Engineering Software Assemblies for Participatory Democracy: The Participatory Budgeting Use Case

James Holston, Valérie Issarny, Cristhian Parra

Engineering Software Assemblies for Participatory Democracy: The Participatory Budgeting Use Case

James Holston
University of California
Berkeley, CA, USA
jholston@berkeley.edu

Valérie Issarny
Inria
Paris, France
Valerie.Issarny@inria.fr

Cristhian Parra
Inria &
University of California
Berkeley, CA, USA
Cristhian.Parra@inria.fr

ABSTRACT

The worldwide use of the Internet and social networking has transformed the constraints of time and space in human interaction: we can now be heard at a massive scale unprecedented in human history. As a result, ICT may enable citizens to undertake both government through direct assembly and collective action at a scale and an efficacy previously considered impossible. Our research concerns this opportunity to leverage a new sort of political life. We focus specifically on how software systems may enable participatory democracy, that is, the participation of citizens in democratic assembly, action, and governance. As an initial step, we have developed a service-oriented software platform, called *AppCivist-PB*, focused on a specific, yet representative use case of participatory democracy, namely, Participatory Budgeting (PB for short). PB is an allocation process used in many cities around the world through which they commit a percentage of their annual budget (often 5%) to implement citizen-proposed projects. In PB, residents of a city (or a higher level territorial organization), brainstorm, develop, and select project proposals that local government institutions are required to fund and implement. The key contribution of *AppCivist-PB* is to enable the cohesive creation of both citizen and software assemblies that together implement a given participatory budgeting campaign.

Keywords

Software composition, Online participatory democracy, Democratic assembly, Participatory budgeting, Collaborative proposal making, Deliberation, Versioning, Voting.

1. INTRODUCTION

Information and communication technologies (ICT) are profoundly changing the nature of human social and environmental interactions. One such change concerns innovations in the way that citizens both interact with government institutions and engage in greater self-government through democratic assembly and collective action. Our research focuses on this transformation of politics, asking how new social media can contribute to new forms of democracy. The pervasive use of ICT suggests that they present an unprecedented opportunity to rethink the constraints of time and space that are generally thought to make the exercise of a more direct and engaging democracy at a large scale practically impossible [10]. In effect, ICT challenge the assumption that citizens of large political units must be content with systems of representative democracy that typically produce a more passive and legalistic citizenship than an active and participatory one.

To consider this challenge, we undertake a pragmatic and modest investigation of how ICT – and more precisely software systems – can contribute to enabling direct democracy at a large scale. Our research has two immediate objectives. One is to engineer software that leverages the reach of the Internet and the powers of computation to enhance the experience and efficacy of civic participation. The second is to use the ICT software platform to induce the associational forms of a new digitally-inspired citizenship among residents. As our focus is on the problem of democratic organization and action in contemporary cities, we are interested in fostering a citizenship among residents regardless of their national immigration status; in effect, an urban citizenship [11]. Thus, a key problem to investigate is whether the digital methods of assembly that the software platform implements can engage residents on the basis of a digitally-inspired urban citizenship, that is, a citizenship facilitated by online communications yet based on local residence. The objective is to engineer new forms of digital democratic assembly that favor the emergence of a citizenship for which participation in the making of the city (understood as the sum of activities of residents) is both the context and the substance of a sense of belonging and for which nation-state membership and national immigration status are irrelevant.

Our research is multi-disciplinary in nature, bringing together anthropologists and computer scientists to co-investigate how to build software systems that promote the development of such digital democratic assemblies and citizens. Our initiative is further rooted in the principles of social activism in that we want to provide citizens with new software systems that help them articulate projects, deliberate directly among themselves, and mobilize activities. A number of digital tools – and in particular social networks and web-based content management systems – already support aspects of social activism [20]. However, these tools need to be customized as much as composed to

Accepted for publication in Proc. ICSE-SEIS'16

become really useful for activists. To that end, we introduced the high-level design of the *AppCivist* service-oriented software platform in [16]. *AppCivist* is built around the vision of letting activist users compose their own applications, called *Assemblies*, using relevant Internet-based components that enable various aspects of democratic assembly and collective action. Starting from a social science perspective [12], we identified the following high-level categories of functions for *AppCivist Assemblies* [16]: Mobilizing people, Co-creating proposals, Acting collectively, and Communicating. We also identified examples of Internet services that support, at least partially, these functions. For instance, we refer to social networking that facilitates reaching out and communicating; Google Apps for Non Profits (`http://ww.google.fr/nonprofits/products`) that eases collaboration; and CrowdVoice (`http://crowdvoice.org/`) that is dedicated to mobilizing activists.

However, the development of *AppCivist* led us to conclude that existing relevant Internet services are all lacking sufficient support for more grassroots-oriented collaboration and online co-creation activities. For example, the common practice in the co-creation of proposals, especially for the ideation stage, relies on face-face meetings. Although a digital platform can and even should encourage offline meetings, it should also make it possible for users to complete full-fledged proposals at the Internet scale without them. Thus, the digital co-creation of proposals should enable online and at a large scale what people accomplish offline and at a small scale. Otherwise the software system does not leverage the powers of the Internet and computation.

Enabling people to gather over the Internet to elaborate ideas jointly as proposals for action is a challenging topic, one that needs to be explored with the user community [3]. [15] identifies this collaboration as the key role of Design Thinking in social software engineering, conceived as designing software systems with, in contrast to for, the users. However, as the target user community for direct participatory democracy is both vast and diverse, it is difficult to claim co-design in software with a representative population of users in advance of a specific application. Instead, *AppCivist* addresses design thinking by allowing end-users to assemble relevant software services and components that have themselves been co-developed with social scientists who have studied democratic processes and social activism in the field [12]. In addition, we have concentrated on developing the first instance of *AppCivist* for Participatory Budgeting (PB), as a representative use case of participatory democracy [7]. As a result, we are able to account for various initiatives in citizen participation, including lessons learned from existing PB campaigns worldwide since their emergence in Brazil in the late 1980s.

