

Context-Based Services Discovery in Mobile Environments

Y.Bouزيد¹, H. Harroud¹, M. Boulmalf¹, and A. Karmouch²

¹Alakhowayn University in Ifrane, Morocco

²University of Ottawa, Canada

{Y.Bouزيد, H.Harroud, M.Boulmalf}@aui.ma, Karmouch@site.uottawa.ca

Abstract— Mobile and Ubiquitous computing have increased the need for mobile users to expect accessing preferred services, whenever they want and wherever they are. Users do not have to explicitly specify and configure their working environment each time they move from one location to another. The necessary adaptation to cope with the changing environment should be initiated by services rather than by users. This paper presents a context-aware platform for supporting mobile users with personalized services. The platform is capable of handling different types of context sources (i.e. sensors, readers, agents), offers sophisticated mechanisms in matching the mobile user's preferences with services that are enabled at the visited location, and provides these services in personalized and adaptive manner to the user conditions. As a proof of concept, we present an context sensitive instant messaging prototype where a user is provided with services and information of interest based on location and time. The platform monitors a set of policies to dynamically adapt each application behavior to the situation of its use.

I. INTRODUCTION

The emergence of ubiquitous computing, enabled by the availability of portable devices and advances in (wireless) networking technologies, has increased the need for personalized and adaptive services in mobile environments. Users are not anymore only using computing facilities on their desktop machines in a relatively predefined office environment, but they require having access to various services as they move from one location to another, from one device to another and from one network to another. Making use of contextual information is essential to cope with and timely react to changes in such environments and hence achieve adaptability, reliability, and seamless service provisioning.

Context information may include any information that characterizes the user operating-environment. Baldauf in [1] presented a survey on context-aware systems that contains various definitions for the term. Most of these definitions are based on concrete examples and categories, and it is still difficult to determine, whether a particular kind of information can be regarded as context. In practice, four types are commonly introduced: the location which includes people and entities that are in or nearby (*where the user is*), the identity which determines the user's profile, role and preferences (*who is the user*), the activity that is occurring or may occur around (*what*), and the time (*when*).

The development of context sensitive services and their deployment remain challenging as they require appropriate paradigms in interacting with different sensing entities, in gathering, interpreting and disseminating different types of contextual information, and self-adapting to changing environments.

Users, device and session mobility are also challenging as services need to dynamically adapt to network changes and heterogeneity, occasional disconnections, resources availability (i.e. bandwidth), and device capabilities [2]. The necessary adaptation to cope with the changing environment should be initiated by services rather than by users.

This paper describes the approach we have adopted for supporting mobility and context awareness, so that mobile users could be provided with personalized services while moving from one location to another. This approach focuses on the development of a platform that allows developers to spend less effort on the characteristics that are common across various applications for mobile users and focus on the specific objectives of these applications. Managing context and mobility at the platform level clearly facilitates context aware programming by providing tools for discovering relevant context, for processing it, and for disseminating appropriate information to mobiles users' services in various domains such as e-Tourism and e-Transport applications.

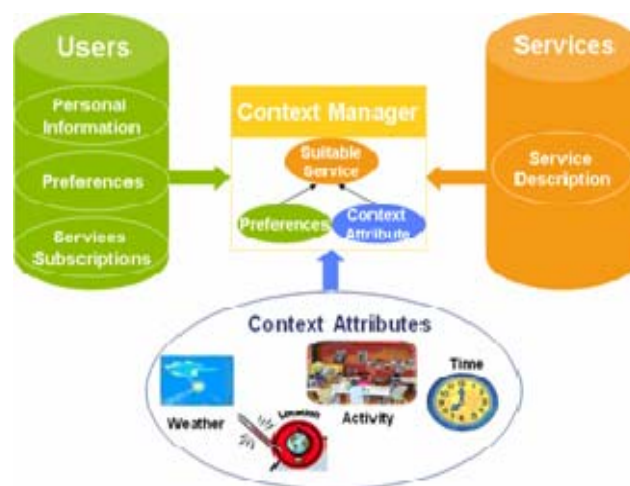


Fig 1: Generic view of the framework components.

A generic view of our context aware platform is shown in fig 1. The user repository holds user’s personal information, services to which she/he has subscribed, and her/his preferences regarding service provisioning. For instance, the user might specify a time of the day where she/he wants to receive a specific service.

