IM-2.0
Fusion of Web 2.0, Semantic Web and Instant Messaging

Mark Williamson

Department of Computing, Maths & Physics,
Waterford Institute of Technology,
Cork Road, Waterford, Ireland
mwilliamson@tssg.org
http://www.wit.ie/

Abstract. Many internet users' have multiple Web 2.0 accounts detailing their online status with little linkage between the varying sites.
IM, Web 2.0 and the Semantic Web are public technologies with features that allow users to create content, share content and communicate with one another. Combining the functionality and strengths of these technologies can only form a synergy that can benefit the end user, by promoting openness and collaboration in the world of internet communication.
This paper examines the aforementioned technologies and their similarities, proposing a new Java Service named *IM 2.0* that enables internet users to quickly and easily add their varying Web 2.0 and IM accounts, thus enabling the creation of a singular unique open life stream page which can be subscribed to by friends and interfaced with by software applications. As well as broadcasting Web 2.0 updates through the instantaneous communications conduit of IM.

Key words: Web 2.0, Instant Messaging, Semantic Web, Chatbot, Subpresence, RSS, RDF

1 Introduction

Instant Messaging (IM) and Web 2.0 technologies are becoming more and more popular with internet users and are continuing to grow. According to Compete [1], in January 2009 there were approximately 2.2 billion monthly visitors to the top fifteen social networking sites worldwide. Radacati estimate that there are 933 million Instant Messaging users today which is expected to grow to 1.5 Billion by 2010 [2]. The vast majority of Web 2.0 and IM technologies are public tools whereby users can communicate and share information. This shared information varies from an individual's whereabouts, current activities, thoughts and opinions to personal content that they have created such as videos and pictures.

The Semantic Web, which is considered an evolving extension of the World Wide Web, is used for collaboration and sharing of data across the internet. It is about using common formats to describe information and how it relates to other sources as well as real world objects.

There is a potential intersection between IM, Web 2.0 and the Semantic Web in functions offered such as communication amongst friends, relaying what a user is currently doing, sharing content and developing an online community of contacts. This paper will target these popular and expanding technologies attempting to fuse them into *IM-2.0*. This can be achieved by:

- Collecting context based updates of a user's recent Web 2.0 online activity;
- Generating a unique user RSS feed and semantic web RDF page available openly on the web for people to subscribe to and technology to interface with;
- Presenting Web 2.0 activity through IM custom presence messages, IM notifications and responses to IM requests.

2 State Of The Art

This section highlights the origins of each technology as well as discussing current research and trends developing in their respective areas, prior to providing an analysis and contrast of current aggregators of Web 2.0, IM and the Semantic Web.

2.1 Instant Messaging

IM grew from Unix utilities such as talk and zephyr and gained popularity in the mid-1990s after Mirabilis introduced ICQ, a free instant messaging utility. IM functionality includes the ability to instantly view a person's availability and subsequently initiate a real time communications chat session. The server manages all IM communication and services between clients. The integral services it offers are packet routing and user account management such as contact list storage [3]. This provides access to contacts from any device on any network infrastructure. Clients typically act as user agents displaying information to the end user in the form of sub-presence, and responding to user requests. Client accounts may also offer services autonomously, which is referred to as a chatbot or IM bot.

Chan *et al.* [4] discusses the ability of chatbots to engage in IM conversations which can act as a transport for value added information. A bot would parse the IM conversation looking for specially formatted commands and when recognised, executes a task with the results being sent back to the participant of the conversation. IM is predominately used between people who are known to each other such as friends and associates. This sort of communication is secure and

reliable as it provides peace of mind that the recipients are trusted contacts and that messages reach their intended destination.

It provides one's ad hoc set of contacts with awareness of the individual's online state, which in turn serves as an estimate of their availability for conversation. The type of information broadcasted through IM sub-presence encompasses these main categories [5]:

- Identification ("who am I")
- Information about self ("this is what I'm doing")
- Broadcast Information ("directing information to the community")

Several other categories including mood, comments, activities, location, messages, quotes and notices are used to group information broadcasted through sub-presence. The detailed amount of information that can be depicted through such a small piece of text about one's self is surprising and impressive. Many people show more than one textual element in a sub-presence message. This is referred to as sub-presence holding multiple entries [5]. An example of this would be the message "listening music - U2" where listening to music could be categorised as the activity and U2 the source of the activity.

There is an increasing movement towards interoperability and openness through the development of the XMPP (Extensible Messaging and Presence Protocol) protocol. GTalk, the IM client offered by Google, is XMPP compliant which is the protocol of focus during this project.

XMPP is an open technology for real-time communication, which powers a wide range of applications including instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data [6]. XMPP was invented by Jeremie Miller in 1998, refined in 2000 by the Jabber open-source community and formalized by the IETF in 2003, resulting in the publication of the XMPP protocol in 2004 [6]. XMPP has been very successful and has been proven to be secure, flexible, extensible and robust as there are millions of adopters making it a standard.

The XMPP community is very active and continues to define extensions that add functionality to the protocol through an open standards process maintained by the XMPP Standards Foundation.

2.2 Web 2.0

Web 2.0, a phrase coined by Dermot McCormack in 2003 [7], refers to a perceived second-generation of Web based communities and hosted services such as Social Networks including Bebo, MySpace, Facebook, Twitter as well as content management systems like YouTube, Flickr and Web logs such as blogger. Other examples are Wiki sites and location based services such as the interactive features available through Google Maps. O'Reilly Media titled a series of conferences around the phrase and it has since become widely adopted. Though the

term hints at or implies the idea of a new version of the Web, it does not refer to an update to the World Wide Web technical specifications, but to changes in the ways system developers have used the web platform. O'Reilly depicted four levels in the Web 2.0 sites hierarchy [8]:

- Level-3 is suggested to be the most Web 2.0 orientated and exists only on the internet. These types of services become more useful as the user base increases. Examples are eBay, Wikipedia and Skype;
- A level-2 application can exist offline but can gain many advantages by going online. Flickr is an example of this as an online community has access to user generated content which would not be possible if photographs were managed locally;
- A level-1 application can and does exist successfully offline but provides more collaboratively functionality online. An example of this is Google Docs;
- A level-0 application is an application that works exactly the same offline as it would online once access to all the required data and information is cached locally. Example applications include Google Maps.

Bart Decrem, a founder and former CEO of Flock has stated that he regards Web 1.0 an internet solely to provide information. Web 2.0 allows more than just information retrieval but encourages users to create added value content which can be shared seamlessly with other internet users.