Building on the service-oriented principles for the *AppCivist* platform set out in [16], this paper details the development of *AppCivist-PB*, from design to prototype implementation. This development leads us to elaborate on the following key research contributions throughout the paper:

1. *State of the art survey and analysis of software systems that contribute to enabling participatory democracy:* Section 2 describes key functions associated with software systems that support participatory democracy [13]. It then surveys a number of representative web-based systems that allow a given user community to collaborate around proposal and decision making, though these systems are not necessarily designed for participatory democracy. This review concludes that no existing system offers an adequate bottom-up approach to digital proposal making. Such an approach would allow groups of citizens to self-assemble on the basis of common interests and enable the resulting *citizen assemblies* to initiate ideas and elaborate on them using convenient *assemblies of software services*.

2. *State of the art survey and analysis of digital tools oriented towards Participatory Budgeting:* Section 3 focuses on the specific use case of PB that is the main focus of our paper. PB has emerged as a significant field of innovation in democracy and local governance in the last 25 years [8, 4]. After its initial development in Brazil, the practice has spread worldwide with major initiatives in capital cities such as Paris. However, leveraging ICT to enable truly urban-scale participation in PB campaigns remains unrealized. We discuss how *AppCivist-PB* utilizes the concepts of citizen assembly and software assembly to address this challenge.

3. *AppCivist-PB software architecture enabling citizen and software assemblies:* Section 4 introduces the overall software architecture of *AppCivist-PB*. Following the design of *AppCivist* introduced in [16], *AppCivist-PB* strictly adheres to the architectural principles of service orientation. In that framework, citizen assemblies allow registered users and groups of users to self assemble into higher-level groups to coordinate idea generation and to elaborate proposals through versioning. In a complementary way, software assemblies adhere to the well-known principle of service composition, configuring software services and components oriented towards the implementation of functions supporting participatory democracy.

4. *AppCivist-PB prototype:* Section 5 complements the above with the description of the first *AppCivist-PB* prototype. This prototype permits an assessment of the effectiveness of *AppCivist-PB* in supporting actual urban-scale PB campaigns, such as the one of Paris in 2015. In addition, the prototype provides an opportunity to experiment with developing service wrappers to integrate third-party services into its software assemblies. In the near future, we intend to automate this integration as much as possible, building on our background in the synthesis of mediators [6].

Finally, Section 6 concludes with a summary of our contribution compared to related work and of our future initiatives.

2. SOFTWARE SYSTEMS MEET PARTICIPATORY DEMOCRACY

As depicted in Figure 1 from [23], participatory democracy may be positioned with respect to other models of political democracy in a two-dimensional analytical space, according to the primary means (from direct to representative) and goals (from opinion formation to decision making) of democracy. Participatory democracy is a combination of direct and

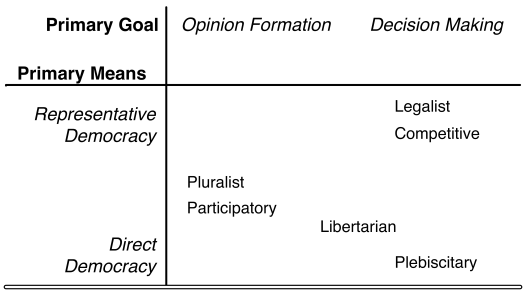

Figure 1: Models of political democracy from [23]

representative democracy, emphasizing both opinion formation – where the support of citizenship is central – and decision making – where the plebiscitary process of deliberation and voting either by an entire electorate or by its representative (e.g., a randomly selected jury) is binding. A necessary condition of participatory democracy is the presence of informed citizens. Indeed, participatory democracy is a process of collective decision making in which informed citizens have the power to decide on policy proposals [2]. Thus, supporting ICT systems must inform users about proposals and activate a collective process of deliberation in which they debate and vote on them. Such systems must also be designed in a way that reduces the digital divide so as to allow the community of users ready access to their online features. Key to the design of supporting digital systems is therefore the provision of adequate instruments of information and deliberation that are accessible and user-friendly

Building upon the survey presented in [13], together with our classification introduced in [16], we suggest that the following key functions need to be addressed in any software system supporting participatory democracy, that is, systems oriented towards *online participatory democracy*:

- *Information access:* This function deals with the provision of relevant information and makes it easily accessible and understandable to users. The function is typically implemented by a content management system, which must promote user engagement as well as access to and sharing of knowledge.

- *Communication & Engagement:* ICT extend the opportunity for participation from same-place and same-time to any-place and any-time. Nevertheless, dedicated tools are needed to ensure that the relevant group of users is reached and adequately prompted to engage. The mobilization of citizens in this sense depends on digital communication tools, while the various phases of participatory democracy processes (from ideation to contribution making, deliberation, voting, and implementation) should benefit from the diversity of today's communication media.

- *User & group management:* This function must implement a notion of citizenship – understood as membership in a political community or group – that corresponds to the given instance of participatory democracy. The creation and organization of groups that users join as they produce ideas and proposals is critical to ensuring that the participatory process is inclusive and results in collective decision making.

- *Contribution & Decision making:* Online participatory democracy must allow users to collaborate in the development of proposals, a collaboration that spans the phases of proposal making, versioning, deliberation, and voting. What we call contribution making is therefore grounded in collaborative editing. Making this kind of editing available to large groups of citizens is one of the core challenges in designing ICT systems to support participatory democracy.

- *eExpertise:* The role of experts is indispensible in assessing the relevance of the proposals that citizens develop. Although proposal assessment may possibly be partially automated, expert citizens must be involved in the deliberative process and their involvement should be facilitated by any ICT system designed for participatory democracy.

- *Trust management:* The trustworthiness of the digital system supporting participatory democracy is another core challenge. It affects all the above functions because citizens must trust each one as a prerequisite for engaging it. Similarly, it should also be possible to assess the trustworthiness of participants.

There already exist many web-based systems supporting one or a subset of the above functions. The interested reader can find a working list that we maintain at `http://tinyurl.com/civic-tech-platforms`. Another reference is the list of tools for online engagement in [14]. In what follows, we give examples of each of the two major categories of representative systems we find:

- *Systems for civic participation* (see Table 1): The major purpose of the systems in this category is to support contribution making, which includes proposal making, discussion/deliberation, and voting. However, some systems, such as *Agora Voting*, only enable a single sub-function.

