LCCAReader: Software Proposal for Ubiquitous Applications Through RFID Systems

Samuel Souto[1], Emanoel Costa[1], David Beserra[1], Ramon Nascimento[1], Alberto Araujo[1]

[1]Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)
Av. Bom Pastor, s/n, Boa Vista – 55.292-270 – Garanhuns – PE – Brasil

samuel.romeiro@gmail.com, emanoel.rc@hotmail.com, {dw.beserra, ramonsantos.pe, albertoaepa}@gmail.com

Abstract. *The use of RFID technology for identification has grown in many parts of the world. Its applications cover different economic areas ranging from student cards, inventory management until control robotic systems. But the difficulty in finding software commonly used to control RFID systems has also increasing. In this paper we present the development stages of software able to communicate with UHF RFID's systems offering functions able to remedy difficulties encountered in the literature and commercial applications. The results will be used in developing applications on identification and location of objects.*

1. Introduction

Pervasive computing can be defined as the computing branch who is present in an environment, interacts with the ambience as if it was part of it, but it is not perceived when used [Santini, 2008]. Intertwined with this context, arise the Ubiquitous computing (UbiComp), defined as the integration of mobility with distributed systems mostly, intelligent and highly integrated with computers and their applications for the benefit of final users, becoming invisible for those them. [Weiser, 1991] [Satyanarayanan, 2001].

One of the technologies that are strongly related to these concepts are those based on radio frequency communication. RFID stands for Radio Frequency Identification. Basically it uses radio waves to perform the communication between a moving object, containing labels, commonly called *Tags*, and a reader [Santini, 2008]. RFID have been gaining ground on the world stage as identifiers that use radio waves to collect, modify and transmit information ensuring a rapid flow of data in a short period of time.

RFID readers are classified according to their frequency band of operation. Its application will have relationship with that characteristic. Thus, there are LF readers (Low Frequency readers) which have a frequency range of 30-300 kHz and are commonly used for identifying animals and close reading of items with high water content. HF (High Frequency readers), operating in a frequency range of 3-30 MHz and used for access control to buildings. UHF (Ultra High Frequency readers) operating in the range of 300MHz-3GHz and used to identify boxes and cases. Finally, Microwave

having a frequency range of 3GHz and used for vehicle identification [Glover &. Bhatt. 2007].

One of the most important components in an RFID system is its software. This component is responsible for managing the information trafficked by the duty cycle of RFID. Such software should work as a middleware. That is, provide a communication interface between the reader and user with maximum possible transparency. The interface of an interactive system comprises the entire portion of the software in which the user holds conceptual or physical contact during its use [Barbosa and Silva, 2010].

In theory, the user doesn‟t need to possess a working knowledge about the RFID system components, but does need to know how to operate the software.

However it is not difficult to find testimonials from people talking about how hard is to work with software responsible of hardware management, such as done with RFID systems. This difficulty is caused by several factors. Either by the deployment, having as main objective, specific functions of the equipment used, either by excessive restrictions on the functionality of the system. Develop a software interface with technical languages hinders the software dissemination capable of hardware controlling systems between users with little or, even, none specific knowledge for this computing branch.

The main goal of this paper is to present the stages of development software called LCCAReader. Such system is capable of performing communication with radio frequency identification readers and able to remedy difficulties encountered in the literature and commercial applications where RFID plays an essential role in the management and data transferring.

2. Related Works

In the literature we can find some article involving the development of software for use in hardware systems, such as RFID, and applications that are similar to the proposed paper. Trindade & Sobrinho (2008) presented the stages of software deployment able to control tires. The authors also show its application. In the research it was proposed to use the system for assets control by the carriers. Trying to avoid malicious exchange of new tires for old and worn. The operation of this system is achieved by the *Tag* identification located on the tires of trucks and monitoring the status of them through a database.

Many studies have been conducted in order to seek innovative and creative applications for RFID. An example is the intelligent use of RFID as done by Luimula *et al.*, (2010). This research was conducted in three different stages and was used RFID to control robots through identification of ways that the robotic system should make, only using the computer and the RFID system to control the machine. The survey showed all stages of development an RFID system, from conception (literature review), deployment of a robotic control system, until the effective control of the robotic platform. The results showed that these systems can be used to control robots efficiently.

