13th ICCRTS
"C2 for Complex Endeavors"

Title of Paper

Expanding the Dynamic Collaboration in Teams of Portal and Non-Portal Based Users Using Semantic Based Tools and Constructs

Topic(s) : Collaborative Technologies for Network-Centric Operations,
Networks and Networking
Architectures, Semantics

Name of Authors
Shon D. Vick
Osbaldo Cantu
Amen Ra Mashariki
Christopher Steen

Point of Contact
Shon D. Vick

Name of Organization
Johns Hopkins University Applied Physics Laboratory (JHU/APL)
11100 Johns Hopkins Road
Laurel, MD
443-778-8378
shon.vick@jhuapl.edu

Title of Paper

Expanding the Dynamic Collaboration in Teams of Portal and Non-Portal Based Users Using Semantic Based Tools and Constructs

Abstract

Collaboration is integral to effective C^2. Previous JHU/APL research focused on dynamic collaboration within a portal environment between homogeneous users. This paper details continued research on how both non-homogenous, disadvantaged users, operating in a mobile environment as well as portal based users could share information for effective C^2. It also addresses how the effectiveness of this integration could be enhanced using semantic tools and constructs. The work on non-portal based applications treats the effect of team members operating with unequal capabilities potentially using different devices. It explores the adoption of RSS, mobile devices, and other technologies. Various architectures are explored including the Java 2 Micro Edition (J2ME) and associated technologies for evaluation of collaboration tools and technologies. The portal based work is centered on a geo-spatial display and an associated messaging system.

Various directions enhancing collaboration were considered that would enable integrating deeper semantic content into the information shared for C^2. Such tools for future evaluation are Wikis, Blogs and semantic based messengers. Among the benefits with the provision of semantic meta-data are personalized presentation of content (e.g. based on personal preferences), intelligent search, improved interoperability between systems (e.g. integration of several applications in a federated process, or with the support of smart agents).

Keywords: Collaboration, Disadvantaged Devices, Social Software, C2, Semantics

1 Introduction

This paper discusses several research efforts at The Johns Hopkins Applied Physics Laboratory (JHU/APL) to support collaboration for C2. The first project explored in Sections 2-4 developed a prototype to evaluate the feasibility of an integrated capability that synchronizes text and geospatial displays using a limited form of semantic web technology. The second research effort described starting in Section 5 examined an architecture that supported disadvantaged users (i.e. users with lower system capabilities). Each effort is characterized in terms of its background, a technical description of its architecture and development environment for its prototype. These technical descriptions assess system characteristics, strengths and limitations. Future work for each system is also suggested.

A final part of this paper is an in depth analysis of the fundamental limitations of these two architectures in their present form with an analysis of what technologies should be explored to exploit more semantically rich representations.

2 Portal Based System Background

The term "User Defined Operational Picture (UDOP)" corresponds to a variety of capabilities for users in the military, intelligence, and civilian domain. Simply stated, UDOPs provide users with the ability to choose what data they want to visualize in their user interface. This study focused on the user interface of visualization applications. Ideally, UDOPs provide users with a display of data they can interpret quickly and effectively, given that the user has customized the display to match their preferences and needs. In military context, a user would have the ability to define what he or she has on their visualization display in order to enhance their situational awareness of the mission. When geo-spatial visualization technologies are coupled with technologies such as web applications, text chat and natural language processing as well as data exchange standards that promote cross-platform interoperability, UDOPs can then provide a holistic, integrated environment that supports both individualized and shared *sense-making* and situational awareness.

The objective of this research was to demonstrate the feasibility of developing an integrated capability in a web portal environment that synchronizes text and geospatial displays using semantic web technology. The primary goals of the project were:

- integrate the geospatial display and chat clients into a single web application,
- interface chat with geospatial display based on semantic relationship

The secondary goal of the project was to convert chat and geospatial display to web parts and portlets for deployment into a web portal environment.

3 Portal Based Architecture

The research team examined various options for each of the architecture components. The project was built upon four components that comprise the architecture. Figure 1 depicts how the frameworks and services interact with each other, the location of each component, and the data exchanges.

Web application framework: The research team picked EXT[1] to host the web application over DOJO[2] and Yahoo UIs. The EXT code resides at the portal server, but is accessed (along with user interface methods) in the web browser by the client. EXT allowed the team to develop the proper "piping" between each of the other frameworks and provided the user interface requirements management requested for visualization capabilities.

Chat framework: Two components were used to deploy the full chat capability. For the server-side, JABBER was used for the management, processing, and exchange of chat messages. Because of its ease of use and familiarity to most users, the Microsoft instant messenger was used to view the chat text in the web browser on the client side.

Geospatial framework: Leverages various public services over the Internet. For familiarity and richness in satellite imagery, Microsoft's Virtual Earth 5.0 APIs were used for the base-maps and navigation tools. The framework included the ability to switch between various geospatial services, using other services such as ArcGIS Server and Google Maps.

