Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Tomasz Marek Rudowski

On Liquid Democracy in OSS decision making

Master's thesis in MACS
Supervisor: Mariusz Nowostawski
June 2022

NTNU
Kunnskap for en bedre verden

Tomasz Marek Rudowski

On Liquid Democracy in OSS decision making

Master's thesis in MACS
Supervisor: Mariusz Nowostawski
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Kunnskap for en bedre verden

On Liquid Democracy in OSS decision making

Tomasz Marek Rudowski

Abstract

We make decisions every day and some of them have impact on our surroundings in a various scale; in the context of an Open Source projects the decision outcome that is consider fair and of high quality is an art of balancing and interpreting the will of the community, experts and one's own experience. Establishing a secure and trustful decision-making model would make more individuals content about the fairness of the community choice and increase the quality of decision.

Liquid Democracy and emerging forms of delegated democracy are interesting alternatives to a traditional direct democracy, majority voting or dictatorial decision making. The goal of the study is to analyse, extract benefits and apply Liquid Democracy to Open Source management in order to achieve quality and fairness of decision and increase chances of the success of the project.

The exploration of decision models and implementation of the simulation framework allowed to conduct experiments which confirmed the possibility of reaching higher quality of outcomes in a collective decision model that employs the delegation voting mechanisms.

Sammendrag

Vi tar beslutninger hver dag, og noen av dem har innvirkning på omgivelsene våre i ulike skalaer. Beslutningsresultatet, som anses som rettferdig og av høy kvalitet i Open Source prosjekter, er en kunst å balansere og tolke viljen til fellesskapet, ekspertene og ens egen erfaring.

Etablering av en sikker og tillitsfull beslutningsmodell vil gjøre flere tilfredse med rettferdigheten til fellesskapets valg og øke kvaliteten på beslutningen.

Liquid Democracy er et fremvoksende alternativ til en tradisjonell flertall eller diktatorisk beslutning. Målet med studien er å trekke ut dens fordelene og bruke dem til å støtte administrasjon av Open Source prosjekter for å oppnå kvalitet og rettferdig beslutning og øke sjansene for å lykkes med prosjektet.

Utforskningen av beslutningsmodeller og implementering av simulerings rammeverket tillatt å gjennomføre eksperimenter som bekreftet muligheten for oppnå høyere kvalitet på resultater i en kollektiv beslutningsmodell som benytter delegasjonsstemmemekanismene.

Contents

Figures

Chapter 1

Introduction

Making a collective decision is a challenge which exists in the modern society on various levels; from making a decision in a team, a school class or on the larger scale like local or national communities. Quality and fairness are characteristics of both the outcome and the process of decision; they have impact on surroundings in terms of how well they represent the will of the community.

1.1 Project Objectives and Scopes

This project aims to evaluate if it is possible, concerning social and technological limitations, to model a decision-making system that could be considered as fair and help to make better collective decisions; it focuses on aspects after the individual decision has been made.

The scope of the project is within the collective decision-making processes supported electronically supported (e.g. e-voting). In context of an e-voting system the process of gathering individual choices and generating the results are divided into phases which are addressed individually concerning voters' acceptance of the entire voting process; privacy and anonymity are among considered aspects.

The application context is a community decision in Open Source Software (OSS) projects, the research considers different stages of development where decisions can influence the shape of the project (e.g. features, patches).

The project explores collective decision-making processes and how they are or could be supported with modern technology. One of the decision models is called Liquid Democracy and would allow, in addition to making a decision directly, delegation of the vote to another person. The main concerns regarding Liquid Democracy are in which terms would it be better to operationalize fairness of decision, increase its quality and what are technological possibilities and challenges; the project focuses on those fields.

1.2 Keywords

collective decision, e-voting, liquid democracy, open source

1.3 Problem Description

Democratic choices give equal rights (one vote) to each eligible person in a direct voting, so an expert in the decision field cannot get more influence power unless manages to convince others to own knowledge-based decision by being a final representative; literally becoming an available decision choice for each voter, i.e. a candidate for a decision maker position.

After the final decision of electing representatives (decision makers) is made, the voters lose control over the elected ones until next elections; although the support could be revoked afterwards their representatives keep the gained decision power from the election day until the end of the term.

An alternative direct decision making through citizen law initiatives might be difficult and expensive to complete, concerning e.g. gathering support signatures without using e-channels.

Less formally connected groups, like e.g. OSS communities, also need to express their expectations, regardless of how the final decision might be made, e.g. by a higher authority; making a decision acceptable for the decisive authority and at the same time considered fair and of high quality by voters, is an art of balancing and interpreting the will of the community, experts and own experience.

1.4 Justification, Motivation and Benefits

From a society perspective establishing a secure and trustful decision-making model would make more individuals content about the fairness of the community choice.

The encouragement to use e-voting system, as an alternative to the traditional polling-stations, might also increase community members' utilization of other e-governance systems.

An electronically supported voting system that would gain the trust of users could increase a number of active voters', i.e. those who would not participate in e.g. direct citizen initiatives.

A voting system that would allow revoking of support or other form of the control over representatives, could help the individual to decide; knowing that one does not have to wait until the next election to make a change.

Benefits of the fair e-voting system could be gained on various levels and in various contexts; in smaller scale it could be employed when e.g. deciding the direction (or just current focus) of the development.

An improved collective decision-making system could be implemented in an OSS project, e.g. as a part of decision to accept or reject a patch during code review; such a tool could be useful for the decisive authorities to make a high quality decision and to maintain community's acceptance for the project progress, which may be important for its survival.

1.5 Research Question

Q0: How can Liquid Democracy improve quality and fairness of decision-making processes in OSS development?

Hypothesis: Liquid Democracy CAN improve quality and fairness of decision-making processes in OSS development.

Supporting Research Questions

Q0.1 What are the current limitations of OSS decision making?

Q0.2 How Liquid Democracy could be used in OSS decision making?

Q0.3 What are the key architectural variables in delegative democracy models and how do they influence the decision making process?

1.6 Tasks

T1: Prepare the taxonomy of decision making in OSS development (models, stakeholders' interactions, etc.).

T2: Operationalize the *Fairness* and *Quality* of decision making.

T3: Prepare the model of collective decision making inherited from the Liquid Democracy and compare it with findings from task T1.

T4: Combined results of tasks T1-T3 to address research questions.

1.7 Contribution

The principal contributions of the thesis are:

- Literature review concerning collective decision making and governance in the open source projects.
- Creation of the decision model inherited from Liquid Democracy principles.
- Development of the simulation framework for collective decision making process.

- Conducting experiments on decisions models implemented in the simulation framework.

Chapter 2

Background

This chapter presents the summary of findings from literature reviews and previously conducted studies. The first part consist of the analysis of the decision making processes in open source projects; it relates to the first (T1) task. The next parts that explore the participants' perception of collective decisions procedures, concern both trust to the system and the decision outcome; findings presented contributed to the operationalization of *Fairness* and *Quality* terms (task T2) and creation of the voting model (task T3). The introduction to the Liquid Democracy principles, included in this chapter, presents the perspective on this decision model concerned in this thesis and points threats to be considered in the process of decision model proposal (task T3).

2.1 Governance in Open Source Projects

The project management and decision fields in open-source projects development process have been previously studied during the IMT4894 - Advanced Project Work course; the content presented in this section is based on findings from the study. First in this section, common governance models and external influence on decision are described; followed by development activities presented in two activities groups: before coding (requirements, design) and related to code contribution (patch, code review and integration).

Activities and individual participation in open-source projects has been explored by Cheng and Guo [1]; two contributor groups: active and supporting, and six main activity categories: knowledge sharing, issue reporting, code tweaking, progress control, issue coordination and code contribution have been recognized. Concerning different weight on each of the recognized activities authors classified community members into multiple categories e.g., Engage Issue Reporter or Progres Controller.

Another classification of members of an open-source community into two categories based on role within the community has been used by Eckhardt et al. [2]; this simplified classification will be used in this thesis and to describe two subgroups of the community during the experiment settings (Section 6.1).

Committers. The group with more rights and privileges, but also more responsibilities (e.g. project resources, code integration). Also referred to as core developers.

Contributors. Common members of project's community (e.g. code contribution, testing).

2.1.1 OSS development and governance

Decision-making models, using a case of the Python project, have been studied by Keerpathi et al. [3]; two general approaches to decision-model could be noted: the first one involves a single body that make a decision, while the second is a collective decision. In the case of a decisive body, it could take a form of a committee of e.g. experts (*meritocracy*), or could be a single person (*benevolent dictator*) having either permanent or time limited decision rights.

Concerning general governance rules being a part of software engineering, authors like Izquierdo and Cabot [4] studied their influence on open-source projects and argued for those rules not being widely implemented. Governance is being supported through a practical approach to e.g. issue/bug tracking systems and communication tools, without a formalize definitions; which could emphasize understanding and transparency of decision making.

In the context of a commonly used in OSS patch technique (development of a solution to a specific feature or bug), Izquierdo and Cabot [4] presented three decisions to be made: selecting an issue to be worked on, reviewing presented solution and selecting candidates for release.

Decisive roles in OSS projects are often related to the internal hierarchy of collaborators; there might be people with special rights, responsibilities or a dedicated role (e.g. "approvers" or "verifiers" in Android, Bug Squad in GNOME, leaders in Eclipse). The special position might be received based on individual contribution (meritocracy model, e.g. Apache Web Server project) [4].

Open-source projects governance have been studied by Herbsleb [5] from a perspective of global software development; coordination in a cross-site collaboration (on various levels, e.g. by component, by area of expertise etc) is positively influenced by a decentralized decision-making, with a key factor of transparency when sharing responsibility.

Chulani et al. [6] studied governance of software development on an example of Linux project evolution; three aspects has been discussed: governance, management and processes (relations presented on Figure 2.1). Authors presented the "star-shape" form of the chain of responsibility with Linus Torvalds in the center followed by rings of core developers, committers etc; members of each level has specific responsibilities like from conceptual integrity (Torvalds) or policy management to managing a group of contributors (core developers).

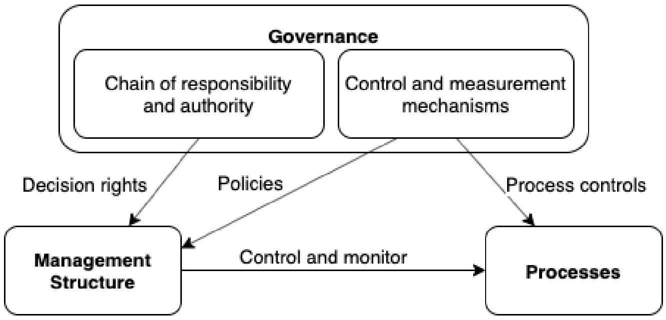

Figure 2.1: Governance, Management and Processes. Based on Chulani et al. [6]

2.1.2 External influence

An external influence has been recognized as a factor that may determine decisions in OSS projects. Example of such is existence of foundations, acting as a form of umbrella over open-source projects; some foundations may also through cooperation with commercial organizations try to impact decisions in underlying open-source projects. The influence of external actors will be discussed in Section 4.8.6 concerning asymmetric voting power of decision makers.

Izquierdo and Cabot [7] studied the influence of foundations, and although they may play various roles and differs also in the level of engagement; foundation support may be realized by providing enhanced communication possibilities, recommendations, policies or more direct form of e.g. technical body (councils) with an advisory role; board members are usually elected for short terms by majority voting by other members.

Eckhardt et al. [2] explored external impacts on decisions on the example of the Eclipse ecosystem. The project is governed by the Eclipse Foundation through councils and boards established in cooperation with commercial actors, e.g. the Board of Directors (the highest authority, decisive in the context of strategy and policies) consists of six people chosen by project committers and ten originating from foundation members (companies like IBM or Google).

Some of the governance bodies having a supportive role for individual projects in the ecosystem are: Planning Council (concerning e.g. releases), Eclipse Architecture Council (revise and mentor role for technical aspects), Project Management Committee (management and confirming development process policies) [2].

Individual projects differs participant roles on privileged *committers* (part of Development Team, often employees of foundation member company) and regular *contributors* (could be promoted to join *committers*). Meritocracy principles are involved in promotion, since acceptance is given by Project Management Committee based on work and skills [2].

2.1.3 Preparation phase

One of the fields where decisions are to be made is the initial phase of an open-source project. Here two of the activities: requirements elicitation and design phases are presented.

Requirements elicitation

Requirements engineering in open-source projects tends to be influenced by a negative effects of distributed collaboration, and the more informal: *just-in-time* approach is argued by Bhowmik et al. [8] to be beneficial (keeping the balance between *up-front* resources and agile development). This approach is argued to address changes over time within the community (members and/or expectations) itself. This represents discussed in the study importance of social interactions and *cross-group knowledge* to embrace diversity and avoid bias of experts in a field when introducing a new requirement.

Similar importance of social interactions and the transparency of requirements engineering process is argued by Dabbish et al. [9].

Heck and Zaidman [10] explored requirements engineering in open-source projects; authors argued that commonly used for this purpose issue tracker systems are insufficient and often need to be supported av extended policies or additional systems where preliminary decisions are made before the new requirements is registered as a new issue.

Design

Building a group consensus has been argued by Moghaddam et al. [11] as more important in design than technical cases; such consensus is according to authors more difficult to achieve in distributed communities.

The authority of developer and proposing of a concrete solution has been presented by Ko and Chilana [12] as the most important factors for the final design decision.

2.1.4 Code contribution

A common development model in open-source projects is defined in various studies as a *pull-based* or *fork-based*, since one of the steps is creating a *fork* of the original repository; contributors work on solutions on the *fork* and propose changes in a form of a *pull-request* back to main repository. The decision regarding including proposed contribution involves code review process. Accepted changes goes the through a code integration process [13].

Padhye et al. [14] argues that this development model for independent contribution could be easy practiced in collaboration ecosystems like GitHub which take advantage of benefits (included functionality) of "modern decentralized version-control systems". GitHub, due to extended support for distributed pull-based development, is also argued by Vasilescu et al. [15] as being the most open platform allowing pull-based contribution to any hosted project.

Development practice in open-source projects, studied by Zhou et al. [16], may have negative consequences caused by either permanent separation ("*hard fork*") or creating *forks* from already separated code base instead of the main line ("*fork of forks*"); authors propose higher project coordination and dividing projects in smaller modules to reduce number of rejected *pull-requests*.

Zhou et al. [17] explored forked-based development in open-source projects; they argued for embracing the transparency of decisions to avoid pitfalls common in social coding like work being lost or done redundantly, or not optimized time of creating a new fork.

Code review

Peer reviews (a common technique of code inspections) created a base for the technique of code reviews widely utilized in OSS development; Rigby et al. [18] discussed it as an important factor for maintaining code quality, although their findings suggests low involvement of project community in the process, which needs then be compensated with reviewer's expertise.

Rigby et al. [18] presented two approaches to code review process: (argued to result from "asynchronous nature of OSS") "review-then-commit" (RTC) and less commonly used "commit-then-review" (CTR). RTC process includes upfront discussion on community forum, followed by decision if patch is worth reviewing, so the the code review itself is performed for chosen contribution. The CTR starts with merging patch and then publishing changes for community to perform the review afterwards.

Hirao et al. [19], argued that a collective decision-making has only a supportive role for the process of code review; it has been revealed that a decision made by community through a majority voting is not always followed by the decisive body. Authors argued for increasing minority voice impact on the final decision through including voter's preferences and position.

The code review process includes choosing a reviewer for proposed contribution; often it is required that one of the core developers must accept the pull-request ("committer's review" [14]). The elicitation of reviewer may be a challenge, Ying et al. [20] discussed existing practices based on either user work history - finding most active committers (concerning contributions, reviews) or by analysis of user relations with other contributors; authors proposed the recommendation algorithm to increase efficiency of the decision process to choose a suitable reviewer concerning his/her "*expertise and authority*".

Gausios et al. [21] recognized code review as a most common part of the process of including contribution (code integration) in projects organized on pull-request development model principles; important factors for patch acceptance are: the quality of code (could be interpreted as following "the project's current style" and non-functional traits, e.g documentation), realisation of project's goals and social aspects like the reputation and the work history of contributor.

2.2 Electronically Supported Voting

The review by Wang et al. [22] analyzes e-voting challenges considering four aspects: requirements that the system should fulfill, common roles related to the voting process, voting stages and usability of the system.

Requirements. Authors considered following requirements as essential: correctness of counting of each ballot, protecting privacy of voter, assurance that only eligible voter can participate and could cast a ballot only once, robustness of the voting system regarding potential voters' misbehavior, possibility to verify that the vote was counted.

Roles. Concerning roles beside the voter him/herself, Wang et al. [22] recognized following parts of the voting system: the registrar - controls eligibility of voters, the pollster - voter's agent, the validator - responsible for certificates attached to encrypted ballots, the tallier - collects submitted ballots.

Stages. The e-voting process includes following stages: the registration of voters, validation of voters' eligibility to participate, processing ballots by collecting, verifying and counting them and at the end resolving of the potential claims [22].

Usability. Usability of the system is considered as an important aspect since the voters' attitude may affect overall evaluation of the entire solution. Wang et al. [22] suggests possible research direction in order to effective implementation of an e-voting system; the main areas are securing privacy and enhancing usability that will align with user education to ensure their trust in the solution.

2.2.1 Trust and participation

Findings from previous exploration of systems used for remote voting in political elections, conducted during the IMT4887 - Specialisation in Web Technology course, are presented below. The study focused on factors to empower trust to voting system and participation in elections. Four of the measures found in the explored studies: transparency, privacy, verifiability and possibility to update vote, have been recognized as relevant for further discussion in the context of collective decision-making system in OSS projects. Transparency and privacy influence on trust and participation will be discussed in Section 4.8.6 concerning possible improvements to the proposed decision model.

Transparency

Study conducted by Carter and Belanger [23] suggest that doubts towards the voting system have a significant social impact and may affect individual voting system. Santamaria-Philco and Wimmer [24] also addressed the influence of clearness and trust towards voting procedures on individual choices of voters; authors recognized the transparency of process flow in the system and the transparency of decision outcome as essential factors in building trust.

Transparency of the voting system could be achieved through information sharing on various levels: from user manuals or Q&A information sites to technical documentation of the system (Volkamer et al. [25]); authors noted the risk of exposing system's vulnerabilities.

Privacy

Voter's privacy and anonymity have been addressed in explored studies as an important part of the voting system. A specific solution: the separation of duty has been presented by Volkamer et al. [25]; different part of an election process (e.g. voting, vote telling etc) could be realized by independent distributed systems (with limited scope of work). Authors argued also that protecting the final results of election is an important element of system security.

An opposite perspective on keeping the final decision outcome secret has been presented by Vlachokyriakos [26]; according to the study, reveling the current state of voting outcome is an important "social stimuli on the quality of decisions made".

Verifiability

Ability to verify own vote has been discussed in explored studies; Volkamer et al. [25], Cortier and Lallemand [27] and Adeshina and Ojo [28] addressed similar fields and propose measures that could be summarize in three main aspects of what could be verify by each voter:

Eligibility. Only valid (registered) voters may participate in election.

Universal. All recorded votes are properly tallied.

Individual. Voter's vote is included in final outcome.

Vote updating

In the context of explored political elections, possibility to update one's vote is mainly discussed in relation with coercion avoidance (Volkamer et al. [25]), which may occur while voting remotely (i.e. outside the controlled polling station environment). The practice of re-voting exists in countries that applied electronically supported voting, like in Estonia where only the last vote is counted towards the final decision (Kitsing [29]).

2.3 Liquid Democracy

Findings presented in this section concerns how the Liquid Democracy is defined in the literature and had a direct influence on creating the decision model (presented in Chapter 4).

Brill [30] presented the Liquid Democracy as a part of a more general field of the Interactive Democracy, which was defined as an upgrade of the democratic principles (for collective decision-making), more suited to the modern society and common use of information technology, persuading to higher participation and more responsive. The study presented elements of the collective decision-making system which should be considered in design: selection of final decision options and organizing available voting behaviour options (including how they are presented on the ballot) and "aggregation methods" i.e., how the final decision outcome is calculated. Brill equated Liquid Democracy with the delegative voting which "aims to reconcile the idealistic appeal of direct democracy (where every voter votes directly on every issue) with the practicality of representative democracy (where voters vote for delegates, who then vote on the voters' behalf on all issues".