To achieve the goal of developing management software for hardware, such as the purpose of this paper, RFID equipment analysis involved in its deployment is required. In literature we can find some work involving this topic. Ramakrishnan (2003) verified the performance of *Tags* identified by passive UHF readers having as protocol

identification EPC (Electronic Product Code), in different environments. Including measurement of maximum reading distance, orientation sensitivity, read rate and performance variation. In the results, Ramakrishnan concluded that the system, used by him, recorded a reading beam on average 18 meters.

All studies previously reported have shown results that make possible the development of software whose purpose is to cater to a context or specific application. Even though the studies in the literature showing us good results about the integration between hardware and software, the difficulty relative to location of researches involving the development of ubiquitous computing systems, more specifically talking about RFID systems, enabling its use for other needs or welcome other realities is large and stumble into obstacles that slow your progress. Therefore, such questioning was used as motivating agent for improvement of found studies and development new research related to these topics.

3. LCCAReader: Design and Development.

Most of the computing area and, in particular, the sub-area of software engineering, is interested in building interactive systems more efficient, robust, error-free and easy maintenance [Barbosa & Silva, 2010].

The LCCAReader, name of the proposed software, is a system that has as main feature the communication between personal computers (PC) and UHF RFID devices, manufactured by Think Magic Company. This software has to perform this information exchange using the RS232 port of the computer. This is because the reader that has greater identification range, used in this study, identifying the world's most used radio frequency's communication protocol (EPC Global), is the Mercury 5e, manufactured by this company. The system also uses an API that provides communication with the serial port described above and consequently the reader desired.

Observed the key attributes required by other RFID systems, the software should have the following features: restricting access to the system, data and objects registration, user registration, RFID readers communication (produced by Think Magic) and reading/writing *Tags*. The description of each function is given below:

- **User Registration**: Registration of all users who have access to the system. During the registration, will be asked some information, such as password and login name
- **Restricting Access to the System:** Being an extension of the functionality „User Registration", the user will need to supply username and his password to proceed with system access.
- **Data and Object Registration:** The LCCAReader will have, as one of its proposal, the function to record data and objects which is related to a *Tag* number. As the user performs the identification of these labels, the information contained in the database will be displayed on the screen for him.
- **RFID Readers Communication:** The proposed system has the capability to perform communication with the reader desired, provided that it is compatible with the software developed.
- **Reading/Writing *Tags*:** Besides the function of communicating with the reader, the system can identify *Tags* and submit information, if any. Another important feature

- is the writing *Tags*. If the *Tag* identified is for the type Active (able to store data about an object within it), the system will provide an interface to perform data update contained within this *Tag*.
- **Graphic Interface:** Interface in which the user will requirehis actions to the system.
- **Database Communication:** The proposed software will have access to the database in which user's data, objects' data and *Tags*' data will be stored and available.

We used, as process of software development, the method Incremental Development. In this method the system is divided and developed in parts (called increments). Immediately after the specifying software requirements for each increment, the architecture of the system is designed, that is, their behavior is specified such as call model function, distribution of package, among others. Following the flow model for structuring software, we have the stage of development and validation. All increments are validated to be integrated with the rest of the system. If there is any error in the validation of the system altogether, the whole process is redirected to the stage of development. This cycle continues until all the software meets the initial system requirements.

In order to improve the understanding of the entire system, we developed some UML diagrams (Unified Modeling Language). UML is a set of notations that are intended to support the modeling of object-oriented systems, providing a partial representation of the system. We develop two diagrams that enabled a greater understanding of the LCCAReader's functioning. They are Use Case diagram and Sequence diagram.

The sequence diagram (Figure 1) represents the sequence of the system's global behavior detailing all the steps that a particular user needs to run to access or require any function. In the figure, we can see that it's necessary present information to the system in order to validate this access for a given user.

The Use Case diagram (Figure 2) shows the functions that can be controlled and managed by the user. We can quote as an example, the role of "Managing *Tags*". This functionality according to figure 2, comprise remove, insert, update, and retrieval data associated with the *Tag* numbers recognized by the system.

