Determining User Presence using Context in a Decentralized Unified Messaging System (IPAD-UMS)

Saguna
Caulfield School of Information Technology
Monash University
900 Dandenong Road,
Caulfield East, Victoria, Australia 3145
+61-3-99034144
Saguna.Saguna@infotech.monash.edu.au

Prem Prakash Jayaraman
Caulfield School of Information Technology
Monash University
900 Dandenong Road,
Caulfield East, Victoria, Australia 3145
+61-3-99031151
Prem.Jayaraman@infotech.monash.edu.au

Arkady Zaslavsky
Caulfield School of Information Technology
Monash University
900 Dandenong Road,
Caulfield East, Victoria, Australia 3145
+61-3-99032479
Arkady.Zaslavsky@infotech.monash.edu.au

1. INTRODUCTION

UMS can be described as a system that allows for communication between users via a number of communication technologies over heterogeneous network infrastructures[1]. It integrates networking technologies such as Ethernet, ATM, IEEE 802.11, GSM and UMTS alongwith various devices such as laptops, tablet PCs, desktops, mobile phones, smartphones, PDAs, fax machines and landline phones. In other words, it allows the users to access their different types of messages by logging into one inbox from these different devices[2]. The user is able to create, modify, access and administer messages belonging to various formats on a single device to which he/she has immediate access. This form of '*anywhere*', '*anytime*' access to '*anykind*' of messages enables the UMS to become ubiquitous and pervasive[3, 4].

There are several UMS available in the market[2] which incorporate some of the above mentioned communication technologies in different combinations. However, existing Unified Messaging Systems (UMS) are fraught with critical shortcomings and drawbacks which inhibit them to furnish a complete solution for satisfying all the communication needs of a user. Nearly all existing UMS are centralized, vendor-specific, organization oriented, expensive and highly complex to integrate. Thus, in this paper we present our novel UMS called the Intelligent Presence-Aware Decentralized Unified Messaging System (IPAD-UMS) that is decentralized, context-aware and addresses the problems prevalent in existing systems. It collects and processes a wide array of complex context information such as user and device location, message subject, message priority, user profile settings, etc. It also determines presence intelligently to deliver messages to a device closest to the user irrespective of time and location. Thus, IPAD-UMS is a novel system which is presence-aware, platform-independent, software based, easy-to-integrate, extremely low in cost and user-centric. Experimentation was conducted to test and validate IPAD-UMS capabilities.

2. CONTEXT-AWARE IPAD-UMS

Our proposed system deals with different types of messages such as e-mail, IM and SMS within the decentralized environment where an incoming message received on any of the devices is delivered to the device closest to the user based on context-awareness. The user can then reply to the incoming message from that device and the message is sent out in the same format in which it was originally received by our IPAD-UMS. Figure 1 shows our proposed architecture for the IPAD-UMS.

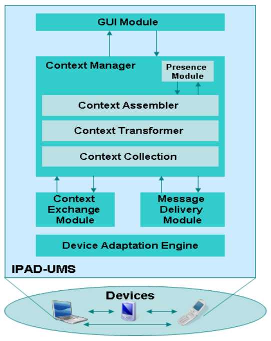

Fig. 1. IPAD-UMS Architecture.

The Context Manager (CM) makes intelligent decisions on behalf of the user for message delivery using context-awareness and by determining Presence from various sources in a novel way. The Device Adaptation Engine (DAE) works during the installation of the system on the device and installs the required components based on the available resources. The user interaction is done via the system GUI. The Message Delivery Module (MDM) and

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3991
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3991

the Context Exchange Module (CEM) are used to exchange messages and context information between the devices. The following sub-sections provide a detailed discussion of our IPAD-UMS architecture.

2.1 Context Manager

The architecture of the CM as shown in figure 2 is divided into several component blocks and follows a layered approach.

2.1.1 The Context Collector (Monitor Layer)

The Context Collector (CC) collects context information from the user and the device environment. It collects information such as: user and device location, calendar context, user profile context, presence information, etc. The collected set of context information is then presented to the Context Transformer (CT) for further processing.

2.1.2 The Context Transformer (Context Layer)

The Context Transformer (CT**)** gathers the collected data and converts it into meaningful information for our system. The CT contains the *Location Module (LM)*, the *Profile Module (ProM)* and the *Presence Module (PreM***)**. Each module uses the raw data that is relevant to it and converts it into useful information that can be used by the *Decision Module (DM)* present in Context Assembler (CA).

