

# Context-Enhanced Service Provisioning in Wireless Networks

Mauro Brunato

Department of Computer Science and Telecommunications, Università di Trento

Via Sommarive 14, I-38050 Trento — Italy

brunato@dit.unitn.it

## Abstract

We describe a context-aware architecture under development at the Department of Computer Science and Telecommunications of the University of Trento.

The middleware system described in this report can integrate various data about the user and his/her context (e.g., position) and provide them in a transparent and system-independent way via a SOAP interface to dynamic web pages and other client applications.

The main components of the architecture are described, and the main development choices are motivated.

## I. INTRODUCTION

While wireless local area networks (WLANs) are becoming ubiquitous, and promises of always-on networking become more and more realistic, the main feature of these communications media, i.e. user mobility, is often underused. This fact differentiates WLAN access networks from cellular phone systems, where the user's ability to move while communicating is a key factor. WLANs, on the contrary, are mainly used from the desktop, as a cheap cable replacement.

In spite of this fact, there is a growing belief that wireless systems can easily provide an additional value to common network usage. Beyond the ability to freely move while staying connected, with some limits due to the roaming difficulties of Wi-Fi systems, the system's ability to infer the user's (possibly changing) location can result in many innovative services [1], [2]. In this report we describe a middleware architecture that enables high-level services to take into account the user's context, in particular his position.

This report is structured as follows: in Section II we provide an overview of the system components and the overall architecture; in Section III we describe how context data (user location) can be retrieved by the system. In Section IV we briefly describe an experimental context-aware service based on this architecture.

## II. CONTEXT-AWARE ARCHITECTURE

The Uni-Fy AAA platform [3], [4] deployed at the CS department enables network managers to fully exploit new services that take advantage of context information. Such services can provide additional benefits to wireless users, promoting mobility and helping network operators and researchers to collect usage data.

A sample system, aimed at providing context data (in particular location) to user applications at the Faculty of Science of the University of Trento, is described in Figure 1. The system is structured into three layers: *network*, *middleware* and *application*.

The *network*, at the bottom of the picture, is the collection of all networking equipment, in particular devices related to wireless access: APs, AAA gateways and mobile clients. We only consider components that are "context-aware", in the sense that they are able to provide some information that helps to build context representations. For instance, APs usually provide association information that can be used to estimate clients' positions; clients, on the other hand, may optionally localize themselves by means of specialized hardware (GPS receivers, RFID tags) or by exploiting radio properties of the wireless medium; the AAA gateway can complement location data with the identity of the users, their traffic and time since login.

The *context broker* is the main component of our architecture. Its purpose is to obtain context data by correlating information obtained by polling network devices and provide coherent context information to applications. It is composed of two logically separate components: the *context database* and the *middleware service*. The first basically contains a database of context information about users. Currently, time, identity and location data are considered. Such data are built by an application that receives information from network devices by using a mixture of methods, including SNMP polling and Syslog trapping (see Section III for a description of methods); data are matched by common tags (for instance, APs and the AAA gateway both report IEEE802 MAC addresses, while client applications use IP addresses that are also known by the AAA gateway). Currently, user locations are inferred in an asynchronous manner. In fact, active logging functionalities of some AP models have been found to be unreliable (particularly on earlier models), so users are not always localized as soon as they connect, but shortly thereafter by a polling cycle. Information stored in the database is not directly accessed by applications. A *middleware service* is interposed in order to provide a standard protocol for querying, independence from the DB representation and additional functions such as location tracking, confidence estimates, search functions or integration with user profile information. The service is implemented as a SOAP Java servlet, thus appearing as a standard web service to upper layer applications.