The service repository carries information about the available services. The platform operates by evaluating changes in the different context attributes (e.g. weather, user location, time, end-user device...), and uses the user preferences to find matching services. As a result, the user is provided with personalized services that fit her/his current context and meet her/his preferences. When activated, a service continuously receives appropriate context information from the platform and therefore adapts itself accordingly.

The paper is organized as follows. Section 2 relates our work to existing approaches. Section 3 describes our context ontology and the user profile structures. Section 4 presents our overall architecture for providing personalized services in mobile environments. Section 5 describes a prototype scenario that illustrates the feasibility of our approach. In section 6 we summarize and point to future work.

II. RELATED WORK

Several approaches have been proposed for building prototypes of context aware platforms [3] [4] [5] [6]. Dey, et al. proposed in [3] the Context Toolkit framework to support collecting and transforming contextual information using widgets, interpreters and aggregators. Each widget is responsible for acquiring a certain type of context information. The aggregators collect context from widgets and interpreters and act as proxies to applications.

COMPASS [4] is a context aware mobile personal assistant which provides users (e.g. tourists) with context-aware recommendations and services. It retrieves and provides information about the user’s context by contacting appropriate context services (i.e. location, user and time contexts) and uses a *registry* that contains information about the third party services providing the content such as museums and restaurants information.

Chen et al. used OWL in their Context Broker Architecture (CoBrA), where a broker agent is responsible for maintaining and aggregating a shared model for context information [5]. The broker agent facilitates the distributed reasoning capabilities for service agents that make use of CoBrA by including a knowledge model and therefore removes the need to deal with the reasoning part for each service and application.

Kuck et al. presented in [6] an approach for the context-sensitive discovery of web services based on the matching of the user’s context and enhanced service descriptions, stored in UDDI repository. Service descriptions contain inferred information about textual contents of a WSDL description as well as feedback information (e.g. the time of service recommendation).

Our work to supporting mobility and context aware computing complements these research projects by integrating the implications of mobility and the context to the platform middleware. Applications and services developed upon the platform can easily be deployed in various scenarios including e-Tourism and e-Transport.

The platform can also be used with third party services which need to register to the platform. These services are then advertised to users depending on the time and their location and the current devices. At the registration phase to the platform, each service subscribes to specific context information so that it can adapt itself accordingly.

III. CONTEXT MODELING AND USER PROFILING

A. Context Modeling

A well defined representation of context has to be developed to allow the service provisioning process to learn about the different context attributes and provide context aware services. Modeling context in a flexible structure with explicit entities and relationships is essential in supporting efficient context management. Context modeling must also enable reasoning mechanisms on raw context. Ontology-based models allow definition of common vocabularies among various environments, and allow representation of relevant information for defining entities, properties, and semantic associations. We use an ontology built upon Ontology Web Language (OWL). The core ontology of context is depicted in Fig 2. A detailed description of the context ontology can be found in our previous work [7].



Fig 2: Context View Ontology

The Global Skeletal Ontology (GSO) represents a high level abstraction with two different classes of context: Simple Context and Composite Context. The Simple Context class represents a group of information referring to the same kind of context provided by one or more context sources. We defined basic ontologies to express simple context information about location, time, user, device, and network. Composite Context corresponds to different Simple Context entities bound by different types of associations and aggregations. For example, Presence is aggregated from user ontology and preferences. The Environment Context subclass provides typical context information about an environment (i.e. Home, Office...).

Context suppliers differ in the context information they can provide based on the domain within which they exist, their resources, and available context sources. These differences are expressed through the Global Concrete Ontology (GCO). GCOs are extended from the general GSO, providing domain-specific ontologies to model the

providers' acquired context information for easy access and sharing.

The GSO is further applied to derive Client Profiles (CP) and Client Views (CV). CPs are used by services to model their characteristics and capabilities, thus supplying the platform with information necessary to make decisions as to what context information services may be interested in receiving. CVs provide services with information related to their accessible contextual information. Informing services of the context information they may request from the platform is done through CVs.