Often Web 2.0 services providing this ability are referred to as RIA's (Rich Internet Applications), which are applications that look and feel like desktop applications but exist online.

One of the main driving forces of Web 2.0 is the advancement of technologies and development methods. One particular technology that has supported the emergence of Web 2.0 is asynchronous JavaScript and XML (AJAX) a term first coined by Jesse James Garett in 2005 [9]. AJAX used a group of technologies to bring dynamic web pages and dynamic loading of web pages to the internet. This essentially meant that internet users no longer had to wait long periods for a web page to refresh or reload after clicking an option as AJAX allowed small portions of the page to be updated when required. Technologies used in AJAX include Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), Document Object Model (DOM), Extensible Markup Language (XML), EXtensible Stylesheet Language Transformations (XSLT), XMLHttpRequest and Javascript [10]. XML is a general purpose specification for allowing a user to define custom extensible mark-up language to aid information systems in sharing structured data on the Internet. Really Simple Syndication (RSS) is a form of XML existing as a family of Web feed formats used to publish frequently updated works such as blog entries.

Application programming interfaces (APIs) provide a mechanism for programmers to make use of a set of modules without having access to the source code. APIs continue to help Web 2.0 services grow and expand while also facilitating the creation of new services through various mash-ups of existing data

and services. Many APIs take a RESTful approach when accessing information on the web. Representational state transfer (REST) is a lightweight or simplified programming model, which facilitates the creation of loosely coupled systems. REST is an approach that allows a client and server to exist in a stateless architecture, which provides a simple communications interface using XML and Hypertext Transfer Protocol (HTTP). Every resource is identified by a Uniform Resource Identifier (URI) and the use of HTTP lets you communicate with the web server using GET, POST, PUT and DELETE command requests [10].

2.3 Semantic Web

The Resource Description Framework (RDF) is a framework for representing information on the Web. It was designed by the World Wide Web Consortium (W3C) for the purpose of depicting metadata (data about data) in a uniform and standard data model (defining data objects and relationships among data objects, including how the data is represented and accessed).

Every RDF assertion is decomposed in three parts known as an RDF triple. The triple consists of a subject, a predicate and an object. The subject denotes the resource. In Semantic Web applications and popular applications of RDF like RSS and FOAF (Friend of a Friend) resources are represented by URIs that show where the data is located in the World Wide Web. Furthermore, the subject could describe information originating from any resource. The predicate is a binary relation; it describes traits or aspects of the resource and describes a relationship between the subject and the object. The object is either a URI identifying a resource or a literal value. Each triple can be seen as a labelled arc and joining these arcs builds a graph that describes the identified resources and their relations. For example, one way to represent the notion "The dog has a name of Jessie" in RDF is as a triple consisting of: a subject denoting "The dog", a predicate denoting "has a name of", and an object denoting "Jessie" [11]. RDF data is often persisted in relational database or native representations also called Triple stores or Quad stores if context (i.e. the named graph) is also persisted for each RDF triple.

OWL (Web Ontology Language) is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. The Semantic Web is a vision for the future of the Web, in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web. The Semantic Web will build on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data. The W3C-endorsed OWL specification includes the definition of three variants of OWL, with different levels of expressiveness.

– OWL Lite was originally intended to support those users primarily needing a classification hierarchy and simple constraints. For example, while it supports

cardinality constraints, it only permits cardinality values of 0 or 1. It was hoped that it would be simpler to provide tool support for OWL Lite than its more expressive relatives, allowing quick migration path for systems;

- OWL DL was designed to provide the maximum expressiveness possible while retaining computational completeness (all conclusions are guaranteed to be computed), decidability (all computations will finish in finite time), and the availability of practical reasoning algorithms;

- OWL Full is based on a different semantics from OWL Lite or OWL DL, and was designed to preserve some compatibility with RDF Schema. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right; this is not permitted in OWL DL [12].

One of the major goals of the semantic web is to bring knowledge representation capabilities to the web. The web in its current form as a web of documents is very different to what is envisaged as the web of data or the Semantic Web. A web of linked data allows us to combine data from different sources to create structures on the fly. Linked Data is published on the web in a style that emphasises data re-use and builds connections between related data and its sources [13].

The Semantic Web and Web 2.0 can potentially complement one another and will need each other to scale beyond their respective strongholds. For example, if normal Web 2.0 services such as blogs added a few extra semantic fields in order to describe the blog entry it would depict the data much more accurately, allowing the data to fit into current semantic ontologies much more closely. Providing this extra information and using Semantic technologies enables greater interoperability, control, correctness and consistency of the data that can be transferred over the web. The information can be saved in a standard database needing only a SPARQL Protocol and RDF Query Language (SPARQL) endpoint to hook up to it or alternatively export the data as RDF while exporting it as HTML or RSS. In RDF form a SPARQL query can be executed directly on the page [14].

Semantically-Interlinked Online Communities (SIOC) is advocated as the ontology to add semantic information to the current web, which brings various social applications together in the hope of forming a social semantic web. SIOC types and properties were designed to describe Web 2.0 data. The main terms used in SIOC are Post, User group, Item, User, Role, Site, Forum, Container and Space. Combining these terms to describe data could result in users creating content items that reside in containers on data spaces, which could be broken down further into; a user is a blogger who posted an item on a blog which is located on a blogger site. Bojars *et al.* [15] point out the difficulties in knowing what accounts a user holds on different web services as well as the process of trying to gather the content created. They promote that the vocabularies of FOAF and SIOC used in conjunction can be used to describe the services a user is registered on as well as the content created. SIOC and FOAF exporter

tools can be used to construct RDF describing a person's Web 2.0 services and content. This RDF then contains information from all the services targeted by the semantic exporter within a single location available in a machine readable manner. Semantic technologies can join data from different sources as well as point to additional content of a similar nature using the "rdfs:seeAlso property".

FOAF is described as being the most popular ontology used to represent information about an individual. It can be used to describe people, their relationships as well as their activity. FOAF has dedicated a number of properties to describe a profile belonging to a user by using tags such as family name, nickname and interests etc. The "knows" property is used to connect people and build a social network. As well as those descriptive properties there are also elements to describe online accounts such as web blogs etc. Furthermore, the FOAF "knows" property can be extended using the "Relationship" ontology detailing a relationship into groups of friendship, family and professional etc [16].

