



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	Semantically Interlinked Notification System for Ubiquitous Presence Management
Author(s)	Mehmood, Qaiser; Ali, Muhammad Intizar; Mileo, Alessandra
Publication Date	2013
Publication Information	Qaiser Mehmood, Muhammad Intizar Ali, Ollie Fagan, Owen Friel, Alessandra Mileo (2013) Semantically Interlinked Notification System for Ubiquitous Presence Management ODBASE 2013 Graz, Austria,
Publisher	Springer
Link to publisher's version	http://dx.doi.org/10.1007/978-3-642-37401-2_7
Item record	http://hdl.handle.net/10379/4127
DOI	http://dx.doi.org/978-3-642-41030-7

Downloaded 2024-04-25T14:03:51Z

Some rights reserved. For more information, please see the item record link above.



Semantically Interlinked Notification System for Ubiquitous Presence Management ^{*}

Qaiser Mehmood¹, Muhammad Intizar Ali¹, Ollie Fagan², Owen Friel², and
Alessandra Mileo¹

¹ DERI, National University of Ireland, Galway, Ireland
{qaiser.mehmood, ali.intizar, alessandra.mileo}@deri.org

² Cisco Systems, Galway, Ireland
{ofagan, ofriel}@cisco.com

Abstract. Presence based notification systems play a pivotal role in any collaborative working environment by providing near real time information about the status, locality and presence of the collaborators. Instant Messaging (IM) tools provide a simple low cost solution to support communication and collaboration in the working environment. With the wide adoption of smart phones, there is an increasing trend of using IM clients on mobile devices for exchanging and managing presence information. However, in modern organisations the use of collaborative tools is not limited only to IMs, but various other tools are also used, such as project management tools, meeting agenda, calendar and alike. In this paper we design, develop and implement an agile platform for ubiquitous presence management that relies on semantically interlinked vocabularies, existing information extraction tools for semantic enhancement of heterogeneous enterprise collaborative tools, continuous query processing for semantic integration and event detection in dynamic environment, and semantic transformation via query language for event-triggered notification actions based on the users context. We showcase our proposed solution by demonstrating a real world scenario for Cisco's Unified Presence System and we evaluate our solution in terms of requirements for Ubiquitous Presence Management System (UPMS) compared to related solutions, and overhead of each additional processing step, namely semantic transformation, continuous query processing and notification.

Keywords: Semantic Presence Management, Notification Systems, NFC, Instant Messaging, XMPP, XSPARQL, Stream Query Processing.

1 Introduction

Instant Messaging (IM) has been an effective way of communicating and collaborating in the modern enterprises [17]. Flexible and interoperable solutions for online IM-based communication systems rely on the standard XML-based

^{*} This work is jointly funded by Science Foundation Ireland, Grant No. SFI/08/CE/11380 (Lion2) and Cisco Systems, Galway, Ireland.

Extensible Messaging and Presence Protocol (XMPP) protocol for message exchange [19].

Advent of the modern communication technologies and smart phones have facilitated IM users by providing communication facilities on-the-go, increasing the number of mobile IM users. According to the forecast from Juniper Research, mobile IM users will exceed 1.3 billion by 2016³. This phenomenon that sees IM at the centre of the communication infrastructure is also observed in enterprise solutions for unified communications, where IM represents a flexible and easy-to-use tool for communication, presence management and collaboration.

The more IM becomes popular as a collaborative tool, the higher the need for solutions that can enhance collaboration by seamlessly integrating and correlating IMs messaging and semantic presence with information from other collaborative tools (including calendars, emails and project management tools) with sensors for physically contextualised presence [3] (including sensors embedded in mobile devices and other localised sensors).

In previous works, we focused on the use of semantics to enhance presence management and facilitate integration and exchange of semantic presence information [2, 8]. Semantic enrichment in IM communication paves the way for ubiquitous presence management, because it provides a standard and interoperable way of interlinking and sharing heterogeneous presence knowledge.

However, in order to materialise the idea of Ubiquitous Presence Management System (UPMS), we need to design and implement an agile solution that can cater for the richness of physical contexts (location, ongoing meetings, available resources) and integrate these physical contexts with other potential sources of contextualised presence such as calendars and other collaborative tools [9]. Physical contexts are not only rich in terms of variety of the information produced, but they also have an intrinsic dynamic nature we need to cater for. While variety is addressed by using semantic technologies for data integration and linking, we tackle the contextual dynamicity by resorting to the paradigm of event-based notification systems: dynamic knowledge from physical and collaborative contexts is interlinked and captured as events, which trigger appropriate actions or notifications.

To facilitate a solution with high interoperability and flexibility, we need to rely on existing standards for IM communication and semantic enrichment. In other words, we need to be able to manipulate and use XMPP (XML-based) and RDF to capture, encode transform and generate IMs, detect semantic events and generate notification actions [9, 21, 4]. We enable these capabilities by relying on a complete mapping of XMPP constructs into RDF [7], a SPARQL-based continuous query processing engine for semantic event detection and aggregation [15, 14], and a semantic query and transformation language that can deal with XML and RDF data (XSPARQL [1]).

The proposed UPMS architecture combines and applies the existing solutions for semantic information extraction from collaborative tools, continuous query processing and data transformation in the area of IM communication and based

³ <http://www.juniperresearch.com/viewpressrelease.php?id=311&pr=248>

on the standard XMPP protocol, resulting in a flexible and interoperable solution for Ubiquitous Presence Management.