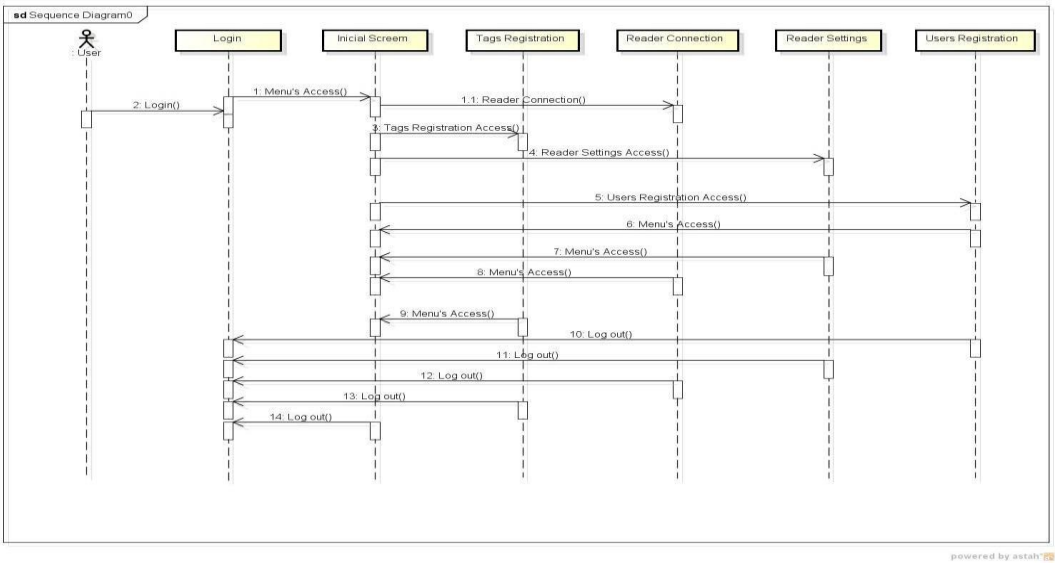

Figure 1. Sequence Diagram

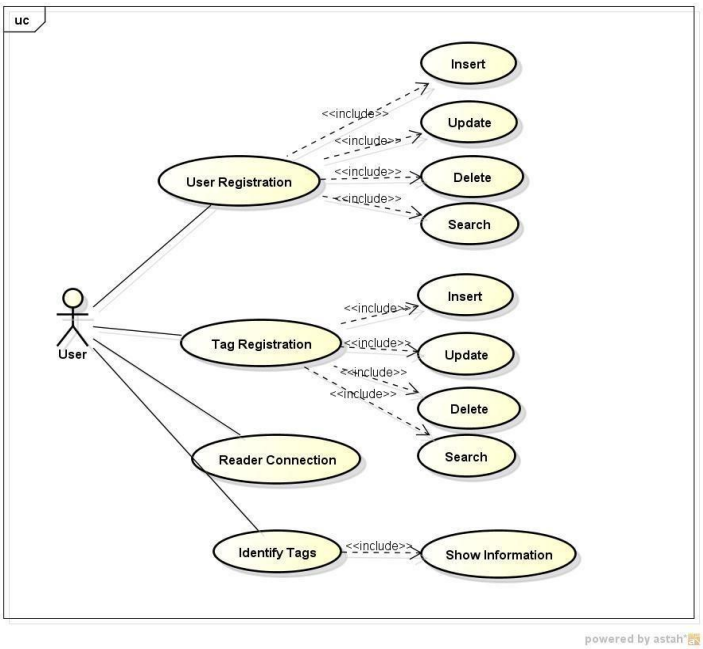

Figure 2. Use Case Diagram

The LCCAReader was developed using the programming platform NetBeans IDE 7.0.1 and Java Development Kit (JDK) 1.7_01. The communication with the reader Mercury 5e was based on a Think Magic's library equipment. The serial communication functions were implemented using the C programming language, We decided to do that because of the fact that this language is more flexible for information exchange with the RS-232 port than the Java language.

The development was divided into four increments representing all four main system functionality: user registration, *Tag* registration, RFID Readers communication and read/write *Tags*. All increments were developed in parallel, properly validated and finally incorporated into the system source code. For a better understanding, we divided the software development in four primary steps described below.