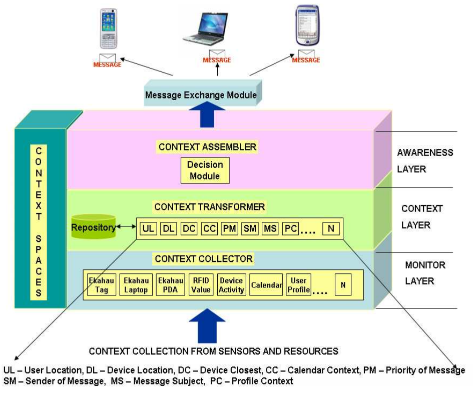

Fig. 2. The IPAD-UMS Context Manager: Layered approach.

2.1.2.1 Location Module

The Location Module (LM) is used to determine the most accurate location of the user and the device. In our case, the location of the user and device is determined via the Ekahau Tag, RFID tags and the calendar information. After collecting the raw information from the CC, the CT assigns *confidence levels* to each of these. These levels can be assigned differently in different situations and the process of assigning particular confidence levels can be justified accordingly. For example, in the case of finding user location, the information collected from Ekahau Tag was given the confidence level of 0.25. This was due to the fact that the correctness of Ekahau is limited up to 4-5m, which can sometimes mislead the CM. Similarly, our system can assign confidence levels for both calendar and RFID as well. The confidence levels from matching location sources are added to make a sum of 1. The added confidence levels for each location are then matched against the threshold. The final user location value thus, found is the most accurate and reliable. The location of devices (laptop, PDA and smartphone) are also determined in a similar way.

2.1.2.2 Profile Module

The Profile Module (ProM) is responsible for handling various profiles pertaining to the system such as sender profile and message profile. These profiles are used to categorize the senders/contacts and the subjects of incoming messages into various groups according to their importance level as shown in table 1 and 2. The most important senders/contacts are present in Group 1; this is followed by Group 2 and 3 respectively. The same approach is followed for subject groups. Using the User Profile GUI, the user can anytime move a particular sender/contact or a particular subject from one group of importance to another.

Table 1. Sender profile with priority values based on importance to user.

Sender Group	***Sender Group Members***	***Priority Value*** P_{sender}
Group 1	Arkady, Karan, Prem, Paul,… N	1
Group 2	Shubhi, Niti, Nakul, Mimi, … N	2
Group 3	Rob, Alice, Mark, Tim, … N	3
Group 4	Unknown Senders	4

Table 2. Subject profile with priority values based on importance to user.

Subject Group	***Subject Group Items***	***Priority Value*** $P_{subject}$
Group 1	Urgent, Meeting, Project … N	1
Group 2	Reminder, Deadline, …. N	2
Group 3	Lunch, Dinner, Movie, … N	3
Group 4	Un-listed/Un-registered Subjects	4

2.1.2.3 Presence Module

The Presence Module (PreM) handles the user's and his/her contacts presence. The presence of the user is determined either by the user or is set automatically by our intelligent CM. When the CM sets the presence, it is either based on user location or on user activity or both taken together. This is done in a similar way using confidence levels as the final location was determined in the LM. In our system, presence is divided into two levels of hierarchy. At the first level, presence of a user is divided into 4 basic types which are *Busy*, *Away*, *Available* and *Not Available*. At the second level, these basic presence types can be further divided into several states. For example, states such as *'In Room'*, '*Reading'*, *'surfing the internet'*, etc. Our proposed architecture can make use of several types of context information along with *"vague"* presence information to correctly determine the right presence status of the user.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3991
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3991

This is described in detail in the next section. Within the CT, lies the Repository (R), for storing all the relevant context data to be used by the relevant modules in our system. Once all the context information is transformed according to the requirements, it is then passed onto the Context Assembler (CA) for final processing.

2.1.3 The Context Assembler

The Context Assembler (CA) is one of the most important components of the CM. The Decision Module (DM) in the CA decides whether the message should be delivered at a particular time or not.

2.1.3.1 Decision Module

The DM decides whether an incoming new message should be delivered to the user or not depending on the context information. When a message arrives on a particular device, the decision is taken by the CM to deliver it to the most suitable device. If the CM decides to deliver the message, it is sent using proper routing mechanisms to the device closest to the user else the message waits in the R for the context to change for it to be delivered.