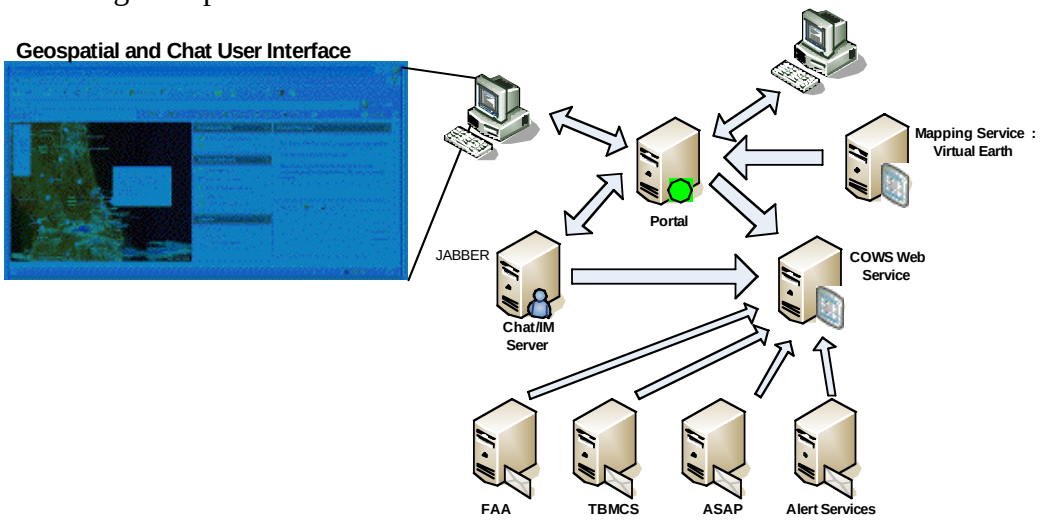

Figure 1 Architecture and Data Exchange

[1] http://extjs.com/
[2] http://dojotoolkit.org/book/dojo-book-0-9-0

Data services: The research team chose to use data exchange standards, which allow for interoperability across platforms, criteria critical to this research project. Three different types of datasets and exchange standards were selected as test cases: (1) a web service that produced FAA tracks (2) the tiling service that produced Doppler radar of the entire US and (3) GeoRSS feeds of events throughout

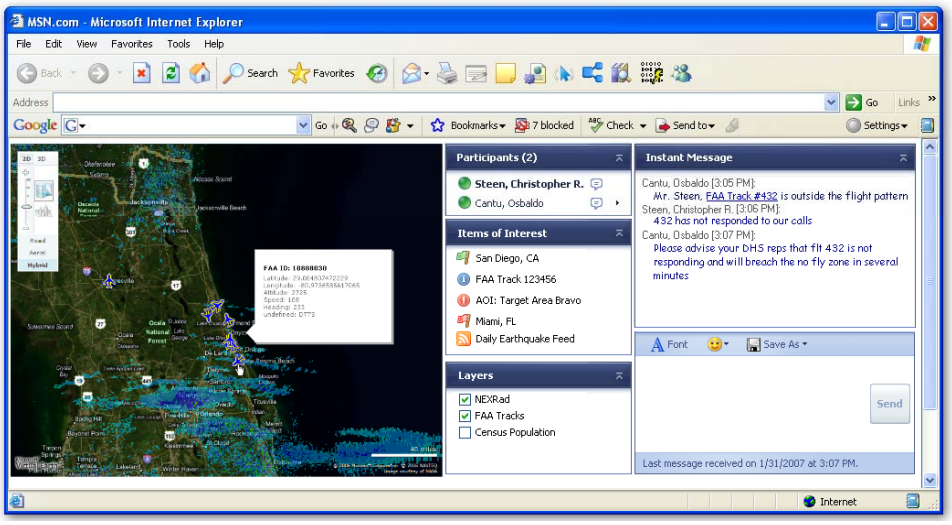

Figure 2 Geospatial display and Chat (A Collaboration Manager)

3.1 The Collaboration Manager (CM)

In the current incarnation of the system, geospatial collaboration is effected by a Collaboration Messenger (CM) as shown in Figure 2. Tool users communicate using the CM as an example of Social Software. The term Social software is often applied to a range of web-enabled software programs. It is difficult to precisely define the term social software but it usually taken to encompass systems that allow users to interact and share some body of knowledge (BOK) with other users in a dynamic way. Wikipedia, MySpace, Facebook and Google Talk are well known examples from the non-government domain. In a C2 setting such a BOK can be used to disseminate situational knowledge, tactics, and procedures and so on.

The CM allows a user to communicate and share information with other users listed on a User Roster (UR). Information is exchanged through text and a construct called a Point of Interest (POI). POIs may exist in a database or be defined by a user. An entry in the UR is composed of username and email addresses. These are needed by the CM to allow collaboration amongst users. These are used essentially for routing of messages; text or POI. Additionally there is a notion of a room which is simply a collection of users taken from the roster. Users may enter and leave a room. All information exchanged within a room is exchanged between all members of the room and no one else. The Jabber protocol[3] is used to exchange textual data and the POI between CMs. There is no

[3] http://wiki.jabber.org/index.php/Main_Page

semantic information or meta-data in this exchange. So for example the system cannot route text or POI to everyone on the UR or to a room with some interest or need to know based on the interests of the users described in the UR because there is no explicit representation for those interests nor can they be inferred.

