Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

In the Net of Collaboration: Formation of Learning Groups in Online Social Networks

Steffen Brauer

Master Thesis

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Steffen Brauer

In the Net of Collaboration: Formation of Learning Groups in Online Social Networks

Master Thesis eingereicht im Rahmen der Masterprüfung

im Studiengang Master Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Thomas C. Schmidt
Zweitgutachter: Prof. Dr. Michael Köhler-Bußmeier

Eingereicht am: 4. Dezember 2015

Steffen Brauer

Thema der Arbeit

In the Net of Collaboration: Formation of Learning Groups in Online Social Networks

Stichworte

Soziale Netzwerke, Gruppenbildung, computergestüztes kollaboratives Lernen, Diaspora

Kurzzusammenfassung

Durch den Aufstieg sozialer Netzwerke hat sich der zwischenmenschliche Informationsfluss im Internet entfernt von starr bereitgestellten, moderierten Strukturen. Das Bilden von Gemeinschaften trägt hierbei ein hohes Potenzial für den Erfolg von Technologie-gestüztem Lernen. Motiviert durch die Möglichkeiten des gruppenzentrierten, aufgaben-basierten Lernes im Internet, befasst sich diese Masterarbeit mit dem Problem die richtigen Lernenden in einem sozialen Netzwerk zu finden. Hierzu werden entsprechende Metriken ermittelt und ein automatisierter Ansatzes für lernorientierte Gruppenarbeit präsentiert. Der Gruppenbildungsprozess schlägt die Zusammenarbeit auf der Grundlage geeigneter Werte in der Tag-basierten Wissensbewertung, gemeinsamen Lernstil und Nähe im sozialen Netzwerk vor. Das Verfahren wird mit Hilfe empirischer Daten der Stack Exchange Plattform evaluiert.

Steffen Brauer

Title of the thesis

In the Net of Collaboration: Formation of Learning Groups in Online Social Networks

Keywords

online social networks, group formation, computer-supported collaborative learning, diaspora

Abstract

The rise of Online Social Networks has changed inter-personal information flows away from rigidly provisioned, moderated structures. The online community-building of social networks bears a significant potential to the success of technology-assisted learning. Motivated by exploring the realm of group-centered online collaboration with task-based learning, this thesis addresses the problem of finding the proper peer learners in an OSN by identifying relevant metrics, and presenting an automated approach for learning-oriented team building. The group formation process proposes collaboration based on appropriate scores in tag-based knowledge ranking, common learning styles, and proximity within the social network. The method are evaluated with the help of empirical data from Stack Exchange Network.

Contents

1 Introduction

Over the past decade, qualification trends in society have shifted responsibility for learning and education more and more into the hands of individuals. While educational institutions have lost their unique tie to knowledge distribution, the new technologies and the open Internet have granted easy access to content, communication, and collaboration in learning. Online Social Networks (OSN) stimulate their users to socialize with friends and communicate with each other. Discussions in groups are user-triggered and do not need a moderator or facilitator. OSNs enjoy an overwhelming popularity among students.

In current eLearning environments like eLearning Content Management Systems (LCMS) and Computer-supported collaborative learning (CSCL) physically distributed users are able to access structured content and collaborative tools. By adding inter-group communication, learning material can be manipulated by group collaboration using Internet communication technologies like text chat, instant messaging, audio- and videoconferencing.

These applications usually demand for an instructor who prepares, holds and analyses courses. A important task of the instructor is to create a learning group constellations that is reasonable from a didactic point of view. A common knowledge of the members enables collaboration via discussing on topics on the same level. In addition, an instructor tracks the peoples learning progress by analysing course results, which makes instructors a critical resource in current eLearning scenarios [1]. Also for this reason, the deployment of LCMS' and CSCLs is commonly limited to dedicated courses or schools.

Project Mindstone[1] has worked out how to open the learning process and the building of learning groups to become part of social Internet eco system. Such an eLearning-enabled OSN allows users to self-pace learning on topics of personal interest, and teams of personal choice. To follow the non-hierarchical paradigm of the social Web, any kind of instructor is omitted on the platform. A instructor-less concept leads to questions about the design of such an eLearning system [1]:

1. How to stimulate a team building process that is effective for learners?

[1] http://mindstone.hylos.org

2. How to provide access to the relevant content for a learning group?

3. How to facilitate a consistent learning progress, include feedback and corrective actions?

This thesis focuses on the first question. Before investigating how to find learning groups, the question what an effective learning group is has to be answered. There are many possible aspects that can influence the quality of a learning group. Often criteria like skills and learning style are taken into account, but OSN provide the possibility to account for social relationships between the learners when building groups. Here, former collaboration [2] or a representation of trust between users [3] are possible factors. In this thesis each learner of a ideal group is motivated to collaboratively learn on a certain topic. The learning style of the members is appropriate to form a balanced group and the learners background on the topic is compatible among the whole group. Also group members are well connected in the underlying network. The contributions of the present thesis are:

Group Formation Approach The presented approach of finding learning groups in the eLearning-enabled OSN is divided in two parts. First, the social network is searched and the approach tries to find a minimal number of suitable candidates for the formation of a group, which an initiator shaped on a chosen topic. Based on the candidates, the second part tries to optimize a constellation of collaborators for a successful group learning experience. Both steps are grounded on metrics that are calculated from user configuration and statistics in the underlying online social network.

Comprehensive Evaluation The group formation approach is evaluated on two large empirical data set extracted from the data of the Mathematics and Superuser site in the Stack Exchange Network. The evaluation covers the compliance of the requirements, the algorithmic parametrization as well as the group quality according to stability and in comparison to the empirical groups in the data sets.

1.1 Previous Work

This thesis marks the end of several other publication stressing its objectives. The overall context is defined by the project Mindstone[2]. It

> seeks novelty beyond traditional elearning applications: Assuming content and organization in place, mobility-compliant dialog functions are designed and implemented to facilitate ubiquitous contextual learning. Understanding new features

[2] http://mindstone.hylos.org

> of the world is fun and part of a life long learning process. Mindstone addresses this need by designing a content-centric social network.

How online learning can be socialized is issued by the paper of Roreger and Schmidt [1]. They identify the key objectives of transferring classic eLearning environments into a social network setting. The questions ,concerning the objectives, developed are listed above. Answering the first question stated by Roreger and Schmidt was conceptionally answered by Brauer and Schmidt [4] by introducing an early version of algorithm presented in this thesis. Besides the group formation approach in the context of an eLearning social network, the paper also included a first evaluation on synthetic data. The concept of an eLearning platform modelled in a unified graph structure was introduced in [5] as well as a content recommendation approach. Also the categorization of Personal Learning Networks was taken into account to analyse the relations in OSN in a learning context. To improve the understanding of groupings in Online Social Networks, the Circle structures of Google+ were evaluated in [6]. A report covering the implementation of the group management component of the eLearning-enabled OSN was also created before this thesis. The thesis summarizes the theoretic work on eLearning-enabled OSN and states an updated version of the group formation approach. New contributions are the implementation and integration of the approach in the platform and the in-deep evaluation on empirical data.

1.2 Structure of this Thesis

The remainder of this thesis is structured as follows, starting with an introduction to Online Social Networks, including their graph representation and vertex groupings. The state of the art of eLearning environments is presented and Personal Learning Networks are introduced and analysed applying a scheme for classic social networks. Based on these two theoretical building blocks, the concept of an eLearning-enabled OSN is presented. Chapter 3 argues related work from the fields of group formation, search in social networks and recommendation system and introduces the group formation approach including the formal model needed to quantify the learners and the two phases Candidate Selection and Group Optimization. A report of the implementation can be found in Chapter 4, starting with the system architecture and continuing with a technical review on the open source social network diaspora*. The details of the implementation of the Group Formation Engine and Graph Repository are also content of this chapter. The evaluation of the group formation approach is the objective of Chapter 5. Based on a report how and why the data of the Stack Exchange platform is transformed to suitable data set, the compliance of the requirements, the algorithmic parametrization and the quality of the found groups are investigated. The thesis finishes with a conclusion.

2 Background

This chapter covers the background on how to form learning groups in a eLearning-enabled Online Social Network. Starting from a graph theoretical point of view, Online Social Networks and their typical characteristics are introduced in Section 2.1. This is followed by the state of the art of eLearning environments and an introduction of Personal Learning Networks. Afterwards the eLearning-enabled Online Social Network is introduced in Section 2.3.

2.1 Online Social Networks

Nowadays Social Networking Sites (SNS), like Facebook[1] or Twitter[2] are used by millions of people and take an important role in the interpersonal communication [7]. Besides Social Networking Sites, social relations in the online world can also be maintained implicitly via different services like E-Mail or Chat. The general representation of social relations in the Internet is called Online Social Network.

2.1.1 Formal Representation

To analyse the structure of OSNs, graph theory is often used to formalize the network structure and make statements about its characteristics. Such a graph consists of several vertices that are connected by edges. The vertices in an OSN are Internet identities of the users. Musial and Kazienko [8] define this mapping as follows:

> "An internet identity iid is a short digital, verified, authenticable, unambiguous and permanent representation of the physical social entity - a concrete human or group of people, who are aware of the single internet-based system. The task of an internet identity is to transfer the physical entity form the real to the virtual world."[8]

The vertices are connected by edges indicating a social relation. The semantic of the relation highly depends on the OSN and can be directed or undirected. While most platforms support

[1] https://www.facebook.com

[2] https://twitter.com/

more than one type of relations, it is possible that two Internet identities are connected via multiple relationships (for instance: direct friendship, commented on same post). The aggregation of all relationships between two $idds$ is called a *tie*:

> "A single tie $t \in T$ may contain up to N different social internet-based relationships $T = R_1 \cup R_2 \cup ... \cup R_N$."[8]

The vertices (Internet identities) and edges (Ties) together build the graph $G(V, E)$, where V is the set of vertices and E the set of the edges.

Based on the graph representation, researchers have found several characteristics in Social Networks using the methods of graph theory. These characteristics are often common to other types of networks, like the Internet or biological networks. The Small World Problem published by Milgram 1967 [9] is one of the early objectives in social network research. Milgram discovered that the inhabitants of the US know each other through 6 other people. This phenomenon is known as six degrees of separation [10]. From the graph perspective, this observation means that the diameter of the social network including all people in the US is 6. The diameter of a graph is the largest distance between any pair of vertices. To find the diameter of a graph, all shortest paths between all pairs of vertices need to be calculated.

The degree $d(v)$ of a vertex v in a graph G is defined as the number of edges connected to v. In directed graphs, the in- and out-degree are distinguished. Previous research on Online Social Networks indicated that the distribution of the degree often follows a power-law distribution [11, 12, 13]. The semantic of a power-low distribution (also known as the 80-20-rule) is that 20% of the vertices have 80% of the edges, but 80% of the vertices just have 20% of the edges. Foudalis et al. [14] studied the degree distribution in graphs over time and found that vertices which have already a high degree tend to establish more connections over time than vertices with a low degree.

Granovetter [15] established the concept that the ties in a social network can be distinguished into strong and weak ties. The observation shows dense clusters of vertices that are interconnected by only a view bridging edges. As these weak ties serves as connections between clusters, they have a high impact on how information traverses the network [10].

2.1.2 Vertex Groupings

Vertex groupings in social networks indicate a common context that is present in the graph structure by highly interconnected groups of vertices. While one objective of this thesis is to develop an approach of how learning groups can be found in the eLearning-enabled Online

Social Network, a deep understanding of existing grouping-mechanisms in Online Social Networks is needed.

Community as a Communication Feature

Groupings of vertices in graphs are called communities. In OSN communities can be anaylsed from two different perspectives: The community as a communication feature of on Social Networking Site and as a structural property of the social graph. Communities as a communication feature are explicitly created and maintained by the users on the platform. Most Social Networking Sites enable their users to join such communities that are also called groups. In which users can post messages or share content. The visibility of the contributions is often limited using different types of groups or communities. There are public groups, to which every member of the OSN can join freely, and private groups that require an invitation by a member or creator [13]. These possibly large communities have motivated various work related to OSN. The main research subject in the feature perspective are the social dynamics of communities. Based on observations of communities Kraut et al. [16] and Preece [17] collect details on how to start and maintain successful communities, how social norms and behaviour reflect in them and how to encourage commitment and increase engagement.

Brzozowski et al.[18] found that most active users in Google+ describe their motivation for joining communities is finding like-minded people and relevant content. A more differentiated classification of communities was introduced by Lazar and Preece [19]. Here, communities are formed according to

- a common attribute of the members in the community (e.g., interest)
- the software platform used to support the community (e.g., newsgroup or Internet Relay Chat)
- reflection of offline communities (e.g., a church group)
- the designed boundedness of the community provider

The context of communities in the social network was considered by Brzozowski et al. The authors state that ”on one hand, communities enable users to circumvent the graph to reach others who share their interests. On the other, social network users may already be destined to connect to others who share their cultural identity and values, so communities in an OSN may simply reflect existing ties”[18]. So communities as a communication feature in OSN serve as arrival and connection point for users, who wish to recreate existing ties or create new based on common attribute.

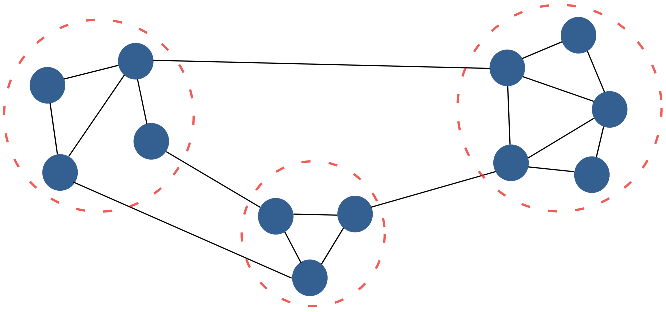


Figure 2.1: Example Graph including 3 Community Structures

Community as a structural Property

Communities created by a communication feature in a OSN are explicitly labelled by a common attribute the users share. But without labels, the common attribute of the users leads to a high connectivity in the social graph. Community detection aims to find the communities hidden in the graph structure. Besides OSN, communities can be found in different types of networks like biological or information networks. All communities structures refer to a hidden common attribute of the involved vertices that causes a high connectivity [20]. A general definition of communities in graphs is that a community is a set of vertices with many connections within it and just few connections from the community to the remaining network [21]. Figure 2.1 shows a example graph including 3 communities.

Fortunato [22] quantifies the definition of community C in the graph G with n as the number of vertices in the graph $n = |V|$. The number of vertices in the community is $n_C = |C|$. The edges are defined as m_C for edges in C and c_C for edges on the boundary of C connecting the communities to the remaining network. Applying the definition from a above an ideal community has a high value for m_C and a small one for c_C.

While this approach on defining communities focus on the edges another approach of defining communities focus on the vertex similarity. Lorrain and White [23] define two vertices as structural equivalent if they have the same neighbours but are not directly connected. Fortunato [22] uses the Jaccard Index to measure the overlap between the neighbouring vertices of v and u:

$$J(v, u) = \frac{|\Gamma(v) \cap \Gamma(u)|}{|\Gamma(v) \cup \Gamma(u)|} \tag{2.1}$$

The intersection of the neighbours $\Gamma(v)$ of vertex v and $\Gamma(u)$ of u is divided by the union of the neighbours. In a community structure the intersections of the neighbours is indicating a high connectivity and cohesion. In contrast to the previous quantification of communities, the focus on vertices considers indirect connections via a common neighbours and not only direct connections.

Selective Sharing

Another type of vertex groupings in OSN is selective sharing. In OSNs, the audience of direct communication and private messages is explicitly known, content sharing and status messaging are commonly distributed via implicit replication on the platform. A possible large audience of a post is often not considered by the user, but may have significant impact on the personal life. The so called over-sharing of content can be avoided using features for selective sharing [24], which are provided by several Social Networking Sites. The concept of selective sharing was designed to support a context-dependent publication behaviour. In different contexts like work, family, or friends, a person can act differently according to appropriate norms and accepted conventions. These contexts so called facets [25] or foci [26] describe different social aspects in the life of a person and provide the theoretical background of selective sharing features. In preparation of this thesis the selective sharing feature of Google+ was analysed [6]. In Google+ the groupings of contacts are called Circles. While other Social Networks like Facebook support groups for selective sharing, Google+ forces the user to put new contacts in circles. It thus inverts the application logic of communities. Whereas groups in traditional OSNs form optional overlays of the social graph, circles are mandatory sub-structures of a users ego-network in Google+. The ego-network of a user in the social graph describes the personal view on graph including the owner (the 'ego'), the neighbours (the 'alters') and the edges connecting these vertices [10]. Some users try to avoid this categorization by putting all contacts in one circle, but the majority actively adopts this perspective change when building social contacts [27].

While in traditional communities users join on their own will, the circles are created by users from their ego-network. The work done in [6] tried to explore the effects of these different building mechanisms on the social networks and on the processes of selective sharing therein. Based on a Google+ data set [28] with shared circles, the structures of circles were characterized. Comparing two social networks of circular structures with two data sets that are built from traditional communities, the paper could show that circles

- form pronounced community-like structures in Google+ and
- attain an individual structural signature

In particular, circles are significantly less separated from the remaining network than classical communities. Selective sharing in Google+ is thus more diffusive and less confined.

Summary

Communities as a communication feature in OSN serve as arrival and connection point for users, who wish to recreate existing ties or create new based on common attribute. The common attribute of the community reflects in a high connectivity of the community members. In classic communities users can join on their own will. New feature for selective sharing in OSN support the creation of communities within the ego-network of a user. Learning groups in the eLearning-enabled OSN are communities designed for communication. Recommending members for these groups has to take advantage of the structural definition of communities to aim for members connected by common learning topic or former interaction. The scope of the recommendation may vary from the ego-network to the whole network according to the concert topic the learning group should work on. Selecting members from the ego-network may increase the connectivity of the group.

2.2 eLearning

In the same way the nature of the Web had move from static content sites that are created and maintained by a few administrators, towards dynamic content, maintained by the users themselves on different platforms for communication and content creation, eLearning is moving from technologies that support courses with sequential content paths, managed by instructors towards to dynamic, personal environments based on social media services.

2.2.1 State of the Art of eLearning Environments

In the previous work of the development of the eLearning-enabled OSN, the Learning Content Management System (LCMS) [29] Hypermedia Learning Object System (hylOs) [30] was developed. LCMS allow physically distributed users to access structured content. Modern LCM-systems organize content in eLearning objects [31] that interrelate to form an instructional or semantic network [32, 33]. hylOs is an adaptive eLearning content management system and runtime environment, built upon an information object meta-model [34] tailored from the IEEE LOM (Learning Object Metadata) standard [35]. hylOs comprises instructional design concepts and tools, a content acquisition and analysis engine for semi-automated generation and annotation of eLearning Objects, as well as an Ontological Evaluation Layer for concluding relations between eLearning Objects. Based on meta data, taxonomies and an intrinsic ontology, the system provides automatic reasoning to produce semantic learning nets [36]. hylOs provides adaptive eLearning functions and may attain any look & feel by applying appropriate XSL transforms. Variable content access views like instructional learning paths or individual content explorations based on semantic nets may be compiled for the hylOs repository. Traditional hyper-references, which provide a separate layer of content traversal, may be adapted to a learning context within hylOs. Links are represented within contextual containers, each one suitable to express a narrative of a specific hyper-linking scheme [37]. A disadvantage of LCMS is that they are mostly under the control of a institution and administrators. They also leave limited options for learners to manage an own learning space that facilitates their learning activities as well as connections to others [38].

Personal Learning Environments The focus on eLearning research is moving from a content-centered perspective to a learner and social interaction focused perspective [3]. Besides the technological trend of integrating social media aspects in eLearning, learning theories based on social connections via the Internet gain more interest [39]. One result of this development are Personal Learning Environments (PLE). They consist of a set of several Web services, which can be used to access content from social media services like Twitter, Blogs or discussion forums. Besides the passive content consumption, communication takes an important role.

Martindale & Dowdy [40] state that users of these services can now interact with content as well as with others learners in a shared environment. This enables them to organize content that has meaning to them and easily share that content and their own interpretation of it. Furthermore, learners can interact with other people with shared learning goals. The application interoperability is a problem in this scenario [41]. Dabbagh & Kitsantas see PLEs as "a manifestation of a learners's informal learning processes via the Web, or, as a single learner's e-learning platform allowing collaboration with other learners and instructors and coordination of such connections across a wide range of systems" [38]. Definitions of PLEs describe them as "a specific tool or defined tool collection used by a learner to organize his or her own learning process" or "a metaphor to describe the activities and milieu of a modern online learner" [40]. Both types of definition share that PLEs give the control of the own learning process to the learners.

In contrast to LCMS, Personal Learning Environments focus on the learners personal characteristics and commit the control to them [38]. More generally, LCMS and CSCL try to map the learner to the system, but PLEs are created by the learner during the learning process and follow no model, which was created by instructor or engineer and are maintained via services not primary designed for learning.

2.2.2 Personal Learning Networks

A new term in the area of PLEs are Personal Learning Networks (PLN). They describe the social relationships, which are created by the learner during interaction through the PLE. Couros [42] uses the definition that "personal learning networks are the sum of all social capital and connections that result in the development and facilitation of a personal learning environment". Warlick [43] stresses the importance of aggregation. New learning content does not have to be found, but is delivered to the learner via the feeds of the other learners in the PLN. While this increases the availability of content, Warlick although mentioned the importance of a variance of information sources.