Our main contributions in this paper can be summarised as follows: (i) we identify the requirements for UPMS, (ii) we semantically enhance XMPP messages based on standard vocabularies for sensor data and personal information management, (iii) we design and implement an agile platform for ubiquitous presence management that can detect semantically relevant event and trigger actions over the standard XMPP communication channel solely by using SPARQL-like queries in both event detection (CQELS) and data transformation (XSPARQL), (iv) we implement and evaluate the requirements and the performances of our solution by using NFC-enabled smartphones for physical presence and calendar data for collaborative contextual presence.

Structure of the paper: We identify the requirements for UPMS and emphasise on the motivation of our research by describing a general usecase scenario in Section 2. We present the conceptual design of our Ubiquitous Presence Management System in Section 3. Section 4, we demonstrate implementation and evaluation of a particular instance of the overall usecase scenario of Section 2. We position and advocate the importance of our work by comparing it with the existing work in Section 6 and discuss further steps and open challenges in Section 7.

2 System Requirements and Scenario

In order to achieve the general idea of semantically enhanced Ubiquitous Presence Management System, we identify a list of requirements, classified into three broader categories: user experience, interoperability and knowledge discovery. These requirements provide a guideline for the evaluation of performance, functionalities and potential impact of semantically enhanced UPMS.

User Experience: Mobile devices are now being widely used to provide and update the presence over the IM clients. Despite the rapid increase in performance of the mobile devices, user experience may suffer from applications designed to offer poor personalisation or unacceptable latency. User experience is thus strictly related to context awareness (or personalisation) and performances.

- **Context Awareness:** Presence-based services for mobile devices require regular updates because of the frequent change in the user’s context. It is very problematic and cumbersome for the IM client users to regularly update the presence status after each and every change in their context. UPMS should have the ability to automatically detect user’s context from diverse presence sources, and automatically update the presence status accordingly.
- **Performance:** Performance is a critical feature for UPMS relying on instant messaging clients because instant messaging applications are expected to perform within minimal time latency of realtime or near realtime, and the management of rich presence sources can add to this by making latency unacceptable for a positive user experience.

Interoperability: Interoperability is among one of the essential features for UPMS to be able to deal with incremental presence sources and to be applied to different scenarios. Different collaborative tools are designed independently to serve a specific purpose, however providing interoperability among these heterogeneous and autonomous tools is a challenging task.

- **Heterogeneity:** Presence management systems should not only have to deal with heterogeneous communication protocols e.g. XMPP, SIP etc., but should also be capable to provide an integrated view over heterogeneous data formats.
- **Extensibility:** Rapid advancement in mobile technologies is resulting in greater interest for the design and development of new mobile applications for presence management. An ideal presence management system should be extensible to easily incorporate any new feature or support new types of collaborative tool.
- **Adaptability:** Adaptability refers to the ability of the system to adapt the changes in its environment without any or with minimal external involvement, which is an important non-functional requirement of UPMS.
- **Domain Independence:** Most of the existing presence management systems are aimed to address the problems related to any specific domain. Domain independence of the semantic presence management system will facilitate its reusability and utilisation in any domain.

Knowledge Discovery, Sharing and Linking: Semantic technologies provide a systematic way to interrelate heterogeneous information from diverse sources for better discovery. Different ontologies are defined to semantically annotate and interlink the data from various domains and ease data sharing and reuse.

- **Event Management and Notification:** Event management allows the system to identify relevant pieces of knowledge (events) and related actions to be performed when such events occur. UPMS should facilitate its users in the registration of any event and specification of a set of actions to be performed on the occurrence of that particular event.
- **Recommendation:** *Mobile Recommender Systems* offer personalised, context-aware recommendation for the mobile device user by considering various parameters associated with the mobile device users. UPMS should be able to provide better (presence-aware) recommendations by combining text analysis and information extractions with semantics, e.g. for expert finding).

Ubiquitous Presence Management Scenario. Our company SmithCom is a very dynamic multinational corporation that has to deal with a many customers across the globe developing IT solutions and providing targeted customer support. The employees of the company are involved in a many meetings

involving technical documentation, contracts, sales, IT support, financial planning and more. To optimise communication, collaboration and resource management, SmithCom had put in place an infrastructure that supports Ubiquitous Presence Management.

For a final review of the products beta release Alice organises a virtual meeting with Sean and the German lead developer Hans. Alice uses her presence-enhanced meeting scheduler to create the meeting, specify the list of attendants, set meeting priority, preferred time and meeting duration along with resources needed for the meeting, while the scheduler suggests a meeting slot according to current participants calendars. The UPMS checks for available resources and sends an email to book them for the meeting. If the required resources are not available, alternative meeting times are provided to Alice according to her availability and Sean and Hans calendars.

Ten minutes before the meeting starts, a reminder is generated for the attendees. The UPMS detects that Hans is not at his laptop, so he receives an SMS on his smartphone on his way to the office. In a similar way, Sean presence indicates he is in a priority telephone conference for another half an hour, so he receives the same reminder by email. Alice, as a meeting host, receives a summarisation of relevant documents, reports and recent communications associated with the upcoming meeting. Based on her preferences and semantic presence, a list of pointers to relevant documents is provided through the IM client.

When the meeting starts, the UPMS detects that Hans is on his way to the meeting room, and informs Sean and Alice that he will arrive in five minutes. During the meeting, one of Seans team members sends him an IM. According to Seans policy, based on Sean's users groups, the profiles of the people Sean is meeting with and his preferences, the IM system shows him as busy for his team members and the message is blocked until the end of the meeting.

During the meeting, Alice's secretary Claire receives a request by the CEO for a meeting. UPMS determines that Alice doesn't have her phone with her, since her physical location is different from her phones location, but she is close to her laptop, so the UPMS produces a non-intrusive alert on it and the CEO is provided with an estimated time for a call and a time slot is booked into his calendar.