4 Future Work

As shown in Figure 2 the geospatial display and chat clients were integrated into a single web application. The geospatial display primarily uses Microsoft's Virtual Earth for the imagery basemap though other services were tested as well. The chat client uses Microsoft Instant Messenger interface for the client-side. The research team successfully tested the chat client inside the portal based web application. This evaluation included a test that involved chat with another application unrelated to project web application. This testing demonstrated that the possibility for integration with disadvantaged users is certainly viable. Much more experimentation is needed to evaluate the viability of this technology framework as a means of command and control.

An interface between chat and geospatial display based on a very simple semantic relationship was attempted. The research team was successful in linking the data that was displayed in the chat interface to the geospatial display. For example, if the chat text showed FAA Track #12z3, the user could click on the link and the map would reset and be represented in the geospatial display. The team was also successful in developing the Chat Object Web Service (COWS) that served as the mediating web service between the data services and the chat and geospatial frameworks. A natural language application was used to implement a limited real-time parser that could identify chat messages related to geospatial features (e.g., cities). However semantically richer interaction between the geospatial display or even between two chat sessions was not accomplished. Section 9 addresses issues involved in such semantic integration.

5 Background - Dynamic Collaboration for Non-Portal Users

Collaboration can be viewed as the backbone of effective Command and Control. The focus of previous research efforts at Johns Hopkins APL (JHU/APL) was on collaboration conducted within a portal environment. This section of the paper discusses a different approach that explores how a disadvantaged user (i.e., a user with lesser capabilities) operating in a mobile environment could interact and share information with a collaborative team operating in a portal environment. The value of this effort is that it extends previous work to better address heterogeneous teams; (i.e. teams with members operating with different and likely not equal capabilities. The following specific objectives were established for this effort:

- Define for the effort what it means to be a "disadvantaged" user.
- Build an RSS data server as well as a mobile RSS consumer/reader.
- Allow mobile devices to consume web services with or without standard protocol.
- Develop a method to allow a "disadvantaged" device to display data on a screen that was possibly meant for a portal user.

Achieving these objectives required significant research to gain an understanding of the current capabilities of key technologies. Meeting these objectives yielded a prototype capability that demonstrates a “disadvantaged” user (perhaps a disaster relief responder) interacting with a collaborative team operating in a portal environment.

6 Non-Portal Architecture

The technologies employed for this exploration included the Eclipse Java Integrated Development Environment (IDE), the RSS parsing library, JSR 172 compliant protocols, the Java 2 Micro Edition (J2ME) software development kit, and the Apache Tomcat Web server. To facilitate the deployment of Java across devices with a vast array of features and capabilities, it was necessary to change the architecture significantly from the all-in-one nature of J2SE and J2EE. J2ME is not simply a Java Virtual Machine (JVM) with APIs; J2ME is a framework on which a J2ME “configuration” targets a range of devices with a specific set of capabilities, e.g. mobile phones. A “profile” selects a configuration and a range of APIs that facilitate the development of a range of applications targeted at a specific device or range of similar devices, depending on the configuration specified.

Essentially, there are two kinds of configurations available to J2ME. The first is for devices that are resource limited, typically mobile phones. This configuration is called Connected Limited Device Configuration (CLDC). It consists of a Virtual Machine and basic set of APIs for use with devices that typically have 128K - 512K memory available, limited power, a simplified user interface and network connectivity like GPRS, Wi-Fi or Bluetooth. The second configuration is the Connected Device Configuration (CDC) which is the framework to build applications that can be shared across consumer and embedded devices such as PDAs and set top boxes. This configuration is targeted at devices that have >512K memory available and are connected to a network resource. The configurations are nestable. An application made using the less capable configuration (CLDC) should be able to be executed on the more capable configuration with little or no changes.

The Mobile Information Device Profile (MIDP) provides a standard run-time environment for the most popular Java-enabled portable information devices such as mobile phones. The goal of the MIDP profile specification was to provide a complete set of APIs and a properly configured Virtual Machine so that applications written using this specification can have the greatest compatibility possible with as many devices in the mobile market. Essentially, if an application was built and runs to MIDP specification, it should run on any phone certified for J2ME and MIDP 2.0 compliant.

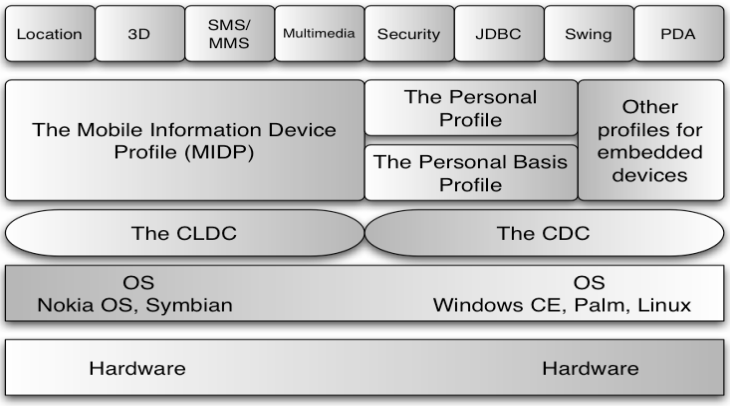

Figure 3 Non-Portal User Project Architecture

It isn't always feasible to develop on target machines (e.g. resources may not be available). To accommodate this, Sun has created a MIDP emulator which emulates devices capable of meeting the specifications needed to run an application that is developed for the MIDP specification. Emulating a mobile device that implements MIDP1/2 is not an entirely simple affair, as device manufacturers have varying levels of security for untrusted/signed third party applications. For this reason, manufacturers often provide their own emulators that either emulates a specific device, or series of devices. These often have configurable options that allow the developer to emulate the security of a specific device with regards to untrusted, unsigned programs as well as trusted, signed third party applications. Restricted APIs like the Connector API and the FileConnection API are not available to untrusted applications. Applications under development can be signed for a limited amount of time using the wireless toolkit; this is solely to allow applications to be tested on their designated devices.