  Many of the other functions associated with online participatory democracy in addition to contribution making are implemented as built-in functions of these systems in relation to the underlying user experience and management. However, we do not identify any striking feature compared to classical content management systems regarding the implementation of those functions.

  In their support of contribution making, these tools are primarily oriented towards enabling discussion about and voting on a given proposal. They provide little if any support for users to work cooperatively on the ideation or elaboration of proposals (e.g., collaborative editing).

- *Systems for online collaboration & discussion* (see Table 2): An increasing number of systems support online collaborative editing. These include the well-known Wikis (`www.wikispace.com`) and GoogleDocs (`docs.google.com`). We also note that *Etherpad* is often used by social activists and that *Discourse* is an example of a system for online discussion. *StackExchange* has proven successful in engaging Internet communities to exchange in discussion about a variety of primarily technical topics in the form of questions and

System	URL	Purpose & key feature	Assessment
Liquid Feedback	liquidfeedback.org/	Decision making with a sophisticated voting system that includes delegation.	Little or no support for grassroot-oriented online collaboration and co-creation of proposals. Also unclear if its decision making processes and voting scheme have much flexibility.
Loomio	loomio.org/	Decision making with discussions organized within groups.	Little or no support for bottom-up proposal development and associated collaborative editing tool. Voting is also limited to plurality voting.
DemocracyOS	demo.democracyos.org/	Deliberation and voting oriented toward legislative consultancy for political parties.	Similar to Loomio regarding the limited support for bottom-up proposal development.
Open Town Hall	opentownhall.com/	Cloud-based online civic engagement tool oriented toward collecting feedback for governments, with the aim of increasing and diversifying online public participation.	Little or no support for bottom-up proposal development; primarily intended to assist governments in collecting feedback rather than to support citizen decision making.
Open IDEO	openideo.com/	Online design thinking for social good.	Contribution making initiated by sponsors instead of citizens. Little support for collaborative editing in that edition is handled by a given master, while secondary contributors may get involved through comments only.
Agora Voting	agoravoting.com/	Digital voting through different voting schemes.	Limited to voting.

Table 1: A sample of web-based systems oriented toward civic participation

System	URL	Purpose & key features	Assessment
Etherpad	etherpad.org/	Real-time collaborative editing.	Not tailored for decision making but well-known and used by activists for online collaboration.
Discourse	discourse.org/	Maintaining discussions with a rich system for supporting comments.	Not tailored for decision making but well-known and used for easy integration of discussions in websites.
StackExchange	stackexchange.com/	Question & Answer framework for exchanges about topics that are primarily technical.	Not intended for decision making and collaboration but well-known as an example of a reputation-based system for social knowledge management.
Quora	quora.com/	Question & Answer framework driven by experts and addressing a broad range of topics.	Not intended for decision making and collaboration but well-known as an example of a reputation-based system where expertise is primarily assessed according to qualification (e.g., PhD in physics; software engineer in a renowned company).

Table 2: A sample of web-based systems oriented toward online collaboration and discussion

answers. Although systems in the second category lack support for decision making that is central for online participatory democracy, they are well suited for the implementation of collaborative proposal editing and discussion.

Our state of the art analysis of web-based systems leads us to identify a rich list of examples that may be useful for supporting online participatory democracy. Nevertheless, we also found that there is no existing solution that includes all the essential functions. For this reason, we advocated for the development of the *AppCivist* service-oriented platform that enables the assembly of relevant software systems [16]. Moreover, the development of the *AppCivist* platform led us to identify the central role of collaboration in the participatory process. Indeed, we believe that collaboration must drive contribution making from its initial phase in ideation through to the deliberation and voting phases. The support for collaboration directly affects the organization of users online in that citizens, individually and in groups, must be able to assemble "on the fly" as they meet online, identify common interests, and elaborate their contributions. This pervasive collaborative feature is lacking in all the software systems we have studied. To address this need, we have developed *AppCivist* to enable the co-creation of both *citizen assemblies* and *software assemblies*. The latter embody the notion of service composition, while the former are the digital counterpart of the physical face-to-face meetings that are often central in processes of participatory democracy. In the prototype of *AppCivist-PB*, we have tailored this development for the requirements of Participatory Budgeting.

3. PARTICIPATORY BUDGETING USE CASE

Participatory budgeting processes are among the most illustrative, real-life experiences of participatory democracy [2]. Participatory Budgeting (PB) has its beginnings in the late 1980s, when some Brazilian cities started to experiment with processes of citizen participation in decisions about how to better allocate part of the city's budget [22, 4]. Although PB takes different forms, they can all be considered as refining the following base process: residents of a city propose spending ideas, volunteers or delegates develop those ideas into proposals, residents then vote on the proposals, and the government finally implements the winning projects.

Since the 1980s, PB processes have spread around the world as a set of administrative reforms and, more recently, as a "best practice" in mainstream international develop-

System	URL	Key features	Assessment
Participare	participare.io/	Focused on the voting of proposals and the configuration of eligible voters.	Little support for bottom-up proposal development and diverse voting systems.
Citizen Budget	citizenbudget.com/	Involves residents in decision-making processes and also allows citizens to monitor implementation of selected projects.	Little support for bottom-up proposal development. The tool is primarily oriented toward budget management for selected proposals.
BiPart	monza.bipart.it/	Implementation of PB in 4 phases: collecting problems for the territory (citizens signal problems and map them; other citizens suggest ideas of solution in comments), creating proposals by delegates, evaluating proposals, and voting for proposals.	Little support for bottom-up proposal development. The process by which delegates create proposals is not transparent, and citizen assemblies are managed by cities rather than by citizens themselves.

