published as: Thomas Staubitz, Sebastian Woinar, Jan Renz, Christoph Meinel: TOWARDS SOCIAL GAMIFICATION - IMPLEMENTING A SOCIAL GRAPH IN AN XMOOC PLATFORM, In Proceedings of 7th International Conference of Education, Research and Innovation 17-19 November, 2014, Seville

TOWARDS SOCIAL GAMIFICATION - IMPLEMENTING A SOCIAL GRAPH IN AN XMOOC PLATFORM

Thomas Staubitz[1], Sebastian Woinar[2], Jan Renz[3], Christoph Meinel[4]

[1,3,4]Hasso Plattner Institute Potsdam (Germany)
[2]SAP Innovation Center SAP SE (Germany)

Abstract

High dropout rates and poor user participation are some of the major challenges for xMOOC providers. Similar to others, openHPI also faces these problems.

Gamification is a mechanism that recently has been successfully employed to raise user motivation in web applications. A concept to gamify the openHPI platform has been proposed and implemented. Parts of it are already published on the platform. It is proposed that further gamification elements could be improved by applying the factor *relatedness*. Introducing a social graph to the platform, allowing users to establish connections to other users, is seen as a means to increase this relatedness. For example, the gamification element of a leaderboard gains a lot of relevance when a user finds herself amongst a list of her friends than a list of random strangers.

The paper at hand introduces the implementation of a social graph in the context of openHPI's MOOC platform.

1 INTRODUCTION

Typical Massive Open Online Courses (xMOOCs[1]) apply a structure similar to university courses with fixed time ranges, hard deadlines and frequent homework assignments for a fixed group of enrolled students. A MOOC usually has a length of 6-8 weeks and offers about two hours of video material per week. In general, MOOCs are free of charge and everybody is welcome to enroll. The partial synchronicity of MOOCs and the periodical tasks are some of their success factors.

In 2012, the Hasso Plattner Institute (HPI) launched its own MOOC platform: openHPI[2]. By now, twelve courses have been delivered to the public, a sister platform—openSAP[3]—has been established and both platforms have satellite sites in China[4]. During 2013 and 2014 the platform has been rewritten from ground up. From March to September 2014 all instances of the platform have been moved to the new code base. The paper at hand is situated in this context. We will use the term openHPI v1 to address the old openHPI platform and openHPI v2 to address the new version of the platform.

A challenge that openHPI shares with other major MOOC platforms is that a large share of participants is not very active during a course. Required tasks are completed but especially active participation in the forums leaves room for improvement [1]. In average, users at openHPI are 83% passive, 14% rarely active and only 3% very active [3]. A survey amongst openHPI participants also confirms this [2]. As one approach to improve this situation Willems et al. presented a first gamification concept for openHPI [1]. The concept has not yet been published on the productive system in total, but certain elements, such as the pinboard (an optimized discussion forum) and an improved approach to display the users' progress are already in use.

---

[1] For a distinction between xMOOCs and cMOOCs see e.g. http://massiveopenonlinecourses.com/xmooc-vs-cmooc/. From now, for reasons of brevity the term MOOC will be used as a synonym of xMOOC.

[2] https://open.hpi.de

[3] https://open.sap.com

[4] https://openhpi.cn, https://open.sap.cn

The proposed concept does not yet exploit the full potential of gamification, however. The factor *relatedness* had been left out completely in the initial approach; social networking features were not included in the draft [1]. The paper at hand discusses a proposal to fill this gap: The introduction of a social graph into openHPI, with the goal of expanding and maintaining the users' motivation and increasing their level of participation.

Learners on openHPI usually attend more than one course. Often, social relations between users already exist [2]. Implementing a social graph to manifest existing relations, therefore, appears to be promising. Section 2 will introduce some of the theories that led us to start working in this direction. Section 3 will discuss our selection of the database beneath openHPI's implementation of the social graph and dive deeper into the implementation of a social networking service (SNS) and its integration into openHPI. Sections 4 and 5 will conclude our findings so far and give an overview on our future plans.

2 PSYCHOLOGICAL BACKGROUND

2.1 Motivation

A crucial key to understand why participants are willing to be active in a course and participate on the platform is the question about their motivation. What keeps them going on to finish a course? It is differentiated between two basic types of motivation:

Intrinsic motivation: Has its origin within a person. It refers to a behavior, which is shown in absence of external incentives. This is central to inherent human tendencies to learn and develop.

