Uniting Online Social Networks with Places and Things

Michael Blackstock
Rodger Lea
Media and Graphics Interdisciplinary Centre
University of British Columbia
Vancouver, Canada
{mblackst, rlea}@magic.ubc.ca

Adrian Friday
School of Computing and Communications
InfoLab 21
Lancaster University
Lancaster, UK
adrian@comp.lancs.ac.uk

ABSTRACT

As the Web of Things (WoT) broadens real world interaction via the internet, there is an increasing need for a user centric model for managing and interacting with real world objects. We believe that online social networks can provide that capability and can enhance existing and future WoT platforms leading to a *Social WoT*. As both social overlays and user interface containers, online social networks (OSNs) will play a significant role in the evolution of the web of things. As user interface containers and social overlays, they can be used by end users and applications as an on-line entry point for interacting with things, both receiving updates from sensors and controlling things. Conversely, access to user identity and profile information, content and social graphs can be useful in physical social settings like cafés. In this paper we describe some of the key features of social networks used by existing social WoT systems. We follow this with a discussion of open research questions related to integration of OSNs and how OSNs may evolve to be more suitable for integration with places and things. Several ongoing projects in our lab leverage OSNs to connect places and things to online communities.

Keywords

Social networks, internet of things, web of things

1. INTRODUCTION

The Internet of Things, initially focused on identifying objects with RFID tags and wireless sensor networks, has evolved into a broader vision of integrating the digital and physical world akin in many ways to the vision for pervasive and ubiquitous computing. Like the pioneering work in wide area and pervasive systems such as CoolTown [14,19], ActiveCampus [7] and the GUIDE Project [6] and more recent work in ubiquitous systems [16] the IoT is beginning to embrace web standards [11,17], moving from incompatible islands of connected things to global connectivity over the web. This global network has been referred to as the "web of things" [10].

Embracing the web architecture to give the physical world an online presence is not a new idea. The Cooltown project leveraged the simple and effective model of the web by providing a software layer called a "web presence" to integrate the physical environment with the web. In Cooltown, a web presence was software running on a server that provided a web user interface to a person, place or thing. Unlike other ubiquitous computing systems based on web services (WS-*) [3,7,13], and more recent IoT standards efforts such as DPWS [17] Cooltown not only embraced the HTTP protocol but the spirit of the web as a distributed system. Using Web protocols for naming objects (URLs), representing objects (HTML) and linking related people, places and things to each other using hyperlinks similar to how modern RESTful APIs and architectures expose application resources to application developers.

Recent academic and commercial efforts under the label the "Web of Things" has aimed to create an open Web 2.0-style infrastructure to allow web developers to create innovative IoT products and services [2,10,22]. Embracing the web affords the opportunistic use of connected things such as sensors, actuators and other digitally connected physical objects to create *mashups* where users can interact and collaborate with each other and the physical world [8,12]. Users and mashups can be consumers of not only user-generated data but also machine-generated data.

The emergence of online social network (OSN) web sites and systems such as Twitter, Open Social and Facebook are often considered another key component of Web 2.0[1]. OSNs allow people to build and maintain connections with family, friends and coworkers. On these systems, users establish profiles to identify themselves, establish relationships with other users, and share content with others such as photos, videos, messages and status updates. Just as ubiquitous systems and the Web of Things have recently embraced web protocols and RESTful APIs to identify devices and establish physical world interaction for physical mashups, social networks have exposed their data models to applications to build social applications and link web sites to online communities [15].

More recently commercial applications and systems are beginning to link social networks to the real world using mobile phones. Applications like Foursquare and Facebook places are using GPS-equipped phones to allow users to check-in to real world locations to find friends and earn rewards. Recent efforts leverage online social networks to share things on the web [9] using authentication mechanisms like Open Authentication (OAuth)[2] to support participative sensing mashups (e.g. [9,21]). Just as Cooltown and others web-enabled the physical world, social networks are bringing online communities into our everyday lives as well as connecting the physical world to online communities [9,20].

We believe online social networks can be leveraged to enable a *social* WoT. They can be used as an entry point for users and applications to interact with things on the web. For example, Facebook can be used as the entry point for sustainability applications that connect to sensor data sources and host "sustainability challenge games" for mobile and desktop users. In

[1] http://en.wikipedia.org/wiki/Web_2.0. Accessed: 01-29-2011

[2] http://oauth.net/. Accessed: 01-29-2011

some scenarios it is useful to access social graphs from the real world. This could be used by public displays in social settings to indicate social connections between people in places like coffee shops and bars.