Table 3: A sample of web-based systems supporting participatory budgeting campaigns

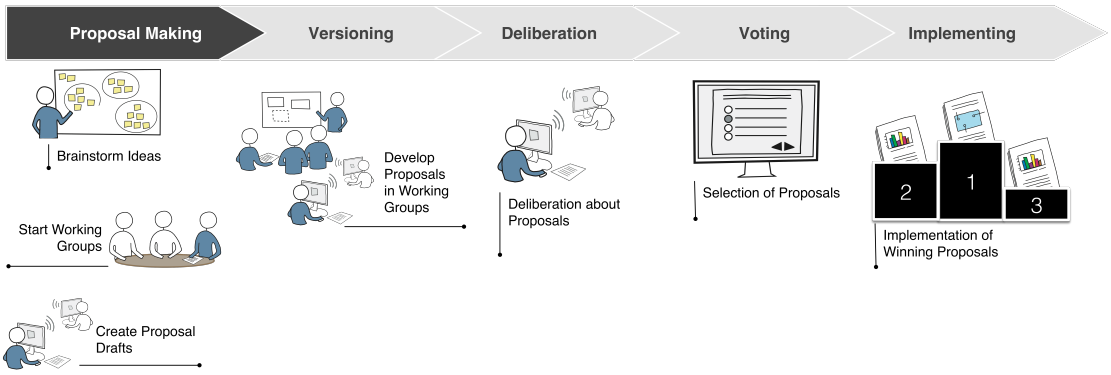

Figure 2: The Participatory Budgeting workflow

ment [9]. Today, more than 200 cities in Brazil [9] and at least 1000 cities around the world [22, 21] (see `http://tinyurl.com/nmtb893` for a map) institute some form of PB. In Europe, Portugal is exemplary, with Lisbon being the first European capital to implement PB and extending the practice to more than 67 municipalities in 10 years [22]. Large cities like Paris are also starting to catch up and recent research has highlighted the expanding number of PB around the world [24]. Most commonly, PB happens at the municipal level, although some provincial and regional examples also exist. Undoubtedly, PB has grown to become a global phenomenon, even if there is no standardized set of practices and processes [9]. This diversity is both criticized [9] and praised [22], the latter view stressing that each PB process is the product of a local reality.

Although a large array of ICT tools exist to support citizens' engagement [14], their use in PB is still limited and scattered. Mostly, ICT have been leveraged for communication or promotion purposes (through multiple channels such as TV, radio and social media) and for facilitating voting for citizens (usually, with custom-made web sites or SMS) [17]. Table 3 lists examples of existing software systems aimed at supporting the implementation of participatory budgeting within cities

With *AppCivist-PB*, we want to enable city governments to configure the *software assemblies* that best match the requirements of the kind of PB campaign they want to support, while leveraging existing software services and components. However, from the overall perspective of participatory democracy, our goal is primarily to facilitate the elaboration of proposals by citizen assemblies that form according to the citizen interests. In other words, we want to support the process that is graphically depicted in Figure 2 and that emphasizes collaborative contribution making at all stages of the elaboration of proposals by diverse citizen assemblies. The collaborative process must in particular facilitate the assembly of groups (or sub-assemblies) on the basis of commonalities among the proposals, which is essential if one wants to sustain city-scale participation and be inclusive of citizen contributions.

To illustrate the issue of city scale, we take the example of the 2015 "*Paris Budget Participatif*" (`https://budgetparticipatif.paris.fr`), the second of the five years that Paris has committed to PB. Parisian residents initially submitted more than 5000 proposals in 2015. A process of pre-selection reduced the number to approximately 600 projects, out of which 8 were selected at the level of the city for a total budget of about 35M€. In addition, 181 less expensive projects were selected at the level of Paris' 20 districts for a total budget of about 59M€. Considering that the City of Paris initiated PB in the first year with 15 proposals, and that residents submitted 5000 proposals in the second, we expect that citizens will submit even higher numbers in subsequent years as they learn more about the process. Hence, PB clearly presents the problem of city scale for software design.

4. APPCIVIST-PB ARCHITECTURE

AppCivist-PB helps users assemble proposal making and selection workflows, like the one depicted in Figure 2, using

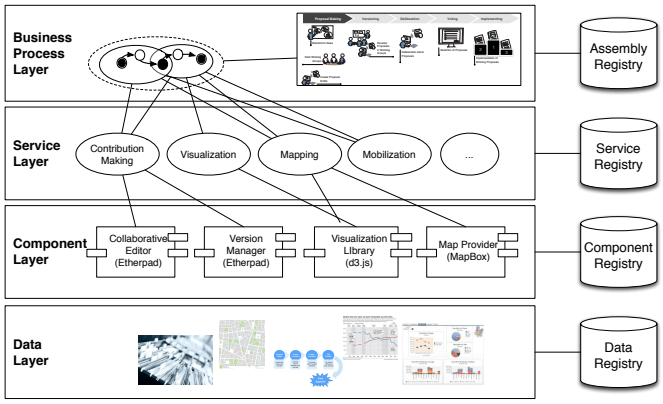

Figure 3: AppCivist-PB service-oriented architecture

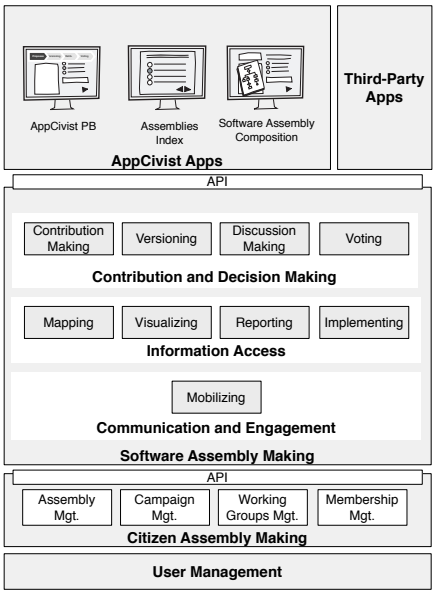

Figure 4: AppCivist-PB architecture

service-oriented architecture (SOA) principles (Figure 3). The composition principles of SOA allow for various implementations and instances of these workflows, including the possibility of integrating and linking different workflows for the same PB campaign. For example, a city might create and manage its own workflow to receive proposals and facilitate deliberation and voting by registered residents; at the same time, citizen groups (typically activists) can create their own, independent, workflows to co-create, develop, and promote proposals for the city, following their own collaboration practices.