Gölz et al. [31] described Liquid Democracy as a collective decision-making system, which introduces the concept of delegated voting into a classical democratic procedure; voters can either support the final decision option or another voter (taking then the role of a delegate). The delegate can also become a follower, instead of casting direct vote, so the chain of delegation may emerge.

Authors [31] argued for advantages of Liquid Democracy while considered how democratic decisions systems may be evaluated using epistemic and egalitarian approaches. Concerning epistemic metrics, good decisions should be made in the system; therefore, delegating votes to more competent voters will increase the quality of collective decision. Egalitarian metric concerns individual voter's interest in the collective decision; here the advantage of Liquid democracy is argued, that through a delegating voting one could easier express own preferences while lacking sufficient information regarding decision issue [31].

Kahng et al. [32] defined the Liquid Democracy as "a practical compromise between direct democracy (...) and representative democracy"; authors explored the process of delegation voting from an algorithmic perspective to reveal on what terms it performs better than a direct voting.

Two, "positive gain" and "do no harm", properties were defined in the study [32]; the research assumed, that the delegation will happen only towards more competent voters, so the first of metrics, the "positive gain" will be satisfied by the delegation process itself – i.e., will perform at least as good as direct voting. More problematic are situations that could jeopardize the benefits of delegation voting ("do no harm" property). One of such examples is existence of a super-voter (or a group of popular voters) i.e., a delegate(-s) who wields through the delegation chains the voting power of all voters; this leads to the single point of failure, since the decision is made by that one voter (or a limited group) voter - Kahng et al. [32] concluded that such situation is not to be avoided unless limitations to who voter can delegate voting power to are introduced.

Gölz et al. [31] also addressed the possibility of gaining through delegation too much voting power in the hands of “super-voters”, which may question benefits of Liquid Democracy; the study concerns the concept of delegating votes to more than a single delegate to avoid voting power concentration.

Brill [30] pointed other issues they may emerge during the collective decision-making process; the existence of delegation cycles, delegate abstain during a single decision round and on general level the possibility of “inconsistent outcome”, defined as “globally incompatible set of decisions”. Author proposed using a default vote or ranking delegation to avoid loss of vote due to delegation cycle or abstention of the delegate. The term of “strategically manipulated” elections has been presented by Brill as possibility for voters “achieve a preferable outcome by misstating their preferences”; author named the delegation transparency here considered as possibilities to see the final decision option being supported by the voter through a delegation chain (combined with the chance to change the voting options) as a factor increasing the risk of the strategic manipulation of elections. The threats to the proposed decision model are addressed later in Section 4.7 of model creation chapter.

Brill [30] noticed, based on findings from a previous work by Schelling (referred in the study to Nalebuff [33]), that the unwanted aspect of delegative voting which is the burden for delegates that may occur with the decision power exceeding the power of a single voter; the presented solution include giving a delegate opportunity to accept only chosen followers instead of all willing to give their voting power.

Brill [30] argued, that since the voting rules cannot satisfied all wanted properties, the design of the voting system within the context of the interactive democracy need to consider the trade-offs between satisfied and violated properties. Additionally, another finding coming also from the social choice theory was presented; the author argued that “way preferences are aggregated matters”.

2.4 Collective Decision Making

Research concerning decision-making processes exist in social science literature, but are also present in the fields where the decision itself is applied; those research explore the problem from both methodology and practical approach perspectives in the context of specific environment of the decision.

2.4.1 Choosing the decision model

Falessi et al. [34] explored decision-making techniques in the field of software architecture; the study contributed to the meta-decision-making problem i.e., "deciding how to decide".

Three main types of decision-making techniques that are commonly used in software engineering: "keeping the first available alternative", "selecting among a finite number of alternatives", "selecting among an infinite number of alternatives", have been presented in the research [34]. The study allowed also to extract four common elements in explored decision-making techniques:

Quality Attribute Description. Interpreted as the ability to gain a common understanding of subject, i.e. system traits in the scope of the study, by utilizing terms, use cases and/or measures.

Quality Attributes Importance Description. Applied when system attributes are not "equally important for a stakeholder", may be weighted using different criteria and final weight value calculations, but also not present at all if the technique doesn't consider ranking quality attributes),

Fulfilment Description. Since Quality Attributes are not equally addressed by explored techniques this category is used for describing the level of fulfilment, e.g. determining the acceptance as discrete yes/no criterium, a listed scale e.g. from "bad" to "good", or a fulfilment ratio/percentage.

Uncertainty Description. The reliability of a specific solution/alternative to reach expected fulfilment level.

Falessi et al. [34] presented a set of difficulties related to the process of designing the system architecture and evaluated the quality of explored decision-making techniques; authors concluded that no technique is best suitable for addressing all potential difficulties and the choice of decision model should involve choosing which of those challenges are the most likely to be relevant in decision process, and so finding best alternative from presented ranking list.

2.4.2 Collective decision procedure

The importance of collective decision-making has been recognized by Matsatsinis et al. [35] due to its common occurrence in the various organizations. Authors explored previous studies concerning "multicriteria decision making methodology" and "preference aggregation methods". The practical implementation of those methods and support systems for group decisions has been argued; three elements of collective decision support systems has been proposed: problem representations, preferences aggregation, and decision process organization. The study presented a new methodology based on combination of existing methods; the proposed framework included following stages:

Setup. Decision options and rules for the process are decided.

Assessment of group members' preferences. Aimed to extend participant knowledge and her/his preference awareness.

Calculation of relative utility values for each alternative. Individual for each participant in decision-making process.

Calculation of a group relative utility value for each alternative. Individual utility aggregation in form of group ranked order.

Measurement of satisfaction. Utilizes satisfaction indexes to evaluate if the decision consensus is reached, unsuccessful results may lead to either restarting procedure from first stage or stopping "without reaching consensus".

2.4.3 Non-human decision-making processes

Collective decision-making has been studied also from the perspective of systems, where decisions are made by non-human participants e.g. self-organized group of robots.

Valentini et al. [36] studied collective decision-making concerning decision accuracy and consensus time. The weighted voter model, inherited from a classical democratic election, has been proposed as a solution to the best-of-n decision problem; the decision problem has been scoped in the study to two alternative options only (sites A and B), where one has been defined as being of higher quality than the other. Authors applied the model to a self-organized system – a swarm of autonomous robots (agents) utilizing an area divided in three sites: A, B and the neutral (base) in between; agents may move free and when are done evaluating quality (scouting site A or B) come back to the neutral zone to announce discovery, other (neighbouring) agents may than change their opinion (preference) of the site. When the consensus on choosing the higher quality site is reached, the decision problem is recognized as solved.

The approach is argued as being able to reach better accuracy of decision and have a positive influence on consensus time. The key difference from a "general democratic model", which consists of random change of opinion to a neighbour opinion by a randomly picked agent, is introduction of a positive feedback mechanism to increase a chance for picking a higher quality site; technically solved by correlation of announcing state time with a predefined in the experiment site quality value [36].

2.5 Perception of Decision Fairness

Lee and Baykal [37] studied how people perceive decisions made by "mathematically proven fair division algorithms"; authors argued that low perception of fairness (only one third of participants perceived the decisions as fair) is caused by insufficient impact of social factors on the decision supporting algorithm.

Experiment presented in the study [37] were conducted using Spliddit[1], a web-based tool for "fair division problems"; fairness of decision outcome was evaluated from two perspectives: group and individual. The decision fairness has been perceived as higher in the group of participants who included discussion before the mediation has been conducted by the tool; the discussion preceding the decision has been argued to enhance responsibility for decision outcome and awareness of the priorities and needs of both their own and the others. Research confirmed that participants with higher interpersonal virtues considered discussion mediated decision as fairer than others; another hypothesis concerning the positive influence of computer programming skills on perception of fairness of algorithmic decision has not been supported by results.

Complex social factors like altruistic behaviours, the will to compromise are presented as being weaken in the system that lack "social transparency"; understood in the context as possibility to reach beyond knowledge of own satisfaction level only. Authors concluded that although social factors are difficult to implement in the decision system, their existence is important to maintain positive effects of such behaviours [37].

Lee et al. [38] explored utilization of algorithms for efficiency of goods allocation services from a human perspective; regarding how those automatic processes influence motivation and perception of decision fairness among involved people to design fair algorithms. Authors argued that the perspective of fairness, based on cultural and individual factors, depends on the context and the same algorithm may be perceived differently by multiple stakeholders. The study considered fairness as an essential value of human society and presented two models to operationalize it.

Equality. The individual differences between concerned people (here recipients of resources) are not to be considered; argued in the study as more relevant in cooperative environments, as being able to satisfy fundamental needs, enhance harmony between group members and "reduce negative socio-emotional behaviors".

Equity. Group member performance or needs are concerned in the goods allocation process.

Lee et al. [38] considered two solutions for improving perception of fairness for efficient algorithms; one regards establishing common rules that would be

[1]Spliddit - http://www.spliddit.org/ - [accessed 2022.05.04]

defined globally by involved parts, and the other allowing to locally override global rule to apply more fairly the local context.

Lee et al. [38] argued that motivation to use system is important and that the results might be unfair although algorithm itself is able to produce a fair decision. Concerning the perspective of the entire decision system transparency of the procedures run by algorithm are consider as essential for trust and usability. Another aspect discussed in the study is the need to maintain meaning of the work (decision context) and social interactions, which may be weaken due to increased automatizing of processes.

2.6 Decision Quality

Decision quality has been discussed and approached in literature from various perspectives. Kamis and Davern [39] defined the decision quality through evaluation of outcome difference from the optimal result. Authors considered two perspectives: an objective one – concerning choosing the best of alternatives, and the subjective one understood as the best suitable option for an individual participant of decision-making process. The choice of more than one perspective to define the decision quality results from the challenges with operationalizing of decision quality; authors argued that this would help to mitigate errors or bias using only one of them e.g. “similarity to expert judgment” for objective decision quality. The study argued factors that impact the quality of decision: decision-maker characteristics (knowledge), environment (presentation of alternative choices) and usability of the system supporting decisions (interactivity and information control).

Yates et al. [40] defined the decision as “a commitment to a course of action that is intended to produce a satisfying state of affairs” and describe the high-quality decision as “more satisfying” than alternatives. Authors argued the quality of decision may be perceived and defined through different perspective and conducted experiments concerning both decision outcome (product) and the decision process (difficulty) approaches. The findings from the study showed the participants’ subjective perspective on decision quality relates to the outcome (i.e., “good decisions produce good outcomes and bad decisions bad ones”) including result satisfaction which was noticed higher for easier decisions to make (concerning the decision process part of the research).

Yates et al. [40] suggested that even in the subject literature there is no agreement on the unitary definition of decision quality and previous studies refer to following aspects concerned by their authors as important:

Abstract rationality. Related with decision analysis, may refer to e.g., “consistency with axioms of probability theory”.

Accuracy. Decision maker’s evaluation of the difference between what is considered (in wider perspective) as normative and the chosen option.

Pertinent decision quality. The concept of defining decision quality through relations between decision and experience utilities i.e., between how the option is evaluated before the individual decision is made and how the chosen option is evaluated when experienced.

Decision process. Concerns with the elements of the decision process itself that would influence decision makers to be more satisfied with chosen option.

The adequate decisions may be reached by employing various decision-aid practices explored by Yates et al. [40] with concern on how the quality of decision is addressed and interpreted in each of them.

Decision Analysis. The quality of decision is understood as an integral part of the logical and rational process of decision-making. Decision Analysis aims to simplify and structure the process of making hard decisions and concerns organizing the problem e.g., in the form of decision tree, uncertainty assessment, metrics for value and utility, sensitivity analysis (influence of judgement differences on recommendations), and information processing.

Debiasing Techniques. Approaches problem similarly to Decision Analysis, but mostly concerns with judgement role and avoiding errors like overconfidence (over own judgement accuracy) and hindsight bias (too optimistic afterwards about what could have been anticipated)

Social Judgement Theory. Differs from Debiasing Techniques in terms of interpreting own judgement as consequence of perceiving elements of decision subject and assuming that they are somehow naturally connected.

General Decision Support Systems. Commonly composed of three elements responsible for: information providing, information processing, decider interactions. Characterized by effectiveness (outcome improvement through extending provided information) and efficiency (decision-making time and costs).

Group Decision Support Systems. Extender the general decision support system with "group process component" responsible for interactions between deciders; it has been argued that such systems aim to mitigate occurrences of process loss due to those interactions. Quality of decision could be interpreted from different perspectives; they might consider the time to reach the group consensus, group dynamics (interactions and confidence) or individual satisfaction of participants of the decision process.

Expert Systems. Based on qualitative principles (unlike the previously named decision-aid practices) and focus on a problem solution or performing a concrete task. The core of the system is a knowledge base and the data processing (knowledge engineering). Those systems are argued to be utilized as a part of a broader decision support system, due to ability of giving recommendations to deciders (substitute to a human expert).

Yates et al. [40] argued that utilization of decision-aid systems may be valuable if their concern apply to what is perceived as the quality of decision by those who decide (therefore may vary); although reaching consensus is acknowledged as indicating that the decision-aid systems supported a good decision. The good decision was defined in the study through five criteria:

The aim criterion. The final decision corresponds with the goals for making the decision (as perceived by a decisive body)

The need criterion. The needs of those who the decision concerns (e.g., beneficiary) are satisfied (alignment with the decision goal is not necessary).

The aggregated outcomes criterion. Accumulated results that may also include wider perspective and impact than aim and need criteria concern.

The rival option criterion. The outcomes of the final decision are better than any other option considered during the decision.

The process costs criterion. Regards minimization of costs for reaching the final decision.

The good decision process was defined by Yates et al. [40] as the one that is expected to lead to the good decision (as described above); although is was argued in the study that not all good decision processes would guarantee the good decision but based on statistic “the process tends to produce” it.

Chapter 3

Method

This chapter describes methods utilized in the thesis. It consists of two parts; the first concerns the theoretical part of the field to which this study applies and includes the procedures for the literature searches that result in the theory background, presented in the previous chapter.

The second part relates to the practical part of the study i.e., creating of decision model and simulation framework for the experiments. This part presents the theoretical background for the creation of the simulation of the social system; concerning the general application of the findings, they are presented and discussed in this chapter instead of the previous one - exploring the domain specific knowledge.

The final part of the chapter describes the elements of the social system to be included in the decision model creation and voting community representation in the simulation framework.

3.1 Literature research

The background chapter presented findings from literature search conducted during previous studies and new queries concerning collective decision making and proposed in this thesis liquid democracy model. In this section methods used for search are described.

3.1.1 Governance in OSS

The systematic literature review conducted during the IMT4894 - Advanced Project Work course has been used in this thesis to describe the context of the study. The APW study focused on answering the questions: what are the decision-making fields in OSS development, and how are the decisions in OSS projects made?

The following query which was run ACM Database Library had an initial number of results (1089, concerning “all fields” scope) decided to be sufficient:

("open source" OR "open-source" OR "OSS" OR "FLOSS") AND ("decision making" OR "decision-making" OR "governance" OR "SDG" OR "management") AND ("patch

submission" OR "patch acceptance" OR "requirements elicitation" OR "code review" OR "MCR" OR "pull request")

After the first filter (paper title, abstract and matching keywords fragments of the text were evaluated) 300 papers remained; further filter criterium concerned relevance for OSS management and excluded papers presenting management tools proposals – the final set of 224 papers established foundation for the presented results.

3.1.2 Electronically Supported Voting

Concerning that collective decisions are to be made in communities consist of remotely located contributors and stakeholders, the choice of a platform for a potential decision support system has been scoped to electronically supported voting systems; the literature search results presented in background chapter summarized the essential elements and concerns for designing a voting platform. Existing e-voting solutions has been explored in results for the “e-voting technology requirements” term search in ACM Digital Library and Semantic Scholar.

The findings, briefly introduced in background chapter, were utilized in the decision model creation and implementation of experiment framework.

Trust and participation issues in e-voting was the subject studied previously during the IMT4887 - Specialisation in Web Technology course. Due to extensive number of results for the “e-voting” term (21743 for “any field” scope), has been additionally limited to “author keyword” scope (3919 results) and further to newest published (619 from 2018 or newer). For specific subject within the voting systems, additional search parameters were used: “trust” (117 results - any fields, 2011 or newer), “usability” (108 results for author keyword, 23 of them from 2016 or newer).

The concerns presented in the background chapter have been addressed in the decision model creation process presented later in the thesis.

3.1.3 Collective decision making

Literature search regarding general collective decision making, liquid democracy has been conducted using ACM Digital Library, Research Gate and Google Scholar. The goal of this part of the study was to increase understanding of how the decisions are made, and what techniques could be employed in creating a decision support system regardless of the context.

A part of explored studies, concerning collective decisions in non-human environment influenced technical decisions made during the implementation process of the experimental framework.

3.1.4 Fairness and quality of decisions

Literature background for the operationalization of fairness has been done using ACM Digital Library. The search query (“fair AND decision” in title or abstract) produced 441 results; further filtering was applied as follow: publish date (321 since 2011), research article type (233) and publishing type – "journal" (30), and final screening for relevance produced the list of 17 articles explored for relevant information.

Findings from the study have been used in the thesis in two contexts. The first is a perception of fairness to the decision supporting system itself, here the transparency of the decision procedure and trust to decision algorithm have been evaluated as important. The second relates to the decision model i.e., concerning the choice between equality and equity, the first has been chosen as more relevant for the democratic decision making. This choice influenced the proposed liquid democracy model to aim the equal starting position for all voters (initial voting power) and equal rights regarding available behaviour during the decision process.

Initial results for decision quality queries in ACM Digital Library revealed inconsistency in the interpretation of the "quality" term and the multiple perspectives on the subject. Therefore, the extended search using Google Scholar and Research Gate has been conducted. Various definitions and the decision-aid techniques have been briefly introduced in the background chapter.

Concerning the definition of decision quality in the thesis, the accuracy of chosen option and the evaluation of individual decision before voting and the experience of chosen option during the collective decision are in focus. Regarding decision-aid practices the decision analysis aim to simplify the decision process has been evaluated as the most relevant for the thesis. Decision process aspect of the quality definition, and group decision support systems have been also considered, although this approach would be more relevant in case of future work with a concrete implementation of the voting support system in the real OSS community.

3.2 Experimental Framework

3.2.1 Simulation of social system

Building of the conceptual decision model and implementation of the experimental framework has been grounded in the concept of society simulation. Awad and Alvarez [41] argued that, although the modelling of social behaviour is challenging, society simulations are getting credit and being utilized in social science due its usefulness in cases with no obvious consensus on the solution to the problem or when the flexibility of testing new theories by various researchers is required. One of the methods is a "computer simulation laboratory for social science"; a technique that employ "agent-based models" to explore and predict social changes through combination and comparison of social theories. Authors recognized a social system as a set of social entities in a specific environment and impact on them either among themselves (relationships) or by another social force; four elements should be considered:

Composition. Set of elements (roles, organizations) characterized by a state (concerns temporal state of entity's properties).

Structure. The relationships between elements of social system. Regards additionally mechanisms to change the state of an entity.

Interactions with the environment. Concerns external entities and their relations with the social system (input and output)

Behaviour. The external perception of the social system. Behaviour of both individual entities and their composition are treated as distinctive.

In the context of this thesis, the society system represents the collective decision-making process and has been composed of voting members of an OSS project community. Decision process participants are characterized by personal attributes relevant to the collective decision calculation and voting process. The structure i.e., relationships between the elements are represented by the delegation voting mechanisms. Interactions with the environment have been interpreted as the discussed later issues concerning voting preparation (e.g., available voting options) and the consequences of decision outcome. To be able to observe the behaviour of the voters during the decision process, the experiment framework has been created.

