FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Title of the Dissertation

Ângelo Miguel Tenreiro Teixeira

PREPARAÇÃO DA DISSERTAÇÃO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Prof. João Correia Lopes

January 20, 2021

Title of the Dissertation

Ângelo Miguel Tenreiro Teixeira

Mestrado Integrado em Engenharia Informática e Computação

January 20, 2021

Abstract

TODO abstract

Keywords: keyword1, Keyword2, keyword3

Acknowledgements

TODO ACKNOLEGDEMENTS

Author

"Until I began to learn to draw,
I was never much interested in looking at art."

Richard P. Feynman

Contents

List of Figures

List of Tables

Abbreviations

ADT	Abstract Data Type
ANDF	Architecture-Neutral Distribution Format
API	Application Programming Interface
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CORBA	Common Object Request Broker Architecture
UNCOL	UNiversal COmpiler-oriented Language
Loren	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed vehicula lorem commodo dui
WWW	*World Wide Web*

Chapter 1

Introduction

As of today, millions of users follow their teams' games online to keep up-to-date regarding the events of a match [3]. Some of those had a special connection to their hometown team, but since they play in way lower leagues and without much exposure, oftentimes the users end up missing information and losing the passion they once had for the hometown team.

There is a specific group of users, however, that keeps following the games of the smaller teams, and most importantly: sharing updates about them. One platform that allows users to do that, as of today, is zerozero.pt, from ZOS. This enables the most passionate fans that still watch the smaller leagues to share what is going on in the game, reporting the events and building the game's history, totally community-driven. This tool exists and is somewhat outdated, hence the opportunity to build something better.

The goal is to allow multiple users to report the events that happen in a sporting event, which show up for everyone following that match in real-time. As internet connectivity is often poor inside stadiums, the tool must allow offline work, which is synced whenever possible. This can generate many data inconsistencies, which must be handled by the tool.

This project will provide an approach to this problem and the following sections provide more details on the key-objectives of the project. In Chapter 2, a comparison with a similar project is made, as well as a *State of the Art* exploration on the multiple scopes of this project.

1.1 Offline Availability

As previously stated, internet connection in stadiums is poor most of the time. Thus, the users must have the option to interact with the application and synchronize once possible. This will obviously lead to data consistency issues (i.e. two users report a goal, changing the result to "1-0" for example, but one of them is offline, so when it finally synchronizes, the result is already "3-2" and it should not be overwritten.)

More information on this and a proposed solution will be stated in Chapter **??**.

1.2 Conflict Resolution

Another objective of the tool is to provide users with automatic conflict resolution when possible. Some strategies are depicted in the State of the Art section, in Chapter 2.2. Here, it is important to preserve the truth and the most up-to-date versions of data. In this scenario, there might not be a source of truth present to verify and validate all inputs, so other strategies must be used, such as an agreement-based implicit voting - if nobody questions a user's input, it must be true until stated otherwise.

Additionally, different strategies can be used to solve conflicts automatically, thus improving the user experience. More on the proposed solution can be found in Chapter **??**

1.3 Reputation System

The third key-objective of the application will be the reputation system. Currently, there already exists a ranking concept, as well as a "trusted" user, which is the equivalent to the maximum reputation and should be considered as the source of truth in case of conflict.

But what about the cases where two "non-trusted" users' inputs conflict, or even the case of two "trusted" users? Who should win? To resolve conflicts, an answer to these *conundrums* is fundamental. Ergo a new reputation system is required, and more details are available in Chapter **??**.

1.4 Summary

TODO summary???

Chapter 2

Background and Literature Review

This section will dive deep on previously done work related to this project. First, a Background is provided, for the reader to have context on some relevant work and information that precedes the findings present in the following sections. Second, since the goal is to develop a complete application, there will be an analysis on the specific problems, and how they have been solved in the literature. Then, there will be a comparison between a similar work and similar existing applications.