In this paper we describe some of the key features of social networks and some existing social WoT systems. We follow this with a discussion of open research questions related to integration of OSNs and how OSNs may evolve to be more suitable for integration with people, places and things. We then describe several ongoing projects in our lab that leverage OSNs to connect places and things to online communities.

2. ONLINE SOCIAL NETWORKS

To better understand how online social networks can be integrated with the physical world we need to understand the services provided by current social network platforms. The primary services are as follows:

- Identity and authorization services
- APIs to access and manipulate the social network graph and publish and receive updates
- Container facilities for hosting third party applications.

All social networks store identity information about their users. This typically includes profile information, privacy settings, and connections with family, friends, colleagues or followers. OSNs store content such as messages to each other, updates (e.g. to their "wall" or "tweets"), photos, videos, events, fan pages and other objects. All social networks support authentication to prove user identity. In some cases third party sites can use an OSN authentication service as an identity provider.

OSN platforms also provide authorization services to control access to user data from third party web sites that access the OSN on behalf of end users. Most systems use the OAuth standard to allow third party sites as well as mobile and desktop applications to access OSNs on behalf of registered users. To support different types of applications, OAuth supports various client (application) profiles. For web applications, a third party site obtains authorization from the user, for example by redirecting the user's browser to an authorization page. If the user authorizes the site, the browser is redirected back to the third party site with an access grant that they can exchange for an access token.

OSNs support APIs for third party applications to extend the system by accessing and manipulating social network resources. Some systems use proprietary APIs while others aim to provide open APIs with a goal of supporting cross social-network compatibility. Currently social networks allow users to update their status, to 'like' resources on the web, create new connections and interact with on line social network applications, largely one-way interactions. An important API supported by all OSNs is the ability to publish or receive updates to the user's activity feed or stream.

OSNs like Orkut or Facebook can act as containers for third party applications. When an OSN acts as a container for applications, the application becomes part of the OSN user interface and overall social experience. The application user interface hosted on an external web server or implemented using markup and JavaScript is loaded in the UI of the OSN by either referencing an iFrame as illustrated in Figure 1, or it is hosted by the container, and sent to the browser on demand.

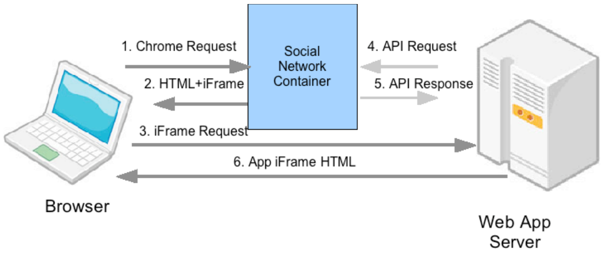


Figure 1. Third party application contained in OSN hosted on another server using an iFrame.

In the next section we describe several popular OSN APIs highlighting their unique approach to providing social network services to applications.

2.1 Facebook

Facebook is the most popular OSN platform today with over 500 million users. Facebook is often used by third party sites as an identity provider. By linking their identity to Facebook, a user need not register on each site individually and avoids the need to create yet another user id and password.

For authentication and authorization of third party web sites and applications, Facebook supports OAuth 2.0[3]. OAuth 2.0 is currently an "Internet Draft": created in an attempt to simplify the original OAuth 1.0 and 1.0a specifications[4] by leveraging standard TLS and SSL encryption to exchange credentials and access tokens between service providers and their clients. In addition to simplifying authentication for developers, OAuth 2.0 includes support for various flows, for web servers, native desktop and mobile apps with a UI, as well as devices where user authentication is not practical or possible such as PVRs set top boxes, home automation equipment, sensors and actuators.