2.4 Web 2.0 Aggregators

Since this work commenced, a number of other services have launched trying to leverage and add Web 2.0 information to their own offerings. These types of services could be seen as Web 2.0 aggregators. Established companies such as Microsoft are now allowing their subscribers to add external web service updates to their own respective accounts. Furthermore, start up companies SocialURL [17], Power [18], SwitchAbit [19] and Gnip [20] are also trying to gain a niche in this consumer area.

Microsoft through MSN is offering its users a service dubbed "Web activities", which aggregates what you've been doing on other websites into your own MSN profile for your contacts to see. The service supports many other popular Web 2.0 sites including Facebook, Flickr, Wordepress, Twitter, Dailymotion, ilike, Last.fm as well as custom blogs in the format of RSS or ATOM. A user simply chooses what service he/she wishes to give MSN access to and then approximately every half an hour MSN will check the service to see what content has been created which in turn is published to their MSN profile.

SocialURL and Power attempt to bring all your social networking sites together under one site. It allows you to add your accounts from other sites such as Bebo, MySpace, Piczo, hi5, Cyworld, LinkedIn, Xuqa, Youtube, Blogger, Digg, Friendster, and Twitter etc. The media generated on the external sites is then collected and embedded in your SocialURL and Power profile respectively. SwitchAbit opt to take a different approach when compared to MSN Web activities or SocialURL. Instead of polling and gathering content from other sites to be shown in one place they route the content to multiple social services. SwitchAbit explain this functionality by comparing it to the "Cc:" feature used in email. The idea is to write content and publish to multiple services once. Gnip recognises that many companies are trying to aggregate and mash up content from multiple sites. It is argued that the task of polling this information is very difficult and resource hungry. Their solution is to perform the hard work for you by polling the services on your behalf and then alerting you of updates. A callback address

is setup and this is used to inform you of updates on sites Gnip are polling for you.

As can be seen there is many opportunities and directions that can be taken in this area. So far it seems that none of the main players in this area are either leveraging the viral and instantaneous communication aspects of Instant Messaging or the information representation of the Semantic Web. This paper argues that utilising these technologies will further enhance the user experience and information sharing of the collaborative internet.

3 Goals

3.1 Reflection On The State Of The Art Trends

Etzioni and Etzioni state that in order to form and sustain bonds, a community of connected individuals need what they call "interactive broadcasting". This is composed of two major elements [21]:

- The ability to broadcast messages to many people within the community simultaneously;
- The ability for those addressed by the message to provide feedback, not just to the message originator, but to other message recipients as well.

This was addressed by fusing information from various sites into a single informative page, which also links back to the original data natively supporting feedback through comments and ratings. As well as that, IM-2.0 has the ability of replying to incoming IM messages in situations where the individual is unable to do so, helping to sustain friendship bonds while promoting participation.

Supporting openness by providing RDF that encapsulates links to a person's online context while also providing the format to amalgamate these links with other RDF data available on the web or vice versa on the fly, takes a tiny step towards reaching the idea of the Semantic Web. The RDF is constructed utilising semantic ontology's such as SIOC, FOAF and Versitcard (VCard) but could be expanded to use many other ontology's that are designed to describe various other forms of information.

Earlier the idea of sub-presence containing multiple entries was mentioned. The project utilises this type of multiple entry formula for depicting Web 2.0 information in the IM world, by first recording what a Web 2.0 service is capable of doing. For example regarding Blogger, a person is only able to create a new blog entry, so IM-2.0 will depict this as Blogger - New Blog - Blog Entry. It is envisaged that this sort of formula will work for all Web 2.0 services, so that the formula will contain Service - Activity - Event, providing enough information for people to ascertain the context of the update. Further detailed online context can be requested from the IM-2.0 service by making requests with certain commands.

Chatbots have the ability to converse with people and provide information. IM-2.0 has turned a normal user account into a chatbot in particular situations. This could be referred to as a typr of hybrid mode or IM-2.0 mode the account assumes when active.

3.2 Research Question

The combination of the research and experimentation work that has been carried out thus far has proven that the aim of the project has been fulfilled. Below are the research questions that drove the direction that the research took.

- How can web 2.0 personal context information be structured for usage in an IM Domain?

Updates were read from Web 2.0 services using APIs and RSS which where then broken down into categories such as Title, Description, Date, URL, Web 2.0 Service, Type of Update etc., which seamlessly fitted into existing semantic ontology's that could depict updates from Web 2.0 services.

- How can Web 2.0 context information transparently be projected into an IM infrastructure while allowing a user to use their favoured IM client?

XMPP supports concurrent user account authentication and sessions, a user provides their account credentials to IM-2.0. This enables a user to be logged into their IM client while IM-2.0 maintains a session using the same account concurrently. Developed using XMPP extensions, resulted in both sessions being able to alter and update sub-presence and presence which is broadcasted to the account's contacts. In certain situations the IM-2.0 server will only receive IM messages. This allows the server to respond to certain messages, while broadcasting sub-presence to IM contacts. The server is also powered by semantic technologies allowing it to understand the users RDF page and send this information into an IM infrastructure.

3.3 Problem description - Reasoning for the IM-2.0 service

The aim of turning a normal XMPP GTalk IM account into a hybrid IM service, allowing the account to act as a normal human controlled account as well as a chatbot service is currently possible once a certain IM client or plug-in is used. Popular IM clients such as MSN Live, AIM, Pidgin (provides XMPP access) and Trillian Professional (provides XMPP access) offer users the ability to create plug-ins for their respective clients. These platforms can be used to create a plug-in to offer IM service functionality, thus extending the functionality of the client and IM account. The creation and installation process is too technical for the average IM user which IM-2.0 caters for.

Web 2.0 users register on each website separately and must then promote their pages and content to their audience in an ad hoc way. The most common mechanism is via initiated email messages from the Web 2.0 site. The immediate drawback to this is "spam" filters whereby the email may end up in the recipient's junk folder. Furthermore, these unique sites have little crossover or connectivity forcing people to have a scattered online context.

4 Scenarios For IM-2.0

The following scenarios describe the functionality of the IM-2.0 service from a user's perspective. The features explained will include the web services that are supported by the system, the IM protocol implementation and how semantic web technology is supported. It is assumed in the scenarios that an individual has already registered and added all the supported web services to their account.