Categorization of PLN

To categorize and analyse Personal Learning Networks, the scheme of Musial and Kazienko [8] is applied. As mentioned above, a simple OSN consists of a finite set of internet identities IDD and social relations of one type R. Typical OSN like Facebook support the creation of

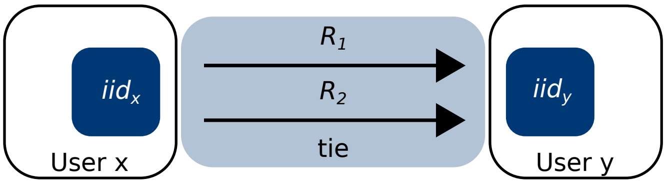


Figure 2.2: System-based Social Network (SSN) [8]

different types of relation. People can send direct messages, comment on same post or like one. Such a OSN, including the Internet identities of users and several kinds of relation is called a System-based Social Network $SSN = (IID, T)$ (Figure 2.2). In the context of PLN, a SSN is one service used by the learner to maintain the PLN through the PLE.

Warlick [43] defines three types of relationships in PLN:

- Personally maintained synchronous connections,
- personally and socially maintained semisynchronous connections and
- dynamically maintained asynchronous connections.

Synchronous refers here to the participation of both internet identities in the creation and management of the relation. The first type refer to communication between two iids, which communicate directly via private messages or email. The second relationship type is described by Musial and Kazienko [8] as a quasi-direct relation. Here the relation includes a meeting object. This can be a communication medium like a blog post both $iids$ commented on. While the first type have to be maintained by the connected Internet identities, this is not required by a quasi-direct relationship. The third type describes the connection of users via content or content sources. An example for this type of PLN conncetion is a RSS Feed. Here users subscribe to a source and new content is delivered to them. Considering all the kinds of connections mention in these types of PLN, they all are directed. On Twitter, users follow each other, Content on Google Docs and in Wikis is edited by users and users subscribe to RSS Feeds. A PLN is also a directed graph with different vertex types including learner, content and communication artefacts.

Using the Internet identity of a user in only one system limits the scope of an analyses of PLN to only one Internet-based system. To add the possibility of having a mapping of multiple accounts on different platforms to the same physical entity, the concept of virtual Internet identities is introduced:

> "Virtual internet identities aggregate distributed internet identities existing in different internet-based systems. A virtual internet identity $viid$ corresponds to all

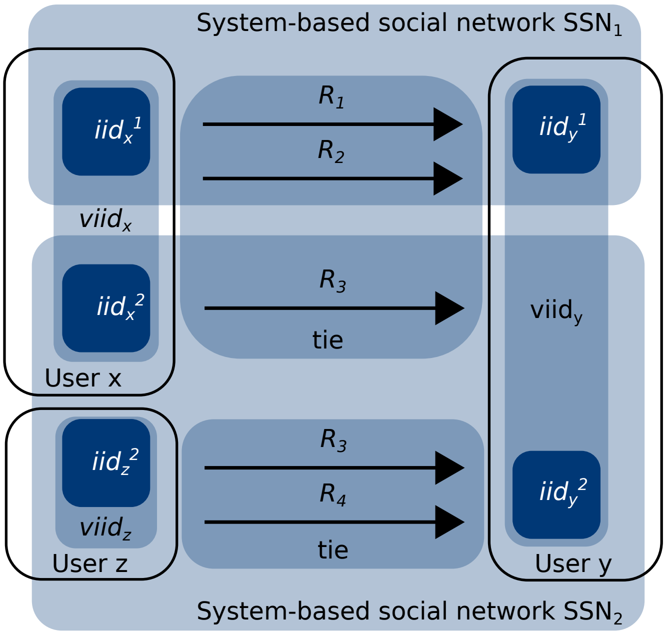


Figure 2.3: Internet Multisystem Social Network (ISN) [8]

> internet identities iid related to the single physical social entity. Simultaneously, each internet identity is related to only on virtual identity."[8]

By using virtual Internet identities to join the $IIDs$ of different SSN a multi-platform model of social networks is created:

> "An internet multisystem social network ISN for the set of m system-based social networks $SSN_i = (IID_i, T_i), i = i, ..., m$ is the tuple $(VIID^M, T^M)$ where $VIID^M$ is the set of virtual internet identities related to the same physical social entity. T^M is the set of ties."[8].

A discussion of the distributed nature of PLNs as well as their formal characteristics is possible using the concept of an ISN. Figure 2.3 shows the scheme of an ISN. Assuming that the users x and y use the SSN_1 and SSN_2, the relations $R_{1..3}$ form their PLN, where the learners are represented by their $VIIDs$. The awareness of the learners that the concrete iid of another learner is part of a $viid$ is not always given. Some services have separate fields in the profiles to link $iids$ in other services. Besides the awareness of the iid, the awareness of ties may also be hidden on the boundary of $SSNs$. In Figure 2.3, user y and x are active in both $SSNs$, but user z uses only SSN_2. While all three users can communicate via SSN_2, user z is not aware of the relations R_1 and R_2 in SSN_1 between x and y. Depending on the privacy settings of SSN_2, user x may be not aware of the relations between z and y. In this scenario, only user y is able to see all relations. So besides the distributed nature of PLN, the visibility of relations is bound to the learners PLE.

Formal Characteristics of PLN

While the analysis of PLN is limited by the scope according to different services and privacy settings, this section focus on the analysis of PLN from an service independent position, as an ISN connecting learners through their PLE.

The concept of weak and strong ties, established by Granovetter [15], can be transferred to PLNs. It argues that the ties in a social network can be distinguished into strong and weak ties. Learners can hold connections to other people in their learning groups or they former interact with. This connections are high frequented by direct messages or participating in discussions. While these connections form strong ties, a learner may also hold connections to important scientists in the fields of interest. These connections are mostly based on consuming content shared by the important personalities. Because of the probably low number of common contacts by the high connectivity via the important person, these connections can be called weak ties.

The example of a learner and a famous scientist can also be used to describe the reciprocity in PLNs. Reciprocity measures the likelihood when having a edge from vertex x to y to have another edge from y to x. Applied to the example, learners on the same eye level shall have a higher reciprocity because of their peer discussions or using the same content, while learners with a high difference in eye level tend to a low reciprocity, because the learner with not much experience in an field of interest mostly consumes the content from the learner with high experience.

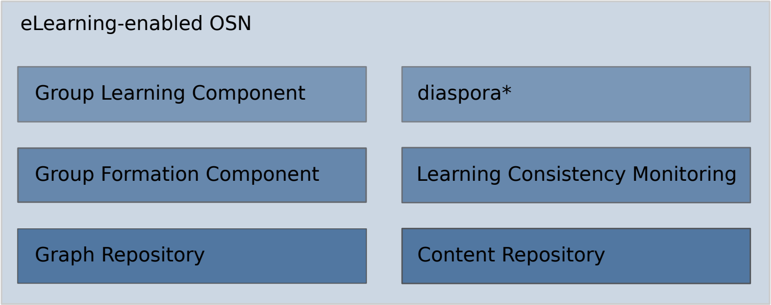


Figure 2.4: Conceptional Overview of the eLearing-enabled OSN

2.3 eLearning-enabled OSN

The concept of an eLearning-enabled OSN was drafted Roreger and Schmidt [1] and complemented by Brauer and Schmidt [5]. This Section presents the concept and the different components of the system. The objective of the eLearning-enabled OSN is to provide an environment for learners to maintain their PLN combined with formal open learning groups. Learners browse the network, create connections to others with the same interests and can start groups to work on collaborative tasks. These tasks, called topics in the model, are created by other learners and cover any kind of activity like creating a piece of content or exchange their knowledge in discussions. While it is not possible to provide all offers that other services implement, the goal is to create an open platform, which can be easily extended and allow integration of other services.

Figure 2.4 shows an conceptual overview of the eLearning-enabled OSN. On the top layer are OSN diaspora* and the group learning component. The open source project diaspora* [3] is a implementation of a Social Networking Site that serves as the environment for the eLearning extensions by providing elementary social networking features. The learning groups and topics are managed by the group learning component. It includes the extension of learner profiles and features for creating groups and topics as well as a group invitation system and group discussions. The middle tier of the eLearning-enalbed OSN consist of the group formation component and the learning consistency monitoring. Possible constellations of learning groups are provided by the group formation component. Starting from the group initiating learner, the engine searches for possible group members based on learning style, knowledge and position in the social graph. The task of the learning consistency monitoring is to preserve the consistency in the sense of learning path flow. To do so, it adapts the content representation to the context of the learner and monitors the groups according to their cohesion. A integration of a content

[3] https://joindiaspora.com/

network is implemented by the Content Repository. It manages the creation of learning objects and is able to reason relations between them. The graph, containing all eLearning related entities is stored in the Graph Repository. In the following, the representation of the learner and the components of the eLearning-enabled OSN are introduced in detail.

2.3.1 Learner Representation

Learners are the active part in the eLearning-enabled OSN. They create content, topics, form groups or assign tags to the entities. To provide relevant content and possible co-learners, the learning style and knowledge of a learner is tracked by the system. A natural aspect to evaluate collaborative potentials in learning is harmony in learning style. Learning style models, though, are sometimes criticised according to their reliability, validity, and implications for pedagogy. Coffield et al. [44] presented a review of learning styles and conclude that there is a lack of theoretical coherence and a common framework. Nevertheless, the use of learning style in eLearning application for selecting a certain content representation can improve the learning experience [45, 46, 47].

In the eLearning OSN, the learning style is employed in agreement with Felder and Silverman's theory (FST) [48]. This work is widely accepted as a standard way to assess learning styles. A key feature of this theory is that it does not try to force a learner into one specific category of leaning, but variably assigns preferences to a learner in the four predefined dimensions:

- "Active or Reflective" (Processing)
- "Visual or Verbal" (Input)
- "Sensing or Intuitive" (Perception)
- "Sequential or Global" (Understanding)

In each of these dimensions, the learner can have three different strengths, i.e., fairly well balanced, moderate preference, and a very strong preference.

Measuring learning styles within an eLearning application has many advantages. In the eLearning-enabled OSN it is used as an instrument for customizing search, select a presentation of content and use it in the group formation to create groups with a common learning style. The authors of FST propose to use a questionnaire to determine the learning style, but this has proven to be inappropriate for eLearning environments [49]. Questionnaires are also not able to detect changes in the behaviour of a learner over time.

To represent the learners knowledge the platform uses tags. In contrast to ontologies, which require experts on the subject and lack flexibility, a lightweight approach is employed that combines knowledge annotations for learners, content and topics. This largely increases flexibility, While ontologies are only able to represent a Web of a special topic, tags can jointly describe content, the competence of a learner, or the context and style of a content object [3].

Clements et al. [50] distinguish between individual and collaborative tagging systems. In individual tagging systems user-generated content is published where the creator is able to assign tags to the content. Users who do not publish content do not build relations to tags and may be left out by recommendations that are build on the tagging system. In collaborative tagging system all users are able to assign tags to every piece of content. Content in these systems has a more diverse description of content and more users have assign tags enabling creation a tagging profile. To handle the problem of synonyms and typing faults, Gruber [51] recommends an auto-completion feature.

2.3.2 Open Source OSN diaspora*

diaspora* [4] is an open source implementation of a distributed Online Social Network. It is written using the Ruby on Rails Framework[5], which applies a 3-tier architecture to the application including recent Web technologies. It is also easily extendible using rails engines that are loosely interconnected to the main application, but have full access to its code. This enables our extensions to reuse some parts of diaspora* and to deeply integrate it into the existing system, but the code is clearly separated. This is an advantage of extending diaspora* in contrary to using an API of a large commercial social network with limited data access and little possibilities of integration.

The social graph of diaspora* is distributed over a network of independent, federated servers—so called pods—that are administrated by individuals so called podmins [52]. When joining the network, a user has to choose a pod. The personal data are only stored within the database of the chosen pod. There is no centralized indexing instance, which has access to the whole social graph and personal data of all users.

Features

diaspora* supports the core features of a typical OSN like sharing content with others, private messaging and user discovery. In the following, we want to introduce the features that distinguish diaspora* from other OSNs.

[4] https://joindiaspora.com/
[5] http://rubyonrails.org/

Contacts A user can start *sharing* with another user. The semantic of this relation is that the posts of user A are delivered to user B. This relation is directed and needs not to be acknowledged by user B. But user B receives a notification when user A started sharing and can also start sharing with user A. With these directed edges, the social graph of diaspora* can be described as a directed push network between users. The persons user A is sharing with are called *contacts*. The sharing relation is not restricted to a pod, but can be established between all registered diaspora* users by using the WebFinger protocol. An in-depth discussion of how users connect in diaspora* can be found in Section 2.3.6.

Aspects When user A starts sharing with user B, A has to attribute an *aspect* to B. This feature implements selective sharing, discussed in Section 2.1.2. Besides the selective sharing of posts, diaspora*s aspect feature implements a selective reception of posts generated by the contacts in each aspect.

Tags During the registration process, a new diaspora* user can select different *tags* to follow. The followed tags have individual streams, a collection of posts that contain all posts visible to the user that are marked with the tag.

In social networks, tags are used to mark posts and other content with keywords to describe its context [3]. Marked with a '#', tags are interpreted by the network and posts and content marked with the same tag can be found by clicking on the tag.

The scope of tags in diaspora* is limited to the visibility of posts. When a post is not visible to the user, because she does not share with the author or the post is not public, the tag connection is hidden. Another limitation is that only public posts are visible for local users, which are stored on the local pod. Posts are stored on the senders and recipients pods, which leads to an imbalance of public information on different pods, because public information is not spread to all pods. This is a problem of the current diaspora* implementation, which could lead to a development against the distributed nature towards to only few large pods.

2.3.3 Group Learning Component

The learning groups are in the center of the group learning component. It covers all features, which are needed to enable the communication within and management of the learning groups. A learning group can be open or closed. If the group is open, other users can join in the active state at any time and the group can be found by all other users of the learning network. If the group is closed, other users have to be invited to join. Only members are able to see the group in the system. During the learning process, members can join a group according to its

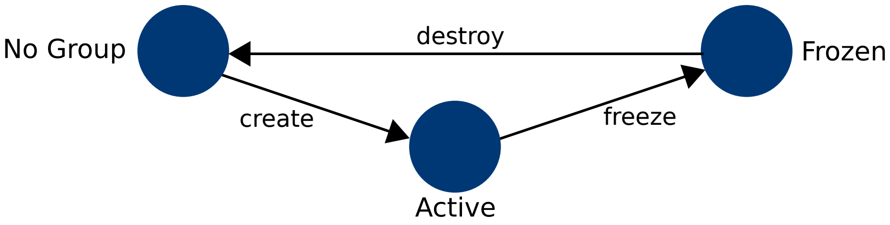


Figure 2.5: Life cycle of a learning group

visibility or leave it. The group members define, when the goal of collaboration is reached by voting for it. The final result should be any kind of artefact. A learning group can have three different states (see Figure 2.5): The group does not exist, or it is active, or it is frozen. When a user creates a group, a name and a description need to be assigned. In addition, the creator has to decide, whether the group is open or closed. Now the state of the group changes from *no group* to *active*. In the active state, content can be shared within the group and members can discuss about the topic. It is possible that new members can join according to the visibility of the group. To terminate the active state, all members have to agree that the task of the topic is completed. In this case, the group freezes and changes to the *frozen* state. Now it is not longer possible to join or post something to the group. But the group is still visible to its members and can be used as an archive. Members of a frozen group can request to delete it. When all members agree, the group is deleted and the state is again *no group*.

Based on the concept of the learning groups, a specification was developed, which describes the features that have to be added to diaspora* to enable a group learning experience. Besides the learner, *topic* and *group* are the main entities. A topic consists of a title, a description (including a task), a goal and several tags, which characterize the context of it. It is possible to create a group on a topic or select it in the group creation process. A group has a name, a description, a topic the members are working on and a state.

2.3.4 Learning Consistency Monitoring

The learning consistency monitoring introduced by Roreger and Schmidt [1] outlines a answer for the question on how to facilitate a consistent learning progress, include feedback and corrective actions. The extension aims to add ubiquitous learning to the eLearning-enabled OSN. This enables the learner to learn anything at any time leading "to learning in several contexts like different places, different levels of noise or different levels of concentration" [1]. The learning consistency monitoring will adopt the learning content from the content network to the current learner context.

Besides the adopting of content, the extension also monitors the collaboration in the learning groups. Roreger and Schmidt propose to use lexical markers based on the work of Reffay et al. [53]. Here the group quality was verified by tracking lexical markers and the authors found that the occurrence of 'we' in discussion indicates a intensive communication inside the group. Roreger and Schmidt also formulate the possibility to analyse the homogeneity in content consumption and interactivity patterns in order to track the group cohesion. The authors assume that groups "operate rather diverse, remain unusually quiet or dominated by individuals indicate a lack of social success"[1].

2.3.5 Content Network

The Content Repository, created by With [54], implements a semantic content network within the eLearning-enabled OSN. It includes a management component that helps to organize and serves learning objects. Using a linking module, semantic relations between the learning objects are created by meta data following the LOM standard [35] and by reasoning rules based on existing relations. The concept of content networks extends the work on hylOs and the included Ontological Evaluation Layer. The learning objects are stored together with groups, topics and learners in the eLearning Graph.

2.3.6 Graph Repository

Diaspora* uses a MySql database instance to store its models. The scheme used here is normalized and optimized for read and insert operations, but lacks usability and performance by analytical queries, because of join operations and selection on large tables. This issue is addressed in the field of Business Intelligence. Here an analytical database has an optimized scheme for querying historical aggregated business data to support the controlling. While analytical databases often have a multidimensional scheme, in the domain of social networks a graph scheme is a adequate choice. It is easy to map the eLearning-enabled OSN concept to the graph database scheme and the data representation allows to traverse the graph without large join tables by just following the edges. To enable an easy-to-use and fast analysis of the elearning graph in diaspora* and the learning extensions a analytical graph database is set up. Its task is to hold a copy of the entities of the eLearning-enabled OSN in a graph structure to analyse the social graph and serve as a data source for the group formation component.

The entire network is modelled as a directed graph with vertices of different types (see Figure 2.6). These types cover all relation kinds between content, learner profiles, groups and topics. The center of the network are the learner vertices. To represent the actual users of the

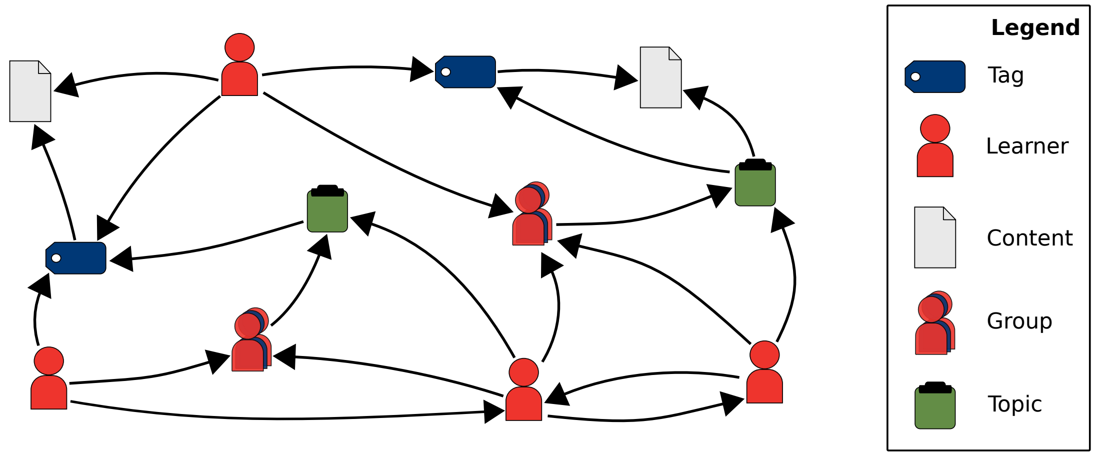


Figure 2.6: Example of an eLearning-enabled Online Social Network

platform, they hold a profile with common OSN attributes extended by the learning style. As the learner vertices represent the active roles in the network, they create or edit other vertices and form edges to them. Besides learners, content is mapped to vertices. Content vertices can refer to resource outside the platform or eLearning objects in the connected Content Network. While learner and content form edges by editing relations, their contextual relation are mapped using tag vertices. The weight of this connection shows the relevance. Topic vertices can be chosen for collaborative group learning. If a group is successfully created, a group vertex is created and all members connect themselves to it.

The (technical) links are typed accordingly. This unified approach (cf., [55, 56]) adds many implicit relations to the network, for instance by editing the same content or participating in discussions. The direct neighbours of a learner describe the personal context in the network. This context extents the profile of the learner exclusive information. This surrounding of a learner in the network have a high potential for discovering new learning partners and interesting content, which can be discovered by simply browsing the network. It also enables algorithms to measure the strength in connectivity of two vertices by accounting for shared neighbours or distinct paths that connect them.

To distinguish the relevance of different edges in the network, weights are used. How the weight is determined depends on the connected vertices.

Learner → Learner The weight of edges between two learners is grounded on the communication artefacts exchanged. Each artefact send from one learner to the other increments the weight. Due to the directed edges in the graph, a answer on a communication artefact, creates or strengthens the edge in the opposite direction (see Figure 2.6).

Learner → Tag To weight the edges between a learner and a tag, each tagging event of a learner is used. When a learner tags a topic or content object, the edge between the tag and topic or content object is created an the weight of the edge from the learner to the tag is incremented.

Learner → Topic The direct relation between learner and topic is defined by an creation or manipulation action. Each action increments the weight by one. A contextual relation is estimated by the Knowledge Rank (Section 3.2.2). It measured the likelihood of the tags shared by the learner and the topic and their weights. The contextual relation is not suitable as a weight, because updating it is a complex task involving the surrounding tags.