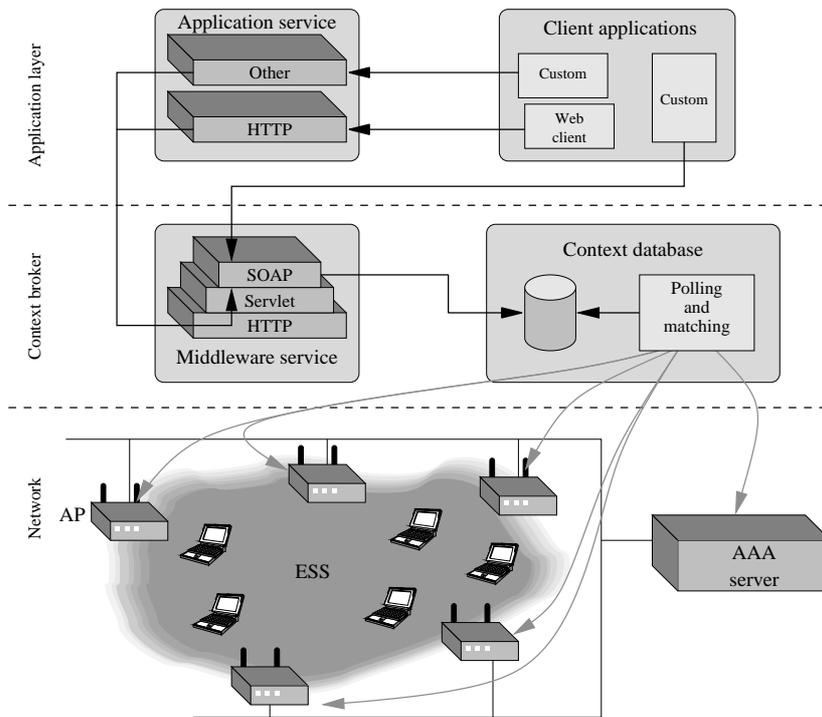


Fig. 1. Block diagram of the context-enhanced service framework

---

`http://laica.science.unitn.it:8082/axis/services/InfoClients?method=getLastByIp&in0=172.31.194.155`

---

Fig. 2. A query URL.

Finally, in the *application* layer (upper part of Figure 1) several types of user services can be found. The context middleware can be queried by a web server running a (PHP, ASP, JSP, CGI) script, by some other application service, or by context-aware custom applications that connect directly to the middleware layer.

#### A. Access to middleware services

Communication between the middleware components and network devices is strongly hardware-dependent and uses a variety of synchronous and asynchronous methods, some of which shall be detailed in Section III; one of the middleware's purposes is to hide this complexity from the user by presenting a common interface.

The choice of SOAP as the query protocol is motivated both by the availability of soap-enabled libraries for most languages (the web service is implemented in Java, test clients have been written in PHP) and the ability of testing the service's functionalities by using a common browser. Figure 2 shows the URL used to query the sample system about the last occurrence of a given IP. The corresponding answer is provided in Figure 3: the MAC address the IP was last assigned to (the six initial octets of the line within tag `getByIpReturn`) is shown; response fields are separated by pipe symbols, and the third field denotes the code of the associated AP.

A list of available methods is given in Table I. The `currentTime` query is provided for synchronization; the `getByIp` query returns information about all clients that have been assigned a given IP address, and is mostly used for debugging. The `getLastByIp` command returns data about the last occurrence of a given IP address, as shown in the previous example. The `numClients` command asks data about the current load of an AP.

Finally, method `setCoord` lets a client force its own  $(x, y)$  coordinates, provided that the client has some technique for inferring them.

### III. CLIENT LOCALIZATION

Client devices such as laptops, smartphones and PDAs can be located by means of several different mechanisms. Since the TWELVE project is oriented towards IEEE802.11 networks, we are not considering custom localization techniques (GPS, RFID or ultrasonic transceivers): localization shall be therefore provided at no additional cost by the networking equipment, either with cell granularity or by finer estimation methods.