Facebook supports two APIs for developers: a legacy RESTful API as well as the new Graph API[5], recommended for all new applications. The Graph API is interesting in that it provides a very simple and consistent REST API for all Facebook resources. All objects such as users, photos, events, and Facebook pages are assigned a unique id by the system, either the user name or a unique id. For example to get information about Brett Taylor:

```
https://graph.facebook.com/btaylor?access_token=…
```

To get information about the current user, you use the "me" keyword. In both cases graph API returns a Javascript Object Notation (JSON) representation of the object. For example:

```
https://graph.facebook.com/me?access_token=22…
{
    "id": "69..........",
    "name": "John Doe",
    "first_name": "John",
    "last_name": "Doe",
    "gender": "male",
    "locale": "en_US"
}
```

The URL can be simply extended with a keyword to request the corresponding object properties or relationships. For example to retrieve a user's public profile photo:

```
http://graph.facebook.com/DoloresPark/picture
```

[3] http://tools.ietf.org/html/draft-ietf-oauth-v2. Accessed: 01-29-2011

[4] http://tools.ietf.org/html/rfc5849. Accessed: 01-29-2011

[5] http://developers.facebook.com/docs/reference/api. Accessed: 01-29-2011

Similarly, to list all of the Facebook pages the current user likes:

`https://graph.facebook.com/me/likes?access_token=…`

To publish new data, a client POSTs HTTP requests containing name/value pairs to the appropriate URLs. To delete data, the client uses an HTTP DELETE request[6].

Facebook's Graph API now supports notifications to support detecting some changes in the graph. Applications subscribe to certain resources (currently users, pages, and application permissions) to be notified of changes to the properties of these objects. Applications can register interest by adding subscriptions to the application object. A subscription contains the URL for the callback, the type of object to monitor (user, permissions, or page), and the properties or connections of the object to monitor.

Facebook supports several types of applications: desktop and mobile applications, and third party web sites or canvas applications that are contained in the Facebook UI. Canvas applications provide dynamic content associated with users that have installed the application, or within a Facebook page container that hosts the application.

Using canvas applications, the user experience of Facebook is extended to include functionality that may not have been anticipated by the Facebook designers while maintaining a consistent user experience. These applications effectively extend the reach of Facebook to external resources and services like social applications, games, and utilities while maintaining the overall Facebook user experience.

2.2 Open Social

Open Social is a set of APIs for building social applications on the web. The Open Social effort was started by Google, MySpace and others and is now supported by LinkedIn, Yahoo Friendster and others.

Unlike the Facebook Graph API, Open Social[7] is an open specification with reference implementations available[8] allowing anyone to create not only applications but also their own social network containers. Using these open APIs social application developers can write applications that "live" within any Open Social container; applications created for Orkut can also run on others like MySpace with little or no modification.

The specification supports several compliance levels that define a set of capabilities. This allows Open Social containers to supply only the capabilities that their users need while maintaining compatibility with others at the same compliance level. This specification allows remote clients to access server side data using REST and RPC protocols, OAuth 1.0 security, perform content upload, identify resources, perform service discovery and call services. There is nothing specific to social networks in this specification. Building on this specification, the Social API Server specification defines the services that a social site must provide for remote applications such as people, groups, activities, application data, albums, content and others.

To be a host for applications that are integrated into a container user interface - similar to Facebook canvas applications, a system must comply with the Core Gadget Container specification. When a system complies with this specification it is able to render *gadgets* that are integrated into its user interface and supports a JavaScript library for building applications embedded in containers. To support social networking features in these hosted applications the system must comply with the Social Gadget Specification that supplies a social network object model for gadget applications using a JavaScript API.

When a system complies with all specifications, it is a full Open Social container, able to handle remote and embedded gadgets, or a combination of both and provides a full set of APIs including a model for social networking that includes people, groups, friend connections and content.

2.3 Twitter

Twitter has relatively few features when compared to Facebook or Open Social. The purpose of twitter is to connect users who send short messages called *tweets* (up to 140 characters) to *followers* who are interested in reading these messages in reverse chronological order. Like Open Social, Twitter uses OAuth for third party applications to make use of twitter services on user's behalf. Unlike Facebook and some Open Social systems, twitter does not act as a container to host applications within its user interface.

Twitter has three APIs - two REST APIs and a long polling API. One is for interacting with the system as a user, and the other for searching for tweets. A third API provides streaming services for delivering real time feeds using HTTP long poll[9].

2.4 Discussion

Social networks provide significant functionality that may be leveraged when integrating the physical world with the web. First, they include mechanisms for identifying users and storing information about those users that is easily accessed by applications. They can provide access to users' trusted connections, for sharing and rating pages, links photos and other content. Both Facebook and Open Social offer user interface container frameworks. These frameworks provide a way for applications to extend the social network user interface with plug in applications using gadgets or iFrames. This capability allows these applications to extend the social network user experience in ways that may not have been anticipated. Finally, OSNs supply a conceptually uniform namespace and associated APIs for interacting with people, places, pages, and content on the web.

