

A Context Aware Platform for Mobility in Immersive Environment

Invited Paper

T. Pavani
WILAB
University of Bologna
Italy
tpavani@deis.unibo.it

F. Marchesi
WILAB
University of Bologna
Italy
fmarchesi@deis.unibo.it

D. Dardari
WILAB
University of Bologna
Italy
ddardari@deis.unibo.it

A. Conti
ENDIF, University of Ferrara
WILAB, University of Bologna
Italy
a.conti@ieee.org

O. Andrisano
WILAB
University of Bologna
Italy
o.andrisano@ieee.org

ABSTRACT

This article describes the architecture and functionality of a hardware/software platform for context aware services. The implementation of the platform requires several aspects to be taken into account at different levels: technology, algorithms and context-aware applications requiring reliable position estimation.

Within the Italian project on Virtual Immersive Communications [1], we realized a multiuser immersive guide taking into account several solutions and implementation for all levels. In fact, our immersive guide communicates toward heterogeneous wireless networks and collects information on the user position both indoor and outdoor through properly setup communication technologies, furnishes location information through robust algorithms, and enables access to context aware services through a graphical service-browser which interacts, following a client-server protocol, in a multiuser environments.

Keywords

Context Aware, Service, Immersive, Localization

1. INTRODUCTION

The concept of context awareness in the computer science, is related to systems capable to recognize the current environmental state and reconfigure themselves consequently. In this paper, we consider the context awareness concept with the meaning of perception of the physical surroundings: consequently the participation to the context is conditional to

the capacity to interact with it by means of contextual information.

This view of the context awareness used to model the “immersive environment” in which users can transparently interact with virtual and physical entities, has been exploited through the design and implementation of a “testbed”, called Mobile Immersive Environment (MIE) and developed within the Italian Project on Virtual Immersive Communication (VICOM) [1].

The context-aware system we realized grants to the users an augmented immersive environment and a model of ubiquitous computing through multiuser indoor/outdoor virtual guide services. This example of multi-user context-aware service permits to each user to localize himself in both outdoor and indoor environments. This goal is achieved by the creation of a context containing information produced by every participants: while users share their positions and any eventual public data, other special virtual entities share informations about available services and about the physical environment which users are currently located in. Opportune applications allow users to manage and use these information, enhancing both users’ perception of environment and communication capability.

Due to the requirements, the location information in MIE plays a significant role in the definition of the context that could be logically extended on large geographical distances: therefore the platform has also the role of showcase for all the technologies developed during the project researches with special attention to those with localization capabilities [2].

Our choice for indoor localization is to experiment wireless sensor network (WSN) with a received signal strength indication (RSSI) based distance estimation, whereas in an outdoor environment the localization task is simply accomplished by the Global Positioning System (GPS) [3] integrated with UMTS.

The structure of the MIE platform is three-layered as depicted in Fig. 1. The first layer (technology layer) manages all the devices used for localization and communication. The mid-layer, the algorithm layer, collects data from the technology layer, and processes contextual info for the upper layer, the service layer, responsible for providing interfaces

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IMMERSCOM 2007 October 10-12, 2007, Verona, Italy
Copyright 2007 ACM ICST 978-963-9799-06-6 ...\$5.00.

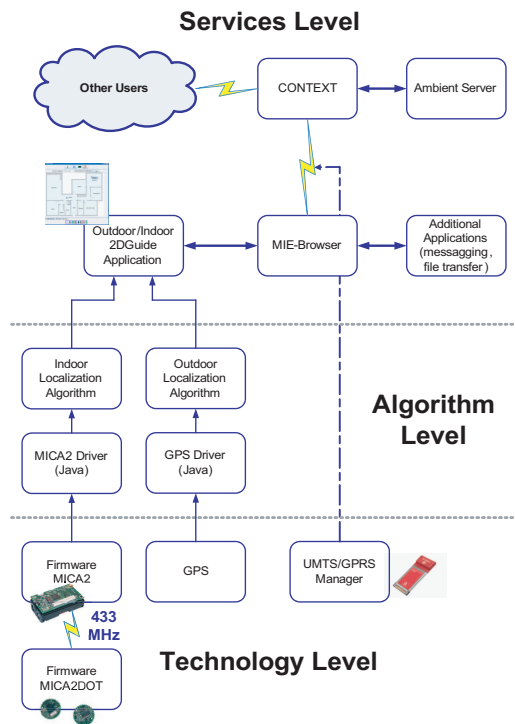


Figure 1: Context-aware platform structure organized in three levels

and services to the user.