The DM first finds the device closest to the user. It then calculates the message priority ($P_{message}$) based on the sender of the message and the subject of the message. The sender and the subject belong to a certain group within the Sender Profile and the Subject Profile respectively as shown in tables 1 and 2. The priority values for the sender and the subject are added to achieve the priority of the message as shown in equation 1.

$$P_{message} = P_{sender} + P_{subject} \qquad (1)$$

The $P_{message}$ is then checked against the Message Priority Threshold ***(****t_p*) as shown in table 4 and the message is delivered accordingly, if $P_{message} <= t_p$. Thus, user presence, message priority and device closest to the user are integral to the message delivery decision. Once the decision is taken, the message is then passed to the Message Delivery Module (MDM) which delivers the message to the most appropriate device.

3. PROTOTYPE IMPLEMENTATION

Our proof-of-concept prototype implementation was done using the three devices, i.e. the laptop (A Toshiba laptop with Intel Pentium 4 CPU running on 3.02GHz, 1.12GB of RAM), the PDA (HP iPAQ h5550 PDA) and the smartphone (Nokia 6260). Each of these devices had the GUI, CM and MDM as shown in figure 3. The implementation was done using the different Java Virtual Machines (JDK1.5, Insignia Jeode and JVM for MID) and various APIs such as Swing, Java Mail, Smack API for GTalk BlucoveJSR82, AvetanaOBEX, EkahauSDK, etc. As soon as the system is started, the GUI allows the user to set their presence and profile. Meanwhile, the system also checks for the incoming messages and tracks the user and device location using the Ekahau Tag. If there is a new message on any of the devices, it is then delivered to the device closest to the user. For example, a new e-mail arrives on the laptop. The CM then determines the device closest to the user. If the closest device is discovered to be the smartphone, then the new message is delivered to the user on the smartphone by our system.

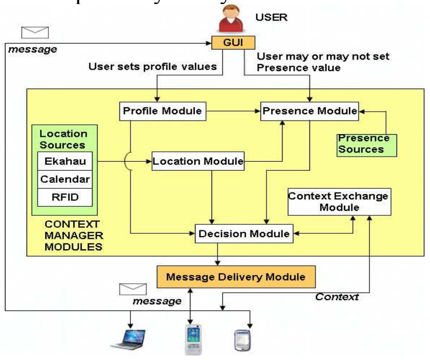

Fig. 3. IPAD-UMS implementation on the devices.

The GUI on that device alerts the user regarding the incoming message. The messages are delivered to the user based on the current user presence and other relevant context information such as the user location which is determined by our CM. The user can also reply to the new messages from the closest device. This message is then sent to the sender in its original format. Our devices are capable of using the Google Talk service (GTalk) for IM and determining the user's and his/her contacts presence information. Context information is exchanged between all the devices whenever any of the devices detects a change in any type of context.

4. EXPERIMENTATION AND RESULTS EVALUATION

We conducted several experiments for our system's evaluation. For the scope of this paper, we present the outcomes of five such experiments which clearly validate our system as shown in table 3. The experiments were conducted within the office and the meeting scenario. Also for experimentation, we considered context information related to incoming messages such as senders and subjects as shown in table 1 and 2 to determine message priority. Figure 4 shows different combinations of senders and subjects which can affect message priority $P_{message}$. For both senders and subjects, the group of highest importance was assigned the value, 1 and the group with least importance was assigned the value, 4. The combination of Group 1 senders with subjects from both Group 1 and Group 2 generated a message priority of **high (H)** ($P_{message} <= 3$), and the message was always delivered to the user regardless of his/her presence state. When Group 1 senders were combined with subjects from Group 3 and Group 4, the message priority was **medium (M)** ($P_{message} <= 5$) and the message was not delivered when user was *"busy"*, but it was delivered when he/she was *"away"* or *"available"*. The message priority can also be calculated as **low (L)** in