Learner → Group The edges between a learner and a group vertex is weighted by the participation of the learner in the collaborative work. Similar to the learner → learner relation, the amount of communication artefacts is measured.

Topic → Tag The weights of tags per topic are assigned by the learner, who edits or creates the topic. This is a critical task, because it determines the overall context the topic is embedded in.

Group → Topic A group vertex has just one out going edge to the topic the group is working on. The uniqueness is not present in the opposite direction, because several groups can work on the same topic.

Learner → Content The weight of edges from or to a content objects are defined by With [54]. The weight from a learner to content is influenced by three factors. The learner

- creates or edits content,
- views the content or
- adds the content object to the favourites.

Each action increments the weight.

Topic → Content A edge between a topic and a content vertex is created by a learner, who marks the content as relevant for the topic [54].

Tag → Content This edge represents the tagging of a content vertex. Keywords from the Content Network are represented as tag vertices in the Graph Repository [54].

Content → Content The connections between two content vertices are created by the reasoning engine in the Content Network. How these edges are created is discussed in [54].

3 Group Formation Engine

The Group Formation Engine supports the learner to find group constellations to work collaboratively on a specific topic. While most group formation algorithms arrange the groups without any control of the learner in the background, the presented approach recommends group constellations to the initiator of the group formation process. The learner can selected a preferred constellation and invitations are sent to the constellations members. With this procedure, the learners are in charge of the group formation with being supported by group recommendations.

The input of the group formation process is the initiating learner and the chosen topic. From this starting position, the Graph repository is traversed looking for suitable co-learners. The challenge of forming a group for an effective mutual learning process lies in finding those people that are not only interested in the same subject area, but are thematically at eye level and match in relevant social dimensions. The approach aims at harvesting appropriate candidates from the graph model of the eLearning-enabled OSN. In this perspective, the problem of group formation in OSNs was frist formulated by Roreger and Schmidt [1] as finding a subgraph of the full social network that fulfills the following conditions.

1. Each learner is motivated to collaboratively learn on a certain topic.
2. The learning style of a learner is balanced among the group.
3. The background on the topic is compatible among group members.

This thesis adds a fourth condition.

4. Group members are well connected in the underlying network.

Condition 1 identified by Roreger and Schmidt addresses the intrinsic motivation of a learner. A simple approach introduced by the authors is to set a flag for each learner indicating a personal interest in collaboration.

Condition 2 of Roreger and Schmidt is based on the learning style of the individual learner. It requires a mechanism to find a sub graph of learners with a balanced learning style. Grouping people who learn in a similar style is proposed by the authors.

Condition 3 presented by Roreger and Schmidt stresses that learning groups should have a common knowledge base. In the present approach the relations including tags in the eLearning-enabled OSN are used.

Condition 4 is introduced in this thesis and considers the connectivity in the network. Learners form relations with each other based on former collaboration or interest in mutual activities. A group of learners that is well connected in the network builds a trustful environment and indicates a positive collaboration based on former interaction.

Besides these requirements, the algorithm should comply to the scale of real world OSN with millions of user. This requirement demands, that the algorithm operates on a sparse subgraph and do not require knowledge about the whole data structure. To handle the problem of finding a group starting from an initiating learner and a selected topic, the approach splits the problem into two sub tasks: Candidate Selection and Group Optimization. The Candidate Selection aims to search the Graph Repository for suitable peer learners. It deals with the sub problems of when is a learner a suitable peer learner, how many learners have to be found to build a group and how to effectively search the network. At the end of the selection, a set of possible group members is created. Based on this set, the Group Optimization tries to find the best constellations of the members and suggests them to the initiator.

The next section discusses related work in the field of group formation in collaboration scenarios. Besides the context of eLearning, forming groups of people with the goal of collaboration is also a topic in a professional context. The focus on group formation is extended by general work on search and recommendation in OSN to gain insights on how to handle problems in the context of large graph structures. Then, the requirements defined above are formalized and metrics are formulated quantifying the requirements. The subtasks Candidate Selection and Group Formation are introduced in detail.

3.1 Related Work

3.1.1 Group Formation

Group formation in the context of eLearning is a well studied field. Cruz and Isotani [57] preformed a literature survey on group formation in the context of collaborative learning, using the method of systematic mapping of literature to summarize and catalogue the work done in this field. They retrieved 48 papers concerned with concepts of group formation, including a previous publication of the approach presented in this thesis [4]. Their results show that most approaches use probabilistic algorithms. Here, the biggest share holds Genetic Algorithms, followed by the swarm intelligence algorithms particle swarm optimization and ant colony optimization. Also the data mining algorithm k-means is used by several techniques. The authors also listed approaches as unspecified and others, covering semantic web and other computational techniques. It was also observed, that the majority of approaches also evaluate their solution mostly on real scenarios. Formulating the problem of source code availability, the authors stress the lack of instruments to compare and evaluate different approaches. Only 2% of papers introducing algorithms provide their source code and 23% provide some kind of pseudo-code.

Most approaches concentrate on forming groups for a classical learning model of a classroom, including a small set of students, and an instructor. In this context, the approaches of Ounas et al. [58, 59] nad Monreno et al. [60] gained popularity. Ounas et al. stress the group formation problem as an constraint-satisfying task that is solved by ontologies and given rules that define the desired group constellation. Moreno et al. use Genetic Algorithms to find suitable groups from a small set of students and also underline the importance of the parameters used in the algorithm. Halimi et al [61] introduce solearn, a social learning network. Based on social relations and activities, it provides intelligent recommendations for the best collaborators, tutors or learning tool but no group formation approach is introduced.

Clique-based Group Formation

Arndt and Guercio [62] introduce a group formation process, which creates groups from of a social network of classroom members. The goal of this approach is to assign each member to a group. The minimum group size is defined by an instructor.

The students in a class are modelled as vertices in an undirected graph G. The edges between the vertices describe friendship relations. Each vertex in G has an associated vector, which contains the learning preferences for a student. The result of the group formation progress is, that all members of the class are assign to a group. Given a minimum size of members in a learning group the algorithm builds groups as follows: In the initialization phase of the algorithm, an empty set is created, which contains the sub graphs of the learning groups. Also MIN is defined as the minimum number of students in a group. Based on G, all cliques are found with a minimum size of MIN. Each of these cliques is removed from G and added to the set of groups. Cliques in graph theory describes a set of vertices, which are all connected to each other. The authors state, that cliques represent a good learning group, because everyone knows each other. After all cliques are removed, the relaxation phase tries to find k-edge-connected components in G, where k is stepwise reduced. The phase also stops, if G is empty or no more components are found. If a k-edge-component is found, it is removed from G and added to the set of learning groups. K-connected-components are weaker connected than cliques. The coalesce phase checks, if G contains more vertices as MIN. If this is the case, G is added to the set of groups. Are less vertices in G as MIN, a vertex is removed from a group with a higher number of members than MIN and added to the group of left vertices. This is done until $|G| > MIN$. In the last phase of the algorithm, each vector of a group member is replaced by an averaged vector of all group members, to transform the course material based on the group preferences to grantee a shard ubiquitous learning experience.

This approach does not facilitate the requirments introduced above, because it is bound to a small size of a classroom. Moreover, the groups created by the algorithm have some weaknesses. Even tough the students provide a vector of their learning preferences, it is not considered in the group formation process. Only in the last phase, the vector of all group members is averaged to enable a shared ubiquitous eLearning experience. Another weakness is that the groups have different qualities. The groups generated in the first phase are densly connected. This feature decreases during the algorithm and the last group is created by the left vertices with possible no connection among each other.

Multi-Objective Team Composition

Another relevant group formation approach is introduced with the goal of building expert teams in a professional context. Dorn et al. [2] try to build expert teams based on the skill of the users, and their relation within the social network. The edges, which indicate earlier interaction, take an important in role in this scheme. Authors also introduce a recommendation mechanism which serves the purpose of routing to another expert, if the desired person is not available. This paper shows that multi-objective team composition is NP-complete. Thus Dorn et al. introduce heuristic optimizations to solve the problem of finding the best team configuration. The group formation approach in total is divided into three main parts: network establishment, candidate selection and heuristic optimization. The divition of the group formation problem into a candidate selection and heuristic optimiziation phase is adopted in this theis.

Network Establishment The first step is to establish a social network with user profiles covering information about the provided skills and the expert availability. The edges in the network are weighted based on former interaction of two experts. Figure 3.1a shows an example network with different edges weights and unavailable experts, who are represented by red marked vertices.

Candidate Selection Based on the network, candidates are selected based on their availability and the required skills. Figure 3.1b shows a set of selected candidates. These

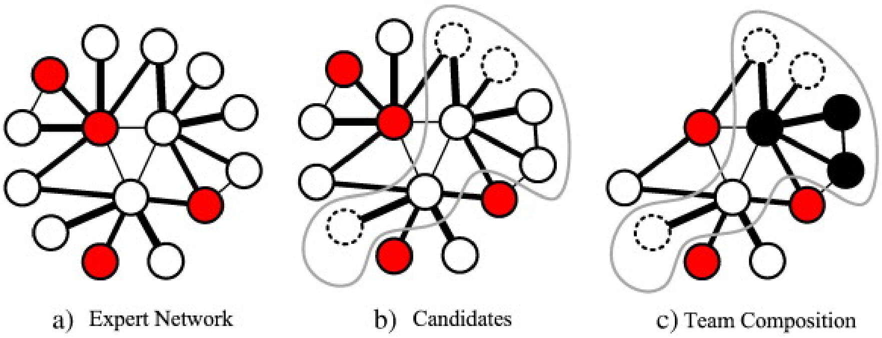


Figure 3.1: Parts of group formation in [2]

candidates are top ranked experts for the skills, but may be just loosely connected in the network.

Heuristic Optimization To find a team, which is better connected than the top ranked experts, the team composition tries to identify a group with a high skill coverage and a high connectivity in the network using Genetic Algorithms [63] and Simulated Annealing [64].

An important role in this approach takes the skill-dependent recommendation model. If a selected user is not available, the algorithm selects another based on the interaction structure of current team members customized for a given situation. To minimize manual management, a self-adjusting trade-off model is introduced which determines the trade-off between interaction distance and recommendations. The presented work includes a proof, that this step of the approach is NP-hard. Because of this challenge two heuristics are introduced: Genetic Algorithms and Simulated Annealing. While Genetic Algorithms apply crossover and mutation operations on a given team configuration and evaluates the team fitness, Simulated Annealing maps each team configuration to a temperature, where a smaller temperature indicates a better team configuration. The evaluation of Dorn et al shows that both heuristics are able to find good team configurations, but Genetic Algorithm performs better and produce more stable configurations.

The work of Dorn et al. [2] presents a promising approach for team composition. The main evaluation focuses on the comparison of Simulated Annealing and Genetic Algorithms and the evaluation of the mathematical modelling. The presented approach is able to form groups on large networks with low computational cost because of the applied heuristics. It is mentioned that the algorithm should be able to search the network very fast, but the paper does not recommend a approach to do so.

3.1.2 Search in Social Networks

While the approaches for group formation are mostly focus on small social networks covering a classroom, they can not be applied to large networks. But the requirement of group formation for searching in large social networks with millions of users demands effective search algorithms to find suitable group members.

Zhang and Ackermann [65] compare several algorithms for search in social networks in the context of finding an expert that matches a vector of required skills. To evaluate the algorithms, they generate a test network from an e-mail data set, where the vertices are generated by the senders and receivers and augmented with keywords spotted in the mails, and edges represent email exchange between vertices. Besides the computational costs of this algorithms, authors also measured the social impact. They found that social network search algorithms, which take the degree of a vertex into account, perform better by finding one expert in the social network. Table 3.1 shows the evaluated search strategies grouped in three families.

Family	Name	Heuristic
General computational	Breadth First Search (BFS)	Broadcasts query to all neighbours
	Random Walk (RWS)	Selects next vertex randomly from neighbours
Network structure based	Best Connected (BCS)	Selects next node based on highest degree from geihbours
	Weak Tie (WTS)	Traverses the weak ties
	Strong Tie (STS)	Traverses the strong ties
	Hamming Distance (HDS)	Picks neighbour with most uncommon friendlist
	Cosine Similartiy (CSS)	HDS normalized by the degree
Similartiy based	Information Scent (ISS)	Picks neighbour with highest match of profile and query

Table 3.1: Search algorithms used in [65]

Figure 3.2 shows the ratio of successful queries as a function of different search path lengths. While Breath First Search(BFS) has the lowest search path length, although Hamming Distance Search(HDS) and Best Connected Search(BCS) performs very well. The search strategies with the highest averaged search path are Random Walk Search(RWS) and Strong Tie Search(STS). Zhang and Ackermann also analyse the relevance of the out-degree of a node. They found that the out-degree is important for all search algorithms. Specially HDS and BCS have a high number of nodes with a high out-degree, because their metrics select nodes based on a high degree. Besides the computational cost and the importance of the out-degree, Zhang and Ackermann evaluate the social cost of the algorithms. The Authors introduce a metric to measure the number of people used per query. While BFS, RWS and STS use eight to nine people per query in median, BCS, HDS and CSS only need three to four people to find an expert. WTS and ISS median lies between six and five.

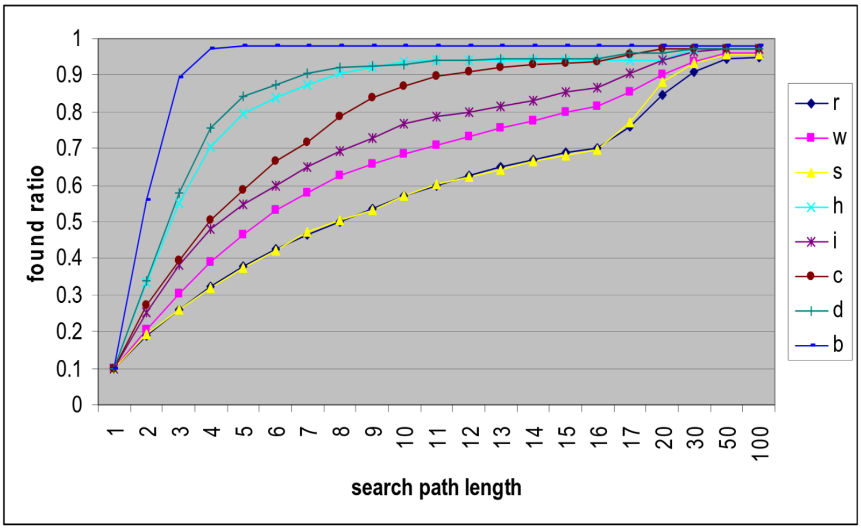


Figure 3.2: Zhang and Ackermann [65]

The results show, that STS performs worse than WTS. Zhang and Ackermann focus on the role of weak ties by searching on social networks. They remove the weak ties and run a new evaluation. The removal of weak ties lead to 23% of queries which could not be finished because of the less connected network. A second observation shows the sensitivity of the search strategies to the out-degree. The 10 users, who have the highest out-degree were removed and a new evaluation was run. A high difference was found in the average path length by BCS, HDS and CCS, which is caused by their metric for choosing a next node. All other search strategies show no significant difference.

The work of Zhang and Ackermann is relevant for the purpose of group formation, because it is necessary to search in the social network effectively. But the results are not directly transferable. In their approach, the authors try to find one expert within the network, but in our approach we are seeking a group, which is built on an impact of their relation in the social network. So a search algorithm, which tends to have a depth-first search nature, would find good candidates with a high distance to the initiating node. Another interesting contribution of this work is the method of creating of a test network. Zhang and Ackermann generate their test data from an E-Mail data set and extract the required information. This approach enables them to generate suitable real test data without implementing an own network or crawling a commercial one.

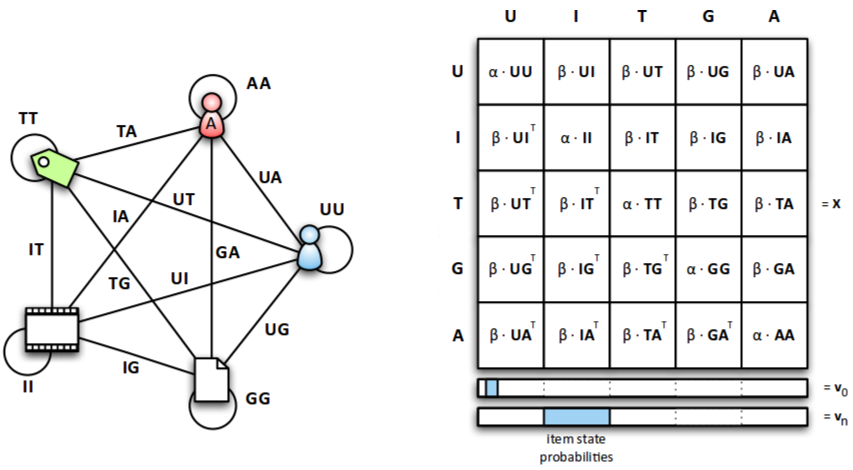
3.1.3 Recommendation Systems

When defining group formation as a recommendation task, learners are ranked in the sense of their suitability to the initiator and topic. Collaborative Filtering [66, 67] is the most promising method and an exhaustively studied research area. These algorithms perform well by discovering common interests among users, and they can forecast a rating a user would give to content items like movies or songs close to a realistic judgement. Algorithms are based on user interests. Such systems lead users to discover new content beyond their immediate interests. However, on the second look, when implementing a recommendation solution in an OSN, it becomes obvious that these systems cannot use the whole potential of the interconnections and linkage on the graph model, because they solely rely on explicit user ratings at their core. It is important to justify the reason of the recommendation, making it reliable to the context of the learning group, which can become a tremendous task in a Collaborative Filtering system.

Movie Recommendation using Random Walks over the Contextual Graph

A recommendation algorithm that is based on taggings in a social graph is introduced by Bogers[56] called Context Walk. By modelling the browsing process of a user on a movie database website, the algorithm takes walks starting from a movie, or any other entity in the graph, and browses the contextual graph, until it finds an suitable vertex and stops the browsing process. The process is extended by the possibility of self-transitions, that increase the influence of the initial state and keep the user in the vicinity of the original recommendation task. This approach is based on the work of Clements et al. [68], who introduce a personalized Markov random walk over in a social graph using tagging relations. Using this approach on recommendation, the eLearing-enabled OSN can be traversed towards relevant learners based on the interconnection of the network.

Figure 3.3 shows the contextual graph and the transition matrix used in the algorithm. The vertices in the graph are the union of users, items (movies are called item in the context of recommendation tasks), tags, genres and actors connected by different edges. The edge type is determined by the vertices involved: An edge between a movie and an actor in unweighed, indicating that the actors played a role in the movie. In contrast, a edge between user and item can have two different semantics: If it is binary, it indicates the user has watched the movie but it can also have a weight, that holds the rating of the user for the movie. In comparison to the graph in the eLearning-enabled OSN, the contextual graph shows different types of vertices representing user, tags and other domain specific entities.



$$\begin{array}{c|ccccc} & U & I & T & G & A \\ \hline U & \alpha \cdot UU & \beta \cdot UI & \beta \cdot UT & \beta \cdot UG & \beta \cdot UA \\ I & \beta \cdot UI^T & \alpha \cdot II & \beta \cdot IT & \beta \cdot IG & \beta \cdot IA \\ T & \beta \cdot UT^T & \beta \cdot IT^T & \alpha \cdot TT & \beta \cdot TG & \beta \cdot TA \\ G & \beta \cdot UG^T & \beta \cdot IG^T & \beta \cdot TG^T & \alpha \cdot GG & \beta \cdot GA \\ A & \beta \cdot UA^T & \beta \cdot IA^T & \beta \cdot TA^T & \beta \cdot GA^T & \alpha \cdot AA \end{array} = X$$

Figure 3.3: Contextual graph and the derived transition matrix of ContextWalk algorithm [56]

The walk in the graph G is a stochastic process, in which the initial state is known and the next state is governed by a probability distribution[68]. The distribution is represented by the transition probability matrix X, where the probability of going from vertex i (at time t) to vertex j (at time $t + 1$) is represented by $X_{i,j} = P(S_{t+1} = j|S_t = i)$. The weights from the contextual graph are represented by sub-matrices in X (Figure 3.3). The sub-matrices are partitioned by edge type: For instance, the matrix UI contains the weights from the user to the item vertices. Following this scheme all edges are mapped. Self-transitions are named UU, II, TT, GG, and AA and happen by the probability α. To ensure that the transitions probabilities sum to 1, the sub-matrices are row-normalized $\beta = \frac{1-\alpha}{\delta-1}$, where δ is the number of different vertex types.

To a start a random walk on G, a initial state vector v_0 is needed. This vector encodes the starting point by a 1 and remains the other values by 0. Then v_0 is multiplied by X to calculate the transition probabilities v_1 after taking on step on the contextual graph. It is also possible to calculate multi-steps by multiplying the initial state vector with X^t after t steps or by iteratively multiplying the state vector for the previous step t by the transitional probability matrix: $v_t + 1 = v_t X$. the state vector contains the probability distribution over all vertices after any steps. By removing the items already rated by the user, the rank-order of the probabilities are the recommendations for the user.

The movie recommendation approach of Borgers introduce a way of performing Random Walks on a multipartite graph. The graph of the ContextWalk algorithm matches the representation of relevant entities as vertices and weights at the edges used in the eLearning platform. A limitation of the approach is that the transition probability matrix gets very large for real world social networks and the multiplication action is not handleable. But the random walks on the contextual graph are a promising method to find candidates in the graph model.