The paper is organized as follows: section 2 describes the most relevant elements that operate in the MIE environment as well as their mutual interaction; sections 3, 4, and 5 discuss the structure of technology, algorithm, and service levels respectively; in section 6 we present the results obtained through on-field test and measurements; finally, in section 7, our conclusions are given.

2. DESIGN OVERVIEW

2.1 Architecture Implementation

The elements that operate in the MIE environment are called “MIE Devices”. Three particular MIE Devices, the most relevant in the test bed scenario, are User Device, Service Device and Ambient Server:

- a User Device could be any device which has enough computational and communicational capabilities to interact with the context and the services that are offered and, at the same time, has a relatively small size to allow the mobility of the user (e.g., hand-held or Tablet PC);
- a Service Device is an element that operates in the context to publish a service and manages the relative incoming requests. The service offered could be relative to a physical locally managed device (e.g., a Service Device could be a pc equipped with a printer that publishes, in the context, a Printing Service) or just a virtual function (i.e., a shared directory for store data on a server);

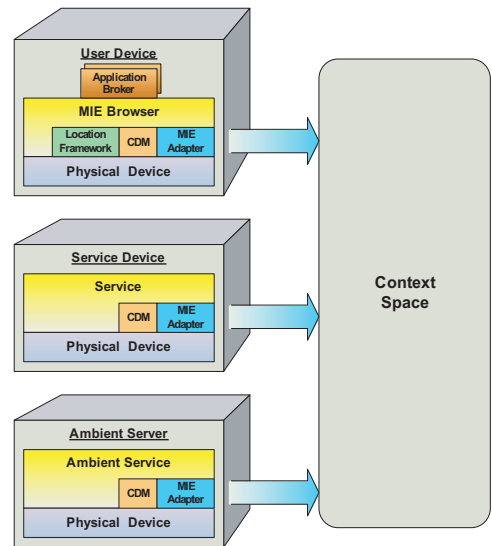


Figure 2: Testbed system architecture

- the Ambient Server purpose is to transpose, in the context, the physical environment basic characteristics. Its main tasks are: to publish information about the geometry of the real places, to offer an interface to the context to the “non-intelligent” element that are present in the surroundings (i.e., automatic door, notice board, etc.), and to provide information of functionalities (e.g., opening time of an office).

In Fig. 2 these three devices are shown with details on their internal structure. A MIE Device is composed of several internal components: some of them are specific to the particular function that the MIE Device should perform in the MIE environment like the Service, the Application Brokers, the MIE Browser and the Location Framework. Some other components are common to all the platform such as the Context Data Manager (CDM) and the MIE adapter which will be discussed in the following.

2.1.1 MIE Adapter

This component manages the communication of the MIE Device with the context. Its duty is to configure and adapt any existing network capabilities available on the MIE Device and to mediate the communication between others components and the context.

2.1.2 CDM

The Context Data manager is the component that manages and retrieves the contextual information needed by other components in the MIE device to perform their tasks. Besides this function, a CDM has to publish the contextual data for which a particular MIE Device is responsible (e.g., a Service Device must keep updated and available the contextual information about the service it offers).

2.1.3 Service

This component offers a service and publishes its availability in the context. In a client-server view, this module is the server part of the service whereas the client is the relative Application Broker offered by this Service. In this way

there is no need for the User Device to know how to communicate with a Service: the interface (i.e., the Application Broker), once obtained and loaded on the User Device, takes care of everything. If the particular service doesn't require the server part, this module just offers the Application Broker to download and publish the relative information on the context. The Ambient Server is a Service as well but with such a specific and unique role in the environment that it is defined as a particular type of MIE Device.