In addition to this core functionality, we can also leverage several proven technical approaches in the area of security and interaction. Using OAuth, OSNs can securely authenticate third party applications (web, mobile, desktop and device-hosted) to access user information such as profiles and social connections. OSNs expose easy to use RESTful interfaces built on top of HTTP for easy web application integration and development (mashups). Most, if not all commercial and research systems follow this lead. As a result of these technologies, social networks, and especially twitter provide a way to publish and subscribe to soft real time updates or feeds about users and their activities.

[6] if the DELETE method is not supported by a client, it can use an HTTP POST and append method=delete to the URL.

[7] http://www.opensocial.org/specs. Accessed: 01-29-2011

[8] http://shindig.apache.org/. Accessed: 01-29-2011

[9] http://dev.twitter.com/doc. Accessed: 01-29-2011

3. OSNs AND THE WEB OF THINGS

Several commercial products and recent research efforts have begun to integrate social networks with the physical world. FourSquare[10], for example, allows users to create and check-in to real world places using their GPS-enabled smart phones. Based on how many times they check-in, users are given awards and product discounts. Google Latitude[11] allows users to track the location of their friends. Twitter and Buzz associate short messages with locations. The Pachube system [18] allows users to share, store and find sensor feeds on the web.

Research projects such as the Social Access Controller [9] have used social networks to enable the discovery and sharing of web-enabled things. While SAC connects to a number of social networks to extract the users and groups for sharing things, the system does not "live" within one or more social networks; the user interacts with the SAC user interface, not the OSN interface to connect things with friends. Like the OSN's mentioned, the SENSE-SATION project [21] leverages OAuth and a RESTful interface to access sensors on mobile phones for building participative sensing applications where end users are involved. While SENSE-SATION supports "social context-aware" applications, it does not attempt to integrate its facilities for exposing phone sensors with social networks directly, except perhaps as another client of the system. The SenseShare system [20] proposed to use Facebook as the front end to their sensor sharing system. While we believe that OSN user interfaces should be used, it is important to be open to multiple OSNs. Furthermore, it is not clear that exposing the WoT as just another Facebook application is the best approach. The Product Empire [4] game links things to people by taking a similar approach to FourSquare, Similar to how people "check in" with Foursquare, users enter information about products they scan with barcodes. Over time the system builds up a "crowd sourced" up-to-date product repository. The House that Twitters [1] sends various updates to a twitter feed from sensors in the home allowing users of Twitter to monitor the home remotely.

These systems leverage several aspects of Social Networks and Web 2.0 technologies. They use or provide a RESTful, HTTP-based API to make it easy for external developers to access and manipulate places and things exposed by these platforms.

4. TOWARD STRONGER INTEGRATION

While the Internet of Things, and more specifically the Web of Things has a history of leveraging Web 2.0 functionality such as RESTful APIs and OAuth, they have only begun to leverage existing social network systems and standards. Commercial systems like Foursquare, Facebook Places, Google Latitude and Yelp have begun to bring location and places into OSN, but so far, making objects or things first class citizens of our on-line social world is still on the horizon. Moreover, while we can often find information about the places around us, the interaction points for OSNs are currently limited to desktop and mobile computing devices. Essentially our social networks are not yet fully integrated into our real world environment.

A stronger convergence of the real world with OSNs will enable new applications and opportunities for social interaction between places and their owners, end users and the devices they control and have access to. For this to occur, stronger integration and continued convergence of the real world into OSNs is necessary.

However, to fully unify these worlds, there are several challenges that need to be addressed. In particular, while places and user location have begun to manifest themselves within on line social networks, is not clear how *things* or *objects* should manifest themselves in the on-line world. Conversely, it is not clear how social networks can or should manifest themselves in the real world. To begin to address these issues we need to consider three important research challenges.

Two-way interaction between OSNs and the WoT. First we need to look at how things will interact with the online social world. How will a broad set of devices and objects ranging from passive objects like books or products on a shelf to sensors, actuators and home appliances both identify themselves and interact with OSNs in a two-way exchange?