3.2 Formal Model

Based on the requriements stated above, this section introduces the metrics used in the group formation process. To describe the graph model, the nomenclature of section 2.1.1 is used. The graph G includes the vertices $v \in V$ and edges $e \in E$. Whether a vertex v is a learner, tag, topic or content is determined by the function $T(v)$. The weight of an edge e is returned by the function $W(v)$. An overview of the nomenclature used in the algorithm can be found in Table 3.2.

Notation	Description
V	Set of vertices
E	Set of edges
$T(v)$	Function to determine the vertex type
$W(v)$	Weight of edges
$D_L(u, v)$	Distance in learning style
$D_K(v, \tau)$	Distance in Knowledge / Knowledge Rank
$P(v)$	Normalized edge weight per vertex
$A(v)$	Availability of a learner
$J(u, v)$	Jaccard Index
$D(u, v, t)$	Combined Distance
$G_{Fit}(t, g)$	Fitness function for the group g

Table 3.2: Nomenclature

The motivation of a learner to start or join a collaborative topic is modelled by the availability flag, which is true, when the learner is motivated, or false if busy or currently not interested. This flag can be set manually, or the system can automatically detect availability based on other topics a learner works on or other metrics based on learner activity. To formalize the availability of a learner, the function $A(v)$ is used, which return $true$, if learner v is available, or $false$ if not.

3.2.1 Learning Style

In the graph model, the learning style is represented as a vector $L(v)$ with an entry for each dimension of the Felder and Silverman Theory (See section 2.3.1). Possible values are 1, 0 and

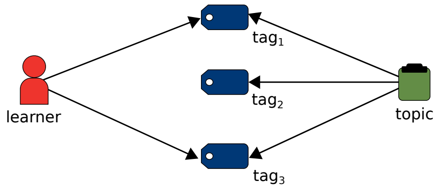


Figure 3.4: Learner Tag Topic

-1 indicating a positive or neutral or negative characteristic in each category. The learning style distance $D_L(u, v)$ between two users u and v is evaluated as the Euclidean distance between the vectors and normalized by its maximum possible value 8:

$$D_L(u, v) = \frac{1}{8} \sum_{i=0}^{4} |L_i(u) - L_i(v)| \tag{3.1}$$

3.2.2 Knowledge Rank

The knowledge model used in the group formation approch is build on tags, which are a explicit vertex type in the Graph Repository (See Section 2.3.1). A learner can tag content and topics. This action triggers the creation of an edge between a learner and a tag. A example sub graph of the relations between learners, tags and topics is shown in Figure 3.4. The tags assigned to a topic define its knowledge context. A relation between a learner and a topic is established or reinforced by each tagging action. Comparing the tags connected to a learner and a topic makes is possible to rate the common knowledge context. Shared tags indicate that the learner is familiar with the skills or background needed to work on the topic. On the other hand, tags assigned to the topic but unconnected to the learner may indicate a missing requirement for a successful participation in the collaborative work.

More formally, to calculate the distance D_K in knowledge between a topic and a user, the first step is to match the learner's tags to the topic. In the example, the tags 1 to 3 are assigned to the topic, but the learner only has a relation to tag 1 and 3. The weight of a edge, that does not exists is 0. Using the graph model, the tags of the topic t are defined as all neighbours with the type "tag" $\tau = \{E(t, v) | T(v) = \text{tag}\}$. After tags match, the correlation of the topic and the learner tags is calculated as the scalar product of the edge weight vector of the topic $W(t, \tau)$ and the edge weight vector of the learner $W(l, \tau)$:

$$D_K(l, t) = 1 - \langle W_t, W_l \rangle \equiv 1 - \sum_{i=0}^{|\tau|} W(t, \tau_i) \cdot W(l, \tau_i) \tag{3.2}$$

This value indicates, how the displayed knowledge of a learner correlates to a certain topic. Note that the normalized scalar product is 0, if the learner admits full activity in the topic, and 1, if learners activities do not overlap with the topic.

3.2.3 Distance in the Social Graph

Besides learning style and common tags, the cohesion of the group candidates in the social graph is a important factor in the group formation approach. While the algorithm aims to find a dense group, a metric is needed to describe the connectedness for the members of a group constellation in the graph model. A easy approach is to apply the understanding of a classic network community (Section 2.1.2): A ideal group is densely connected inside and has less connections to the remaining network. While the size of a learning group is set by the initiator and should be small, using this metric would rate constellations on the border of the underlying network community better and constellations inside it. So metrics considering a sparse connectivity to the outside of the group do not meet the requirement of the group formation approach. A learning group may be a small grouping of vertices in a larger community build around a common field of interest. A metric based on edges is also not suitable for the graph model, that includes learners, tags, topics and content. The goal is to rate the overall connectedness of initiator and possible group members. A metric, that only considers the edges between the learners would not cover the many indirect ties via meeting objects (tags, content and topics). When this indirect connections should be included, the problem of finding a sub graph, that includes all relevant vertices and edges occurs. A possible solution would be to find the shortest path between all constellation members, but the shortest path between the members is always 1, because they share the tag vertices, required by the topic. Another solution is to find all edge disjoint paths between all members. While this measure would give a exact picture of the connectedness, the calculation is very expensive even for small graphs.

The calculation complexity can be reduced by focusing the 1-hop-paths between the members. When switching from edge focus to vertex focus, this kind of metric measures the common neighbours of the vertices. As introduced in Section 2.1.2, Fortunato [22] uses the Jaccard Index to measure the overlap between the neighbouring vertices of vertex v and u:

$$J(v,u) = \frac{|\Gamma(v) \cap \Gamma(u)|}{|\Gamma(v) \cup \Gamma(u)|} \tag{3.3}$$

$J(v, u)$ can be applied to pair of vertices but not to groups, so it is applied to all pairs of members in the group constellation. This measure for connectivity is used in the group formation approach to describe the social distance of a group constellation.

3.3 Approach

To access the problem of group formation more easily, the group formation approach is divided in two parts. First, it searches the eLearning graph and tries to find a minimal number of suitable candidates for the formation of a group, which an initiator shaped on a chosen topic. Based on the candidates, the second part tries to optimize a constellation of collaborators for a successful group learning experience.

3.3.1 Candidate Selection

The first step of the group formation approach is the candidate selection. Its task is to extract possible group members from the underlying eLearning graph. To reduce the complexity of group formation, it is necessary to select a small set of well suiting learner vertices. Starting at the initiator, the network is searched for vertices with a common learning style and knowledge base as evaluated by Equation 3.4.

$$D(u, v, t) = \omega * D_K(v, t) + (1 - \omega) * D_L(u, v), \tag{3.4}$$

The total distance D between an initiating vertex u and a chosen topic t with a possible candidate v is calculated as the weighted sum of these two parts, where ω is a weight parameter to adjust between the relevance of of the knowledge rank and the learning style. This ‘learning distance’ $D(u, v, t)$ shall be small enough and will serve as the selection function for candidates.

The choice of the search algorithm is essential for the group formation process. Because several algorithms optimized to social networks try to find special vertices, the group distance in the social network is here relevant.

To reduce the complexity of the Candidates Selection, it can be parametrized with the maximal number of candidates and a threshold, which determines whether a node is added to the candidate set. These parameters determine the quality and complexity of the result. If the threshold is high, the candidates are near to the initiator, but may have a higher distance in the sense of learning style and knowledge. For a low threshold, the search algorithm will select nodes that have a higher distance in the social network, but are closer in the sense of learning style and knowledge. By choosing a low threshold the performance decreases.

The position of a learner-vertex is not used in the candidate selection, because in this phase of the algorithm the candidates are a loosely coupled set and no statements can be made about group membership. So it remains open what the final group density will be. It is noteworthy that the initiator plays no special role and could be the least connected part of the group.

To select the best-suited search algorithm it is necessary to take the overall requirements into account. By considering the density of group members in the social network, a team with experts on their topic at an equal learning style, but with a low density in the network does not satisfy the requirements. Also this team configuration would have high computational cost. In the following the search algorithms are listed, which are selected to be suitable in the present scenario.

Breath First Search (BF)

Breath First Search is a classic way to traverse a graph. It starts at the initiator and explores the graph by visiting each neighbour, before moving to the next level neighbours. A more memory-efficient variant of BFS is iterative deepening Depth-First Search. Instead of holding all neighbours in queue, it only operates on the neighbours of the current vertex and uses a index on the visited vertices to track which vertex neighbours should be visited next. Starting from the initiator, BFS will probably find the nearest candidates, because it traverses the social network with stepwise increasing distance.

Random Walks (RW)

Random Walks introduced by Adamic and Adar [69], traverses the social network by random paths. In contrast to BFS, the distance to the initiator in Random Walk increases very fast. In classic Random Walks the probability of which vertex to select next is uniformly distributed. This could lead to a selection of candidates who have a high distance to the initiating node. This phenomenon can be reduced by restarts.

Context Walk (CW)

In contrast to classic Random Walks the Context Walk model used in Borgers [56] and Clements et al. [50], the transition probabilities are derive from the edge weights. To have comparable probabilities for all edges of a vertex, it is necessary to normalize the weights of the edges. The weight of each connected edge e is divided by the sum of the weights of all edges.

$$P(e) = \frac{W(e)}{\sum\limits_{k \in E(v)} W(k)} \tag{3.5}$$

Now the local probabilities over all edges outgoing from a vertex at this point in time can be calculated. Based on this distribution the Context Walk can pick a edge randomly to proceed.

Node Type-based Selection (NT)

Besides the search algorithm, a simple approach can be used to select the candidates. Using the network structure shown in Figure 3.4, starting from the selected topic, all learners are selected, that share the tag vertices. This approach should find candidates, that have a high Knowledge Rank but may be less connected than candidates found by the other search algorithms.

3.3.2 Group Optimization

After a set of candidates has been selected, the next step is to find a group constellation that is densely connected and optimized according to distance in learning style and Knowledge Rank to recommend it to the initiator. To achieve this goal, a set M of all candidate groups is generated that satisfy the constraint on group sizes. Then the metrics defined in the previous section are optimized according to the entire group.

The rating of all possible combinations is just possible for small sets of candidates and group size, because the number of possible combinations increases very fast. Using the binomial coefficient, defined as

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}$$

,where n is the number of candidates and k the group size, the complexity of rating the combinations becomes more clear. Assuming there are 10 candidates for a group of 4 learner, 840 combinations have to be ranked. But when 20 candidates are selected for the same group size 19380 combinations occur and for 30 candidates 109620 group constellations. The combinations increase even faster, if larger group sizes are selected. This phenomenon shows, that the number of candidates is a curial parameter in the group formation process and rating all group constellations is inappropriate. To handle the scalability problem, Genetic Algorithms (GA) are used to find a appropriate group constellations. Besides the work of Dorn et al.[2] (Section 3.1.1), Genetic Algorithm are also employed for finding group constellations for a set of group candidates by Moreno et al. [60].

The procedure of GA is inspired by the Evolution principles of Darwin. Individuals of a population mutate their genes from generation to generation and individuals with higher fitness are more likely to survive and multiply in the next generations [63]. Mapped to a Genetic Algorithms, a set of team configurations is represented as a population of chromosomes or individuals. Each chromosome is a group of learners represented as genes. In each generation, crossover and mutation operations are performed on the population, which increase its size.

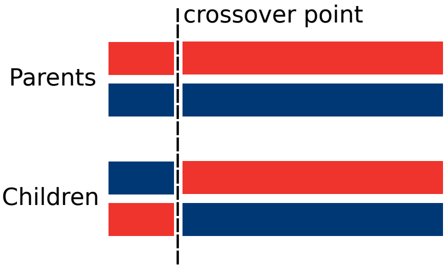


Figure 3.5: Crossover operation on a chromosome

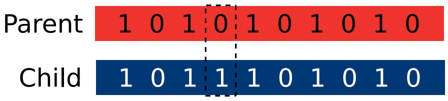


Figure 3.6: Mutation operation on a chromosome

Crossover A crossover operation generates new chromosomes by randomly selecting two from the current population according to the cross over rate. The chromosomes are split at a random point and the parts are exchanged (Figure 3.5).

Mutation A mutation exchanges only one gene in the chromosome with another. Applying this to the present approach another learner is selected from the candidate set (Figure 3.6).

After the operations, the fitness of all chromosomes is evaluated and the fittest are selected by keeping the population size constant. It is also important to check the group size, because small groups tend to have a overall smaller group distance. So all groups with a size lower then specified are removed from the population.

The fitness of each chromosome is evaluated for each suitable group $g \in M$ for a given topic t. The value of the function G_{Fit} should be minimized to achieve a overall small group distance.

$$G_{Fit}(t,g) = D_{KG}(t,g) + D_G(g) \tag{3.6}$$

The fitness of a group can be separated into two parts. The first part D_G measures the distance in learning style and the connectedness in the network by using the corresponding metrics for each possible pair of group members.

$$D_G(g) = +\binom{|g|}{2}^{-1} \sum_{u,v\in g} \{D_L(u,v) + J(u,v)\}\,. \tag{3.7}$$

The second part sums the Knowledge Rank for each group member and the topic.

$$D_{KG}(t,g) = \frac{1}{|g|}\sum_{v \in g} D_k(v,t) \tag{3.8}$$

Note that $G_{Fit}(t,g)$ is renormalized and attains values between 0 and 2. When the operations finish, the fitness of all chromosomes is evaluated and the best are selected for the next generation. After sufficiently many generations have been run, the best group constellation is recommended to the learner, who can now send invitations for joining the group to the selected candidates.

4 Implementation

This chapter starts with the system architecture and introduces the structure of the diaspora* implementation. The development of the Group Formation Engine and improvements on the Graph Repository are the remaining content.

4.1 System Architecture

The goal of the system architecture is to separate the eLearning-enabled OSN into several components according to their concerts and minimize the coupling between them. Figure 4.1 visualizes the system architecture of the eLearning-enabled OSN. The components diaspora* application, group learning, group formation and content network introduced in Section 2.3, are located in the same environment processing the request forwarded by the HTTP Interface. The joined environment is realized by using the Ruby on Rails Framework for implementation. Using the Ruby on Rails framework, it is possible to separate different components of the application into so called engines [1]. These engines are miniature applications inside a hosting application. To clearly distinguish between the applications, the engines are isolated by different namespaces. Using this feature, all components consist of their own Model-View-Controller

[1] http://guides.rubyonrails.org/engines.html

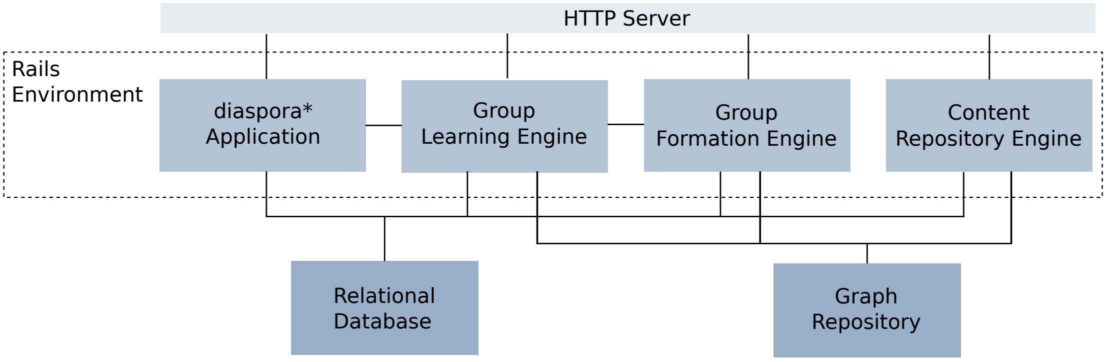


Figure 4.1: System Architecture of the eLearing-enabled OSN

structure. To store the operational data, the platform uses a relational database. The Graph Repository, used to store analytical data, is connected to eLearning related components by using the same interface. The coupling of the components are indicated by the links. The Group Learning Engine adds the eLearning specific functionality to the diaspora* Application by extending its user model. Besides this interface, the Group Learning Engine uses the run model of the Group Formation Engine to start and manage the group formation process. The Content Network developed by With [54] does not have a coupling to other engines. To access the functionality of the different components a learning sidebar is included to the overall layout that includes hyperlinks to the relevant views of the engines. Having little changes in diaspora* simplifies the installation and update process. This is achieved by adding hooks in the loading routine of the engine, which add the migration, locales paths and other dependencies to the main application.

4.2 Diaspora* Pod Architecture

Diaspora* consists of a network of distributed instances, so called pods. All pods are equal peers in the network and use the same interface. The architecture of a pod follows a classic 3-tier architecture: Database, Server and User Interface. Diaspora* is created using Ruby on Rails, which is a Model-View-Controller (MVC) framework, providing default structures for a database, web service and web pages. Besides MVC, Rub on Rails emphasizes the use of other software engineering patterns, including convention over configuration (CoC), don't repeat yourself (DRY), and the active record pattern.

4.2.1 Database

The database stores the models created in diaspora*. While all migrations are handled by the Ruby on Rails framework, the database needs less configuration and maintenance. Per default, the framework maps the model to a table in the database and creates a migration script. Possible implementations are MySQL[2] or PostgreSQL[3]. Thus, diaspora* recommends MySQL, it is chosen as the transactional database for the eLearning-enabled OSN.

4.2.2 Application

The application handles the incoming HTTP requests and passes them through the Rails framework to the application code. Requests can come from users or other pods. The incoming

[2] http://www.mysql.com/

[3] http://www.postgresql.org/

requests are mapped to a corresponding controller following the REST paradigm [70]. The inner of diaspora* is designed according to the Model View Controller Pattern. Here the model represents an entity and handles the storage in the database. A view generates a presentation of a model using predefined templates. The manipulation of models and rendering of views is handled by the controller. Besides the Web interface the server component of diaspora* handles the messages exchanged between the pods using the Salmon protocol[4].

4.2.3 User Interface

The User Interface (UI) is implemented on the server and client side. Static content is rendered at the server and sent as HTML code. Posts and messages are queried as JSON Objects by Javascript using backbone.js [6] and handlebars [7]. Both approaches are not clearly separated from each other.

4.2.4 Background Jobs

Diaspora* uses background jobs to distribute messages to other pods and send E-Mails to the users. To do this, is uses the Sidekiq[8] gem. The Sidekiq client runs in the container application and allows to create defined workers and push them into different queues. The key value store Redis[9] is used to manage the queues and workers. Jobs are pulled from the queue by the Sidekiq server and processes them. Thus the server is started in the Rails application, the server has access to the full application API.

4.3 Graph Repository

The Graph Repository is responsible for storing the learning objects according to the scheme defined in Section 2.3.6. During the development of the Group Learning Engine, the Graph Repository was implemented by the Rexster graph server and a custom client for its API. Rexster is part of the TinkerPop2 [10] graph computing framework and provides a REST API to different graph database implementations. Rexster was chosen, because its intern database can be configured according to the applications requirements and it was developed using the

[4][5]

[6] http://backbonejs.org/

[7] http://handlebarsjs.com/

[8] http://sidekiq.org/

[9] http://redis.io/

[10] http://www.tinkerpop.com/

popular Blueprints[11] interface. The reason for changing to another implementation is the lack of an import interface and that the REST API is assigned as deprecated in the newest version of the Tinkerpop Framework. Also the usage of the querying interface was not well suited for the usage in the eLearning-enabled OSN.

After re-evaluating the possible implementations of the Graph Repository, the graph database Neo4j was chosen. It uses an own language to import and query the database, called Cypher and also supports large batch imports by an own import tool. Another advantage of choosing Neo4j is comparison to Rexster is its well-supported client libraries. The neo4j-core[12] gem is used in this thesis to insert vertices and edges and query the Graph Repository.

4.4 Group Formation Engine

The Group Formation Engine implements the group formation approach introduced in Chapter 3. It operates on the Graph Repository and is triggered by the Group Learning Engine. The first use case of the engine is that a learner browses the topics on the platform and decides to start a group on a certain topic but do not know suitable peer learners. At this point the Group Formation Engine starts to search the Graph Repository for candidates. In the Candidate Selection, the engines traverses the Graph Repository and measures a score at each learner vertex. If the score lies under a given threshold, the vertex is added to the candidate set. If a given number of candidates is found, the Group Optimization is started and candidates are grouped and the constellations are optimized using Genetic Algorithms. The different group constellations are suggested to the learner. And group invitations are send to the learner in the initiator selected group.

4.4.1 Requirements

Based on the use cases specified above and the objective to scale to real world social networks, the following functional and non functional requirements are formulated. The functional requirements are:

1. The learner can explicitly trigger a group formation run to find a constellation for a given topic.

2. The learner can specify the group size for the desired constellations.

[11] https://github.com/tinkerpop/blueprints/wiki
[12] https://github.com/neo4jrb/neo4j-core

3. When group constellations are found for a specific topic, they are shown in the topic view of the group learning engine.

4. Learners can select the preferred constellation and group invitations are sent to its members.

Besides the manual start of a group formation run, runs are automatically started.

5. The Engine starts runs from random learners with topics that share tags with the learner.

To analyse the performance of the group formation engine a separated interface can be used.

6. All runs can be listed via a separated administrative interface.

7. The results and parameters of each run can be displayed in the administrative interface.

Scalability and fault tolerance are addressed by the non functional requirements:

9. The process of finding constellations should be separated in small independent building blocks to enable an easy distribution.