Compared to traditional SOA, *AppCivist-PB* distinguishes itself by enabling the assembly of software services dedicated to the support of online-facilitated participatory democracy by and for relevant citizen assemblies (Figure 4).

4.1 Citizen Assembly Making

The primary purpose of *citizen assemblies* is to promote the contribution of citizens at the scale of Internet-based processes. To that end, we have introduced the concepts of *campaigns* and *working groups*, which allow citizens of a *citizen assembly* (e.g., Paris residents) to organize themselves into focused groups. A *campaign* is an initiative that the citizen assembly undertakes to achieve a specific goal within which collective action is organized by *working groups* of citizens. *AppCivist-PB* provides a platform to support this collective action from its ideation to its implementation. These concepts are inspired by practices that are common in town hall meetings and in participatory budgeting cases that emphasize face-to-face meetings among citizens for the purpose of developing ideas into full-fledged project proposals. We also introduce the ability to configure citizen assemblies with different kinds of resources. This is handled through the generic notion of *resource space*, which constitutes a shared space of connected resources.

For illustration, consider the Paris PB campaign. The mayor's office of the City of Paris constitutes a citizen assembly and creates within it the "Paris PB Campaign 2015" within which Paris citizens can contribute proposals. For the City of Paris, what matters most is having a space where proposals can be submitted, deliberated, and ultimately voted by citizens (as depicted in Figure 2). Similarly, a neighborhood association in the Belleville district of Paris defines its own citizen assembly which would like to submit proposals to the "Paris PB Campaign 2015". The Belleville assembly can then create a campaign of its own to elicit proposals from its members to improve its neighborhood. The Belleville assembly is interested in versioning its proposals, and also in receiving technical assessment on them. To that end, the citizens of the Belleville assembly create their workflow around proposal making and versioning, which allows them to form working groups to develop specific brainstormed ideas into full proposals. After they have selected their proposals, they can automatically submit them to the City of Paris, by linking their workflow to that of its assembly. Additionally, *AppCivist-PB* will automatically forward the Belleville proposals to experts for technical assessment and then return their feedback. At that point, the Belleville assembly might decide to engage in additional rounds of versioning to improve its proposals before they are submitted to the City of Paris campaign. Applied in this way within *AppCivist-PB*, SOA principles have enormous potential to enable truly grass-root oriented processes of collaboration and civic engagement.

Figure 5 outlines the core concepts underlying citizen assemblies, which are managed using the functions introduced in Figure 6. In addition, and according to the functions associated with online participatory democracy surveyed in Section 2, *eExpertise* is currently addressed in our system by explicitly recognizing the role of *technical experts* from whom citizen assemblies can voluntarily ask for feedback about their proposals. In the long term, we intend to leverage peer assessment based on techniques that are being developed in the context of massive open online courses [18]. We will also elaborate trust management based on the state of the art in the area [19], while our current system primarily relies on underlying user management, authentication, and authorization.

4.2 Software Assembly Making

Software assemblies allow implementing PB workflows through the composition of services oriented toward online

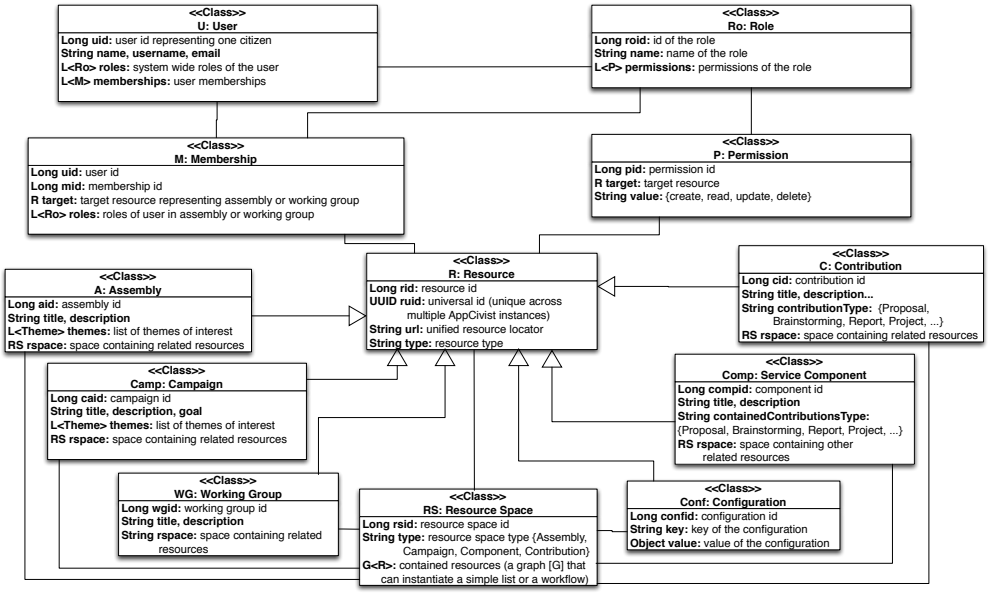

Figure 5: AppCivist-PB Class diagram

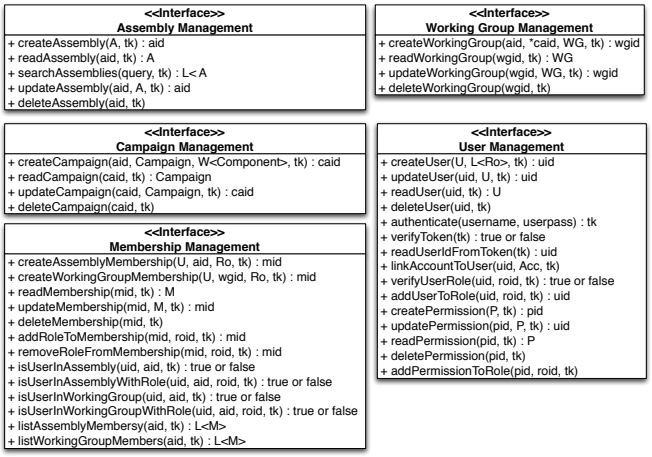

Figure 6: Services enabling citizen assemblies

participatory democracy. These services enable the functions we defined in Section 2, namely, *Communication & Engagement* supported by *Mobilizing*; *Information access* supported by *Mapping*, *Visualizing*, *Reporting* and *Implementing*; and *Contribution & Decision making* supported by *Contribution making*, *Versioning*, *Discussion making* and *Voting*.