Background

TODO * web (arch) * real time (sockets?) * relevant technologies * pagerank, parallelism to reputation system

2.1 Offline Availability

2.2 Conflict resolution

TODO Creative conflict resolution in realtime collaborative editing systems TODO A Consensus-Driven Group Recommender System

2.3 Reputation System

There are multiple examples of how reputation can be used in multi-user systems and how it can affect the group dynamics. Many refer to it as a solution to "Group Recommendations", which are based in **trust** among participants whereas others mention its ability to induce cooperation. Haveliwala [10] shows how the PageRank algorithm can be personalized so that each link among nodes has a different weight, in order to express a dynamic preference among nodes. Andersen et al. [5] demonstrates multiple trust-based recommendation systems and how they comply with a set of relevant axioms. Most importantly, it shows how the aforementioned personalized PageRank (PPR) algorithm can be used to simulate a trust network among peers, by linking users with differently weighted connections. The greater the weight, the more a user trusts another, and the most likely it is for the Random Walk algorithm to choose that "path of trust". The latter also shows that PPR satisfies three out of five relevant axioms: **Symmetry**, **Positive Response**, **Transitivity**, but not Independence of Irrelevant Stuff and **Neighborhood Consensus**.

- **Symmetry.** Isomorphic graphs result in corresponding isomorphic recommendations (anonymity), and the system is also symmetric

- **Positive response.** If a node's recommendation is 0 and an edge is added to a + voter, then the former's recommendation becomes +.

- **Transitivity.** For any graph (N, E) and disjoint sets $A, B, C \subseteq N$, for any source s, if s trusts A more than B, and s trusts B more than C, then s trusts A more than C.

- **Independence of Irrelevant Stuff (IIS).** A node's recommendation is independent of agents not reach- able from that node. Recommendations are also independent of edges leaving voters.

- **Neighborhood consensus.** If a nonvoter's neighbors unanimously vote +, then the recommendation of other nodes will remain unchanged if that node instead becomes a + voter.

Dellarocas [9] shows examples of how multiple platforms handle their user reputations mechanisms. It also states that reputation systems can prevent badly intended users and deter moral hazard by acting as sanctioning devices. If the community punishes users that behave poorly and if the punishment compensates the "cheating" profit, then the threat of public revelation of a user's cheating behavior is an incentive for users to cooperate instead. It further elaborates on the reputation dynamics of a multi-user application:

- **Initial Phase** - Reputation effects begin to work immediately and in fact are strongest during the initial phase, as users try and work hard to build a reputation on themselves. Reputation effects may fail, however, when short-run users are “too cautious” when compared to the long-run ones and therefore update their beliefs too slowly in order for the long-run user to find it profitable to try to build a reputation.

- **Steady state** (or lack thereof) - In their simplest form, reputation systems are characterized by an equilibrium in which the long-run user repeatedly executes the safe action, also known as the Stackelberg action, and the user's reputation converges to the Stackelberg type (always collaborating and no cheating).

These dynamics have important repercussions for reputation systems. Dellarocas goes on to say that if the entire feedback history of a user is made available to everyone and if a collaborator stays on the system long enough, once he establishes an initial reputation for honesty will be tempted to cheat other users sometimes. In the long term, this behavior will lead to an eventual collapse of his reputation and therefore of cooperative behavior.

Bakos and Dellarocas [6] present a model for a reputation system which explores the ability of online reputation mechanisms to efficiently induce cooperation, when compared to contractual arrangements relying on the threat of litigation. It concludes that the effectiveness of a reputation mechanism in inducing cooperative behavior depends on the frequency of transactions that are affected by this mechanism, reminding that a minimum degree of participation is required before reputation can induce a significant level of cooperation. After this threshold is reached, however, the power of reputation manifests itself and high levels of cooperation can be supported.

Dellarocas [8] concludes that reputation mechanisms can induce higher cooperation and efficiency if, instead of publishing updated ratings as soon as they are available, they only update a user's public reputation every n transactions, meaning a summary statistic of a user's last ratings. In settings with noise, infrequent updating increases efficiency because it decreases the adverse consequence of artificial negative ratings. At the same time, however, infrequent updating increases a user's short-term gains from bad behavior and thus the minimum future punishment threat that can sustain cooperation.

