Abstract—The emergence of wireless sensor networks (WSN) and RFID technology are starting to make the ubiquitous computing vision a reality, as they provide the means to obtain information about physical phenomena and entities. However, there is a great need for higher abstraction levels, as the integration of these technologies into current ubiquitous computing systems is still primitive. This problem is addressed by TinySOA, a service-oriented architecture that allows programmers to build their applications by using a simple API provided by language independent Web services. TinySOA contributes to the development of ubiquitous computing applications by facilitating the means to obtain primary context mainly through WSN and RFID. We describe the development of an application using our implementation into a production line scenario.

I. INTRODUCTION

In 1991, Mark Weiser coined the phrase: “The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it”. With this phrase he established his vision, in which augmented environments with computing resources provide information and services, when and where they are needed [1]. This vision is known as ubiquitous computing. To achieve this, one of the goals is to “upgrade” the physical world by providing devices that will allow to understand and support the users’ needs.

Recently, there has been an active proliferation of low-power and low-cost devices such as wireless sensor networks and RFID. This devices are starting to make a reality the ubiquitous computing vision. As more of these new devices are introduced, the problem to manage and efficiently analyze the information they generate is starting to gain importance. Also, there is a great need for higher abstraction levels, as the integration of these new technologies into actual application development is still primitive, mainly because of the low level issues involved. This problem is addressed by TinySOA, a service-oriented architecture that allows programmers access to sensor networks and RFID resources from their applications by using a very simple API provided by Web services.

This article is organized as follows: in the next section we briefly describe WSN and RFID technologies and talk about two of the main model approaches in the design of platforms for ubiquitous computing scenarios; in section 3 we detail context awareness; in section 4 we talk about some of the related work; in section 5 we talk about TinySOA, its functionality and components; in section 6 we introduce the TinySOA extension to add RFID technology; in section 7 we talk about the implementation aspects; in section 8 we describe the development of a sample application in a production line scenario using an augmented reality (AR) representation of product boxes; and finally we present some conclusions and future work.

II. UbiComp Infrastructures

Previous research in ubiquitous systems has reached significant advances in the exploration of new applications, establishment of architectures or usability studies. But most of them required considerable personalization by deployment and diagnosis experts, or were developed as proprietary systems. To explore programming models for ubiquitous computing it is essential to improve on productivity, quality and interoperability. There are two models, the context-driven model and the service-oriented model, that have been proposed and studied by many authors and have proved to be good foundations to implement ubiquitous scenarios [2], [3].

From the technologies that provide a good base for ubiquitous computing two can be highlighted: wireless sensor networks and RFID, which are briefly introduced next.

A. Wireless Sensor Networks

WSN are composed by many small devices that integrate a very limited processing unit, a very small storage, a low range radio, and one or more sensor units. These sensors can gather information about temperature, acceleration, barometric pressure, humidity, and so on. These small devices work together to process large amounts of information and to communicate between them until reaching far ending points such as gateways.

B. RFID

Radio frequency identification technology allows to read and write information in very small devices known as tags. The main purpose is to serve as a mean to associate a physical entity with its corresponding digital one. There are two modes of this technology: passive and active. In the passive mode, the functionality is very similar to that provided by barcodes, where the tag must be very near to a reader to allow it to obtain its data. In the active mode, a reader can obtain the tag information at far distances (usually 1-10 meters). The first mode doesn’t need any power source in the tags, as does the second mode. There is a hybrid model which works like the passive one but it allows larger distances.
III. CONTEXT AWARENESS

Context can be defined as “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves” [4]. And context computing is “software that adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time” [5]. If computer access to context is allowed, it can improve the communication and the human-computer interaction, which will make possible to produce more useful computational services.

There are simple context states such as cold, hot, humid, that can be directly measured, known as atomic or primary contexts. More complex, derived contexts (or secondary context) are defined as combinations of atomic contexts.

IV. RELATED WORK

It has been more than 15 years since the ubiquitous computing vision was introduced and it stills looks like science fiction. It would be expected that at this time there would be lots of these scenarios already deployed and working. However, and even though currently there are infrastructures and tools somehow ubiquitous (such as Internet), this vision still doesn’t materialize. Some authors argue that the main barrier is the integration between all these technologies [6].

One of the projects that exemplify this integration is GSN [7], which focuses on the integration with WSN, RFID and wireless cameras. However, it doesn’t provide an external API to allow the development of new applications. Another integration work between WSN and RFID is [8] where they propose an integration model based on a standardized framework to share information between the two technologies.

V. ABOUT TINYSOA

TinySOA’s architecture [9] facilitates the use of wireless sensor networks in traditional application development by using a service-oriented model. Its implementation provides a set of Web services and tools that make possible to send and receive information from a deployed sensor network1.

