic assembly. An empty seat in a parliament is painfully visible, while an inactive member of an online community is entirely invisible. Decisions that require a percentage of members to vote thereby give disproportionate power to non-voters.

But by counting only those who vote, we give disproportionate power to those who participate actively in the process. This provides a strong incentive for democratic participation, which is not a bad thing, but it will also favour those members who have the right combination of ample time, political passion, and computer literacy³ - let us call them the “non-representative majorities”, in that they might represent a majority of activity but not the majority position or the majority of informed opinion. In a WebTing, such non-representative majorities already have significant advantages. Weak consensus and supermajority votes strike a balance between these concerns, giving

² This, we should recall, was the fate of LambdaMoo’s experiment in direct democracy. (Curtis 2002)

³ This effect can presently be observed in the election campaign of US parliamentarian Ron Paul, a libertarian anti-war candidate for the Republican presidential nomination. While opinion polls indicate that Paul has little support in the general population, web readers will easily get the opposite impression due to a very active cadre of tech-savvy Ron Paul supporters. On the other hand, Howard Dean’s campaign for the 2004 Democratic nomination managed to translate initial support from virtual communities into mainstream acceptance and a strong showing in public opinion polls. (Trippi 2004) Such “netroots” campaigns may, in some - but not all - cases, reflect the fact that politically active “early adapters” increasingly turn to the Web for news and organisation, rather than by any significant difference in political preference between active Internet users and the general public.

minorities sufficient power to counteract a majority that might not be representative without hindering active majorities in taking decisions that would have the general support of the community.

When relying on weak voting standards in asynchronous communication the duration of a vote becomes important. When does voting end? When are enough votes cast that we can say for sure that opinion has been sufficiently aggregated? The answer will depend on the context and importance of the vote. For procedural decisions, the consequences are not great, and there are advantages to speeding up the process. Furthermore, if procedural decisions can be easily reversed, a short time-frame but a high standard (consensus or supermajority) might be used - 70% of votes in two days. For final decisions, those that might enact a law, longer time frames are needed, but the guarantee that most members will have a chance to cast a vote lessens our concern about the non-representative majorities, and allows us to reduce the required standard. For example, 60% of votes cast during two weeks might be required to pass important legislation.

While I have argued for weak but high standards in WebTing, the exact level of these standards needs to strike an appropriate balance. On the one hand, WebTing decisions should have the highest possible level of legitimacy. On the other; voting standards cannot be allowed to hinder the community in taking necessary decisions, or hand too much power to minorities. User trials, that test different levels of standards, combined with interviews, to determine the perceived legitimacy of decisions, are needed to resolve this question.

5.2.3 The minimum consensus standard

Not all issues require lengthy deliberation. Some decisions might already - at the onset - have a consensus of the community, or be fairly simple matters to resolve. Democratic institutions use various forms of “fast-track” procedures, eliminating deliberation or reducing voting options to a simple “yes” or “no”, in order to handle these issues.

I will propose a fast-track procedure for online deliberation: the minimum consensus standard. This allows voting to conclude with an affirmative if the first n votes are all in agreement. For example, 10 positive votes (10% of a community of 100) and no negative votes might be enough to take a procedural decision. A higher standard, say 30% of a community, might be needed for final decisions. The minimum consensus standard cannot entirely ignore the aspect of time, or it could easily be abused by a coordinated faction that is large enough to meet the minimum consensus before other members have the time to react. A minimum voting time of 24 hours should be sufficient to eliminate this risk.

5.2.4 Direct democracy, elections and representation

WebTing facilitates direct, rather than representative, democracy, and as such has no features for electing representatives. It does so mainly because it can, because direct democracy is possible online in a way that it is not offline. Not every citizen of a village can or will participate in referenda on all kinds of topics, but asynchronous communication allows for conversations within larger groups of citizens - and furthermore, such communication allows members to be involved to the degree that suits them. Some might discuss and vote on every issue. Some might discuss and vote on issues of

particular interest to them. Some might not participate at all, but trust their peers to take decisions for them. As long as the possibility of participation is guaranteed, the process should retain legitimacy.

We might also think of WebTing as the tool of an elected assembly. In this scenario, representatives would be elected through some other mechanism, but would use a WebTing to facilitate their decision-making. While I have not designed for this particular scenario, it should be a simple question of limiting membership of the Ting only to those elected.

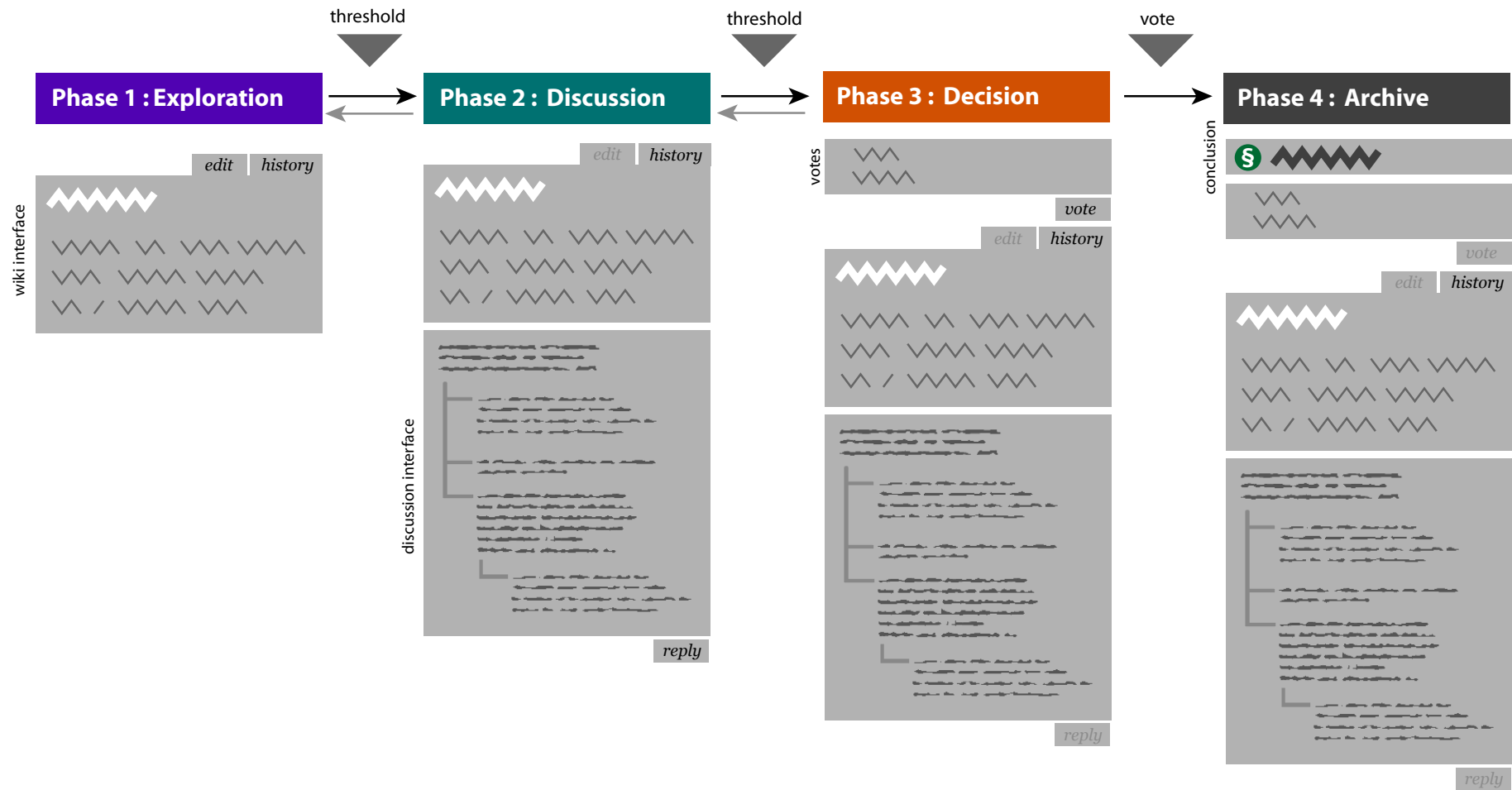


fig. 5c - The transformation of an issue through a WebTing process

5.3 The WebTing process

The requirements (chapter 1.5.2) presented two significant problems: the need for a “conclusion mechanism” and a “decision mechanism”. The WikiTing design failed to address these problems sufficiently inside the wiki paradigm, while discussion-style conversation entails heightened risk of abusive conversation, reducing the potential for deliberation. In the WebTing design, these problems are resolved by separating deliberation into a three-phase process, consisting of an Exploration phase, a Discussion phase and a Decision phase. This process is unique to WebTing, and constitutes the main design innovation presented here.