10. Exceptions in a single run should not influence the stability of the Group Formation Engine.

4.4.2 Data Model

The data model of the engine captures the identifiers of the involved entities and the parameters of the algorithm (Figure 4.2). The parameters of the algorithm are included in the database and are not separated to a configuration file, because the results of the algorithm highly depend on its configuration and the differences should be comprehensible over time. To identify every entity, they hold an id as primary key. The Run entity serves as the starting point for the Group Formation algorithm. It holds the threshold, count of candidates as well as the key for the topic and initiator. The search strategy can also be selected. The detected candidates are stored with their keys and ranks for knowledge, learning style and combined distance.

An Optimization entity holds the parameters for the Genetic Algorithms (Section 3.3.2)as well as the group size and a relation to the origin Run. The found Constellations are stored with their fitness and social distance together with its Constellation Members.

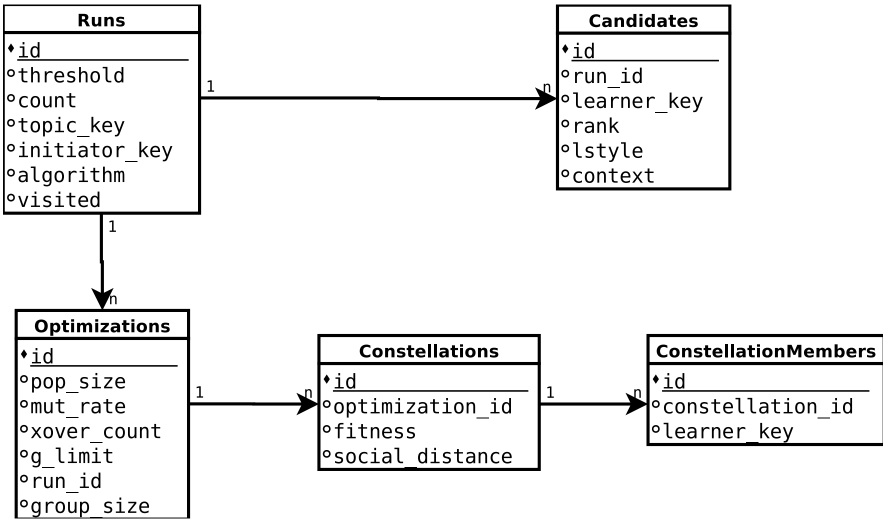


Figure 4.2: Data Model of the Group Formation Engine

4.4.3 Software Overview

The Group Formation Engine has two components, which are connected by sharing the models. The REST interface servers creation and evaluation actions and the workers perform the group formation process.

Following the Ruby on Rails MVC structure, the code artefacts in the Group Formation Engine can be categorized into Models, Views and Controllers (Figure 4.3). The models correspond to the entities of the data model.

Serving a REST interface, the task of the MVC strucutre is to create and manage group formation runs. The interface is logically divided into the run and optimization controller, that serve the functionality for the learner and the evaluation controller, that serves adminitrative needs like listing all runs and showing their details. The actions of the evaluation controller are only usable for users with diaspora* admin role and only these users are able to delete runs, optimizations and constellations. Learners are able to create runs and optimization and view their results.

To initiate a group formation run via the REST API, a new run model is created through the run controller using the new view and create action and a new worker is created.

4.4.4 Workers

The tasks of finding candidates and group constellation cannot be completed within the response cycle of an HTTP request. To decouple these task, the Group Formation Engine takes

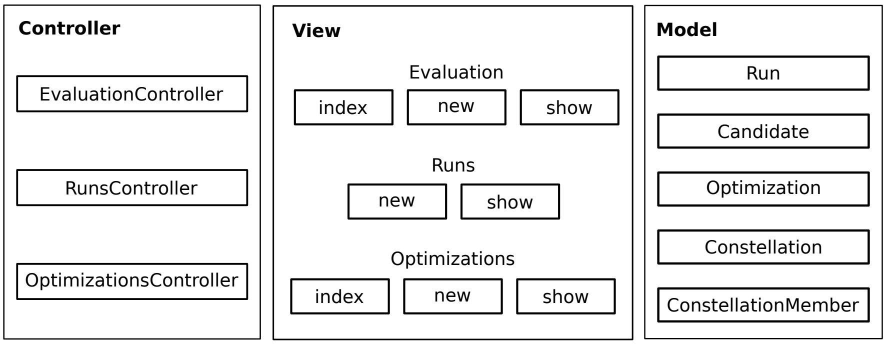


Figure 4.3: MVC Entities of the Group Formation Engine

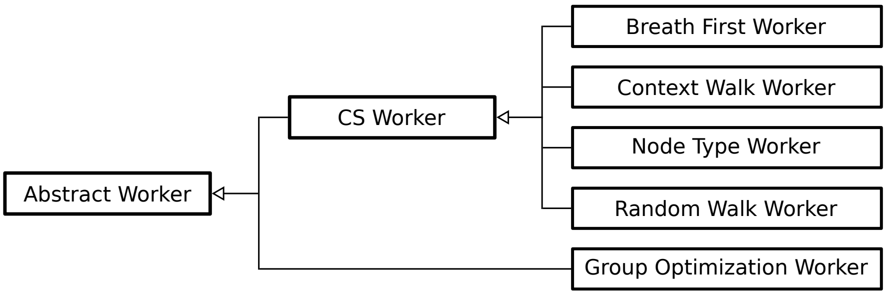


Figure 4.4: Inheritance structure of the Workers

advantage of the existing background processing in the diaspora* application. Here, the sidekiq gem is used to handle asynchronous jobs, that serves a distributed management environment and API to easily create and run workers. The setup of the Group Formation Engine uses two custom queues: The Candidate Selection and Group Formation queue to control the availability of free workers for each phase of the algorithm.

The inheritance structure of the worker classes in the Group Formation Engine can be found in Figure 4.4. The Abstract Worker is the parent class of all workers. It includes methods for loading the tags of the topic and getter methods for the initiator and topic. Also the implementation of the knowledge rank is located in the Abstract Worker. To calculate the knowledge rank, it is necessary to select the tag vertices that are shared by the topic and the learner. In the first version, all tags from the learner where loaded and then matched to the tags of the topic. Using this approach, irrelevant tags are loaded an a unnecessary large list of tags has to be iterated. To avoid the loading of unneeded data, only the tags are loaded from the Graph Repository, that are shared between the learner and the topic. The improvement of the data loading strategy increases the complexity of the database query but decreases the time needed to calculate the Knowledge Rank for one topic learner pair.

The CS Worker directly inherits from the Abstract Worker, that holds the common methods for all candidate selection strategies, like setting up the worker and a method for writing the candidates to the database as well as a method for checking if a learner is suitable candidate according to the threshold. The workers derived from the CS Worker hold *preform* methods according to their strategy:

Breath First Worker Classic implementations of Breath First Search hold a queue of the all explored neighbours that are not visited yet. In context of a large social network, this queue may become very large in a small amount of time. Besides its size, at each vertex, all neighbours have to be loaded from the Graph Repository, which can be a bottleneck when large edge lists have to be loaded. To increase the performance of the Breath First Search, in this implementation, the queue is only filled, when it is empty. An additional index on the list of visited vertices indicates which neighbours have to be loaded next. It is incremented after each refill action. Using this approach, the queue size remains small and neighbours do not have to be loaded at each new vertex.

Context Walk Worker The Context Walk Worker selects the next vertex according to the edge weights. The random selection is handled by the PickUp[13] gem. To calculate the probabilities it necessary to load all edges and neighbours. As argued above, loading all neighbours at each vertex becomes a bottleneck for large networks. At the Context Walk selection strategy it is needed to calculate the relative weight for each edge. To optimize performance, the data to load can be minimized. Instead of loading the edges and the neighbours, a custom Cypher query is used to only load the weights and neighbours ids. This does not affect the quantity of data but the size of each entry.

Node Type Worker The Node Type Worker selects the Tags from the Topic and then aggregates the connect Learners. The list of Learners is iterated, to check which learner is a candidate.

Random Walk Worker Similar to the Context Walk Worker, the Random Walk Worker selects the next vertex randomly, but a uniform distribution and not the edge weight. This eliminates the requirement of loading the neighbours to the worker and the selection process can be handled in the Graph Repository. To implement this using a Cypher query, each neighbour obtains a random number and the list of neighbour and random number is ordered by the number. The next vertex to visit is the first in the list. This

[13] https://github.com/fl00r/pickup

relocates the processing from the worker to the database and reduces the data to only one vertex entry.

Group Optimization Worker The implementation of the Group Optimization follows the same pattern as the Candidate Selection. The start is triggered by the creation of a Optimization object and a GroupOptimizationWorker is created and started. The implementation follows the description in section 3.3.2. Each chromosome is a random combination from the candidate set with a group size decremented by one. The initiator is not a represented in the chromosomes but is added only in the fitness evaluation, so it is ensured that he or she is present in the group constellation.

The social distance metric, implemented in the Abstract Worker, measures the shared neighbours of a group constellation, which requires to load all neighbours of all constellation members. Thus the metric is only used for the candidates in one optimization, a cache was implemented, that stores the neighbours of a learner after they were loaded once.

5 Evaluation

The objective of this thesis is to answer the question how to stimulate a team building process that is effective for learners. While the previous two chapters introduced the concept and the implementation of the team building process, this chapter aims to show the effectiveness for learners. As a pre-study to real-world deployment of the eLearning-enabled OSN, an evaluation of the Group Formation Engine is preformed. The questions answered in the evaluation are:

1. Does the implemented group formation approach operate compliant with its requirements?
2. What is a effective algorithmic parametrization according to performance and group fitness?
3. How is the quality of the groups is sense of fitness, stability and similarity to real groups?

The first question concerns if the requirements of the approach are compliant by the implementation. To do so, the scores for the distance in learning style, knowledge rank and social distance of the members of the found group constellations are evaluated.

The group formation approach introduced parameters which may have an influence on the quality of the group constellations as well as on the performance. The second questions investigates these influences by comparing the search strategies and its parameters as well as the parameters used in the Genetic Algorithms.

The group quality is issued by the third question. The different perspectives on quality are the fitness values, stability of the group constellations and the Knowledge Rank and social distance in comparison to groups in the evaluation data.

Evaluating the Group Formation Engine raises the problem of proper test data. The ideal method would be to motivate a reasonable number of learners to join the eLearning-enabled OSN and use the data learners generate. However, it is not in the scope of this theses to run such an evaluation. So suitable synthetic or empirical data has to be used. Suitable means that the evaluation data has similar characteristics as data created during real usage of the eLearning-enabled OSN. While the group formation approach traverse the Graph Repository the structure of the graph is relevant. The graph structure consists of the following dimensions.

- Distribution of the different vertex types
- Interconnection of the vertex types
- Range and distribution of edge weights
- Overall size of the evaluation graph

There are several models for generating social graphs with real world features (for instance [71, 14]). Previous evaluation of the group formation approach [4] used a synthetic graph consisting only of learners. The other entities were not mapped to the graph. By introducing the eLearning graph model, the network became more complex by including different vertex types. This complexity exceeds the synthetic graph models and emphasizes the need of empirical evaluation data. The challenge by using empirical data is to find a data set, that matches the model of the eLearning Graph or transform a suitable data set to the desired data model. This Chapter starts with a report on how the empirical evaluation data set was created based on the data available on the Stack Exchange platform and how the learning style was assigned to the learner vertices. The evaluation continues with answering the evaluation questions.

5.1 Stack Exchange

Stack Exchange [1] is a popular Question Answering network hosting more than 130 different sites related to individual issues. On Question Answering sites, users can ask question to a specific problem and receive different answers from the community. In the case of Stack Exchange, different answers are discussed, commented and rated, which leads to a high quality.

The concept of Stack Exchange sites matches our eLearning-enabled OSN in several aspects. While both platforms are based on collaborative problem solving, there is a different motivation for collaboration: The creator of a question on Stack Exchange has a specific problem instead of motivation to learn and discover something new. Also questions are only answered once but the topics are designed to be reused by several groups.

Users, involved in the discussion process by posting questions or answering and commenting on them, form a loose group, whose common attribute is the participation in the question answering process. This kind of temporal user groups maps well to the concept of learning groups.

Another similarity aspect is the tagging system. Posts (questions and answers) on Stack Exchange are tagged by their creators. These tags help the users to find questions, they may can answer and to describe the context of the question.

[1] http://stackexchange.com/

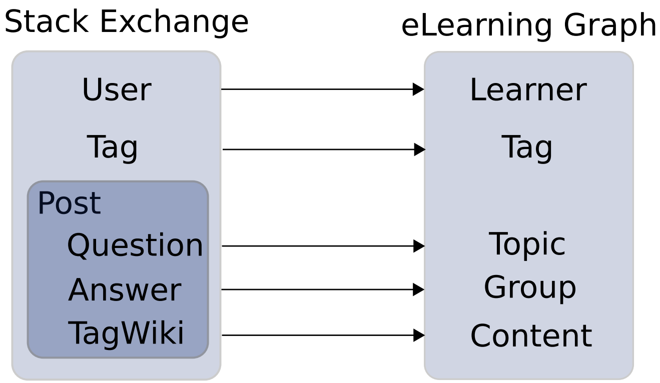


Figure 5.1: Mapping of Entities from Stack Exchange to eLearning Graph

Most of the tags used on Stack Exchange have a info page, that describes it. These info texts, maintained by the community, also include other tags. These info pages are mapped to the content vertices in the network.

Besides the common aspects, there are differences in the concept of Stack Exchange and the eLearning-enabled OSN. While most of the differences could be solved by transforming the Stack Exchange data, a essential difference still exists: Each question has only one answer threat. Mapped to the present platform, there is just one group per topic. This limitation has to be considered in the evaluation. Also data describing the motivation or availability of a user is not included in the data set.

5.1.1 Mapping to eLearning Graph

In the following the mapping of the entities to the vertex types of the eLearning graph is formulated. A overview can be found in Figure 5.1.

User On Stack Exchange, a registered user can ask questions and give answers or comment on both. While the site focuses on Q & A, users cannot establish a direct relation between each other explicitly, but ties are created indirectly by answering questions or commenting on posts. The user can be easily mapped to the learner vertex. In the eLearning graph, the weight of the edges between learners is grounded on the communication artefacts exchanged. Each artefact exchanged increments the weight. Mapped to the Stack Exchange world, the artefacts are posts and comments. An edge is created if a user replies to a question of another user or comments on a post. The weight is increased by each event. The requirement of a learning style representation can not be fulfilled using Stack Exchange data but is addressed in the next section.

Post The post is the central entity in the Stack Exchange data model. It covers all types of content, users can create on the platform. The different types are *Question*, *Answer*, *Wiki*, *TagWikiExcerpt*, *TagWiki*, *MonderationNomination* and *WikiPlaceHolder*.

The posts of the type question contain the problems or issues the users publish to the community. While the motivation to formulate a question and creating a topic in the eLearning graph is different, they are both the starting point of collaboration. Thus, the questions are mapped to the topics. The corresponding groups are the aggregated users, who participate in answering the question. So the group vertex can be created and the edges from the group to the topic and from users to the group are mapped accordingly. The weights of group to users edges are determined by the amount of posts and comments of the user in the answering process.

Besides in the question answering process, the post entity is used to maintain a wiki. The wiki includes descriptions for the tags, that can be used to specify the context of a post. While the textual content of the wiki posts includes tags, it is possible to extract the relations between the different tags respectively their wiki posts. These relations are used to map the content network in the eLearning graph: The TagWiki posts are mapped to content vertices and the tag relations are used for the content to content edges.

Tag Tags help the user to specify the context of a post and can be mapped one to one to the tags in the eLearning graph. The user to tag edges refer to the taggings in the user's post and the quantity corresponds to the weight. Using the same approach as for content vertices, the content to tag edges are extracted from the textual TagWiki posts. In the eLearning graph, topics also have connections to tags. Here the tags of the question post are used. A post can be tagged with a maximum of 5 tags. The weights are determined by the reversed index in the tag list.

Editing Relations The edges between user and topic as well as user and content indicate a creation or editing action in the eLearning graph, where the weight represents the number of these actions. In Stack Exchange posts can be edited by every user and the history of the posts is tracked, so these editing relationships can be transferred, because content and topics are mapped by posts.

5.1.2 Transforming the Data

The Data of the Stack Exchange network, can be queried via a web interface [2]. It enables querying a snapshot of the Stack Exchange database using SQL. The created queries can include parameters and can be shared to get insights into the data. While the evaluation aims to use the whole data and transform it to the eLearning model, the SQL interface can be a bottle neck. Besides the Web Interface Stack Exchange also provides their data in zipped XML-Files, which can be download[3]. The XML dumps include all Stack Exchange member sites, but the evaluation just focus on the Mathematics and Superuser site, that are the second and third largest members. They have a large data set that is also manageable with the resources available in this thesis. The data provided is organized in several large XML Files. To import the files of the Mathematics and Superuser site of Stack Exchange into the Graph Repository and run the evaluation, the data has to be extracted and transformed. The files, that include the relevant data for the graph model have a size of 4.8 GB and 2.8 GB unzipped.

The first step of the transformation is to load the data in an environment, that enables data selection and reorganization. A installation of a MySQL DBMS was used to import the XML Files via the build-in XML import command. After the importation, the database has the scheme shown in Figure 5.2. The data for badges and votes is not loaded in the database, because it is not relevant for the transformation to the graph model. The scheme approaches from Posts, PostHistory, Comments, Users and Tags.

Entities

Users and Tags are adopted one to one. From the Post table, the Topic and Content tables are derived according to the mapping above. To create the Group table, the Post entries answering one question are aggregated and a relation table between the Group and the User table is created that hold the amount of Post for each question.

Relationships

The one to many relations between the entities are modelled by foreign keys in the tables. The many to many relation of taggings between the Post and Tag entities is modelled in a different way. While Tags are listed in a explicit table, the taggings are represented by a list of the names of the Tags in the Posts table. To extract the tags from the field, a Ruby script was used. The script caches the 1,212 and 5.112 tag entries from the data sets and iterates the Post

[2] http://data.stackexchange.com/
[3] https://archive.org/details/stackexchange

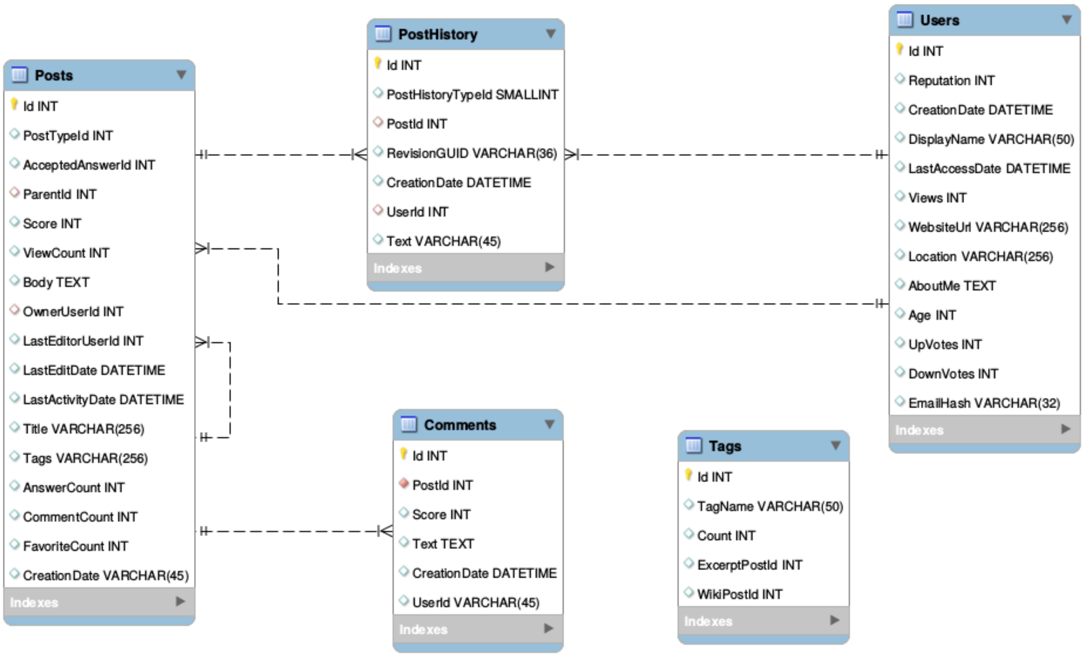


Figure 5.2: Database Scheme of Stack Exchange after XML file import

table. For each Post, the tags from the Tag field are extracted. For the relation between users and tags a entry in a temporary table with the user and tag id is created by using the script from above. When the script is finished, the entries for the same combination of user and tag id are grouped and counted. The result set is stored in a separate table. The same procedure was used for the topic to tag relation. To extract the tags from the content posts, the body field of the post was parsed, because the tags are encoded by hyperlinks to their wiki pages in the description text and not explicitly listed in the tag field. These links are extracted and used as the tag field in the script named above.

Content to Content The relation between content, defined by With [54], are reasoned from a content network and a reasoning engine. In this evaluation, the relation is mapped to the taggings between tag wiki pages.

Content to Topic With also introduced a explicit relation between content and topics, that shows content recommendation for a topic. In the evaluation data, this relation is deviated from the tagging relation between the question tags and the corresponding wiki pages.