Table 3. Experimentation Results

Exp	UL	DC	UA	Presence: t_p	Sender:P_{sender} and Subject:$P_{subject}$	$P_{message}$	Delivered:Y/N
1	R8	Smartphone	Coffee in Lounge	Away : 5	Karan : 1 and Deadline : 1	2	$P_{message} < t_p$: Y
2	R5	Laptop	Discussion (Supervisor Room)	Busy : 3	Nakul : 2 and Invitation : 3	5	$P_{message} > t_p$: N
3	R6	Smartphone	Lunch (Kitchen)	Away : 5	Arkady : 1 and Meeting : 1	2	$P_{message} < t_p$: Y
4	R1	Laptop	In Room Reading Paper (User Room)	Available : 8	Mark : 3 and Dinner : 3	6	$P_{message} < t_p$: Y
5	R9	PDA	In Alley	Away : 5	Mimi : 2 and Pictures : 4	6	$P_{message} > t_p$: N

certain cases. This shows all possible scenarios for message delivery with Group 1 senders.

Table 4. Presence Groups/Categories with Message Priority Threshold values

Group No.	*Presence Type*	*Message Priority Threshold (t_p)*
1	**BUSY**	Less than or equal to 3
	Discussion, Reading Document, etc.	
2	**AWAY**	Less than or equal to 5
	Lunch, Coffee, In the Alley, etc.	
3	**AVAILABLE**	Less than or equal to 8
	In Room Reading Paper, etc.	
4	**NOT AVAILABLE**	None

Similarly, all possible combinations of messages belonging to Group 2 senders with subjects from all four groups and all possible combinations of messages belonging to senders from Group 3 and 4 with subjects from all four groups were considered.

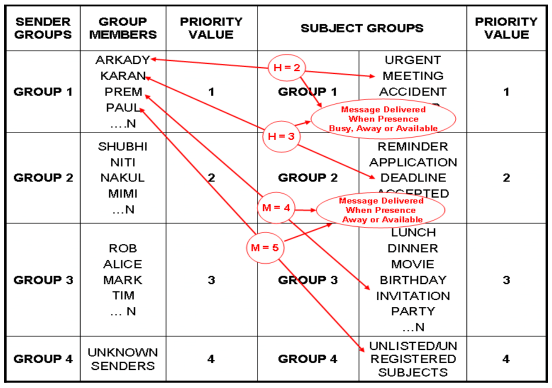

Fig 4: Combination of Group 1 Senders with various subjects with message priority.

Based on the message priority that is determined by adding the sender and subject priority, the messages are eligible or not eligible to be delivered in different presence states. The message priority and in which presence states they can be delivered to the user can be determined using equation (1). The advantages of this decision making process are that the user can move the members of a group from one group to another when he/she wishes to give more importance to a particular sender or subject. The other possibility is that the thresholds, t_p (consider table 4) for the delivery of message can be raised or lowered according to the way in which a user perceives the priority of message for delivery and the level of importance. Table 3 presents the experiments which were conducted to validate our prototype system. The table gives the user location (UL), device closest (DC), user activity (UA), user presence (Presence), Sender (P_{sender}), Subject ($P_{subject}$), priority of message ($P_{message}$) and Message Delivered (D:Yes/No) for all the experiments.

5. CONCLUSION

In this paper we present our novel context-aware UMS called the Intelligent Presence-Aware Decentralized Unified Messaging System (IPAD-UMS). Our proposed system solves problems which are inherent in current commercial UMS available in the market. It is context-aware, decentralized, user-centric and extremely low in cost. It is not vendor specific and can be extended for both personal and business use. For evaluation, we conducted several experiments within the office and meeting scenarios using three most widely used communication technologies, i.e. e-mail, SMS and IM on laptop, PDA and smartphone. The experimental results clearly validate the proposed system. Thus, IPAD-UMS as a decentralized, pervasive and ubiquitous system is a step towards providing the users with seamless connectivity.

6. REFERENCES

[1] P. P. Jayaraman, P. Hii, and A. Zaslavsky, "Message-On-Demand Service in a Decentralized Unified Messaging System," in *WONS 2006 : Third Annual Conference on Wireless On-demand Network Systems and Services*, 2006, pp. 68-77.

[2] K. Schreiner, "Unified Messaging: Will It Finally Meet Its Promise?," *IT Professional,* vol. 9, pp. 56-60, 2007.

[3] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," *Computer, IEEE Computer Society Press,* pp. 25-31, March 2003.

[4] J. M. Wams and M. van Steen, "Pervasive messaging," in *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, 2003, pp. 499-504.