The phases act to constrain different kinds of conversation at different times, thereby affording and prescribing a kind of conversation that is conducive to deliberative decision-making. Regeczi (2004), summarizing research on group decision support systems (GDSS), argues that decision-making is composed of several distinctly different activities (e.g. idea-generation, negotiation) - requiring different skills of its participants. Support for these different activities cannot be socially constructed through communication software such as e-mail or chat but needs to be embedded, separately for each activity, in the design of the tool used for decision-making. This is precisely the purpose of the WebTing phases.

Separation into phases further helps us communicate the expected, or prescribed, process to users, and visibly separates e-deliberation software from other kinds of software. This should lead to users taking a *tabula rasa* approach when encountering a WebTing : treating it on its own terms and developing appropriate genres, rather than

assuming that the genres of USENET or weblogs or discussion fora are appropriate here as well.

5.3.1 Issues

An **issue** is the main frame of deliberation in the WebTing system, somewhat analogous to “threads” in discussion fora, or the items of a meeting agenda. An issue is a document, a web page. It retains a single, unchangeable URL and a single title. An issue can be opened by any member (unless constitutionally constrained), and initially consists only of a title and an interface without content. The issue title should imply a question - e.g. “Where do we stand on GMOs?” or “Should Julius Cæsar be made dictator for life?” - that will be resolved through deliberation.

The issue title is written by the user who creates the issue. This mechanism admittedly grants the initiator some influence in framing the discussion (“Should glorious Cæsar be recognized as dictator for life?”). But the initiator’s power is balanced by the fact that the same power may be exercised by anyone: other members are able to initiate issues about the same topic with a different frame (“Should we give in to the blackmail of the tyrant Cæsar?”). Many issues about the same topic will reduce the quality of discussion, and decrease the likelihood that any issue reaches a decision. This provides an incentive for issue initiators to phrase the title in neutral language, in order to deter competition and attract sufficient interest that a decision might actually be taken.⁴

⁴ The problem of manipulative titles is not unique to e-deliberation: How would the “USA Patriot Act” have fared if it was instead called the “Destruction of Civil Liberties Act”? If the incentives described above work as intended, these problems might be smaller in a WebTing than in offline democratic assemblies.

5.3.2 The three phases

As an issue moves through the process, content is added by users, discussions occur and votes are cast. Although all content is retained throughout, each phase constrains the ways by which members can create content by providing a unique interface, only available in that phase:

Exploration phase: A wiki-style interface is used to collaboratively gather information relevant to the issue.

Discussion phase: Users propose possible outcomes of the issue (“propositions”), and discuss them in a tree-shaped discussion format. No further edits can be made to the Exploration phase text.

Decision phase: All discussion is closed, but the propositions from the discussion phase are gathered and users are invited to vote on them.

A special meta-mechanism, the “process column” allows for short messages to be exchanged at any time, and for users to alter the progress of an issue by proposing and voting on “motions”.

5.3.3 Thresholds

A “threshold” can be thought of as the mechanism that moves an issue from one phase to the next, or as the separation between phases. The name invokes the metaphor of a doorway - once we pass the threshold, we are in another “room” of the issue, one that presumably invites a different kind of activity. They are also analogous to some of the practices found in real-world deliberative bodies, such as set limits on the length of discussion. While thresholds might involve voting, they are not concluding decisions, but matters of procedure.

WebTing should support two types of thresholds: time thresholds and vote thresholds.⁵ A **time threshold** will be crossed after a given quantity of time has passed, for example two days or two weeks, with a minimum of 24 hours.⁶ A **vote threshold**, on the other hand, requires active member input for an issue to pass from one phase to the next: once a supermajority of affirmative votes are received, the issue moves to the next phase. The vote threshold needs to additionally be constrained by a time limit, so that the threshold is not crossed until both the sufficient amount of time has passed and the sufficient amount of votes are cast.

Users can change their cast votes at any given time. If a threshold fails to accumulate enough votes by the expiration of the time limit, the vote remains open until enough users have changed their votes. Vote thresholds might additionally support the minimum consensus standard, described in 5.2.3.

Several other kinds of thresholds might be considered - e.g. simple majority votes, or consensus votes - but based our discussion of voting in 5.2, these two seem sufficient. Threshold configuration - that is: how and by whom are thresholds are determined - will be further discussed in section 5.2.5

⁵ I originally thought of thresholds as being either “automatic” or “user-controlled”. But I have not found good examples of automatic thresholds that do not depend on time, nor good examples of user control that does not include some form of voting.

⁶ I considered the idea of a time threshold based on dates rather than time limits - e.g. issues will pass thresholds every Monday or every 14th of the Month. But if several issues are open, this would lead to bottlenecks of activity immediately before the threshold date.

5.3.4 Motions

The “process column”, at the right side of the issue page, is exempt from the normal flow of the process, it is instead an interface for discussing and modifying the process. The process column allows users to discuss procedural issues using short, to-the-point messages - “We need more time for discussion” or “this issue isn’t attracting much attention. Should we close it?”. Users can act on such opinions by proposing and voting on “motions”, a formal proposal for some change or exception to be made in the process.

Vote thresholds allow the following motions:

- Close this issue.
- Return this issue to the previous phase.
- Move this issue to decision phase.

Time tresholds allow these, and two additional motions:

- Move this issue immediately to the next phase.
- Extend this phase for N time. (where N is the same as the duration of the time threshold)

In the list presented to the end user, rather than reading “next phase”, “previous phase” etc. the text should refer to the name of the phase - e.g. “Return this issue to exploration phase”.

Motions should meet the same standard of voting as the vote treshold, but the “minimum consensus standard” is especially applicable for motions. A user might regret having opened an issue, and propose a motion to close it. Or the issue might be a simple one, and users might want to move straight to decision phase. Such cases can be handled quickly through a minimum consensus vote.

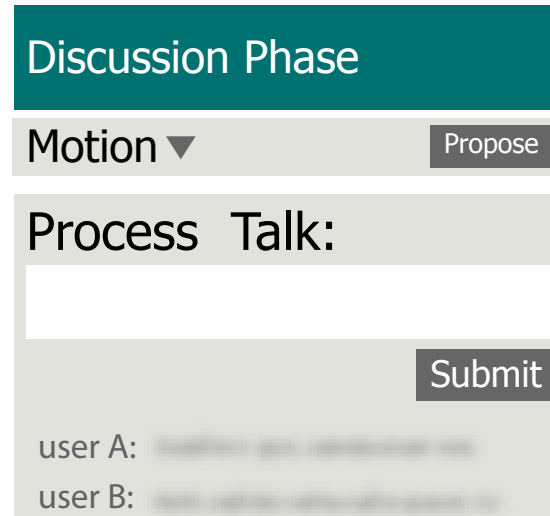


Fig.5d - The “process column”. The current phase is indicated at top, followed by the motion interface, and a chat-style interface for discussion of the process.

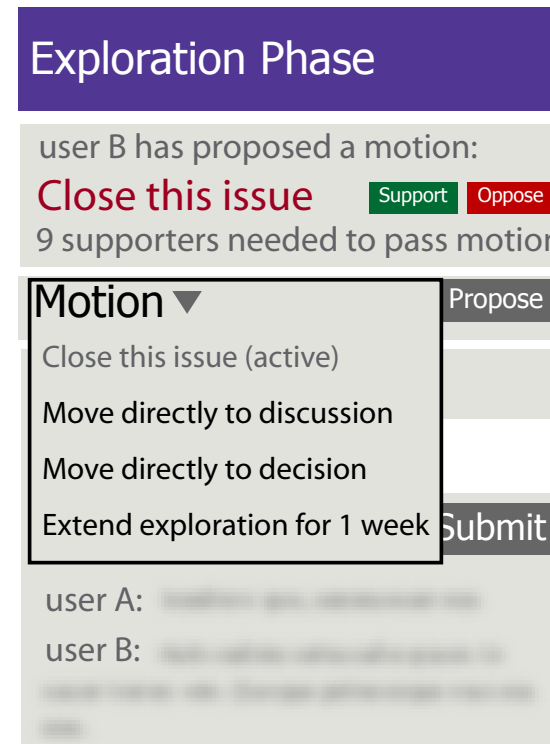


Fig.5e - The process column showing an active motion, and the drop-down list of new motions.

5.3.5 Configuration and constitution

A number of the parameters discussed - such as voting standards or the duration of time thresholds - cannot be standardized, but must instead be adopted to the needs of different communities and to serve different kinds of issues.