Association and aggregation rules are maintained by a set of policies. For example, a policy may involve the device, the location, and the presence ontology such as: if the battery level of the device in use is less than 25% and the presence of a local device is true and its proximity is 'nearby' then the transfer of the active session must be notified.

B. User Profile Representation

In addition to context modelling, a precise representation of user profile helps deliver a user experience tailored to its needs and preferences. Personalization is an important aspect in context-aware systems. It requires knowledge about user interests and service preferences.

We have represented a flexible user profile in a XML Schema based on the set of common rules to specify profiles and data components as introduced in the Data Description Method (DDM) of 3GPP Generic User Profile [8]. The schema in fig 3 is a simplified representation of our adopted user profile consisting of user, device, and preference components.

The user component defines data elements grouped under identity information and contact information. The device component describes the different characteristics of the user's mobile device. The substructures under this component describe hardware, software, browser and WAP characteristics of the device.

The preference component describes the user's preferences with regards to the available services. This component can hold for instance descriptions of the user preferred hotels, transportation modes, attractions, etc.... The preference section in fig 3 describes the user preferred types of restaurants and its level of appreciation for each restaurant ranging from 5 being the user's favourite type of restaurants and 1 being its least favourite one.

The preference section also specifies the location and time that are most suitable for receiving specific services. For example, the user profile in the XML sheet above specifies that she wants to be informed about the presence of a Fast Food Restaurant only in the evenings and only when the Fast Food restaurant is no more than 5000 meters away from her current location.

```

<Profile>
  <user:User>
    <user:Identity>
      <user:FirstName>Lisa</user:FirstName>
      <user:LastName>Smith</user:LastName>
    </user:Identity>
    <user:Contact>
      <user:Email>Lisa.Smith@gw.edu</user:Email>
      <user:TelephoneWork>524-324-5321</user:TelephoneWork>
    </user:Contact>
  </user:User>
  <device:Device>
    <device:HardwarePlatform>
      <device:ScreenSize>80x80</device:ScreenSize>
      <device:ImageCapable>true</device:ImageCapable>
      <device:Keyboard>PhoneKeypad</device:Keyboard>
      <device:Markup>WML</device:Markup>
    </device:HardwarePlatform>
  </device:Device>
  <preference:Preference>
    <preference:PreferredRestaurants>
      <preference:RestaurantType>FastFood</preference:RestaurantType>
      <preference:AppreciationLevel>5</preference:AppreciationLevel>
      <preference:LocationDistance>5000</preference:LocationDistance>
      <preference:time>evening</preference:time>
    </preference:PreferredRestaurants>
  </preference:Preference>
</Profile>

```

Fig 3: User Profile

IV. THE OVERALL PLATFORM ARCHITECTURE

The development of the platform is based on a generic architecture that supports context aware service discovery. The architecture is represented by a generic layered stack that describes the main functionalities of our context aware system. Fig 4 shows a five layer model that separates the concerns of each layer among acquiring, processing context information, and providing users with services that best fit their current context. The architecture handles a variety of sensing devices, uses a context model that can be extended with new context data types, makes use of a generic description of services and user profiles, and provides services to a wide range of mobile users.

A. Context Data Acquisition

The data acquisition layer is in charge of collecting context attributes from various ubiquitous front end data acquisition hardware (e.g. RFID readers, sensors, and other automation devices). This layer listens for signals that hold information describing the context.

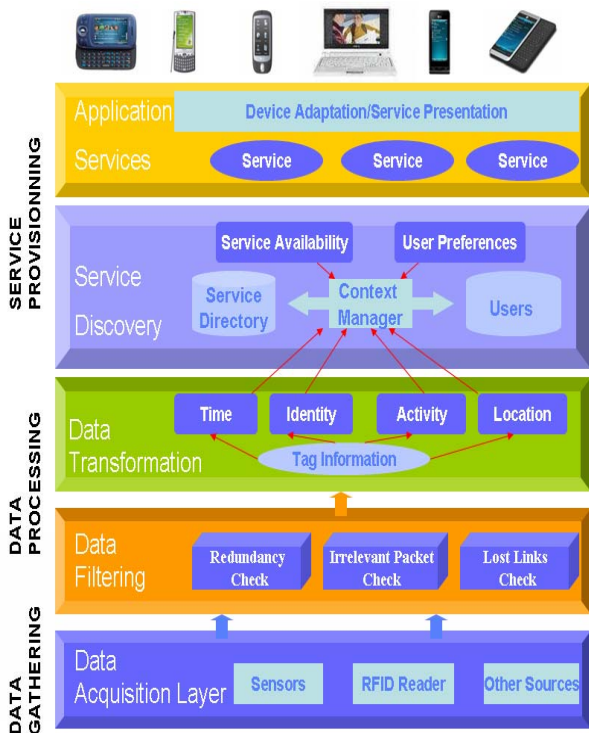


Fig 4: The overall Platform Architecture.