Later, Claire needs Alice to sign some papers but the UPMS shows her busy in her office despite no meeting scheduled. The reason for this is that a sales representative together with a new customer dropped by Alices office, and the UPMS can detect the two employees in the room via their tags and that a conversation is going on among a set of people (sensors can measure the level of noise without transmitting the conversation), therefore she instructs the presence service to notify her when Alice is available.

3 Ubiquitous Presence Management System

In this section, we give an overview of the conceptual design of our proposed ubiquitous presence management system (UPMS). UPMS is capable of manag-

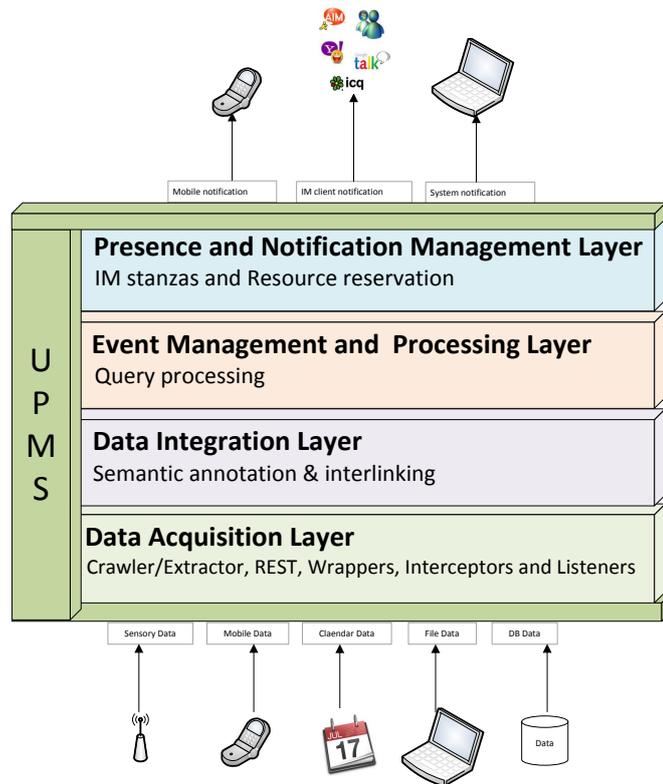


Fig. 1. Ubiquitous Presence Management System

ing and maintaining unified presence after gathering information from various data sources and sensors. It can also generate near realtime notifications which are triggered on the occurrence of any specific event. Figure 1 depicts a simplified architecture of the UPMS, demonstrating the inputs and outputs of the system. UPMS has ability to gather information from various collaborative tools, sensors, instant messaging clients, email clients or other collaborative tools which expose an interface to access their data. Once the information is acquired, it is semantically interlinked using various ontologies and live queries can be executed over the integrated data to monitor the occurrence of any event, and on the occurrence of that particular event, it can trigger various actions and notifications. UPMS consists of four processing layers, namely; (i) *Data Acquisition Layer*, (ii) *Data Integration Layer*, (iii) *Event Management and Processing Layer*, and (iv) *Presence and Notification Management Layer*. We discuss each of these layers and their underlying functionalities in detail below:

3.1 Data Acquisition Layer

The *Data Acquisition Layer* receives data as input to the UPMS. Data can be structured, unstructured and even raw data from the sensors. Data acquisition layer has to cope with heterogeneous data formats and data transmission protocols. UPMS in its current state provides four different mechanisms to receive data from multiple data sources,

- (i) **Data Listeners:** Data Listeners continuously listen, intercept and filter the data from a communication channel and pick out the required information.
- (ii) **RESTful Client:** Most of the collaborative tools including Google Calendar expose RESTful interfaces as a mean of communication with their clients. UPMS provides a RESTful client to collect data from the collaborative softwares and tools which support RESTful services.
- (iii) **Crawlers & Extractors:** UPMS can acquire data from various sources including desktop integration applications and email services using crawlers and extractors.
- (iv) **Data Mappers:** Data received from sensors is usually in raw format, UPMS contains data mappers to transform the sensors data into the desired format. In our usecase scenario, UPMS contains a mapper to map presence and geo-location into semantically interlinked data.

3.2 Data Integration Layer

UPMS relies on Semantic technologies to provide integrated access over heterogeneous data acquired from multiple sources. In our usecase, we semantically annotate the data from each source using predefined ontologies which provide a mechanism to semantically enrich the data of any specific source. SIOC (<http://rdfs.org/sioc/ns#>) and SIOCC (<http://rdfs.org/sioc/chat#>) provide semantic enrichment of the data related to the online presence, user accounts, instant messages, chat sessions and chat topics, which makes them ideal to semantically annotate XMPP messages received from the IM client. NCAL (<http://www.semanticdesktop.org/ontologies/ncal>) semantically annotates and interlinks meetings data described in calendars. SPITFIRE (<http://spitfire-project.eu/ontology/ns/>) allows to interlink and map various sensors data e.g geo-location. Once data is semantically enriched, the resulted RDF data is stored into an internal RDF data store within the UPMS. The data from this store can be queried using any specific query language defined to access linked data.

3.3 Event Management and Data Processing Layer

An event, which is often defined as "significant change in the state", can be registered at the event management component of the UPMS. Once an event is registered at UPMS, a continuous query is performed over the integrated data

to check the occurrence of the event and as soon as the event is triggered, the event management component sends request to the presence and notification management component to perform the actions as defined for that particular event.

At the heart of the UPMS is a continuous query processor namely CQELS query engine [14], which queries live stream of data and integrated linked data. An event can be registered at UPMS by providing a CQELS query, which continuously monitors the occurrence of the event. The CQELS query filters live stream of data coming from various sources and combines it with the static interlinked data to find the desired query patterns. Upon the occurrence of the event, results of the query are generated and passed to *Presence and Notification Management Layer*, which is responsible to perform the actions defined in the event.