A small application that allows a user to register a number of RSS feeds was developed. The device/application checks every so often to see if there is anything new on these feeds. If there is something new, the new update is added to a "last updates" section of the application. The application allows the user to register and delete RSS feeds and check the latest information from the RSS feeds. Ideally, this application would also have a background mode of operation by which the application can check the RSS feeds periodically and alert the user if there were any updates. A few very small MIDlets were developed to first explore some of the functionality that would later be employed to create the RSS reader. These applications were essentially proofs of concept for elements such as the GUI and the Record Management Store, enabling the lead developer to better understand how these elements are implemented in J2ME. One such application was developed with the aid of GUITests, a small MIDlet that served as a sample implementation of an interface using the LCD GUI API. Another application was a small MIDlet that stored a simple string in the RMS. This small MIDlet implemented an interface for the RMS that stored a simple string that could be retrieved again once the application had closed and started again. These served as valuable tools in deciding how to approach the development of the project.

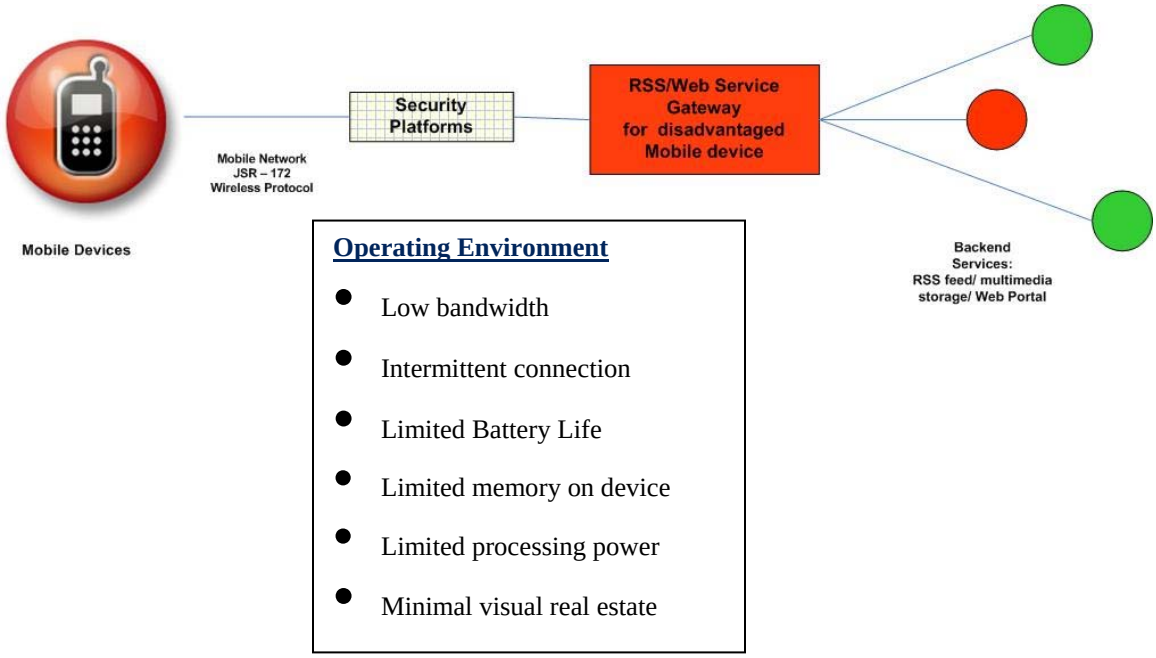

Figure 4 Proof of Concept Depiction

7 The Non-Portal Development Environment

Eclipse, an open source, vendor neutral IDE was chosen as the development platform running on top of the Java SDK1.5, with compile options set to Java 5.0 compliance. Eclipse provides support for a variety of plug-ins, which allows the IDE to be used for much more than just Java development. One of plug-ins used in the development of the application was the EclipseME.org J2ME plug-in, which implements functionality that aids in the development of J2ME applications. This functionality includes fully integrating the Java wireless toolkit, which provides configurations and profiles for mobile development. The plug-in also integrates the MIDP2 emulator that is provided with the wireless toolkit which can then be launched from within Eclipse. The second plug-in used in developing the application was the Nokia Development Suite version 2.0. The device used as the test platform was a Nokia 6630 mobile phone, which is a Series 60 phone. This plug-in provided support for emulating the Series 60 model phones which run the Symbian operating system. It also provided tools for deployment of packages, automatic JAR and JAD file creation and the ability to sign the application for a short period of time for testing purposes. The Eclipse IDE along with the two plug-ins described above formed the Development Environment under which the application was developed. With Eclipse providing error checking and compilation management, the J2ME plug-in provided the CLDC and MIDP2 APIs, and the Nokia Developer Suite2.0 provided emulation and deployment packages.