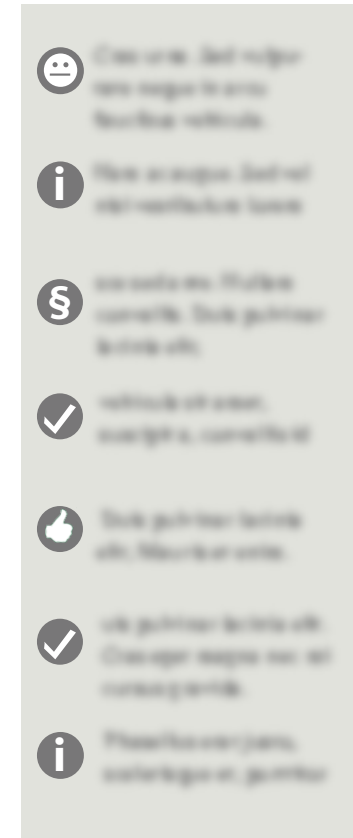
This problem is also a solution to another question : how can the process be authored? How may users change the “constitution” of their Ting? This might be left entirely to the site owner, but I will suggest here a tentative “constitutional mechanism” for WebTing.

The constitutional mechanism has two aspects:

a) Issues may be differentiated into different classes, with different voting standards, with each class having a different meaning for the community. One class might be a “law”, a binding decision. Another class might be a “statement”, a shared opinion that is not a decision. The law might require a 70% majority, and have time thresholds of one month per phase. A statement might require a plurality, and have time thresholds of three days per phase. When a user opens a new issue in the WebTing, the first choice is what kind of issue to open - a proposal for a law? A proposal for a common statement?

b) Definition or re-definition of classes might be accomplished through the same process as other issues. In other words: a “constitutional issue” will be a pre-defined class of issue, installed with WebTing. The constitutional issue allows users to formulate propositions about voting standards, durations and other variables. The default voting standard for a constitutional issue should be set high - at 80% for example - but this might of course be altered through the same process.

Fig. 5f: a list of issues with icons representing their class. Failed or inconclusive issues have icons indicating this status.



5.4 The WebTing User Interface

The following pages contain blueprints for the most important, and frequently used, views of the WebTing user interface. Less common views, such as the issue archive or administration page, are not included - since the main ideas of the design can be illustrated and tested through these “primary” views.

These “blueprints” are neither wireframes, nor exact equivalents of industrial blueprints. In theory, wireframes are meant to let designers and developers consider the basics of a user interface without getting distracted by, or spending time on, visual design issues. The wireframes I produced for CCEU, though, took a considerable amount of time - and while they were *exact* - they did not communicate *clearly*.

Unlike wireframes, the blueprints presented here use colour and shape in order to communicate what the interface is meant to do. The difference between these blueprints and an actual design can be seen by comparing blueprint 5.4.1 - the front page - to Figure 5f, a mock-up of what the front page might look like as an actual website.

WebTing Front Page

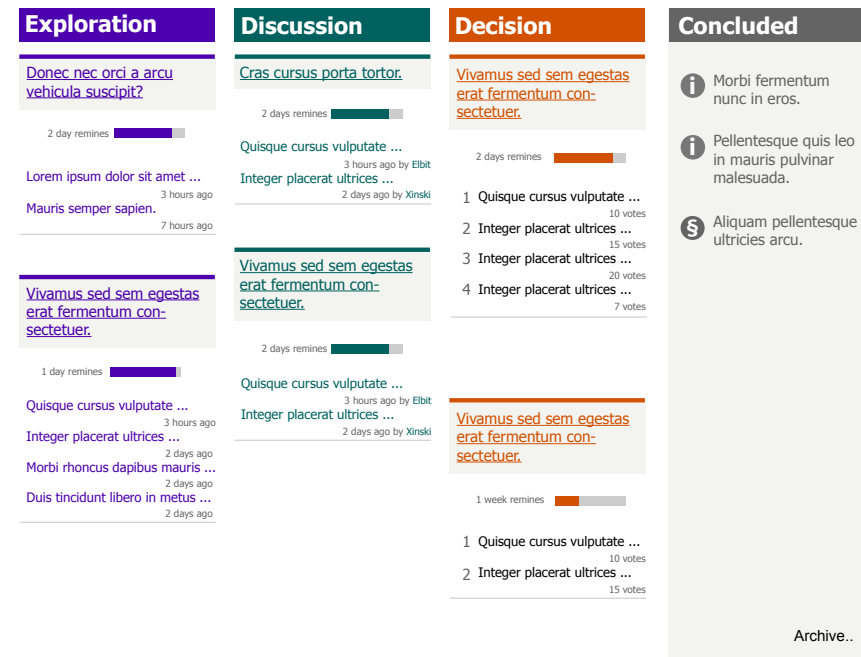
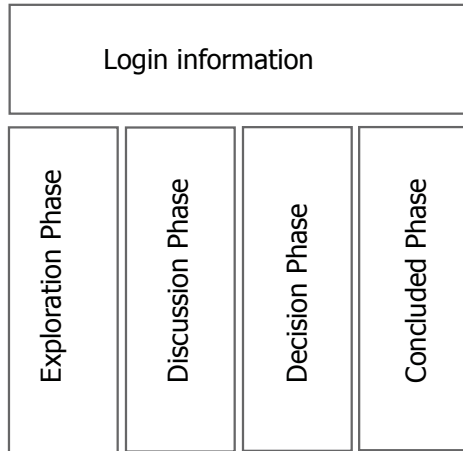


Fig. 5f: what the WebTing design might look like as a web page.