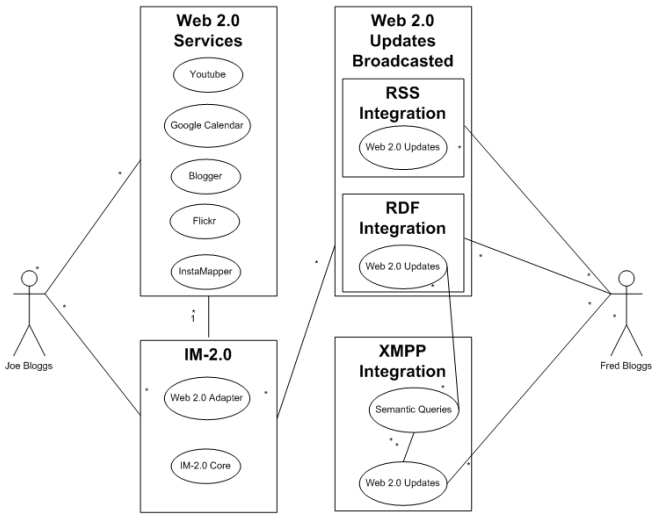

Fig. 1. Simple Overview - IM-2.0 reads data from Web 2.0 services and broadcasts this information through RSS, RDF. XMPP integration layer reads the user's RDF and sets the user's GTalk sub-presence to reflect their latest Web 2.0 event.

4.1 GTalk IM-2.0 Message & Presence Scenarios

A button on the IM-2.0 users account management page can turn on or off the ability that IM-2.0 has to access a GTalk accounts presence and messages. When the feature is turned on, any IM-2.0 update would also be propagated to a GTalk supporting client represented as a sub-presence message. The new event can then be seen by their IM contacts. Furthermore, when the GTalk user logs out of the client, all further messages sent to the account would be transported to the session maintained with IM-2.0, enabling automated responses to be generated based on the content of the message. The responses sent may contain simple help text or further detailed information regarding the initial sub-presence message alteration.

4.2 Google Calendar Event Scenario

- A new calendar event is created on the users Google Calendar account;
- The calendar entry will subsequently be consumed by IM-2.0 which will now have reference to this latest event;
- This data is exposed through RSS, RDF and IM.

4.3 Blogger Entry Scenario

This is a scenario developed to illustrate the functionality brought to IM-2.0 via the Blogger adapter.

- A person creates a new entry on their blog;
- As a consequence, IM-2.0 will eventually record the details of this new entry;
- IM-2.0 broadcasts the blog entry details via other outlets as seen in Figure 2 which is showing the published entry titled *Project Demo*.

Fig. 2. Sub-presence detailing the Web 2.0 update.

As mentioned in the GTalk scenario, the individual's IM contacts can now see this new sub-presence message and can optionally IM the user a special IM-2.0 command to get further information regarding the update as seen in Figure 3.

4.4 Flickr Picture Scenario

- A Flickr user uploads a picture;
- The image is detected by IM-2.0;
- Resulting in an RSS, RDF and optionally IM sub-presence update reflecting the contextual information of the aforementioned update.

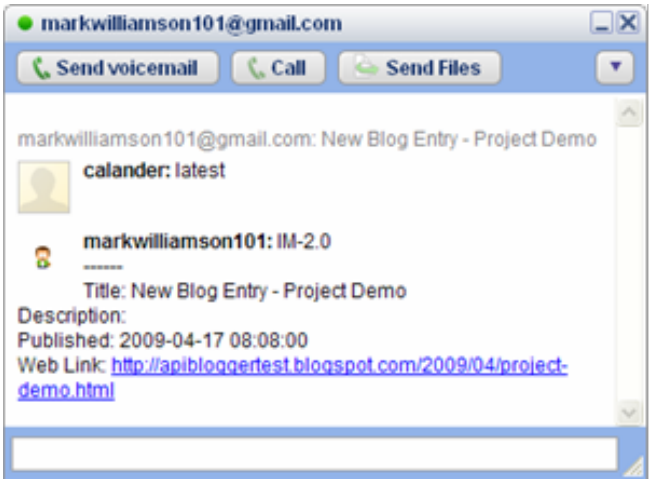

Fig. 3. IM message showing extra information

4.5 Youtube Scenarios

The following user stories carried out on Youtube would be digested by IM-2.0 and stored for broadcasting:

- A video is uploaded;
- A new favourite video is selected;
- The user subscribers to a new channel on Youtube;
- A new playlist is created and published.

4.6 Instamapper Location Scenarios

The following scenario was developed to illustrate the varying web services that could be supported IM-2.0. InstaMapper is a service that offers the ability to track a GPS-enabled cell phone online in real time. This is achieved by installing an InstaMapper application on a GPS enabled handset. The handset will then transmit the phones GPS co-ordinates to the specific InstaMapper account available online for viewing, as well as access through the InstaMapper API.

IM-2.0 would routinely check the account to get the latest known position of the user and save the co-ordinates. Subsequently when the user's IM-2.0 home page is accessed by anyone, the user's position would be shown on a Google Map, which can also be directly edited by the owner of the account who has the ability to drag the location marker around the map in order to correct or update their position Figure 4. This new position either retrieved from InstaMapper or manually set on the account holder's home page is broadcasted through the users IM sub-presence. This is achieved by performing a reverse geo-tagging on the GPS co-ordinates to acquire the nearest place name. An example of this is shown in Figure 5.

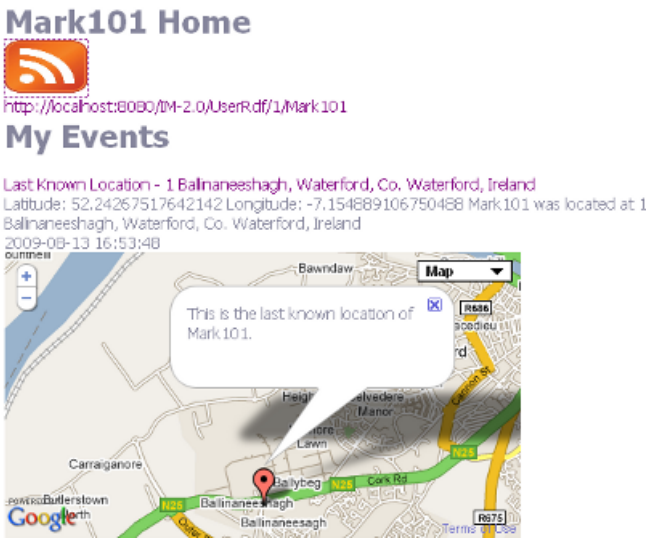

Fig. 4. IM-2.0 home page detailing the locaton of the user.