The simulation is defined by Awad and Alvarev [41] as specific variant of modelling i.e., the process of defining the elements of the system, relations between them, and the state transitions mechanisms; the simulation is expected to interact with the environment ("generate an input/output behaviour") utilizing, in the case of computer simulations, algorithms ("rules, constraints, equations, and instructions").

Those elements have been employed during the iterative process of model and simulation framework creation; they are addressed in Section 5.2 concerning the technical design.

3.2.2 Society system specification

Awad and Alvarez [41] presented three approaches to specify a society system; two of them Differential Equations and Discrete Time are presented as traditional and long existing in the history of social science (both concern with respectively continuous and discrete "time step paradigms".

The third one Discrete Events System Specification is recognized as created lately to take advantage of computer technology. This approach employs algorithms to define states and simple behaviours of people. An event represents a change in the system and its occurrence might be a source of new events in related objects i.e., a single event (discrete "time step") may be a cause of the transition of the multiple states. Authors argued that the tempo of changes for variables should be adjusted to represent the "real human behaviour" i.e., periods of activity and inactivity [41].

The simulation framework created in this thesis is a web-based application according the third approach. It defines states and behaviours of voter representations along with the algorithms responsible for voting process in one of predefined voting models. A single voting round has been treated as a discrete event in the experiment concerning multiple (chained chronologically) decisions. The state of system elements is composed of multiple attributes that change in own pace according to the results (outcomes) of the recently made decision. The implementation of the simulation framework is presented later in Chapter 5.

3.2.3 Computational model

Awad and Alvarez [41] described a computational model as a "self-sufficient piece of analysis" which relies on the following parts:

Computational template. Target representation including necessary equations and techniques. Unlike five other parts (strictly adjusted to target) it is expected to be general and target system independent i.e., possible to apply in another context or domain.

Initial justification. Theoretical foundations and intuitions concerning modelled subject and stakeholders. Includes questions to be answered with their importance justification.

Construction assumption. Concerns handling of issues not certain under modelling e.g., employs abstractions or approximations.

Correction set. Refinement of the computational template aiming to simplify the system e.g., adjustment to less strict constraints or excluding specific relations between entities.

Interpretation. Binds the computational template to the subject of the study.

Output representation. Describe how the results are presented to stakeholders e.g., graphical interface.

3.2.4 Computational model principles in experimental framework

The simulation framework has been designed in a way that make it possible to apply also in another context i.e., beyond the decision outcome quality. Implementation of entities, voting algorithms and state transition has been conducted concerning potential generalisation and re-usability of the framework. The initial assumptions concerning the conceptual model, hypothesis regarding advantages of delegative voting has impact on the iterative process of the development of the framework aiming to be able to observe if the group makes better decisions over time. The construction assumptions included in the simulation concerns abstract representations of voters' psychological traits and decision area knowledge. The implementation of conceptual decision model has been simplified; the voting procedures has been adjusted as follows: it has been decided to behold only two options to vote on (correct and incorrect argued in literature e.g. by Valentini et al. [36] as sufficient), delegation chains have been reduced to one step only (to avoid complicated calculations within a single voting round which would require e.g., reconsideration if the voting round can be still treated as a single event). The framework is an interpretation of how the decision group behaves, and how the behaviour and decision outcome affect the voters.

3.2.5 Analysis of the results from experiments

The simulation framework has been designed to include tools for monitoring the population state and observe both partial (during the experiment) and final results. The results concerning changing states of attributes, graphical reports for calculated values with impact on final decision (e.g., voting weight), final decision outcomes (after each round and as a series) from the experiments run in the simulation are presented in a form of web-based interface; usage of analytic tools for designing the experiment are presented later in Chapter 6.

Additionally, the raw data has been extracted via API-s (serving as a data source for the website) or directly from the database via MySQL CLI.

Final results of the experiments has been analyzed using a Jupyter notebook[1], where statistical measures and diagrams has been used as a supplement to implemented analytic tools in the simulation framework; experiment's results are presented in Chapter 7.

[1] Jupyter - https://jupyter.org - [accessed 2022.05.29]

Chapter 4

Model

This chapter presents the conceptual collective decision model, that inherits from Liquid Democracy principles of vote delegation introduced in Section 2.3. The elements of the initial model representing voters, voting principles and procedures are presented in the first part of this chapter. Further, the threats to the proposed model, that were found during the literature study are discussed. The next part of the chapter presents summary of the study concerning exploration in search for the candidate technology for implementation of the conceptual decision model. The final section of the chapter concludes findings and argues the necessary adjustments to the proposed collective decision model.

4.1 Initial Decision Model

The model of the collective decision on multiple proposals could be represented as a forest data structure, where each decision proposal is a root of a tree (of support) and each voter is represented by a node. During the election time participants choose one of voting behaviour options; the state of collective decision may fluently change during a given decision time, as voters cast new votes or change their mind. When the decision time is over the largest tree will indicate the winning proposal. The final decision which each individual voter contributes to is calculated by traversing up to the root of the delegation tree; if proposal node is not finally reached the vote will count as abstain.

4.1.1 Essential elements of the voting model

- Everyone can express will to be a delegate
- It is possible to abstain and cast a vote directly or by delegation
- It is possible to change own vote
- One should know who is supported by him/her
- One should know how many supports him/her, but not who does
- One should know what the final supported issue is.

4.2 Voting Behaviour Options

The model allows one of three available individual decisions. The first is to vote directly on any available proposal, this is similar to any direct individual decision e.g. as in the classical majority voting; concerning decision data structure it means joining the node representing voter directly as a child to chosen proposal (root node). The second option represents the will to participate actively in decision while not being sure which proposal is the best or simply relying on somebody one trusts, by delegating vote; this will results in the situation where the delegate would have both own and follower's voting power to enhance his/her individual choice; data structure that represents this situation is a follower node being a child of a delegate node. Finally, there is a possibility to abstain, i.e. neither supporting any of the proposals directly nor delegating own voting power to other voter.

4.3 Delegates

It is free to express that one will be a delegate for others. Being a delegate means possibility to gather additional voting power, which could then be used on all three voting options, as described above; in case of the delegate abstain, all followers' votes will be lost; in case of choosing proposal, it will count as all followers and their followers (as long as relevant) vote directly on the proposal; the last option to delegate the decision is considered as transferring all accumulated voting power owned and from the group of supporting voters.

4.4 Flexibility

Collective decision could be treated as a changing representation of the will of the group. Multiple factors, including general ones e.g. publishing current state of decision (live results board) or individual choices e.g. following or being against majority, might influence voting decisions of the community members, during a single election round. Therefore, to represent the truly fluid state of community will, each voter might change his/her voting option within a given time, between the start and the end of a single collective decision round. Final decision might be calculated and revealed, if requested, during a specific election round; i.e. each voter may consider this factor, which will enhance flexibility e.g. if someone will definitely not vote for a specific proposal, but is uncertain about others so has chosen to be a follower.

4.5 Privacy

Model provides privacy for voters in a way that own direct choice is hidden from others. Voters anonymity may remains if required, but when one decided to be a delegate it is necessary to reveal identity in order to allow others to transfer their voting power. It is possible to introduce two profiles for each voter - first (private) for casting a vote and the second (public) for others to refer to when the voter is a delegate.

Individual decision. Voting system provides information about chosen individual voting option; it includes currently supported delegate, if relevant.

Voting power. Voter is informed of his/her current voting power, in case of being a delegate the accumulated voting power (i.e. the of the sub-tree, where delegate's node is as root) is known but the identity of followers are hidden.

Final decision. Final decision, either as calculated at the end of election round or published live during the decision time, should protect individual decisions of delegates; i.e. each follower knows own choice (delegate voter) and final choice (one of proposals or none if abstained), but not what is delegate's choice (nodes between direct delegate and root node are hidden).

4.6 Decision Model Representation

Decision model and voting process are represented as a forest data structure, as mentioned previously; Figure 4.1 shows the initial status of the collective decision, along with basic elements which will be used for explaining model details further in this section.

Each node represents a single voter marked with public identifier (e.g. "*A*"), current voting power (e.g. "*1+2*", which means "*1*" own and "*2*" from followers), and currently chosen voting option ("*null*" if abstained, another voter's public identifier if being a follower, proposal node public identifier if voting for proposal); the special proposal nodes are marked as squares and consist of public identifier only (the number shown on the side of proposal nodes indicates the current support for each proposal, i.e. accumulated voting power, which is not stored in the node).

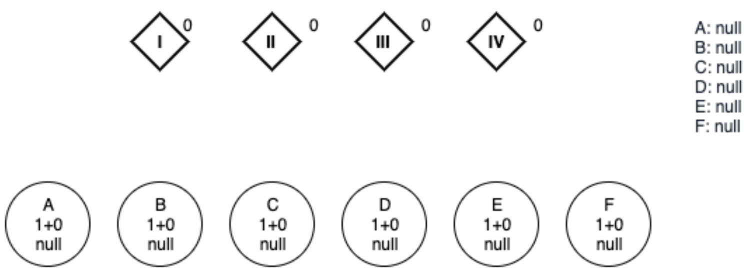

Figure 4.1: Representation of the collective decision status at election start

4.6.1 Choosing voting option

Figure 4.2 shows collective decision status while elections are in progress and the voter *C* decides to delegate vote to voter *D*. The status before the delegation is as follows: *A* and *B* supporter directly proposal *I* - giving it total support of two voters, *D* and *F* delegate their votes to *E* - giving him/her a voting power of three voters' (including own), which is further transferred to proposal *IV*; collective decision is: two votes for *I* and three for *IV*. The act of casting a delegation vote by *C* results in that the direct option in *C-node* is changed from *null* to *D*, all delegates in a chain up to the proposal node *IV* receives an extra voting power from additional follower (*C*), and an additional vote counts for proposal *IV* (which will also become a current final proposal supported by voter *C*); collective decision is: two votes for *I* and four for *IV*.

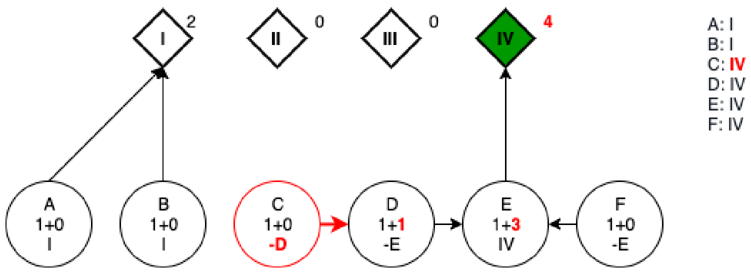

Figure 4.2: Vote delegation while an election is in progress

4.6.2 Changing voting option

The flexibility of voting gives possibility to change one's mind during the election, the process of changing individual decision (changing previously chosen voting behaviour option or keeping the behaviour option while changing action target - i.e. different delegate or different proposal) is presented on Figure 4.3; here voter *D* decides to change delegate from *E* to *A*. Updated collective decision status is represented by detaching the entire sub-tree (with *D-node* as root) from *E* and attaching it to *A*. Both voter *E* and proposal *IV* lose support of two voters (size of *D* sub-tree), voter *A* and proposal *IV* gains support of those two votes; collective decision is changed now from two votes for *I* and four for *IV* to four votes for I and two for IV. As may be noticed, a single decision change by voter *D* results in change of final supported proposal for voter *C* as listed to the right on Figure 4.3.

4.6.3 Privacy of individual decision

Voters should only have access to information necessary to participate in elections i.e. choose any of available voting options and be aware of finally supported proposal. To achieve it, the accumulated voting power of all followers is presented; as shown on Figure 4.4 additional three votes increase voting power of voter *D* to

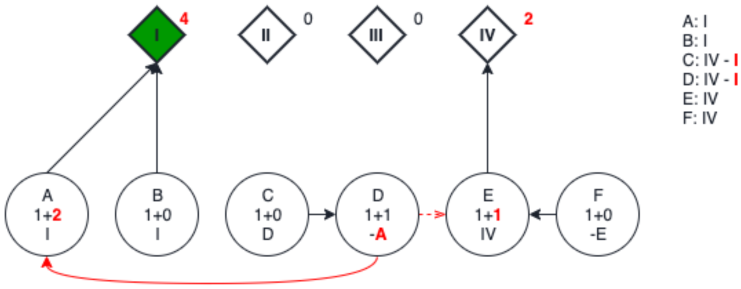

Figure 4.3: Individual decision change while an election is in progress

four (*1+3*), but identity of followers (*A*, *B* and *C*) is hidden; public identifier of voter *E* is available - so *D* can delegate all his/her voting power, but the individual choice of *E* is hidden for voter *D* - he/she may only know that by delegating vote to *E* the final proposal is *III*.

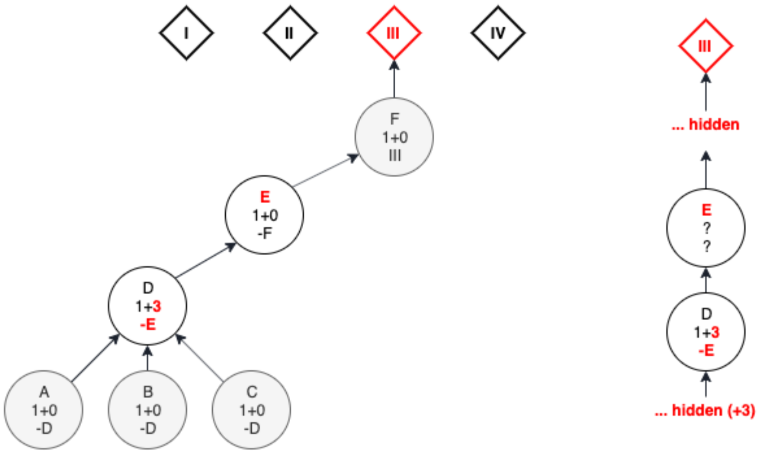

Figure 4.4: Protecting privacy of individual decision

4.7 Threats to Decision Model

This section presents threats that has been noticed during development of the delegation decision-making model and will be considered in the final model proposal described below in Section 4.9.

4.7.1 Delegation cycle

In decision models which allow multi-step delegation, i.e. a voter could be at the same time a delegate and follower, a delegation cycle might occur. Emerging delegation cycle is illustrated on Figure 4.5. The example of initial situation is that voter *D* is both a delegate for voter *C* and a follower of voter *A*; voter *C* has also announce will to be a delegate which voter *A* wants to use by changing his/her voting option to be a follower of *C*. All involved voters will after this action end up in an infinite loop of delegation; their votes will be unresolved (which in practice will mean losing a vote as in abstain option) or system has to prevent by rejecting vote change which will result in creation of delegation cycle.

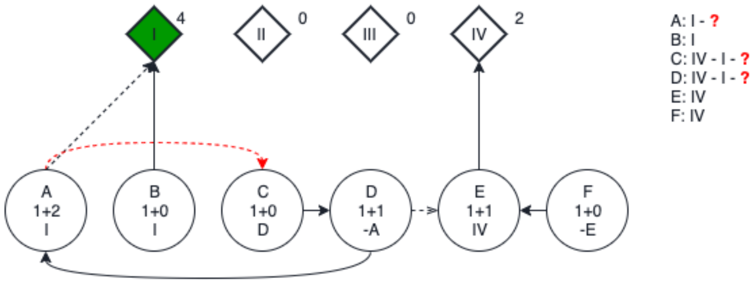

Figure 4.5: Emerging of delegation cycle

4.7.2 Late change of voting option

Potential threat of flexibility exploitation has been introduces by allowing changing already cast vote during the election. The unwanted situation occurs when a delegate change his/her vote too late for other voters (in the entire sub-tree of supporters) to react; this could be a result of willingly action to sabotage an opposite proposal by pretending supporting it or simply waiting too long to decide. A part of a problem could be solved naturally as in traditional elections, the delegate (and his/her choice) will still be valid but might lose voters' trust during the next decision round. Mechanism that could help to avoid that situation, e.g. for voters who delegates their votes but are against a specific proposal, would be time limitation for changing own vote, while voting directly on final proposal.

4.7.3 Frequent vote change for exploration purpose

Within a relatively limited by size and known community, which is about to make a collective decision, it would be possible to exploit flexibility of vote change to map delegation cycles and final decisions of voters, especially when combined with live results. Regardless of level of privacy that would be decided, this exploitation might be addressed by limiting flexibility (e.g. introducing a time delay for vote change) and replacing on-demand actions to check own finally supported proposal or live results with frequent (e.g. hourly) publishing current state of decision by the voting system.

4.8 Technology Exploration

Previous study conducted during IMT4889 Specialisation in Mobile/Wearable Technology course was dedicated to the exploration of technology candidates for decision models, i.e. those which might be used for implementation of the model itself in a given context of the collective decision in OSS projects. In this section summary of findings from this study and discussion regarding impact of tested technology on the theoretical decision-making is presented; those chosen to be introduced in final decision model will be referred to in Section 4.9.

4.8.1 Technology scope

The IMT4889 course study focused on distributed computing technologies, blockchain in particular; exploration scope was the Etherum platform and Truffle Suite[1] and Solidity[2] development environment.

4.8.2 Representation of data

The process of collective decision making could be represented by elements already available in the technology chosen for exploration. A single voter would have an account, related to a single or multiple voting round, and might be represented by a Solidity *struct* type consists of elementary types like *address* (used as identifiers) and *uint* (unsigned integer for other attributes e.g. voting power). It would be necessary to use two identifiers (*address* fields): one known for voter only (or not possible to be connected to the real voter - anonymous account), that could be treated as a private ID, to sign transactions and verify voting rights; the second one might be used while being a delegate, i.e. an optional public ID, revealed to other voters, so they may follow. Proposals would also receive an public account each so voters may refer to during the election, in the same way as delegates are chosen.

4.8.3 Voting process

Each voter's action (voting behaviour options) is registered as a *transaction* and may be implemented using *smart contracts*. Actions can be defined as active or passive; active actions relates to voters interactions with the system that are meant to introduce changes that may affect outcome, e.g. changing a delegate or voting behaviour option; passive actions are not meant to do such a change, e.g. checking currently supported proposal or live results.

[1] Truffle Suite - https://www.trufflesuite.com/docs/truffle/overview - [accessed 2021.11.27]
[2] Solidity documentation - https://docs.soliditylang.org/en/v0.7.2/ - [accessed 2021.11.27]

4.8.4 Voting power flow

Concerning elements of the initial decision model, the voting process has been explored using two, reference and currency-like, approaches. They differs in interpretation of voting power flow between voters and between voters and proposals; consequences of those two choices are as follows.

Reference approach

Data object representing a voter would have an additional *address* field which refers to current target of voter's voting power, since tested Solidity environment doesn't allow undefined values it would be set to the value of own public address instead of *null* for the default abstain behaviour. Each active voting action would toggle current voting target in the voters node. Passive actions like "What my currently supported proposal is?" are realized through traversing voters tree structure using reference to parent node. Parent nodes have no reference to child nodes (concerning privacy i.e. a delegate knows sum of followers power but not who they are) therefore calculating voting outcome would require extended computing. A basic version of collective decision-making, which assume equal voting power of each voter, might simplify calculating current (and final) decision outcome by calculating the size of a relevant support tree.

Currency approach

The currency-like approach identifies the individual voting power as a trait one can earn and spend; initial (and earned) voting power might be either kept as long one abstain or transferred to a delegate or proposal account, in a similar way as money flow between accounts in a banking context. The status of election outcome would be easy to check, the "richest proposal" wins.

4.8.5 Technology exploration findings

Practical implications of choosing reference or currency approach has been summarized in this section; findings are discussed in relation to decision model elements previously presented in Section 4.1.

Voting process

Both approaches can be implemented using *smart contracts*, assuming existence of a publicly known *address* for voting target; each active actions would be registered as a *transaction* and either set value of vote target in the node representing the voter (reference approach) or transfer available voting power from the voter node to the target node account (currency approach). Votes registered in a transaction log cannot be altered and all history is public, this will be addressed later in voting privacy section.