Extrinsic motivation: Refers to a behavior performed to obtain some outcome separable from the activity itself. A third party externally causes it.

2.1.1 Self Determination Theory

The Self Determination Theory (SDT) by Deci and Ryan is a macro-theory of human motivation, emotion and development [6]. According to this theory people are innately curious and interested creatures who possess a natural love of learning and who desire to internalize knowledge, customs and values. Educators, through introducing external controls into learning climates, often undermine this native encouragement. Through rewards and punishment students change their cognition of tasks, which were intrinsically fun before. For example supervision, monitoring and performance evaluation are common means in schools, which can re- duce the original motivation of students [7].

Deci and Ryan identify three intrinsic motivators:

> **Competence**—Humans want to be efficient and good in what they are doing.
>
> **Autonomy**—They want to be in command of their life.
>
> **Relatedness**—They have a universal desire to interact and be connected with other.

2.1.2 Drive Theory

Daniel Pink has developed the drive theory. He also differentiates between intrinsic and extrinsic motivation. Rewards can be a great tool for simple, straightforward tasks, but when they require conceptual, creative thinking, rewards are disadvantageous. Especially monetary bonuses can be negative and prevent creativity [8].

The drive theory also identifies three intrinsic motivators:

> **Autonomy**—Humans want to make their own choices.
>
> **Mastery**—They like to improve.
>
> **Purpose**—They like to make meaningful contributions

2.1.3 Relatedness Autonomy Mastery and Purpose

The Relatedness Autonomy Mastery Purpose (RAMP) framework by Andrej Marczewski merges insights of SDT and Drive Theory [9].

Relatedness—Derives from the SDT, where it stands for relatedness and purpose. Humans have a universal desire to interact and be connected with other.

Autonomy—Is represented in both theories with the same term: humans want to have a choice in what they are doing. Their motivation is suppressed when they feel controlled.

Mastery—Matches SDT's concept of competence and Drive theory's concept of Mastery.

Purpose—Is mentioned in both frameworks - in SDT it is part of relatedness. People want to feel that they make a difference. They care about the outcome of their doings.

2.1.4 Extrinsic Incentives

Rewards are a proven way to spur students to put forth effort. But often this is behavior control and does not increase the motivation of learning. Some educators have refused extrinsic motivational methods from the beginning. It can be seen as bribing of the students for doing something they should do anyway because it is the right thing to do**—**meaning it is in their own or in society's interest. Rewarding students for learning undermines their intrinsic interest. If students are rewarded for doing what they already were doing for their own reasons, the intrinsic motivation is decreased to continue that behavior in the future. It can happen that they develop a minimax mentality: they do what will bring them the most rewards with the least effort. The effects of rewards depend on what rewards are used and how they are presented [12]. When students become aware of being bribed, they start to consider the bribing necessary for the activity, as they are not expected to overcome it without rewards. The students adopt the view that the activity itself is not worth performing in absence of extrinsic rewards. That way, the initially present intrinsic motivation is undermined [13]. Expected tangible rewards undermine intrinsic motivation for engaging in an interesting activity regardless of whether these rewards are contingent or completion-dependent [4]. Rewards have strong negative effects on subsequent intrinsic motivation to engage in interesting tasks. And for uninteresting tasks they have no significant effects at all. Verbal rewards enhance intrinsic motivation when they are primarily informational. When verbal rewards are controlling, intrinsic motivation is decreased [4]. Rewards are more effective for increasing the duration and intensity of effort than for improving the quality [13]. As a result, rewards can be better used for routine tasks, where steady performance or quantity is important, than creativity or craftsmanship. Students should not be granted with rewards as primary incentives for things the students should continue to do on their own - for example watching educational television or read quality books [13].

A very powerful but problematic extrinsic incentive is the competition between students. Competitions can be for tangible prices or just for the satisfaction of winning (being better than the others), between individuals or between groups. They are usually structured around test scores or other performance measures, to be able to announce a winner. Brophy states a number of arguments against the application in regular classrooms:

- The salience of competition can lead to a focus on the competition itself rather than the task.
- Students often do not have the choice to participate or not in the competition. They lack autonomy.
- The application of competition is more appropriate for routine practice tasks than for creative tasks.
- Competition can be only effective if everyone has a good chance of winning. The teams need to be balanced by ability profiles.
- Competitions have always a winner. But they create losers as well - usually many more losers than winners. Individuals suffer at least temporary from embarrassment. Those students who lose constantly lose not only the game, but also confidence, self-esteem, and enjoying the task [13].

