Visualizing Online Social Networks in the Context of Web 2.0

Adrian N. Buzgar and Sabin C. Buraga

***Abstract*—The following paper presents an approach to visualizing social networks in the context of Web 2.0 (social Web) applications. We propose a Web application architectural model that is suited to the task at hand. Also, several examples are presented, by integrating Flex free libraries used to enable (semantic) visualizations of social network data.**

Keywords—Web 2.0, online social networks, information visualization.

I. Introduction

IN a departure from the static model of the previous generation of online content, Web 2.0 is a decentralized social platform that offers rich interaction and information exchange between both users and applications. The term has grown to encompass many technologies and design principles that adhere to this trend, and this is in most respects synonymous with the social Web.

Some of the catchphrases related to this trend deserve mentioning. For instance, '*cooperate, don't control'* maintains the idea of the Web as a platform, since with the increasing availability of *open APIs* and *Web services*, reusing data and business logic from the *cloud* into *mashups* has favored integration between many popular applications.

Another such phrase is *'some rights reserved'*, which suggest the use of open standards and licenses, both of which encourage *remixability* and *hackability*, two new so-called buzzwords that come in contrast with the proprietary, monopolizing software policies of the business models prior to Web 2.0.

Certain aspects commonly associated to Web 2.0 have arguably been present even before the term was introduced. Also, the fact that the Web is meant as more of a social creation than a technical one was a central point stated by its founder, Sir Tim Berners-Lee [2]. Nonetheless, the extent to which individuals can connect and interact online has seen a positive change, and from this point of view, Web 2.0 is an important step forward.

The current generation of Web applications can be described as a sum of online social networks. This underlying structure of users related through common interests or other criteria is present in virtually all popular Web applications, regardless of their purpose.

Internally, most such applications offer simple mechanisms for viewing the connections between given users. For instance, LinkedIn [3], a well-known repository for online portfolios, lists contacts in an alphabetically-indexed list, with optional filters. Indirectly connected users are only shown as counted up to a predetermined limit of three hops. This is a justified design choice, as it reduces server-side complexity. However, this model, present in many similar applications, limits the way a user can navigate through the network of profiles to a hop-by-hop fashion which flattens and conceals the actual structure of the social network.

Our current research focuses on the development of a visual online social network, i.e. one which shows its entire structure dynamically to all users, aside from the usual functionality, which in this case will be externalized by means of *mashups* of other online services that are presumed already familiar to the vast majority of users. Our purpose is to add visual immersion to an online social network, and to facilitate a better navigation of the data provided by users.

The following sections propose an initial approach to visualizing the structure of an academic social network, i.e. a model of the Faculty of Computer Science.

To accomplish this goal, we chose to integrate in our application two Flex libraries able to provide visualizing techniques for our data set. More details are available in Section III.

The paper concludes with final remarks and further directions of research.

II. Related Work

At first glance, since the data structure of a social network is a non-directed graph, the problem of visualizing its contents is reduced to that of displaying a graph. However, depending on the purpose or aspect of interest of a given visualization, different layout algorithms need to be used to obtain the best perspective over a given data set. This issue has gained some interest over the past years, with a number of interesting results surfacing [4-5].

One such example is the Vizster application developed by Jeffrey Heer and Danah Boyd [4], which presents a choice of different available visualizations for data gathered from the

Friendster online community.

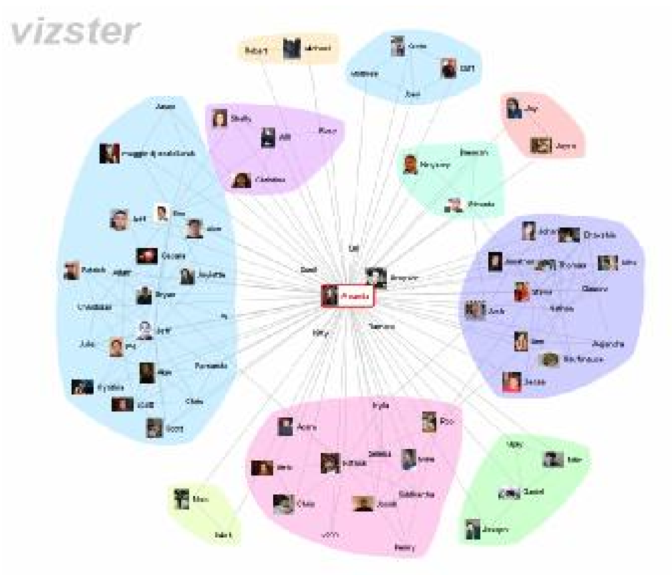

Fig. 1. Vizster application use case. A clustering algorithm progressively groups profiles related to the central selected user.