Voting flexibility

Implementation of voting process using reference might be more intuitive and simplified; a single transaction is necessary for individual decision change and when multiple individual decisions are made only the last one matters. Currency approach will be more difficult to handle support withdraw, e.g. when the delegate already spent received voting power; the solution to consider may be a *withdraw from contract* pattern[3] all transactions marked as pending until the withdraw is requested, but challenges remain e.g. if someone tries multiple withdraws or the necessity to analyse more data if withdraw affects longer chain of delegations (potentially up to the final proposal through multiple delegates).

Voting privacy

Transactions i explored technology are public, so each action e.g. changing vote target status or sending voting power is available to see for all. In reference approach it is possible to combine voting procedure with *blind auction*, i.e. register a hash of supported target with *salt* known for voter only, but this will make it necessary to reveal all *salts* anyway for calculating the collective decision outcome; it is possible to combine *blind auction* with voting flexibility - last vote counts. Another solution may be to use to addresses: private for own choice and public for others to delegate voting power to; an external authority would be necessary for calculating election outcome and mapping private and public accounts. Currency approach will encounter a problem with secret transfer of voting power between a public and private account (i.e. hiding an individual choice of a delegate); also regarding necessity to withdraw transferred voting power. The findings concerning transaction transparency in the explored technology introduced a question for further consideration, i.e. the value of protecting individual choice (the same way as in general/political elections) in the specific context of collective decision in OSS.

Passive actions

Transaction logs could be used to fetch information regarding current state of election and voter relevant data, e.g. last registered transaction will indicate current voting target; to be noticed, an action of reading information even if originally encrypted will reveal it to public. Calculating own voting power, in reference approach, will be difficult in secret voting (hiding followers identity) due necessity of mapping follower's private (voting power source) and public (amount of follower's voting power if he/she is also a delegate) accounts. Checking the final supported target of own vote is similar in both reference and currency approaches, and requires tracing transactions in the log; therefore choice between secret and

[3] Solidity documentation – withdraw from contract pattern – https://docs.soliditylang.org/en/develop/common-patterns.html#withdrawal-from- contracts – [accessed 2021.11.28]

transparent voting is more relevant to consider in further model improvements, also concerning avoidance of the delegation-cycle (search for cycle existence will requires revealing the secret as in other passive actions)

Vote telling

As presented previously the final (or partial) results could be calculated by analysis of either voter nodes references or voting power transactions but those operations will reveal any encrypted information if privacy of voters (and their decisions) is to be considered. If it would be decided that privacy remains until final vote telling, all delegation-cycles will be unable to avoid during the election; studies by Zhang and Zhou [42] or Brill [30] argue for counting those votes as incorrect.

4.8.6 Possible improvements

Exploring the blockchain technologies allowed to discuss potential improvements that could be made in the initial collective-decision model; in this section those which emerged in the IMT4889 course study are presented.

Transparency and privacy trade-offs

Trust and participation concerns presented in Section 2.2.1 suggested a significant social impact of transparency of decision process flow and decision outcomes. The findings from literature search (presented in Section 2.2.1) suggested that the privacy could be interpreted from two perspectives: either as a important aspect of system security (impacting voters' trust towards the system) or as an obstacle for participation (concerning live results as a social stimuli - Volkamer et al. [25]). The implementation choice should be preceded by further analysis of those elements of the initial collective decision-making model that might appear as in contrary to each other; transparency and privacy of decisions trade-offs emerge if blockchain technologies are to be used. Keeping individual decision hidden would could during election (e.g. to avoid concerns that some voter could simply follow majority choice, or if a voter would not show who or what proposal is chosen) is possible but without use of external authority all will be published by vote telling. Secret voting will also minimize benefits (e.g vote change flexibility or live results) and increase negative effects (e.g. loss of votes due to delegation-cycle) of Liquid Democracy.

Multiple choice

Implementation of data structure and voting process in explored technology allow assumption that it would be possible to extend an individual choice with a multiple voting target option; this would be more flexible and easy to realize in the currency approach, by simply allowing spending only a part of available voting power (limited to a single, decided in advance, voting power unit). Reference approach would require a set of nodes one refers to (instead of a single target). Both could be further ordered either by amount spent or a position in reference list to achieve a ranking voting trait, as e.g. Borda-Count.

Asymmetric starting position

Concerning voting power, the principles of a classical democracy assuming equality of voters. In the OSS context it is possible to notice either existence or potential benefits of an asymmetric starting position, i.e. unequal initial voting power either emerging from investments (stakeholders' share) or contributor's merits. Possible external influence on the OSS projects has been presented in Section 2.1.2. Voting power can also be treated as a value changing between election rounds and following community trust, expertise in form of reputation; Allaho and Lee [43] explored expert recommendation framework that include relevant skills evaluation and social factors.

Initial phase

Preserving privacy within a single election round would require existence of a governing authority responsible for assigning voting accounts for eligible voters (bind between a user and a voting account may also be done randomly) and protect the election from Sybil attacks (voters using multiple voting accounts to increase own voting power). Trusted election governance authority could also help to preserve voting privacy (at least for those who doesn't want to be delegates) by taking over vote telling, i.e. confirming that voter has the voting power he/she claims.

4.9 Refined model

Findings prom previous study, conducted during IMT4889 course, suggest that blockchain technology is still a candidate for decision system implementation; assuming various complexity, depending on chosen model interpretation (reference or currency), that environment is capable of realizing single elements of the theoretical model, but very challenging (if not too much effort comparing to potential gains) will be to satisfy them all at the same time in a single voting system. Moura and Gomes [44] discussed distributed architecture and *smart contract* transactions for security purpose and building trust towards decision process and outcomes due to transparency.

Concerning trade-offs between transparency and privacy discovered during the IMT4889 course study (Section 4.8.6) and the decision context (cooperative goals of collective decisions in open-source projects), it has been decided that (regardless of final choice of technology for implementation) it would be beneficial to maintain voter's identity public and his/her decision transparency as mandatory for those who wants to be delegates, and as optional for others. This results that two voter's profiles (accounts) will be necessary: one (private) for voting, and second (public) for receiving support; the public one could be dynamically created during the election after choosing "being a delegate" option. As long as there is no voting power delegated to the public account it could also be removed during election, if voter doesn't want to be marked as a potential delegate any more. The connection between voter's accounts will be published, so it will be possible for followers to trace delegate's decisions and the further flow of their voting power. This form of transparency will allow also introducing of traits as reputation or trust ranking, based on followers feedback after the collective decision round.

Voting transparency for delegates will prevent delegation cycles, but concerning other threats to decision model described in Section 4.7, limitations to voting flexibility will be introduced in form of voting change delay. The duration of delay should be decided and published during the election setup, since many factors might impact this decision, e.g. voting time window. Delay solution would also help (in addition to possible negative feedback) to prevent promoting (by a delegate) one proposal and switching to an opposite one too late for supporters to react.

Chapter 5

Simulation Framework

One of the goals for the study was to create a simulation framework that could be primary used to run experiments concerning theoretical decision model presented in previous chapter. The simulator application has been both designed and implemented in a way that others could either create and run own experiments using settings described further in this chapter or repeat experiments presented later in Chapter 6.

The scope of implemented experimental simulation and functional requirements of the simulator application are presented. Additionally, the technical design of the system, data representation of voters and algorithms used for running elections in predefined modes are explained. Finally, implemented analytic tools and the development process (choices made to reach the final state of the simulation application) are presented.

5.1 Scope of the simulator application

Simulator application is supposed to model an open-source project community, focusing on collective decisions. Voters' traits, impacting individual decision, might be randomized, but possible to control through the framework - i.e. concerning asymmetric starting position regarding either voting power or expertise in the field where the collective decision belongs to. Although in real life scenario consequences of decision may interfere with each other and profits or loses may be not know immediately, the system should be able to simulate a long term impact of using a specific decision model, i.e. evolution of community; it has been recognized as a valuable asset to the simulator and included in a form of running multiple decision rounds which depend on previous election outcomes (*learning mode*). Finally, randomized results are possible to repeat using a *performance mode*, which allows to run multiple decision rounds in parallel from the same staring point, e.g. after running *n* elections in a *learning mode*, switching to *performance mode* and running multiple versions of election number *n+1* which won't affect population traits.

5.1.1 System requirements

By confronting initial goals and purposes of the framework with first prototype versions of simulator, following functional requirements has been establish through an iterative process:

Population generator. System should be able to generate a representation of a decision makers community with support for both random and customized individual voters' traits.

Multiple decision models. System should implement both classical majority and delegation voting procedures.

Randomized individual choice. Individual voting behaviour options and correctness of choice are randomized - plausibility is based on individual voter's traits.

Simplified decision outcome. Collective decision outcome would be generalised to two proposals: correct and incorrect (based on *Expertise* trait).

Simplified behaviour options. Voting options has been limited to a single option and supporting votes only, i.e. no abstain options is available and no delegation chains.

Chained decisions. System should be able to simulate a set of decisions as if they were chained chronologically.

Learning stage. Elections run in this mode should modify population and/or voter traits to simulate decisions timeline, i.e. impact of election's outcome on future decisions.

Performance stage. No modification should be made in this mode to achieve possibility to generate multiple decision results on a given set of input data (voters' traits).

Constant *Expertise* trait. This trait is equally used in all decision models, so it has been decided that the level of *Expertise* trait should not change, regardless of running multiple chained elections.

5.2 Technical Design

This section describe the final design of simulation framework as a result of an iterative development process (presented in Section 5.3). Voting community is represented in the framework by a *Population* and *Voter* objects. A single collective

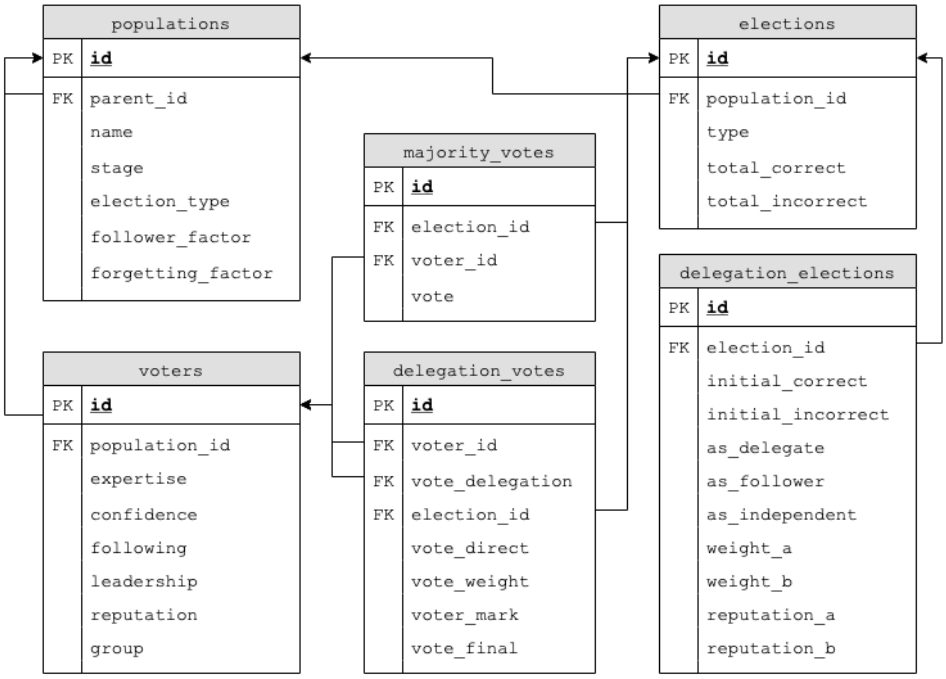

Figure 5.1: Simulation framework. Representation of data objects

decision is represented by an *Election* object; addition data required for delegation decision models are stored as object extension in *DelegationElection* objects. *MajorityVote* and *DelegationVote* represent voters' individual choices. Entities relations and fields overview are presented on Figure 5.1 and explained further in this section.

5.2.1 Population template

It has been decided that to maintain the ability to replicate experiment results it is important to introduce a *population template*, i.e. the root object of all experiments which is not meant to run elections. Child populations (in *learning mode*) are then created from the template. When the required milestone for population is reached, the child population switches to *performance mode*, which locks changes

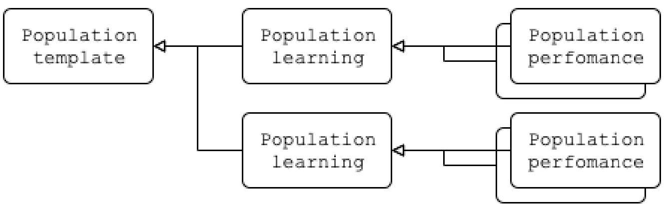

Figure 5.2: Simulation framework. Population hierarchy.

and opens for creating own child populations to run multiple versions of elections from a stable state. Population hierarchy is shown on Figure 5.2

Population attributes

Attributes as *election_type* or *stage* indicate voting model binding and current mode (*learning* or *performance*), therefore as irrelevant for population template get a *null* value. The *follower_factor* and *forgetting_factor* attributes are used under delegation election in the *learning* mode to adjust delegate's *Reputation*; i.e. may be treated as a common setting that is not directly used by a population template but by a child populations bound to delegation voting type.

Voter traits

When the population template is created (Figure 5.3 shows a create population form in simulator GUI), it is possible to adjust reputation modifiers and generate voters for entire population family (inheriting form the template).

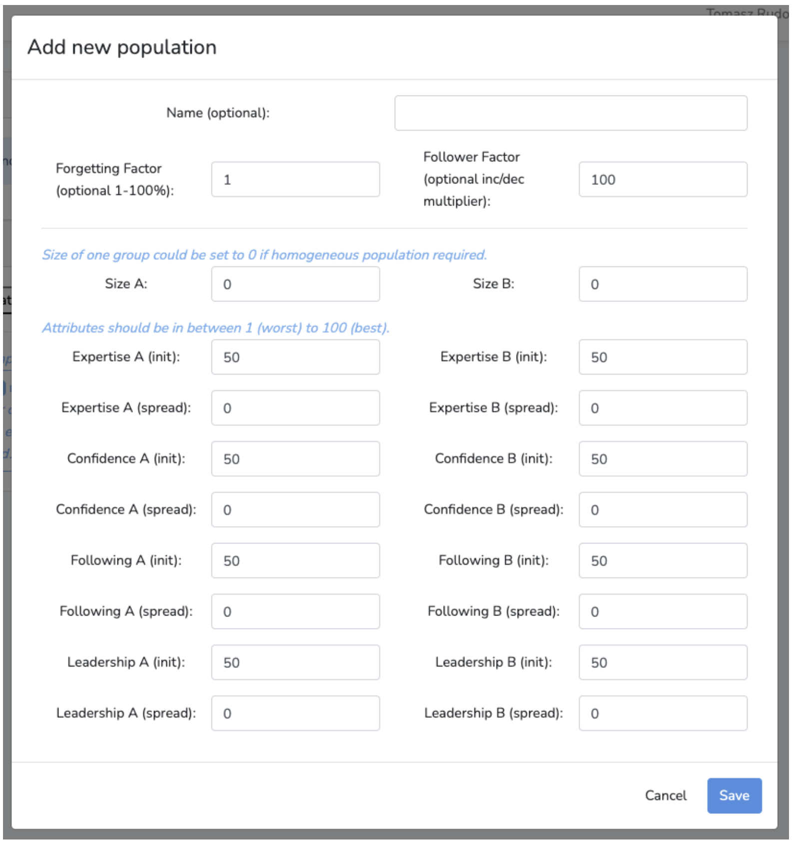

Figure 5.3: Simulation framework. Create population template (simulator GUI screenshot).

It has been decided to give a possibility of generating voters' groups (*A* and *B*) within a population that could be defined independently; this will allow to run experiments on populations consist of e.g. both regular and expert voters. Individual attributes of a *Voter* object are defined with two values, first the starting one (*init*) and the second which indicates how much the initial value may differs (*spread*). The final attributes that are possible to define may have integer values (1-100); *Expertise* indicates voter's probability of choosing a correct proposal while *Confidence*, *Following* and *Leadership* impact voting behaviour. The additional *Reputation* attribute has an initial value of *0*, which indicates no initial bias for choosing a delegate; detailed impact of population and voters attributes on decision outcome will be explained later in Section 5.2.5.

5.2.2 Populations in learning stage

The functional goal of creating a child population from a population template is to gain possibility to run a set chronologically ordered delegation elections from the same starting point; only delegation voting models relevant in the *learning* stage are available. Data representing voters will be duplicated, so the their traits will change only in the scope of the child population.

5.2.3 Populations in performance stage

It is possible to manually switch the mode of the child population from *learning* to *performance* stage; this represents the situation when the population stops evolving and it is required to run multiple, independent elections using each time the same factors. The performance stage functionality has been implemented by locking the population to a *performance* mode election type, so repeated parallel in time elections could be run. The population will also be treated as a new template, so creating a child of the child population will be available, in case more milestones in the evolution process of root population template would be required by an experiment.

5.2.4 Election

A single collective decision is represented by an *Election* object (entity relations on Figure 5.1); after each election round (regardless of decision models type) final number of correct (*total_correct*) and incorrect (*total_correct*) choices are stored. In case of delegation elections, additional data is registered: the number of direct correct/incorrect votes before applying delegation modifications (*initial_correct*, *initial_incorrect*), the number of voters taking each of the delegation model roles (*as_delegate*, *as_follower*, *as_independent*), weight of votes (*weight_a*, *weight_b*) and reputation (*reputation_a*, *reputation_b*) by a group identifier (details explained below in voting models description).

Majority vote. A single vote in majority election is stored as *MajorityVote* object; it requires only information if the choice of voter has been correct or no (*boolean* value of *vote* attribute).

Delegation vote. Additional information is stored in *DelegationVote* object: correctness of choice before (*vote_direct*) and after (*vote_final*) delegation, and attributes used by election algorithms (*vote_weight*, *voter_mark*).

5.2.5 Voting models

Simulator framework implements four voting models; the first is a classical majority election (*m-type*), the second is a delegation model not affecting voters' traits (*performance mode: d1-type*), two last ones are delegation models that differs in the scope of changes to voters' traits (*learning mode: d2-type, d3-type*).

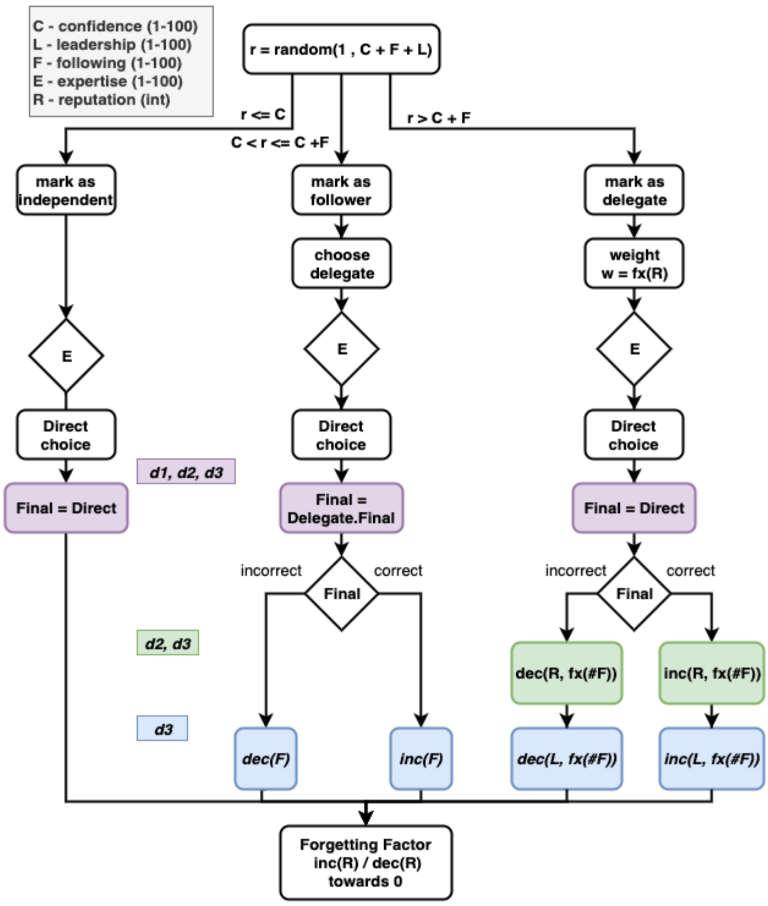

Figure 5.4: Simulation framework. Election algorithm for delegation models with options for performance (*d1*) and learning (*d2*, d3) modes

Majority voting

Majority election is implemented using a simple algorithm, that iterates through all *Voter* objects within the *Population* and registers voters' direct choices as *MajorityVote* objects. The correctness of direct choice is determined by voter's *Expertise* trait; since this parameter is defined as a probability of correct choice (in percent), the random integer in range 1 to 100 is generated and compared to the value. Majority elections run always in a *performance* mode.