Extending OSN APIs and Models for the WoT. Second we need to consider how social network applications and the social network itself manifests things and objects in their APIs and data models to enable strong integration.

Real World User Interface Issues. Finally, and perhaps most importantly are user interface considerations ranging from how users interact on line with social networks that include places and things to how the real world exposes the on line social connections. How will the user experience of OSN containers need to be extended or evolve to support things and objects? How can real world objects expose social networks to end users in a way that maintains privacy while encouraging social interaction?

4.1 Two-way OSN-to-WoT Interaction

The largely one-way interaction between users and OSNs needs to evolve to include two-way interaction to receive updates and control things. Since many things are mobile, either carried with users or moved from place to place, how can we make on line relationships between users, places and things as dynamic as they are in the real world? Interaction should including the discovery of new things, and the establishment of relationships between things, users, communities, and places that mirror the real world where useful or desired.

To support two-way interaction we must make sure that data integrity and timeliness is maintained. The state of a thing and/or its relationship to other social network resources should be consistent for all users interested in that object. When a user controls an object, the object state should change in a timely manner, allowing applications such as remote control and real world automation.

How should smart objects such as appliances, sensors, controllers and gateways connect to OSNs and how should they authenticate themselves? Should each object or gateway be considered just another OSN application? If so, then is OAuth suitable? In some cases, there is no user interface on a device to authenticate it with the OSN. In others it may not be practical for users to authenticate everything they own - in their home for example. Device gateways may be suitable in these private environments to discover and authenticate and connect objects to OSNs on their owners' behalf. The OAuth 2.0 assertion flow or device profile may be used to allow users to authenticate things like set top boxes to access their OSNs using a browser on a mobile phone or laptop for example.

[10] http://foursquare.com/. Accessed: 01-29-2011

[11] https://www.google.com/latitude. Accessed: 01-29-2011

If the real world is updating the social network, and social network applications and users are interacting with the real world, how can we ensure data integrity and timeliness - ensuring the state of an object is reflected in a reasonable time in the network and is the same across all sites that have an interest in the thing? We can expect to leverage real time messaging and protocols such as that used by twitter and instant messaging systems [5]. Perhaps applications and the social network system itself on behalf of interested users can subscribe to certain relationships or thing updates.

4.2 Extending OSN APIs and Models

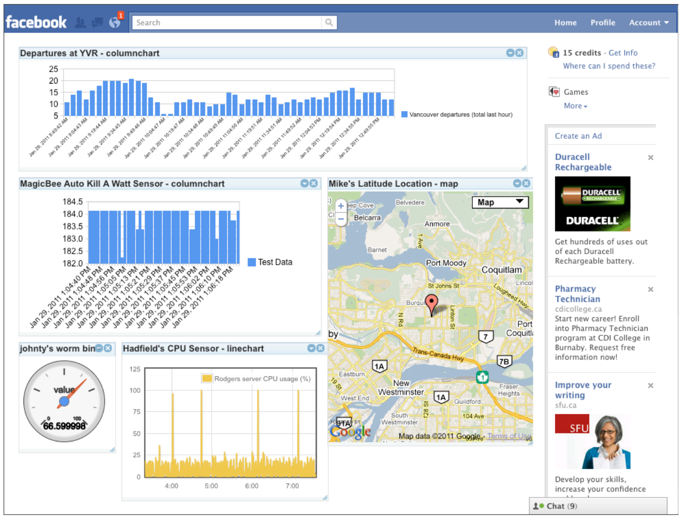
Today social network service and data models support not only users and social relationships, but also communities, content and web pages. For greater convergence with the real world their models and APIs should be extended to include physical places, and things as first class entities and appropriate inter-relationships. This will allow OSNs to become a 'social-overlay' for the Web of Things, enabling users to share places and things that they own, and discover things in the physical world that they can interact with.

In Facebook for example, extensions to the Graph API may be required. Open social will likely need additional services for object discovery, connections and two-way interaction. As Twitter is appropriated for sensing and actuator application, extensions may be required to the publish subscribe API to handle multiple twitter feeds per user, associate feeds with places and things, or change follower relationships based on location.

A challenge here will be to support appropriate location models that allow a high degree of flexibility for users, but still allow applications and mashups to do useful and interesting things with as little work as possible. In some cases, applications will need GPS coordinates, in others they may want to use place names and relationships for example. We should allow users to establish new, meaningful places and assign them names and meaning that make sense to the communities they belong to.