8 Non-Portal Future Work

The are several level of considerations for future work. Advances that are incremental and those that are more radical in nature are both considered in the next two sections. Incremental improvements include advances in geospatial data rendering that would allow an application the ability to receive geospatial data renders in gif or jpeg form are a next logical step. This can be done using the same format and technology that is currently used to send multimedia data over the RSS wire..

Advances is scheduling that allow a program to automatically check for updates to an RSS feed, and if found, download the feed for later perusal by the user will be possible. MIDP 3.0 is currently in the middle of the Java Community Process and this specification will allow for background MIDlets and scheduled execution

In RSS each item in a feed is generally a small description of a more extensive article. For each entry, the feed should also contain a link to the full article that the item describes. Future work should include implementing a way for the application to open an external program, like a web browser, that could follow the link to the web page and provide the complete article.

Often a user is interested in not one source but in several and wants them combined or *mashed up*. Typically such *mashups* are done by hand. The ability to open up an internal program based on a link has been discussed. A more powerful construct would be to have a software agent brokering such feeds, combining potentially different (but semantically related) feeds by a set of user supplied descriptions or rules. RSS feeds of Blogs and Wikis may be of great use for sharing information in a team. Section 9 describes relevant technologies for incorporating semantically rich content into such items.

9 Limiting Semantics Limits Effective Collaboration

In Section 4 and Section 8 limitations of the current prototypes were evaluated. The fundamental defect in each is the same: there is no semantic processing. To examine the role of semantic content let's re-consider the POI construct in the Collaboration Manager introduced in Section **Error! Reference source not found.**. As described there is very little meaning associated with a POI. It is essentially just a feature on a map. Its only context is given by its location on the map. There are no higher level semantics associated with these POI. So for example if you had an element that represented a particular troop strength (TS) with certain resources (R) you couldn't poise the query – show me all locations with TS > x and R > y; because you have no meta-data representation for that information. In like manner in the Non-Portal prototype there is no way for a program to *reason* about a feed because there is no context associated with the feed or when or how it is intended to be used. This limited ability to describe a POI makes them of less use in describing or communicating situational awareness as part of the context of a particular C2 domain.

Semantic Social Software (SSS), the combination of social software and Semantic Web [4](SW) technology has recently been gaining significant attention in the SW community. This balance of this document discusses enhancing social software like the CM or the Non Portal system described in Section 5, by adding semantics based tools and constructs.

9.1 The Semantic Web – The Basic Framework

To understand the potential for using semantic context it is important to understand the evolution and basic structure of the Semantic Web (SW). The SW is a mesh of information linked up in such a way that may be easily processed by machines, on a global scale. You can think of it as being an efficient way of representing data on the World Wide Web (WWW), or as a globally linked database.[5] The Semantic Web is an evolving collection of knowledge, built to allow anyone on the Internet to add what they know and find answers to their questions. Information on the Semantic web, rather than being in natural language text, is maintained in a structured form which can be processed by both computers and people.

The primary visionary and inventor of the Semantic Web is Tim Berners-Lee who was also the inventor of the WWW, URIs, HTTP, and HTML. There is a World Wide Web consortium (W3C) working to improve, extend and standardize the system, and many languages, publications, tools and so on have already been developed.[6] These WWW technologies were used in both prototypes described above.

The SW however has several important technologies beyond eXtensible Markup Language (XML). The Resource Description Framework (RDF) for XML lets everyone create their own tags - hidden labels annotate Web pages or sections of text on a page. Software agents can make use of these tags in sophisticated ways, but the agent developer has to know how tag is used. In contrast XML allows users to add arbitrary structure to their documents but says nothing about what the structures mean.

Meaning is expressed by RDF, which encodes it in sets of triples, each triple being rather like the subject, verb and object of an elementary sentence. In particular, The RDF lets content be expressed using structured metadata statements describing URIs. These triples can be written using XML tags. In RDF, a document makes assertions that particular things (people, Web pages or whatever) have properties (such as "is a sister of," "is the author of") with certain values (another person, another Web page, etc). This structure turns out to be a natural way to describe the vast majority of the data processed by machines. Subject and object are each identified by a Universal Resource Identifier (URI), just as used in a link on a Web page. (URLs, Uniform Resource Locators, are the most common type of URI.) The Semantic Web also supports the notion of ontologies.

---

[4] Semantic Social Software: Semantically Enabled Social Software or Socially Enabled Semantic Web? , Sebastian Schaffert, Salzburg Research NewMediaLab Salzburg, Austria (http://schaffert.eu/download/paper/Schaffert2006_SemanticSocialSoftware.pdf)

[5] http://infomesh.net/2001/swintro/

[6] http://www.w3.org/2002/03/semweb/

An ontology formally defines the relations among terms. The most typical kind of ontology for the Web has taxonomy and a set of inference rules. Ontologies can enhance the functioning of the Web in many ways. They can be used in a simple fashion to improve the accuracy of Web search; a search program can look for only those pages that refer to a precise concept instead of all the ones using ambiguous keywords. More advanced applications will use ontologies to relate the information on a page to the associated knowledge structures and inference rules. It is these tags and the establishment of there C2 ontologies that would allow software agents to behave intelligently. The intelligent fusion of feeds in non-portal system is a good example of this.