Front Page

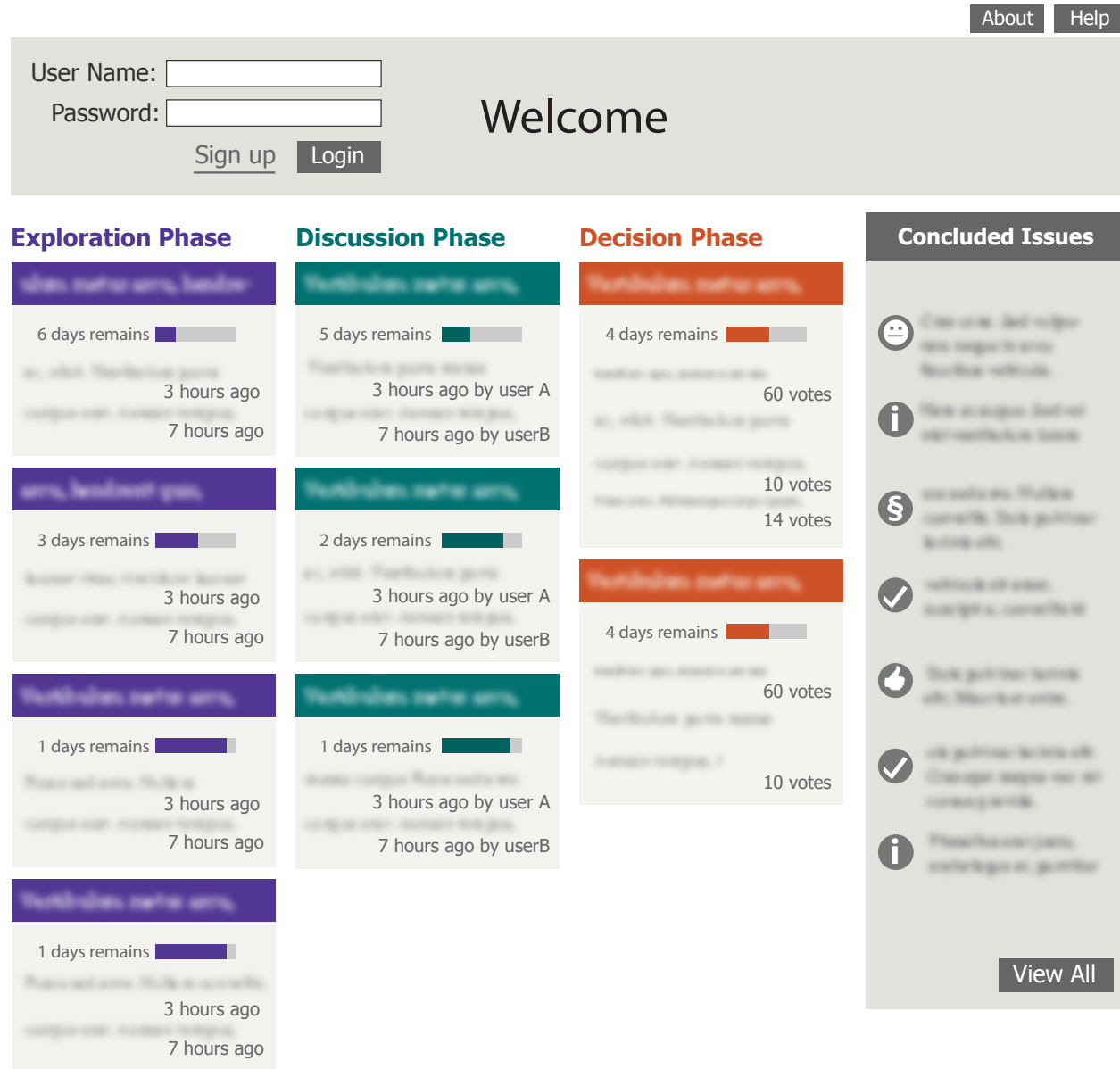


5.4.1 Front Page

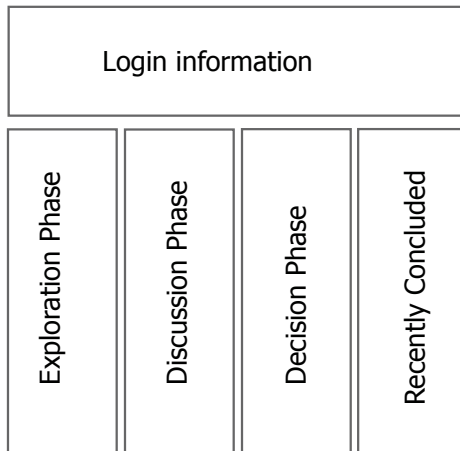
This is the first page to greet any visitor. All currently active issues are displayed, ordered by phase. To the right is a list of recently concluded issues, with a link to the archive.

Each issue is displayed with a link to the issue page, information about recent activity, and distance from threshold.

When an issue is opened, or changes phase, it is listed at the top of the column of the appropriate phase.



Front page



5.4.2 My Front Page

This is the front page visible to users who have logged in. Issues that a user is personally active in are highlighted.

This simple mechanism allows users to keep a spotlight on issues of particular concern. In a large and active WebTing, it also serves to filter information.

Logout About Help

You are logged in as: User A

★ 22 updates on your issues

Exploration Phase

- 6 days remains
- 3 hours ago
- 7 hours ago

Discussion Phase

- 5 days remains
- 3 hours ago by user A
- 7 hours ago by userB

Decision Phase

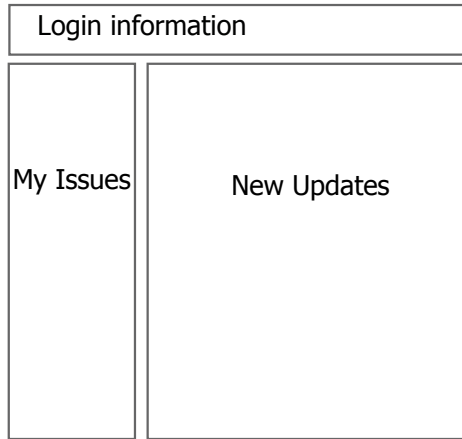
- 4 days remains
- 60 votes
- 6 votes
- 10 votes
- 14 votes

Concluded Issues

- This was not voted on by anyone. No votes.
- This was not voted on by anyone. No votes.
- This was not voted on by anyone. No votes.
- This was not voted on by anyone. No votes.
- This was not voted on by anyone. No votes.
- This was not voted on by anyone. No votes.
- This was not voted on by anyone. No votes.

View All

My Page



5.4.3 My Page

“My page” has two functions: it is the personal page of a user, but it is also the page visible to other users who click on a username.

In this view, only the user’s selected issues are displayed, alongside a list of recent updates of those issues. Contributions by the user are highlighted.

The dual purpose makes this page both a filter, and an identity. By constraining the possibilities of self-mediation to contributions, it is expected that users will express identity and status through constructive contributions.

Logout
Home
About
Help

You are logged in as: User A

My Issues:

Issue title

6 days remains

3 hours ago

7 hours ago

Issue title

5 days remains

3 hours ago by user A

7 hours ago by userB

Issue title

4 days remains

60 votes

6 votes

10 votes

Issue title

1 days remains

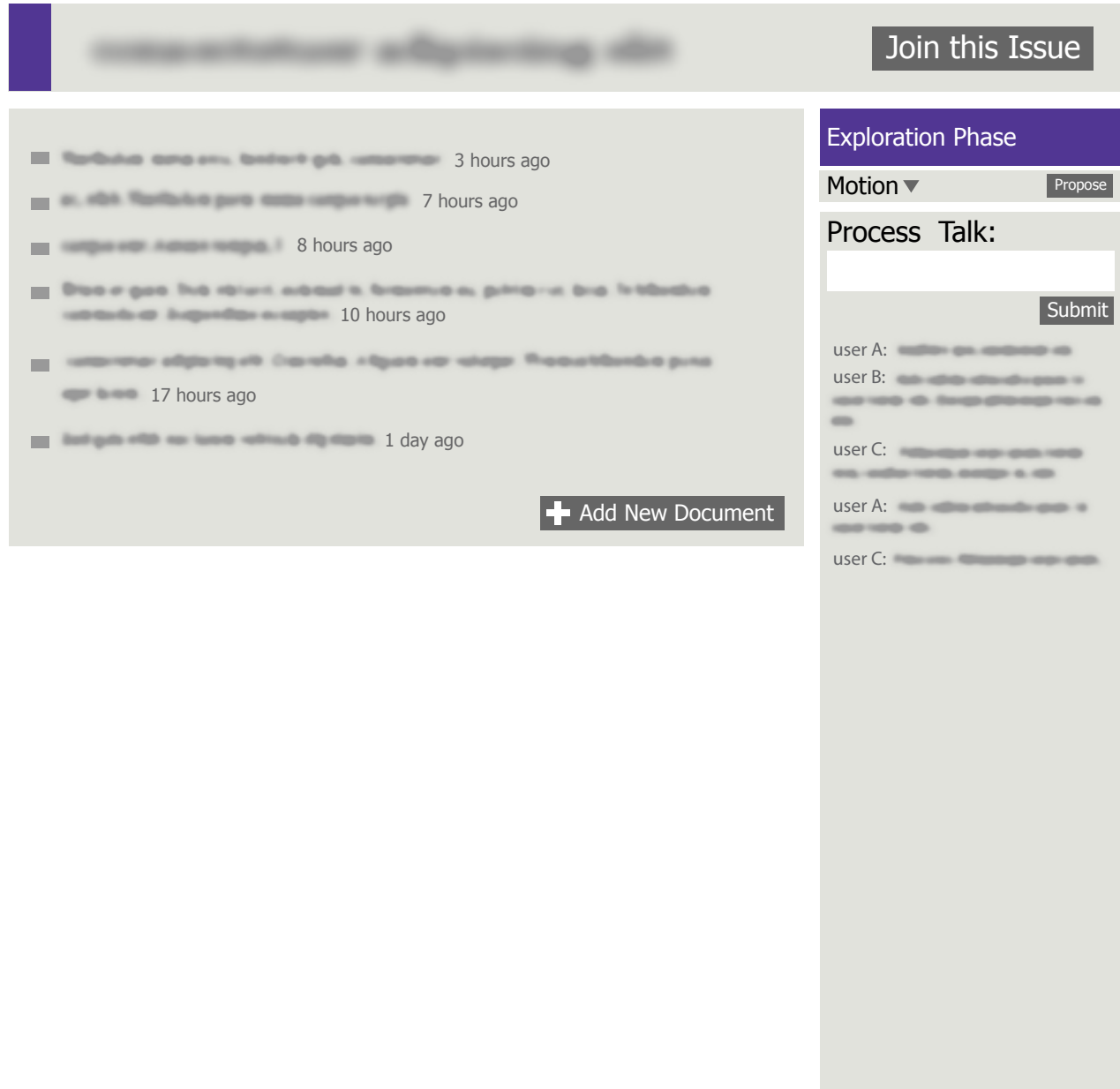
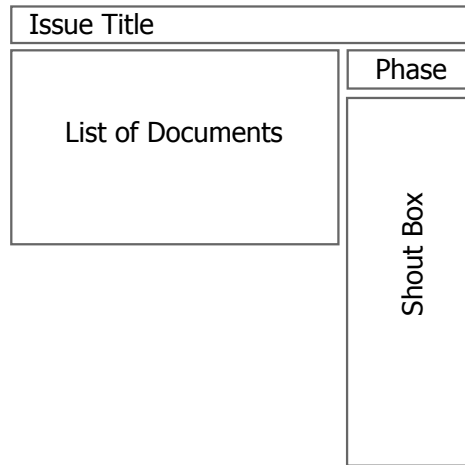
3 hours ago

