International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 3 (2011) pp. 427-434

Interactive 3D Graphs for Web-based Social Networking Platforms

Nikita Mattar and Thies Pfeiffer

A.I. Group, Bielefeld University,
Universitätsstr. 25, 33613 Bielefeld, Germany
{nmattar,tpfeiffe}@techfak.uni-bielefeld.de

Abstract: **Social networking platforms (SNPs) are meant to reflect the social relationships of their users. Users typically enter very personal information and should get useful feedback about their social network. They should indeed be empowered to exploit the information they have entered. In reality, however, most SNPs actually hide the structure of the user's rich social network behind very restricted text-based user interfaces and large parts of the potential information which could be extracted from the entered data lies fallow. This article presents results from a user study showing that 3D visualizations of social graphs can be utilized more effectively – and moreover – are preferred by users compared to traditional text-based interfaces. Subsequently, the article addresses the problem of how to deploy interactive 3D graphical interfaces to large user communities. This is demonstrated on the social graph application FriendGraph3D for Facebook.**

***Keywords*:** social networks, interactive graphs, information visualization, 3d graphics, web technology

I. Introduction

Online social networking platforms (SNPs) are, as of 2011, very popular. Millions of users have entered their profiles and defined their relationships, if not more. This sensitive information is collected in large databases where it is in principle easy to digest by data mining. There exists a cornucopia of scientific tools to analyze and visualize social networking data on a large scale, some of which will be discussed in the related work. Also, there is a huge interest in those data for marketing and especially advertisement. Consequently, the provisioning of private data to the public and the potential abuse of this data by third parties is source of persistent debates, both in science and public.

In contrast to the general discussion, the focus of our work is not on the data in the large. Instead, we focus on individual users and their personal social network. Our perception is that users enter many details about themselves and their social network. In doing so, they make use of descriptive functions offered by the SNPs to declare facts about their lives, their jobs, their residences and with whom they are friends (or not). Some platforms also provide means to declare in a case-based fashion what users like or where they currently are.

In addition, besides these declarative statements, users act on the SNPs: they communicate with friends, comment on pictures or events, and digest digital media. All these actions implicitly add to the overall picture of the users' profiles including but not restricted to their personal social networks.

At the same time, we also perceive that most of this information is either not accessible or not presented in a way easily to consume for the majority of the users. What is displayed on the so-called profile pages of individual users is only a glimpse of the profiles that could have been created from the declared or collected data. Thus, the data entered by the users is not turned into information that can be easily digested and used by them.

For example, most social networking platforms simply lack presentations beyond serialized lists of first grade relationships. While this approach is suitable for finding known entities in one's social network, it does not provide a straight answer to questions like, e.g., which of one's friends are also friends or what common interests they share, and it makes it difficult to discover something new in the structure of the network.

We expect at least two benefits from a rich presentation of the data collected by the SNP. Rich is meant here both in terms of completeness and accessibility, i.e. as much of the collected data and of the possible inferences which could be made should be reported back to the user and the way of presentation should make this data easily accessible to them. The first benefit is that users would get a much clearer picture of what they are actually revealing, which would enable them to correctly consider their position regarding their privacy in the first place. The second is that if they choose to continue using the SNP services, they could make much more use out of it, such as in the examples given in the previous paragraph.

In this article, which is an extended version of a previous contribution to the IADIS WVR3DW conference [1], we concentrate on the relations of users to others, i.e. the social network per se. In the first part of the paper, we argue that graph-based visualizations, which exist since the early 20th century, are well suited to emphasize the structure of the social network. They allow for an intuitive exploration and increase efficiency in certain tasks. We substantiate this claim and extend it from 2D to 3D graphs by reporting on a small user study, where we compared the current text-based interface with a 3D graph interface of the SNP last.fm. In the second part, we address the problem of disseminating such an interface to a large community by giving an explicit example on how such a 3D interface can be deployed as App on the SNP Facebook.

II. Related Work

The visualization of social networks as graphs has a long tradition in social sciences. Freeman [2] reviews the evolution of techniques for visualizing social network data as graphs and attests first examples for the beginning of the 20th century. Shape, color, position and the size of nodes can be identified as the main features for presenting distinct views on the data. However, the presentation of still images of a

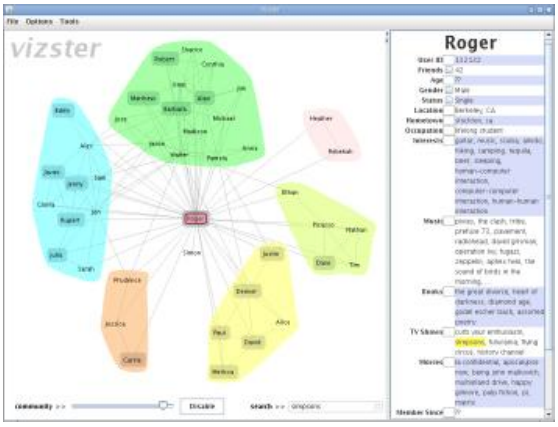


Figure 1. Viszter is a standalone Java application for end-users. It displays interactive 2D graphs of social networks. The graph structure and detected clusters are clearly visible. A rich set of auxiliary information and many filtering functions are available.