2.2 User Types

In the 1990s Richard Bartle analyzed players of Multi-User Dungeons (MUDs) and categorized them into four types to better understand their behaviors—achievers, explorers, socializers, and killers [10]. Marczewski transfers these player types into user types for applications in a non-game context. He argues that in pure games, players want to play the game from the beginning. They are having fun and are playing it willingly, which is not necessarily the case for applications [9]. In the case of MOOCs it could be argued that the users are closer to game players as they are following the courses out of their own will. On the other hand, with emerging enterprise MOOCs and companies encouraging their employees to participate in MOOCs [11] this scenario could be changing. So in any case, having a closer look on Marczewski's player types might be useful.

Marczewski differentiates four intrinsically motivated types—socializers, free spirits, achievers, and philanthropists—plus four extrinsically motivated types—networkers, exploiters, consumers, and self-seekers [9]. There is no „one size fits all" approach for these types as their motivation to go on is inherently differing from each other.

3 SOCIAL NETWORKING SERVICE

The social networking service (SNS) enables users to establish connections to other users. These connections are the basis for all features that require a social context, such as a social leaderboard or an activity stream. The SNS will enable users to import existing friends from social networks as well as to create new connections within openHPI.

3.1 Components

3.1.1 Social Login

The possibility to login with a third party authentication provider is a convenient and by now common way of keeping the amount of user credentials manageable. Figure 1 shows the button to login via an existing SAP ID, which enables SAP employees and customers to login with their existing data. A similar possibility will soon be available for users of online platforms, such as Facebook or Google+.

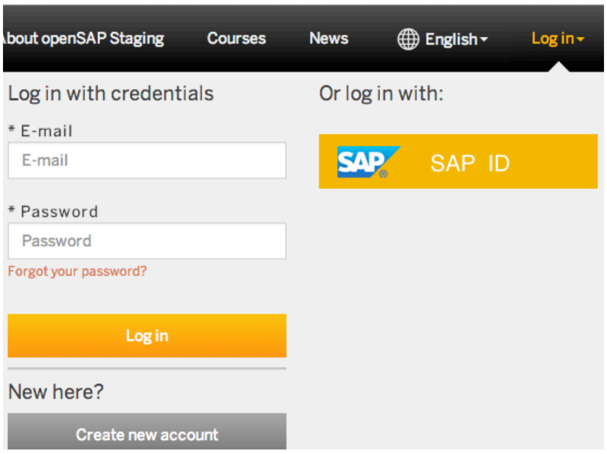

Fig. 1: Social Login via existing SAP ID on openSAP

3.1.2 User Profiles

User profiles are crucial elements in social networks [16]. They enable individuals to present themselves and on the counter part, allow the discovery of new friends. openHPI's user profiles do not have to include the same information as on conventional social platforms. While conventional social platforms, such as e.g. Facebook enable users to find friends who visited the same school or university, openHPI as a MOOC platform does not intend to go that far. The users' main objective is to take courses and gain new knowledge. Instead of explicitly searching for people, who attended the same school, they rather search directly for friends of whom they know that they are registered. Already in 2013, without any SNS features on openHPI, about 33% of the participants knew, that some of their friends attended the same course [2]. The primary goal of the SNS is to support those users to find and connect to each other.

3.1.3 User Search

The user search widget supports the detection of known users. Students can explicitly search for users by entering display names (or nick names), clear names, or email addresses (more details will follow below). The user name and the state of the friend request are displayed (see Fig. 2). For some users this might be a problem in terms of violated privacy feelings. There are different options how to handle the users' identity on a socially connected platform. Trade-offs have to be made between the users' ability to freely use the platform and the reliability of their activities. If participants can hide behind nicknames and conceal their real identity, the barrier to participate in discussions is lowered but also the quality of these discussions might decrease. Especially in a learning environment users should be encouraged to ask questions and give high quality answers. Furthermore, in the context of a MOOC platform, the users also have the option to receive a certificate. On the certificate they want their real name to be displayed.