- **Serial Communication**: This phase was focused on develop an algorithm capable of performing serial communication with external devices. The algorithm execute the following steps: first it's looked for a directory with all the DLLs (functions used by programs to perform tasks) necessary to communicate with the RS-232 port. This is required so the software can continue with its execution. Soon after this check, the code makes a call to system native methods (functions implemented using C language) and provides the information exchange through the serial port. If this process is not, by some reason, running, failures in connection with the port may be one of the reasons for this error, the algorithm returns a message stating the problems and possible causes of the error.
- **RFID Reader Connection:** Main function of the LCCAReader software, the algorithm of the connection with the reader has the following operating cycle: a Java class named 'Reader' is called and instantiated. In order for this to occur, it's necessary the instantiation of one specific method named "create" (contained within this class). This function requires a parameter to execute its role; this variable is the URI, port address where the reader is located.
- Thereupon, there's a call for the method 'conect' (also contained within the class 'Reader'). This new incoming call is performed so that the software can achieve the communication with the reader. If the answer is positive the algorithm makes another call for method, this time for one named 'paramSet'. Such function required as parameter, the region to be identified. Think Magic's manufacturer readers claim a specific region ID to perform the identification. After these processes, the reader should be connected successfully. If there is any error during the process, the execution is stopped and an error message appears.
- **Reading/Writing *Tags*:** The reading/writing *Tags* algorithm, subdivision of the connection code with the reader showed above, was developed in this increment. After making sure that the reader is connected, this being the first step of the algorithm, the method 'read' (implemented in class 'Reader' discussed earlier), is called passing, as parameter, the reading time. If the reader is not connected an error message is sent to the user alerting him about event. For the accomplishment of writing, the process is the same. As the information that will be displayed on the screen include data contained within the *Tags*, if the user wants to change them, just edit the desired field and the algorithm makes sure that the changes will be saved.
- **Graphic Interface:** The graphical interfaces developed for the software is presented through the figures 3, 4, 5 and 6. The figure 3 shows the screen corresponding to the function of *Tag* registration. Once the user accesses this screen you can navigate through sub-functions related to registration functions, such as insertion, update and search for *Tag*s.

The LCCAReader allows the user to edit some connection information related to the reader. This function is illustrated through figure 4. In this function the user can update the name of the reader as well as specifying what type of *Tags* the system can identify. The main function of the software is shown in Figure 5: reader connection. At first, this frame will show the possibility of running a function: communicate with some reader. At the time that this communication is established, other functions will be

presented to the user such as reading *Tags*, information verification in the database, among others.

Finally, the figure 6 shows the Unit Tests for registration functions present in the software. Through these tests it was concluded that the system meets all the functions satisfactorily, since, during the execution of the evaluation, the database was being monitored, as well as the execution of each command.