3.4 Presence and Notification Management Layer

Presence and Notification Management Layer is responsible to perform all the actions after the occurrence of the event. Results of the CQELS queries are passed by the *Event Management and Processing Layer* to the *Presence and Notification Management Layer* in the form of RDF triples. The presence and notification manager transforms the RDF triples into the required data format to disseminate the notifications. In our usecase scenario, we use RDF to XMPP mapping to generate notification messages (XMPP stanzas) for IM clients. Presence and notification manager is also capable of interacting with various tools and their underlying network protocols for the communication.

4 Implementation

As a proof of concept, we have implemented a usecase scenario of Section 2 using Cisco's Unified Presence System (CUP). CUP server is an XMPP based communication system to provide better means of communication within an enterprise [16]. In this section, we describe the NFC based usecase scenario for SmitCom and give a detailed description of UPMS implementation for this scenario.

4.1 Usecase Scenario

In order to facilitate localisation and presence management, employees of Smith-Com are equipped with NFC-enabled smartphones running an NFC-enabled IM client, and meeting rooms are identified by a unique NFC tag. Resources are managed using IM messages for booking them in particular time slots for particular events.

The sales manager Alice schedules a meeting using *Gcal* in one week time, in room *James Joyce*. The meeting involves Bob, Charlie and Dave and needs a projector and a white board, which are available and reserved by the UPMS.

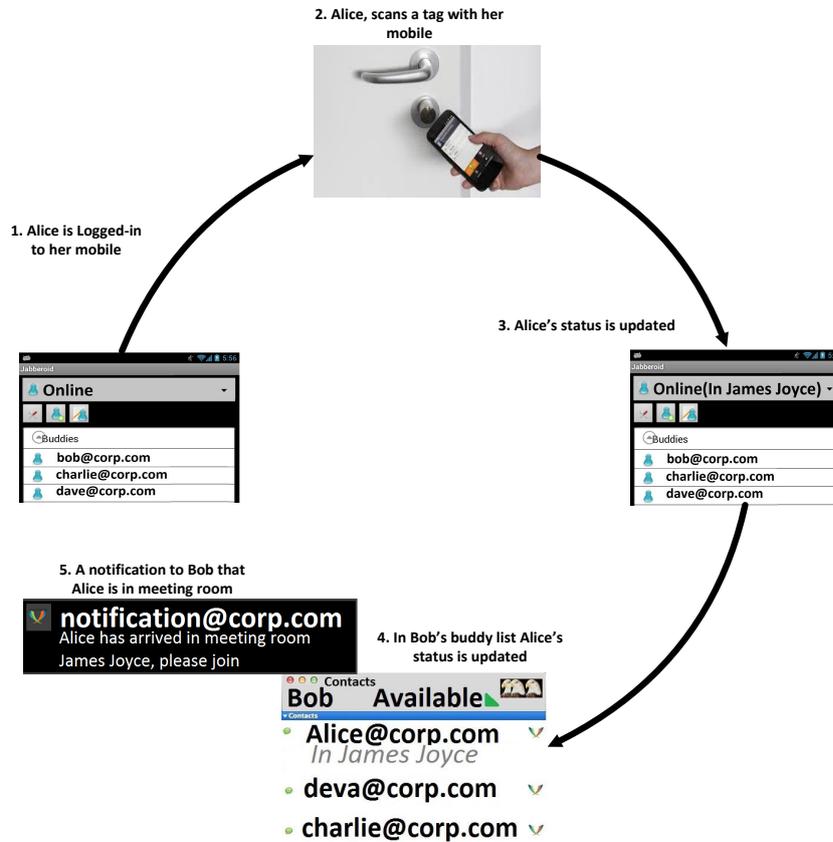


Fig. 2. Usecase Scenario

Alice enters the meeting room *James Joyce*, fifteen minutes before the meeting, and she scans her mobile device against the wall-mounted NFC tag in the room. Her status on the IM client is automatically updated and made visible to her buddy list according to her privacy preferences. Being Alice the host of the meeting, she gets an IM message with the agenda for the meeting and a list of links to IM chats and emails related to the meeting, while the list of attendees (Bob, Charlie and Dave) receive an IM message advising that the meeting host has arrived and they should join the meeting in *James Joyce* room. The meeting can start perfectly on time with very little communication effort between the employees. Figure 2 depicts the proof of concept scenario described above.

4.2 System Architecture

Figure 3 represents the system architecture of the implemented usecase scenario, which depicts the residing components encapsulated by UPMS layer architecture

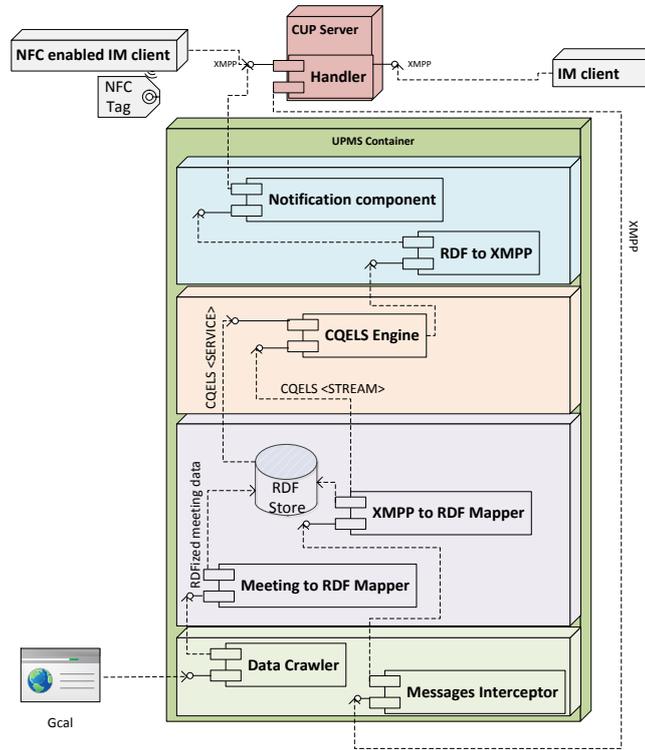


Fig. 3. System Architecture (Cisco’s Unified Presence System for SmithCom)