In [4], tests were made in order to understand the reputation issues for users. These were made in Waze, a GPS-like driving assistant with crowd collaboration for road events. Even though this and zerozero.live are somewhat different, some paralelisms can be made and some gathered information still applies. They concluded that it was hard for users to recognize where the information came from, and if it was reliable at all. Furthermore, users did not care much about their reputation when submitting information (i.e. if they heard about some road event, they would publish it without verifying it), maybe this is somewhat different from our use-case of sporting events, as users are either actually watching the game, or following it from a reliable source. Additionally, when users knew the source of data, they tended to trust people in their close circle (e.g. family and friends) and the main conclusion is that the app needed to better convey the reputation of the source to let the consumers know how much they can or should trust the source.

Resnick et al. [13] elaborate about reputation systems and their generic importance on the web. It is more geared towards e-commerce examples where people investigate the reputation before interacting with each other. It mentions three important properties reputation systems should have:

- Long-lived entities that inspire an expectation of future interaction. If the entities are short-lived, their reputation matters little;
- Capture and distribution of feedback about current interactions (such information must be visible in the future);
- Use of feedback to guide trust decisions;

In the zerozero.live case, it might be hard to get expressive feedback from users regarding other users. Therefore, it is important to have some kind of implicit voting in place. Additionally, users are more inclined to express feedback when they disagree than when they agree, which means that the lack of negative feedback must be considered as some sort of positive feedback in order to balance the system. Besides, users won't see the reputation of other users beforehand in order to decide to interact or not, as they simply enter the event without knowing who is also there, so it is important that they can see the reputation, or a variant of it (i.e. some relative reputation based on the current group of users) while they are at the event (e.g. Showing it next to the user's name).

Melnikov, Lee, Rivera et al. [11] present a dynamic interaction based reputation model (DIB-RM), which is further evaluated in [14]. It presents a method to measure reputation as a function of user interaction frequency, also contemplating a reputation decay if the users stop contributing to the platform.

The aforementioned method is also present in [7], where the authors present a way to harness the "wisdom of the crowds", very much in line with what is required in zerozero.live, since there is no express authority during the event. It presents an example of a document sharing system and the approach to rank the documents based on the amount of readers, the reputation of the author, the time dynamics of reader consumption, and the time dynamics of documents contributed by the user. This last one manifests indirectly, but is still relevant: it means that if a user has less frequent readers on their documents, their reputation will decrease, so the contribution to the main document's reputation - the one they are reading now - will be smaller. Reputation values scale between 0 and 1, and it sticks to the following rules:

1. Every time a user consumes a document from an author, the author gains reputation according to:

   $$newRep = oldRep + (1 - oldRep) * repReward$$

   repReward is a constant between 0 and 1 and should consider the number of entities in the system. As the paper states: "If the number of expected consumers is in the order of hundreds or thousands, then an overly high value of *repReward* will potentially cause popular content to quickly converge towards 1 making it difficult to differentiate between similarly popular content.".

2. Every time a user consumes a document, the document gains "reputation" - meaning popularity in this case - according to the same formula of (1):

   $$newRep = oldRep + (1 - oldRep) * repReward$$

3. In order to take time dynamics into account, reputation should decrease over time, so that a "rich-get-richer" paradigm can be avoided. This is achieved by the following equation (both for users and for documents):

   $$newRep = oldRep * decayCoeff^k$$

$decayCoeff$ represents how much the reputation will change, and k is the amount of time units that have passed since the last reputation update, i.e. for a time unit of "days", k will be 0 in the first 24h, 1 in the next day, 7 in a week, and so on. This decouples the algorithm from the logistics, since the algorithm can now run in a fixed frequency, independently of the time units, and every time it re-calculates, it will give an accurate value. However, if for example the time unit is "day", and the algorithm updates every week only, there will be an offset of 6 days in which the value will be outdated.