In our platform we have built a network of sensors that deliver different kind of information to the base station such as temperature, noise, etc... The sensor network is also used to roughly determine the position of a mobile user by fixing some nodes at specific locations (i.e. buildings) and exploiting links that mobile nodes, carried by mobile users, form with fixed nodes while nearby.

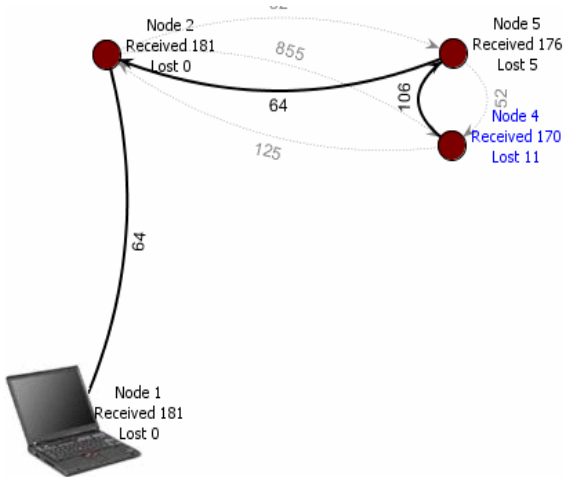


Fig 5: Mobile and Stationary Nodes Distribution

Fig 5 provides an example of sensor network nodes distribution. In this configuration, nodes 1, 2, 5 are stationary nodes, while node 4 is mobile.

Because the data is received in raw format that may not be understood by the service discovery module, it is first forwarded to a data processing layer that transforms it into

meaningful information. An example of incoming sensor network packets is shown in fig 6.

```

1236675155187 Message <MultiHopMsg>
[ sourceaddr=0x1 ]
[ originaddr=0x5 ]
[ seqno=0x9fd ]
[ originseqno=0x511 ]
[ ttl=0x6 ]
[ id=0x21 ]
[ data=0x95 0x1 0x0 0x0 0xe 0xb 0x4 0x0 0x3 0xfe 0x4 0x0 0x3 0x0 0x2 0x0 0xa5
0x0 ]
1236675155703 Message <MultiHopMsg>
[ sourceaddr=0x1 ]
[ originaddr=0x2 ]
[ seqno=0x9fe ]
[ originseqno=0x55e ]
[ ttl=0xc ]
[ id=0x21 ]
[ data=0xf9 0x1 0x0 0x0 0x3f 0xb 0x3 0x0 0x3 0x6e 0x4 0x0 0x3 0x0 0x5 0x0 0x64
0x2 ]

```

Fig 6: Incoming Sensor Network Packets Log File.

B. Context Data Processing

The basic goal of context processing layer is to generate concise and accurate information about the context that would be used in the context-sensitive service discovery. Mechanisms used by the data processing layer include filtering and transforming the received raw data.

1) Data filtering

Different filtering methods could be applied to raw data depending on the types of used sensing sources and on the nature of the data required at the application level. The filtering layer holds a filtering policy repository to offer the flexibility to handle different filtering format. In our platform, the filtering has three main functions:

- Duplicate Removal: Since the data is sensed continuously, the sensors may re-send the same data read multiple times. This happens when no changes are taking place in the network. The duplicate removal limits the amount of data to be processed by reporting the data read once.
- Irrelevant data Removal: Some of the data received by the sensors is of no use to the service applications. For instance, the data received that reports about links formed between two stationary nodes does not provide any additional information. We are more interested in knowing about links between stationary and mobile nodes.
- Lost links removal: Because a mobile node may change its location, we need to continuously update the mapping of mobile node to physical locations to keep the latest information about its current position. While a mobile user is in proximity to a building, the base station receives data describing the link between the mobile user node and the stationary node in that building. If the base station stops receiving data about the existence of that link within a time frame window, the link is reported as lost. This means that the user has moved to another location.