Delegation voting

Algorithms for all implemented delegation models extend the common operations described here and the preceding delegation model (as they will be described below) in the terms of changes applied on voting populations; overview for process flow for the delegation models is presented on Figure 5.4.

The initial step of delegation decision is to randomize each voter's behaviour and mark them as belonging to one of three groups: independent, followers and delegates; the choice is stored by using the *voter_mark* field of *DelegationVote* object. A random integer is generated in a range from 1 to the sum of *Voter* traits: *Confidence*, *Following* and *Leadership*; those traits indicate also thresholds within the range that determine individual voter's behaviour.

Concerning delegates group, the *weighted_delegation_array* is created based on the current *Reputation* value of the delegate voter. *Reputation* is an integer and has a initial level of 0, but can rise or drop during *learning* stage; to indicate a minimal chance of being chosen it has been decided that the *Reputation* value for purpose of calculating weighted delegation chance will be adjusted to be at least 1. The final *weighted_delegation_array* consists of delegates *Id* appearing as many times as adjusted *Reputation* value.

Concerning followers group, each voter choose one of available delegates; the random number in a range from 1 to a size of *weighted_delegation_array* is generated, so delegate voter *Id* is received by index (*weighted_delegation_array[random - 1]*). Reference to a delegate is stored as *vote_delegation* attribute of *DelegationVote* object. Additionally, the number of followers is stored in delegate's vote as *vote_weight*.

The state of total sums of *Reputation* and vote weights, for each of voters' groups, are noted at the beginning of an election and saved in the *DelegationElection* object; those attributes: *weight_a*, *weight_b*, *reputation_a* and *reputation_b* are used by analytic tools explained later.

Voter's *Expertise* is tested as in the majority election and stored as *vote_direct* in *DelegationVote*; *vote_final* is set to the same value for independent and delegate groups, all followers use his/her delegate's *vote_direct* as own *vote_final*.

Delegation in performance mode - (*d1*)

The algorithm for the first delegation model (*d1*) run the basic common operations presented above. It takes into consideration current voters' traits (that could be potentially adjusted by other models), but leave them unchanged; this allows running multiple elections for analysis purpose.

Delegation in learning mode - (*d2*)

The second implemented delegation model builds on assumption that the *Reputation* of a delegate voters changes between elections (chronologically chained). Adjustment to *Reputation* is calculated by multiplying the number of followers and the *follower_factor* of the current *Population*; the product of the multiplication either increase or decrease delegate's *Reputation* depending on his/her choice (*vote_direct*). It has been decided to simulate the flow of the time between elections and therefore the *forgetting_factor* (here of current *Population*) has been introduced; it is defined as a value in a range from 1 to 100, that is interpreted as a percentage adjustment of current *Reputation* towards *0*, i.e. gain for negative and loss for positive *Reputation* at the end of election round. This model assumes that chances for the individual voting behaviour remains unchanged, but promote delegates with the history of correct decisions.

Delegation in learning mode - (*d3*)

This delegation voting model extends the previous one by adding a step that adjust the *Following* attribute of followers depending on correctness of delegate's decision, and the *Leadership* attribute of all delegates having followers (increase/decrease based on own direct choice). Those two adjustments represents the change in the voter's behaviour for next election; a voter who followed a delegate who made a correct choice will more likely be a follower next time, if delegate's choice was incorrect the probability drops; for delegates, leading others to correct/incorrect solution increase/decrease chances to choose an option to be a delegate next time.

5.2.6 User interface and analytic tools

This section describes functionality of simulator application through web interface; three basic views are presented below. More analytic tools are described later in Section 6.2.1 concerning design of the experiment and voting settings.

Population index

Home page of web application (Figure 5.5) consists of an index of all population templates; additional index filter allow to include child populations in *performance* stage. By selecting one of them summary of child populations of the template and voters traits are fetched; for child populations in *performance* stage election stats are relevant to show and as for the regular (top-level) templates the link to population template view is available. It also possible on this view to create a new population template; creation form (Figure 5.3) has been previously explained in Section 5.2.

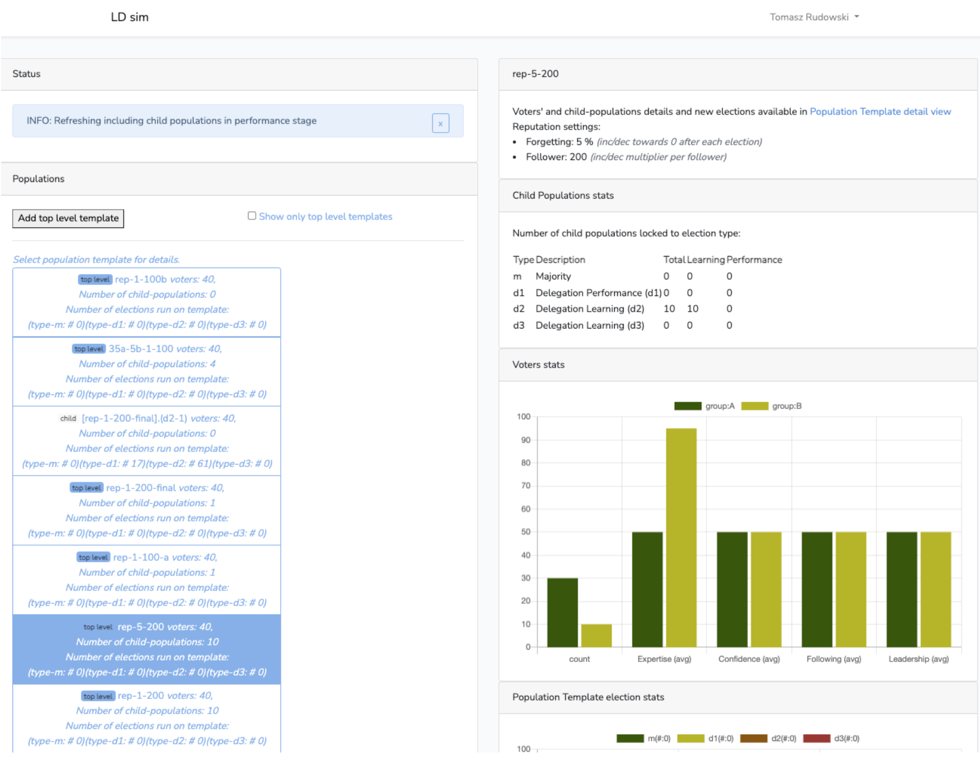

Figure 5.5: Simulation framework. Population templates index page (simulator GUI screenshot)

Population template details

The page with details concerning population template (Figure 5.6) consist of tools for creating of child populations (bound to one of predefined election types) and running a custom number of elections; this speed up the process for learning child populations. The graph on this page is used for analysis of vote weights; selector for election type and optional auto-update after a set of elections is run are included. Data concerns weights by predefined group of voters (*A* and *B* in current population) as a timeline, i.e. for chronological set of elections; values are gathered from all child populations that are bound to the selected delegation model and are limited by the minimum number of elections cross relevant child populations to include all in calculations. Multiple values can be shown on the graph: average, minimum and maximum sum of delegation weights (based on *Reputation* as described previously) by a group; additionally, ratio between group weights is calculated and how this ratio changes while more elections are run (selected graph option for *Group B - weight share* and *Group B - DIFF weight share* on Figure 5.6).

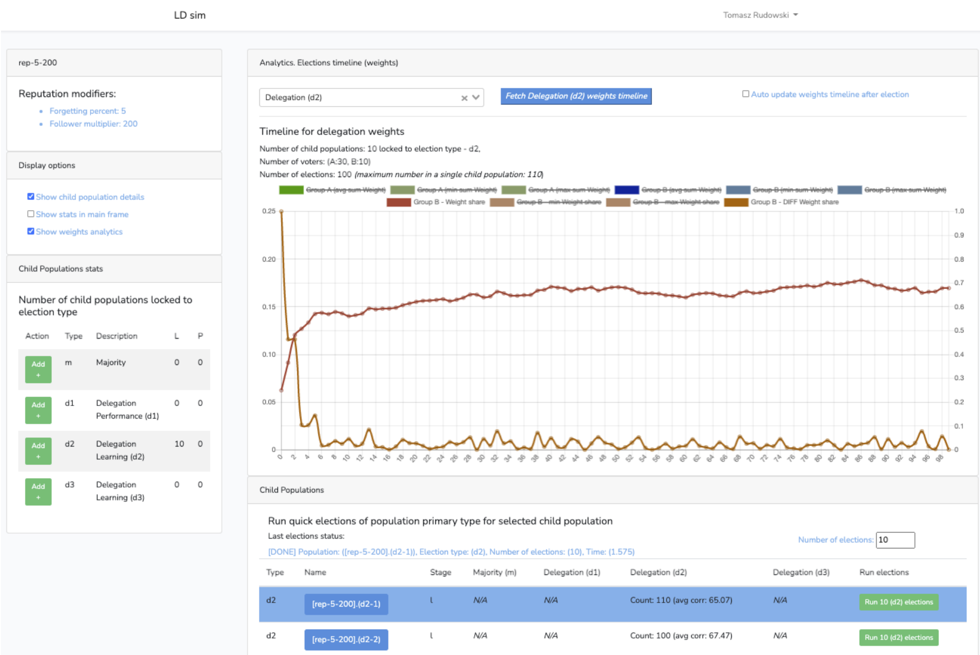

Figure 5.6: Simulation framework. Population templates details page (simulator GUI screenshot)

Child population

It is possible to run elections also using the detailed view page for a child population (Figure 5.7); this will allow to observe changes regarding final outcome and population evolution for each (or custom decided number) of elections. The page consists of multiple elements that could be shown/hidden by choosing options in a side panel; voters' data and timeline graphs might be marked for auto-updating after a set of election is run.

Population stats. Summary of voters attributes by group and elections outcome by type (hidden on Figure 5.7 screenshot).

Last election chart. Details for each election when run in a *multiple election* mode (hidden on Figure 5.7 screenshot).

Voters details. Detailed graph with data concerning current values of voters' traits and the sum of chosen behaviour option (independent, follower or delegate) so far (first graph on Figure 5.7 screenshot).

Election outcome timeline. Timeline of outcome by a selected decision model (a grey line is set to the value of population average *Expertise* for comparison reason). Data may be presented also using a moving average using additional parameter for timeline data fetching (second graph on Figure 5.7 screenshot).

Voting weights timeline. Presents changing data concerning voters' *Reputation* by group: average weight per voter, sum of group *Reputation* and weight share for group *B* (third graph on Figure 5.7 screenshot).

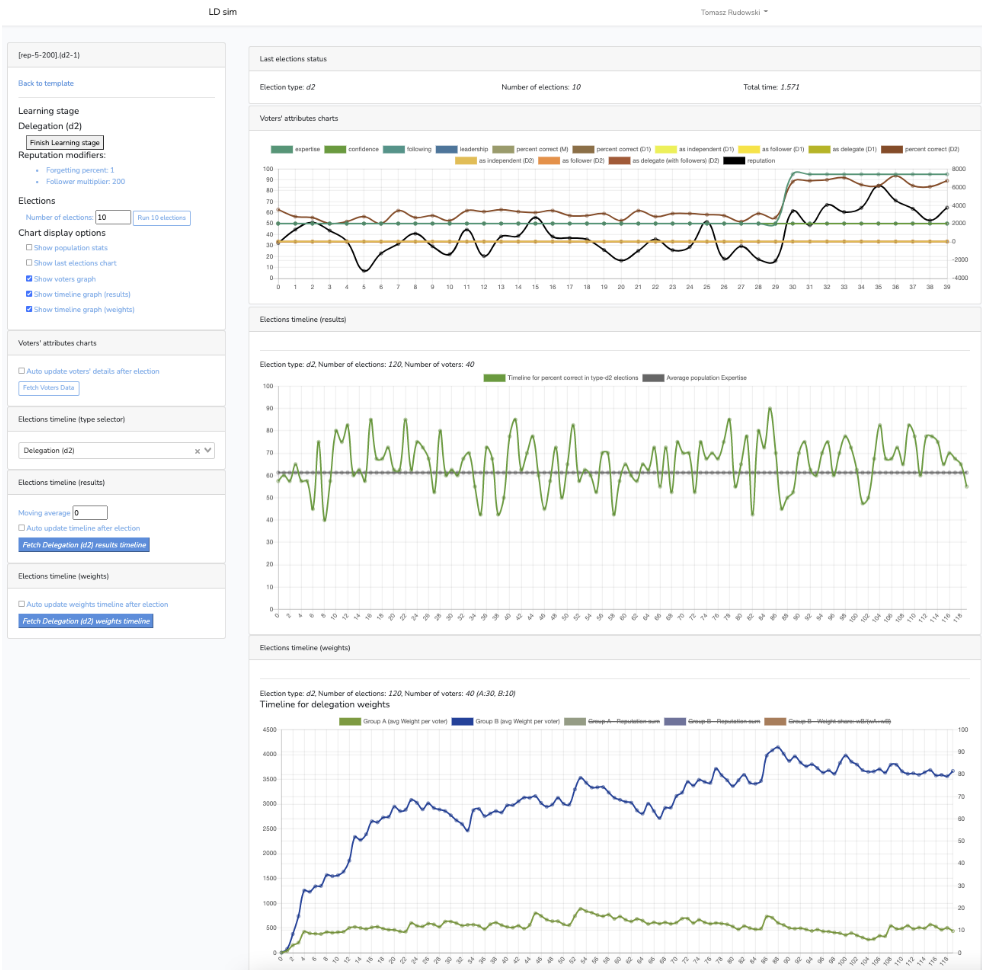

Figure 5.7: Simulation framework. Child population details details page (simulator GUI screenshot)

5.3 Development Process

The development of the simulation framework has been an iterative process. For each iteration the stable version[1] of framework has been tested, by conducting experiments. The necessary adjustments to the algorithms, tools that are relevant for monitoring of election progress and changes in the decision group itself have been either implemented or noted as the possibilities for future improvements. The technology used for implementation was a full-stack open source framework – Laravel (PHP) with MySQL database as the persistent storage.

5.3.1 Initial implementation

The first development iterations resulted in the initial version of the simulation framework including only one delegation and one majority voting models; the initial delegation model did not consider changes to voters' base attributes. Concerning future comparisons, the possibility for defining two groups of voters independently was added.

The voting algorithm was limited to a single delegation only to simplify implementation; this version of voting procedure allowed to maintain a single collective decision as a discrete event (concerning computational model principles), where the chosen behaviour and voting target options represents the final individual decision of a voter. By allowing the creation of delegation chains, the necessity of implementing control mechanisms to avoid delegation cycles would emerge; this would require concerning all individual decisions to depend on time e.g., by noting a random "vote time" to all initial voting behaviours (and/or voting targets) to simulate unwanted voting action.

Another element of the correction set in computational model was the excluding the abstain option. Although both corrections are relevant to consider in future studies e.g., experiments concerning the decision support system in a real group of decision makers, the simulation framework aims to compare the decision models a limited number of parameters. The implementation concerns that all votes are legitimate and can be counted towards the final collective decision outcome; votes that might be lost due the delegation cycles or abstain where not within a scope of the planned experiment.

The final limitation of the decisive group representation was that the individual attributes has been decided to remain unchanged to allow run multiple election rounds concerning the same setup while randomizing voter's voting behaviour and correctness of the individual (initial) choice.

5.3.2 Testing the initial version

Assuming equal starting values of the traits that have impact on voting behaviour (Confidence, Following and Leadership) for all voters, and equal Expertise values

[1]Project site with information concerning current deployment of the framework available for running experiments - https://git.gvk.idi.ntnu.no/tmr/ld-simulator [accessed:01.06.2022]

within each of two groups: experts and regulars that differ concerning this attribute, the preliminary experiments were run on the initial version of the framework. The experiments include multiple elections (decision rounds) where the percentage of correct answers in the entire decisive group (correctness of the group decision outcome) where compared between the models.

Assumed independence of Expertise and the other voter's traits caused which also remain constant through the initial experiments, caused as expected, that the collective decisions were similar in the majority and the basic delegation models. The results could be explained by the initial setup of the experiment i.e., voters' attributes. Every voter has an equal chance to decide to be a delegate and as one has an equal chance to be chosen by any of voters who declared themselves as followers. This leads to the situation that through delegation as many regular voters will contribute to results with higher delegate's expertise as experts that will use lower expertise of regular voter being a delegate. The graphical representation of the observed phenomena is presented (and compared with other implemented models) on Figure 5.8.

Randomize Expertise tests may cause a single delegation election has the higher (or lower) percentage of correct answers, but an average of multiple rounds on the same input placed the basic delegation model on the same correctness level as a classical majority voting i.e., based on an individual Expertise of voters (the reference level of collective decision correctness has been defined as an average Expertise of the population).

This basic delegation model has been then decided to be utilized as the model for the performance mode (*d1*) and the second delegation algorithm (including group evolution) was designed for the framework.

5.3.3 Learning mode

The second delegation model (*d2*) assumed that each collective decision has an impact on future decisions made in the same group and that the individual choice of a follower may be logically motivated by the perceived expertise of the available delegates.

For the implementation simplicity a single new attribute (Reputation), concerning the history of decisions made as a delegate, has been added to a voter representation in the system. The existence of this dynamic attribute could be grounded either by an objective perception that the choice made by a delegate has been correct or by an individual voter's perception concerning the correctness of the choice i.e., feedback to community that increase (or decrease) the perceived expertise of a delegate.

The default starting value of the Reputation attribute has been decided to be neutral (0), to represent the lack of delegate bias, but it could be adjusted if an experiment scenario requires it. Adjustments to Reputation has been implemented in two steps, that are included at the end of voting procedure for each election. The sum of initial Reputation values for each group are additionally stored in the election data to allow the future analysis of group evolution. The first adjustment is relevant for delegates only and concerns the product of multiplication of the number of followers and the follower factor of the population; this has been decided as representation of the correlation of reputation change with the number of voters directly affected by the delegate's decision. The value of the factor has been decided to be represented by an integer value, where the minimal value of 1 and higher factor values increase the tempo of Reputation changes. The second adjustment concerns all voters, it builds on assumption that previous actions (choices) will be forgotten in time. The adjustment to a voter's Reputation is applied either to already modified value (for delegates) or to initial one (for others), then the final state of the attribute is stored in the voter's object representation in database.

The experiments, conducted using the current state of framework, revealed that initial interpretation of forgetting factor as a value to be subtracted form positive or added to negative Reputation value had side effects. The first observed effect was the difficulty to gain reputation in the beginning of the learning stage; it was easy for delegates to lose Reputation due to a single incorrect choice or lack of followers- The second one concerns attribute's changes having a lower impact on voters with very high or very low level; voters that after many election rounds reached the Reputation (negative or positive) level that differs in order of magnitude from the constant forgetting factor would become no longer depended of it i.e., late correct or incorrect choices won't matter as much as for delegates balancing around the 0-value of the trait.

It has been decided then that the forgetting factor will be represented by an integer value which contributes to calculations as a percentage of current Reputation to be adjusted after each round; the minimal value of 0 has been treated as disabling of the forgetting factor from an experiment, and the maximum value of 100 makes the delegation voting Reputation independent e.g., gives possibility to run the third delegation model (*d3*) experiments with behaviours attribute changes as the only group evolution factor.