2.1.4 Application Broker

This component represents the service interface which the user can interact with. The most relevant feature of this component is that it could be loaded on demand on a MIE Device: a user could browse the context to look for the available Application Broker to download and run on his device. An Application Broker could be both the client side of a complex service or just a ready-to-use application.

2.1.5 MIE Browser

This component has a crucial role in the User Device; it is the graphical environment in which the Application Broker are loaded and managed and also the responsible for coordination of the other component (CDM, MIE Adapter).

2.1.6 Location Framework

This component is generally present in every user device and its task is to manage the various localization technologies which could be present in the physical device (e.g., GPS, WSN, etc.). It can load different Localization Plugins and gather the generated position data.

2.2 Operative Approach

One of the main concepts of MIE test-bed is the progressive discovery of the surrounding environment and the real-time adaptability to it: the management of the Services that can be available to the user strictly relates to that philosophy.

The presence of a service is not previously known by users' terminals, but "announced", in the context space, thus a user is not limited to use a pre-set of applications.

Information about available services are collected by the local Device and presented to the user: once a service is selected for execution, the user Device takes care of connecting to the relative Service Provider, download and launch the Application Broker.

To enhance the experience of the user in the MIE environment, an Application Broker interoperation system is integrated in the test-bed. Each Broker has the ability to declare some operations that it can do with particular system entities: e.g., after the Message Application Broker download a new "Send Message to" operation will be available selecting a Person in the context.

One of most relevant point of interest in the definition of the context given in the platform, is the Location of a particular contextual resource. Any element in the MIE environment could have its position: a User or a local device will be located in a particular place and even a non-physical context participant, like a Service, could be associated to a certain spot.

To accomplish the goals of MIE Test-bed, the context is modelled to support the concept of "location" information and was demanded to almost every MIE Device to produce

and manage, among the others, the information about its position. The MIE Devices possess a component named Location Framework that is designed as a manager of separate Plugins, each one delegated to a particular localization technology.

3. TECHNOLOGY LEVEL

In the development of the MIE Test-bed, particular attention has to be paid to the localization technique requested to gather position information in a MIE Device. The technology level of the MIE platform structure manages the localization and communications technologies. For each localization technology we developed a Localization Plugin to be used within the Localization Framework.

3.1 WSN Plugin

The technology selected for the localization platform in indoor environments is a WSN where information are exchanged through a RF signal. A small number of sensors play the role of "bridge" toward high level applications or services. These nodes are called sink nodes. In our system, there are nodes deployed in known locations, called *anchor nodes*, whereas the sink nodes are located on the users' terminals in unknown positions due to the user movements. Each sink node, controlled by the localization algorithm, interacts through broadcast signals with anchor nodes to obtain RSSI information from sensors in electromagnetic visibility. The sink node on the user terminal collects the answers from anchor nodes and, at the same time, samples the RSSI of this answers. All anchor nodes are set to the same RF Output Power. The RSSI are sampled individually by each sink node, thus the same packet from an anchor node give a different information to different sink nodes. All RSSI from sensors in electromagnetic visibility are sent to the user terminal as input values for the localization algorithm. Starting from these measurements, localization techniques can provide an estimation of the user position.

The following issues have to be carefully considered:

3.1.1 WSN planning

The deployment of anchor nodes is a key aspect in realistic environments and it has been investigated in order to have a good radio coverage and, at the same time, limiting the number of sensors deployed. The impact of nodes' distribution on the performance of WSN for environmental monitoring is described in [4].

3.1.2 Channel Modeling

The first step of localization algorithms based on RSSI is to map RSSI input into relative distances between the anchor nodes and the sink node. Converting RSSI values into distances is a critical step in indoor environments which have complex electromagnetic propagation. Thus we decided to keep the model of radio channel as simple as possible and, instead, to design robust algorithms that take some kind of error, due to multi-path fading, in the RSSI-to-distance conversions into account. This choice allows us to perform a quick tuning of the channel model when the system is utilized in a particular indoor environments.