---

```

<soapenv:Envelope>
  <soapenv:Body>
    <getLastByIpResponse soapenv:encodingStyle="...">
      <getLastByIpReturn xsi:type="soapenc:string">
        00 13 CE 1B 00 3E|172.31.194.155|4|Mickey Mouse|mickey@unitn.it|0|0|20|2006-05-10 13:26.06
      </getLastByIpReturn>
    </getLastByIpResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

---

Fig. 3. Response to the `getLastByIp` query.

TABLE I  
INFOCLIENTS QUERIES.

Method	Action
<code>currentTime</code>	Get the server's time.
<code>getByIp</code>	Get clients who have been detected to use a particular IP address.
<code>getLastByIp</code>	Get the last client who has been detected to use a particular IP address
<code>numClients</code>	Get the number of clients associated to a given AP.
<code>setCoord</code>	Force the client's coordinates.

### A. Cell granularity

In order to be connected to the network, a mobile terminal needs to be associated to an access point. Association is always client-driven, and a client shall usually choose the AP having the highest SNR with respect to its location. The assumption that a client associates to the nearest AP (or, more generally, that it is possible to assign to each AP a specific “association domain” in the physical space with little overlap to neighboring APs) is therefore reasonable most of the times. Some APs offer a load-balancing feature that allows them to refuse clients, forcing them to associate to farther APs with a lower load; however, this feature is useful only where AP density is high; thus, we can easily assume that the position of the associated AP is a good estimate of the mobile terminal's position, with an average error within a few tens of meters.

The basic form of client localization is therefore implemented just by looking which access point the mobile terminal is associated to. Most APs provide functions that enable authorized parties to discover which clients are associated at a given time. The four main techniques are described in Figure 4.

1) *Asynchronous polling*: The simplest case, requiring almost no configuration on the AP side, is just asynchronous polling (Figure 4, part a). The context database periodically polls all databases by a common network administration protocol such as SNMP; all APs, in turn, provide an updated list of associated client, usually identified by their interface's MAC address.

2) *Syslog notifications*: Most APs can also be configured to send appropriate Syslog notifications when some classes of events occur (Figure 4, part b). The context database IP address must be inserted in the AP configuration; since most APs consider association as a low-priority event, APs must be instructed to send non-crucial notification. One drawback of the system is that syslog packets do not require acknowledgments, so notifications may be lost. This problem can be overcome by the RADIUS authorization method.

3) *RADIUS authorization*: Many APs also provide some MAC-based authorization functionalities (Figure 4, part c): before accepting a client's association, an AP may check its MAC address against a RADIUS server's list. The context database can operate as a RADIUS server by sending authorization answers to such queries, while recording them. If MAC-based authorization is not required, the RADIUS server forges positive answers regardless of the request. This method is less elegant with respect to Syslog notification, because it makes use of a functionality (MAC-based authentication) for different purposes, but it overcomes the possible loss of information discussed in the previous paragraph; a missing RADIUS authorization shall force the AP to reissue the request, thus operating as an implicit ARQ mechanism.

4) *Client notification*: A specialized client program can be placed in the mobile terminal. Most wireless interfaces provide an API allowing user-level applications to retrieve fundamental connection data such as the cell's ESSID and the AP's ID (usually a MAC address). The client program shall send relevant information to the context database when it detects a change in connection parameters, or even periodically. Installation of a custom application on a mobile terminal is often discouraging, but this method lets the user decide whether information should be sent or not, thus enabling a higher control of privacy.