Introduction of this adjustments to the Reputation based on the percentage of previous value and the performance concerns for keeping calculation within integer values, led to reconsideration of values that the follower factor should have. It has been decided that the follower factor should be represented in hundreds and that the product of percentage changes due to the forgetting factor will be rounded to an integer value; this should replace the need of using float number calculations to the second decimal place.

The changes in the representation of the forgetting factor help to stop the linear rise of expert's Reputation and necessity to keep good decision and the number of followers related to a current Reputation to maintain its level (and the voting weight); the final implementation of the forgetting factor has been interpreted as being an expert and an active leader to maintain the trust of community.

The final values for Reputation factors chosen in this study will be explained later in chapter describing the environment setup for the experiment.

Concerning assumption that the individual voting behaviour is equally probable (and equal for both regular and expert groups), it is expected that size of behaviour groups and ratio between regular and expert voters in each remain as in basic delegation voting (*d1*). The vote weight based on the delegate's Reputation has been introduced to the voting procedures and opened the possibility to create a learning mode for the population, which would represent a set of decisions made after each other. Since there is an expertise level difference between regular and expert voters, it has been expected that the reputation, correlated with the previous correct choices, will increase more significantly for experts (Figure 5.8 presents expected changes to a number of voters supporting the correct option).

First experiments regarding comparison of the second delegation and classic majority voting confirmed this hypothesis so additional adjustments were made so the framework may become more realistic. Two parameters (follower and forgetting factors) has been included in the calculations concerning Reputation gain or loss after each voting; the purpose was to control the dynamic of this attribute and avoid divergence to infinity which has been interpreted as the situation when a voter has a guarantee to be or not to be chosen as a delegate.

Customization of those parameters during the setup of the population template, for the purpose of the experiments, will be discussed later in the following chapter.

Dynamic traits of voters

The third delegation model implemented in the framework was designed to address changes that may affect the individual behaviour of voters in the time perspective. The simplified representation of this phenomena assumed that a follower who contributed to a better decision thanks to delegation should have an increase chance to choose being a follower next time, and that the delegate who led to a correct decision should have an increase chance to be a delegate during the next elections.

Since the attributes are represented as integers the minimal change of the value (+/- 1) has been decided in each situation that led to the need for modifications; the minimum (1) and maximum (100) values for attributes are maintained. The final implementation of the framework includes changes to voter's traits at the end of the election round (similar to how the Reputation is modified). The Leadership attribute adjustments are relevant for delegates and Following for followers:

Increasing Leadership. Delegate has followers and the final choice was correct.

Decreasing Leadership. Delegate has no followers or has followers but the final choice was incorrect.

Increasing Following. Follower's choice was incorrect, and the delegate's choice was incorrect.

Decreasing Following. Delegate's choice was incorrect.

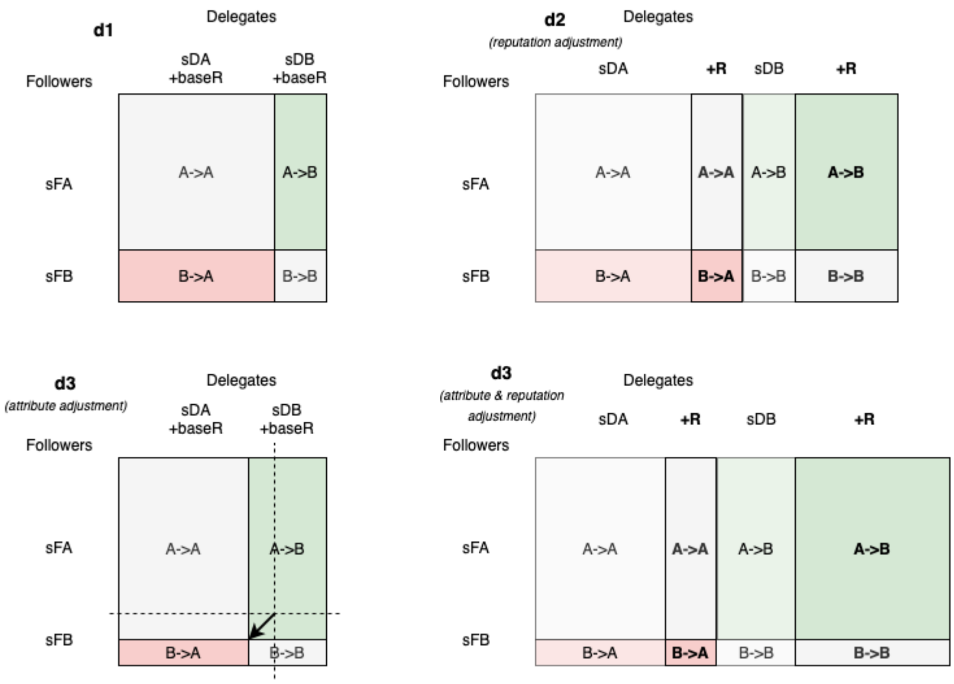

Figure 5.8: Simulation framework - Example of impact on final votes for two homogeneous subgroups of voters

5.3.4 Summary of implemented models

The graphical representation (Figure 5.8) of differences between implemented models and expected changes to final votes summarizes the goal of the development process. It regards two example groups of voters: regular (A) and experts (B) and Expertise difference only (random factor is excluded). This example assumes that all voter traits influencing behaviour are equal, so the sizes of subgroups are correlated: number of followers from group A (sFA) equals the number of delegates from A (sDA), and similar for expert group B (sFB equals sDB). Grey areas represent the same level of Expertise which results in no changes in the correctness of final vote (voter from one of the groups delegates the vote to another voter form the same group). Positive impact is expected when a regular voter delegates the vote to an expert (green area), and the negative impact when the delegation occurs in the opposite direction.

The figure representing the delegation model in the performance stage (*d1*) shows that the impact is expected to be neutral (B to A delegations are compensated with A to B delegations) assuming no initial Reputation of voters.

The second model of delegation (*d2*) which assumes the gain of Reputation is higher for the expert group (B) which is presented as the change of size of the group of delegates (+R).

The third model (*d3*) is presented in two stages that contribute to expected increase of the share of correct decisions. The first stage represents the increasing number of regular voters being a follower and experts choosing a delegate role. The second one applies the Reputation adjustments as for *d2*-model.

5.3.5 Development of GUI

The initial interface for running the experiments included the main page with the possibility to create a new population and a single population details view, where both majority and delegation (*d1*) elections could be run and compared. This was a sufficient for the first version of the framework and the representation of the constant voters attributes.

Concerning findings from preliminary experiments, especially the necessity of creating two learning and performance modes and the data structure of the social system represented in a single experiment instance, the population template view with additional options and monitoring tools has been implemented.

Later changes included mostly functionality to enhance usability and optimize the experiment process e.g., by adding the possibility to run multiple election rounds and changing the mode to performance on the child populations directly from the template view.

Last modifications to the system was the support for creating multiple generations of child populations, as presented in the Chapter 6 the design process of the final experiment.

Chapter 6

Experiment

This chapter consists of description of work done to reach settings used for experiment. The main goals for the experiment were to create a representation of a decisive group, allow it to evolve in independent directions from the same starting point through child populations, and finally determine if the delegation voting can help to reach the higher quality of decisions. The experiments aims to confirm, that the share of votes supporting the correct answer is higher than expected i.e., an average group Expertise for the classical majority voting; which would be argued as the delegation model has performed better than majority voting. Concerning the interpretation of the decision quality as by Kamis and Davern [39] i.e., the outcome difference from the optimal result. It has been decided to concern the optimal result as the support percentage for the correct answer on the level of expert group Expertise i.e., the ideal situation for choosing the right option where all decision are made by those with highest knowledge of the subject.

The first part of experiment included decisions concerning: the composition of the decisive group, the settings for learning mode of delegation voting, and the model itself. Alternative settings and potential new experiment scenarios will be discussed in chapter concerning future work.

6.1 Population Settings

Concerning representation of the community it has been decided that the Population created in the simulation should consist of two groups: regular voters and experts. Those groups refer to classification on committers and contributors presented by Eckhardt et al. [2] (Section 2.1). Although the possibility to randomize voters' attributes has been implemented (as presented previously) it was decided to exclude this factor from decision model comparison; the results of the preliminary experiments included levels of noise, inherited from voters' traits spread, that increased difficulty to determine how the model itself influence the quality of decision. Therefore two homogeneous groups, that differ in size and Expertise level were created.

6.1.1 Voting behaviour settings

To avoid bias in individual decision on which of available voting behaviour to choose, the values of Confidence, Following and Leadership were decided to have equal values for voters in both groups. The choice of initial values had a significant impact on the results from preliminary experiments conducted using the *d3* decision model which concerns that those attributes may change. Concerning that the algorithm for voting behaviour utilize the sum of those three attributes (as presented in Section 5.2.5), adjustments due to voting outcome has more impact when those attributes are closer to their minimum (1) than maximum (100) values; therefore, (regardless of choice of the model) they all have been set on the value of 50.

The initial value of Reputation has been decided to be the same for all voters and set to 0 to avoid an initial bias, as in the case of behaviour attributes.

6.1.2 Correctness of individual decision

The level of expertise for regular voters has been decided to be 50 i.e., they have equal chance to vote on correct as on incorrect option. For the purpose of the experiment the Expertise values for the second group was set on 95; the margin of 5% of incorrect choices and the difference to regular voters was assumed to be realistic and sufficient.

6.1.3 Size of subgroups

The decision concerning the size of each of the subgroups has been made as a compromise; if the group was too small, the impact of randomized choices of one of voters could have high impact on the collective decision, while too big group could potentially lead to performance issues. The group of 40 voters, consist of 30 regular and 10 expert voters, has been decided as representative.

6.2 Voting Settings

Based on findings from experiments with the third (*d3*) delegation model, as mentioned above, the decision to limit factors for the evolution of the population was made. The delegation model chosen for the experiment was *d2* i.e., concerning dynamic Reputation trait in the learning mode.

6.2.1 Reputation adjustments factors

The last parameters that have been decided were those impacting the dynamics of Reputation change over time i.e., forgetting and follower factor. Multiple population templates were created regarding differences in those two parameters only: the forgetting factor was set within the range 1 to 10, and the follower factor set to 100 and 200. Ten child populations were created or each of templates and then a hundred delegation voting (*d2*) rounds were run on each child population. Each of the settings pair were then evaluated using a weight analytic tool on population template page of application (as presented previously on Figure 5.6); the values for the delegation weight analysis are calculated as an average of all child populations (of chosen election model type) per voting round (minimum and maximum values for each round are also available).

Findings from those experiments were expected to determine both reputation factors values and the evolution time (learning mode) necessary to reach a stable state of the decisive group, consider here as the point since when the gain of Reputation due to successful delegation can no longer compensate the loss due to the forgetting factor. The stable level of delegation weight ratio between groups has been assumed as the point where population could be moved to the performance mode to evaluate the collective decision outcome for the trained population. The state was evaluated by observing the timeline of the delegation weight per group and the delegation weight share of the expert group. The parameters were expected to be suitable the already defined subject of experiment i.e., group compound and voters attributes defined as described above.

Delegation weight analysis

Two situations that were aimed to be avoided in the experiment were discovered by analysing population templates close to the edge case scenarios (additional figures from analysis in Appendix A).

The first test scenario, that assumed more weight on the follower factor (200) than on the forgetting factor (1), led to nearly linear progress of the total delegation weight for the expert group for all hundred voting rounds in the learning mode (Figure A.3). It has been interpreted as the necessity to run many more election rounds in the learning mode before switching to the performance delegation model.

Concerning factors values on the opposite side of the tested scale (follower factor: 100, and forgetting factor: 10), the ratio between the delegation weight of the groups stopped increasing linear as described above, but higher difference between rounds have been observed, especially regarding weight per group (Figure A.2).

It has been decided that since the preliminary observation of the changes in the population for the follower factor set to 100 and the forgetting factor to 5 (Figure A.1), more fine-grained tests could be run on this setting.

Impact on the experiment scenario

Ten more child populations have been created to double the amount of input to delegation weight analysis (per decision round in learning mode) and a hundred more elections have been run to double the learning time for each child population (Figure 6.1).

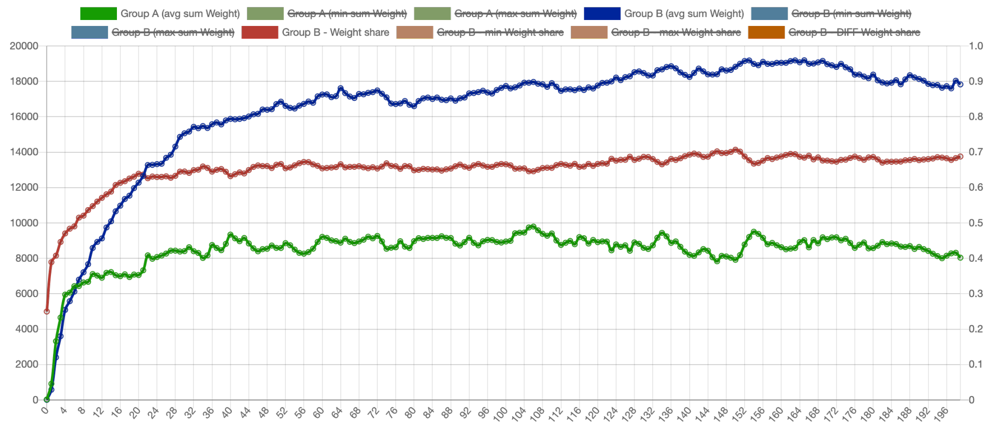

Figure 6.1: Experiment settings - Delegation weight analysis

After the analysis of the delegation weight, the three phases of population development have been observed:

The initial phase. Delegation weight of the smaller group of experts is lower than of the regular voters. Delegation weight rises quickly for both groups.

The learning phase. Experts have higher delegation weight and the ratio increases. Delegation weight rises for both groups.

The stability phase. Delegation weight changes per group are smaller. The weight ratio stabilizes.

The initial phase is assumed to be the result of the population setup: Reputation of all voters are close to initial value (0), so the weight of each group is related to its size. The sum of individual delegation weight for has been found as higher for the regular voters than for experts during the first six rounds; weight share of all experts rise from initial value of 25% (group size ratio) but remains below 50%. Single final decisions have high impact on Reputation gain or loss due to delegation while the forgetting factor has not such a high impact on the value (concerning lower values in first rounds of learning mode).

Concerning the second, learning phase, more collective decisions has been made i.e., more votes have been delegated to both experts and regular voters (based on the assumed voters' traits, one third of voters each round are followers). The weight levels are not on the high levels yet, so the regular voters as delegates have higher chances to find followers. The difference in Expertise between groups implies that experts will gain Reputation more often than regular voters. The weight share for the expert group is over 50% (i.e., experts are expected to be chosen as delegates more often than regular voters) and reaches the level of 63% after the 19th election round.

The final, stability phase, relates to the situation when the Reputation have reached levels that changes due to the delegation (following factor) no longer can compensate the passage of time (forgetting factor). The weight share for the expert group starts to be stable, during the next elections remain within the range: 63-67% for elections rounds 20-100, and 64-70% for elections rounds between 100-200

Final experiment settings

Following settings are chosen for the final experiment:

Behaviour traits. All voters in the population have all (Confidence, Following, Leadership) equal 50.

Reputation. The starting level for all voters is equal 0.

Population. Consists of 40 voters divided into two groups with the same level of Expertise in each group: 10 experts with Expertise equal 95 and 30 regular voters with Expertise equal 50.

Delegation model. The d1 delegation model (concerning Reputation adjustments only) will be used during population training (learning mode).

Reputation adjustment factors. Voting settings for Reputation adjustments has been set to: followers factor equal 100, forgetting factor equal 5.

6.2.2 Training scenario

The final experiment scenario has been decided to explore the group dynamics and the collective decision outcome in all phases observed in the preliminary experiments. This would require the multi-step learning process explained below; the necessary framework modifications, including promoting of a child population to a template population has been implemented.

The first step includes creating of child populations (first generation) from the base population template and training them for seven rounds. Afterwards the learning progress will be stopped i.e., first generation populations switch to performance mode and d1 delegation decisions will be run for each. Those populations will be then promoted to new templates for the purpose of the next phase.

Child populations (second generation) will be created for each of new templates and will be further trained from the point where their parent population stopped. The training will continue for thirteen rounds (to reach total of twenty rounds from the base population template).

Concerning higher impact of randomize behaviour and election settings in the first rounds, the first generation should consist of more child populations than the second one. It has been decided that the size of the first generation will be twenty and the size of the second generations will be ten each.

The results are to be compared with those from elections (majority and delegation in performance mode) run on the root population; is has been decided to run 500 election rounds of each control model type.

Chapter 7

Simulation Results

This chapter presents results of the experiment. It concerns the description of data produced by the simulation framework and the methods used for analysis.

7.1 Simulation Data

During the experiment a total of 223 populations were created including root population, first generation (20), second generation(200) and control populations for root (majority model - 1, d1 model - 1). Concerning forty voters in each population the total of 8.840 objects representing them were created. Each first generation population run seven *d2* elections and each second generation population run thirteen *d2* elections (giving the total of 2740 collective decisions in a learning mode i.e., 109.600 of individual votes). Concerning performance mode, each of populations in both first and second generation run ten d1 elections (giving the total of 2.200 collective decisions i.e., 88.000 individual votes).

Data concerning the performance mode elections has been gathered for further analysis. Two series of data (one for each generation) has been created using SQL queries directly in the database storing the experiment data and stored in CSV files. Data series included a calculated percentage of correct answers per election.

Two control data sets have been created by running five hundred of each majority and *d1* elections directly on the root population template i.e., using unmodified voters' attributes. This additional data increased the number of collective decisions (by 1.000) and individual votes (by 40.000) generated by the experiment. Results from control elections have been fetched to CSV files as for the two learning mode decision models.

7.2 Results Analysis

Four CSV files with experiment results and control data have been imported into a Jupyter notebook. Data has been processed using following libaries: NumPy[1],

[1]NumPy - https://numpy.org/ - [accessed 2022.05.29]

Scippy Stats[2] and Matplotlib[3].

7.2.1 Measures

This section presents measures calculated based on data series representing each of decision models. Values fetched from the database are in range 0 to 1, and represent the share of correct choices in a single majority or *d1* (performance) election.

Majority voting - root population

- Number of elections: 500
- Mean: 0,6132
- Standard Error of Mean: 0,0032
- Standard Deviation: 0,07
- Probability of outcome better than population average expertise: 50,4 %

Delegation voting (performance - *d1*) - root population

- Number of elections: 500
- Mean: 0,6124
- Standard Error of Mean: 0,0047
- Standard Deviation: 0,10
- Probability of outcome better than population average expertise: 50,0 %

Delegation voting (performance - *d1*) - first generation

- Number of elections: 200
- Mean: 0,6545
- Standard Error of Mean: 0,0081
- Standard Deviation: 0,11
- Probability of outcome better than population average expertise: 64,4 %

Delegation voting (performance - *d1*) - second generation

- Number of elections: 2000
- Mean: 0,6668
- Standard Error of Mean: 0,0023
- Standard Deviation: 0,10
- Probability of outcome better than population average expertise: 70,0 %

[2] Scipy Stats - https://docs.scipy.org/doc/scipy/index.html - [accessed 2022.05.29]
[3] Matplotlib - https://matplotlib.org/stable/index.html - [accessed 2022.05.29]

7.2.2 Distribution of decision outcomes

The analysis included generation of the distribution of decision outcomes (share of voters supporting the correct answer) for each of four voting models based on the values listed above. The relevant mean value has been used as the location and the standard deviation as the scale parameters of the normal probability density function. Calculation results have been used to create a single plot (Figure 7.1) with the comparison of the data series.

The *majority* model and the performance mode delegation (*d1*) run on the untrained (root) population shows similar outcome, which corresponds with expectations that in this case the delegation will not increase the decision quality; as presented previously in framework development process summary Section 5.3.4. Those results are considered as a control group.

As could be observed on the Figure 7.1, concerning the performance mode delegation (*d1*) employed to evaluate the outcomes of decisions made by trained populations (both first and second generations), the curves are positioned on the right side of those representing outcomes of the control group decisions, indicating the higher share of support for the correct answer.