Figure 7 depicts the interfaces that are associated with all these services. The parameters of these interfaces refer to the model defined by Figure 5, except for some minor elements that we have not included for the sake of simplification; e.g., *tk* refers to authentication token, *Geo* refers to geographical coordinates, and *D* refers to a set of data points and their descriptions.

Using these interfaces, we will enable the implementation of *Third Party Applications* (Figure 4) that will leverage our open API, in the same way that we develop our own

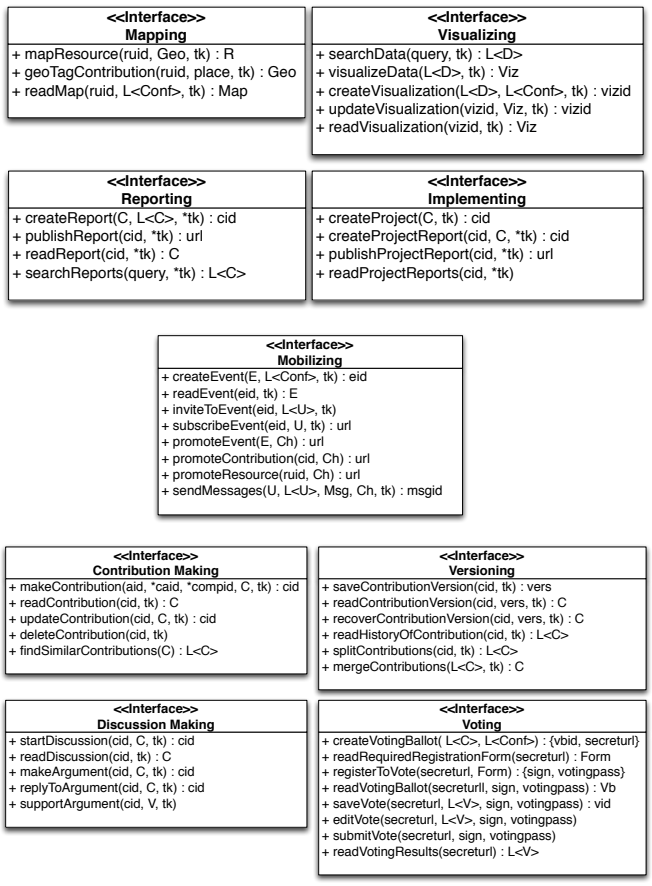

Figure 7: Services composing software assemblies

user interface implementation of *AppCivist-PB*. Eventually,

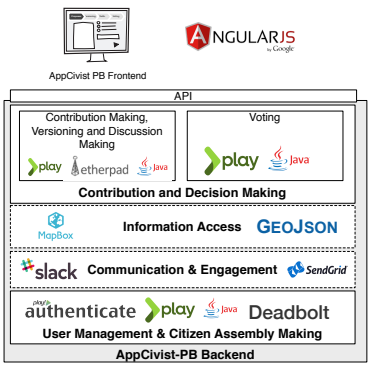

Figure 8: AppCivist-PB prototype

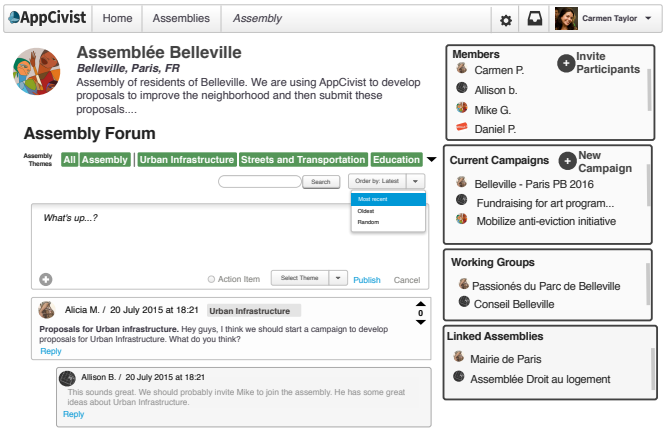

Figure 9: The citizen assembly space

we also aim to integrate third-party services into our own architecture, making them available through our API. This integration requires implementing associated mediators, if possible automatically [6]. However, most existing systems (see examples in Tables 1 and 2) are implemented as full-stack applications without open APIs, as opposed to composable services. Thus, as a first step, we are experimenting with wrapping a few systems to support PB workflows, as the next section discusses.

5. PROTOTYPE IMPLEMENTATION AND EVALUATION

We have implemented a first version of *AppCivist-PB* with the goal of evaluating how our design of citizen and software assemblies enables grassroot processes of collaboration among actual citizens. These processes are supported by campaign workflows created with the prototype. From a software engineering perspective, each of these workflows represent a classical SOA composition. The reuse of existing web-based systems, however, is far from trivial in the application domain of participatory democracy, which is why the prototype does not yet support the full range of SOA composition. Instead, our focus is first to ensure that the assemblies and campaigns created with *AppCivist-PB* support a satisfactory user experience for citizens.

5.1 Software Prototype

Figure 8 depicts the specific components and technologies we use in our implementation. The API backend has been built with Java while the web user interface employs traditional web technologies (HTML5, CCS and Javascript). In particular, we have used *Play* (`www.playframework.com`) and *AngularJS* (`angularjs.org`) as development frameworks.

In the prototype, *Contribution Making* and *Versioning* are supported using *Etherpad* (Table 2). For *Communication & Engagement*, we are integrating *Slack* (`slack.com`) and *SendGrid* (`sendgrid.com`). *MapBox* (`www.mapbox.com`) and the *GeoJSON* data standard (`geojson.org`) are used for mapping in *Information access*.