Thus, the following relationship between RSSI and the distance has been considered:

$$RSSI = -K_0 - K_1 \log_{10}(d) \quad (1)$$

where $RSSI$ is expressed in dBm and d in meters. This channel model is dependent on the two parameters K_0 and K_1 . We have carried out several experimental measure campaigns and tests in order to match the model (1) to measured data through the identification of the optimal values for K_0 and K_1 at the used radio frequency and particular environments.

3.2 GPS Plugin

The GPS Plugin that has been developed, acts as a driver for the GPS receiver connected to the user terminal and forwards the information to the Location Framework. Due to the characteristic of the device that was used, a Garmin Etrex Vista [5], this plugin was also added with the capability of read and offer information about the current orientation to the Location Framework.

To gather the data from the device, the GPS Plugin was developed to decode the common format used in the GPS devices output: the NMEA 0183 standard [6].

3.3 GPRS/UMTS

In order to grant to the system user interaction with the context as well as mobility in every environment, the technology level manages a wireless communication card with GPRS/UMTS capability. The nature of this technology solution permits a multi-user system with mobile communication capability in both outdoor and indoor environments.

4. ALGORITHM LEVEL

In this level the data collected in the technology level is processed in order to obtain location information useful for the service level. For the GPS technology this processing is done via hardware by the GPS receiver, so location information for the outdoor environments is directly passed to the service level with only a minimal reorganization. On the contrary, data collected from the WSN need more processing to furnish a reliable location information. Locating and tracking mobile users in a indoor environment is a challenging problem. In fact indoor environments are geometrically complex realities from a propagation point of view. Besides, the environment is dynamic too: doors and windows opening and closing, people walking or sitting as well as furniture that can be moved or changed. The combination of complexity and dynamism result in a non-predictable positioning of users. Thus, we realized and tested two algorithm at different complexity both based on RSSI values from a WSN.

4.1 Low-Complexity Deterministic Algorithm

One of the first localization algorithms we realized and tested with our WSN platform is a deterministic algorithm, named Min-Max, characterized by a low computational complexity [7]. The Min-Max algorithm is directly derived from a standard lateration algorithm, following the idea that standard lateration is computationally expensive and not reliable in case of distance estimation errors. In our application of the Min-Max algorithm, the RSSI values are used to find relative distances between the users and the anchor nodes through (1). Thus relative distances are used to construct a square surrounding each point with known coordinates (in our case an anchor node), with the square side length equal to the relative distance from the object to be located, then the intersection of the squares creates a rectangular

box around the object. The bounding-box is obtained as given by

$$[\max(x_k - d_k), \max(y_k - d_k)] \times [\min(x_k + d_k), \min(y_k + d_k)]$$

where k represent the ID of the generic anchor node that answered to the broadcast signal, (x_k, y_k) are the coordinates of the node k , d_k is the relative distance between the node k and the object to be located. The center of the bounding box is chosen as the object position. This algorithm provides an accuracy near to standard triangulation, but reduces computational complexity.

In order to increase the robustness, that is preventing erroneous localizations results, the instantaneous position of user is also estimated considering a certain number of previous user's positions (averaging).

4.2 Bayesian Filtering

The Bayesian filtering offers a powerful mathematical instrument for the location problem through WSN [8]. Here the localization problem is modeled as a dynamic system where the vector state \underline{s}_n , at discrete time n , is given by the coordinates (x_n, y_n) of the user. In particular, the state at time n is

$$\underline{s}_n = f(\underline{s}_{n-1}, \underline{s}_{n-2}, \dots, \underline{s}_1; \underline{L}_n), \quad (2)$$

which is a function of previous estimated positions and actual RSSI vector \underline{r}_n . Specifically, the vector \underline{r}_n is the vector of RSSI in dBm received from the K_n anchor nodes which answer to the broadcast packet sent by the user's node at time n . At time n we can define the following function

$$Bel(\underline{s}_n) \sim p(\underline{r}_n | \underline{s}_n) \Sigma_{\underline{s}_{n-1}} p(\underline{s}_n | \underline{s}_{n-1}) \cdot Bel(\underline{s}_{n-1}) \quad (3)$$

which is called Belief function and represents the a posteriori probability distribution over the random variable \underline{s}_n [8]. Through the Belief function it is possible to identify the most likelihood state at time n among all possible states.