4. Users with higher reputation matter more when calculating the document reputation changes:

$$newRep = oldRep * repConsumer * B$$

B is a constant within [0, 1] representing to what extent the user reputation $repConsumer$ will influence the document's reputation.

This system can be adapted and applied in zerozero.live if we map user inputs in an event as documents. However, we will be ranking users instead of inputs - "documents" in the analogy - even though they will also have reputation values. This will be explained in more detail in Section 3.2.1.

2.4 Similar platforms

On a basic level, this is a sporting-event following app. A similar platform would be 365scores.com [1], which offers the following of the same events in real-time, however it does not offer the community-input feature of this proposed work.

Another platform that enables live viewing of sporting events is mycujoo.tv [2]. This one enables the teams themselves to livestream the game with video, and mark specific events as they happen, so that the viewers can revisit those moments in the video. It, too, lacks the community input feature when inserting the events; it is more geared towards the clubs sharing ability, rather than the fans'.

This leaves zerozero.live as a singular app that will allow fans to contribute with the games' events in real-time, increasing engagement, which can be complemented with the enormous football-related database which can provide real-time statistics about the game.

2.5 Similar work

Castro, João [12] has developed an application with the same goal, as a Master's Thesis as well. This work, however, will not be a continuation of Castro's work or use any of its code. It will benefit solely from the insights it can give, being a work with the same goal, with high importance in terms of literature review.

Castro's work focused mainly on the reputation system as a conflict resolution strategy (i.e. the user with the most reputation wins an argument over the user with less reputation). While this is a valid approach to start with, in the real world it has a lot of limitations such as highly-reputed users abusing their power. Further discussion about reputation systems in the literature is shown in Section 2.3. This work, however, intends to apply a different technique that, while harnessing the advantages of a reputation system, aims to prevent the problems that could arise when used by real users. One of them would be using different conflict resolution strategies, depending on the conflict strategy (i.e. a conflict in the game score is way more important and thus cannot be solved by blindly applying a reputation comparison than, say, a mistake on the player substitution). The way of solving conflicts in terms of User Experience is also a matter of study, as we don't want to fact-check every user input and disturb every other user experience with it, will at the same time guaranteeing the most true story possible. Finally, this work will have an "Offline Availability" goal as well, which is of great relevance in the real world, as the connectivity is not always the best, and many consistency problems result from it thus, it's only fair that it is included in the areas of study regarding this application.

Chapter 3

Problem Statement

problem section intro here

3.1 Problem Definition

TODO US here

3.2 Proposed Solution

TODO arch + techs here mockups?

3.2.1 Reputation System

As was mentioned in the Literature Review (Section 2.3), an effective method to achieve a fair reputation system, which takes into account the time dynamics of user interactions as well as their current reputation, is to implement a personalized PageRank algorithm, which takes into account the reputation of users when calculating vouching or invalidation, in order to achieve a weighted voting system so as to provide long-term reputable users with a prize for their good behavior. Recalling the system present in [7], there are 4 rules involved in adapting the system to our use case:

1. Every time a user consumes a document from an author, the author gains reputation;
2. Every time a user consumes a document, the document gains "reputation" (i.e. popularity);
3. In order to take time dynamics into account, reputation should decrease over time, so that a "rich-get-richer" paradigm can be avoided (both for users and for documents);
4. Users with higher reputation matter more when calculating the document reputation changes;

With this in mind, I propose the following rules to adapt this to our scenario:

- Every time a user agrees with an input, he will improve the input's reputation according to rules 2 and 4;

$$newInputRep = oldInputRep + (1 - oldInputRep) * maxRepReward * userRep * userRepInfluence \quad (3.1)$$

- Every time a user disagrees with an input (either by inputting a real-conflicting input or reporting as false/inaccurate) he will worsen the input's rep according to rules (2's reverse) and 4;

$$newInputRep = oldInputRep * (1 - maxRepPunishment * userRep * userRepInfluence) \quad (3.2)$$