III. Towards a Visual Online Social Network

While the example given previously allows a user to much more thoroughly perceive and use the social data already available within the system, the obvious next step would be to also allow for real-time changes (e.g., users registering, connecting to other users or to interest groups, adding personal data) that would immediately be reflected in the interface.

In order for such functionality to be offered in a scalable Web application, a careful choice of technologies must be made. Commonly used development platforms like J2EE [6] tend to keep most if not all of the business logic on the server side. However, in this case most of the complexity of the application lies in the algorithms used for visualization, not in the usual CRUD operations. A *thick client*, which runs in the browser instead of the server machine, is a better choice. For this reason, we have chosen the Adobe Flex platform. Previous results [7] have convinced us that the flexibility it offers in terms of designing user interfaces by far outweighs the downside of having to install the Flash Player plug-in on the client machine.

In the current stage of our project, we have experimented with available solutions to visualizing data in Web 2.0 applications, in the form of two free libraries for the Flex platform. We will next present each of these.

A. *Prefuse – Flare* [8]

Flare [8] is an open-source ActionScript library for Flex which provides data management, visual encoding, animation, and interaction techniques. It is an actively developed conversion of Prefuse, a previous Java library.

One of the innovations of Flare is the use of elastic tension between notes of a given graph visualization, resulting in a *force-directed layout*. This allows a user to drag any node in a given graph and pull the rest of the structure after the selected node.

Overall, the number of available options for information visualization in Flare is impressive. For any general-purpose Flex application, this would improve the user interface layer greatly. However, in our scenario, the comprehensive process necessary to configure certain visualization after data-binding meant that Flare remained our second choice.

B. *KapLab Visualizer* [9]

Visualizer is a free (closed-source) library for visualizing and interacting with graphs. It offers a more limited array of layouts, but the process of data-binding and configuration was more straight-forward. In the next examples, after integrating this library in our solution, we have used a virtual academic social network in the form of an XML (Extensible Markup Language) document containing 40 students and 10 professors, and their profile data.

Although, as shown in Figure 2, the data set consists of a tree, in contrast with the more complex structure of an actual social network, which often contains local meshes or forests. Nonetheless, relationships between users can be kept in separate data structures (e.g., *friend-of*, *member-of*) which can be reduced to trees.

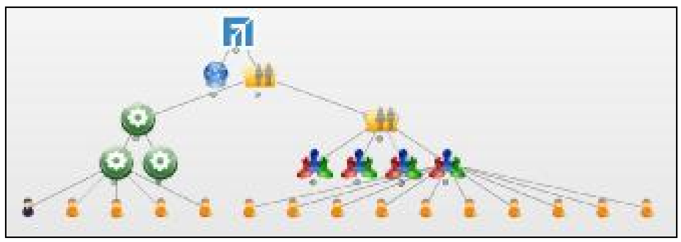

Fig. 2. Tree layout. Most of the structure was collapsed to keep the figure width within bounds.

A layout suited to visualizing the hierarchical structure of the data (i.e. departments, research groups etc.) is the *balloon layout*. This plots leaf nodes to form circles around their first ancestor. Note that the same property does not hold for higher levels of the tree.

In the case of a larger hierarchy, the process of successively expanding all nodes automatically zooms out to keep all *bubbles* within the viewable area. This raises the problem of visual scalability, a solution for which is the use of *semantic zooming* [10], which dynamically reduces detail levels (e.g., simple avatars instead of avatars with attached names, as seen above) to keep the entire visualization comprehensible.

An alternative to the *bubble layout* is the *organic layout* shown in Figure 4, which pushes leaf nodes towards the extremities of the visible area in order to disallow nodes from coming close to overlapping, again serving to improve the overall aspect.

