Mobile Architecture for Identifying users in Ubiquitous Environments Focused on Percontrol

Valderi R. Q. Leithardt[1,2,3], Guilherme A. Borges[1], Iván M. Carrera[1], Anubis G. M. Rossetto[1], Carlos O. Rolim[1], Cláudio F. R. Geyer[1]
[1] Institute of Informatics, GPPD, UFRGS
[2] Institute of Technology Senai - POA, FATEC
[3] University of the Vale do Rio do Sinos, UNISINOS
Porto Alegre, RS, Brazil
{vrqleithardt, gaborges, ivan.carrera, anubis.rossetto, carlos.oberdan, geyer}
@inf.ufrgs.br

David Nunes, Jorge Sá Silva
University of Coimbra
Coimbra, Portugal
dsnunes@student.dei.uc.pt; sasilva@dei.uc.pt

***Abstract*—In pervasive and ubiquitous environments, the process of identifying users can be done using various devices and techniques like radio frequency identifiers (RFID), bluetooth devices, smartphones and others. In this context, it is necessary to implement management techniques and control of these environments, as well as customize the environment characteristics to meet user preferences.**

In this work, we present a solution for control and manage various users, devices and environments. We base our research on Percontrol (a system for pervasive user management), which was only intended to identify users using Wi-Fi, and now it is capable of managing temperature, luminosity and other preferences, measured by a Wireless Sensor Network (WSN) embedded to Percontrol, and the data treatment is done by a Artificial Neural Network (ANN).

***Keywords*—Ubiquitous Communication; Intelligent applications; Artificial Neural Networks**

I. INTRODUCTION

In 1991, Mark Weiser [1] claimed that the most useful technologies would become so integrated onto our daily lives that it would be as if they just vanished, since using them would become second-nature. With this idea in mind, Weiser proposed a new form of computation, known as pervasive computation. By vanishing, it is meant that the device becomes imperceptible, i.e., the user no longer needs to worry about the technological resource in use. From this perspective, pervasive computing frees users from worrying about secondary tasks and allows them to focus on their primary task.

The development of microelectronics and wireless communications has been for great benefit for Wireless Sensor Networks (WSNs), in a pervasive/ubiquitous environment. WSNs are usually composed by a large number of wireless nodes spread through an area where the events or phenomenon to be monitored is located [2]. A sensor node is a device with reduced size meant to be cheap and easily deployed in risky locations or with difficult access. The small size of the nodes severely limits their hardware capabilities; sensors usually present many limitations with their battery, processing power, storage and communication interfaces [3]. These restrictions with sensor nodes affect the amount of nodes present in a given network, and said network's performance, leading to a need for developing communication protocols that could handle these limitations while being adequate for the application and its associated routing mechanism.

In these pervasive/ubiquitous environments, routing is a task handled by the sensors themselves and not managed by a specific routing device. Sensors can carry data throughout the network in an power efficient way through wireless radios, or light pulses cooperating with each other. Sensorial data is transmitted throughout the network in hops, i.e., a sensor transmits its data to the node that is closer to final recipient of the data, trusting that the transmission will have no interruptions. This transmission mechanism is useful to distribute the transmission's energy costs throughout the whole network, avoiding the need to use high power signals [2]. This makes the use of WSNs in pervasive/ubiquitous environments a more complex challenge, since these environments are populated with many other types of devices and communication mechanisms [1]. All this diversity in devices and communication types is managed by the pervasive/ubiquitous application; an example this kind of application is Percontrol [4], a pervasive system for controlling user attendance through mobile devices and wireless networks.

The main contribution of this work is an analysis of the viability to use Artificial Neural Networs (ANN), as an improvement of a previous work [5], by using a base technique to handle user profiles and manage devices on a pervasive environment also using WSNs to capture environmental data. The results from this analysis are innovative and serve as initiative to further research. The present work is organized as follows: section II presents the Percontrol system and introduces several important concepts regarding ANNs and WSNs. Section III describes some concepts of routing in WSNs. Section IV gives an overview of the state-of-the-art in the area while Section V presents the application scenario. Section VI contains information about an example of use and shows some preliminary results. Finally, Section VII gives a conclusion and alludes to future work.