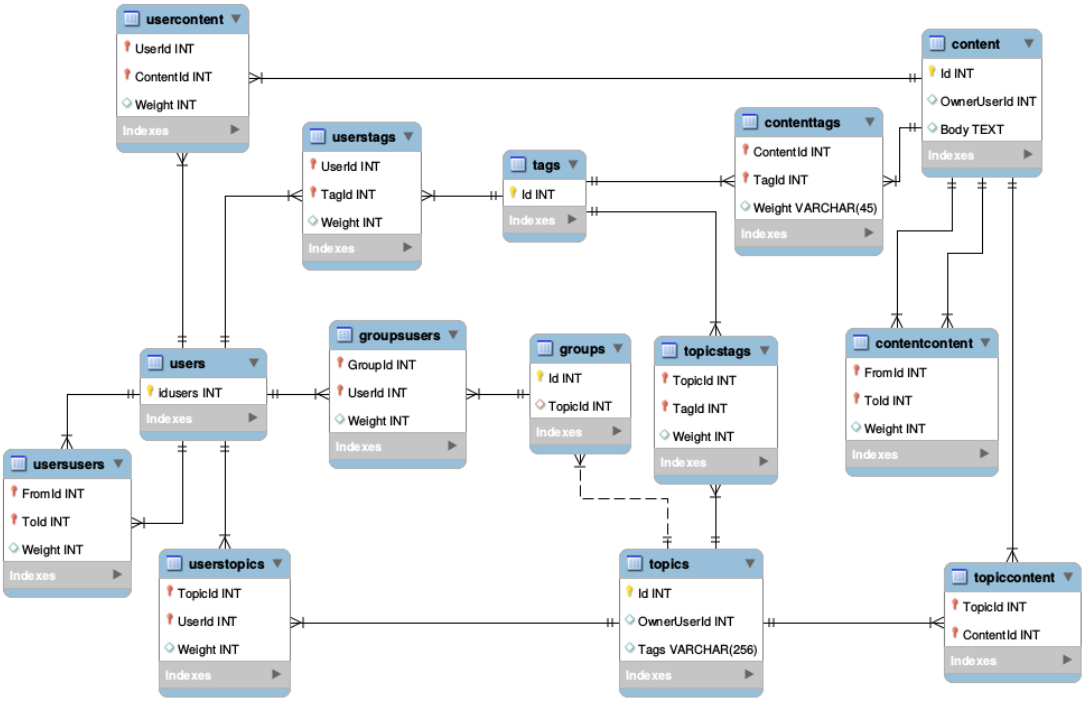


Figure 5.3: Transformed database scheme of Stack Exchange sites

User to User As sated above, the user to user relation is based on the amount of post responds and comments on posts from one user on the other's. For the relation via posts, the owner id of the question and of the answer post are joined and grouped. This grouping is also done for owner ids of comments joined to posts.

Figure 5.3 shows the data scheme after the transformation process. It includes all graph model entities and the edges connecting them in separate tables.

To transfer the data to the graph database, the tables are exported to CSV files. Using the Neo4j batch importer, the graph is created in the Neo4j specific database scheme. The amount of edges and vertices in the raw data and in the Graph Repository vary because only data, that holds all necessary attributes was transferred.

5.1.3 Properties of the Mathematics and Superuser data sets

The data from the Mathematics forms a graph with 1.081.473 vertices and 6.180.564 edges and the Superuser data includes 863.775 vertices and 3.603.825 edges. The distribution of the vertices and Edges according to their type can be found in Table 5.1.

Vertex Type	Mathematics	Superuser
Content	1.105	2.761
Group	405.750	230.184
Tag	1.212	5.112
Topic	473.746	278.046
Learner	199.660	347.652
	1.081.473	863.755
Edge Type		
Content → Content	312	806
Content → Tag	312	1.032
Group → Learner	1.461.598	796.433
Group → Topic	405.750	230.184
Learner → Content	1.904	4.648
Learner → Learner	1.336.155	814.923
Learner → Tag	535.436	653.670
Learner → Topic	463.295	264.742
Topic → Content	984.710	35.640
Topic → Tag	991.092	801.747
	6.180.564	3.603.825

Table 5.1: Quantity of vertex and edge types in the evaluation graphs of Mathematics and Superuser

The Tables shows that the distributions of the vertex types are very different on the Mathematics and Superuser site in the Stack Exchange Network. The Superuser site has 4 times more tags than the Mathematics site while the overall amount of vertices is smaller. Another difference is the ratio of learners and topics or groups. While in Mathematics, there are twice as much topics as learners, in Superuser are 100.000 more learners than topic. This indicates that the learners in Mathematics are generally more active. In contrast to the eLearning-enabled OSN where topics can have several groups, on Stack Exchange a question is only answered once, so this ratio would be different in a real world deployment. The share of Content vertices is the smallest in both evaluation graphs. The amount is derived from the number of tags, because the wiki entries of the tags were used to create the Content vertices.

The activity of Mathematics users is also visible in edge distributions. Even if the number of learners is smaller in Mathematics, their connectivity by learner to learner relations is much higher than in the Superuser site. In contrast the Superuser site shows a higher characteristic for tagging and diversity in tags, indicated by the amount of tag to learner and topic to tag relations.

5.2 Learning Style

The learning style cannot be derived from the Stack Exchange data. So another way to assign the learning style to the learner vertices has to be found. A random assignment of the different dimension preferences in the Felder and Silverman learning style theory could lead to a test network with unrealistic characteristics, as was indicated by findings of Derntl and Graf [72]. The authors started from a blog as a learning diary to a course and tried to identify correlations between the blogging behaviour and the learning style of the students. By comparing the blogging behaviour and the active reflective dimension, they found a correlation to the number of blog posts. Active learners tend to write more blog posts than reflective. On the other hand, reflective learners read more posts than active. In addition active learners tend to follow the chart of rated blog posts because of their social orientation. These findings indicate a correlation between the degree of a user in the social network and the value in the active resp. reflective dimension of the learning style. While the authors state this correlation, it is not possible to transfer it to the evaluation data because of missing data.

Another problem in assigning learning styles is to choose the preferences of dimension values. Felder and Spurlin [73] collect several studies of measuring the learning style of students using the Felder and Silverman theory. Figure 5.4 shows the average of learning styles preferences for each dimension. In this thesis, these averages are used to represent the learning style of the learners. For each learner vertices the dimension preferences are randomly selected based on the probability distribution for each dimension. The limitation of the missing correlation of learning style and position in the social graph have to be considered in the evaluation. To reduce the complexity of the evaluation, here the learning style is a property of the Learner vertices. In the production mode, the learning style is stored at the Learner model in the Group Learning Engine.

5.3 Evaluation Environment and Experiment

The evaluation environment consists of a database server running a Neo4j database instance, and an application server running the MySQL database and a Ruby on Rails application with the Group Formation Engine mounted. To analyse the results, the statistical programming language R is used to extract the data from the MySQL database and perform the evaluation. To generate runs of the Candidate Selection, a worker for Sidekiq which starts a candidate selection process and sleeps for 1 minute was implemented. There are two different kind of starts. Either 4 runs, each using a different search strategy, or 4 equal parametrized runs are started. The parameter for the runs of the Candidate Selection and Group Optimization are

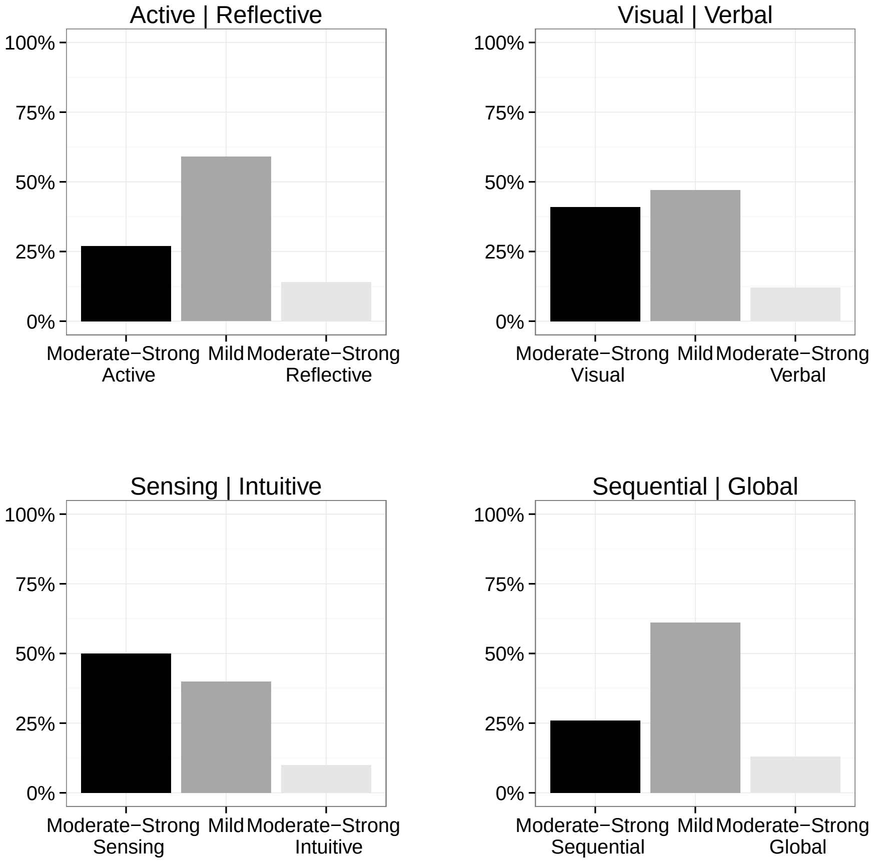


Figure 5.4: Average of learning style preferences based on studies collected by Felder and Spurlin[73]

selected according to Table 5.2. Just like the Candidate Selection runs, optimization processes are started in a separate worker every 3 minutes. Based on a finished Candidate Selection run the parameters are selected randomly. Using this approach 53.759 runs for the Mathematics and 35.970 for the Superuser data set were performed. In the present experiment it was assumed that the learners in the network are always available for collaboration.

Phase	Parameter	Range
Candidate Selection	Learner	random
	Topic	random, connected to learner
	Candidate Count	1 - 30
	Threshold	0.3 - 0.9
	Search Strategy	BF, CW, NT or RW
Group Optimization	Population Size	10 - 100
	Mutation Rate	0.1 - 1
	Cross-over Count	0.1 - 1
	Generations	1 - 100
	Group size	2 - Candidate Count

Table 5.2: Parameter and their Range used in the Candidate Selection and Group Optimization

5.4 Requirement Check

The first question in the evaluation is, whether the implemented group formation approach operates compliant with the requirements formulated in Chapter 3.

1. Each learner is motivated to collaboratively learn on a certain topic.
2. The learning style of a learner is balanced among the group.
3. The background on the topic is compatible among group members.
4. Group members are well connected in the underlying network.

The data of Stack Exchange does not include any indicator with regard to the motivation of a learner to join a group. The evaluation will skip this first requirement as it may not affect the group formation essentially. It is assumed that all learners in the network are motivated for collaborative learning. Adding unmotivated learners would increase the number of vertices that have to be visited in order to find the defined number of candidates. Different metrics were introduced to quantify the requirements. A compliance of the requirements can be shown by evaluating the distribution of the metric scores measured for the members of the group constellations found in the experiment.

The Cumulative Distribution Function (CDF) of the metric values scored by the learners in the found group constellations is shown in Figure 5.5. The distance in learning style is here applied to the initiator of the group and each member. The score measured in the Mathematics and Superuser data sets is overall small. The maximum value is 0.3 and the smallest value 0

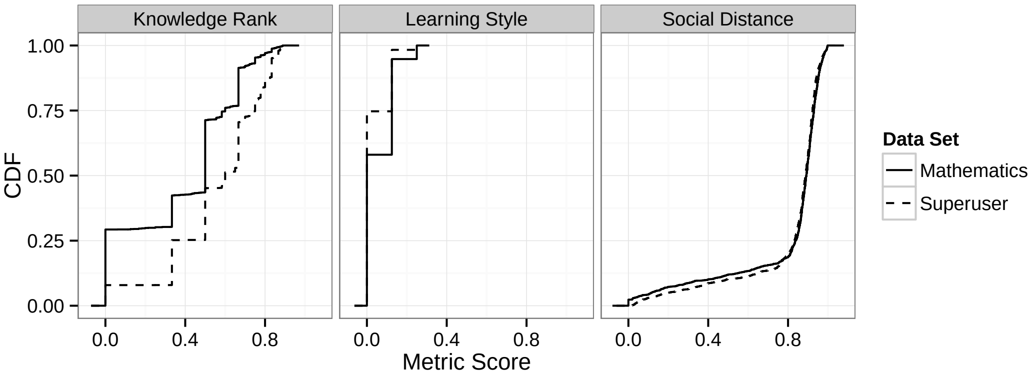


Figure 5.5: CDF of the Scores for Knowledge Rank, Distance in Learning Style and Social Distance of the Members in the Group Constellations found in the Experiment for the Mathematics and Superuser Data Set.

is measured by 60% of the initiator candidate pairs. The steps in the CDF are based on the 3 possible values for each dimension in the Felder & Silverman model. While a balanced learning style is required for the learning groups, the small scores show that this requirement is met by the distance in learning style.

The condition for a compatible background on the topic is addressed by the Knowledge Rank. Here the scores represent the correlation of the tags the learner as well as the topic are related to. The scores for the learners in the Mathematics and Superuser data set have accumulations at 1/3, 1/2 and 2/3. These steps are grounded in the amount and weights of the Topic and Learner vertices. As stated above, Topics can have 5 Tags at maximum. In the Mathematics data set all Topics have at least one Tag, 65% have 2 Tags and 30% have 3. This leads to relative weights of (1/1), (1/3, 2/3) and (1/6, 2/6, 3/6).The degree distribution of tag vertices per Learners shows similar relative weights as the topic vertices. Most learners have a tag degree under 5 even though some learners have a lot of tags assigned, too. In the calculation of the Knowledge Rank the relative weights are multiplied resulting in the peaks. Comparing the scores for the two data sets, the Mathematics scores are more pronounced at the peaks and generally lower indicating a better connection of learner, topics and tags. The difference is maybe caused in the different amount of tags in both data sets. Superuser has 5 times more tags than Mathematics but less topics resulting in a more diverse context of the topics. However, the scores of the Knowledge Rank show that the approach is able to find learners that share the tags with the topic and also have scores in the bottom half of its scale indicating a high relation of the tags and its weights for the topic and the learner.

To measure the connectedness of the members in a group constellation, the social distance metric was introduced. In contrast to the distance in learning style and the Knowledge Rank, the social distance is not measured for a member of a group constellation, the initiating learner and the topic, but for a whole group constellation. The scores for the social distance show a high peak by values from 0.8 to 0.9 for both data sets. But besides the peaks, scores are measured for the whole possible range. While the social distance measures the overlapping of the neighbours for two learner vertices the high scores could be grounded in the comparison of vertices with a large list of neighbours. Even if the scores are high for the social distance metric, scores under 1 indicate that the ego-networks of the learner at least overlap with some common vertices. The scores of the metrics are influenced by the threshold parameter that determines the upper limit of adding a learner to the candidates. In the present experiment the threshold parameter was sampled on its range from 0.3 to 0.9, in the production the parameter could be set to a small value and is able to find sufficient candidates. The scores measured for members of the found group constellations show that the evaluated requirements are full filled according to metrics capturing the aspects of balanced learning style, compatible background and connectedness of the learners in the social network.

5.5 Algorithmic Parametrization

In the experiment Candidate Selection and Group Optimization are started with different configuration from the range of their parameters. This section evaluates with parametrization according to performance and group fitness.

5.5.1 Search Strategies

The parameters in the Candidate Selection are the threshold, the number of candidates and the search strategy. While the threshold and the number of candidates are used to limit the run time of the algorithm, the search strategy is essential for the performance and quality of the Candidate Selection. To evaluate the quality of the candidates found by each search strategy the weighted combined rank of distance in learning style and Knowledge Rank are compared for the search strategies in both data sets (Figure 5.6).

Both data sets show the same order of search strategies according to the scores for the combined distance. Note Type-based selection (NT) selects the candidates with the lowest scores from the eLearning graph followed by Breath First Search (BF). Both CDFs overlap at certain points. The separation of both distributions is more present in the Superuser data set. Candidates found by Context Walk (CW) and Random Walk (RW) generally score higher at

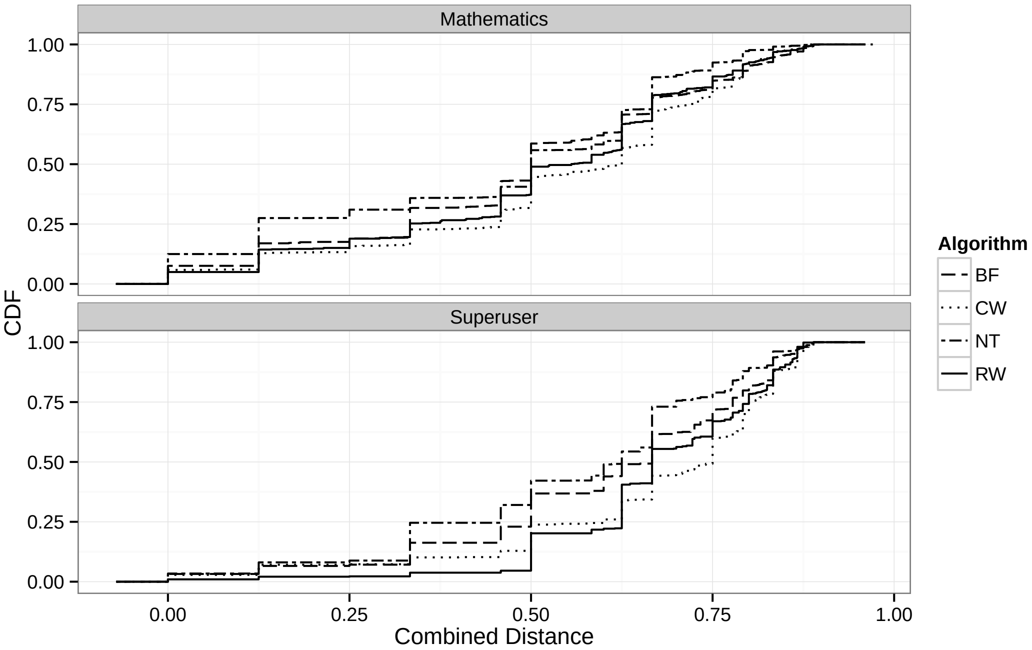


Figure 5.6: Combined Distance of the Candidates found by each Search Strategy in the Candidate Selection of the Experiment. Breath First (dashed), Context Walk (dotted), Node Type (dash-dot) and Random Walk (solid)

the combined distance. Comparing the combined ranks for both data sets, the Superuser data scores are higher plus results for the individual search strategies are more separated from each other. In conclusion, the best candidates according to the combined distance, used in the Candidate Selection are found by the Node Type based selection.

Besides the quality of the found candidates, the performance of the different search strategies is relevant to decide which suits best to the problem of finding a set of special vertices in a complex graph, especially in the relation to the number of vertices to be found. To analyse the performance of the search strategies, the number of vertices visited in each run is evaluated for possible parameter values for threshold and candidate count compared by the search strategy. This enables to profile the performance for each strategy and shows the influence of the parameters in the Candidate Selection. Fig. 5.7 shows the vertices visited by the different strategies to find the number of required candidates. The Context Walk strategy shows no raise of vertices visited when the candidate count raises in the scale of the figure. In view of the data, it shows that the visited vertices have a mean of ca. 200 for the Context Walk. Node Type performs second best by having a higher number of vertices for higher candidate counts.

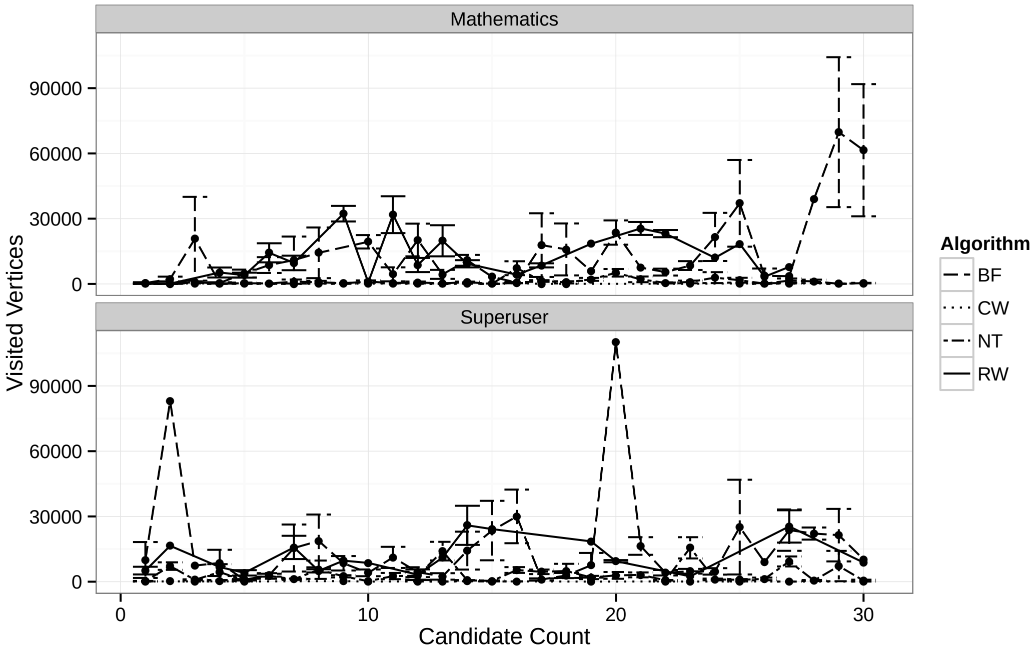


Figure 5.7: Number of Visited Vertices per Candidate Count. Breath First (dashed), Context Walk (dotted), Node Type (dash-dot) and Random Walk (solid)

In contrast to the first two strategies, Breath First and Random Walk seem to follow no clear behaviour handling the candidate count increases and need to visit more vertices, at which the Breath First strategy needs most.