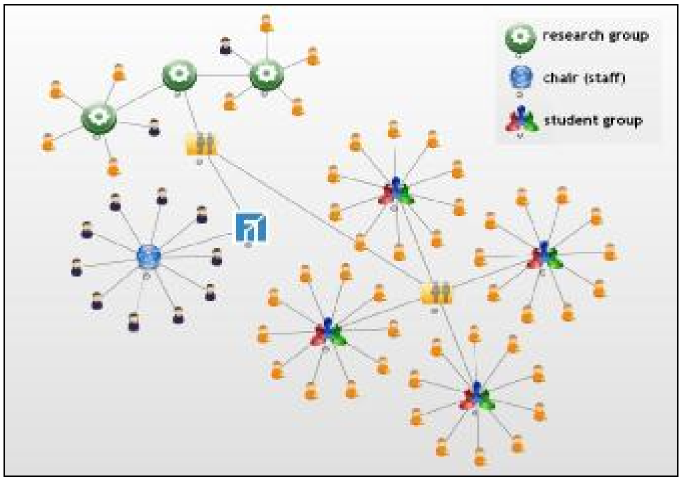

Fig. 3. Baloon layout. Research clusters have been moved to keep the zoom level from decreasing.

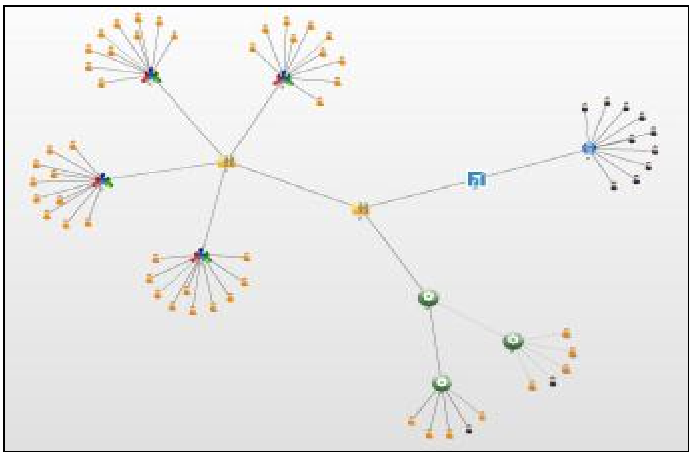

Fig. 4. Organic layout. Note that depending on the structure of the data-set, the root of the tree may not be shown centered (the FII – Faculty of Computer Science – logo in this case is the root).

Two forms of circular layouts are available, named *circular* and *radial* – see Figures 5 and 6. The difference between these approaches consists of the location in the visible space of the nodes above the tree's leaves. In the first example, all nodes are placed in a circle, which favors the display of one-to-many relationships (e.g., can clearly show all the users that share a certain feature with the currently selected user, by drawing lines to connect them). In the second example, child nodes are pushed towards the exterior of the circle, while parent nodes are kept inside the circle.

IV. Conclusion and Further Work

In this paper, an original study was presented in order to increase the level of awareness regarding modern visualizing techniques based on the Adobe Flex platform.

After certain details about related solutions, in Section III we gave different examples concerning available layout algorithms for social networking data sets. Our case study focused on visualizing a model of an academic community. As previously stated, an important facet is the management of large data spaces through the use of semantic zoom techniques. Therefore, another direction of interest is to test our approach for large communities of users, to study the scalability and performance of visualization algorithms.

The next step in this research work is to integrate real interaction mechanisms so as to add practical functionality to the visual user experience of this social Web application.