as shown previously in Figure 1. UPMS has the flexibility to interact with multiple collaborative tools for data acquisition as well as generate the notifications to the respective targeted systems. In this prototype, we have demonstrated message interceptor for the XMPP stanzas sent from/to the CUP server, a meeting data crawler/extractor from Google Calendar (*Gcal*) for data acquisition, and a data mapper and notification dispatcher to disseminate notifications.

4.3 Process Flow

In this section the application process flow is explained describing all the steps involved to successfully execute the SmithCom usecase scenario as described previously. Each step is presented with a detailed description of the technologies involved.

Calendar Data Acquisition: Alice, a user is using a *Gcal* to schedule meetings. She is a registered user with $\langle \text{alice}@gmail.com \rangle$ account for *Gcal*. When

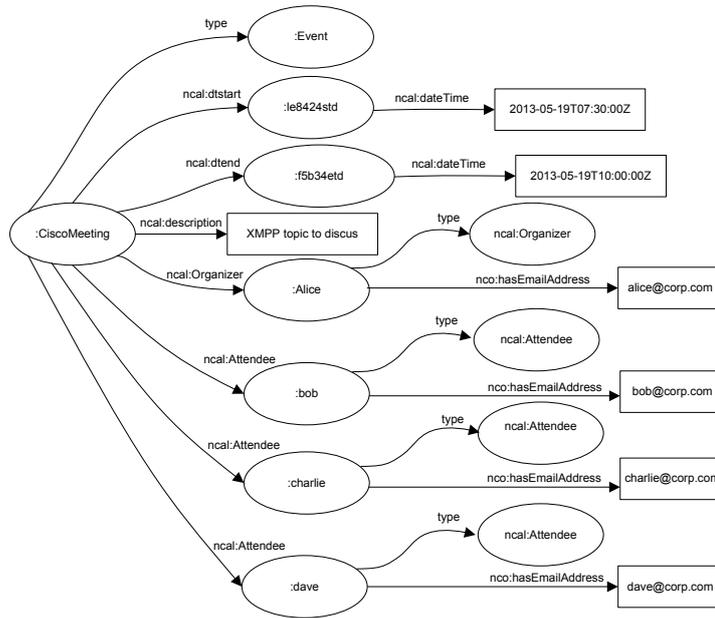


Fig. 4. Calendar Data Mapping

Alice, as a host, creates a meeting, the RESTful client crawls the meeting data from *Gcal*.

Calendar Data Mapping: *Data Acquisition Layer* transfers the extracted data to the *Data Integration Layer*, where a *Meeting to RDF Mapper* transforms the meeting data into linked data by employing two ontologies defined for semantic desktop integration: (i) NCAL (<http://www.semanticdesktop.org/ontologies/ncal/>) to represent event extracted from the calendar, their start/end date and time, and participants in the event, (ii) NCO (<http://www.semanticdesktop.org/ontologies/nco/>), to represent persons and their e-mails addresses. Semantically interlinked data is stored in an RDF data store, which resides in *Data Integration Layer* of the UPMS. Figure 4 shows a semantic representation of a *Gcal* data for a meeting organised by Alice.

NFC-enabled IM Client: Our custom built NFC-enabled IM client has the ability to detect NFC tags. When a tag containing data in a supported format is tapped, the IM client acquires the data from NFC tag. The IM client generates two XMPP stanzas: (i) a *presence* stanza to update the user's status as acquired from NFC tag, and (ii) an *iq* stanza containing geo-location information acquired from the NFC tag. These stanzas fully comply with the existing standards defined in XMPP core and its XEP extension [19, 10].

```

prefix ncal: <http://www.semanticdesktop.org/ontologies/
2007/04/02/ncal#>
prefix nco: <http://www.semanticdesktop.org/ontologies/
2007/03/22/nco#>

SELECT ?attEmail ?starttime
WHERE {
    STREAM <http://deri.org/streams/lsm> [NOW]
    {
        <http://smithcominternal.com/example#aliceSmack>
        <http://www.semanticdesktop.org/ontologies/2007/03/22/nco#imID>
        ?host.
    }

    SERVICE <http://lsm.deri.ie/sparql>
    {
        GRAPH <http://lsm.deri.ie/cisco/eventdata#>
        {
            ?E ncal:organizer ?organizer.
            ?organizer nco:hasEmailAddress ?hasemail.
            ?hasemail nco:emailAddress ?host.
            ?E ncal:attendee ?attendee.
            ?attendee nco:hasEmailAddress ?email.
            ?email nco:emailAddress ?attEmail.
            ?E ncal:dtstart ?start.
            ?start ncal:dateTime ?starttime.
            Filter(str(?starttime)="2013-05-19T07:30:00Z")
        }
    }
}

```

Listing 1. A Sample CQELS Query

XMPP Interceptor: In our usecase scenario, UPMS is implemented as an extension of the Cisco’s CUP server, which is responsible for the information exchange between IM clients. *Message Interceptor* is responsible for intercepting and filtering the XMPP stanzas sent to/from the CUP server. The filtered XMPP stanzas are then passed on to *Data Integration Layer* of the UPMS for further processing.