9.2 Exchanging Data and Semantics

With diverging applications it is often hard to exchange content between different systems. The existing informal or semiformal structures like hyperlinks or simple unadorned protocols like JABER may be augmented by machine-readable formal descriptions ("metadata") that make explicit the actual meaning behind a connection. Such metadata allows for more sophisticated services, like improved search and navigation (e.g. queries on the structure, context visualization, derived.[7] This would allow more intelligent fusing of feeds as well.

There are various mechanisms to enable/allow higher level meaning to be exchange between two messengers. Among these are Wikis, Blogs and Messengers (like CM). Among the benefits with the provision of semantic meta-data are personalized presentation of content (e.g. based on personal preferences for how to render geo data, interpret links in feeds, etc), intelligent search, improved interoperability between systems (e.g. to integrate several applications in a company process, or with the support of smart agents). The following section discusses each of these three mechanisms with an emphasis on Messengers.

10 Tools for Sharing Concepts

This section discusses three tools that would allow the semantic sharing of a BOK. These include Wikis, Blogs and Semantic Messengers.

10.1 Wikis

A Wiki is a website or similar online resource which allows users to add and edit content collectively sharing a BOK. Wiki are social web sites enabling a potentially large number of participants to modify any page or create a new page using their browser.[8] A Wiki without Socially Enabled Semantics (SES) supports an online collaboration model and related set of tools that allows any user to edit some content within the BOK quickly[9]. In the current incarnation knowledge about a POI or significant text knowledge could be shared through a Wiki by augmenting the BOK with additional information and context.

---

[7] . Semantic Social Software , Schaffert, op cit

[8] SweetWiki : Semantic WEb Enabled Technologies in Wiki, SemWiki2006, Michel Buffa, Gael Crova, Fabien Gandon, Claire Lecompte and Jeremy Passeron, pp 77-88

[9] mobileman.projects.supsi.ch/glossary.html

While this approach has some merit there are drawbacks as well. The user of a Wiki that does not support SES may find navigation and search difficult as the BOK grows large[10]. Tagging items in the BOK with meta-data about the interests or other attributes of the poster, indirectly related items, inconsistent items, and other problems have been noted by researchers in various domains to improve the usefulness of a Wiki[11].

As an example of a SES based Wiki a system called SweetWiki will be discussed. Some Wiki systems have replaced early web technologies (e.g. HTML, HTTP and URIs) by a Wiki markup language (i.e. WikiML or its variants). Among other features mark-up facilities allowed for simple remote edition and storage facilities. SweetWiki abandons WikiML in favor of a tool that is built around RDF, RDFS, OWL, and SPARQL.

A SweetWiki approach that relates semantic content associated with POIs, tagged or un-tagged textual information or UR with entries in a Wikki would be a powerful mechanism to share semantics in a dynamic way.

A simple topology for the use of a Wiki in conjunction with the CM would have all users within a room share a Wiki. Messages exchanged between users in a room could have content that is tagged to refer to content in the Wiki. This is illustrated in Figure 5.

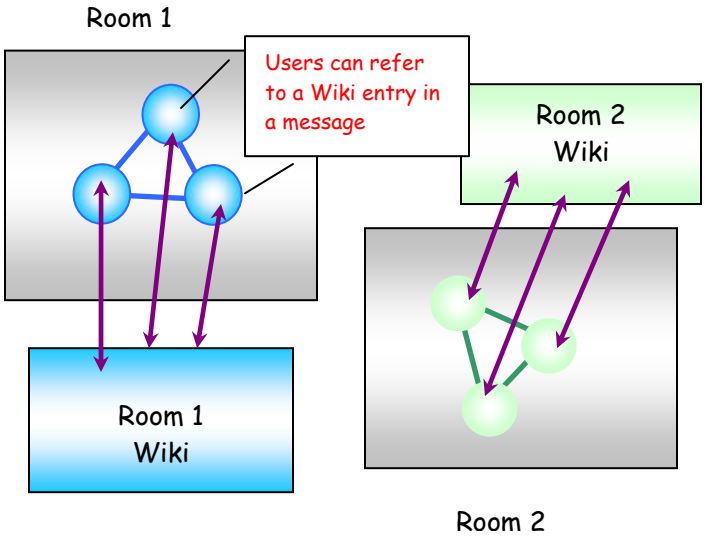

Figure 5 Wiki Topology

Alternate topologies are possible. An architecture that had one Wiki for the entire system, at one extreme, and a Wiki for each user, at the other extreme is also possible,

[10] SweetWiki: Semantic Web Enabled Technologies in Wiki Michel Buffa Mainline Group Laboratory, University of Nice and Fabien Gandon Acacia Group, INRIA laboratory, Sophia-Antipolis, (http://www.wikisym.org/ws2006/proceedings/p135.pdf)
[11] Towards a Semantic Wiki for Science, Christoph Lange Computer Science, Jacobs University Bremen, ch.lange@iu-bremen.de (http://kwarc.info/projects/swim/pubs/kwepsy-swim-abstract.pdf)

10.2 Semantic Blogging

Blogging is a popular and effective approach to information sharing. An extension of blogging would allow the mark-up of blog content;. A mapping of such content to concepts and relationships which are formally described using an ontology could be made by the creator of a blog.. Such labeling provides the potential for a common semantics in ways that an unstructured Blog does not. It also allows for rich categorizations (of data) and subscription (i.e. tell me when X talks about Y). Figure 6 shows another example that could be represented with rich semantic labeling. Here there is a portioning of a set of blogs that allows each user to have their own blog but restrict access from other users.