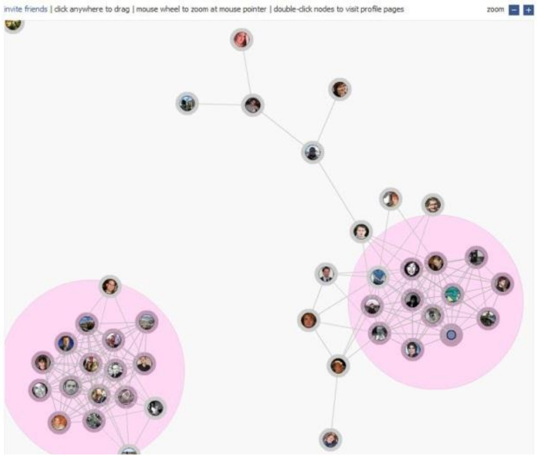


Figure 2. Social Graph is a Facebook App based on Flash. It generates an interactive 2D graph of one's friends and highlights detected clusters.

network is not always suitable if one has to extract something meaningful. Today's social networks are very complex; they support many different attributes and relations connecting the individual nodes. Consequently, Perer and Shneiderman [3] stress the importance of adequate mechanisms for interactive exploration of social network data. They present a rank-by-feature framework that enables network analysts to systematically detect certain patterns in the structure of a network. In a later work Perer and Shneiderman [4] present findings from several case studies evaluating their Social-Action framework. SocialAction provides a combination of visualizations and a toolkit for social network analysis mainly designed for researchers. They conclude that researchers can benefit from tools that integrate visualization and tools for the exploration of networks.

While today graph-based tools for the scientific analysis of networks on a large scale are broadly available there is still a lack of sufficient tools for end-users of social networks to gain a deeper insight in the structures underlying the data they are providing. Heer and Boyd [5] try to address this problem with their Vizster tool (Fig.1). Vizster provides an interface designed for end-users to visualize and explore social networks. They evaluated Vizster in two settings concluding that the possibility of the interactive exploration encourages users to playfully examine their networks in more detail.

Playful examinations are especially encouraged if the interface is highly reactive to user input, such as the physics-based interaction method for manipulating networks presented by Spritzer and Freitas [6]. In their MagnetViz tool they use a magnet metaphor to provide users with means to cluster nodes based on certain attributes. A similar tool, called SONAR, Social Networks in Virtual Reality, was presented by us for the interactive exploration of last.fm in 3D, both on the desktop and in an immersive Virtual Reality installation [7].

A disadvantage of the presented tools is that they are standalone applications which are not fully integrated into the interface of the SNP. This makes such tools less accessible for the typical user. One of the rare applications which embed into SNPs is SocialGraph [8], a Flash application, or App, on Facebook, which uses a dynamic force-based layout to arrange personal friends on a 2D plane (Fig. 2). A special feature of SocialGraph is the identification of clusters of friends, which are highlighted by circles. Thus SocialGraph provides a unique benefit over the web interface: it allows the users to clearly identify the social groups they are a part of. Interaction, however, is restricted to moving around nodes. A second example is the TouchGraph Facebook Browser [9], which provides a visualization of friends and groups (Fig. 3). Both apps require special browser plugins (Flash for SocialGraph and Java for TouchGraph), which restricts their use to certain platforms and makes them less accessible.

Besides task-specific advantages of graph-based interfaces to SNPs, the online-feedback provided by an augmented visualization of the network also influences the communication behavior of the users. This has, inter alia, been investigated in the frame of the EU-funded (FP6) PASION project (Psychologically Augmented Social Interaction Over Networks) [10]. Martino and colleagues tested the effect of visual feedback about the users' centrality within the network on their communication behavior [11]. They found that the feedback had a positive effect and led to an increased communication activity which remained stable over multiple sessions.

III. Interactive Graphs for Social Networks

Based on our findings on SONAR [7], which have been previously published in German, so we report the essential findings in 3.2. for your convenience, we advocate the use of 3D graph interfaces for SNPs. This, however, rises the following questions:

a) What kind of information can be extracted from a user's social network?
b) How do users benefit from the use of interactive 3D graph-based interfaces?
c) How can these interfaces be deployed to end-users such that they are seamlessly embedded into web-based interfaces of SNPs?

A. *Overt and Covert Structures of Facebook's Social Graph*

In order to answer the first question (a) we examine the social network Facebook. Facebook is one of the most popular SNPs and due to the broad range of personal information which can be shared, it can be used to illustrate what kind of data is present, but is hidden from the user by the interface.

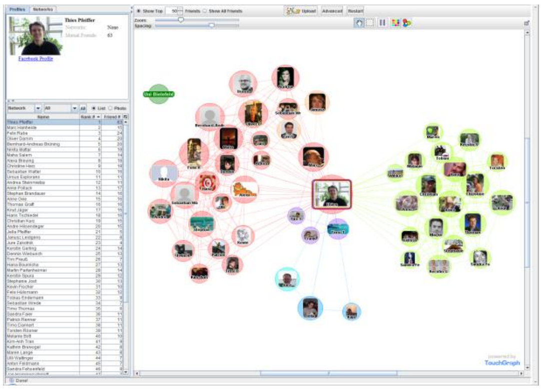


Figure 3. Facebook TouchGraph is a Java Applet which can be launched as Facebook App. It offers similar functions as Social Graph, but adds support for Facebook Groups and provides more details about friends.