2) Data Transformation

Raw data presents little information until they are transformed into a form suitable for application-level interactions. So, from an application perspective, it is desirable to provide a mechanism that turns the low-level captured data into meaningful input. The transformation layer contains pre-defined rules for transformation depending on the type of the raw data. For instance, the geographic x-y coordinates obtained from a GPS are translated into physical locations (street, city...).

This layer presents flexibility with regards to transformation rule definition, since they can be added, changed and deleted in an easy manner. Transformation rules are represented using policies. In our implemented prototype, we used transformation rules that translate stationary node's tag ID into a building name (i.e. location) and a mobile node's tag ID into a mobile user (i.e. identity).

The Context processing layer provides the upper service layer with meaningful and accurate descriptions of context attributes to allow context sensitive service discovery.

C. Context Service Provisioning

In this layer, the context manager component has the role of matching the user context (e.g. location) with the appropriate services. The context manager also takes into consideration the user preferences to offer personalized services to each user.

Upon receiving the context information, the context manager sends a request to the appropriate third party services (i.e. web services) to find services that better match the user current context. The context manager would for instance ask a web service to find all the restaurants that are close to a user location. The web service return a list of services that best fits the user current context. The context manager then classifies received services based on how much they meet user preferences according to her/his profile. The result of this process is that the user is provided with a list of services that match her/his current context and that are tailored to her/his preferences. Fig 7 shows a simplified sequence diagram of context discovery process.

A list of identified services is then provided to the user depending on her/his current situation. The activation of a service requires an adaptation of its presentation to the user's device. Different mobile devices have different features in terms of the size of their screen, the content type (e.g HTML, WML) supported by the mobile browser. Context policies, policies for which conditions make use of context attributes, can be associated with each service, so that content and presentation adaptation are automatically enforced by the policy enforcement points (PEPs) [9]. For instance, policies will enforce transforming the web page materials into an appropriate markup language that the mobile device can interpret by holding a set of XSLT stylesheet (Extensible Stylesheet Language Transformations) for each type of mobile device.

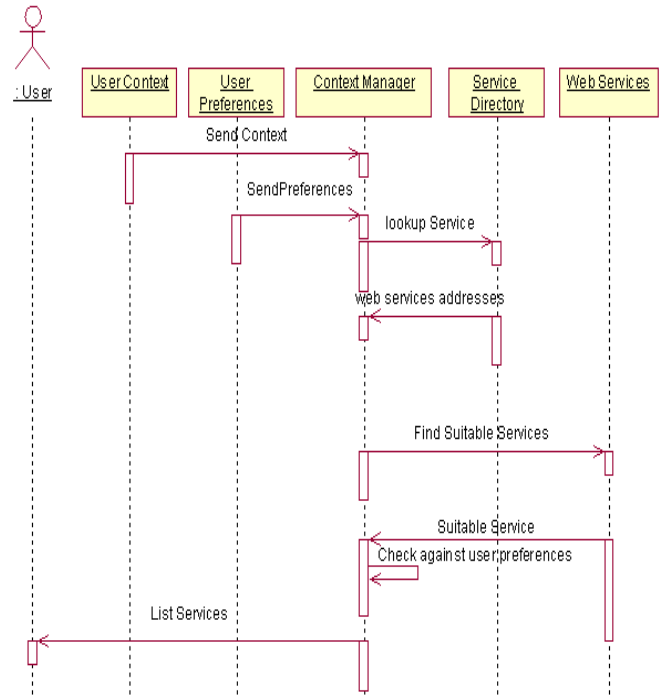


Fig 7: Sequence Diagram Describing Service Discovery Algorithm.

V. PROTOTYPE SCENARIO

Our platform will be deployed in an application for e-tourism that will assist tourists during their travels by providing them with context sensitive services. We expect to integrate user preferences, the current time, and the user current location in the proactive formulation of suggestions on PDA devices about nearby points of interests (e.g. museums, restaurants...).