The second parameter of the candidate selection phase is the threshold. Using the combined rank, the visited learners have to score under the threshold to be added to the candidate set. Fig. 5.8 shows the relation of the threshold and the visited vertices. To make it more readable the error bars are removed in the figure. The behaviours of the search strategies follow the relation between candidate count and visited vertices. Again Context Walk performs best by simply forming a line at the bottom of the figure. In addition, the Node Type strategy performs similar to CW but with some peaks for a low threshold. The unclear but high behaviour of BF and RW is also present.

Summarizing the different characteristics of the search strategies in the candidate selection, Node Type and Breath First find the candidates with low scores. Evaluating the performance of the search strategies shows that Context Walk visits the smallest amount of vertices to find the candidates set even if the candidate count and threshold are chosen small. Second best performs the Node Type based search strategy by having small peaks for high candidate counts

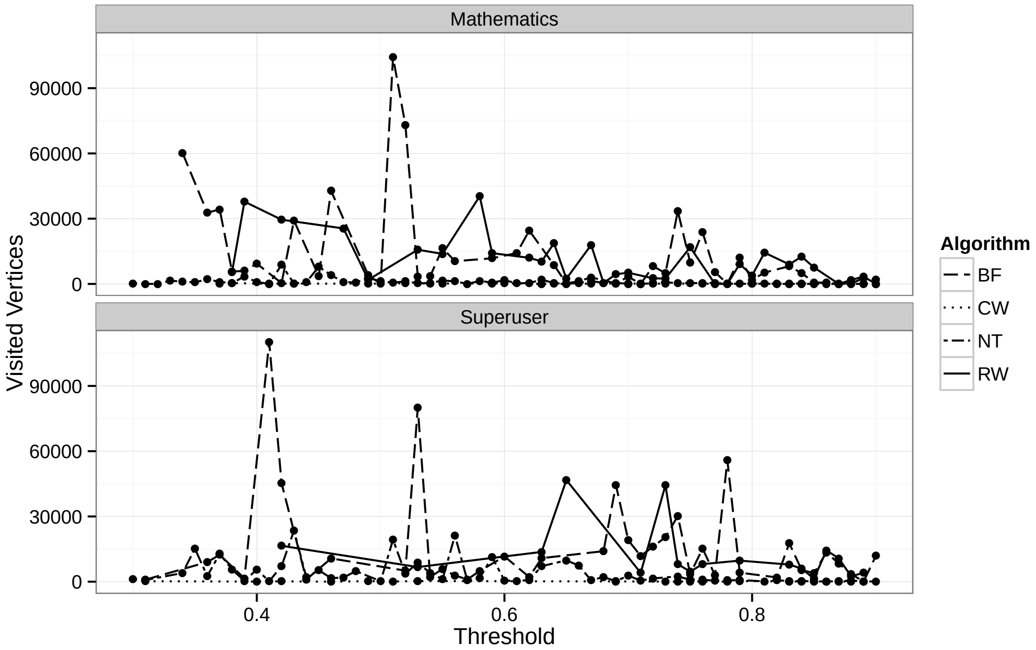


Figure 5.8: Visited Vertices per Threshold. Breath First (dashed), Context Walk (dotted), Node Type (dash-dot) and Random Walk (solid)

and small thresholds. By considering not just the scaling but additionally the complexity of selecting which vertex to visit next, Context Walk has to load the complete edge list of the current node to compute the relative probabilities of the edges to pick the next vertex to visit. In contrast, NT performs one query, when the candidate selection phase is started and fills a list of explicit learner vertices that has to be checked for being candidates. So NT may have the best performance by finding candidates in the graph and also finds the candidates with the lowest combined distance. Evaluating the candidate selection in total, it can be said that the candidate selection is able to find candidates that are suitable for a learning group in context of the presented metrics even for strong constraints for the number of candidates to be found and a small rank. To select the candidates from the eLearning graph, Node Type based selection should be used.

5.5.2 Genetic Algorithms

In order to optimize the groupings of candidates, Genetic Algorithms are used in the Group Optimization. While several parameters are used in the optimization, it is relevant which parameters configuration leads to low fitness scores. To present the characteristics of the

Group Optimization in detail, Figure 5.9 shows the relations between the fitness score and the parameters used in the second phase of the group formation process. The crossover rate defines how many chromosomes in the current generation are split and rejoined with others. It can be seen that a high cross over rate leads to lower fitness scores. Accordingly a high exchange of group parts takes an important role in the optimization. Looking at the influence of the generations in a Genetic Algorithm run, there seem to be no clear correlation between it and the fitness score of the end population. Besides the crossover operations, mutations change the chromosomes in each generation. Here, one member of a group constellation is exchanged with another one from the candidate set. The relation between fitness and mutation rate shows an increase of the fitness with a higher mutation rate. On the contrary to the crossover rate a small mutation rate leads to better fitness scores. The population size determines how many chromosomes are generated at the beginning of the Genetic Algorithm and how many survive from generation to generation. Like the number of generations, there is no clear correlation for the fitness scores and the population size. The size of the learning group is not a parameter for the Genetic Algorithm but yet very important in Group Optimization and Group Formation in general. The Figure shows, that the lowest fitness is scored by groups with 2 or 3 members (the initiating learner is not included in the group size parameter). This can be explained by the fact that in case of 2 or 3 group members the the possibility of sharing the same tags or neighbours is higher than for larger groups. So the group size parameter increases the fitness because of lower possibilities of shared neighbours. After the first high increase of the fitness score the fitness rises slowly but jitters by group sizes over 15 in both data sets.

The parameters show that a crossover rate over 0.5 in combination with a very low mutation rate should lead to small fitness scores. From a performance centred perspective, the optimal group size is over 3 and under 15. Because they seem to do not have an impact on the fitness score of a group constellation, the number of generations and population size could be kept small.

5.6 Group Quality

The previous section checked for the fulfilment of the requirements. The objective of this section is to investigate the quality of the group constellations by addressing their fitness, stability and similarity to empirical groups in the Stack Exchange data sets.

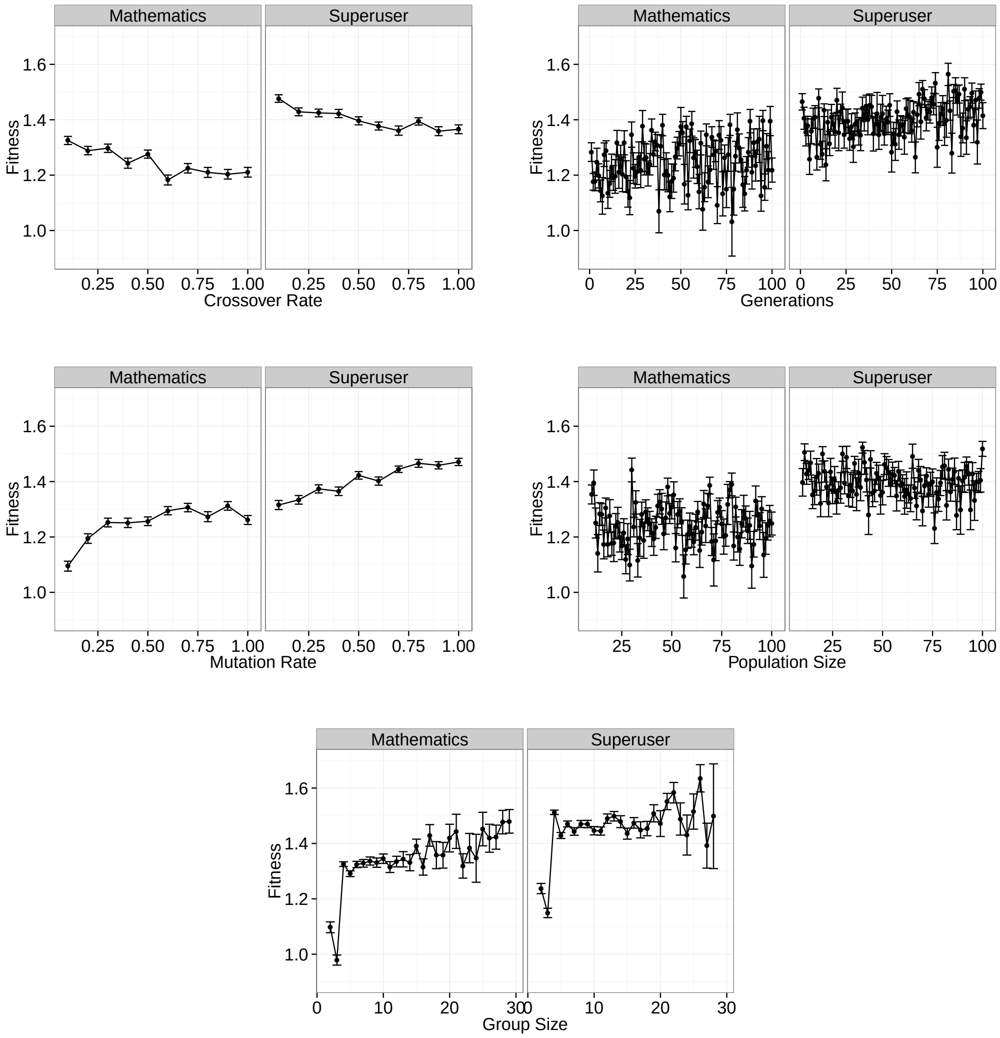


Figure 5.9: Parameters of the Group Optimization

5.6.1 Fitness

In the group formation approach the quality of a group constellation is rated by its fitness value. Used in the Genetic Algorithms, the value includes the distance in learning style and social distance for each pair constellation of group members, the Knowledge Rank for each member and the topic to work on. While the fitness value is able to describe the group quality in the context of the algorithm, an evaluation of the overall group quality goes beyond considering the isolated fitness value. Observing the overall fitness values enables a discussion of not only the isolated metrics scores but an integrated perspective on the group quality. Fig. 5.10. shows

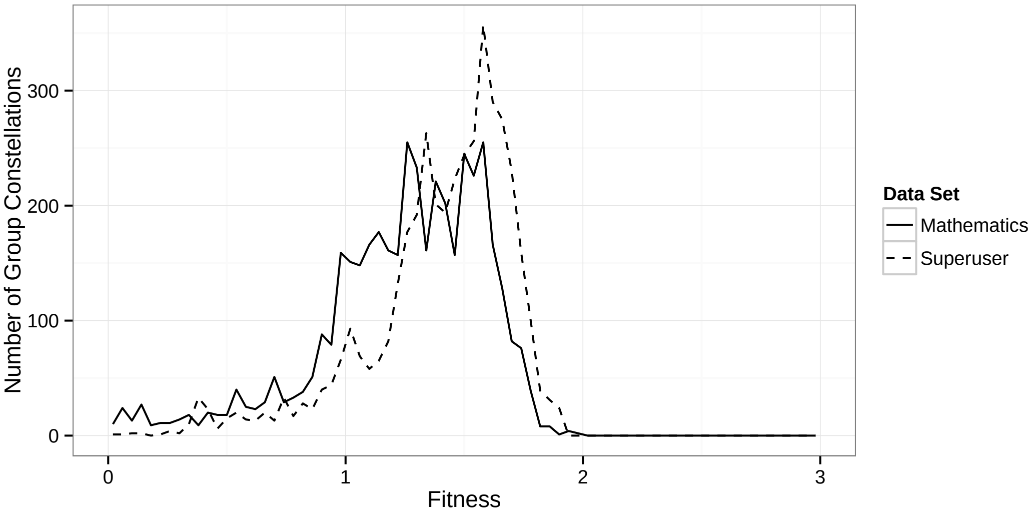


Figure 5.10: Frequency of Fitness Values scored for the Group Constellations in the Group Optimization Phase

the frequency of the fitness values achieved by the group constellations. While the possible scale of the fitness function reaches from 0 to 3, the highest value scored lies about 2 in both data sets. Marking the lower middle of the scale, most constellations in the Mathematics data score between 0.9 and 1.6, nonetheless there are also scores under 1 and even approaching the perfect fitness score of 0. The results of the Superuser data show a more pronounced and clustered peak of the fitness scores between 1.2 and 1.8. The lowest scores observed in the evaluation is 0.0264 for Mathematics and 0.0396 for Superuser. With a peak value for the social distance around 0.8 the average of the fitness values indicate that the found group constellations are optimized in all dimensions according to the requirements. Still, the high variance in the fitness scores may cause a different learning experience for learners in a group with a fitness score of 0.1 and 2.0.

5.6.2 Stability

A found group constellation is a small set of learner vertices collected from the whole eLearning graph. The selection process should find vertices that have the best fit for the initiating learner and the selected topic. This means that the group constellation is not a random selection from learner vertices scoring under the given threshold but rather stands out from the remaining network as a pronounced set of vertices. To evaluate this aspect of group quality the stability of the Candidate Selection and Group Optimization is relevant. To measure the stability of

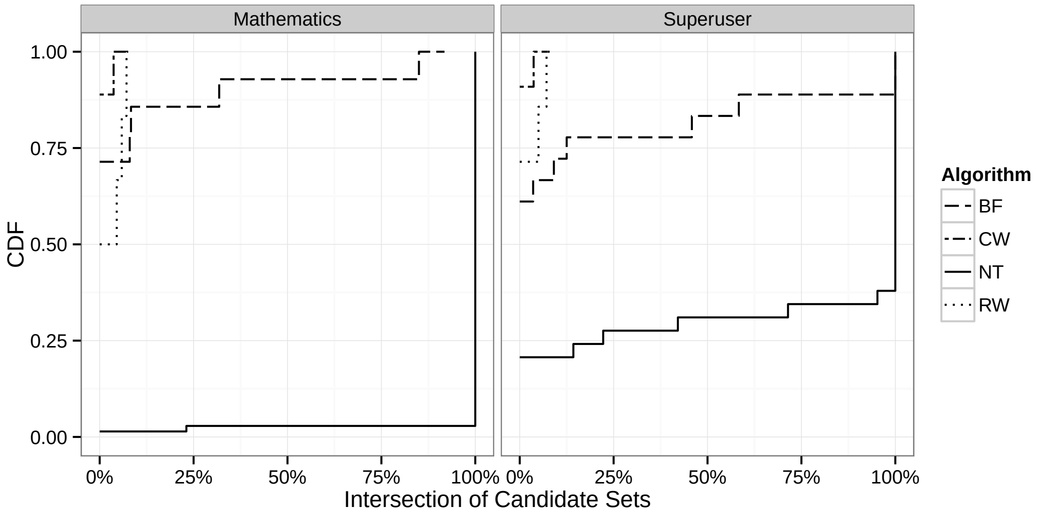


Figure 5.11: CDF of the Intersections of Candidate Sets with same Configuration

the algorithm, the intersections of the candidate sets and group constellations created by each phase are used in this evaluation.

Candidate Selection

The stability of the Candidate Selection highly depends on the strategy used for the graph traversal. Breath First Search and Node-type-based selection have a strict approach to select the next vertex to visit, but Random Walk and the Context Walk depend on a random selection which may have a negative impact on the stability. In this evaluation the stability of the CS is formalized as the percentage intersection of the candidate sets found by runs with the same parametrization. Figure 5.11 shows the CDF of the Intersection of candidate sets per run grouped by the search strategies. As assumed above the selection strategy has an impact on the intersections. However, group constellations found in the Superuser data set are generally less intersected. Context Walk has an intersection of only 10% maximum, the same limit can be found for Random Walk. Even if Breath First and Node Type have a strict selection strategy the intersection for BF is generally smaller than for NT and does not reach 100% in the Mathematics data set. Node Type shows the biggest intersection with mostly 100%. This result strengthens the suitability of NT for the problem of finding compatible learners in the eLearning graph.

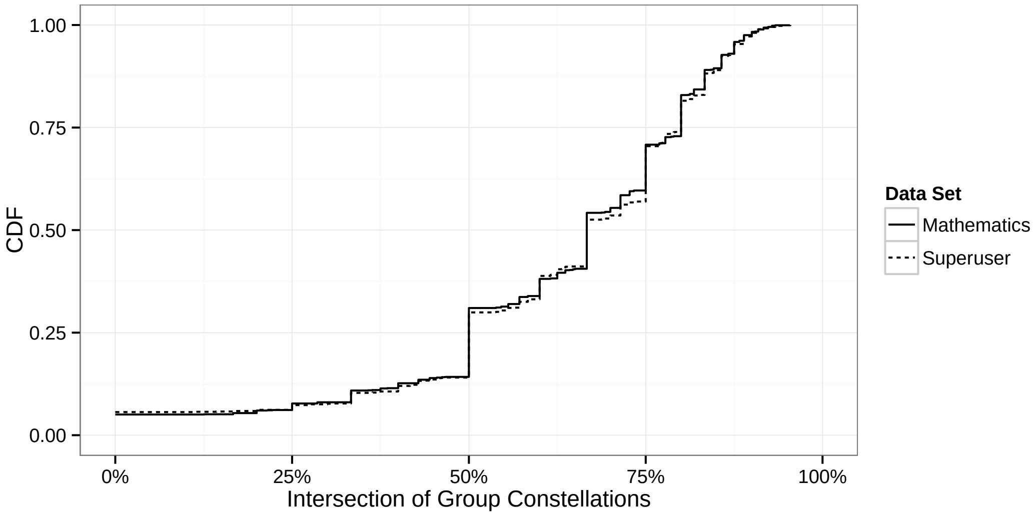


Figure 5.12: CDF of the Intersections of Group Constellations in the Group Optimization

Group Optimization

The stability in the Group Optimization is measured as the intersections of the group constellations. The base of the Group Optimization is built by the runs of the Candidate Selection together with the parameters for the Genetic Algorithm and the group size. Figure 5.12 shows the CDF of the intersections of common group constellations. Compared to the Candidate Selection, the intersections in the GO are higher: 50% of the common optimizations have intersections in their members over 70%. This is implied by the, in comparison to the whole graph structure, actually small set of candidates. Another difference to the stability of the CS is that the constellations in both data sets show the same distributions.

5.6.3 Similarity to Empirical Groups

Using the data from the Stack Exchange Network for the evaluation allows to compare the groups constellation found by the group formation approach to the real groups in the data sets. Dimensions in this part of the evaluation are the group size of the empirical groups as well as the Knowledge Rank of the empirical groups compared to the group constellations and the social distance present in both types of groups. The scores of the fitness function are not considered at this point, because the fitness includes the distance in learning style, which is not included in the evaluation data but assigned randomly.

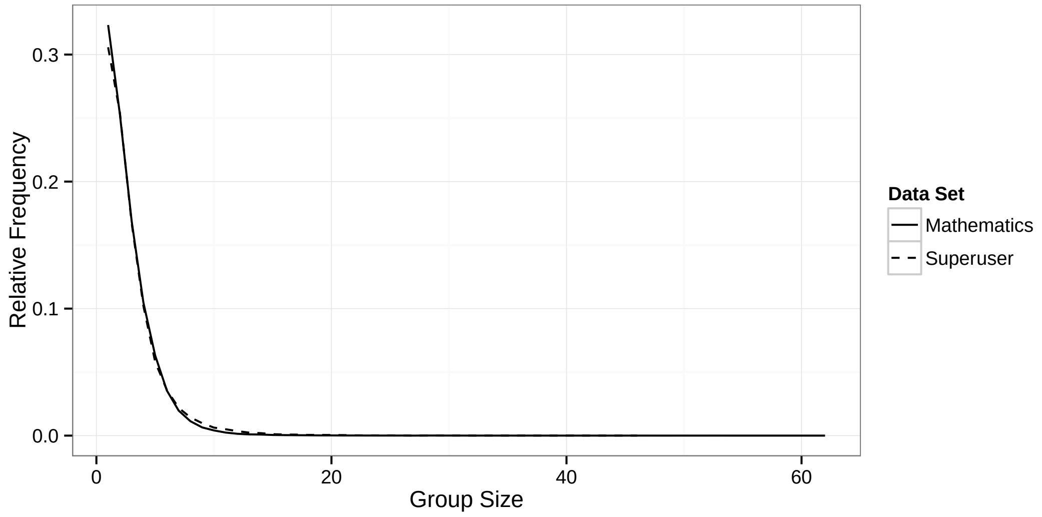


Figure 5.13: Group Size observed in the Mathematics and Superuser Data Sets

Size of Stack Exchange Groups

The size of a learning groups is chosen by the initiating learner in the group formation approach, but the evaluation of the Group Optimization has shown that the best fitness value can be scored for groups with 3 to 15 members. To estimate how the group constellations relate to the real groups in the Stack Exchange platform, the size of the empirical groups is relevant. Figure 5.13 shows the distribution of the group size in the Mathematics and Superuser data. Most groups in the data sets have a size of 2 members, meaning that one users asks the question and another gives a satisfying answer. From this peak, the size fall with a long tail to groups with over 60 members. The figure also shows the unanswered questions of the Stack Exchange platforms where the only group member is the initiating user. While the distribution is very wide, the median of 3 and the mean of 3.654 show that the center of the distribution lies at the bottom end of the range evaluated in the Group Optimization. The sizes of the empirical groups reflect the purpose of question answering on the Stack Exchange platform, but also show that small groups with 2 to 10 members are well suited in the context of collaborative knowledge sharing and learning.