Friendship is the constructing relationship of Facebook's social network. Users create and manage individual profiles on Facebook. They can express their friendship to other users by linking their profiles, which is a reliable declaration as both participants have to agree upon this relation. Figure 4, left, shows the textual list which is the kind of interface element provided natively by Facebook to give feedback on the established friendship relations. This list gives a complete, but spatially limited view, about the relations a user has established. This view, however, is already disclosing information, because it does not consider the relations that have been established by other users in the same way. A graph that also takes these mutual friends into account is depicted in Figure 4, right, for the same set of users. It can easily be seen that the graph-based visualization reveals the clustering of friends, which is an added value and closer to the internal profile of the user known to the SNP.

This friendship-relation is rather static, once accepted it would not change until the relation is revoked. Note also, that the friendship relation not necessarily coincides with a friendship in the real world. It could also express family relations, relations between co-workers or business contacts, among others. However, they are at least two sources of information made available by Facebook which are more conclusive about the relationships: **communication events** and evidences of **real-life encounters**.

The structure-giving friendship-relation is an example of an overt structure, as it is directly accessible to the user and to visitors. But there is a rich covert structure constructed by **communication events** which can be interpreted as relations. For example, users can directly exchange messages or they can upload images (authorship-relation), identify and tag other users on photos or comment images. Comments are a universal tool provided by Facebook. Nearly everything can be commented by other users, be it messages, entries on the pinboard, photos, events or comments themselves. As the friendship-relation offers only a weak qualification of the link between users, the number and kind of social interactions which can be observed on Facebook, might provide a more thorough picture of the actual relationship between individuals and groups. The mining of the real social network from these interactions is currently a hot research topic.

All of these communication events happen in the virtual environment constructed by the social networking platform. The users not necessarily have met in real life. Another service offered by Facebook provides also an index for

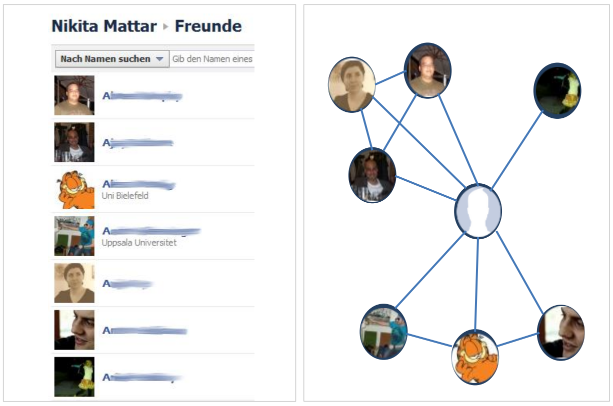


Figure 4. Excerpt of a user's friends list on Facebook (left). Example of a graph visualization with clusters (right).

real-life encounters: users can upload photos and identify other users who are depicted on a photo by tagging them. The presence of two users on one or more photos constructs a relation *have-met-in-real-life*, weighted by the number of occurrences. This relation, however, has to be considered with care. If there is no photo showing a specific pair of users, this does tell nothing about the *have-met-in-real-life* relation. A second way to identify whether two users have met in real-life is to consider the event-management pages. Events, such as parties, can have individual pages on Facebook which can be used, e.g. for organizational purposes. Among other things, users can be invited to events and they can state whether they have participated at the event or not.

However, as stated earlier, there is a lack of convenient tools for users to explore and make use of these overt and covert information to their full extend.

B. *Usability Analysis of 3D Social Graphs*

When trying to answer question (b) several aspects must be taken into account: Can users intuitively handle 3D graph-based interfaces? What kinds of tasks graph-based interfaces are most suitable for? Do users prefer 3D graphs over serialized presentations of information?

Previously, we presented SONAR, an interactive 3D graph visualization of the enriched social network defined by last.fm [7]. While the network constructed declaratively at last.fm is less complex than that of Facebook, it features many prominent high-level relations, where Facebook is dominated by the Friendship relation. Last.fm covers users, artists, albums and songs, as well as relations of different kinds between them. It also offers inferences, such as similar artists and songs, thus augmenting the information declared by the users with an added value.

SONAR is targeted at desktops and immersive Virtual Reality installations. Its implementation is based on X3D, an ISO standard with a declarative language for 3D content [12], and the additional functionality for immersive set-ups provided by the instantreality framework [13]. SONAR allows users to literally immerse into the social network. Relationships between users, their favorite music and artists, or similar artists can be interactively explored (Fig. 5). The users are able to expand and follow the different relations offered by last.fm. They could, e.g., explore all songs of an artist and trace down other users who like a certain song. Furthermore, the user is able to rearrange nodes by dragging them around. Consistency of the layout is maintained by the

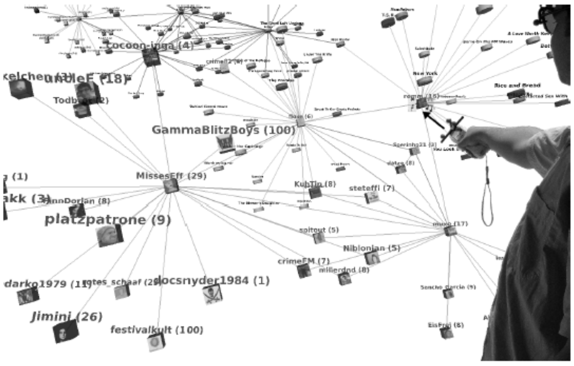


Figure 5. SONAR enables users to interactively explore their last.fm social network in a VR setup.

interactive layouting algorithm. A zoom-in/-out and search functions allow users to keep an overview or to focus on specific subparts of the network.