The architecture consists of 4 main components: TinySOA Node, Gateway, Registry and Server (see Fig. 1). Its operation is as follows: when one of the sensor network nodes turns on, using the Discovery subcomponent it detects and identifies what sensor types are available and announces them by sending a registration message to the Gateway. The Gateway, uses the gathered sensor types to register this information into the Networks Registry subcomponent. This way, the system knows what kind of information the sensor network can provide. Once the sensor types are registered, the Gateway subscribes itself to the services provided by the sensor nodes. When the subscription message arrives to the sensor nodes, they proceed to activate their Sensor Reading subcomponent to start constantly communicating sensed data to the Gateway. Those sensor readings are stored by the Gateway into the Historical Registry subcomponent. The Server uses the Networks Registry to prepare a Web service for each of the sensor networks in the system. Those Web services feature methods to query for readings or to send a maintenance command (e.g. to change the data rate) to the corresponding sensor network. When a query for readings is requested, the Server communicates with the Historical Registry to obtain the requested information and it returns it to the Web services user. TinySOA also supports event detection. For this, the Web services user must send reading values criteria to be detected (e.g. “when the temperature rises to 30°C”).

VI. TINYSOA+RFID

We integrated RFID technology into our current TinySOA implementation. This can allow us to gather primary context that could generate secondary or more complex context. We are using RFID as a context provider because it can be used to identify physical entities. One of the benefits of TinySOA is its component-based structure; this allows the re-deployment of individual components into different devices. In this case, a new component was introduced: RFID Gateway (see Fig. 2). This new component is responsible for the interaction with the RFID hardware allowing the reading and writing of tags. Once integrated, the Networks Registry and the Historical Registry were redesigned to allow the registry of RFID providers and the storage of tag readings. When the Server prepares a Web service for each registered sensor network, it now includes one for each of the available RFID providers. The Fig. 3 shows the new API set for the RFID interaction.

VII. IMPLEMENTATION

The Node component was developed using TinyOS [10]. TinyOS is a component based “operating system” that allows the development of embedded software, specifically, sensor nodes software. All its components are written in nesC, which is an extension of the C programming language. For deployment purposes we used the Crossbow hardware platform [11], which included MicaZ and TelosB motes. The rest

---

1TinySOA is an open-source project available at www.tinysoa.net
of TinySOA’s components were developed using Java [12] and MySQL [13] as the database platform. To implement Web services functionality the XFire toolkit [14] was chosen, because it features low overhead, processing speed and simple services management.

In order to add RFID technology, we used the PIRF-Lite 2.0 middleware, provided by DataBrokers, Inc. [15] as hardware abstraction layer. This allowed us to support several RFID hardware platforms. For deployment, we used a Texas Instruments, Inc. RFID kit [16] and many different RFID tags.

VIII. SAMPLE APPLICATION

When the integration of RFID technology into TinySOA was completed and the implementation ready, a sample application was developed. The objective was to show what can be achieved using WSN and RFID technologies but most important, how an application developer can easily develop ubiquitous computing applications using a well-known paradigm and the language of his choice. The application scenario consists of a production line in which product boxes carry a RFID tag specifying its content and the optimal temperature conditions for the product. The operator using video lenses will look at these boxes and trough augmented reality will see the content of the box, and in its top a bar that displays the current temperature (obtained from a sensor network deployed in the room). Around the temperature bar are two rings that show the maximum and minimum temperatures for optimal product condition. If the temperature is higher or lower than the specified range, an alert will be produced to communicate the problem.

Figure 4 shows a product box with the RFID tag containing what product lies inside and its optimal temperature range. Also, a tracking pattern can be seen on top of the box. This pattern is used by the augmented reality (AR) system to draw a virtual 3D representation of the box. The AR system used in this application was ARToolKit [17]. This toolkit provides a mechanism to draw any OpenGL object [18] on top of a real time video stream, all it needs is a physical pattern to calculate the tracking of the object.

The operator activates the system by passing the box near the RFID reader. The system using TinySOA+RFID, reads the tag information and compares it with the current temperature obtained from the sensor network in the room. After this, a virtual representation of the box showing the product and the range of allowed temperature is drawn (see Fig. 5). The temperature bar can be seen with more detail in Fig. 6. When the current temperature falls above or below the allowed range, the box starts to glow in red.

Besides using TinySOA+RFID for reading tags, it also provides the ability to write information on the tags (actually, a string representation of an object obtained by serialization). To show this feature, a small companion application to allow the operators to write tags with the boxes information was developed.

IX. CONCLUSIONS AND FUTURE WORK

Many platforms and infrastructures have been proposed for the creation of ubiquitous computing applications. However, easy and transparent integration and higher abstraction levels for applications development are still missing. In this article we have introduced our new design and implementation of TinySOA, a framework that allows the development of ubiquitous computing applications through the well-known services
model paradigm. With TinySOA, primary context can be easily acquired through wireless sensor networks and RFID tags, and this context can then be used in applications to derive more complex context. As proof of concept, we have developed a sample application combining RFID and WSN into a production line scenario, showing how different hardware platforms can be transparently integrated into applications, using the language that the developer is used to.

We are currently working on evolving the concepts behind TinySOA to support different technologies, devices, and computation services. The main objective is to provide a high level of abstraction to propose solutions to problems in ubiquitous computing such as immediate event notification, inter-device communication, and others not currently handled by our current implementation of TinySOA.

ACKNOWLEDGMENT

Financial support for this project was provided by the Mexican Council for Science and Technology (CONACyT).

REFERENCES