It is possible to characterize two elements that operate in the Belief function. The first one is the *mobility model* $p(\underline{s}_n | \underline{s}_{n-1})$ that represents the dynamic model for the system. It gives the description of variation of the state $\underline{s}_{n-1} \rightarrow \underline{s}_n$, i.e., the statistical description of user movement. The second one is the term $p(\underline{r}_n | \underline{s}_n)$, that provides statistical information on the position at time n starting from the measure vector collected at time n . This is the *perception model* and operates as an updater for the system state. In fact it updates and corrects the forecast obtained from the elaboration of the system dynamic model.

Once properly constructed the mobility and perception models, the Belief function allows to answer the following question: *what is the probability that, having the probability of previous positions and the RSSI vector at current time, the user position is $\underline{s}_n = (x_n, y_n)$?* In other words, it allows to estimate the probability map from which it is possible to extract the most likely estimation of the current state of the system.

By assuming statistical independent RSSI measurements, the equation we adopted to describe the perception model is

$$p(\underline{r}_n | \underline{s}_n) = \prod_{k=1}^{K_n} p(r_n^{(k)} | \underline{s}_n), \quad (4)$$

where $p(r_n^{(k)} | \underline{s}_n) = \mathcal{N}(r(d^{(k)}), \sigma_p^2)$, and $\mathcal{N}(m, \sigma^2)$ represents the Gaussian distribution with mean value m and variance σ^2 , $r(d^{(k)})$ is the expected RSSI in a point at distance $d^{(k)}$

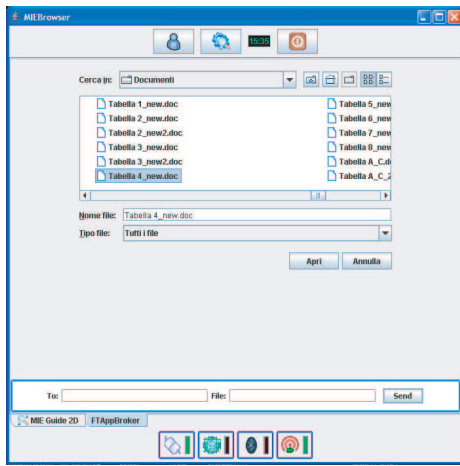


Figure 5: MIE File transfer application broker

adding the “routes” between them: a dialog window allows, then, to set various parameter for the room like, name, type etc.

5.1.2 MIEMessageService

This is an example of client-server service: the Application Broker acts as a simple mail client while the Service Device takes the role of the mail-server.

The service uses a simplified version of the common e-mail protocols (i.e., POP3 and SMTP) and allows some basic operations of mail management: send, get-new, get-all, delete, delete-all. On the server side, the Service Device manages some mailboxes associated to the various service users. The messages are plain text with a limited set of custom tags that offers some text format styles and could incorporate reference to contextual entity too.

5.1.3 MIEFileTransfer

This Application Broker is a simple application that lets the user send and receive files. There is no limitation to the size or type of the file and the recipe could be any other user connected in the MIE context with this Application Broker loaded. Once started, the Application Broker is always active and endlessly waits for new transfer requests: when one arrives, a graphical popup shows some details about the file name, the size and the sender and let the user decide what to do. For file sending a graphical interface, shown in Fig. 5, permits to select the file to send and the destination. The file is chosen through a usual file browsing window, while the destination is selectable from a menu containing the list of context entities.