Thus, users are enabled to get a deeper understanding of their relationships and to identify patterns, which in the case of the music network last.fm might lead to the discovering of new interesting artists. We used this application as test bed to see, whether a 3D interface for social graphs can be understood by novice users and if such an interface can provide advantages over the text-based web interface.

In order to compare the web-based frontend of last.fm and SONAR, we defined tasks dealing with typical problems in social networks: exploration and search.

The tasks were of three categories: (1) "Is A a friend of B?", (2) "User A likes song B, find a similar song.", and (3) "Find three mutual friends of users A and B.".

The first category deals with direct relationships that are typically presented on a single page (cf. Fig. 4, left). Category (2) requires up to two page transitions on the web frontend of last.fm and can be categorized as a straight depth exploration of the network. The third category requires the user to follow at least two relations in parallel, a rather demanding task to perform with the web frontend. The tasks were designed to be of increasing complexity regarding the overview of the structure of the local social network excerpt required to solve the tasks. It was expected that tasks of category (1) were in favor of the text-based interface whereas tasks of category (3) were clearly in favor of the graph-based interface.

In a user study, 16 participants between 20 and 30 years, students at Bielefeld University and active users of SNPs, were asked to perform these tasks [7]. Both applications were tested on the desktop.

It turned out, that in all types of tasks users were more efficient in terms of time-on-task using the 3D graph visualization of SONAR than with the web-based frontend (Fig. 6, top). In addition to performance measures, the users were also asked which tool they preferred for each group of tasks. For tasks like (1), 69% preferred the web-based frontend over SONAR, but for (2) 75% preferred SONAR and for tasks like (3) all participants preferred SONAR (Fig. 6, bottom). When asked directly, 83% of the users also believed that they performed overall faster using SONAR.

The participants of the study were novices to 3D social graphs. Hence, the results for the 3D graph visualization in the study suggest that the 3D graph visualization can easily be understood and be used purposefully. The fact that the 3D graph visualization is also preferred over the web-based

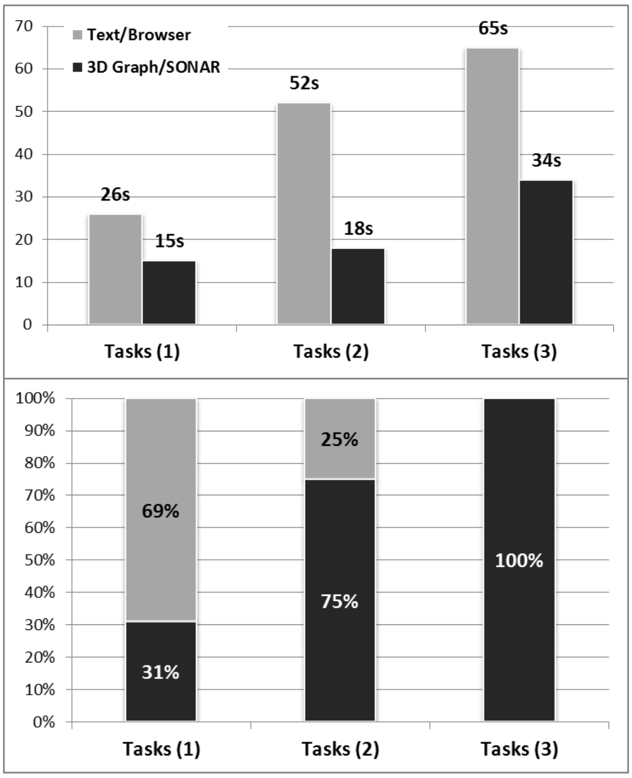


Figure 6. Comparison of time-on-task between SONAR and the web frontend of last.fm (top). User preferences of SONAR and the web fronted for the 3 tasks (bottom).

presentation for tasks were the structure of the social network is highly relevant also suggests that such visualization is suited for typical tasks which are expected to be performed on a SNP.

Since we only compared a 3D graph interface and a typical web frontend, an open question remains if 3D graphs offer an advantage over 2D graph-based interfaces. Halpin et al. (2010) compare a 2D graph interface and a 3D graph for semantic social networks in an immersive Virtual Reality setup. Their findings show that the additional third dimension adds to the overview, supports users in tackling certain complex tasks, and therefore conclude that 3D graphs can be beneficial over 2D graph interfaces.

The positive feedback from the participants of the study of Halpin et al. [14], the evaluation of Vizster [5], and our own results show that graphs not only help to better understand one's social network, but that they are also fun to use. A precondition for a widespread use of such tools, however, is their accessibility. The aforementioned systems are standalone applications, which makes them less accessible for a broad range of users, who may have no access to supported platforms, are not allowed to install applications, e.g. because they are using SNPs in a company or from an Internet Café, or have a general fear of installing software from the Internet. Also, with standalone tools, the user experience a break in the workflow between the web-based frontend and the application.