XMPP to RDF Mapping: *XMPP to RDF Mapper* performs the mapping of the received XMPP stanzas using various ontologies: (i) OPO (<http://online-presence.net/opo/spec/>) to represent online presence information, (ii) SIOC (<http://rdfs.org/sioc/spec/>) and SIOCC (<http://rdfs.org/sioc/chat#>) to represent instant messages and chat messages, (iii) NCO (<http://www.semanticdesktop.org/ontologies/2007/03/22/nco#>) to represent user accounts, (iv) SPITFIRE (<http://spitfire-project.eu/ontology/ns#>) to represent information received from sensors, and (v) wgs84_pos (www.w3.org/2003/01/geo/wgs84_pos#) to represent geo-location information contained within XMPP messages. Figure 5 shows a pictorial representation of the mapping of *presence* stanza while Figure 6 shows mapping of *iq* stanza. XSPARQL (a query language uses to transform the XML into RDF and vice versa) is used to transform the live stream of XMPP messages into RDF on the fly.

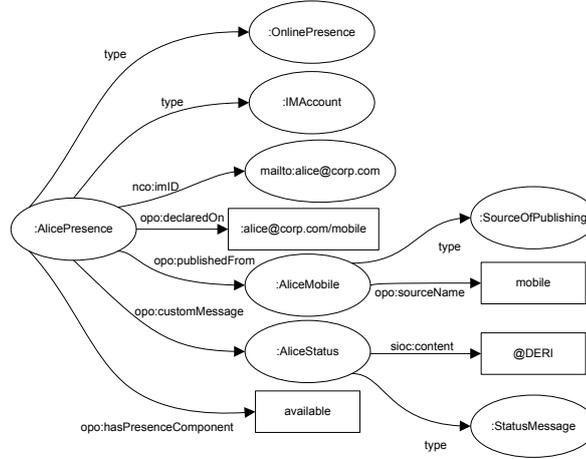


Fig. 5. Presence Stanza Mapping

Integrated Data Processing: CQELS queries perform integrated data processing by performing combined query over the live streams of mapped stanzas and integrated meeting data. UPMS in its current state expects from its users to provide a CQELS query to monitor the occurrence of any event. However, providing an interactive user interface for the event registration and generating CQELS queries automatically is part of our future agenda. Listing 1 shows a CQELS query which monitors the occurrence of our usecase scenario event. CQELS query contains two clauses, (i) STREAM clause filters XMPP messages to find out host of the meeting and (ii) SERVICE clause, queries the integrated data to look for a particular meeting organised by host, meeting start time and list of attendees.

Presence and Notifications Management: CQELS queries produce results on the occurrence of the event, which are processed at *Presence and Notification Management Layer* to achieved the desired outcomes as described in the event. In our usecase scenario, we used XSPARQL to transform the results of CQELS queries into XMPP messages, which are disseminated to all attendees of the meeting.

5 Evaluation

We evaluated the performance of our system using two approaches: (i) by comparing our system with the state of the art presence management systems, and (ii) by evaluating the performance of our system with the increasing number of users (meeting attendees).

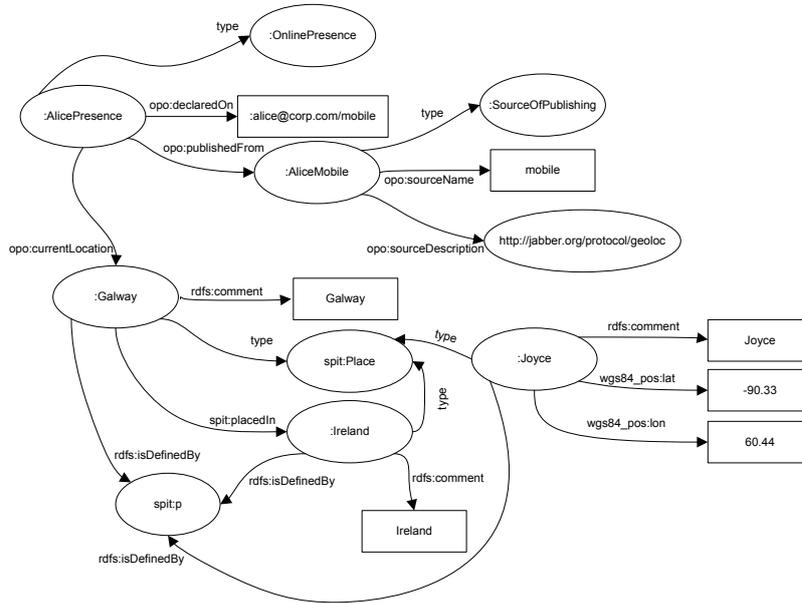


Fig. 6. IQ Stanza Mapping

5.1 Comparison

We used system requirements described in Section 2 as a guideline for the evaluation of functionalities provided by the UPMS. In Table 1, we compare our system with three state of the art presence management systems, namely (i) *Loca-Tag* (ii) *R-U-In?*, and (iii) *Follow Me!*. *Loca-Tag* is aimed to provide automatic update in the presence of the IM client's (Skype) user, reflecting the current location of the user observed using NFC tag [13]. *Loca-Tag* provides rich user experience but lacks interoperability and knowledge discovery features. *R-U-In?* leverages contextual information to provide better "social search" for heterogeneous social networking tools including IM [4]. However, it is domain dependent on social networks and also does not support event management and adoptability. *Follow me!* provides rich presence information by consolidating virtual and physical presence using contextual information [3]. Similar to the previous two presence management applications, *Follow me!* is also limited to provide rich user experience while falling short in providing interoperability and knowledge discovery. Contrary to the previously discussed systems, UPMS is not only aimed on providing rich user experience but it is also equipped with interoperability and knowledge discovery.