### B. Fine-grained localization

Wireless client hardware often provides useful information about the radio environment; many Wi-Fi cards provide APIs that enable client applications to obtain signal intensity data from all APs within range. These data can be issued as SNR levels in a standard unit of measure (such as decibels over one milliwatt, dBm) or in an arbitrary scale.

Probing the radio environment can enable clients to infer a better estimate of their position, usually well below the cell-to-cell distance. Various research and deployment experiences [5], [6] prove the feasibility of this approach.

Unfortunately, fine-grained localization has at least two drawbacks:

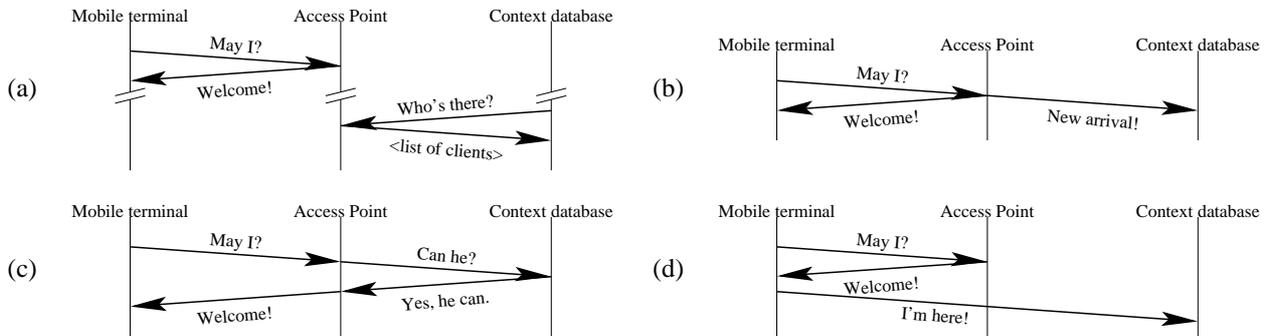


Fig. 4. Different forms of location disclosure: (a) Asynchronous SNMP query by database; (b) Database informed as Syslog server; (c) Database informed as RADIUS authenticator; (d) Client's notification

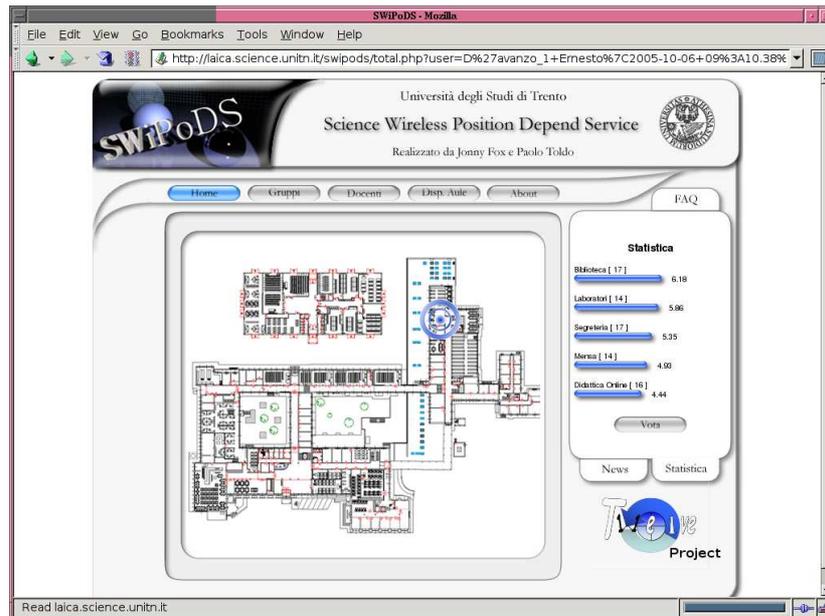


Fig. 5. Example of location-aware service developed by students at the University of Trento. A location map and a context-dependent poll are shown as sample services.

- commercial APs lack the capability of scanning the radio space in search of client signals, and adding this functionality would deteriorate their performance, therefore RSSI measurements are always client's responsibility;
- continuous probing for radio signals is a power demanding activity, which leads to faster battery draught in mobile equipment.

Nonetheless, experimental client applications have been implemented for PDAs and laptops.

#### IV. IMPLEMENTATION

An example of context-aware service currently available at the University of Trento is shown in Figure 5. The captive-portal authentication mechanism implemented in the Uni-Fy access network [3] allows the system to offer the user a link to a local web page whose contents are tailored according to the user's identity, position and, possibly, preferences (either declared or inferred). The web page contains the current user's location, suggestions about the nearest available classroom, a poll system about university services, tools for the organization of student communities. Other envisioned services include the suggestion of less congested wireless locations, restaurant menu (for students nearby the cafeteria and only during lunch hours) and teacher availability.

Another longer-term service will be a wireless video broadcasting system which shall be used to diffuse lectures and talks to students. The system shall take advantage of a mixture of coding techniques in order to ensure good video quality even in the case of high packet loss ratio, while offering optimum performance to properly positioned clients.

#### V. CONCLUSIONS

This report describes ongoing activity at the Department of Computer Science and Telecommunications of the University of Trento.

The context-enhanced service is articulated in a low-level interaction between a *context broker*, which asynchronously polls or receives notifications from access points and other hardware, and a high-level web service for providing context data to dynamic web pages and client applications. Different solutions are possible for client localization. A sample web-based application has also been developed.

## VI. ACKNOWLEDGMENTS

Many thanks to Stefano Sgorlon, Paolo Toldo and Jonny Fox for implementing various parts of the system as parts of their stage and thesis activities.

This work has been supported by the Italian Ministry for University and Research (MIUR) under the PRIN Project TWELVE (<http://twelve.unitn.it/>).

## REFERENCES

- [1] M. Brunato and R. Battiti, "PILGRIM: A location broker and mobility-aware recommendation system," in *Proc. PerCom 2003*, S. Das, Ed. IEEE Computer Society, 2003, pp. 265–272.
- [2] —, "A location-dependent recommender system for the web," in *Proc. MobEA 2003*, 2003.
- [3] M. Brunato and D. Severina, "WilmaGate: a new open access gateway for hotspot management," in *Proc. Third ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, S. Choi, M. M. Buddhikot, and R. A. L. Cigno, Eds. ACM Press, 2005, pp. 56–64.
- [4] D. Severina, M. Brunato, A. Ordine, and L. Veltri, "UniWireless: a distributed open access network," in *Proc. Fourth ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, R. A. L. Cigno, A. Prasad, and S. Das, Eds. ACM Press, 2006, pp. 30–36.
- [5] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*. IEEE Communications Society, 2000, pp. 775–784.
- [6] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Computer Networks*, vol. 47, no. 6, pp. 825–845, 2005.