7 hours ago

New updates:

- Issue title 5 mins ago
- Issue title 1 hour ago
- Issue title 1 hour ago
- ★ Issue title 2 hours ago
- Issue title 2 hours ago
- ★ Issue title 3 hours ago
- Issue title 7 hours ago
- ★ Issue title 8 hours ago
- Issue title 10 hours ago
- ★ Issue title 17 hours ago
- Issue title 1 day ago
- Issue title 1 day ago
- Issue title 1 day ago

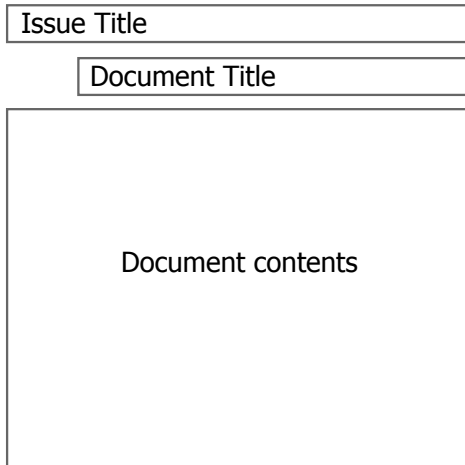
Exploration Phase



5.4.4 Issue in Exploration Phase

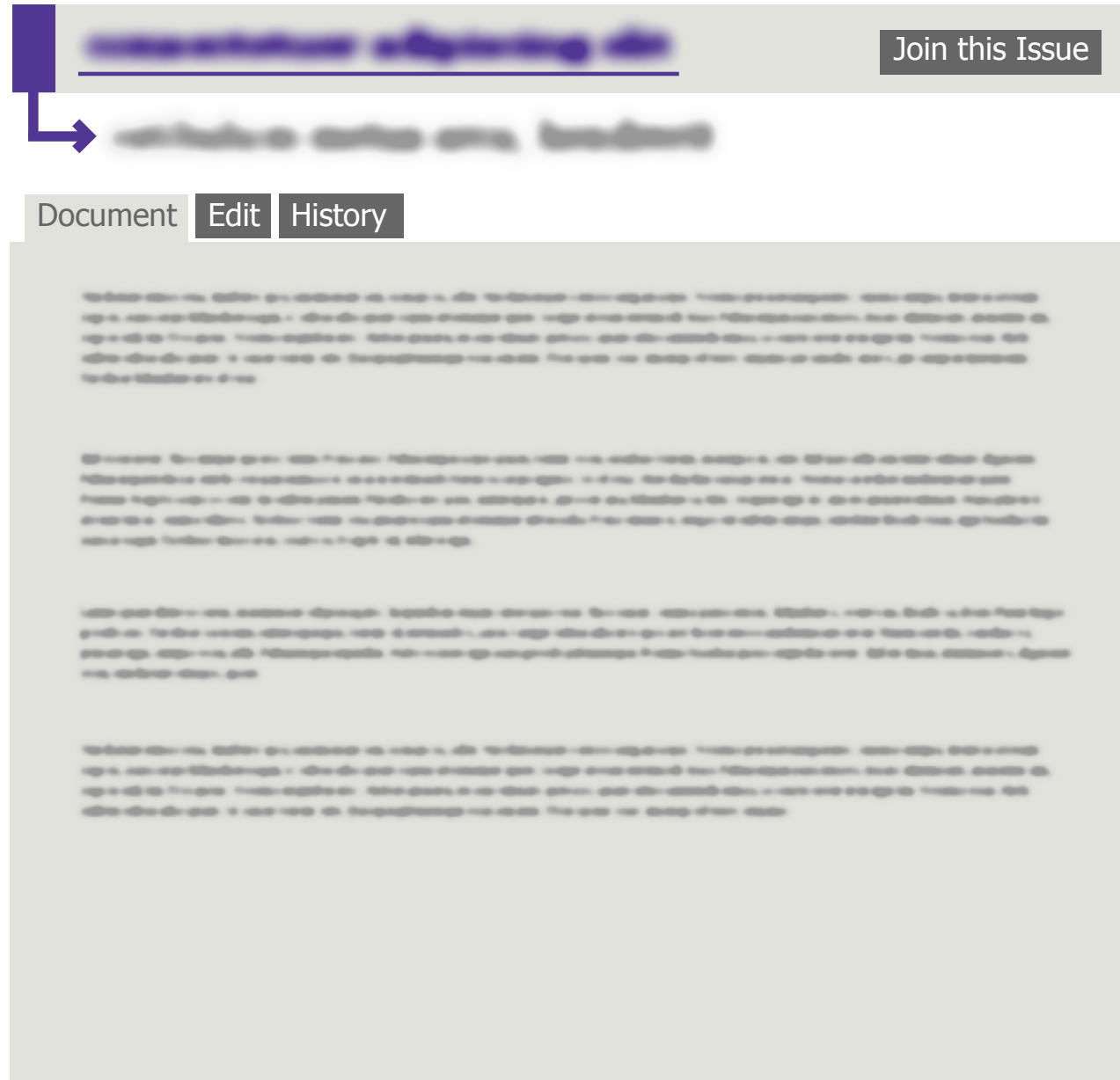
This is the initial state, of an issue. At this point, the interface affords the creation of new documents and the navigation of existing documents. In the tabula rasa, only the button “Add new document” is visible.

Exploration Phase

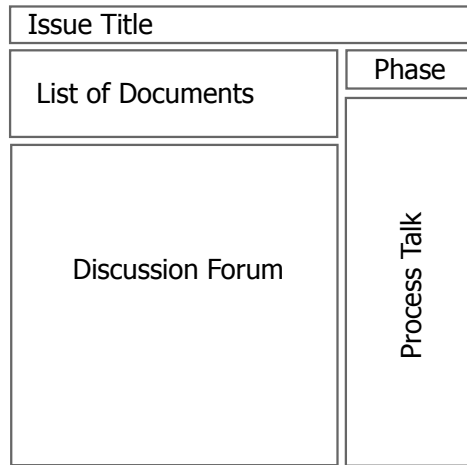


5.4.5 Exploration document

An exploration document is a wiki document. It has the same basic features as wiki systems - an edit button, and a history of revisions, though the creation of new documents may only be accomplished from the issue page. In terms of navigation and information architecture, each exploration document is treated as a sub-page of the issue.



Discussion Phase



5.4.6 Issue in discussion phase

The list of exploration documents is displayed at the top. But documents can no longer be edited or created.

The active interface in this view is the tree-shaped discussion. Each discussion begins not with an opinion or comment, but with a proposition - an suggestion for a resolution. Users may indicate their support for one (and only one) proposition. Propositions with zero supporters, i.e. not even their initiator, are moved to the bottom of the list and excluded from the decision phase. In tabula rasa, the exploration documents and “Suggest New Proposition” button are highlighted.

Join this Issue

2 supporters

10 mins ago by user C Reply

2 days ago by user B Reply

1 days ago by user A Reply

4 supporters

1 days ago by user B Reply

+ Suggest New Proposition

Discussion Phase

Motion ▾ Propose

Process Talk:

Submit

user A: comment on resolution...

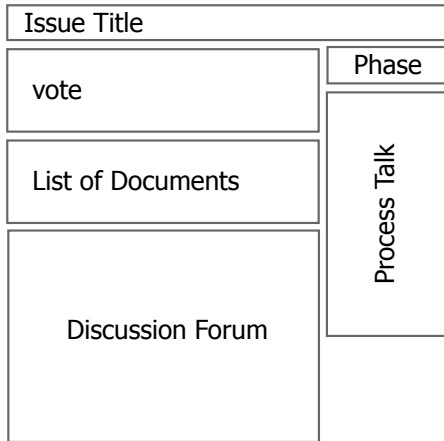
user B: comment on resolution...

user C: comment on resolution...

user A: comment on resolution...

user C: comment on resolution...

Decision Phase



Issue Title
Join this Issue

Need **15 votes** to pass

- 4 votes
- 10 votes
- 3 votes
- 4 votes Abstain

Decision Phase

Motion ▼ Propose

Process Talk:

Submit

user A: ...

user B: ...

user C: ...

user A: ...

user C: ...

2 supporters

...

4 supporters

...

5.4.7 Issue in decision phase

Exploration and discussion are concluded, and the propositions are moved to the top. Users can now vote on the propositions. Votes can be changed as long as the threshold is not reached.

Support for a proposition in the discussion phase is not counted as a vote in the decision phase. Users are asked to review the alternatives, re-think their position, and vote.