Table 1. System Requirements Comparison

Category	System Req.	UPMS	Loca-Tag	R-U-In?	Follow Me!
User Experience	Performance	✓	✓	✓	✓
	Context Awareness	✓	✓	✓	✓
Interoperability	Adaptability	✓	✗	✗	✗
	Extensibility	✓	✗	✗	✗
	Heterogeneity	✓	✗	✓	✗
	Domain Independence	✓	✗	✗	✗
Knowledge Discovery	Event Management	✓	✗	✗	✗
	Recommendations	✓	✗	✓	✗

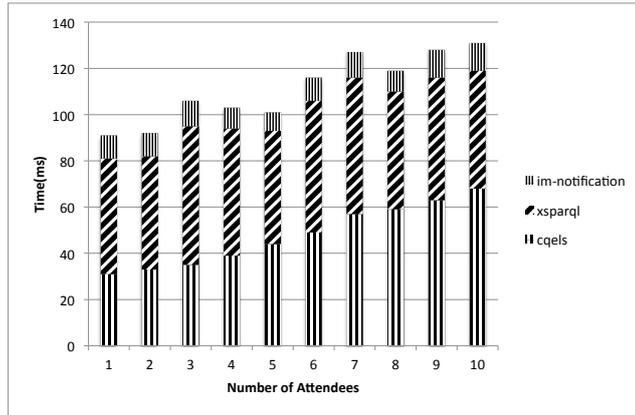


Fig. 7. Processing time in proportion to the number of attendees

5.2 Performance Evaluation

In order to evaluate the performance of our system we setup the testbed to implement the usecase scenario of Section 4. The processing time of the application is subdivided as (i) *XMPP to RDF Mapping Time* (XSPARQL), (ii) *CQELS Processing Time*, (iii) *RDF to XMPP Mapping Time* (XSPARQL), and (iv) *Notification Dissemination Time*. We vary the number of attendees of the meeting from 1 to 10 and calculated the average processing time after 20 observations. Figure 7 shows the average processing time of the notification dissemination to all the attendees of a particular meeting generated after the arrival of the meeting host. We omit *XMPP to RDF Mapping Time* in Figure 7, because it was evident that *XMPP to RDF Mapping Time* is dependent on the size of the presence and iq stanza irrespective of the number of attendees. Average time of the 20 observations for *XMPP to RDF Mapping Time* for *presence* stanza and *iq* stanza was 136ms and 163ms respectively. As shown in Figure 7, *CQELS Processing Time* and *RDF to XMPP Mapping Time* increases with the increase in number of attendees. However, considering the existing time latency, UPMS

is scalable to disseminate notification up to 100 attendees within a delay of less than a second. It is also worth mentioning that *Notification Dissemination Time* is also increasing with the number of attendees, however not very substantially due to better performance of the CUP server.

6 Related Work

In the last few years, NFC based applications are gaining popularity and many applications have been developed and implemented using NFC technology. A generalised approach for the design and development of NFC applications have been described in [6, 5]. The state of the art and future directions for the NFC applications have been discussed in [18]. However, currently most of the existing applications provide domain specific and small scale solutions e.g mobile payment systems, e-ticketing systems or healthcare systems [22, 12, 20]. Loca-Tag enhances instant messaging tools to provide near real time location based presence sharing through the use of NFC enabled mobile devices [13]. A prototype developed in Loca-Tag sends the presence information for Skype⁴ using a Web service. The trend of presence systems, which were developed earlier to serve presence sharing or instant messaging, have been shifted to intelligent presence management in a collaborative environment [11].

In [9], a holistic definition of presence, which they call "Consolidated Presence", has been presented. Consolidated presence is achieved by deploying richer semantic services to combine several presence information from various heterogeneous and dynamic sources. In order to receive richer semantic presence in Cisco's CUP server, a mapping of the core XMPP messages into RDF is formally defined in [7]. All XMPP messages are mapped into RDF using various ontologies e.g. (i) SIOC (<http://rdfs.org/sioc/ns#>) represents user account, IM messages and threads, topics and chat channels, (ii) OPO (<http://online-presence.net/opo/spec/>) represents dynamic aspects of user's presence, (iii) NCAL (<http://www.semanticdesktop.org/ontologies/ncal/>) semantically annotates calendar events, their start and end date, topic, host name, and list of participants, (iv) NCO (<http://www.semanticdesktop.org/ontologies/nco/>) represents known persons and their email addresses, and (v) SPITFIRE (<http://spitfire-project.eu/ontology/ns/>) semantically represents sensor network and its surrounding environment.

However, to the best of our knowledge, there is no existing system which can utilise NFC-enabled mobile devices to update location based presence information for the IM clients and can also integrate the information from various heterogeneous collaborative tools to provide a complete semantically interlinked notification system for ubiquitous presence management.

⁴ www.skype.com

7 Conclusion & Future Work

In this paper, we have proposed, deployed and evaluated a semantically enriched solution for Ubiquitous Presence Management. This work is pioneering next generation communication platforms for the Smart Enterprise, and it is based on i) semantic enrichment of XMPP stanzas to provide a holistic view of heterogeneous presence sources (including sensor data) and knowledge from other collaborative tools, ii) continuous event detection based on CQELS semantic query processing, iii) query-driven data integration and transformation from XML to RDF and back via XSPARQL, and iv) query-driven generation of notifications triggered by appropriate events. We showcase our prototype using NFC-enabled devices and calendar events to dynamically detect meeting status and progress, and notify attendees accordingly, and we present an evaluation based on the implemented proof of concept.

