

DATABASE

TRENDS AND APPLICATIONS

Solutions for the Information Project Team • www.dbta.com

Volume 22, Number 4 • April 2008

APPLICATIONS INSIGHT



GUY HARRISON

OpenSocial Aims to Open Up Social Networks

Social networking sites epitomize many of the characteristics of “Web 2.0.” As well as highlighting new technologies such as AJAX, social networking sites emphasize interactions between users and user-generated content, rather than centrally provided static content. It’s the user-generated content and relationships that define the value of any social networking site.

Given the value placed on the user-generated content, and given how rapidly social networks can gain (Facebook) or lose (Friendster) traction, it’s hardly surprising to find that these sites don’t make it easy to switch from one to another. The most common barrier is data lock-in; it is generally hard or impossible to get your personal data out of a social networking application. Indeed, you may be barred from the site if you attempt to get your data out! For this reason, social networking applications are sometimes referred to as “walled gardens” - they may be attractive and pleasant, but they are walled off from the rest of the Web.

At first glance, the introduction of Google’s OpenSocial API seems to promise a way out of the walled gardens. The OpenSocial API allows interoperability between social networks and for the development of applications that span multiple networks. Google has recently provided OpenSocial support on their own Orkut social network as have MySpace, Bebo, LinkedIn and other prominent social networks.

Facebook remains the obvious absen-

ter in the OpenSocial universe, which says a lot about the underlying motivations of the social networks that endorse it. One does not have to be too cynical to see OpenSocial adoption as a coalition approach to counter Facebook dominance rather than as an altruistic move designed to improve our social lives.

In reality, OpenSocial is more of a development platform than a portability platform. OpenSocial allows third-party developers to create applications that will work on top of any - or many - OpenSocial compliant sites. An application could allow, for example, members of LinkedIn, MySpace and Orkut to collaborate on creating an ultimate top 10 list of Desperate Housewife quotes. By allowing niche applications to simultaneously address the populations of multiple networks, the scope for these applications is increased.

Facebook recently released its own API that allows for similar applications to be deployed to the Facebook platform. These applications may run on some other sites - such as bebo.com - but would not have the widespread compatibility provided by OpenSocial.

So while OpenSocial shows some promise of integrating networks, it doesn’t specifically allow you to migrate across networks. There are emerging movements towards that goal, such as the data portability project (dataportability.org). However, it seems to me that until such time as legislation guarantees consumers the right to extract their personal data from any Web site, social net-

working sites will continue to pursue the walled garden approach.

OpenSocial and the Facebook API are examples of the emerging platforms for small-scale developers - “micro-ISVs” - in the Web 2.0 world. Together with Salesforce.com’s development platform initiative, and Amazon’s cloud computing environments, these offerings allow developers to deploy scalable Web 2.0 applications without having to provide the expensive and complex infrastructure that has until recently been required.

Another interesting aspect of OpenSocial is the Web services model. Like most distributed Web applications, HTTP and XML form the foundation of the Web Services provided. But standards-based Web Service protocols - SOAP, UDDI, WS-Security, etc. - are nowhere to be seen. Instead OpenSocial is based on simpler protocols such as ATOM and RSS. Indeed, standards-based Web service protocols seem to play little role in leading-edge Web 2.0 applications - the very sorts of applications for which they were originally conceived.

Guy Harrison is a chief architect for database solutions at Quest Software, and is a recognized expert with over 15 years of experience in application and database administration, development, performance tuning and project management. He is the author of “Oracle SQL High Performance Tuning” (Prentice Hall) and “MySQL Stored Procedure Programming” (O’Reilly), and is a regular speaker at trade shows and events.