Conclusions

Reflections

This thesis work began with a goal, that of designing tools to enable strong online democracy. With the WebTing and CCEU designs, that goal has at least partially been achieved. What remains is to test the designs:

- To see if the CCEU project can attract and maintain a user community, and if that user community will be capable of deliberating on the future of Europe
- To build a working prototype of WebTing, and expose it to the trial-by-fire of being put to use by an actual community.

There was no simple “recipe” or common process for the kind of design challenge undertaken. A large (and undocumented) part of my work has consisted of a broad search for methods, information, prior works - anything that might cast some light on how to design for online democracy. Several of the threads that were investigated are omitted from the thesis text, as they ended up having no discernable impact on the actual design. These included social psychology (especially studies of group decision-making and coercion), captology - the study of persuasion in digital media, and a large collection of papers on e-democracy, e-government and deliberative democracy.

Design methods for community software

In terms of methodology, the most persistent problem encountered in this design exploration was that of designing interfaces for user-to-user communication. I have outlined some ideas and perspectives that proved useful: forecasting user responses to the *tabula rasa*, predicting and establishing genres, treating interfaces as regulating forces. But the larger questions remain: How does one “sketch” a community? How might one “prototype” conversations that last for weeks or months?

How can one “test” the consequences of design on groups of humans over time? Sketching, prototyping and testing are relatively simple tasks when designing for the single user interacting with a single interface. Not so with multi-user interaction.

Interaction design is often taught and talked of as a *reactive* practice, where the designer is served sets of problems by managers, clients, or team members, and is expected to react to those challenges within the fairly narrow borders set by technology, specifications and the marketplace. I suggest two alternative, and complementary, approaches to interaction design, both of which have been demonstrated in this thesis work:

In general, I propose a *proactive* conception of interaction design, one where designers work to identify design problems where interaction design might yield the greatest benefit. Such cases are characterised by design problems that are non-obvious: problems and solutions that require more than daily experience with computers to identify.

For the specific task of designing community software, I also propose a *prescriptive* form of Interaction Design, where the important questions are not “what are the user’s needs?” or “what are users doing?” but “what might users, without knowing it, need?” and “what should users be doing?”

Will they come?

But the question I have had the greatest difficulty answering is a simple one: “if we build it - will they come?” Is there really an unarticulated need for democratic decision-making tools? The cases presented in the first chapter indicate - but do not prove - that such a need exists.

As I was preparing this thesis for print, I received the clearest confirmation yet: an invitation to join a new group on Facebook (a social networking site) called “Constitutional Facebook”. The founders of the group, fed up by Facebook’s perceived violations of the privacy and IP rights of their users, propose writing a Facebook constitution - a virtual bill of rights for users, with or without the approval of Facebook’s owners. The group, to date, has more than 600 members and is growing rapidly.

The first question raised in the group’s discussion forum was this:

“How should this group be governed?”

The first replies, in summary:

“As a democracy.”

Sources

All URLs, unless otherwise indicated, were referenced on November 16th, 2007

A

Adamic, LA and Glance, N 2005, 'The political blogosphere and the 2004 U.S. election: divided they blog' in *Proceedings of the 3rd international Workshop on Link Discovery*, Chicago, Illinois, [online], <http://doi.acm.org/10.1145/1134271.1134277>

Agre, P 1998, 'Designing Genres for New Media: Social, Economic, and Political Contexts', [online], <http://polaris.gseis.ucla.edu/pagre/genre.html>

B

Bellamy, R and Castiglione, D 2000, 'The uses of democracy : Reflections on the European democratic deficit', in Eriksen & Fossum 2000

Borchers, JO 2000, 'A pattern approach to interaction design', in *Proceedings of the 3rd Conference on Designing interactive Systems: Processes, Practices, Methods, and Techniques*, [online], <http://doi.acm.org/10.1145/347642.347795>

Buchanan, JM and Tullock, G 1962, *The Calculus of Consent: Logical Foundations of Constitutional Democracy*, University of Michigan Press, Ann Arbor

Buxton, W 2003, 'Performance by Design: The Role of Design in Software Product Development', [online], <http://www.billbuxton.com/SoftwareDesign.pdf>

C

Castells, M 2000, *The Rise of the Network Society*, Second Edition, Blackwell Publishers Ltd, Oxford

Cavalier, R 2005, 'Project PICOLA (Public Informed Citizen Online Assembly)', [online], <http://www.online-deliberation.net/conf2005/viewabstract.php?id=13>

Clift, S 2000, *E-Democracy E-Book: Democracy is Online 2.0*, [online], <http://www.publicus.net/ebook/>

Curtis, P 2002, 'The Incredible Tale of LambdaMOO', [online], http://www.g4tv.com/techtv/vault/features/38666/The_Incredible_Tale_of_LambdaMOO.html

Cross, N 2006, *Designerly Ways of Knowing*, Springer-Verlag London Ltd.

D

Dahlberg, L 2001, 'Extending the Public Sphere through Cyberspace: The Case of Minnesota E-Democracy', in *First Monday : peer-reviewed journal of the Internet*, volume 6, number 3, [online], http://firstmonday.org/issues/issue6_3/dahlberg/index.html

Debian.org 2006, 'The Debian Constitution', [online], <http://www.debian.org/devel/constitution>

E

Elster, J (ed) 1998, *Deliberative Democracy*, Cambridge University Press, Cambridge, UK

Eriksen, EO & Fossum, JE (eds) 2000, *Democracy in the European Union : Integration Through Deliberation?* Routledge, New York

EU Commission 2003, *eGovernment Resource Book : Synopses of IST projects relating to eGovernment*, [online], http://ec.europa.eu/information_society/activities/egovernment_research/doc/pdf/2003_1231_en.pdf

F

Fishkin, JS 2000, 'Virtual Democratic Possibilities: Prospects for Internet Democracy', [online], http://cdd.stanford.edu/research/papers/2000/brazil_paper.pdf

Fogg, BJ 1998 'Persuasive Computers: Perspectives and Research Directions', *Proceedings of CHI 98*

- Freedom House 2006, 'Freedom in the World 2006', [online], <http://www.freedomhouse.org/uploads/pdf/Charts2006.pdf>
- Fukuyama, F 1989, 'The End of History?', [online], <http://www.unc.edu/home/rlstev/Text/Fukuyama%20End%20of%20History.pdf>
- G**
- Gibson, James J 1986, *The Ecological Approach to Visual Perception*, Lawrence Erlbaum Associates, Hillsdale, New Jersey
- H**
- Habermas, J 1996, *Between Facts and Norms*, Polity Press, Cambridge, Massachusetts
- Habermas, J 2001, 'Why Europe Needs a Constitution', [online], <http://www.newleftreview.org/A2343>
- Habermas, J 2005 'The illusionary "Leftist No"', [online], <http://www.signandsight.com/features/163.html>
- Hertzum, M and Jacobsen, NE 2001, 'The Evaluator Effect: A Chilling Fact about Usability Evaluation Methods', [online], <http://www.idemployee.id.tue.nl/g.w.m.rauterberg/lecturenotes/JFS-lecture/hertzum-jacobsen-2003.pdf>
- I**
- IETF 2006, RFC 4677 : 'The Tao of IETF: A Novice's Guide to the Internet Engineering Task Force', [online], <http://www.ietf.org/rfc/rfc4677.txt>
- Iyengar, S Luskin, RC and Fishkin, JS 2003, 'Facilitating Informed Public Opinion: Evidence from Face-to-face and Online Deliberative Polls', [online], <http://cdd.stanford.edu/research/papers/2003/facilitating.pdf>
- J**
- Jacobsen, NE Hertzum, M and John, BE 1998, 'The Evaluator Effect in Usability Tests', in *CHI 98 Conference Summary on Human Factors in Computing Systems*, ACM, New York, [online], <http://doi.acm.org/10.1145/286498.286737>
- Jones, JC 2002, *Design Methods*, second edition, John Wiley & Sons, New York
- K**
- Kelly, J Fisher, D and Smith, M 2005, 'Debate, Division, and Diversity: Political Discourse Networks in USENET newsgroups', Online Deliberation Conference 2005, [online], http://www.coi.columbia.edu/pdf/kelly_fisher_smith_ddd.pdf
- Kurvinen, E 2007, *Prototyping Social Action*, Publication Series of the University of Art & Design Helsinki, Helsinki
- Kollock, P and Smith, MA (eds.) 1999, *Communities in Cyberspace*, Routledge, New York
- Krebs, V 2003, 'Divided We Stand?', [online], <http://www.orgnet.com/leftright.html>
- L**
- Latour, B 1987, *Science in Action*, Harvard University Press, Cambridge, Massachusetts
- Law, ELC and Hvannberg, ET 2004, 'Analysis of Combinatorial User Effect in International Usability Tests', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04. ACM, New York, [online], <http://doi.acm.org/10.1145/985692.985694>
- Lawson, B 2004, *What Designers Know*, Architectural Press, Oxford
- Leonard, A 2004, 'Trapped in the echo chamber', [online], http://dir.salon.com/story/tech/col/leon/2004/11/03/echo_chamber/index.html