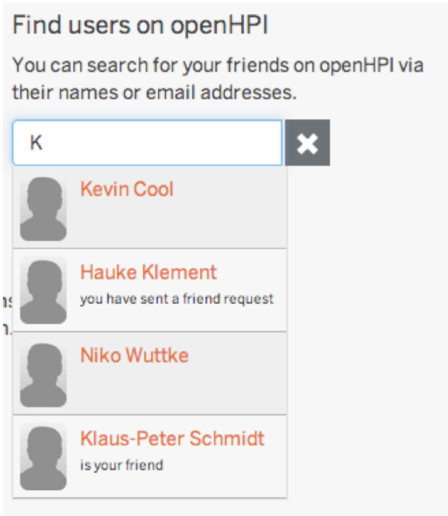

Fig. 2: User search form

Identities can be classified into 3 categories

- Identity—the real identity has to be referenced.
- Pseudo-anonymous—a real identity is pretended by using a newly introduced subject (for example a nickname), by which the connection to the real participator can only be fulfilled by the individual itself.
- Anonymous—no referencing between the activity and the participator is possible [17].

openHPI's SNS uses an identity approach with the possibility to opt-out into pseudo-anonymity. Each user has the possibility to add a display name to her profile, which is stored as an attribute for the user object. Except for the certificates, only the display name is used throughout the platform.

The *display name* is shown in discussion forums, in friend lists or when searched for other users. Users are able to define under which circumstances they can be found on the MOOC platform. As mentioned above, the user profiles, per default, are searchable by real name, email address and display name. The users have, however, the possibility to adjust their settings so that they can only be found by display name.

3.1.4 Friend Request

Friend requests are directed friendship propositions.

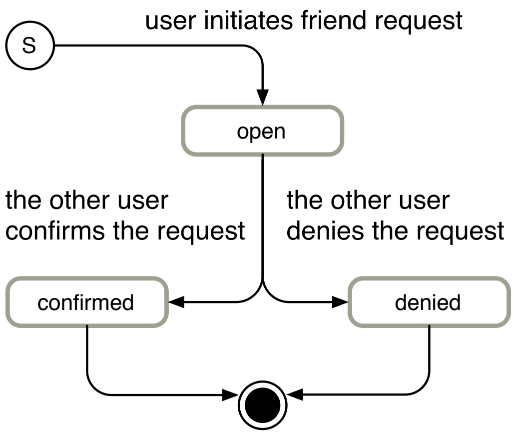

Fig. 3: State diagram of a friend request

Originators invite potential friends to create a link between each other. Friend requests' states reflect the possible conditions: *open*, *confirmed*, *denied*. Upon creation they are *open* (see Fig. 3). When potential friends answer the propositions, they can either deny or confirm them. In case of a confirmation, the friend request's friend is added to the originator's friends. As a result, a friendship between them is created. In case of a rejection, the friend request's state is just changed. When friend requests have been denied, new friend requests can be created again. Initially, this has been implemented differently, to prevent SPAM and obsessive usage. But then users, who accidentally deny a friend request, would not be able to create a friendship with the originator anymore. Other social networks handle this in a similar way.

3.1.5 Invitation

While friend requests enable users to connect to other users within the openHPI platform. Invitations enable openHPI users to invite their friends from other platforms. Invitations are only created via direct user interactions. They can be confirmed through the internal event of new registrations only, when the invitee signs up at openHPI. In general, there exist two types of invitations: *email* and *external provider*. For this work, the focus has been on invitations via external providers. They can be emitted from the external friend finder. If current users have authorized openHPI to access Facebook, they can send app requests with the Facebook JavaScript SDK.

3.2 Applications of the Social Graph

3.2.1 Leaderboard

Leaderboards are a gamification element that has to be handled with extreme care. If applied carelessly, the result might be contradicting the original intention.

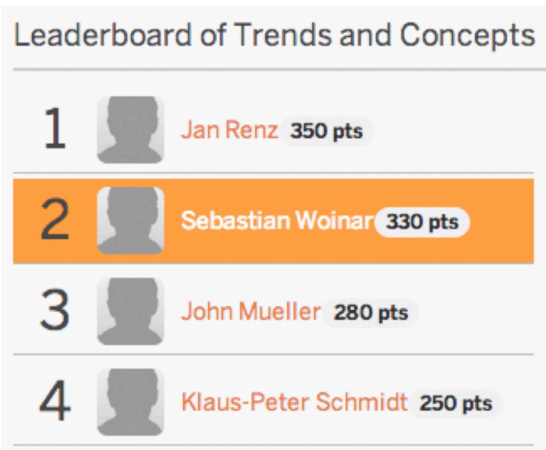

Fig. 4: Social leaderboard

To recapitulate: the intention of all gamification features is to increase and maintain the users motivation to actively take part in the course. Leaderboards come along in different shapes. What they have in common is that they compare the results of a certain user to the results of a list of other users. The difference is which users are included in the list and which are not. Another difference is which points are entering the competition.