This allow to conclude that by introducing the implemented in the simulation framework delegation model (*d2*) and allowing population to evolve while using it to make collective decisions, the higher decision quality could be achieved.

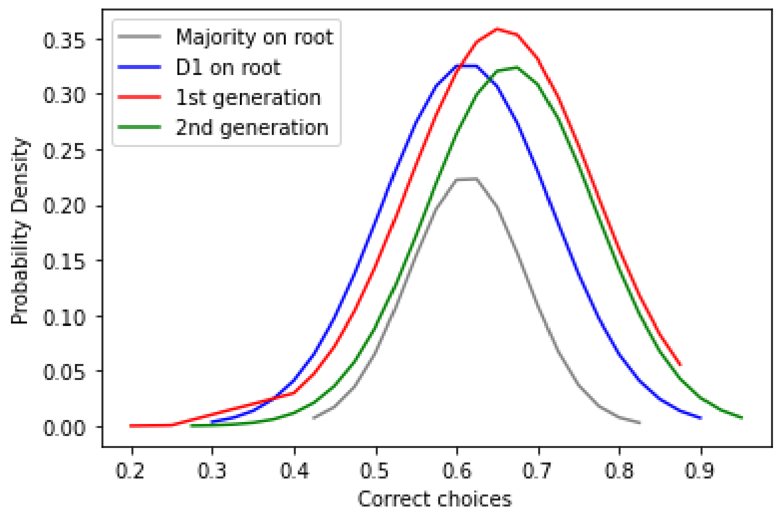

Figure 7.1: Experiment results - Outcome distribution (Jupyter notebook)

Chapter 8

Discussion

The evaluation of contribution of this thesis is presented in this chapter. The first section includes discussion regarding the simulation framework, its composition and limitations. The following sections concentrates on the results received from the conducted experiment and the experiment setup itself.

8.1 Simulator Evaluation

8.1.1 Simulation of social system

The Discrete Events System Specification has been chosen as an approach for creating a simulator framework. The decision was made based on arguments found at Awad and Alvarez [41] (in Section 3.2.3). The concept of treating a single collective decision as an event with consequences for the social system has given both a sufficient generalization regarding design and implementation, and a flexibility for state transitions after each round of elections.

Concerning the design of social system (as argued by Awad and Alvarez [41], presented in Section 3.2.1) it is believed that all four elements have been sufficiently implemented:

Composition. The elements of the social systems (entities) like a Populations and Voters with both constant and temporal states have been implemented and represent an OSS community. The Election entity with constant states represents a single (discrete) collective decision made by community; in case of learning mode the set of Elections is considered as a history timeline of collective decisions. Individual decisions are represented by a Vote entity.

Structure. Algorithms concerning relationships and mechanisms to change the states of entities has been implemented on different levels. Four main voting algorithms (majority, delegation in performance mode, two delegations in learning mode) operate on a single election round level (single event scope), while additional methods propagate relevant changes due to the event (transition of multiple states scope).

Interactions with environment. Concerning the simulator scope, the functionality of creating populations based on the requirements of the designed experiment and receiving results are considered as interaction with external entities; available in population index and population template views of the webapp.

Behaviour. The external perception of the social system i.e., behaviour of both individual entities and their composition, is available through the monitoring tools and partial results available e.g., by running elections in the learning mode in child population view of webapp.

8.1.2 Computational model

The reason for the creation of simulation framework was to gain possibility to evaluate the decision quality of the proposed delegation model.

The computational model approach has been chosen and its elements (Awad and Alvarez [41], presented in Section 3.2.3) are sufficiently implemented for the context of the study.

Computational template

The simulation framework has been designed and implemented to allow conducting custom experiments within various context of collective decision making and group dynamics.

Initial justification

The design of the simulation has included the representation of the conceptual decision model and the tools to help evaluation of the results, aiming to answer the question concerning the impact of the delegation model on the decision quality.

Construction assumption

The simulation concerns abstract representations of voters' psychological traits and decision area knowledge. It is believed, that by employing the random numbers to determine individual actions of voters and the utilization of child populations (simulate the group evolution in multiple directions) the outcomes of the collective decision quality can be generated.

Correction set

Multiple refinements of the computational template aiming to simplify has been introduced in the framework. One of them assumed that only two options will be available to vote for: correct and incorrect: this approach has been employed by Valentini et al. [36] (as described in Section 2.4.3) and argued to be a sufficient scope for the best-of-n decision problem. The choice allowed efficient design and implementation of individual choices (Expertise test) and the collective decision outcomes (percentage of voters supporting the correct option, either directly or by delegation). Another correction was allowing the delegation chains to have one step only, aiming to avoid complicated delegation algorithms within a single voting round. It is believed, that since the single delegation (assuming population evolution and active feedback) has been evaluated as giving a higher decision quality, the positive effect of the decision model would accumulate as the delegation chains grows, but the negative impact of vote loss due to delegation cycles will emerge. Although this correction limited the possibility to evaluate which (positive or negative) effect will dominate and if so under what conditions. Concerning the scope of this study the further development of the framework in this direction has been postponed to future work.

Interpretation

The framework has been developed as an interpretation of how the decision group (OSS community) behaves, and how the behaviour and decision outcome affect the voters.

Output representation

The framework includes various tools for both final decision outcomes and the temporary states of society system entities.

8.1.3 Quality of decision

It is believed, concerning the various approaches to define the decision quality, that essential aspects (Yates et al. [40], as described in Section 2.6) have been included in the simulation framework and interpreted as follows:

Abstract rationality

The algorithms for elections assume logical reasoning i.e., voters aim to reach the correct decision either based on confidence of own expertise or by delegating a vote to others (delegate reputation is treated as a perceived expertise ranking).

Accuracy

Implemented delegation models notes both own and delegate individual choices; the final supported option is compared with the (assumed) correct choice.

Pertinent decision quality

The decision quality aspect concerning the relation between decision and experience is realized by combining the results of own Expertise test and behaviour option; the perception of the experience impacts voter traits accordingly to chosen behaviour and the correctness of relevant individual decisions.

Decision process

The individual satisfaction of the decision process (impacting satisfaction with the chosen option) is difficult to measure in the simulation; the effects of the feedback (on how a voter is satisfied with the chosen option) given after decision is made is implemented as described over, but the evaluation of the process would require testing a real decision support system. Decision quality criteria as presented by Yates et al. [40] (in Section 2.6) that would indicate that the decision support system helps to reach a good decision are considered in the simulation framework:

The aim criterion

The correspondence with the decision goal is realized partially due to limitation of two choices only, although the decision round could represent e.g., a review process here considered her as a discrete value i.e., the chosen options means that it is correct or not to accept the patch. Multiple final options could be included in the framework but concerning the evaluation of decision outcome it might require implementation of alternative votes aggregation methods.

The need criterion

The representation of needs (of those who the decision concerns) is simplified in form of feedback as argued above challenges with implementation of the individual satisfaction of voters.

The aggregated outcomes criterion

By choosing to implement the chronological chaining of decisions, the wider perspective and impact has been introduced e.g., by traits modifications.

The rival option criterion

Concerning the limitation of the simulator framework i.e., two options to vote for, so the evaluation of rival options is just a single (incorrect) option. Although in the real-life support system or after inclusion of multiple voting options the necessity to decide on aggregation methods and evaluation of not chosen options will emerge.

The process costs criterion

The cost of reaching the decision from decisive body perspective is considered as not relevant in simulator framework but could be explored in future work on the voting system based on proposed model.

8.2 Results Evaluation

This section presents the evaluation of the experiment conducted using the simulation framework. The first part concerns how the setup and scenario represent the OSS community and the collective decision process employing the proposed delegation model. It will be also discussed hot the experiment results answer the supporting research question Q0.2 i.e., how Liquid Democracy could be used in OSS decision making?

8.2.1 Population setup

The findings from the literature study presented in Section 2.1 reveals, that one of the commonly used approaches to the decision making is based on some form of meritocracy. The existence of two groups (committers and contributors) that differ in roles (responsibilities) and privileges has been included in the population setup for the experiment.

The interpretation of literature findings suggest that the belonging to the inner circle of core developers or being chosen as reviewer of the proposed code changes or having is often based on the merit, authority, previous experience etc. Therefore the committers are considered as correctly represented as a smaller group with higher values of Expertise traits and that the increasing Reputation levels representing their experience and contribution to the project.

The remaining part of the OSS community i.e., regular contributors are represented by the second group within the social system of the experiment; their traits have been decided to be set to average values, which is believed to behold an unbiased tests of behaviour or chances to choose the correct option.

The creation of two homogeneous groups concerning both Expertise and psychological traits of voters impacting individual behaviour and are not considered as the ideal representation of varied community, but is a sufficient approximation that allowed to eliminate multiple factors from the experiment i.e., isolate the effect of the chosen delegation model on the evolution of the decisive group.

8.2.2 Experiment scenario

It is believed that the experiment scenario as presented in Section 6.2.2 was designed to produce satisfactory amount of data (details in Chapter 7).

The assumptions made during the process of experiments settings analysis and the process of creating the final experiment scenario (as presented previously in Chapter 6) have a limited general value. Both reputation adjustment factors and the number of collective decisions to be made by a population have been proven by preliminary experiments to be related, therefore it is concluded that each experiment setup might require analysis of delegation weight dynamics to determine the stability stage. It cannot be excluded that some specific setup may prevent the population from reaching stability (concerning delegation weight); future work could reveal such cases by running multiple, varied experiments and help to improve simulation framework.

8.2.3 Analysis methods

The evaluation of methods used to reach the results of the final experiment should consider two stages of the experiment.

The first, that have the impact on the data generated is the analysis of preliminary experiments to determine the experiment settings (as presented in Section 6.2). This task has been successfully accomplished by employing built-in data manipulation methods and graphical representations of social system state e.g., diagrams for the current average delegation weight of all child populations (bound to a specific voting model).

The second stage concerns the analysis of results after the experiment was conducted; the process and results has been presented in Chapter 7. Limitations related to currently implemented statistical methods in the simulation framework required additional effort to extract results directly from the database, but it allow also employment of MySQL engine and specific queries to generate aggregated data e.g., retrieving support percentage for the correct option based on stored individual votes. Jupyter notebook, employed in analysis of the experiment results is considered as satisfactory and allowing an extended processing of data beyond the measures presented in this thesis as relevant to determine the advantages of the explored delegation model.

8.2.4 Application of results

The results of the experiment are believed to confirm the possible application of the Liquid Democracy (concerning delegation voting interpretation as presented in Section 2.3) to the OSS governance context.

The decision field that is believed to have a highest potential of application relates to code contribution. As argued in literature (Section 2.1.4), the processes of e.g., code review include collective decisions but with the supportive role only. The tasks related to it are also believed to include more binary decisions, which corresponds with the simplified representation of the social system and the experiment.

Positive impact on community decisions regarding preparation phase cannot be confirmed directly by the results of the experiment. One of such decision field is requirements engineering, argued in literature (Section 2.1.3) as depended on social interactions and diversity, which doesn't comply with the binary outcome considered in the simulation framework and the experiment.

8.3 Decision Model Evaluation

Liquid Democracy has been interpreted in this thesis as a compromise between the direct and representative democracy that involve voting procedures related to a concept of delegating votes. The proposed model of collective decision making process (presented in Chapter 4) is believed to implement the principles of Liquid Democracy. Concerning the object of voters support, it allows to vote either directly on issue or to choose the representative person within the same collective decision process.

8.3.1 Quality and Fairness of decision

The initial assumptions regarding the quality and fairness of the voting model including the processes of vote delegation have been inspired by literature findings. The delegation voting has been recognized as increasing participation in the decision process by being more responsive and including, especially for voters without a sufficient information regarding the decision issue; the delegation process has been argued in literature as a having a supportive role for expressing own preferences.

Concerning the operationalization of decision fairness, the Equality approach has been chosen for both the thesis and the proposed model; the findings from literature study relates this model as more suitable for cooperative environments which corresponds with the perceived composition of the OSS community. Although the Equity approach was not considered as the first choice for the delegation model, some potential application have been noticed. The composition of existing OSS communities may include decisive bodies with significantly increased voting power e.g., leaning towards dictatorship influence on decision by the owner (and/or creator) of the project or attempting to have more impact on decisions e.g., external influence of various organizations or stakeholders. To address those OSS communities, the improvements of the model including voting power adjustments (e.g., asymmetric start) have been discussed. Those improvements have not been included in the conducted experiments, but the simulation framework could be adjusted in the future to include new parameters.

Concerning the decision quality, the interpretation that focus on the outcomes has been employed due to the limitations related to the experiments being conducted on the simulation of the social system. The current state of simulation framework allowed to evaluate results based on the binary (correct/incorrect) decision context. It is believed that the higher decision quality observed during the experiment can be interpolated to a multiple final options decision context, but the evaluation of the collective decision outcome may require new vote aggregation methods. The majority aggregation (implemented in simulation framework) is considered as satisfactory in the case of binary decision outcomes, but future work and experiments would be required to determine (and evaluate) alternative candidates (e.g., ranking based Borda count).

8.3.2 Trust and participation

Trust to the decision process and increased participation are believed to help the complex social factors (like altruism or mutual understanding) to emerge between decision process participant which is considered as an significant factor to voters' perception of decision process quality and fairness. Trust and participation to decision making systems have been argued in literature in relation to transparency, privacy and verifiability terms; all were considered in the process of decision model development.

Transparency and privacy

Transparency and privacy have been discussed together due to the fact that they may address the same issue from two different perspectives. The limitation of this thesis concerning the validation of the proposed model only in the experiments conducted in the simulation framework, allow to only discuss potential trade-offs between them (Section 4.8.6). It is believed, that in the context of open-source communities, promoting transparency instead of privacy has more benefits concerning the complexity of the system and the quality of outcomes e.g., could help to avoid loss of votes due to delegation cycles or limit the effects of misbehaviour (process exploitation to achieve individual goals).

8.3.3 Threats

Literature study and the iterative processes of voting model design revealed potential threats to the proposed delegation model. Due to the limitations of the simulation framework i.e., a single delegation assumption, the impact of delegation cycles could not be evaluated. Although it is believed, that promoting the transparency of the individual choices in the context of open source community decisions may limit the impact.

Experiment limitations concerned also the other threats like the manipulating the outcome by the late change of the individual decision by a delegate, or frequent changes to explore the votes cast by other voters may affect the single decision. The previously discussed trade-offs between the transparency and privacy, may limit them, but final decision concerning them could be a subject of a future study concerning the decision model prototype and the real decisive group environment.

Chapter 9

Conclusions

This chapter concludes the study by answering the research questions and proposes future work that could be done based on the findings from this thesis.

9.1 Conclusions

The exploration of decision models and implementation of the simulation framework allowed to conduct experiments which confirmed the possibility of reaching higher quality of outcomes in a collective decision model that employs the delegation voting mechanisms. The section presents conclusions regarding each of the research questions.

Q0.1 What are the current limitations of OSS decision making?

Based on the findings from literature it is concluded that although the collective decisions made by communities are involved in the various processes (e.g., code review) the decision outcomes play often the supportive role only. Decisive body e.g., a committer may ignore it completely, which can have a negative impact on other participants of the collective decision process.

External influence, existing in some projects, may be perceived as an attempt to shape the project to more structured frame. This impact can have a positive effect e.g., on the quality of the software, when playing a supportive role, but the possibility of final decision bias of e.g., commercial stakeholders could not be ignored.

Various decision fields in the software development in the open source projects context are recognized as strongly related to the group diversity and social relations. The community of engaged and cooperating contributors and committers making decisions together in a way that could be perceived as fair is considered as a significant factor for the success of the OSS project.

Another limitation of decisions made in the OSS context relates to expertise decisions e.g., code reviews. Concerning loosely coupled and decentralized communities finding a suitable expert might be challenge which has been addressed in literature by various recommendation algorithms proposals.

Q0.2 How Liquid Democracy could be used in OSS decision making?

The results of the experiments conducted in this study allow to conclude that the delegation voting could be potentially employed in the collective decisions regarding problems that are expected to be solved based on the knowledge within the decision subject area.

Assuming application of the proposed model over a time (as presented in the learning stage of the experiment) is believed to influence the evolution of an OSS community towards being based on experience and expertise. Therefore it is concluded, that introduction of the Liquid Democracy principles in the collective decisions may cause that the commonly seen meritocracy structure may be legitimated through the collective cooperation or could emerge itself if not existing.

Q0.3 What are the key architectural variables in delegative democracy models and how do they influence the decision making process?

Based on the conceptual decision model following variables have been recognized and included in both the development of the simulation framework and the process of designing the experiment:

Behaviour traits. Individual psychological traits representing the likelihood of choosing one of the voting behaviours i.e., voting individually (direct vote) or taking one of the delegation related roles (delegate, follower). Additional options like double role as a delegate and follower (delegation chain) or abstain options are considered, but not implemented in the simulation framework.

Expertise. The attribute representing the individual experience and knowledge of the voter. The experiment assumed the relevancy for a single decision area and a simple representation as the likelihood of choosing the option in the binary decision. The future study may concern the interpretation of this attribute as a compound trait e.g., a list of preferred options with weights in the case of the multiple decision options.

Reputation. The attribute representing the likelihood of being chosen as a delegate by followers. The interpretation assumed in the simulation framework (and the experiment settings) relates to the history of good decisions i.e., logical reasoning of the voters aiming to reach better decision. The universal interpretation considers social interactions and individual bias towards other community members; this aspect hasn't been introduced in the simulation framework and could be a subject of the future study.

Voting power. The variable considered in two context: the first is considered as the measure of the individual rights and influence in the decision; the other concern aggregated values through the delegation procedure aiming to calculate the final support for any of the voting options. The simulator framework the calculated support for the correct option has been limited to a number of voters choosing it directly or being a follower of the delegate who has chosen the right answer.

Q0: How can Liquid Democracy improve quality and fairness of decision-making processes in OSS development?

Concerning the results from the experiments the delegation can improve the quality of collective decision, which are based on merit principles and limited to a binary decision. It is believed, that there are the decision fields within the context of open source projects that could benefit from it e.g., patch acceptance or reviewer elicitation.

Based on literature study it is concluded, that the implementation of delegation voting may increase participation and help voters establish own preferences while lacking sufficient information about the decision issue. The perceived fairness of both decision process and the decision outcomes is believed to be positively influenced by both direct impact of delegation voting process and the assumed increase engagement in the decision process. The simulation framework and the experiment could not confirm those conclusions, so the future study concerning the real decisive group is suggested.

9.2 Future Work

The exploration of the decision fields and work with modelling of the society system simulation revealed multiple options for future studies, which due to the limitations of this thesis have been only discussed and simplified.

One of the potential directions is further exploration of the designed and implemented in simulation framework delegation models. The experiment was based on a very specific and carefully decided settings, while creating alternative scenarios for the existing framework could give a valuable input to the evaluation of the work done in this thesis or allow to reach new conclusions e.g., regarding what is the relation between group composition (number of members in each group) and the decision outcome and/or group dynamics.

The future work could consider practical improvements of existing framework e.g., new decision models, extended analytic tools.

The future study concerning creating the decision support system prototype and the research in the real decisive group, either within the same OSS scope or in a more general context, is considered as a natural continuation of the work initiated by this thesis and could be referred to the initial research plan of the study.

The initial research plan concerned the creation of such a prototype and conducting experiments within the initially explored context of the corporate open source i.e., a intern projects based on the development methodology inherited from the OSS projects. Lack of possibilities to establish the necessary relations with a company willing to explore the decision support systems, resulted in a modified plan for the research as presented in this thesis.

The future research could explore the initial goals or aim to a more generalised context of the political election applications.

Bibliography

[1] J. Cheng and J. L. C. Guo, 'Activity-based analysis of open source software contributors: Roles and dynamics,' in *Proceedings of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering*, ser. CHASE '19, Montreal, Quebec, Canada: IEEE Press, 2019, pp. 11–18. DOI: 10.1109/CHASE.2019.00011. [Online]. Available: https://doi.org/10.1109/CHASE.2019.00011.