Some blog commentators envisage connecting blogs using semantic links, and the Topic Exchange activity is a step towards the use of shared ontologies.[12] These facilities would be ideal for sharing context across CMs that would far surpass the present mechanism. Several Semantic Wikis systems exist to date these include WikSAR, Semantic Wiki, IkeWiki and Semper Wiki.[13]

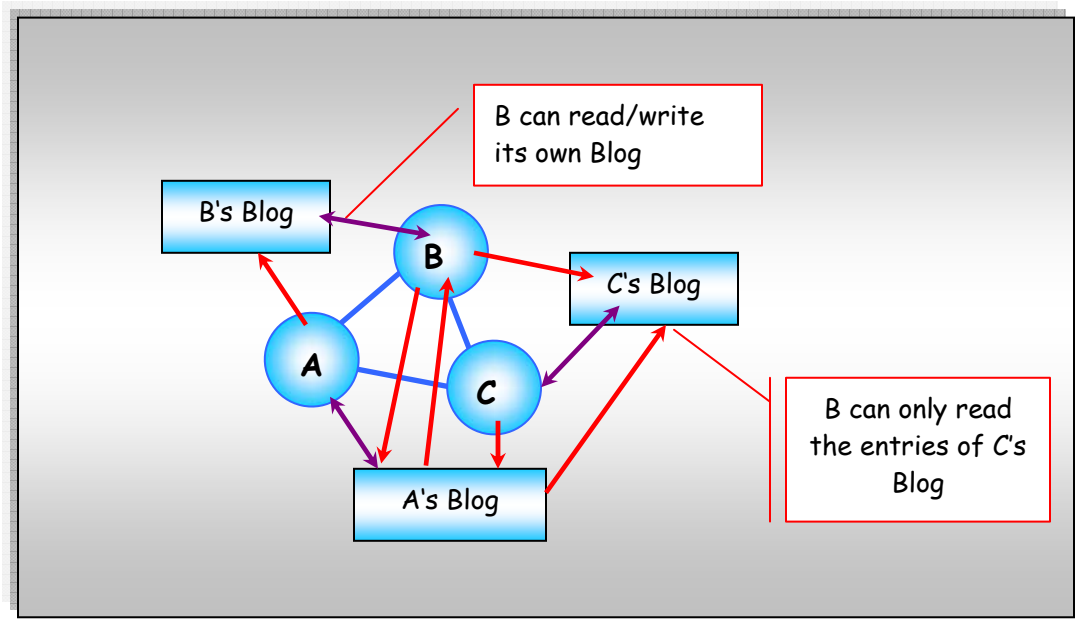

Figure 6 Using Blogs with CM

[12] Semantic Blogging and Bibliography Management, Steve Cayzer, HP Laboratories, Bristol, UK Paul Shabajee, Graduate School of Education and ILRT, Bristol, UK, (http://www.hpl.hp.com/techreports/2003/HPL-2003-130.pdf)

[13] Annotation and Navigation in Semantic Wikis (2006) Eyal Oren, Renaud Delbru, Knud Moller, Max Volkel, and Siegfried., (http://citeseer.ist.psu.edu/cache/papers/cs2/379/http:zSzzSzeyaloren.orgzSzpubszSzsemwiki2006.pdf/oren06annotation.pdf)

10.3 Semantic Instant Messenger

The third alternative considered is a semantic based variant of CM. As designed CM has weak message classification. There are no facilities to search for messages on a particular subject for example or from a user with particular characteristics. Even looking at messages by their subject line can give misleading results because often the subject line has little to do with the actual discourse.

10.3.1 The Problem

Instant messages typically are missing context are rather short, and informal therefore become meaningless without context. Topic switching and interleaving messages are particularities of IM conversation therefore to search for something specific one may have to process some that is general or even irreverent to the subject of the query.

IM messages usually lack semantics Current IM clients do not identify message properties, e.g. the creation date, or sender of a message. Consequently, relations between them cannot be exploited.

10.3.2 The solution

SAM: The Semantics Aware Instant Messaging for the Networked Semantic Desktop has been developed to address some of these problems. SAM addresses the weaknesses in message classifications by allowing a user-definable taxonomy that is used to add semantics to messages by annotating them with entries from the taxonomy. Users can tag a message for classification.

A drawback of this message classification is that the user may have to some effort to annotate messages appropriately. This effort may be lowered by semi-automatic annotation exchange between conversation partners. Additionally an intelligently designed user interface could mitigate this problem.