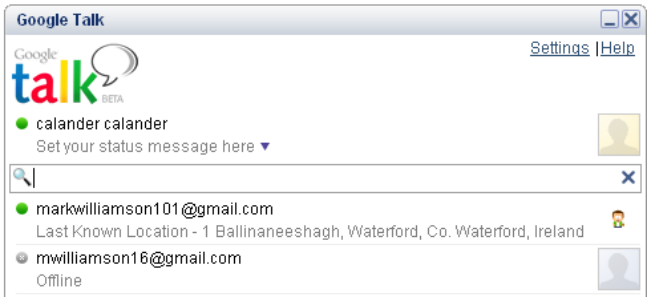

Fig. 5. IM sub-presence showing the location of the user from the user's contacts perspective.

4.7 Semantic Sparql Scenarios

A Semantic web page is available through the IM-2.0 web site which allows the user to execute sample Sparql queries with the results dynamically appearing on the screen. Two predefined queries can be manually executed:

- Dynamically fetching all of the individual's web events;
- Retrieving all of the user's contacts.

Additionally, manual Sparql queries can be constructed which can be used to retrieve information from any internet source that is RDF compliant. This can be used as an introduction and learning exercise for RDF and Sparql. The XMPP integration layer is enriched to understand RDF via automated Sparql queries that it can perform Figure 6.

```
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
PREFIX j.0:  <http://rdfs.org/sioc/ns#>
SELECT ?topic ?about ?site ?version
WHERE
          { ?x j.0:topic ?topic .
            ?x j.0:about ?about .
            ?x j.0:Site ?site .
            ?x j.0:latest_version ?version }
```

Fig. 6. A Sparql query, which defines the ontologies it expects to use to retrieve Web 2.0 updates from the IM-2.0 RDF user page.

5 IM-2.0 Architecture

The architecture diagram Figure 7 depicts all the components that make up the entire system. In short, a Web 2.0 service is adapted into an IM-2.0 module by developing it to adhere to the distinctive web adapter interfaces.

The IM-2.0 framework polling timers log into a users Web 2.0 services to check for newly created content. If new content exists, the details describing it are saved in a database as well as the URL pointing to the update.

This information can then be accessed in the database via data access objects and populated into specific models. These models can then be sent to the RDF and RSS transformer classes to be converted into separate RDF and RSS pages when requested through a browser. Furthermore, an application such as a RSS reader or SPARQL query application could interface with the pages to get information.

The XMPP integration IM account's manager handles all the instances of the GTalk accounts that are logged in. These accounts are communicating with the

GTalk server which in turn is keeping track of all the instances of any one login ensuring all logins share the same sub-presence. The IM accounts manager is semantically enriched and can directly interface with the RDF page gathering the information requierd to automatically update a user's sub-presence and generate responses to IM request.

The web site acts as the view for displaying the information mined back to individuals. Features such as account creation, management, email verification, encryption and authentication are supported. Error checking and avoidance is performed on all areas of the site. As well as that, a person can manually update their RSS and RDF page from their IM-2.0 home page. The ability to search for a particular user based on display names is offered, which subsequently provides the option to follow or to stop following that person. The following concept is similar to the Twitter following concept where a user can simply be added as a contact. Users' contacts latest events are also visible from their own home page.

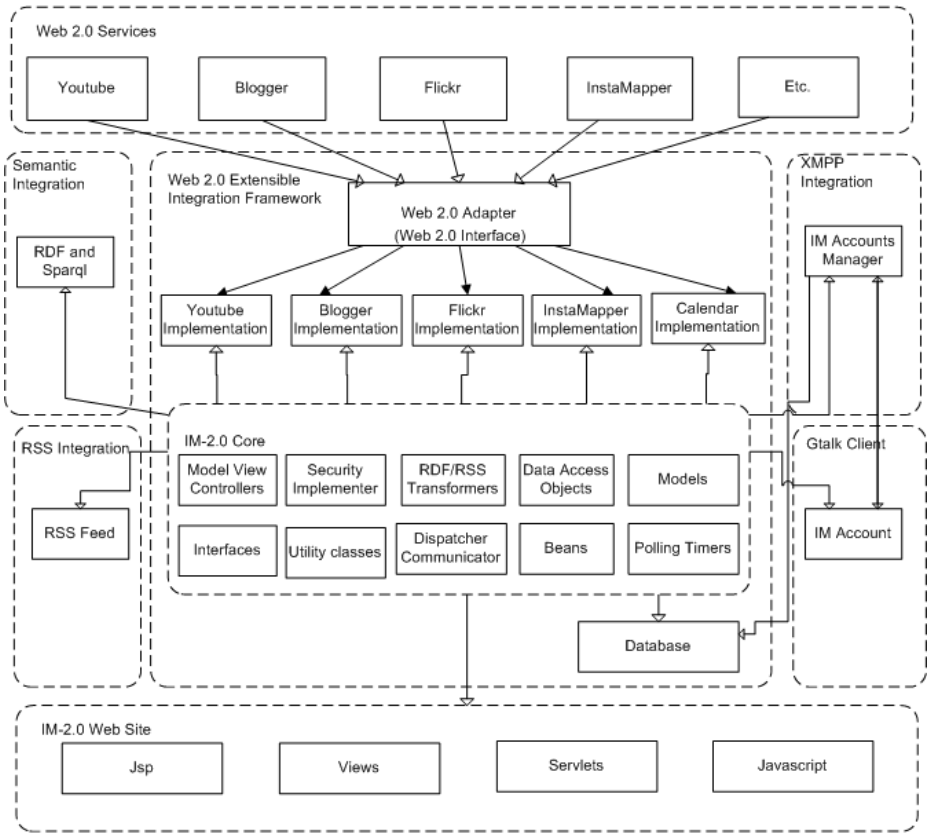

Fig. 7. System Architecture.

6 Framework

The IM-2.0 framework consists of the following main components:

- Web 2.0 extensible integration framework: Framework will encapsulate multiple web 2.0 services by providing a standard interface to each service from a single server;
- XMPP integration layer: An implementation of the XMPP protocol which will support concurrent GTalk account logins and provide IM service functionality;
- Semantic integration layer: Information collected from Web 2.0 services is transformed into RDF using semantic technologies and ontology's;
- IM 2.0 Web Site: Provide IM-2.0 front end in the form of a web site GUI. Provides IM 2.0 registration, account management, activity management, home page, user search and contacts as well as present the online activities of users in the form of a unique user RSS feed / RDF page.