In this section, we describe a context sensitive instant messaging service (C-IMS) that has been implemented for testing the main components of our proposed platform. The C-IMS is an extended version of the buddy space open source in [10] where context information about users and registered services are provided by the platform. The C-IMS integrates therefore information such user location, presence, activity alert, etc... We divided the first floor of the building 7 at AUI University in three locations with respect to our WML research laboratory (3 fixed nodes), and we associated mobile sensor nodes to two users (Balo and Hamid) and two entities (Printer and Projector). We defined three services, namely the printing service, the projector service and friends' presence service, as being the preferences of a user Bob.

Figure 8 shows a snapshot of the C-IMS user interface of user Bob and his associated services. According to his location and time, the user Bob has access to the printing service (i.e. green light), can have access to the projector, but it is currently busy (i.e. red light), and he is continuously notified about the presence of his friends that have been specified in his preferences. His friend Balo is currently in a meeting activity and Hamid is not in the range of the three

locations covered in this prototype, which means that the platform is not aware of his presence status.



Figure 8: a Snapshot of the CS-IMS User Interface

VI. CONCLUSION

In this paper we presented a generic platform for providing users with personalized context aware services in mobile environments. The platform offers mechanisms (i) to register a set of mobile users, their roles and their preferences, (ii) to register a set of available services at different locations, (iii) and by evaluating different context information, provides each user with matching services. The platform also provides activated services with appropriate context attributes, so that they can adapt accordingly. As a result, the user is provided with personalized services that fit her/his current context and meet her/his preferences. When activated, a service continuously receives appropriate context information from the platform and therefore adapts itself accordingly.

We showed the feasibility of the proposed platform architecture using a prototype that implements a typical mobile user instant messaging scenario illustrating the platform's basic concepts.

We are currently in the process of implementing basic services related to e-Tourism and integrating mobile RFID readers as sources of context, so that we can deploy our platform in a working environment (i.e. city of Ifrane) to illustrate how it can be helpful in providing support to tourism, a vital sector in the Moroccan economy.

ACKNOWLEDGMENT

This research work is partly supported by a grant from the Academics Affairs at Al Akhawayn University in Ifrane (Grant n° 92780, 2008).

REFERENCES

- [1] Baldauf M., Dustdar S. and Rosenberg F., 2007. A survey on contextaware systems. In *Int. Journal of Ad Hoc and Ubiquitous Computing*, Vol. 2, No. 4.
- [2] Keeney J., Cahill V., 2003. Chisel: A Policy-Driven, Context-Aware, Dynamic Adaptation Framework, In *Proceedings of the Fourth IEEE International Workshop on Policies for Distributed Systems and Networks*.
- [3] Dey K., Salber D., Abowd D., 2001. Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. In *Human-Computer Interaction Journal*, Vol. 16 (2-4).
- [4] Setten V., M., Pokraev, S., Koolwaaij J., 2004. Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In *Nejdl, W. & De Bra, P. (Eds.)*.
- [5] Chen H.L., Finin T., Joshi A., 2004. A context broker for building smart meeting rooms. In *Proceedings of the Knowledge Representation and Ontology for Autonomous Systems Symposium, LNCS 3137, Springer-Verlag. Technical Specification, Version 6.1.0*.
- [6] Kuck J., Reichartz F., 2007. A collaborative and feature based approach to Context-Sensitive Service Discovery. In *Proceedings of 5th WWW Workshop on Emerging Applications for Wireless and Mobile Access (MobEA'07), Banff, Alberta, Canada*.
- [7] Ridhawi Y., Harroud H., Karmouch A., Agoulmine N., 2008. Policy Driven Context-Aware Services in Mobile Environments. In *Proceedings of 5th International Conference on Innovations in Information Technology (Innovations'08), Dubai, UAE*.
- [8] Group Terminals, 2004. Generation Partnership Project: Data Description Method (DDM) -3GPP Generic User Profile (GUP). In *Samaan N., Harroud H., Karmouch A., 2007. PACMAN: A Policy-Based Architecture for Context Management in Ambient Networks. In 4th Annual IEEE Consumer Communications and Networking Conference (CCNC'07), Las Vegas, USA*.
- [10] Eisenstadt M. et al., 2007. BuddySpace Project. In <http://buddyspace.sourceforge.net>