C. *Technology*

We demonstrated that graph-based interfaces are accepted by end-users and that they support users by accessing information from their social networks. But why, to the knowledge of the authors, do current social networking

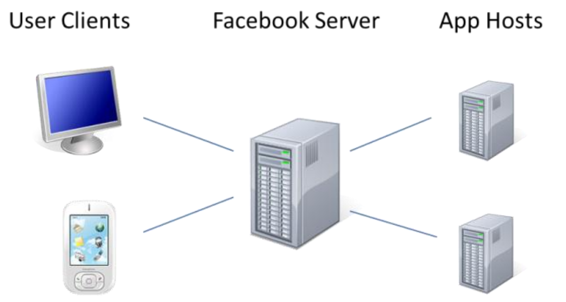


Figure 7. Facebook requests third-party applications from app hosts, embeds them into its own webpage, and delivers them to the client.

platforms not use graph-based visualizations natively? There are two possible reasons we can think of which might hinder SNP operators from making use of 2D or 3D graphs: First, interactive graphics used to require additional tools such as Flash or Java and current web technology restricts the use of these plugins to certain platforms. Second, it was not possible to embed 3D graphics seamlessly into webpages in a platform independent manner.

In order to deploy graph-based interfaces within a SNP some kind of infrastructure is needed. Facebook offers a technology that allows developers to embed their tools into Facebook, so called apps or social widgets. If a user client requests an application Facebook handles the communication with the app hosts, embeds the application into its own webpage, and delivers it to the client (Fig. 7). In that, there is no break in workflow since the user does not have to switch back and forth between the SNP and the application. Since there is no need to install additional software and applications feel like they are a part of Facebook, a broad range of users can be reached.

In fact, several implementations of social graph visualizations exist as apps or social widgets based on Flash. The use of Flash, however, restricts the use of such visualizations primarily to the desktop, as support for Flash is not common on mobile devices (see e.g. the discussion Apple vs. Adobe [15]). Mobile devices on the other side are at the heart-beat of any SNP, as at the time being the networks live primarily from their immediacy. Another disadvantage of apps based on Flash is, that they do not integrate seamlessly into webpages. In that, actions in the user interface of a Flash application cannot easily affect the webpage and vice versa.

Frameworks based on WebGL [16] in conjunction with JavaScript could offer a solution to the concerns mentioned above. WebGL enables web developers to embed rich 3D content into webpages, which can be rendered without using proprietary plugins. JavaScript, on the other hand, allows for interaction between the 3D content and the HTML UI of the webpage. At the beginning of 2011, two major browsers, Firefox and Google Chrome, already support WebGL by default and make 3D graphics available to a broad range of end-users. Browsers with WebGL support for smartphones are under development and first demos for maemo and Android OS have been presented.

IV. A 3D Social Graph App for Facebook

The idea of our application FriendGraph3D [17] is to reveal the underlying structure of the user's Facebook network and to enable the user to interactively explore it. Fig. 8 depicts our

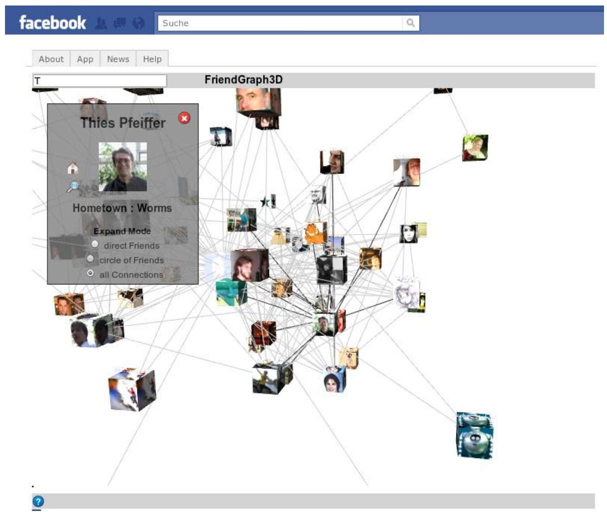


Figure 8. FriendGraph3D enables the user to get a feeling for the structure of his friends network. Additional information is shown in an overlay.

Facebook app in action. Due to the layouting algorithm, clusters of friends are easy to detect and the user can dig deeper into these subnetworks. By zooming in and selecting the level of shown branches the user can customize the view on his personal network. For each friend additional information can be presented as an overlay. A search function assists the user when looking for a specific friend.

In the following, the issue of deploying interactive 3D visualizations on the web is demonstrated, by highlighting the interaction between HTML, JavaScript and the Facebook server on selected features of our Facebook App for 3D social graphs.

A. Integrating HTML5 and X3D: X3DOM

Until now there were only limited possibilities for the use of 3D graphics in browsers. As most web developers are used to the declarative nature of HTML, low-level APIs for 3D content, e.g. plain WebGL, can be cumbersome to integrate within an existing webpage. The possibility of embedding 3D content in a declarative manner similar to 2D content declared in HTML would allow for an easy adaption of this new technology. High-level APIs fulfilling this requirement can assist web developers to mix traditional HTML-styled webpages with interactive 3D content.