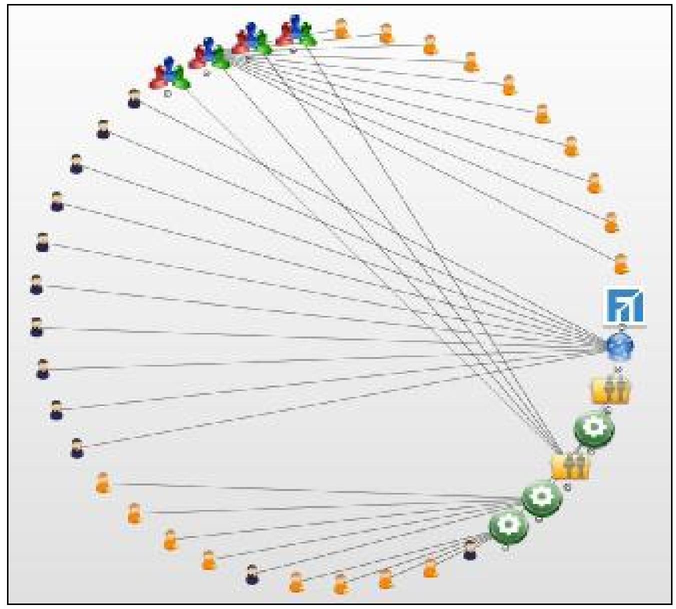

Fig. 5. Circular layout.

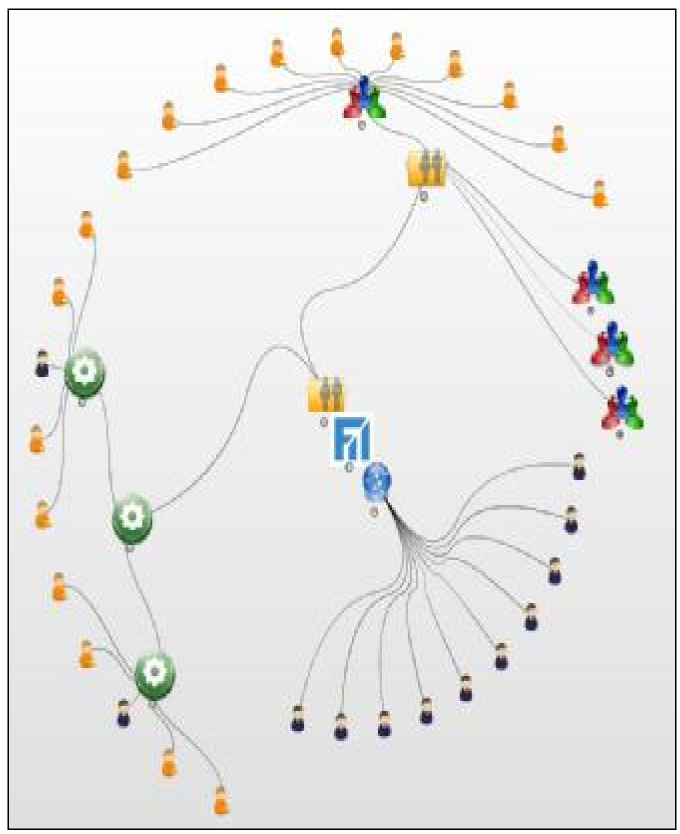

Fig. 6. Radial layout.

REFERENCES

[1] T. O'Reilly, "What Is Web 2.0", O'Reilly Media, 2005, Available: www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
[2] T. Berners-Lee, "The World Wide Web – Past, Present and Future", Japan Prize Commemorative Lecture, 2002, Available: http://tinyurl.com/5hvuaa
[3] Linkedin, Available: http://www.linkedin.com/.
[4] J. Heer, D. Boyd, "Vizster: Visualizing Online Social Networks", IEEE Symposium on Information Visualization, IEEE Press, 2005.
[5] J. W. Park, "Visualization of Online Social Networks Using a Blob", 2007, Available: http://www.gsd.harvard.edu/users/jpark/course/final_jpark.pdf
[6] Java Enterprise Edition, http://java.sun.com/javaee/
[7] A. Buzgar, S. Lazarovici, "MovR: Utilizabilitate Web 2.0", Proceedings of the 5th Romanian Computer-Human Interaction Conference (RoCHI), Matrix Rom, 2008.
[8] Flare Library, http://flare.prefuse.org.
[9] Kap Lab Visualizer, http://lab.kapit.fr/display/visualizer.
[10] B. Bederson, J. Hollan, "Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics", Proceedings of the ACM Symposium on User Interface Software and Technology – ACM UIST 1994, Marina del Rey, 1994, pp. 17-26.