6.1 Web 2.0 extensible integration framework

The IM-2.0 server encapsulates multiple web site services that are freely available over the internet. The ability to scale adding additional web services is also possible. This is achieved by providing a standard interface to each service from the IM-2.0 server side thus enabling the Web 2.0 extensible integration framework to be created that collects and exposes Web 2.0 information for the website, XMPP, RSS and Semantic layers.

Upon researching a number of Web 2.0 services it became apparent that many do not expose all the required information through RSS or semantic web technologies. Hence there is a dependency on third party. It was deemed five services would serve in proving the concept. Blogger, Google Calendar, Flickr, InstaMapper and Youtube were chosen.

The Web 2.0 extensible integration framework was designed in a manner that would be robust and scalable allowing Web 2.0 components to be added as quickly and as simply as possible. With this in mind the IWebService interface was created which must be implemented by any Web 2.0 service module that is being integrated into the Web 2.0 extensible integration framework. This interface provided the bare bone blueprints required to connect to a particular service.

Furthermore a universal method is exposed that would be aware of the features supported by any given service allowing for a variety of tasks to be performed by calling one simple function. Figure 8 code fragments illustrate the methods that must be implemented to adapt a web service for integration as a Web 2.0 module.

A wide range of Web 2.0 services exist, each unique service offering its users different features such as photo upload, calendar event creation or location features as discussed earlier. Instead of exposing implementation methods for all

```
public interface IWebSession {
void initialiseAuthentication(ArrayList<String> listAuth)
void initialiseConnection(String strUserToken)
void initialiseConnection(String strUsername, String strPassword)
void serialiseData()
int getServiceID()
void revokeAuthentication(String strAuthToken)
}
```

Fig. 8. IWebSession interface.

these features through one interface resulting in a complex and large class for a Web 2.0 module which may or may not support half of the methods provided. A number of Web 2.0 feature interfaces were created which would be chosen and implemented when adding a module to the framework.

A study was carried out to make note of the varying features provided through the most popular Web 2.0 services. This resulted in the following interfaces being declared; IWebBlog, IWebComment, IWebContacts, IWebEvent, IWebMedia, IWebMusic, IWebPhoto, IWebProfile, IWebVideo and IWebLocation. Each interface specialised in a particular feature of Web 2.0 activity and provided methods for implementing that feature in a universal manner. Figure 9 illustrates the differences between the IWebBlog and IWebComment adapter interfaces.

```
public interface IWebBlog {
void getServiceBlog()
String createServiceBlog(String strTitle,
String strContent, String strAuthorName, String strUserName, boolean isDraft)
String createServiceBlog(String strTitle,
String strContent, boolean isDraft)
void deleteServiceBlog(String strHref)
}

public interface IWebComment {
void getServiceComment()
void createServiceComment(String strEntry, String strComment)
}
```

Fig. 9. IWebSession and IWebComment interfaces.

In order to add Blogger as a module to the framework the following interfaces would have had to be implemented; IWebSession, IWebBlog and IWebComment Figure 10.

6.2 XMPP integration layer

Using the smack API provided by Jive Software, a XMPP Jabber compliant implementation was developed which supported the ability to log into XMPP

```
public class BloggerInitialisation implements IWebSession, IWebBlog, IWebComment
{
...
}
```

Fig. 10. Adapting the Blogger service.

compliant servers offering standard IM functionality. Such servers include Google Talk.

A number of core interfaces were developed describing features from various aspects of standard IM protocols. These include ISession, IListener, IGroup, IConversation, IContactList and IContact. The interfaces would need to be implemented in order to create a fully compliant functioning IM system. Furthermore, the interfaces and implementations are kept separate allowing for the option of other IM protocols to be integrated without duplicating dependent code. The ISession interface exposes methods to aid in the implementation of an IM protocol Figure 11.

```
public interface ISession
{
  enum ConnectionStatus {CONNECTING, DISCONNECTED, CONNECTED, AUTHENTICATED};
  enum PresenceType {AVAILABLE, UNAVAILABLE, DND, AWAY};
  ConnectionStatus getStatus();
  void connect(String server, int port, String serviceName);
  void disconnect();
  IConversation createConversation(ImpBuddy buddy);
  void authenticate(String username, String password) ;
  void register(String username, String password);
  void setPresence(Type stat,Mode mode,int stat,String subPres,String imAdd);
  IContactList getBuddyList();
  void addBuddy(String buddyName);
  void removeBuddy(String buddyName);
  void createGroup(String buddy, String group);
  void renameGroup(String oldGroup, String newGroup);
  void addBuddyToGroup(String buddy, String group);
  void moveBuddy(String buddy, String group);
  PropertyChangeSupport getPCS();
}
```

Fig. 11. ImpSession interface.

The common methods in the interfaces such as connect, disconnect, createConversation etc. were also implemented as universal commands which could be called from a managing IM session class. Such a class is the IM accounts manager, which manages all the IM sessions on the server. This can be used to easily get any instance of a live IM session, enabling IM-2.0 to manipulate the connection and account.

The IM accounts manager, is semantically enriched allowing it to reach out on the fly to read an individuals RDF page, checking for the latest event details. Figure 12 depicts the methods available to the IM account manager.

```
public interface IIintXMPP
{
 void initXMPP();
 void initXMPP(Subscriber subscriber, IWebServiceAuth webServiceAuth;
 void connect();
 void disconnect();
 IConversation createConversation(ImpBuddy buddy);
 void authenticate() throws Exception;
 void setPresence(Type status, Mode mode, int intStatus, String subPresence, String strIMAddress);
 public ConnectionStatus getMyStatus();
 public WebEvent getMobjEvent();
 public HashMap<String, Integer> getAccountPresence();
}
```

Fig. 12. ImpSession interface.

Research uncovered that Shared Status Messages is a XMPP extension used by Google to enable multiple applications signed in as the same user to report the same status message. This extension was implemented on the IM-2.0 server which enabled all instances of the GTalk account to be logged in simultaneously and change their presence and sub-presence, which is further propagated to all other logged in sessions with out causing any conflicts. This enables the dynamic remote updating of IM user's sub-presence.

6.3 Semantic integration layer

An API called Pellet, has amalgamated other semantic web APIs such as Jena and OWlAPI. It is a Java API complimenting the development language chosen for this project. Pellet supports the creation of many Semantic standards such as RDF, RDF Schema (RDFS) and OWL, including the ability to create and import semantic ontology's. It also facilitates the creation and execution of SPARQL queries.