In contrast to this utilization of existing services, our system uses its own design and implementation for *Discussion Making*, *Voting*, *Citizen Assembly Making*, *User Management*, and some aspects of *Contribution Making*. Although the *User Management* services are our own, we use externally developed user authentication and authorization components, namely, *play-authenticate* (`joscha.github.io/play-authenticate`) and *Deadbolt* (`deadbolt.ws`).

5.2 User Experience

Figures 9 to 11 provide a glimpse of the *AppCivist-PB* user interface[1]. Once citizens create an assembly (Figure 9), they may engage into the collaborative elaboration of contributions via the creation of campaigns (Figure 10). The campaign creation allows many options for each phase of the process: from configuring the proposal template to selecting among several voting systems. The user interface supports all the phases of the PB process from the brainstorming of ideas to final voting (Figure 11). Citizens can organize themselves into working groups and plan their own schedules. They can also synchronize with other assemblies and campaigns, for example, to submit proposals, receive information, and share campaign themes. Similarly, they can select from a number of voting systems the one that best suits a specific campaign.

5.3 Early Evaluation

As a preliminary evaluation of the prototype, we have conducted informal discussions with students, activists, and social science researchers during the course of a student research apprenticeship laboratory hosted by the Social Apps Lab at CITRIS (`citris-uc.org/initiatives/social-apps-lab`) of University of California, Berkeley. We have held these discussions regularly since September 2015. Using a mockup version of the prototype, we have evaluated the front-end design and the backend logic. We have especially engaged the broader question of how the pedagogic principles and practices of democracy that we embed in both the front- and back-end might improve the quality of democratic deliberation and decision making.

One conclusion is that there seems to be a strong tension between usability and pedagogy. This is most noticeable in the process of creating and configuring *assemblies*, *working groups*, and *campaigns*, where having full-flexibility of choices makes the configuration quite complex because

[1]The complete mockup is available at `http://tinyurl.com/appcivist-pb-wireframe`

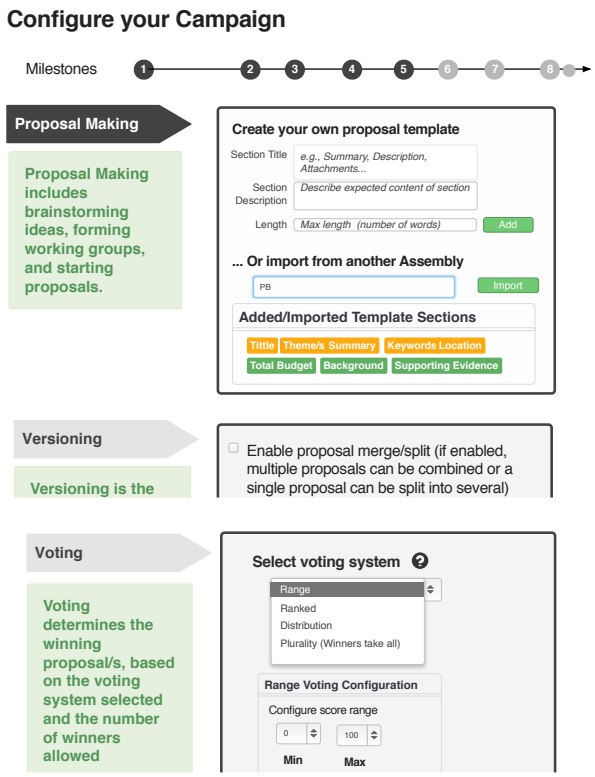

Figure 10: The campaign creation process

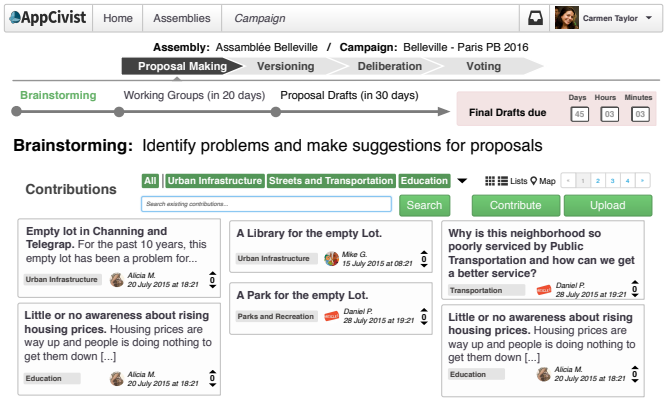

Figure 11: Proposal and decision making: From brainstorming ideas to developing proposals to voting

users have to make many decisions. Some users consider this complexity an impediment to using the software. Others consider it fundamental for learning about democracy and how to craft participatory practices. The software platform 'forces' users to make significant choices about, for example, who should be a member of a working group (anyone or only residents?), who should deliberate or make the final decisions on proposals (all members or random juries?), what type of voting system (range or ranked?). Some users experience making these choices as a burden and disincentive, while others see them as prompts for a deeper consideration both of democracy itself and of their own aims. For the latter group, removing the full flexibility of choice amounts to "dumbing down" a system the purpose of which should be not only to facilitate civic action but also to improve the quality of democratic arguments, deepen knowledge of democratic procedure, expand the scale of direct assembly, and nurture stronger commitments to participatory democracy as a model of self-governance.

In an attempt to balance both perspectives, we developed a "fast-track" version of the PB workflow, which configures carefully considered default options automatically while prompting users about the most important implications of these selections. Among the features that users strongly appreciated is the way that the *AppCivist-PB* prototype enables user collaboration throughout the phases of a PB campaign. They also valued the way it is able to link campaigns that take place in different deliberation spaces, for example, campaigns that develop proposals in the space of neighborhood assemblies while the final decisions about them takes place in a more institutional, city-wide arena. Enabling campaigns to speak to each other across urban time and space appears to make direct democracy possible at a new and vast scale.