Lessig, L 1997, 'The Laws of Cyberspace', [online], http://www.lessig.org/content/articles/works/laws_cyberspace.pdf

Lessig, L 2006, *Code version 2.0*, Basic Books, Cambridge, Massachusets

M

Manovich, L 2002, 'Generation Flash', [online], http://www.manovich.net/DOCS/generation_flash.doc

N

Norman, DA 1994, *Things That Make Us Smart*, Addison Wesley Publishing Company, Boston

Norman, DA 2002, *The Design of Everyday Things*, Basic Books, New York

P

Portland Pattern Repository (2006), 'Welcome visitors', [online], <http://www.c2.com/cgi/wiki?WelcomeVisitors>

Popper, K 1963, 'Prediction and Prophecy in Social Sciences', in *Conjectures and Refutations. The Growth of Scientific Knowledge*, Routledge, London

R

Regeczi, DJ 2004, 'Technological Reinforcements: The Illusion of IT and Strong Democracy', Politics and Information Systems: Technologies and Applications conference, Orlando, Florida, [online], <http://purl.org/utwente/49378>

Reid, E 1999, 'Hierarchy and power: social control in Cyberspace', in Kollock & Smith 1999

Rushkoff, D 2003, *Open Source Democracy: How online communication is changing offline politics*, [online], <http://www.demos.co.uk/publications/opensource democracy2>

S

Shirky, C 2004, 'Group as User: Flaming and the Design of Social Software', [online], http://www.shirky.com/writings/group_user.htm

Smith, ADV 1999, 'Problems of conflict management in virtual communities', in Kollock & Smith 1999

Sunstein, CR 2001, *Designing Democracy - What Constitutions Do*, Oxford University Press

T

Trippi, J 2004, *The Revolution Will Not Be Televised*, HarperCollins Books, New York

Tufte, ER 1997, *Visual Explanations: Images and Quantities, Evidence and Narrative*, Graphics Press, Cheshire

W

Welch, M 2006, 'Farewell to warblogging', [online], <http://www.reason.com/0604/co.mw.farewell.shtml>

Wenger, E 2001, *Supporting communities of practice : a survey of community-oriented technologies*, [online], <http://www.ewenger.com/tech/>

Winn, T and Calder, P 2003, 'A pattern language for pattern language structure', In *Proceedings of the 2002 Conference on Pattern Languages of Programs - Volume 13*, [online], <http://portal.acm.org/citation.cfm?id=1151669.1151673>

Witschge, T 2002, 'Online Deliberation: Possibilities of the Internet for Deliberative Democracy', in Peter Shane (ed.) *Democracy Online: The Prospects for Political Renewal Through the Internet*, Routledge, New York, [online], http://java.cs.vt.edu/public/projects/digitalgov/papers/onlinedeliberation_Nl.pdf

Appendix: a pattern language

I Problems

The architecture of WebTing needs to address the problems of online discourse in general, and specifically the challenges posed by online deliberation. The former are well-known and well-documented, the latter are not. The following list of problems forms the basis of the pattern language – for each problem, at least one pattern offering a solution is offered.

Invisible disputes

Editable texts are subject to many different kinds of editing, some are uncontroversial improvements in grammar and spelling – others reflect the divergent opinions of users. Many edits border the two – new and improved language may encode a different and controversial bias. In a text that undergoes frequent edits, it is difficult to identify the controversial segments, and so to deal with controversy when it arises.

Edit wars

When irreconcilable disputes arise in editable texts, the disputing users may enter what is known as an “edit war” - overturning each others edits. Edit wars are time-consuming - the “winner” is the user who has the most time and patience to monitor and edit the text. They add nothing of significance to deliberation.

Flame wars

A “Flame War” in Internet jargon is “an acrimonies dispute” in a discussion forum. The practice of flaming is counter-productive to online deliberation, as it tends to lock conversation into larger and larger insults rather than factual discussion.

Heat of the moment

Clicking a “send” button is a far easier task than making a public statement to a group of people. Flame wars are often started by inconsiderate users writing and posting provocative, inflammatory messages in “the heat of the moment” - and by those so provoked replying while inflamed.

Massive conversations

In deliberations with large amounts of users, the amount of messages and opinions may easily reach the level where no single user can get an overview of the discussion.

Communication overload

When large amounts of messages are posted in a discussion, the discussion itself becomes difficult to overview and the voice of individual users disappear. Communication overload may occur even in small groups of users, as users with enough time keep posting to the conversation faster than users with little time can follow. Communication overload in turn lead some users to grab attention with excessive rhetoric – inflaming the discussion.

Voiceless individual

In online deliberation, unlike in real-world deliberation, the individual that does not post is in effect invisible and easily ignored by the active members of the community. This is a problem for deliberative democracy, where all stake-holders should be consulted.

Endless discourse

An asynchronous deliberation needs some point where a decision is taken, and the normative document is considered “final”. Wiki’s and discussion fora currently do not provide this functionality. For example, Wikipedia articles undergo continuous re-writes, with the content and quality of the articles fluctuating.

Lack of history

Even in discussion fora that preserve past messages, users easily ignore or forget older messages in the discussion. This can lead to a negative cycle where the same arguments are presented repeatedly, making the conversation less informed.

2 Design Patterns

Features that already exist are marked with an (a). Proposed features are marked with a (p).

Editable texts (a)

Context: Collaboration on a document between geographically separate users.

Problem: With applications such as e-mail, it is difficult to keep track of changes in a document.

Solution: Allow any user to make alterations to the text, but retain a log of changes and allow users review the log and roll back (revert) unwanted changes. Wiki's are the currently best known applications that utilize editable texts.

Propositions (p)

Context: WebTing environment.

Problem: A deliberation platform needs a way for users to author normative documents.

Solution: Allow users to post "propositions", editable texts that may end up as normative documents.

See next: Resolutions

Resolutions (p)

Context: Propositions.

Problem: After a proposition has passed, further edits are not required.

Solution: Propositions that have passed the decision-making phase are closed for further edits and comments. Approved propositions are stored as "resolutions", whereas rejected propositions are stored as "failed propositions" - each one in its separate archival section.

Deliberation phases (p)

Context: WebTing environment.

Problem: If there are many propositions in a WebTing, gaining quick overview of and navigating between them will be time-consuming for users.

Solution: Sort propositions into "phases" according to far advanced they are towards the decision-making stage.

See next: Draft phase, Consideration phase.

Draft phase (p)

Context: Deliberation phases.

Problem: Users need a way to post ideas in a non-binding manner.

Solution: Let initial propositions appear as "drafts".

Consideration phase (p)

Context: Deliberation phases.

Problem: A need to identify which propositions are being seriously considered for a vote.

Solution: When a draft proposition is sufficiently mature, it may enter the consideration phase - signifying that the proposition may be put up for a vote, and that it is important for users to pay attention to it. A draft may move to the consideration phase by an action of the first

author, by a moderator, or by an automated rule - for example, a draft that has been edited by more than 10 different users enters the consideration phase.

Time-limited texts (p)

Context: Editable text.

Problem: Endless discourse.

Solution: Set a pre-determined time limit for each new document. After the time limit passes, the document may automatically pass to decision-making.

Informed Discussion

Discussion forum (a)

Context: Online conversations.

Problem: Users need tools to participate in, and get an overview of, many-to-many conversations.

Solution: Associate each message with the user identity of its author (name, e-mail address or handle) together with relevant data such as subject of discussion and time of posting.

E-mail lists are the simplest form of discussion forum – distributing messages sent by any user to the entire list of messages.

Most of the more sophisticated discussion fora are organized either “tree shaped” or “star shaped”. “Tree-shaped” discussions display message subjects in relation to each other, showing the replies to a message on a level below the original message. The “thread” of subjects and relations is displayed in full on an index page, while a

context-specific section of the thread can be displayed together with each message.

“Star shaped”, or “flat”, discussions show messages in linear succession as replies to a single originating message. A group of such messages is called a “thread” or “topic”, and an index page provides a list of topics.

Examples: Usenet, Mailman, UBBforum.

Comment system (a)

Context: Online conversations.

Problem: Not all feedback to a document will take the form of edits or votes.

Solution: Let the document be accompanied by a comment system, a simple single-thread discussion forum.

Message quarantine (p)

Context: Discussion.

Problem: Heat of the moment.