Global leaderboards are easily implemented. They just show all users of all courses calculating the sum of the achieved points from all courses. These, however, should be avoided, as they will frustrate new users or those that haven't performed so well as they clearly can see that they do not have the slightest chance to get to the top. Leaderboards that are global, but take only points of one particular course into account, would improve the situation for the newbies a little bit, but still are far from perfect. Relative leaderboards include only those users that are within a predefined range of points compared to the current user either within a course or a platform context. These as the current user, in general, is not related to the other, more or less random, users in the list are by nature not very significant.

Social, relative leaderboards list only friends of the current user, who are enrolled in a particular course. All users start at the same time. Courses on openHPI usually have a duration of 4-6 weeks. Therefore, points should not differ too much and everybody has the possibility to win. As the users in the list have a relation to each other, the leaderboard gains significance and relevance.

3.2.2 Activity Stream

User activities are presented in an activity stream, where, depending on the receiving users, the activities of their friends are shown. Following motivation theory as described in Section 2, relatedness and purpose are important incentives and can drive the user's intrinsic motivation. The purpose of the activity stream is to motivate users to take action.

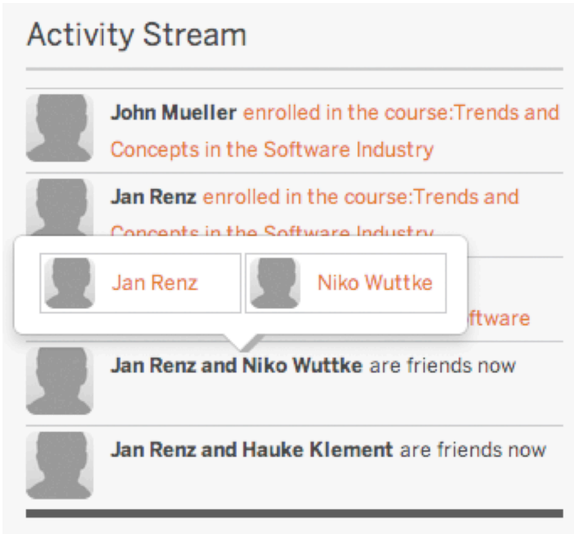

Fig. 5: Activity stream

For example when a user sees that one of her friends has completed an assignment, she is reminded that she had in mind to do that too. The motivation is further increased, as she knows that as soon as she has completed the assignment, this will pop up in the activity streams of her friends, by which they relate to each other.

Activities should be aggregated when multiple friends have completed the same action. For example, if two friends of a user enroll to the course *X* only one activity should be generated that states: *John and Peter enrolled to course X.*

The created activities are filtered and sorted before they are shown to the user. During the filtering, non-relevant activities are rejected, for example *new friendships*, in which the current user is involved—they know, that they entered into a friendship. Only the final result set is I18ned then to avoid unnecessary, expensive computations.

Which activities are shown depends highly on the dynamics of the community. Users should not be overwhelmed by the number of messages, as too many of them will change the users perception. First of all, users focus should remain on the course topic and not on futile events of their friends, and it is very unlikely, that users will read carefully through a long list of activity messages. Therefore, only very important and crucial events should be presented. Table 1 proposes possible user activities and assigns the purpose for which an entry in the activity stream would be appropriate. The event *Enroll in Course* is a powerful incentive for friends, to join the user. The same is true for *Watch a Video*, but the frequency of these video events is very high. Users are provided with more than one hour of videos per week, whereby the comparison of MOOC platforms has shown, that all of the observed learning platforms split the videos into chunks of 5-25 minutes. Thereby, learners have to watch about up to ten videos per week. Even with the aggregation of activities, if users have in average four friends, their activity streams are spammed and they miss other activities. In contrast to *Complete Voluntary Self Test*, just watching a video, does not confirm applied knowledge. Therefore, it was decided not to highlight this activity in openHPI's activity stream. As the pinboard is a crucial part of openHPI, which enables students to consolidate their knowledge and help each other, at least some of the activity there will be shown in the activity stream. Points are the result of previous events, such as the completion of a quiz. As the completion of the quiz is the actual event and way more important than the points that have been earned, only the original event will be shown in the stream.