- Every time a user submits a falsely-conflicting input, meaning that both users submitted the same information resulting in duplicated information, it should act as an explicit agreement with the other user's input, so it should count more, according to an *explicitAgreementBonus* constant, which must be greater than zero to achieve the bonus effect;

$$newInputRep = oldInputRep + ((1 - oldInputRep) * repReward * userRep * userRepInfluence) * (1 + explicitAgreementBonus) \quad (3.3)$$

- The user gains reputation according to the average of its inputs' reputations. Only takes into account the latest inputs, referring to the last event which will trigger the reputation update;

$$newUserRep = oldUserRep + (1 - oldUserRep) * \frac{\sum inputRep}{numInputs} \quad (3.4)$$

- Each user has a reputation decay according to rule 3, the time unit should be 1 week since there's at least one relevant game per week. This prevents users that generate a lot of inputs in a single game to enjoy their reputation boost for many more games, since they need to be consistent every week: it matters more if they make an input every week than 20 inputs once every 2 or more weeks. This decay is on a higher level than the events, creating 20 inputs in an event is roughly the same as 1 input in an event (since the football events last around 90 minutes)

$$newUserRep = oldUserRep * decayCoefficient^{timeSinceLastUpdate} \quad (3.5)$$

- The reputation values are updated at the end of each event, according to the event's history.

3.3 Methodology

TODO mention scrum like stuff, in order to be able to gather feedback with ready products in between sprints (vs kanban which would be more continuous)

3.4 Planning

TODO GANTT here

Chapter 4

Conclusions and Future Work

conclusions intro

4.1 Expected Results

something else?

Appendix A

Loren Ipsum

References

[1] 365scores. Available at https://www.365scores.com/pt/pages/about.

[2] MyCujoo. Available at https://mycujoo.tv/en/about-us.

[3] The Changing Profile of Sports Fans Around The World. Available at https://www.facebook.com/business/news/insights/the-changing-profile-of-sports-fans-around-the-world, 2019.

[4] Orlando P. Afonso, Luciana C.de C. Salgado, and José Viterbo. User's understanding of reputation issues in a community based mobile app. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9742, pages 93–103. Springer Verlag, 2016.

[5] Reid Andersen, Christian Borgs, Jennifer Chayes, Uriel Feige, Abraham Flaxman, Adam Kalai, Vahab Mirrokni, and Moshe Tennenholtz. Trust-based recommendation systems: An axiomatic approach. In *Proceeding of the 17th International Conference on World Wide Web 2008, WWW'08*, pages 199–208, 2008.

[6] Yannis Bakos and Chrysanthos Dellarocas. Cooperation Without Enforcement? A comparative analysis of litigation and online reputation as quality assurance mechanisms. Technical report, 2003.

[7] Elizabeth M. Daly. Harnessing wisdom of the crowds dynamics for time-dependent reputation and ranking. In *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining, ASONAM 2009*, pages 267–272, 2009.

[8] Chrysanthos Dellarocas. How often should reputation mechanisms update a trader's reputation profile? *Information Systems Research*, 17(3):271–285, 2006.

[9] Chrysanthos Dellarocas and R. Smith. Reputation Mechanisms. 2005.

[10] Taher H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, jul 2003.

[11] Almaz Melnikov, Jooyoung Lee, Victor Rivera, Manuel Mazzara, and Luca Longo. Towards Dynamic Interaction-Based Reputation Models. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, volume 2018-May, pages 422–428. IEEE, may 2018.

[12] João Pedro Sousa Castro. Mobile application for online monitoring and collaboration of a sporting event, sep 2020.

[13] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45–48, dec 2000.

[14] Ekaterina Yashkina, Arseny Pinigin, Joo Young Lee, Manuel Mazzara, Akinlolu Solomon Adekotujo, Adam Zubair, and Luca Longo. Expressing Trust with Temporal Frequency of User Interaction in Online Communities. In *Advances in Intelligent Systems and Computing*, volume 926, pages 1133–1146. Springer Verlag, 2020.