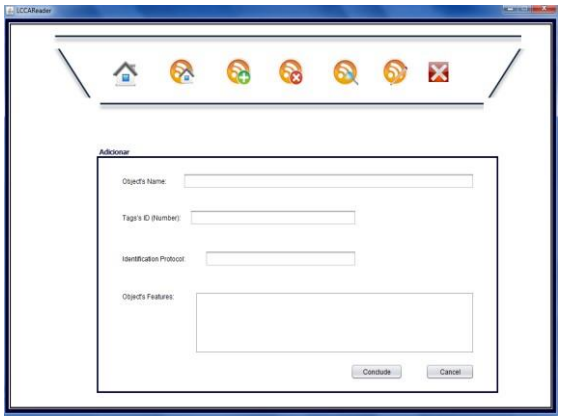

Figure 3. Tag registration screen

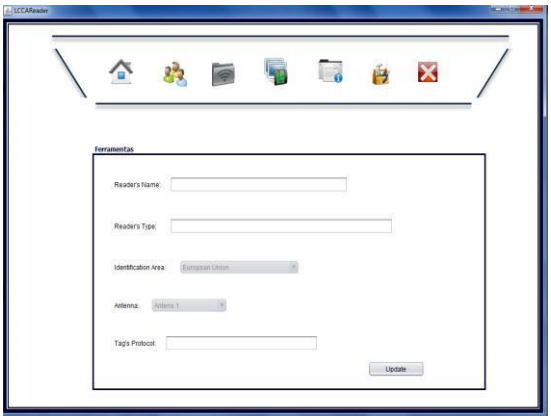

Figure 4. Setting

Figure 5. RFID Reader communication screen

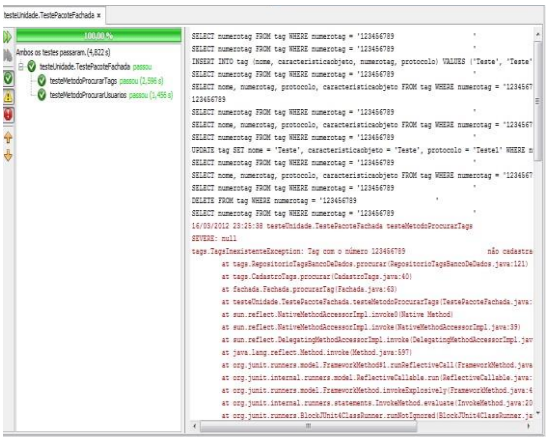

Figure 6. Unit tests screen

4. Final Thoughts

In this work was presented all stages of development of software capable of performing the communication interface with an RFID system respecting usability requirements necessary to ensure communication between users and system. During the planning and requirements gathering, was considered others RFID software with specific applications and even software available by manufacturers of readers. This research allowed us to have clarity of running good software for this computing branch.

As development process, we adopt the incremental method. This decision was made based on the fact that the RFID hardware uses various artifice, in a way, independent of each other. We adopt the parallel development of software. In this type of development, the deployment of the increments is performed simultaneously.

Therefore, the incremental method didn't delay the development process, on the contrary, has showed an improvement, since the validation of a test served to increase the system completely. The validation of a given increment, for example, the *Tags* registration has the same results when the system is fully tested. This feature is possible because the software includes modules for communication, in a way, independent of each other. What communicates all portions of the system is the access to the database. In this way we can advance the implementation schedule.

On the other hand, the system programming presented difficulties on its communication reader step. This occurred because the reader, chosen by us, has protocols for information exchanging through serial port of the computer and doesn't have knowing algorithms messaging. It was necessary to study the entire API used by this equipment, searching for knowledge of its operation. this problem identified took us the need to divide this increment in development units even smaller than we usually did with the aim of reducing the problem in minimum increments and find individual solutions for the errors identified. The LCCAReader presented a satisfactory response to events it has undergone. At the moment the system is in the testing phase and usability issues such as efficiency of algorithms should be remedied in future versions of the software.

The difficulties in the adoption of RFID systems exist, but they are not insurmountable, says Wu et al. (2005). Like most technologies increasing nowadays, is simply a matter of time until the promises of RFID becoming a real thing.

References

Barbosa, S. D. J. And Silva, B. S. (2010). “Interação Humano-Computador”**.** Rio de Janeiro: Editora campus.

Glover, B., and Bhatt, H. (2007) “Fundamentos de RFID”. Rio de Janeiro: Alta Books. Luimula, M., Sääskilahti, K., Partala, T., Pieskä, S., Alaspää, J. (2010) “Remote navegation of a mobile robot in an RFID-augmented environment”. Pers. Ubiquit Comput.

Ramakrishnan, K. N. M. (2003) “Benchmarks for Passive UHF RFID *Tags*”. College of Engineering, Guindy – Anna University, Chennai, Indiana.

Santini, A. G. (2008) “RFID Radio Frequency Identification”. Rio de Janeiro: Ciência Moderna.

Satyanarayanan, M. (2001). “Pervasive Computing: Vision and Challenges”, http://www.cs.cmu.edu/~aura/docdir/pcs01.pdf, July.

Sonntag, N. L. Barbosa, D. N. F. Barbosa, J. L. V. Pinto, S. C. C. S. (2012). “Gerenciador de Objetos de Aprendizagem para um Ambiente de Educação

Ubíqua”. Biblioteca Digital Brasileira de Computação - BDBComp. Avaliable in: <http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=8654>. March.

ThinkMagic (2007) “Mercury4/5 User Guide: (TM-M4/5-X-INSTALL) Rev. 2.4.5”, http://thinkmagic.com, July.

Trindade, J. A., Sobrinho, E. (2008) “Sistema RFID: Controle de Pneus”. Revista de Engenharia da Computação, v. 4, n. 1. 2008.

Wang, S. W., Chen, W. H., Ong, C. S., Liu, L., Chuang, Y. W. (2006) "RFID application in hospitals: A case study on a demonstration RFID project in a Taiwan hospital" System Sciences, in HICSS '06. *Proceedings of the 39th Annual Hawaii International Conference. Volume 8.*

Weiser, M. (1991) "The Computer for the 21st Century". Scientific American, pages 94-104.

Wu, N. C. Nystrom, M. A. Lin, H. C. Y**.** "Challenges to global RFID adoption". (2005). Avaliable in: <http://www.sciencedirect.com>. March.