Behr and colleagues [18] presented the open source framework X3DOM which builds on top of WebGL. In contrast to e.g. WebGLU [19], X3DOM is capable of directly integrating declarative X3D scenes into the HTML5 DOM tree (Fig. 10, bottom left), enabling the use of advanced JavaScript technologies, such as Ajax or JQuery, to interact with the 3D content. Furthermore, recent browsers no longer require additional plugins such as solutions based on Flash or Java among others.

For the embedding of the 3D scene which holds the interactive social graph, the X3DOM integration model [18] is used. Fraunhofer IGD provides a reference implementation based on JavaScript [20] under the MIT/GPL dual license. The project is hosted on sourceforge.net for everyone to contribute.

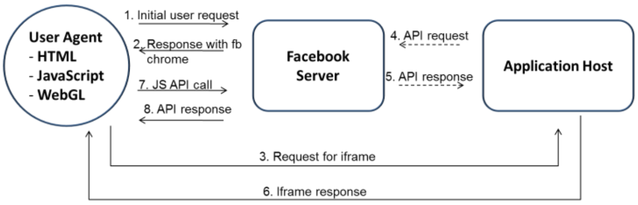


Figure 9. Communication flow between user agent, Facebook server and application host.

B. Communication Flow

As mentioned in 3.3., Facebook does not host third-party applications on their servers. Therefore, an additional server is needed for applications. Following figure 9, the communication flow between the user agent (e.g. browser), the Facebook server and the application host is described.

In order to grant the application access to the user’s data (4) and (5), it is necessary to authenticate the user with an API provided by Facebook. Once this has been done and the user is properly authenticated, the user agent requests an application from within Facebook (1), Facebook responds with its so called Facebook chrome (Facebook's user interface) (2) and an embedded iframe for the application. The user agent requests the content of the iframe (3), the application is delivered by the application host and rendered into the iframe by the user agent (6). Subsequent JavaScript API requests are conducted by the user agent and handled by the Facebook server (7, 8).

The visualization of the social graph is constructed in real-time and asynchronous by retrieving and analyzing relevant data from the Facebook servers. This is all done in JavaScript on the client side. A function (Fig. 10, line 1) is registered as callback function (Fig.10, line 11). Facebook’s server calls this function every time a new friend is detected. Within the function new DOM elements are created (Fig. 10 lines 2-4) and attached to the existing DOM tree (Fig. 10 line 8).

The X3DOM implementation monitors the DOM for changes and mirrors them in the scenegraph which is internally constructed and hidden behind the high-level X3D API. In this case a new node is attached to the scenegraph (Fig. 10, bottom right). X3DOM can optimize the declarative specification of the scenegraph given by the programmer and create a scenegraph which is optimized for performance, tailored either to the WebGL implementation or, if available, to a special purpose X3D plugin.

C. Interaction

As a starting point, FriendShip3D reveals the Facebook’s friendship relationship of the user. That is, the user and the first-grade relationships, the friends that have been registered by the user.

Currently, our application focuses on the exploration of this network. The user is able to pan, rotate, and zoom the network, in order to bring a region of interest into focus. If the user selects one of his friends, by clicking on the 3D representation, an overlay with additional information is shown. The user is able to visit his friend’s profile page, from within the overlay, or center and zoom the view on the selected friend. Furthermore, different levels of connectivity

```
1 : function UserCB(u_infos) {
2 :   var u42 = document.createElement("Transform");
3 :   var u42_s = document.createElement("Shape");
4 :   var u42_b = document.createElement("Box");
5 :   u42.setAttribute("id", u_infos.name);
6 :   . . .
7 :   u42.appendChild(u42_s);
8 :   document.getElementById("users").appendChild(u42);
9 : }
10:
11: FB.Facebook.apiClient.users_getInfo(id,r,UserCB);
12: . . .
13: $("Transform[id^='"+search+"']").each(function() {
14:     highlight($(this)[0], true) } );
```

```
<html>
 <body>
  <x3d>
   <transform id='Thies'>
    <shape>
     <box />
    </shape>
   </transform>
   . . .
```

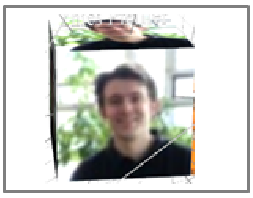


Figure 10. A JavaScript callback manipulates the DOM tree if a new friend is added (top). Excerpt of the corresponding DOM tree (bottom left). X3DOM visualization of a friend (bottom right).

can be selected for visualization: direct friends, circle of friends, and all connections.

D. Search

To allow quick access to a specific friend in the social graph a search box is presented to the user (Fig. 12, left). JavaScript is used to query the friend list (Fig. 11, path a) and the user is assisted with an auto-suggestion feature while he types. Additional visual feedback is provided by highlighting nodes in the 3D graph with an id partly matching the search string. A JQuery statement is used to trigger the highlighting of the nodes (Fig. 10, lines 13-14).

On a successful search request the X3D-part of the DOM tree is altered from within the JavaScript backend (Fig. 11, path b): the viewpoint of the 3D scene is panned in order to bring the requested user into focus.

Since the friend list was retrieved at the start of the application, no further communication with the Facebook server is required. The above JQuery statement solely operates on the DOM tree. At creation time of the node, the user id was set as an attribute (Fig 10, line 5), and therefore can be retrieved through the query above.