A tool called Protege offers a graphical user interface (GUI) providing the ability to create RDF, OWL, RDFS as well as create ontology's and perform SPQRQL quires in order to validate the Semantic content generated from the projects code.

Ontology's such as FOAF, VARD and SIOC provide the majority of the linkage and functionality required to accurately describe new events occurring on users Web 2.0 sites.

RDF was generated using the Jena API, which utilised FOAF, VCARD and SIOC. Jena did not fully have the SIOC ontology embedded but since SIOC is an open ontology it could be imported into Jena for usage. A tool called Schemagen

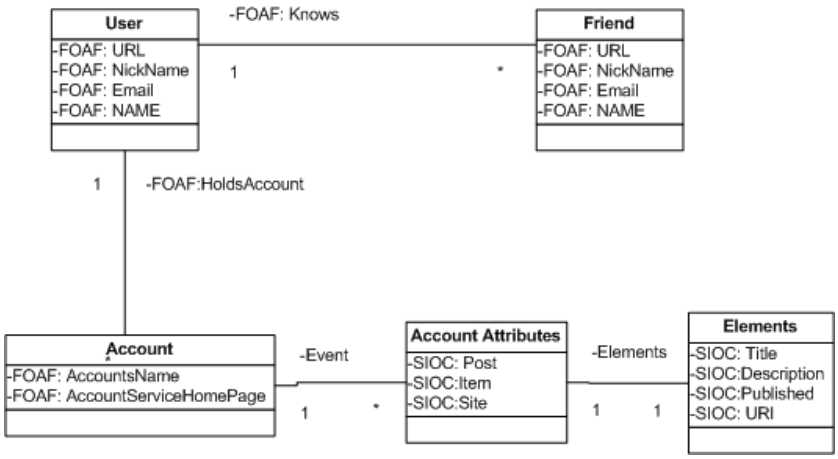

Fig. 13. The information structure available through RDF.

provided by Jena was configured correctly to import SIOC, as Java constants in order to be used through Jena.

This allowed for the creation of a RDF page from database tables describing a user's activity on a number of web sites. An overview of the information depicted in RDF form can be seen in Figure 13. It could be explained as follows, a user has an IM-2.0 account who knows other users on IM-2.0 or on other Web 2.0 services; as well as that, this user holds various other Web 2.0 accounts with certain web 2.0 events occurring on those accounts which can be further broken down into elements describing the event.

This RDF page is generated from information stored in the database. The data is dynamically read into data models which in turn are transferred to the RDF transformer, building a unique user RDF page on the fly, when ever a HTTP request is received by the IM-2.0 server for a particular users RDF page. Using Pellet to generate Sparql queries, the IM-2.0 server can query the RDF page for any particular updates and then broadcast these updates through various technologies. The beauty of RDF lies in the fact that data captured from the IM-2.0 RDF page could also be mashed up into other RDF sources in the event other pages supported semantic technologies. An extract of the RDF can be seen in Figure 14 detailing the creation of a blog entry.

```
– <rdf:Description rdf:nodeID="A6">
  – <j.1:Site>
      http://apibloggertest.blogspot.com/2009/04/project-demo.html
    </j.1:Site>
    <j.1:latest_version>2009-04-17 08:08:00</j.1:latest_version>
    <j.1:about/>
    <j.1:topic>New Blog Entry - Project Demo</j.1:topic>
  </rdf:Description>
```

Fig. 14. An extract of RDF describing a blog entry.

7 Concluding Discussion

Having analysed the relevant literature in the research domains as well as after developing a software solution, it is now clear that IM, Web 2.0 and the Semantic Web were applicable and have brought varying strengths to the IM-2.0 project. The three core technologies exist and operate in the World Wide Web providing varying functionalities with different levels of maturity and adoption. Combining the strengths of each technology has enhanced collaboration and openness between the services choosen to prove the concept. A hybrid system that utilises the strengths provided by each domain has significant merit, yielding a dramatic improvement in terms of maintaining and increasing online communities as well as promoting traffic and participation in internet communities.

IM's inherent communications ability is used to scale up the number and range of people that can simultaneously observe and participate in a person's Web 2.0 world. By making a person's IM account behave like a chatbot in certain scenarios, IM-2.0 has expanded the range of potential user participation, increasing traffic and click through rates to Web 2.0 content and the services respectively. IM also provides a degree of accountability as a person is associated with their IM account, if a contact has added a person as a friend and has held conversations proving that the person is who they say they are, they are more likely to trust information that is provided through the user's IM account such as IM-2.0 updates.

The idea of linking web sites, technologies and promoting collaboration on the internet can only be truly achieved if the Semantic Web idea was adopted and implemented by everyone. Of course this could never happen for all of the content on the internet due to security and privacy concerns, but for freely available content and information, it is the next logical move to take. The Semantic Web can and will be used in the coming years for describing data which will be utilised by developers to create interesting and helpful applications for the general public.

7.1 Opportunities

There are many other opportunities for future research to further expand the IM-2.0 project, considering the vastness of each research area undertaken thus far. The current RDF page that is generated could be expanded to incorporate OWL, which would link data more accurately using data relationship techniques. Furthermore other semantic ontology's could be added in order to broaden the information represented through IM-2.0.

As well as that perhaps a rich front end could be wrapped around the semantic data creating a semantically enriched user profile page originating from multiple Web 2.0 services.

Other Web 2.0 services could be investigated and added to the project such as Facebook, Bebo and Twitter increasing the data mined and exposed through IM and semantic technologies.

Further work could be carried out on the XMPP integration layer in order to automatically switch a user to IM-2.0 mode when they go offline. As well as

that, research could be carried out with an aim of creating an IM-2.0 instant messaging buddy that could alert the user when another person has commented on their Web 2.0 created content or perhaps also allow the user to create content such as blog entries by sending an IM message to the IM-2.0 buddy. As well as that, other IM protocols could be investigated to discover if the IM-2.0 idea could be applicable.