The ontological meta-model provides semantics for IM entities such as persons, messages, conversations, annotations, and message texts as it identifies and relates such entities to each other by meaningful properties such as message context and querying. Such an ontology is depicted in Figure 7.

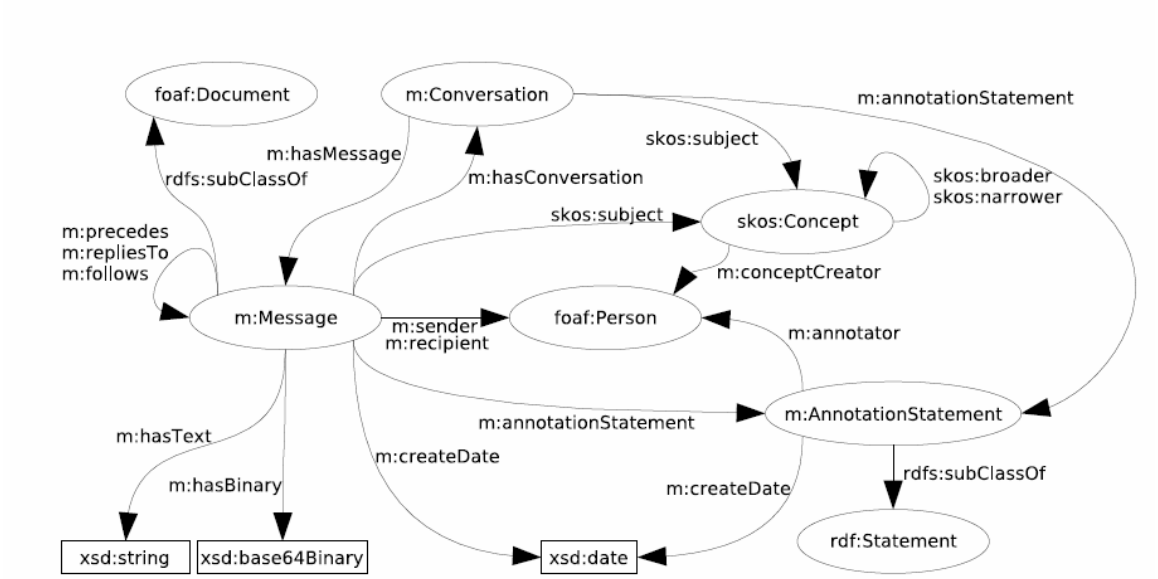

Figure 7 - SAM Ontology

Any message is accompanied by its context, i.e. messages link to their following message, their sender and recipient and so on. Accordingly, messages displayed while browsing or in search results are much more informative thus reducing the user effort of determining whether or not they are relevant.

Notice that the SAM Ontology depicted in Figure 7 makes use of the FOAF. The name 'FOAF' is derived from traditional internet usage, an acronym for 'Friend of a Friend'. FOAF has been designed to allow for integration of data across a variety of applications, Web sites and services, and software systems so should provide little integration problem which the correct incarnation of the prototype. It provides a basic "dictionary" of terms for talking about people and the things they make and do. The Simple Knowledge Organizational System (SKOS) initiative, also referred to in Figure 7 explores in more detail than FOAF the problem of describing topics, categories, "folksonomies" and subject hierarchies.[14] A C2 Ontology might extend this using OWL to express C2 specific constructs.

Semantic Querying: Querying becomes more powerful as the ontological metamodel permits to define what to query for, e.g. one can not only query for messages but also for users or taxonomy entries. Moreover, restrictions on properties can be defined. Each user decides how to annotate a message and which concepts to have in his taxonomy. The user interacts with the messenger in a manner similar to a regular IM except there is a taxonomy tool and ways to markup (annotate a message).

The communication protocol used by the system is Jabber based. It is called Extensible Messaging and Presence Protocol (XMPP) which is a XML-based protocol that is well supported by multiple open source programming libraries. A host of standard tools and

[14] http://xmlns.com/foaf/0.1/

languages were used ranging from OWL and RDF to Jena (a language often used with OWL).[15]

11 Conclusion

This paper has examined two research efforts to enhance collaboration in the C2 domain. A technical architecture and assessment is given for each. While the initial prototypes were very successful in demonstrating the viability of different technologies for C2 using frameworks that are quite different than those used today their evaluation suggests further areas of study. First the approach of each of prototype could be strengthened by the addition of tools to support semantically rich processing.

A summary of technology alternatives were presented for potential future evaluation. Three alternative approaches to adding those semantics were examined: Semantic Blogs, Semantic Wikis, and Semantic Messengers. Each relies on technology for annotating data and then being able to use those annotations to supply context. The approach that is most similar to the present CM architecture is the SAM approach.

Second, an experiment to quantify the difference between portal and non portal, between conventional and social based architectures and between rich and shallow semantics based software based architecture is suggested.

[15] SAM: Semantics Aware Instant Messaging for the Networked Semantic Desktop, Thomas Franz and Steffen Staab ISWeb, University of Koblenz-Landau, Germany, (http://www.semanticdesktop.org/xwiki/bin/download/Wiki/SAMSemanticsAwareInstantMessagingForTheNetworkedSemanticDesktop/11_franzstaab_sam_final.pdf)