Social networks save profile information associated with each user. What profile information will they need to store about things? We may need to define ways to support different profiles depending on their capabilities and use cases. In some cases, we are only interested in the existence of a thing, and information about it. For smart devices like sensors, actuators and home appliances, the data generated and commands supported by one varies widely from one class to another. Like places, things can take on different meanings depending on the user and community that they belong to. Should we allow users to create a "folksonomy" of things and use crowd sourcing techniques (e.g. [4]) to associate various information and semantics to objects? If we do, how can we ensure enough interoperability to build useful applications?

One approach may be to extend existing APIs such as the Facebook graph API or Open Social APIs. To a limited extent the Facebook Open Graph Protocol[12] allows you to add your own pages - these can be used to represent places and things[13]. Using this hook, things, or groups of things can be given a social network presence. The type of thing the page represents is included in meta tags on the page. To date, the only relationship supported is "like" between these page and their users. In the future Open Graph could evolve to include metadata and additional relationships between thing pages, and place pages that are maintained by applications or end users.



Get hundreds of uses out of each Duracell Rechargeable battery.

Start new career! Enroll into Pharmacy Technician program at CDI College in Burnaby. Request free information now!

Develop your skills, increase your confidence

Figure 2. Sensor dashboard hosted in Facebook application.

Open social does allow container providers to extend their social model in a similar manner. Since Open Social depends on establishing a consistent social model for application interoperability, OS containers will have to agree on these extensions for applications to adopt them widely.

4.3 Real World User Interaction

Finally, and perhaps most importantly we need a paradigm to allow OSN users to seamlessly interact with places and things in the real world. We also need meaningful ways to present social network graphs in physical locations in a useful and meaningful manner so that people can leverage and extend their social networks by interacting in the real world.

To do this, OSN containers should evolve their roles as user interface containers to include hooks to extend their interfaces for interacting with places and things. Currently Open Social applications can extend the user profile interfaces, and create applications that can be installed by multiple users, in some cases, creating a community around these applications. Facebook allows canvas applications contained in an iFrame to be installed in application containers and Facebook pages. A page could be appropriated to represent a place, thing, or group of things; this would be an obvious place to support two-way interaction. An example of this is shown in Figure 2 where our web based sensor platform dashboard has been installed as a Facebook application.

Like the CoolTown web presence which offered link/relationships to related places and things, container interfaces for OSNs may evolve to include links that represent not only friends, and "like" relationships, but relationships like "contained-in", "owns", "next-to" that correspond to location and user-to-thing relationships that updated as they happen. Just as the API and data models for OSNs act as a social API overlay for the web of things, OSN containers can act as a user interface overlay for WoT gateways and proxies managing places and things.

To present social networks information to users in the real world is even more challenging. While it is possible to create a bridge

[12] http://developers.facebook.com/docs/opengraph. Accessed: 01-29-2011

between OSN graphs and situated sensors, actuators and displays, how will such installations be used in a meaningful way for the users in these places while maintaining autonomy and privacy?

5. CURRENT WORK

To address these challenges we are beginning to explore ways to integrate social networks with the real world.

First we are leveraging existing social network APIs and facilities as application containers to allow users to visualize and control devices in the real world. Our IoT portal [2] allows users to quickly connect sensors and actuators to the system, and then create dashboard visualizations as Facebook applications. These applications allow OSN users to share and monitor sensed data and control actuators in the real world.

Secondly we are developing a system that is both a social network container application and a container for other applications. This system will allow us to develop a single gateway for the online presence of things that can be installed into a number of social networks – both Facebook and Open Social containers for instance. Using this system we are creating a sustainability application that will host multiple sensor data sources and social "sustainability challenges" for OSN mobile and desktop users.

Finally, in our Interactive Community Displays project we are observing people in coffee shops, bars and study lounges within our university to uncover how members of the community located there will use public displays and how such displays can augment and improve social capital. In one scenario, Bluetooth sensors in a location detect the presence and identity of users nearby. Using this information and their OSN relationships, information visualization is updated to show the relationship between those users on a public display. This allows people to see if there are people they know nearby and how they are linked to others that may or may not be a part of their social connections. So far we have found that these displays served as accessories and initiators of discussion and social interaction.