In this case a search bar was added as an additional UI element to the web page (Fig. 12, left). The integration of the search feature into an existing HTML UI element is possible, due to the fact that the HTML and X3D part of the DOM tree can interact. This enables one to make use of existing HTML-/CSS-based UI frameworks and therefore allows a seamless integration of 2D and 3D UI elements into the look-and-feel of a web page.

E. Layouting into Clusters

Layouting of the graph is done using a simple force-based layouting algorithm. The layout algorithm is time consuming and thus the recently developed WebWorker threads are used to run the algorithm in the background whenever the browser offers this functionality. The background thread, however, does not have direct access to the DOM tree and is thus also not able to read or update the current positions of the nodes in

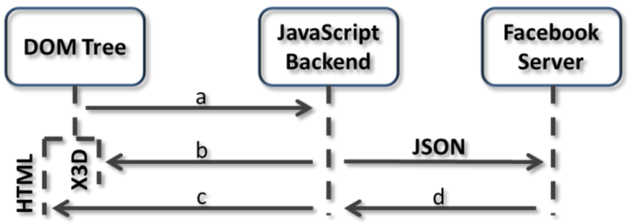


Figure 11. Interaction between HTML, X3D, JavaScript backend and the Facebook Server.

the graph. The relevant information has to be exchanged via messages between the main thread with DOM access and the background thread doing the layout. This ensures at the same time a synchronous update of the 3D scenegraph.

A feature of force-based layout algorithms is that they move together highly connected nodes such that visually detectable clusters emerge (Fig. 12 right). The created clusters are formed by the interconnectedness of the regarding users. Other clustering features are possible, such as geographic distance or participation in certain groups, but are not based on the friendship relation. For an overview of different ways for clustering refer to e.g. [21].

V. Conclusion

We started out from the observation that users of SNPs get little feedback about the data they provide to the platform. We thereby distinguished between declarative and implicit data input, which when taken together can lead to an augmented profile. This profile can be either overt or covert in the SNPs' user interfaces. This observation is considered in greater detail in Section 3.1.

We also argued that more transparency regarding the collected data would on the one hand enable common users to make better informed decisions about their privacy. On the other hand, users would get a greater benefit from the collected data, which could lead, as the related work and our own study shows, to an increase in efficiency. In addition to that, an augmented interface could also open the realm for new tasks which can be performed using the SNP.

An open question is how the underlying data of SNs can be prepared and presented to provide valuable information to the end-user to support those goals. In the social sciences and other disciplines, graph-based visualizations of social networks have been offered as one answer to this question. These graph-based tools, however, are primarily created for scientists and for the analysis of networks on a large scale. Our research reveals that only a small number of tools actually make use of graph-based visualizations in the context of SNPs for end-users.

The related work suggests, that such graph-based interfaces could indeed be helpful and, even more, be enjoyable to use. A claim that is substantiated by our own research presented in Section 3.2. There we compared the user interface of the SNP last.fm with a new designed immersive graph-based interface called SONAR. Users are not only faster using SONAR, they also preferred to use SONAR over using the existing interface when conducting tasks of medium or high complexity.

After the potential of graph-based user interfaces for SNPs has been attested, we discussed in Section 3.3 whether the lack of a broader availability of such interfaces may be due to technological reasons and problems with the deployment to a large user base, as SNPs are primarily web-based and

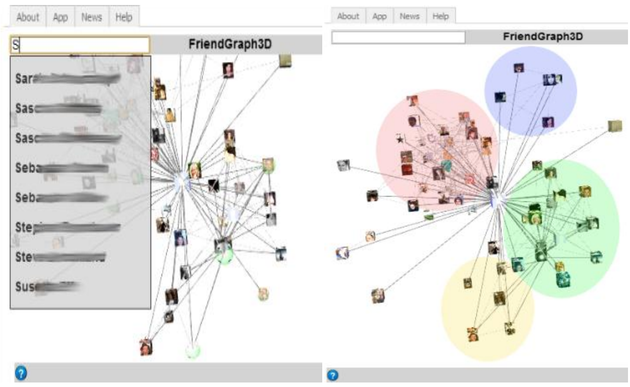


Figure 12. A search box with auto-suggest feature helps to find certain friends (left). Clusters are easy to detect in one's friend network (right).

complex graphical interfaces have been difficult to realize in the past. We argued that WebGL could offer a viable alternative for rich graphical interfaces in the future.

In Section 4, we finally demonstrated on the example of an App for Facebook called FriendGraph3D, how such a graph-based interface for end-users can be realized using the high-level WebGL-wrapper X3DOM. We describe the basic exploration and navigation interactions that have been realized and provide selected examples to highlight the smooth integration between 3D content and traditional HTML/JavaScript content. It is our experience that X3DOM makes creating interactive 3D applications feel quite WWW-ish, an argument which could make this particular Web3D technology appealing to a broad range of new users.

A. Outlook

The presented prototype FriendGraph3D only visualizes a small subset of the covert structure hidden in Facebook. Future work would include the identification of combinations of potential tasks, information relevant for these tasks and appropriate visualizations. The small number of participants in the presented study (Section 3.2), however, is not sufficient to answer such substantial questions. This reveals a further advantage of the presented approach to embed the interface into an SNP as app: We have now access to a very large community to test alternatives, get qualitative and quantitative feedback, and to do long-time observations of usage.