6. CONCLUSION

The pervasive use of ICT presents an unprecedented opportunity to rethink the constraints of time and space that are generally thought to make the exercise of a more direct and engaging democracy at a large scale practically impossible. To that end, an increasing number of web-based systems may be useful for supporting online participatory democracy. Nevertheless, our research found that there is no existing solution that includes all the essential functions. For this reason, we advocated for the development of a service-oriented platform that enables the assembly of relevant software systems in [16].

Leveraging service composition to engineer *software assemblies* that support participatory democracy is a vision that both holds significant promises and faces important challenges. In particular, we have identified that the concept of software assemblies needs to be complemented with that of *citizen assemblies* so as to promote citizen participation in a bottom-up fashion. Using Participatory Budgeting as our initial use case, we have implemented a first prototype, called *AppCivist-PB*, that supports both concepts. This development has informed our understanding about the underlying concepts and the many challenges that software engineering must solve to facilitate the creation of software and citizen assemblies. Further, the availability of the *AppCivist-PB* prototype that we are currently finalizing will allow us to undertake pilot tests with activist groups engaging in PB campaigns.

Another major area for future work relates to the reuse of relevant third-party applications. Indeed, applications in the domain of civic participation and social activism are full-stack web or mobile applications, with their own processes and models of decision making and deliberation that cannot be easily separated and isolated. From a usability perspective, using just part of an application within another (e.g., using just the discussion model of *Loomio* without its integrated proposal/voting scheme) could complicate the different functions we need to support, hindering the overall user experience. Also, some operations provided by third-

party systems are only possible if other operations were executed before, e.g., *Loomio* discussions can only be created if a group was previously described. Most of these requirements are not documented in formal ways. Similarly, there are cross-data requirements between the different resources that are managed by the service, e.g., a *Loomio* discussion cannot be private if it is created inside of a *Loomio* public group. These requirements are also seldom formalized in the service definitions or documentations, which means that some level of "reverse engineering" will always be needed for each service. We intend to leverage our background on service mediation to address this challenge [1, 5]. Finally, while composition might be easy for developers, we need to design the user experience so that it makes it feasible for activists to compose services and thereby easier for them to achieve their goals.

7. REFERENCES

[1] E. M. Andriescu, T. Martinez, and V. Issarny. Composing Message Translators and Inferring their Data Types using Tree Automata. In *Proc. FASE*, 2015.

[2] E. Aragonès and S. Sánchez-Pagés. A theory of participatory democracy based on the real case of porto alegre. *European Economic Review*, 53(1), 2009.

[3] M. Asad and C. A. Le Dantec. Illegitimate civic participation: Supporting community activists on the ground. In *Proc. CSCW*, 2015.

[4] G. Baiocchi, P. Heller, and M. Silva. *Bootstrapping Democracy: Transforming Local Governance and Civil Society in Brazil*. Stanford University Press, 2011.

[5] A. Bennaceur, E. Andriescu, R. Speicys Cardoso, and V. Issarny. A Unifying Perspective on Protocol Mediation: Interoperability in the Future Internet. *Journal of Internet Services and Applications*, 2015.

[6] A. Bennaceur and V. Issarny. Automated synthesis of mediators to support component interoperability. *IEEE TSE*, 41(3), 2015.

[7] C. Boulding and B. Wampler. Voice, votes, and resources: Evaluating the effect of participatory democracy on well-being. *World development*, 38(1), 2010.

[8] Y. Cabannes. Participatory budgeting: a significant contribution to participatory democracy. *Environment and Urbanization*, 16(1), 2004.

[9] E. Ganuza and G. Baiocchi. The power of ambiguity: How participatory budgeting travels the globe. *Journal of Public Deliberation*, 8(2), 2012.

[10] D. Held. *Models of democracy*. Polity Press, 1987.

[11] J. Holston. Urban citizenship and globalization. In *Global City-Regions*. Oxford University Press, 2001.

[12] J. Holston. *Insurgent citizenship: Disjunctions of democracy and modernity in Brazil*. Princeton University Press, 2008.

[13] D. R. Insua, G. E. Kersten, J. Rios, and C. Grima. Towards decision support for participatory democracy. *Information systems and e-business management*, 6(2), 2008.

[14] M. Leighninger. Using online tools to engage- and be engaged by- the public. Technical report, IBM Center for The Business of Government, 2011.

[15] P. Newman, M. Ferrario, W. Simm, S. Forshaw, A. Friday, and J. Whittle. The role of design thinking and physical prototyping in social software engineering. In *Proc. ICSE-SEIS*, 2015.

[16] A. Pathak, V. Issarny, and J. Holston. Appcivist: A service-oriented software platform for socially sustainable activism. In *Proc. ICSE-SEIS*, 2015.

[17] T. Peixoto. Beyond theory: e-participatory budgeting and its promises for eparticipation. *European Journal of ePractice*, 7(5), 2009.

[18] C. Piech, J. Huang, Z. Chen, C. B. Do, A. Y. Ng, and D. Koller. Tuned models of peer assessment in moocs. *CoRR*, 2013.

[19] R. Saadi, M. A. Rahaman, V. Issarny, and A. Toninelli. Composing Trust Models towards Interoperable Trust Management. In *Proc. Trust Management*, 2011.

[20] R. Sandoval-Almazan and J. R. Gil-Garcia. Towards cyberactivism 2.0? understanding the use of social media and other information technologies for political activism and social movements. *Government Information Quarterly*, 31(3), 2014.

[21] Y. Sintomer, C. Herzberg, and A. RÃűcke. Participatory budgeting in europe: Potentials and challenges. *International Journal of Urban and Regional Research*, 2008.

[22] UN-Habitat. *72 Perguntas frequentes sobre Orcamento Participativo (Portuguese Version)*. Urban Governance Toolkit Series. 2009.

[23] J. Van Dijk. Models of democracy and concepts of communication. *Digital democracy: Issues of theory and practice*, 2000.

[24] B. Wampler and J. Hartz-karp. Participatory budgeting: Diffusion and outcomes across the world. *Journal of Public Deliberation*, 8(2), 2012.