Solution: New messages enter “quarantine” for a set period of time (hours or days). While a message is in quarantine, it is only visible to the user who wrote it – who may re-read it, re-think its content, and decide to edit or delete it before publication.

Collaborative discussion (p)

Context: User Factions, Discussion

Problem: Information overload, Voiceless individual, Flame wars

Solution: Set up a discussion where each message in the discussion is authored collaboratively by a faction, rather than by individual users. Each message is quarantined, either until a time limit passes or until a sufficient amount of users vote to publish the message. Until publication, the message is an editable text only visible to faction members.

See also: Message quarantine, Editable text, Discussion forum.

Time-limited discussions (p)

Context: Discussion forum.

Problem: Endless discourse.

Solution: Set a pre-determined time limit for each discussion. After the time-limit passes, the discussion is frozen.

See also: Time-limited texts.

Size-limited discussions (p)

Context: Discussion forum.

Problem: Communication overload

Solution: Set strict limits to the allowable lengths of discussion entries, and to the number of discussion entries a user may post. Thereby - users are forced to ration their words for efficiency.

Iconic reply (a)

Context: Discussion forum.

Problem: Invisible user.

Solution: Allow users to reply to messages or propositions with an icon indicating their level of agreement or disagreement - such as a smiley face for approval or thumbs down for

disapproval. This allows any user to express an opinion without large expenditures of time, and functions as a quick poll indicating the popularity of the proposal.

See also: Straw poll

Hyperlinking (a)

Context: WebTing environment.

Problem: Users need a way to easily reference information relevant to the deliberation.

Solution: Allow users to easily add hyperlinks in their propositions or discussions.

Fact archive (p)

Context: WebTing environment, Hyperlinking.

Problem: Lack of history. Not all information is available on the Internet.

Solution: Allow users to write or upload texts to an “archive of facts” for easy reference. Each fact article may be either an editable text, wikipedia-style, or have an attached discussion thread, allowing other users to comment on the source.

Argument archive (p)

Context: WebTing environment, Fact archive.

Problem: Lack of history.

Solution: Let users save arguments that are of a general nature, and so may be applied to future discussions, in an “Argument archive”. The merits of an article may be discussed in a thread connected to the argument archive.

Powers and decisions**Feature assignment by role (a)**

Context: WebTing environment

Problem: Depending on the use, it may not be desirable to let all users access all functions.

Solution: Assign users different roles, with different levels of power in the system. Role assignment may be automated, or democratic - with users electing other users to specific roles.

Administrator role (a)

Context: Feature assignment by role

Problem: Someone must have access to the technological back-end.

Solution: The “someone” who has this access is an administrator. To enhance transparency, the administrator should be publicly known. To enhance democracy, administrators might be elected.

Moderator role (a)

Context: Feature assignment by role

Problem: Flame-wars, heat of the moment

Solution: A “moderator” may be appointed or elected with powers to delete messages that violate community standards. The actions of the moderator role need to be transparent.

Speaker role (p)

Context: Feature assignment by role

Problem: Massive deliberations.

Solution: There may be situations where users of one WebTing community need to express themselves in another, and larger, community. This may be done through the election of a speaker - who gains the right to participate in the larger deliberation, on behalf of the smaller community.

Facilitator role (p)

Context: Feature assignment by role, time limitations

Problem: Invisible individual.

Solution: The members of a WebTing may elect a facilitator, whose job it is to make sure that all voices are heard. In addition to the social role, the facilitator would have the power to override the closing of debates due to time limitations so that more users might participate.

Straw poll (a)

Context: WebTing environment.

Problem: Users need a way to get a quick overview of how popular or unpopular a proposition may be.

Solution: Allow users to post non-binding Straw polls, measuring the general feelings towards a particular proposal. Straw polls can be posted independently, or in connection to a proposition or debate.

Community votes (a)

Context: Decision-making.

Problem: Deliberations require the ability to make final decisions.

Solution: Allow users to bring proposals - for example propositions or role elections - to a vote in the community. Votes can

take many forms, including consensus, absolute majorities, majorities of those who vote, and pluralities.

Identifying and resolving disputes

Automatic identification of disputes (p)

Context: Editable text.

Problem: Invisible dispute.

Solution: Disputes may be identified by a document or section being reverted more than once. The software may be programmed to recognize such disputes and automatically mark the text as being “in dispute”.

Automatic identification may save users time, and reduce the severity of edit wars. However, not all reverts reflect a genuine dispute, for example many Wikipedia reverts are done in order to remove spam or vandalism.

Human identification of disputes (p)

Context: Editable text.

Problem: Invisible dispute.

Solution: Rather than revert a revert of an edit, a user may click a button to signify dispute. The document or section will then be marked as “in dispute”.

Document forks (p)

Context: Dispute identification.

Problem: A dispute has been identified, and the disputing users are unable or unlikely to find a compromise.

Solution: Split the document into two documents, one document for each side of the dispute. A decision, for example in the form of a vote, must later be taken as to which document is normative.

Process freeze (p)

Context: Dispute identification.

Problem: Two or more users are disputing the contents of a document.

Solution: When a dispute is identified, the document or section in dispute is frozen – no new edits can be made, and no decisions can be taken on the document.

The existence of process freeze in a system may additionally prevent flame wars – users will have an incentive to resolve their differences and seek compromise early on, so as not to trigger a dispute.

Unless implemented together with document forks or user unfreeze, this feature gives veto power to any user, and so forces consensus decision-making.

This pattern is reminiscent of the Wikipedia policy of “protecting” articles that fall victim to heated edit wars (WikiPedia 2006d). However, Wikipedia locks are a moderator principle whereas a process freeze is an automated software function.

See next: Community unfreeze, dispute unfreeze

Dispute unfreeze (p)

Context: Process freeze.

Problem: An editable text has been frozen by a dispute.

Solution: The document will be unfrozen when the user(s) who triggered the dispute un-freeze it. This will encourage disputing users to negotiate with each other.

User unfreeze (p)

Context: Process freeze

Problem: An editable text has been frozen by a dispute.

Solution: The document will be unfrozen when a sufficient threshold of users, for example 50% or consensus-minus-one have voted to re-open the document. The community can thus override the vetoes of individual users, or find a compromise through discussion without requiring the consent of the dispute initiators.

3 Example applications

The following scenarios are examples of how patterns can be combined to design applications for specific purposes.

Scenario 1: Contract Negotiator

Concept: a web application for the negotiation of contracts between two parties.

Patterns: Editable text, proposition, resolution, Human identification of disputes, Process freeze, dispute unfreeze.

Description: The contract negotiator is a simple tool for writing contracts. At its core is an editable text (the contract). If the parties enter a revert war over its contents, a dispute is automatically triggered and the contract freezes. Only when both parties agree on a compromise can the text move forwards to become a final contract.

Scenario 2: Advocacy Forum

Concept: A web application for an online advocacy group which needs to coordinate their member's opinions to make effective statements about their special interest.

Patterns: Editable text, proposition, resolution, community vote (consensus), comment system, automatic dispute identification, dispute unfreeze, user unfreeze, argument archive, fact archive.

Description: The "Advocacy Forum" is a tool for groups of fairly like-minded individuals who need to collaboratively author statements such as press releases and position documents. In the advocacy forum, any user can initiate a proposition - other users can edit and comment on

the proposition. An argument archive and a fact archive are primarily used to contain facts and arguments that the community agrees upon - for future reference. If disputes arise, they are detected automatically and the document is frozen. If the disputing users do not resolve their differences, a threshold of 50% of logged-in users over 48 hours can unlock the dispute. Propositions are brought to a consensus-minus-one vote when any user requests it. Rejected propositions can be brought up for new votes later on.

Scenario 3: National organizer

Concept: An applications that coordinates the deliberations of a large organisation with several chapters.

Patterns: Editable text, proposition, resolution, community vote (plurality), comment system, automatic dispute identification, user unfreeze, argument archive, fact archive, moderator role, facilitator role, speaker role, faction discussion.

Description: The National Organizer is a tool for large organisations with several chapters, such as a political party or large NGO. Members can be expected to both agree and to disagree on a number of issues.

The application comes in two different flavours: the Chapter Organizer and the Main Organizer. A chapter organizer is similar to the advocacy forum, but with the added features of a moderator and facilitator (in order to ensure that aggressive members do not bully the "silent majority"). The argument archive will be used to store arguments frequently used by particular factions of users.

The main organizer has similar functionality, and all chapter members are automatically members of the main organizer. However, only

elected speakers from the chapters are permitted to edit propositions in the main organizer, and discussions occur only as faction discussions. This allows the voices of the different chapters to be heard, and all chapters to participate in the deliberation, without flooding the Main Organizer with messages.