8 Implementation

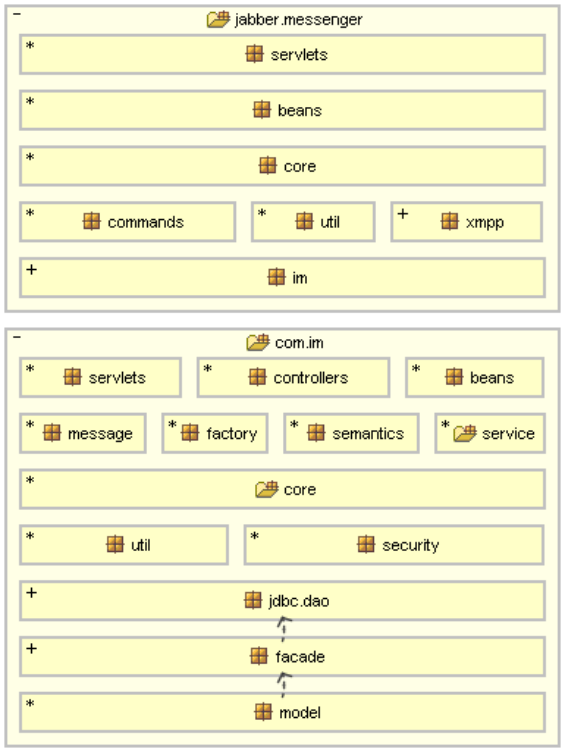

Fig. 15. Packages of IM-2.0 project.

A demo of IM-2.0 has been developed which goes beyond that of just a simple prototype. Many industry standard features that users have come to expect in a fully fledged web site have been incorporated into the project.

Figure 15 is an architecture diagram of the java project created using Structure 101 [22]. It details the packages used in the IM-2.0 project including the dependencies each package has on one another. It demonstrates the thought

and design considerations that have gone into the development of the project to ensure it complies with industry standard techniques. The project consists of over 90 classes and 40 interfaces. Figure 16 is the overview of the database design used for the project. Tables exist to store subscriber information, unique subscriber cryptographic keys, subscriber Web 2.0 credentials, Web 2.0 authentication keys, Subscriber Web 2.0 events, Supported Web services, types of events that can occur on a web service as well as subscribers contacts.

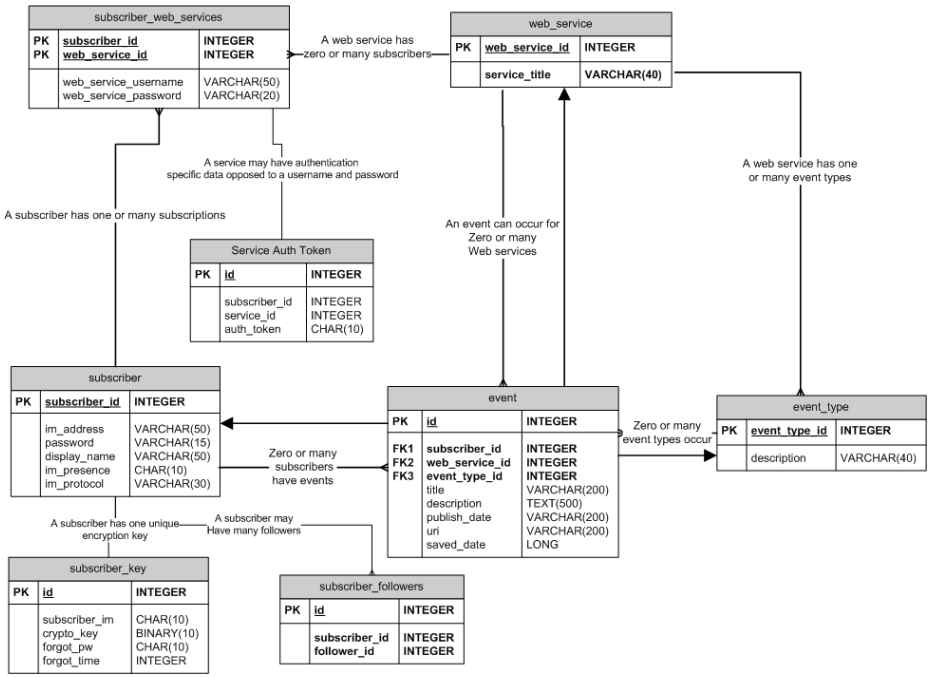

Fig. 16. Database Tables for the IM-2.0 project.

References

1. Social Networks Facebook Takes Over Top Spot, Twitter Climbs, 2008. http://blog.compete.com/2009/02/09/facebook-myspace-twitter-social-network/.
2. IM Gets Down To Business., 2006. http://dev.aol.com/node/150.
3. I Shigeoa. *Instant Messaging in Java - The Jabber Protocols.* Mannng, 2002. United States of America.
4. Sarita Yardi Stephen Chan, Benjamin Hill. Instant messaging bots: accountability and peripheral participation for textual user interfaces. *ACM 10.1145/1099203.1099221*, pages 113–114, 2005.
5. Stephanie Smale and Saul Greenberg. Broadcasting information via display names in instant messaging. *ACM 10.1145/1099203.1099218*, pages 89–92, November 2005.
6. About XMPP, 2006. http://xmpp.org/about/.
7. Who Coined The Word Web 2.0., 2009. http://daftseo.com/who-coined-the-word-web-20/.
8. Levels of the Game: The Hierarchy of Web 2.0 Applications., 2009. http://radar.oreilly.com/2006/07/levels-of-the-game-the-hierarc.html.
9. Ajax: A New Approach to Web Applications, 2005. http://www.adaptivepath.com/publications/essays/archives/000385.php.
10. Paul Anderson. What is Web 2.0? Ideas, technologies and implications for education. pages 10–29, 2007.
11. A Story about RDF and XML., 2001. http://xmpp.org/about/.
12. OWL Web Ontology Language Overview., 2001. http://www.w3.org/TR/2004/REC-owl-features-20040210/#s2.2.
13. N Shabir. Open World Thinking. pages 10–11, 2008.
14. Ankolekar et al. The two cultures: mashing up web 2.0 and the semantic web. pages 825–828, 2007.
15. Bojars et al. Using the Semantic Web for linking and reusing data across Web 2.0 communities. pages 21–25, 2007.
16. Ereteo et al. Leveraging Social data with Semantics. page 4, 2008.
17. SocialURL - Showcase all your web profiles with a single URL, 2009. http://socialurl.com/default.aspx.
18. Power - All your friends in one place, 2009. http://www.power.com/Pub/Login.aspx?ReturnUrl=%2fpriv%2fpower%2fhome.aspx.
19. switchAbit routes your content to multiple social services, 2009. http://beta.switchabit.com/.
20. Gnip - Delivering the Web's Data, 2009. http://www.gnip.com/.
21. A Etzioni and O Etzioni. Face-to-face and computer-mediated communities, a comparative analysis. pages 241–248, 1999.
22. Structure 101 - Keep complexity out and get more done, 2009. http://www.headwaysoftware.com/index.php.