Table 1: User activities

Event	**Reputation**	**Motivation**	**Community**
Enroll in Course		X	X
Complete Homework assignment	X	X	
Complete Voluntary Self Test	X	X	
Watch a Video		X	
Ask Question on Pinboard		X	X
Answer Question on Pinboard	X		X
Accept Friendship Request			X
Receive Badge	X	X	
Receive Points		X	

3.3 Choice of Framework

Some existing social networking frameworks—*Social Stream*[5], *CommunityEngine*[6], Has Many Friends[7], Lovd by less[8] and Insoshi[9]—have been evaluated in terms of their applicability for the given purpose. An active community and a well maintained documentation is required for the application of a

[5] http://social-stream.dit.upm.es/started/#components

[6] https://github.com/bborn/communityengine/

[7] https://github.com/swemoney/has_many_friends

[8] https://github.com/stevenbristol/lovd-by-less

[9] https://github.com/insoshi/insoshi

web framework. It also was decided to use a *bidirectional* friendship model in openHPI as this engages users to rather maintain a few profound relationships, than a large number of superficial connections. To be used in a platform that is built on distributed services such as openHPI, a lightweight and highly configurable framework is necessary, as the responsibilities are split among various services with high cohesion and weak interconnections. For example, the front end has to work completely independent from the functionality in the corresponding service.

Whereas *Social Stream* and *Community engine* fulfill most of the requirements, none of the frameworks is compatible with Ruby on Rails 4—the framework, which serves as the basis for openHPI. The service-oriented architecture of openHPI also has its special requirements in modularity that none of the frameworks completely fulfills. Due to the incalculability of the necessary update process and the limited benefits of using one single module, it was decided that none of the observed frameworks fits our purposes and that we need our own implementation.

3.4 Choice of Database

The distributed service architecture of openHPI v2 allows selecting the database solution, which is suited best for the purpose of each service. As the name implies, social graphs rely on the persistence of relations and entities, which is exactly the data model of graph databases. Graph databases are storages, which use graphs to persist and query data. They consist of nodes and relationships, which both can enclose properties. This is called a *property graph model* [14].

A typical use case for graph databases is the representation of relationships between humans. That way, social communities with people who like, love, hate or slightly know each other can be modeled. The strengths of graph databases come now handy, when complex queries have to be executed on the data. Highly optimized graph algorithms can be used to traverse graphs (visit each node and follow relations) to find for example the shortest path between two nodes. Graph databases have a processing engine to traverse trees and run fast algorithms. Besides that, they have an execution engine for running analytical tasks.

Compared to Relational Database Management Systems (RDBMS) graph databases have a better performance when querying the relations between data. This is due to the fact, that in RDBMS data has to be joined, which is executed on whole tables, whereas in graph databases queries are localized to a portion of the graph [14]. In other words: a relationship in RDBMS joins data from one or more tables. Therefore these rows have to be found and connected. The lookup is linear to the size of the tables. But in graph databases the nodes are connected with each other and the query can be executed in the limited portion of the graph [14].

In the initial implementation of the service, Neo4J, an open source graph database was used to store the friends and friendships. Its free community edition allows the utilization with basic features. Several Neo4j clients for Ruby exist; it has an active community and a comprehensive documentation. Practical experience, however, taught us that from a maintainability point of view it is desirable to stick with one programming language and also one database system in all services. Woinar [15] describes the challenges and possible solutions during his work with Neo4J in detail. Neo4j requires JRuby and even this only slight change caused a remarkable increase in the maintenance effort for the development environment (which has to be multiplied by the number of developers.) JRuby did not allow yet the use of the newest Ruby syntax. We also found out that Neo4Js performance on simple use cases was rather bad. As our use cases mostly are of simple nature—rather: give me a list of all my friends than find the friend of a friend of a friend—the additional effort of using a graph database was not worth the pain, if not to say counterproductive. So, we switched back to a regular, relational Postgres Database for the productive system.

4 CONCLUSION

A social graph allowing users to contact each other was introduced. Psychological research has shown that relatedness is a strong intrinsic motivator, particularly for some user types. It has been decided to add a social networking service to openHPI as it can serve as the basis and social context for a variety of features in—not only—the context of gamification that rely on a social graph. The application of a social graph allows the amplification and adjustment of the original gamification concept with novel game components in MOOC contexts. Social leaderboards, for example, are considered to be more motivating as they visualize a competition amongst friends rather than random strangers to whom the individual user cannot relate. An activity stream showing the actions of friends is expected to motivate users to engage in some action on the platform. At least, it will remind the user that some action has to be taken to continue with the course.