References

[1] Mattar, N. and Pfeiffer, T. 2010. Relationships in Social Networks Revealed: A Facebook App for Social Graphs in 3D Based on X3DOM and WebGL. In *Proceedings of the IADIS International Conference Web Virtual Reality and Three-Dimensional Worlds 2010*, 269-276: IADIS Press.

[2] Freeman, L.C. 2000. Visualizing Social Networks. In *Journal of social structure*, 1.

[3] Perer, A. and Shneiderman, B. 2006. Orderly Analysis of Social Visualizations. *Social Visualization Workshop at CHI 2006.*

[4] Perer, A. and Shneiderman, B. 2008. Integrating statistics and visualization: case studies of gaining

clarity during exploratory data analysis. In *CHI '08: Proceeding of the 26th annual SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, USA, pages 265–274.

[5] Heer, J. and Boyd D. 2005. Vizster: Visualizing Online Social Networks. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, p.5ff.

[6] Spritzer, A. S. and Freitas, C. M. 2008. A physics-based approach for interactive manipulation of graph visualizations. In *Proceedings of the Working Conference on Advanced Visual interfaces*. ACM, New York, NY, pages 271-278.

[7] Bluhm, A., Eickmeyer, J., Feith, T., Mattar, N., and Pfeiffer, T. 2009. Exploration von sozialen Netzwerken im 3D Raum am Beispiel von SONAR für last.fm. In *Virtuelle und Erweiterte Realität – Sechster Workshop der GI-Fachgruppe VR/AR*, Shaker Verlag GmbH, Aachen, Germany, pages 269-280.

[8] Phillips, R. et al., 2010. Social Graph. [Online] Available at: http://www.facebook.com/apps/application.php?id=67692068407 [Accessed 28.02.2011].

[9] TouchGraph, 2011. TouchGraph Facebook Browser. [Online] Available at http://www.touchgraph.com/facebook [Accessed 28.02.2011]

[10] Brugnoli, M. C., Morabito, F., Walker, R. and Davide, F. 2006. The PASION Project: Psychologically Augmented Social Interaction Over Networks PsychNology, 4, 103-116.

[11] Martino, F., Baù, R., Spagnolli, A., and Gamberini, L. 2009. Presence in the age of Social Networks: Augmenting Mediated Environments with Feedback on Group Activity. Virtual Reality, 13 (3), 183-194.

[12] Daly, L. and Brutzman, D. 2008. X3D: Extensible 3D Graphics Standard. In *ACM SIGGRAPH ASIA 2008 courses* (SIGGRAPH Asia '08). ACM, New York, NY, USA, Article 22, 6 pages.

[13] Fraunhofer IGD, 2010a. instantreality. [Online] Available at: http://www.instantreality.org [Accessed 28.02.2011].

[14] Halpin, H., Zielinski, D. J., Brady, R., and Kelly, G. 2010. Exploring semantic social networks using virtual reality. In *The Semantic Web-ISWC 2008*, Springer, pages 599-614.

[15] Copeland, M. V. 2010. Behind the Adobe-Apple cold war. [Online] Available at: http://tech.fortune.cnn.com/2010/01/29/behind-the-adobe-apple-cold-war [Accessed 28.02.2011].

[16] Khronos 2009. WebGL public wiki. [Online] Available at: http://www.khronos.org/webgl/wiki/Main_Page [Accessed 28.02.2011].

[17] Mattar, N. and Pfeiffer, T. 2010. FriendGraph3D. [Online] Available at: http://www.facebook.com/apps/application.php?id=111183342249022 [Accessed 21.02.2011].

[18] Behr, J., Eschler, P., Jung, Y., and Zöllner, M. 2009. X3DOM – a DOM-based HTML5/X3D integration model. In *Proceedings Web3D '09*, ACM Press, NY, USA, pages 127-135.

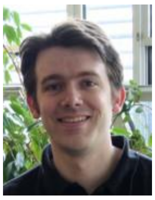
[19] DeLillo, B. 2010. WebGLU. Development library for WebGL. In *ACM SIGGRAPH 2010 Posters* (SIGGRAPH '10). ACM, New York, NY, USA, Article 135, 1 page.

[20] Fraunhofer IGD, 2010b. X3DOM. [Online] Available at: http://www.x3dom.org [Accessed 28.02.2011].

[21] Scott, J. 1988. Social Network Analysis. In *Sociology*, 22(1), 1988, 109-127.

Author Biographies

Nikita Mattar studied Informatics in the Natural Science at Bielefeld University, Germany. In 2009 he received the diploma degree after finishing his thesis on Level Set Surface Editing in the Computer Graphics Group. Since 2009, he is a member of the Artificial Intelligence Group at Bielefeld University, Germany. His current research interests include human-like memory for virtual agents, human-computer interaction, and social networks.



Thies Pfeiffer defended his doctoral thesis on understanding deixis with gaze and gesture in 2010. He is now Akademischer Rat in the Artificial Intelligence Group at Bielefeld University, Germany. From 2006 to 2009 he was researcher in the PASION (Psychologically Augmented Social Interaction Over Networks) project funded by the EU. His current research areas are human-computer interaction, virtual agents, 3d gaze analysis, social networks and interactive visualization.