6. CONCLUSIONS

In a similar manner to how the WoT has leveraged the web and pioneering work in ubiquitous computing, we believe online social networks can be leveraged and extended to enable a social WoT. This will allow the WoT to make use of the social network graph as well as OSN platform services such as authentication, user interface containers, and consistent APIs.

In this manner social networks should continue to extend their reach into to the real world just as the Cooltown and the WoT extends the web itself to physical places and things. Social networks are already adding places to their APIs data models and user interfaces; the next step is to add things and physical relationships to these models. To do this, OSNs need to evolve to include stronger two-way interaction with things, extend their data models and APIs, and the user experience to bring together our on line social presence with the real world.

7. REFERENCES

[1] BBC Things that tweet. *http://www.bbc.co.uk/blogs/technology/2009/06/things_that_tweet.html*. Accessed: 02-02-2011.

[2] Blackstock, M. et al. 2010. MAGIC Broker 2: An open and extensible platform for the Internet of Things. *Internet of Things (IoT) 2010* (Tokyo, Japan, 2010)

[3] Blackstock, M. et al. 2008. Evaluation and Analysis of a Common Model for Ubiquitous Systems Interoperability. *Pervasive Computing*. 180-196.

[4] Budde, A. and Michahelles, F. 2010. Product Empire — Serious play with barcodes. *Internet of Things (IoT) 2010*

[5] Choi, J. and Yoo, C. 2008. Connect with things through instant messaging. *Internet of Things (IoT) 2008*

[6] Davies, N. et al. 1999. `Caches in the air': disseminating tourist information in the GUIDE system. *Mobile Computing Systems and Applications, 1999.* (1999), 11-19.

[7] Griswold, W.G. et al. 2004. ActiveCampus: Experiments in Community-Oriented Ubiquitous Computing. *Computer*. 37, 10 (2004), 73-81.

[8] Guinard, D. 2010. Towards opportunistic applications in a Web of Things. *Web of Things Workshop at IEEE PERCOM 2010*, (Mannheim, Germany, 2010), 863-864.

[9] Guinard, D. et al. 2010. Sharing using social networks in a composable *Web of Things Workshop at IEEE PERCOM 2010*, (Mannheim, Germany, 2010), 702-707.

[10] Guinard, D. et al. 2010. A resource oriented architecture for the Web of Things. *Internet of Things (IoT) 2010* (Tokyo, Japan, 2010)

[11] Guinard, D. et al. 2009. Discovery and On-demand Provisioning of Real-World Web Services. *IEEE Web Services* 2009, 583-590.

[12] Hartmann, B. et al. 2008. Hacking, Mashing, Gluing: Understanding Opportunistic Design. *IEEE Pervasive Computing*. 7, 3 (2008), 46-54.

[13] Issarny, V. et al. 2005. Developing Ambient Intelligence Systems: A Solution based on Web Services. *Automated Software Eng.* 12, 1 (2005), 101-137.

[14] Kindberg, T. et al. 2002. People, places, things: web presence for the real world. *Mob. Netw. Appl.* 7, 5 (2002), 365-376.

[15] Ko, M.N. et al. 2010. Social-Networks Connect Services. *Computer*. 43, (Aug. 2010), 37–43.

[16] Liu, Y. and Connelly, K. 2008. Realizing an Open Ubiquitous Environment in a RESTful Way. *IEEE Web Services 2008* (Washington, DC, USA, 2008), 96–103.

[17] OASIS Devices Profile for Web Services (DPWS). *http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01*. Accessed: 01-29-2011.

[18] Pachube - data infrastructure for the Internet of Things. *http://www.pachube.com/*. Accessed: 02-02-2011.

[19] Pradhan, S. et al. 2001. Websigns: hyperlinking physical locations to the Web. *Computer*. 34, 8 (2001), 42-48.

[20] Schmid, T. and Srivastava, M.B. 2007. Exploiting Social Networks for Sensor Data Sharing with SenseShare.

[21] Shirazi, A. et al. 2010. SENSE-SATION: An extensible platform for integration of phones into the Web. *Internet of Things (IoT) 2010* (Tokyo, Japan, 2010)

[22] Stirbu, V. 2008. Towards a RESTful Plug and Play Experience in the Web of Things. *IEEE Semantic Computing* (2008), 512-517.