5.1.4 Other MIE Application Brokers

Many other Application Brokers for the MIE Testbed have been developed by partners within the project; a short overview of some of them is reported here:

Visual Tag - this Application Broker allows to search for new Services in “physical way”: with a video camera is possible to recognize a visual marker placed on ServiceDevice in the surroundings and to identify and download the Application Broker associated to it. This allows to quickly locating and using a service offered by nearby easily recognizable resources (e.g., a printer).

MIEAvatar - this Application Broker integrates with the MIEBrowser and offers a constant online guide to the information shown. It represents a 3D assistant that can speak any text given using a real-time text-to-speech engine. It is capable of different “emotional expressions” which change the look of the avatar allowing more emphasis on the information given. In addition it adopts a priority mechanism to sort the sequence of the messages that has to be spoken: lower priority messages (e.g., new mail arrived) are queued and spoken as soon as possible while higher priority messages (e.g., fire alarm, etc.) are immediately communicated and can interrupt other messages. Any Application Broker loaded in the MIEBrowser can use the avatar by queuing its messages (i.e., the MIEGuide2D gives directions about the route, the MIEFileTransfer informs about incoming requests, etc.).

MIEVisualGuide - This Application Broker shares the task of guiding the user with the MIEGuide2D, but accomplish its duty in another way: it displays the video captured from a local camera and with a “superimposed” directional hint (i.e., an arrow) that points toward the destination of the user route. The arrow can rotate on two axes thus allowing pointing to a destination situated in a lower/higher floor. This Application Broker is meant to be used in conjunction with the MIEGuide2D to offer a complete guide to the MIE environment.

5.2 Ambient Server

This MIE Device has a crucial task for the positioning support in the MIE context. Its role, as previously mentioned, is to act as a bridge between the physical and contextual environment: each Ambient Server has the responsibility to offer information about the physical place assigned and to allow unconnected device to interact with the context.

To obtain the flexibility and accuracy needed in the coverage of the possible MIE physical environment, this entity is specialized in two logical sub-type related to each other in a hierarchical way: the General Ambient Server is the one that manages the outdoor environment, whereas the Local Ambient Server represents an Indoor place (e.g., a building). Local Ambient Servers could be sub-organized to manage smaller places like floors or rooms.

Each Local Ambient Server is a “leaf” in the hierarchical “tree” whereas the General Ambient Server is the root node: at start-up time a Local Ambient Server register itself to the parent Ambient Server, specifying its competence place and additional data. This mechanism of nested competence registration allows the User Device, only knowing the General Ambient Server at start-up, to be always connected to the Ambient Server managing the place it is located in and to switch between various Ambient Server just asking to the current one responsible for any newly gathered User Device position.

The second task of the Ambient Server is to allow the interaction with the MIE context to those devices which cannot perform it by themselves either because of computational or networking limit. This is accomplished in two steps:

- the physical device is connected to the Ambient Server where a program has the duty to manage it by acting as a Service Provider;
- the Ambient Server, at start-up time, loads a simple

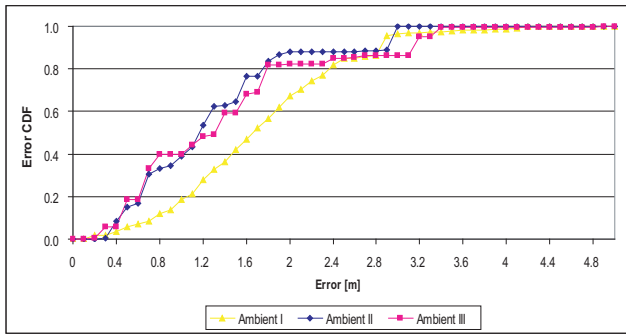


Figure 6: Error CDF comparison for Bayesian filter algorithm in different indoor scenarios

description file produced by the Service Provider and takes care of publishing relative Service information in the context. The device is now accessible as a common MIE Service that could be used either directly or via a specific Application Broker.