Even if the service oriented architecture of openHPI v2 allows implementing each service with the best suitable set of programming language and database, the maintenance effort can easily explode. Even if the term social graph suggests that a graph database might be the best choice, this depends strongly on the use cases in which the social graph will be involved. In some cases the RDBMS, if taken all into account, might be the better choice.

5 FUTURE WORK

The new service has been implemented but is not published yet. To determine if the expected results will be achieved—basically, increasing the users' learning outcomes and the courses' completion rates by maintaining the user's motivation over the complete course period still needs to be evaluated. To run such an evaluation, however, it will be necessary to automatically determine the user types by evaluating the users' actions on the platform, as the improvement of social networking features will not affect each user type to the same amount. Furthermore AB-testing tools will be needed to compare the target group against a control group.

A stimulation to enroll in a course could be given by displaying the number of already enrolled friends for the courses in the course list.

REFERENCES

[1] Willems, C. et al (2014). Motivating the Masses – Gamified Massive Open Online Courses on openHPI. In Proceedings of EDULEARN 2014.

[2] Staubitz, T.; Renz J.; Willems C.; Meinel, C. (2014). Supporting Social Interaction and Collaboration on an xMOOC Platform. In Proceedings of EDULEARN 2014.

[3] Grünewald, F.; Mazandarani, E.; Meinel, C. (2013). openHPI - a Case-Study on the Emergence of two Learning Communities. In Proceedings of EDUCON 2013 IEEE, pp.1323, 1331

[4] Deci, Edward L.; Koestner, Richard; Ryan, Richard M. (1999). The undermining effect is a reality after all - Extrinsic rewards, task interest, and self-determination: Reply to Eisenberger, Pierce, and Cameron (1999) and Lepper, Henderlong, and Gingras (1999). Psychological Bulletin, Vol. 125(6), pp. 692-700.

[5] Meinel, C.; Totschnig, M.; Willems, C. (2013). openHPI: Evolution of a MOOC Platform from LMS to SOA. In Proceedings of CSEDU 2013, pp. 593-598.

[6] Deci, E. L.; Ryan, R. M. (2000). The "What" and "Why" of Goal Pursuits: Human Needs and the Self-Determination of Behavior. In: *Psychological Inquiry* 11, October 2000, Nr. 4, pp. 227–268.

[7] Niemiec, C. P.; Ryan, R. M. (2009). Autonomy, competence, and relatedness in the classroom: Applying self-determination theory to educational practice. In: *Theory and Research in Education* 7, June 2009, Nr. 2, 133–144.

[8] Pink, D.H. (2009). *Drive:* The Surprising Truth about what Motivates Us. Riverhead Books. ISBN 9781594488849

[9] Marczewski, A. (2013). Gamification: A Simple Introduction. ISBN 9781471798665

[10] Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit MUDs. In: *Journal of MUD research* 1, Nr. 1, 19. http://mud.co.uk/richard/hcds.htm/, Last accessed: 2014/09/23

[11] Meister, J. (2013). How MOOCs Will Revolutionize Corporate Learning And Development. http://www.forbes.com/sites/jeannemeister/2013/08/13/how-moocs-will-revolutionize-corporate-learning-development/, Last accessed: 2014/09/23

[12] Kohn, A. (1999). Punished by Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes. Houghton Mifflin Harcourt, ISBN 9780547526157

[13] Brophy, J. (2004). *Motivating Students to Learn*. Taylor & Francis, ISBN 9780805847727

[14] Robinson, I.; Webber, J.; Eifrem, E. (2013). *Graph Databases*. O'Reilly Media, Incorporated, ISBN 9781449356262

[15] Woinar, S. (2014). Social Graph for Gamified MOOCs. Master's Thesis, Hasso Plattner Institute

[16] Lampe, C.; Ellison, N.; Steinfield, C. (2007). A Familiar Face (book): Profile Elements as Signals in an Online Social Network. http://www.pintrestalk.com/ddata/ 2147.pdf, pp. 435–444. ISBN 9781595935939

[17] Collaboratory, I.G. (2013). Gleichgewicht und Spannung zwischen digitaler Privatheit und Öffentlichkeit. http://books. google.de/books?id=IpBWImSlMtkC. ISBN 9783950313932