Next steps for the realisation of a more comprehensive Ubiquitous Presence Management System will include additional collaborative tools and learn user profiles that can be used to suggest interesting events and to configure a personal Ubiquitous Presence Management Dashboard. Leveraging Linked Data, in particular from Social Networks and DBpedia is also an interesting direction for automatic context characterisation. These aspects together with the characterisation of the interplay between physical and collaborative contexts will be investigated in future steps.

References

1. W. Akhtar, J. Kopecký, T. Krennwallner, and A. Polleres. XSPARQL: Traveling between the XML and RDF Worlds - and Avoiding the XSLT Pilgrimage. In *Proc. of ESWC 2008*, pages 432–447, 2008.
2. M. I. Ali, N. Lopes, O. Friel, and A. Mileo. Update Semantics for Interoperability Among XML, RDF and RDB. In *Proc. of APWeb*, 2013.
3. A. Antonic, I. Slivar, and I. P. Zarko. Follow me! A rich presence application for smartphones. In *Software, Telecommunications and Computer Networks (SoftCOM), 2012 20th International Conference on*, pages 1–5. IEEE, 2012.
4. N. Banerjee, D. Chakraborty, K. Dasgupta, S. Mittal, S. Nagar, et al. Ru-in?-exploiting rich presence and converged communications for next-generation activity-oriented social networking. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 222–231. IEEE, 2009.
5. B. Benyó, B. Sódor, G. Frdos, L. Kovács, and A. Vilmos. A generalized approach for NFC application development. In *Proc. of the Second International Workshop on Near Field Communication (NFC), 2010*, pages 45–50. IEEE, 2010.
6. B. Benyo, A. Vilmos, K. Kovacs, and L. Kutor. The Design of NFC Based Applications. In *Proc. of 11th International Conference on Intelligent Engineering Systems, 2007*, pages 277–280, 2007.
7. M. Dabrowski, S. Scerri, I. Rivera, and M. Leggieri. Dx- Initial Mappings for the Semantic Presence Based Ontology Definition, Nov. 2012. <http://www.deri.ie/publications/technical-reports/>.

8. A. Dumitrache, A. Mileo, A. Zimmermann, A. Polleres, P. Obermeier, and O. Friel. Enabling privacy-preserving semantic presence in instant messaging systems. In *Modeling and Using Context*, pages 82–96. Springer, 2011.
9. M. Hauswirth, J. Euzenat, O. Friel, K. Griffin, P. Hession, B. Jennings, T. Groza, S. Handschuh, I. P. Zarko, A. Polleres, and A. Zimmermann. Towards Consolidated Presence. In *Proc. of CollaborateCom 2010*, pages 1–10, 2010.
10. J. Hildebrand and P. Saint-Andre. Xep-0080: User location. *XMPP Standards Foundation*, 2007.
11. W. IJsselsteijn, J. van Baren, and F. van Lanen. Staying in touch: Social presence and connectedness through synchronous and asynchronous communication media. *Human-Computer Interaction: Theory and Practice (Part II)*, 2:924–928, 2003.
12. A. Juntunen, S. Luukkainen, and V. K. Tuunainen. Deploying NFC Technology for Mobile Ticketing Services—Identification of Critical Business Model Issues. In *Proc. of Ninth International Conference on Mobile Business, 2010*, pages 82–90. IEEE, 2010.
13. F. Kbler, P. Koene, H. Krcmar, M. Altmann, and J. Leimeister. LocaTag - An NFC-Based System Enhancing Instant Messaging Tools with Real-Time User Location. In *Second International Workshop on Near Field Communication (NFC), 2010*, pages 57–61. IEEE, 2010.
14. D. Le-Phuoc, J. X. Parreira, M. Hausenblas, and M. Hauswirth. Continuous query optimization and evaluation over unified linked stream data and linked open data. Technical report, Technical Report DERI-TR-2010-09-27, DERI, IDA Business Park, Lower Dangan, Galway, Ireland, 2010.
15. D. Le-Phuoc, H. N. M. Quoc, J. X. Parreira, and M. Hauswirth. The linked sensor middleware—connecting the real world and the semantic web. *Proceedings of the Semantic Web Challenge*, 2011.
16. B. Morgan, S. Lisenbea, and M. Popovich. *Cisco Unified Presence Fundamentals*. Cisco Systems, 2010.
17. B. A. Nardi, S. Whittaker, and E. Bradner. Interaction and outeraction: instant messaging in action. In *Proc. of CSCW*, pages 79–88. ACM, 2000.
18. K. Ok, V. Coskun, M. N. Aydin, and B. Ozdenizci. Current benefits and future directions of NFC services. In *Education and Management Technology (ICEMT), 2010 International Conference on*, pages 334–338. IEEE, 2010.
19. P. Saint-Andre. Extensible messaging and presence protocol (xmpp): Core. 2011.
20. M. Vergara, P. Díaz-Hellín, J. Fontecha, R. Hervás, C. Sánchez-Barba, C. Fuentes, and J. Bravo. Mobile prescription: An NFC-based proposal for AAL. In *Proc. of Second International Workshop on Near Field Communication (NFC), 2010*, pages 27–32, 2010.
21. S. Voidsa, E. D. Mynatt, B. MacIntyre, and G. M. Corso. Integrating virtual and physical context to support knowledge workers. *Pervasive Computing, IEEE*, 99(3):73–79, 2002.
22. J. Zou, C. Zhang, C. Dong, C. Fan, and Z. Wen. Mobile payment based on RFID-SIM card. In *Proc. of IEEE 10th International Conference on Computer and Information Technology (CIT), 2010*, pages 2052–2054. IEEE, 2010.