[2] E. Eckhardt, E. Kaats, S. Jansen and C. Alves, 'The merits of a meritocracy in open source software ecosystems,' in *Proceedings of the 2014 European Conference on Software Architecture Workshops*, ser. ECSAW '14, Vienna, Austria: Association for Computing Machinery, 2014, ISBN: 9781450327787. DOI: 10.1145/2642803.2642810. [Online]. Available: https://doi.org/10.1145/2642803.2642810.

[3] S. Keertipati, S. A. Licorish and B. T. R. Savarimuthu, 'Exploring decision-making processes in python,' in *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '16, Limerick, Ireland: Association for Computing Machinery, 2016, ISBN: 9781450336918. DOI: 10.1145/2915970.2915993. [Online]. Available: https://doi.org/10.1145/2915970.2915993.

[4] J. L. Cánovas Izquierdo and J. Cabot, 'Enabling the definition and enforcement of governance rules in open source systems,' in *Proceedings of the 37th International Conference on Software Engineering - Volume 2*, ser. ICSE '15, Florence, Italy: IEEE Press, 2015, pp. 505–514.

[5] J. Herbsleb, 'Building a socio-technical theory of coordination: Why and how (outstanding research award),' in *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ser. FSE 2016, Seattle, WA, USA: Association for Computing Machinery, 2016, pp. 2–10, ISBN: 9781450342186. DOI: 10.1145/2950290.2994160. [Online]. Available: https://doi.org/10.1145/2950290.2994160.

[6] S. Chulani, C. Williams and A. Yaeli, 'Software development governance and its concerns,' in *Proceedings of the 1st International Workshop on Software Development Governance*, ser. SDG '08, Leipzig, Germany: Association for Computing Machinery, 2008, pp. 3–6, ISBN: 9781605580357. DOI: 10.

1145/1370720.1370723. [Online]. Available: https://doi.org/10.1145/1370720.1370723.

[7] J. L. C. Izquierdo and J. Cabot, 'The role of foundations in open source projects,' in *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Society*, ser. ICSE-SEIS '18, Gothenburg, Sweden: Association for Computing Machinery, 2018, pp. 3–12, ISBN: 9781450356619. DOI: 10.1145/3183428.3183438. [Online]. Available: https://doi.org/10.1145/3183428.3183438.

[8] T. Bhowmik, N. Niu, P. Singhania and W. Wang, 'On the role of structural holes in requirements identification: An exploratory study on open-source software development,' *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 3, Sep. 2015, ISSN: 2158-656X. DOI: 10.1145/2795235. [Online]. Available: https://doi.org/10.1145/2795235.

[9] L. Dabbish, C. Stuart, J. Tsay and J. Herbsleb, 'Social coding in github: Transparency and collaboration in an open software repository,' in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, ser. CSCW '12, Seattle, Washington, USA: Association for Computing Machinery, 2012, pp. 1277–1286, ISBN: 9781450310864. DOI: 10.1145/2145204.2145396. [Online]. Available: https://doi.org/10.1145/2145204.2145396.

[10] P. Heck and A. Zaidman, 'An analysis of requirements evolution in open source projects: Recommendations for issue trackers,' in *Proceedings of the 2013 International Workshop on Principles of Software Evolution*, ser. IWPSE 2013, Saint Petersburg, Russia: Association for Computing Machinery, 2013, pp. 43–52, ISBN: 9781450323116. DOI: 10.1145/2501543.2501550. [Online]. Available: https://doi.org/10.1145/2501543.2501550.

[11] R. Zilouchian Moghaddam, B. Bailey and W.-T. Fu, 'Consensus building in open source user interface design discussions,' in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12, Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 1491–1500, ISBN: 9781450310154. DOI: 10.1145/2207676.2208611. [Online]. Available: https://doi.org/10.1145/2207676.2208611.

[12] A. J. Ko and P. K. Chilana, 'Design, discussion, and dissent in open bug reports,' in *Proceedings of the 2011 IConference*, ser. iConference '11, Seattle, Washington, USA: Association for Computing Machinery, 2011, pp. 106–113, ISBN: 9781450301213. DOI: 10.1145/1940761.1940776. [Online]. Available: https://doi.org/10.1145/1940761.1940776.

[13] Z. Li, G. Yin, Y. Yu, T. Wang and H. Wang, 'Detecting duplicate pull-requests in github,' in *Proceedings of the 9th Asia-Pacific Symposium on Internetware*, ser. Internetware'17, Shanghai, China: Association for Computing Machinery, 2017, ISBN: 9781450353137. DOI: 10.1145/3131704.3131725. [Online]. Available: https://doi.org/10.1145/3131704.3131725.

[14] R. Padhye, S. Mani and V. S. Sinha, 'A study of external community contribution to open-source projects on github,' in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014, Hyderabad, India: Association for Computing Machinery, 2014, pp. 332–335, ISBN: 9781450328630. DOI: `10.1145/2597073.2597113`. [Online]. Available: `https://doi.org/10.1145/2597073.2597113`.

[15] B. Vasilescu, Y. Yu, H. Wang, P. Devanbu and V. Filkov, 'Quality and productivity outcomes relating to continuous integration in github,' in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE 2015, Bergamo, Italy: Association for Computing Machinery, 2015, pp. 805–816, ISBN: 9781450336758. DOI: `10.1145/2786805.2786850`. [Online]. Available: `https://doi.org/10.1145/2786805.2786850`.

[16] S. Zhou, B. Vasilescu and C. Kästner, 'What the fork: A study of inefficient and efficient forking practices in social coding,' in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2019, Tallinn, Estonia: Association for Computing Machinery, 2019, pp. 350–361, ISBN: 9781450355728. DOI: `10.1145/3338906.3338918`. [Online]. Available: `https://doi.org/10.1145/3338906.3338918`.

[17] S. Zhou, Ş. Stănciulescu, O. Leßenich, Y. Xiong, A. Wąsowski and C. Kästner, 'Identifying features in forks,' in *Proceedings of the 40th International Conference on Software Engineering*, ser. ICSE '18, Gothenburg, Sweden: Association for Computing Machinery, 2018, pp. 105–116, ISBN: 9781450356381. DOI: `10.1145/3180155.3180205`. [Online]. Available: `https://doi.org/10.1145/3180155.3180205`.

[18] P. C. Rigby, D. M. German, L. Cowen and M.-A. Storey, 'Peer review on open-source software projects: Parameters, statistical models, and theory,' *ACM Trans. Softw. Eng. Methodol.*, vol. 23, no. 4, Sep. 2014, ISSN: 1049-331X. DOI: `10.1145/2594458`. [Online]. Available: `https://doi.org/10.1145/2594458`.

[19] T. Hirao, A. Ihara and K.-i. Matsumoto, 'Pilot study of collective decision-making in the code review process,' in *Proceedings of the 25th Annual International Conference on Computer Science and Software Engineering*, ser. CASCON '15, Markham, Canada: IBM Corp., 2015, pp. 248–251.

[20] H. Ying, L. Chen, T. Liang and J. Wu, 'Earec: Leveraging expertise and authority for pull-request reviewer recommendation in github,' in *Proceedings of the 3rd International Workshop on CrowdSourcing in Software Engineering*, ser. CSI-SE '16, Austin, Texas: Association for Computing Machinery, 2016, pp. 29–35, ISBN: 9781450341585. DOI: `10.1145/2897659.2897660`. [Online]. Available: `https://doi.org/10.1145/2897659.2897660`.

[21] G. Gousios, A. Zaidman, M.-A. Storey and A. van Deursen, 'Work practices and challenges in pull-based development: The integrator's perspective,' in *Proceedings of the 37th International Conference on Software Engineering - Volume 1*, ser. ICSE '15, Florence, Italy: IEEE Press, 2015, pp. 358–368, ISBN: 9781479919345.

[22] K.-H. Wang, S. K. Mondal, K. Chan and X. Xie, 'A review of contemporary e-voting : Requirements , technology , systems and usability,' 2017. [Online]. Available: http://www.ikelab.net/dspr-pdf/vol1-1/dspr-paper3.pdf.

[23] L. Carter and F. Bélanger, 'Internet voting and political participation: An empirical comparison of technological and political factors,' *SIGMIS Database*, vol. 43, no. 3, pp. 26–46, Aug. 2012, ISSN: 0095-0033. DOI: 10.1145/2351848.2351851. [Online]. Available: https://doi.org/10.1145/2351848.2351851.

[24] A. Santamaria-Philco and M. A. Wimmer, 'Trust in e-participation: An empirical research on the influencing factors,' in *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, ser. dg.o '18, Delft, The Netherlands: Association for Computing Machinery, 2018, ISBN: 9781450365260. DOI: 10.1145/3209281.3209286. [Online]. Available: https://doi.org/10.1145/3209281.3209286.

[25] M. Volkamer, O. Spycher and E. Dubuis, 'Measures to establish trust in internet voting,' in *Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance*, ser. ICEGOV '11, Tallinn, Estonia: Association for Computing Machinery, 2011, pp. 1–10, ISBN: 9781450307468. DOI: 10.1145/2072069.2072071. [Online]. Available: https://doi.org/10.1145/2072069.2072071.

[26] V. Vlachokyriakos, 'Designing voting technology for participation,' in *Proceedings of the 14th Annual International Conference on Digital Government Research*, ser. dg.o '13, Quebec, Canada: Association for Computing Machinery, 2013, pp. 271–272, ISBN: 9781450320573. DOI: 10.1145/2479724.2479768. [Online]. Available: https://doi.org/10.1145/2479724.2479768.

[27] V. Cortier and J. Lallemand, 'Voting: You can't have privacy without individual verifiability,' in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18, Toronto, Canada: Association for Computing Machinery, 2018, pp. 53–66, ISBN: 9781450356930. DOI: 10.1145/3243734.3243762. [Online]. Available: https://doi.org/10.1145/3243734.3243762.

[28] S. A. Adeshina and A. Ojo, 'Towards improved adoption of e-voting: Analysis of the case of nigeria,' in *Proceedings of the 8th International Conference on Theory and Practice of Electronic Governance*, ser. ICEGOV '14, Guimaraes, Portugal: Association for Computing Machinery, 2014, pp. 403–

412, ISBN: 9781605586113. DOI: 10.1145/2691195.2691255. [Online]. Available: https://doi.org/10.1145/2691195.2691255.

[29] M. Kitsing, 'Online participation in estonia: Active voting, low engagement,' in *Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance*, ser. ICEGOV '11, Tallinn, Estonia: Association for Computing Machinery, 2011, pp. 20–26, ISBN: 9781450307468. DOI: 10.1145/2072069.2072073. [Online]. Available: https://doi.org/10.1145/2072069.2072073.

[30] M. Brill, 'Interactive democracy,' in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '18, Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 1183–1187.

[31] P. Gölz, A. Kahng, S. Mackenzie and A. D. Procaccia, 'The fluid mechanics of liquid democracy,' *ACM Trans. Econ. Comput.*, vol. 9, no. 4, Oct. 2021, ISSN: 2167-8375. DOI: 10.1145/3485012. [Online]. Available: https://doi.org/10.1145/3485012.

[32] A. Kahng, S. Mackenzie and A. D. Procaccia, 'Liquid democracy: An algorithmic perspective,' in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'18/IAAI'18/EAAI'18, New Orleans, Louisiana, USA: AAAI Press, 2018, ISBN: 978-1-57735-800-8.

[33] B. Nalebuff, 'Puzzles: Cider in your ear, continuing dilemma, the last shall be first, more,' *Journal of Economic Perspectives*, vol. 2, no. 2, pp. 149–156, 1988.

[34] D. Falessi, G. Cantone, R. Kazman and P. Kruchten, 'Decision-making techniques for software architecture design: A comparative survey,' *ACM Comput. Surv.*, vol. 43, no. 4, Oct. 2011, ISSN: 0360-0300. DOI: 10.1145/1978802.1978812. [Online]. Available: https://doi.org/10.1145/1978802.1978812.

[35] N. F. Matsatsinis, E. Grigoroudis and A. Samaras, 'Aggregation and disaggregation of preferences for collective decision-making,' *Group Decision and Negotiation*, vol. 14, no. 3, pp. 217–232, 2005, ISSN: 1572-9907. DOI: 10.1007/s10726-005-7443-x. [Online]. Available: https://doi.org/10.1007/s10726-005-7443-x.

[36] G. Valentini, H. Hamann and M. Dorigo, 'Self-organized collective decision making: The weighted voter model,' in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, ser. AAMAS '14, Paris, France: International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 45–52, ISBN: 9781450327381.

[37] M. K. Lee and S. Baykal, 'Algorithmic mediation in group decisions: Fairness perceptions of algorithmically mediated vs. discussion-based social division,' in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ser. CSCW '17, Portland, Oregon, USA: Association for Computing Machinery, 2017, pp. 1035–1048, ISBN: 9781450343350. DOI: `10.1145/2998181.2998230`. [Online]. Available: `https://doi.org/10.1145/2998181.2998230`.

[38] M. K. Lee, J. T. Kim and L. Lizarondo, 'A human-centered approach to algorithmic services: Considerations for fair and motivating smart community service management that allocates donations to non-profit organizations,' in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17, Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 3365–3376, ISBN: 9781450346559. DOI: `10.1145/3025453.3025884`. [Online]. Available: `https://doi.org/10.1145/3025453.3025884`.

[39] A. Kamis and M. Davern, 'An exploratory model of decision quality and its antecedents for category novices using multiple-stage shopping engines,' *e-Service Journal*, vol. 4, pp. 3–27, Oct. 2005. DOI: `10.1353/esj.2006.0003`.

[40] J. Yates, E. Veinott and A. Patalano, 'Hard decisions, bad decisions: On decision quality and decision aiding,' in *Emerging perspectives on Judgement and decision research*, S. L. Schneider and J. Shanteau, Eds., ISBN-10 052152718X, Cambridge University Press, Jun. 2003, pp. 1–63.

[41] G. Awad and H. Alvarez, 'A conceptual framework for the modelling and simulation of social systems,' *DYNA*, vol. 87, pp. 189–198, Mar. 2020. DOI: `10.15446/dyna.v87n212.83266`.

[42] B. Zhang and H.-s. Zhou, 'Brief announcement: Statement voting and liquid democracy,' in *Proceedings of the ACM Symposium on Principles of Distributed Computing*, ser. PODC '17, Washington, DC, USA: Association for Computing Machinery, 2017, pp. 359–361, ISBN: 9781450349925. DOI: `10.1145/3087801.3087868`. [Online]. Available: `https://doi.org/10.1145/3087801.3087868`.

[43] M. Y. Allaho and W.-C. Lee, 'Increasing the responsiveness of recommended expert collaborators for online open projects,' in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, ser. CIKM '14, Shanghai, China: Association for Computing Machinery, 2014, pp. 749–758, ISBN: 9781450325981. DOI: `10.1145/2661829.2662032`. [Online]. Available: `https://doi.org/10.1145/2661829.2662032`.

[44] T. Moura and A. Gomes, 'Blockchain voting and its effects on election transparency and voter confidence,' in *Proceedings of the 18th Annual International Conference on Digital Government Research*, ser. dg.o '17, Staten Island, NY, USA: Association for Computing Machinery, 2017, pp. 574–575,

ISBN: 9781450353175. DOI: 10.1145/3085228.3085263. [Online]. Available: https://doi.org/10.1145/3085228.3085263.

Appendix A

Additional Material

Additional material

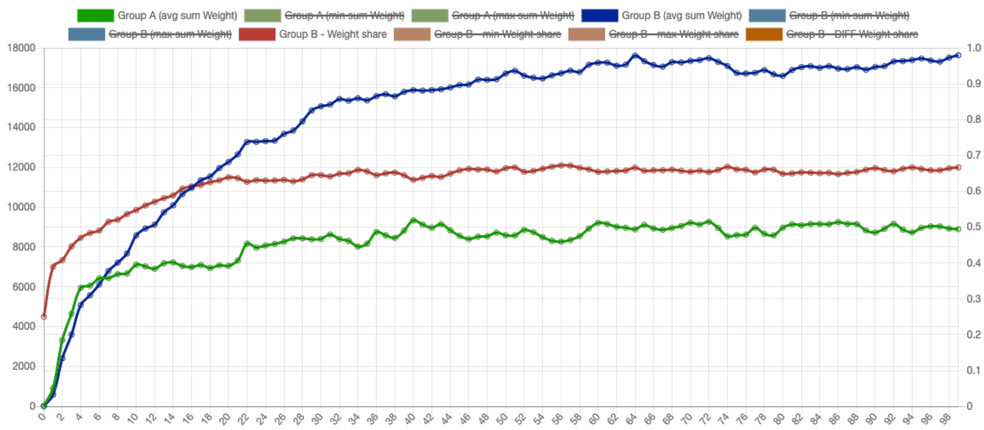

Figure A.1: Experiment settings - Delegation weight analysis for 100 rounds. Forgetting factor:5. Follower factor:100.

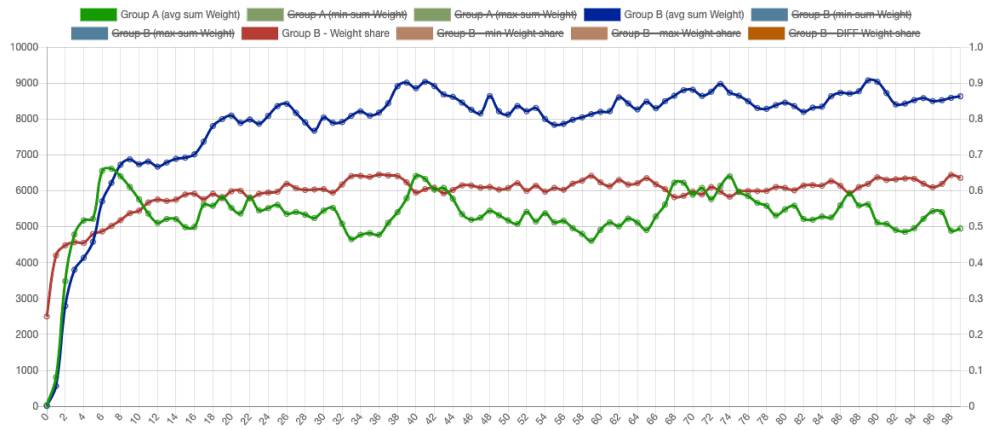

Figure A.2: Experiment settings - Delegation weight analysis for 100 rounds. Forgetting factor:10. Follower factor:100.

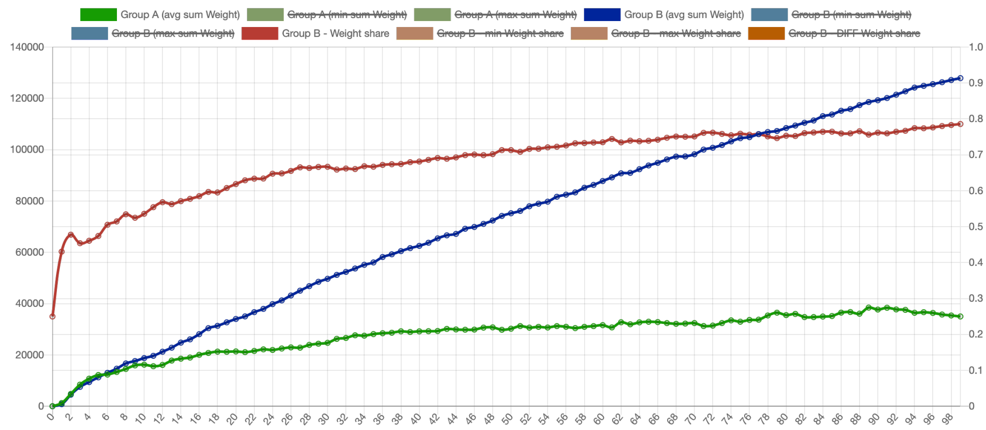

Figure A.3: Experiment settings - Delegation weight analysis for 100 rounds. Forgetting factor:1. Follower factor:100.

NTNU

Kunnskap for en bedre verden