6. TESTBED EXPERIMENTATION

During Project VICom, a testing of small-scale version of the MIE Test-bed was developed by the WiLab research group. This demo version was created to help working on the various components of the final MIE Test-bed: in order to test and see the results of the Localization Plugin and of the MIEGuide2D the entire architecture of the User Device (i.e., MIEBrowser, CDM, MIEAdapter, etc.) had to be completely implemented. To keep a low level complexity, the context and the Ambient Server system were substituted by a server, still keeping the semantic of the information exchanged in the “demo context” the same of the final Test-bed. The Service mechanism was the same of the real architecture as well and all Application Brokers developed by the WiLab group was tested on the demo system.

This scaled version of the Test-bed was used to perform some tests on the final demonstration area of the Istituto Superiore delle Comunicazioni, Ministero delle Comunicazioni, in Rome. The data gathered was used to refine the localization algorithms and to estimate the precision needed by the maps; also the coverage of demonstration places was established leading to the decision of using a General Ambient Server and a single Local Ambient Server to manage the two floor involved in the MIE Test-bed: the decision was taken to contain the computational and networking request and also for the limited density of resources in the physical places considered.

At the same time some localization performance tests have been carried out for our indoor localization architecture. The main metric for evaluating the performance of localization techniques is the precision, that we define as the mean square error of the estimate position with respect to the real position. Since the precision has to be maintained in space an time, the performance of a localization system is also characterized by the accuracy, which indicates the rate of localization precision less than a target value for several space-time positions. In Fig. 6 the measured performance of the Bayesian algorithm in different scenarios is shown in terms of accuracy versus precision by reporting the error cumulative distribution function.

As can be noted, a location precision of a couple of meters can be achieved in most (more than 80%) of locations and environments. This is sufficient to support the context aware services developed. Some videos describing the developing and testing of the context-aware platform are available for download in [9].

7. CONCLUSIONS

The demo version of the Mobile Immersive Environment (MIE) Testbed has opened new possibility for the internal research: due to its modular design it can be easily converted to a testing environment for new algorithms and devices in future research project as a multi-user context aware platform for software developing.

In order to permit to the MIE testbed to localize the user in indoor environments we realized an indoor localization system based on a wireless sensor network (WSN). This localization platform has been experimentally tested in on-field experimental campaign through the use of the aforementioned testbed system.

8. ACKNOWLEDGMENTS

This work has been supported, in part, by MIUR (Ministero dell’Istruzione, dell’Università e della Ricerca scientifica) within the project VICOM.

9. REFERENCES

- [1] Virtual Immersive Communications project, MIUR, [Online]. Available: www.vicom-project.it
- [2] K. Muthukrishnan, M. Lijding, and P. J. M. Havinga. “Towards smart surroundings: Enabling techniques and technologies for localization”. In *Thomas Strang and Claudia Linnhoff-Popien, editors, Proceedings of the First International Workshop on Location- and Context-Awareness (LoCA 2005)*, volume 3479 of Lecture Notes in Computer Science, pages 350-362. Springer Verlag, Berlin, May 2005.
- [3] H. O. Shirer, “Preparing now for the future: U.S. civil radionavigation policy and planning” in *Position Location and Navigation Symposium, IEEE 1998* 20-23 April 1998 Page(s):69 - 76
- [4] A. Conti, D. Dardari, “The effects of node spatial distribution on the performance of wireless sensor networks” in *Vehicular Technology Conference, 2004; VTC 2004-Spring*. 2004 IEEE 59th Volume 5, 17-19 May 2004 Page(s):2724-2728 Vol.5
- [5] “Garmin”, [Online]. Available: www.garmin.com
- [6] “NMEA 0183 Standard”, [Online]. Available: www.nmea.org/pub/0183
- [7] Koen Langendoen, Niels Reijers, “Distributed localization in wireless sensor networks: a quantitative comparison”, Faculty of Information Technology and Systems, Delft University of Technology, 2828 CD Delft, Netherlands
- [8] D. Fox, J. Hightower, L. Liao, D. Schulz, G. Borriello, “Bayesian filtering for location estimation”, *Pervasive Computing, IEEE* Volume 2, Issue 3, July-Sept. 2003 Page(s):24 - 33
- [9] “WiLab”, [Online]. Available: <http://www.wilab.org/vicom.php>