Knowledge Rank

The Knowledge Rank measures the correlation of the learner and the topic by the tags both share. While the scores in the Candidate Selection are discussed above, they are now compared

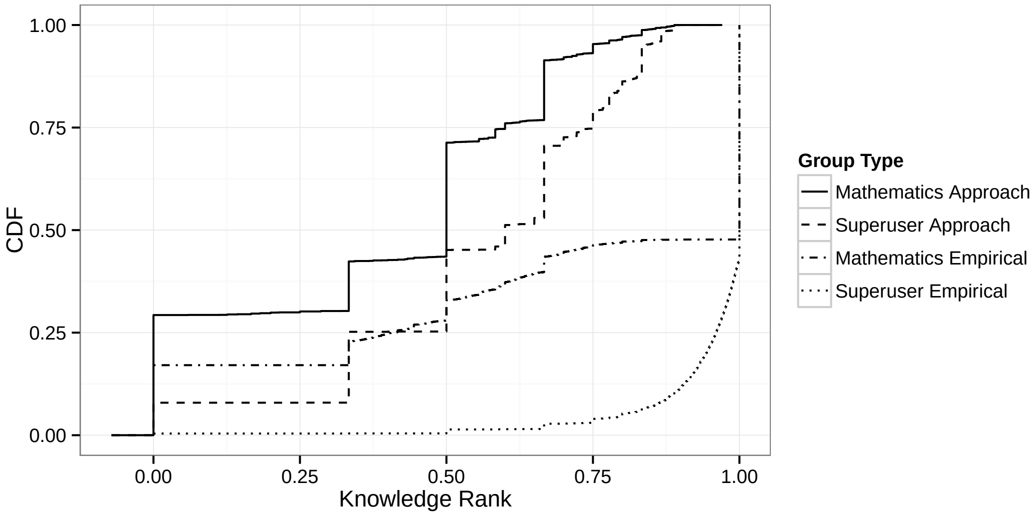


Figure 5.14: CDF of the Knowledge Rank of the Members in the Group Constellations and Empirical Groups

to the scores observed for the members of the real groups. The calculation of the Knowledge Rank for members of the groups shows that the scores for the group constellations are higher than for the empirical groups (Figure 5.14). 50% of scores for the empirical group members score with 1, indicating that no tags are shared with the topic and the member. The group members in the Superuser data barely score lower than 0.8 but also show the peaks for 0.5 and 0.6. In the Mathematics dataset, scores are lower and cross the distribution of the group constellations of the Superuser data set. The difference of the Knowledge Rank for the Mathematics and Superuser group members is caused by the different relation between tags, learner and topics in both data sets. Superuser has 5 times more tags than the Mathematics data by having just half of the learner and topic vertices. The comparison to the empirical groups and the group constellations show a higher connection through the tags in the group constellations. One reason for this result could be that the tags mostly used to categorize topics (or questions) and tagging relations are not uniform distributed over the group members. There could also be a gap between the creators of questions in the Stack Exchange Network who assign the tags and the users who give answers to the question and do not assign tags.

Social Distance

The cohesion of the group members is ranked by the social distance metric. Figure 5.15 shows the CDF of the social distance scored by the group constellations and the empirical

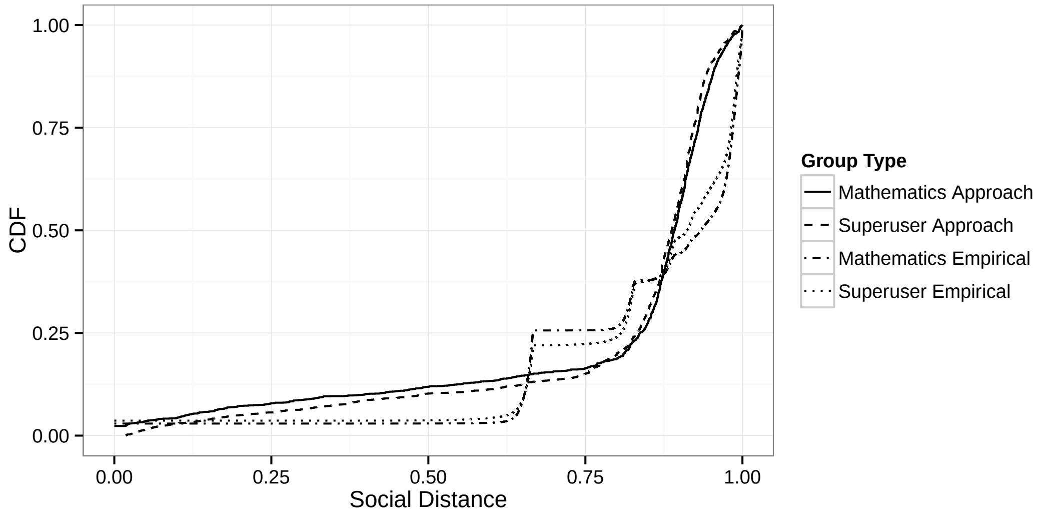


Figure 5.15: CDF of the Knowledge Rank of Members in the Group Constellations and Empirical Groups

groups for both data sets. While the group constellations are able to achieve lower scores than the empirical groups, the peaks of the empirical scores are lower than the peak of the constellations. The empirical groups form a higher amount of better connected groups but the group constellations are generally higher connected.

5.6.4 Summary

The first question of this evaluation was, whether the implemented group formation approach operates compliant with the requirements on group formation in OSN. The scores for the metrics quantifying the requirements for the members of the found group constellations show that the evaluated requirements are full filled. The distance in learning style shows lowest scores, indicating balanced preferences for the group constellations. A broader distribution could be observed for the Knowledge Rank, but it is also able to find candidates with common tags. The social distance peaks for a high distance but ensures a direct connectivity in the graph.

Finding an effective algorithmic parametrization according to performance and group quality was also a subject of the evaluation. The Candidate Selection is able to find candidates that are suitable for a learning group in context of the presented metrics, even in view of strong constraints for the number of candidates to be found and a small rank. The node type based selection strategy shows the best performance. The analysis of the parameters used in the

Group Optimization shows that a crossover rate over 0.5 combined with a very low mutation rate should lead to small fitness scores. From a performance centred perspective, the optimal group size is over 3 and under 15.

The quality of the formed groups in sense of fitness, stability and similarity to real groups was the third issue of the evaluation. The fitness score of the group constellations shows a mean of 1.6, indicating that the group constellations are optimized in all dimensions according to the requirements. To investigate the stability of the group constellations, the intersections of the candidate sets and group constellation was analysed. In this regard the node type based selection reveals the biggest intersection with mostly 100%. This results strengthen the suitable of NT for the given problem. At Group Optimization, 50% have intersections in their members over 70%. This is implied by the smaller set of learner determined by the candidate count of the run. In comparison to empirical groups, the group constellations show a higher connection through the tags. The empirical groups form a higher amount of better connected groups but the group constellations are generally higher connected.

The group formation approach evaluated in this chapter is able to find group constellations in the eLearning-enabled OSN that provide a balanced learning style among the members as well as a common context with the topic to work on. The group constellations are well connected in their inside. Compared to empirical groups in the Stack Exchange data set, the provided constellations show a higher contextual relation and a similar connectivity.

6 Conclusion and Outlook

This thesis answers the question of how to stimulate a team building process that is effective for learners in an instructor-less Online Social Network designed for collaborative learning. Based on previous publications [4, 5, 6] a group formation approach was implemented and integrated in the eLearning-enabled OSN.

The concept of an eLearning-enabled Online Social Network was developed by analysing Personal Learning Networks using a categorization according to data availability and stress the importance of joining the contacts of learners on different platforms. The presented approach of finding learning groups in the eLearning-enabled OSN is divided in two parts. First, the social network is searched and the approach tries to find a minimal number of suitable candidates for the formation of a group, shaped by an initiator on a chosen topic. Based on the candidates, in the second part Genetic Algorithms are deployed to optimize a constellation of collaborators for a successful group learning experience. Both steps are grounded on metrics that are calculated from user configuration and statistics in the underlying online social network. The evaluation of the group formation approach used two large empirical data set extracted from the Mathematics and Superuser site on the Stack Exchange platform. It showed that the presented group formation approach is able to find group constellations that have a common knowledge context, learning style and are well connected in the eLearning graph. Compared to empirical groups, the best constellations show a similar size, but have a higher contextual knowledge similarity. The future work on the eLearning-enabled OSN should include the deployment of the platform and testing all features with real learners. While the code of the group learning, content network and group formation component were handled project intern, the publication of the code could increase the popularity and opens the platform to learners on the Internet. This deployment should go along with work on the unanswered question of Roreger and Schmidt [1] how to facilitate a consistent learning progress, include feedback and corrective actions.

Bibliography

[1] H. Roreger and T. C. Schmidt, "Socialize Online Learning: Why we should Integrate Learning Content Management with Online Social Networks," in *Proc. of IEEE Intern. Conf. on Pervasive Computing and Communication (PerCom), Workshop PerEL.* Piscataway, NJ, USA: IEEE Press, March 2012, pp. 685–690.

[2] C. Dorn, F. Skopik, D. Schall, and S. Dustdar, "Interaction Mining and Skill-dependent Recommendations for Multi-objective Team Composition," *Data & Knowledge Engineering*, vol. 70, pp. 866–891, 2011.

[3] J. Vassileva, "Toward Social Learning Environments," *Learning Technologies, IEEE Transactions on*, vol. 1, no. 4, pp. 199 –214, oct.-dec. 2008.

[4] S. Brauer and T. C. Schmidt, "Group Formation in eLearning-enabled Online Social Networks," in *Proc. of the International Conference Interactive Computer aided Learning (ICL'12)*, M. E. Auer, Ed. Piscataway, NJ, USA: IEEE Press, Sep. 2012.

[5] S. Brauer, T. C. Schmidt, and A. Winschu, "Personal Learning Networks with Open Learning Groups - a Formal Approach," in *Proc. of the International Conference Interactive Computer aided Learning (ICL'13)*, M. E. Auer, Ed. Piscataway, NJ, USA: IEEE Press, Sep. 2013.

[6] S. Brauer and T. C. Schmidt, "Are Circles Communities? A Comparative Analysis of Selective Sharing in Google+," in *Proc. of 34th Int. Conf. Dist. Comp. Systems ICDCS – WS HotPost.* Piscataway, NJ, USA: IEEE Press, June 2014, pp. 8–15.

[7] D. M. Boyd and N. B. Ellison, "Social Network Sites: Definition, History, and Scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, October 2008.

[8] K. Musiał and P. Kazienko, "Social networks on the internet," *World Wide Web*, vol. 16, no. 1, pp. 31–72, 2013.

[9] S. Milgram, "The small world problem," *Psychology today*, vol. 2, no. 1, pp. 60–67, 1967.

[10] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca, "Network Analysis in the Social Sciences," *Science*, vol. 323, no. 5916, pp. 892–895, 2009.

[11] G. Magno, G. Comarela, D. Saez-Trumper, M. Cha, and V. Almeida, "New kid on the block: exploring the google+ social graph," in *Proceedings of the 2012 ACM conference on Internet measurement conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 159–170.

[12] R. Gonzalez, R. Cuevas, R. Motamedi, R. Rejaie, and A. Cuevas, "Google+ or Google-?: Dissecting the Evolution of the New OSN in Its First Year," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 483–494.

[13] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," in *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*, San Diego, CA, October 2007.

[14] I. Foudalis, K. Jain, C. Papadimitriou, and M. Sideri, "Modeling Social Networks through User Background and Behavior," in *Proc. of the 8th int. conf. on Algorithms and models for the web graph (WAW'11)*, 2011, pp. 85–102.

[15] M. Granovetter, "The Strength of Weak Ties," *American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.

[16] R. E. Kraut, P. Resnick, S. Kiesler, M. Burke, Y. Chen, N. Kittur, J. Konstan, Y. Ren, and J. Riedl, *Building successful online communities: Evidence-based social design.* MIT Press, 2012.

[17] J. Preece, *Online Communities: Designing Usability and Supporting Socialbilty*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 2000.

[18] M. J. Brzozowski, P. Adams, and E. H. Chi, "Google+ Communities as Plazas and Topic Boards," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems.* ACM, 2015, pp. 3779–3788.

[19] J. Lazar and J. Preece, "Classification schema for online communities," *AMCIS 1998 Proceedings*, p. 30, 1998.

[20] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[21] J. Yang and J. Leskovec, "Defining and Evaluating Network Communities based on Ground-truth," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, ser. MDS '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:8.

[22] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[23] F. Lorrain and H. C. White, "Structural equivalence of individuals in social networks," *The Journal of mathematical sociology*, vol. 1, no. 1, pp. 49–80, 1971.

[24] S. Kairam, M. Brzozowski, D. Huffaker, and E. Chi, "Talking in circles: Selective sharing in google+," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. ACM, 2012, pp. 1065–1074.

[25] S. D. Farnham and E. F. Churchill, "Faceted identity, faceted lives: social and technical issues with being yourself online," in *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, ser. CSCW '11. New York, NY, USA: ACM, 2011, pp. 359–368.

[26] S. L. Feld, "The focused organization of social ties," *American journal of sociology*, pp. 1015–1035, 1981.

[27] J. Watson, A. Besmer, and H. R. Lipford, "+Your Circles: Sharing Behavior on Google+," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*, ser. SOUPS '12. New York, NY, USA: ACM, 2012, pp. 12:1–12:9.

[28] J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 548–556.

[29] P. Baumgartner, H. Häfele, and K. Maier-Häfele, *Content Management Systeme in e–Education*. Innsbruck: StudienVerlag, 2004.

[30] M. Engelhardt, A. Hildebrand, D. Lange, and T. C. Schmidt, "Semantic Overlays in Educational Content Networks – The hylOs Approach," *Campus-Wide Information Systems*, vol. 23, no. 4, pp. 254–267, September 2006. [Online]. Available: http://www.emeraldinsight.com/10.1108/10650740610704126

[31] D. A. Wiley, "Learning object design and sequencing theory," Ph.D. dissertation, Brigham Young University, Provo, UT, 2000.

[32] M. Engelhardt and T. C. Schmidt, "Semantic Linking – a Context-Based Approach to Interactivity in Hypermedia," in *Berliner XML Tage 2003*, R. Tolksdorf and R. Eckstein, Eds., Humboldt Universität zu Berlin, September 2003, pp. 55–66.

[33] M. Brut, D. Kukhun, and F. Sedes, "Ensuring Semantic Annotation and Retrieval within Pervasive E-Learning Systems," in *Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on.* Piscataway, NJ, USA: IEEE Press, March 2008, pp. 959–964.

[34] B. Feustel and T. C. Schmidt, "Media Objects in Time – a multimedia streaming system – work in progress paper v 1.5," *Computer Networks*, vol. 37, no. 6, pp. 729 – 737, 2001. [Online]. Available: http://dx.doi.org/10.1016/S1389-1286(01)00246-8

[35] "Learning object meta-data," http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html, IEEE, Draft Standard 1484.12.1, July 2002.

[36] M. Engelhardt, A. Hildebrand, D. Lange, and T. C. Schmidt, "Reasoning about eLearning Multimedia Objects," in *Proc. of WWW 2006, Intern. Workshop on Semantic Web Annotations for Multimedia (SWAMM)*, J. V. Ossenbruggen, G. Stamou, R. Troncy, and V. Tzouvaras, Eds., May 2006. [Online]. Available: http://image.ntua.gr/swamm2006/resources/paper06.pdf

[37] G. P. Landow, "The rhetoric of hypermedia: Some rules for authors," *Journ. of Comp. in Higher Education*, vol. 1, no. 1, pp. 39–64, 1989.

[38] N. Dabbagh and A. Kitsantas, "Personal Learning Environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning," *The Internet and Higher Education*, vol. 15, no. 1, pp. 3–8, 2012.

[39] G. Siemens, "Connectivism: A learning theory for the digital age," in *International Journal of Instructional Technology and Distance Learning*, 2005.

[40] T. Martindale and M. Dowdy, *Emerging Technologies in Distance Education.* Edmonton: AU Press, Athabasca University, 2010, ch. Personal Learning Environments.

[41] M. Van Harmelen, "Personal Learning Environments." in *ICALT*, vol. 6, 2006, pp. 815–816.

[42] A. Couros, *Developing Personal Learning Networks for Open and Social Learning.* Athabasca University Press, 2010, no. 6, pp. 109–128.

[43] D. Warlick, "Grow Your Personal Learning Network," *Learning & Leading with Technology*, vol. March/April, pp. 12–16, Mar. 2009.

[44] F. Coffield, D. Moseley, E. Hall, and K. Ecclestone, "Should we be using learning styles? What research has to say to practice," The Learning and Skills Research Center, Tech. Rep., 2004.

[45] M. Budhu, "Interactive web-based learning using interactive multimedia simulations," 2002.

[46] C.-I. Peña, J.-L. Marzo, and J.-L. Rosa, "Intelligent Agents in a Teaching and Learning Environment on the Web," in *Proc. of International Conference on Advanced Learning Technologies (ICALT2002)*, 2002, pp. 21–27.

[47] N. Stash and P. D. Bra, "Incorporating Cognitive Styles in AHA! (The Adaptive Hypermedia Architecture)," in *Proc. of the IASTED International Conference Web-Based Education*, 2 2004, pp. 378–383.

[48] R. Felder and L. Silverman, "Learning and teaching styles in engineering education," *Engineering education*, vol. 78, no. 7, pp. 674–681, 1988.

[49] J. Villaverde, D. Godoy, and A. Amandi, "Learning styles' recognition in e-learning environments with feed-forward neural networks," *Journal of Computer Assisted Learning*, vol. 22, no. 3, pp. 197–206, Jun. 2006.

[50] M. Clements, A. P. de Vries, and M. J. Reinders, "Optimizing single term queries using a personalized Markov random walk over the social graph," in *Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR)*, 2008.

[51] T. Gruber, "Collective knowledge systems: Where the social web meets the semantic web," *Web Semant.*, vol. 6, no. 1, pp. 4–13, Feb. 2008.

[52] A. Bielenberg, L. Helm, A. Gentilucci, D. Stefanescu, and H. Zhang, "The growth of diaspora-a decentralized online social network in the wild," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on.* IEEE, 2012, pp. 13–18.

[53] C. Reffay, C. Teplovs, F. Blondel *et al.*, "Productive re-use of CSCL data and analytic tools to provide a new perspective on group cohesion," in *Proc. of 9th Int. Conf. on Computer-Supported Collaborative Learning (CSCL2011)*, Jul. 2011, pp. 846–850.

[54] N. With, “eLearning in sozialen Netzwerken: Eine Erweiterung von Diaspora um semantische Content-Netze,” Master’s thesis, HAW Hamburg, Germany, 2015.

[55] E. Amitay, D. Carmel, N. Har’El, S. Ofek-Koifman, A. Soffer, S. Yogev, and N. Golbandi, “Social search and discovery using a unified approach,” in *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, ser. HT ’09. New York, NY, USA: ACM, 2009, pp. 199–208.

[56] T. Bogers, “Movie Recommendation using Random Walks over the Contextual Graph,” in *Second Workshop on Context-Aware Recommender Systems*, 2010.

[57] W. M. Cruz and S. Isotani, “Group formation algorithms in collaborative learning contexts: A systematic mapping of the literature,” in *Collaboration and Technology.* Springer, 2014, pp. 199–214.

[58] A. Ounnas, H. C. Davis, and D. E. Millard, “A Framework for Semantic Group Formation in Education,” *Educational Technology & Society*, vol. 12, no. 4, pp. 43–55, 2009.

[59] A. Ounnas, H. Davis, and D. Millard, “Towards Semantic Group Formation,” in *Advanced Learning Technologies, ICALT 2007. Seventh IEEE International Conference on*, july 2007, pp. 825 –827.

[60] J. Moreno, D. A. Ovalle, and R. M. Vicari, “A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics,” *Computers & Education*, vol. 58, no. 1, pp. 560–569, 2012.

[61] C. F.-Z. Khaled Halimi, Hassina Seridi, “Solearn: A Social Learning Network,” in *International Conference on Computational Aspects of Social Networks (CASoN).* Piscataway, NJ, USA: IEEE Press, 2011, pp. 130–135.

[62] T. Arndt and A. Guercio, “Social Network-Based Course Material Transformations For A Personalized And Shared Ubiquitous E-Learning Experience,” in *The 5th int. conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2011)*, 2011, pp. 218–222.

[63] D. E. Goldberg and J. H. Holland, “Genetic Algorithms and Machine Learning,” *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.

[64] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[65] J. Zhang and M. S. Ackerman, "Searching for expertise in social networks: a simulation of potential strategies," in *Pro. of the 2005 int. ACM SIGGROUP conf. on Supporting group work*, ser. GROUP '05, 2005, pp. 71–80.

[66] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992.

[67] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, vol. 2009, no. 421425, 2009.

[68] M. Clements, A. P. de Vries, and M. J. Reinders, "Optimizing single term queries using a personalized markov random walk over the social graph," in *Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR)*, 2008.

[69] L. A. Adamic and E. Adar, "How To Search a Social Network," *Social Networks*, vol. 27, 2005.

[70] R. Fielding, "Representational state transfer," *Architectural Styles and the Design of Netowork-based Software Architecture*, pp. 76–85, 2000.

[71] J. Leskovec, J. Kleinberg, and C. Faloutsos., "Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005, pp. 177–187.

[72] M. Derntl and S. Graf, "Impact of Learning Styles on Student Blogging Behavior," in *Ninth IEEE International Conference on Advanced Learning Technologies (ICALT).* Piscataway, NJ, USA: IEEE Press, 2009, pp. 369–373.

[73] R. M. Felder and J. Spurlin, "Applications, Reliability and Validity of the Index of Learning Styles," *Int. J. Engng Ed.*, vol. 21, no. 1, pp. 103–112, 2005.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 4. Dezember 2015 Steffen Brauer