II. PERCONTROL

Percontrol, as described in [4] and [5], is a system that automatically manages and keeps track of user attendance.

The system detects the entrance and exit of users within an academic or business environment. Percontrol also improves user discovery and localization service, within the local environment based on bluetooth, Wi-Fi and RFID identifiers.

In an academic environment, Percontrol monitors student attendance without teacher's intervention, and in a business environment, it controls the entrance and exit times of employees without the need of their intervention. By resorting to certain aspects of pervasive computing, the system developed in [5] can guarantee reliability and transparency in the detection and localization of users, resulting in minimal impact on both environments. The system identifies users through their mobile devices and controls the environment based on their previously-defined profiles. The user/device presence on a certain location generates an event, which is sent to a WebService listener application. The listener application first identifies the user within the environment and then configures the venue in accordance to the user's profile stored in a database. Environmental data such as temperature and luminosity are acquired through SunSpot wireless sensors [6] in conjunction with Arduino development kits [7].

In the work described in [5], an earlier version of Percontrol was already being developed, but the management of different user profiles and devices on a single environment was yet to be supported. To support this feature, an artificial intelligence module was necessary, which is now presented by using Artificial Neural Networks. Thanks to its new-found intelligence, the system is now capable of adapting the environment to fit the demands of several users, simultaneously. The ANNs receive data from each user, perform pattern recognition and, by resorting to previous event data, make a decision to configure the environment in a way that satisfies most users.

Solving problems with ANNs is an attractive approach. According to [8], the way these problems are internally represented in the network and the natural parallelism associated with ANNs architecture create the possibility of superior performance when compared with traditional machine-learning methods.

In ANNs, the usual procedure for problem-solving begins with a learning phase, where a set of examples is presented to the network from which are extracted the necessary characteristics to correctly represent the information provided [9]. These characteristics are then used to generate answers to the problem in question. An ANN is composed by several simple processing units connected through communication channels that have a certain associated cost associated. Each processing unit performs computational operations on the fed data that through its connections. The intelligent behavior resulting from an ANN comes from the interactions between the network's processing units.

The most important property of ANNs is the ability to learn from their environment and improve their own performance. The entire knowledge contained within the network is stored in the links between the networks artificial neurons; in other words, the networks behavior is mostly conditioned by the weights associated to the connections between individual neurons. The development of a computational solution that not only infers a system's current state but also predicts its future condition seems to be the solution for the environment management problem previously mentioned.

Considering the characteristics metioned above, a neural network is a prediction mechanism that seems to fit Percontrol needs since it can use raw contextual data, learn from it and predict configuration values that will better fit a future situation. The solution proposed by Silva [10] was used as basis to adapt a neural network to the current application scenario. This effort demanded an analysis of the characteristics of WSNs in order to find the best configuration and techniques to associate ANNs parameters to Percontrol's WSNs.

The main components of sensor networks are the phenomenon, the sensor and the observer. The phenomenon is the event of interest to an observer, and it is monitored and processed by a WSN. Several different phenomena can be observed concurrently by a single WSN. Monitoring is performed under certain criteria determining the performance requirements for the nodes to be used. The sensor device is used for collecting the data generated by the phenomenon and transmit it to the next node through wireless communication. Nodes respond to changes in the environment where they are currently in operation, e.g., climatic and eviromental changes. The characteristics and capabilities of the sensor depend on the application and each node's sensibility may differ, depending on the distance to or the exposure time to the event being monitored. The observer is the element that receives the information collected by the WSN; depending on the application's needs, there may be more than one observer at any given time [11]. WSNs can be classified according to their configuration, sensing mechanisms, communication media and processing type [2]. The type of routing is one of the most important research topics for reducing energy consumption and increase performance in wireless sensor applications.

III. ROUTING

Due to the specific features of their utilization, (e.g., their limited battery life), communication between nodes deployed in an external environment is subjected to many obstacles that may stop its effectiveness. Most applications that use WSNs communication are directed towards configurations with stationary characteristics, which differ from traditional ad-hoc networks; routing can have its focus on addresses or data, the latter being a more recent approach, which main advantage is the fact that data aggregation manages to reduce the amount of packets exchanged in the network, and thus, reduce the overall energy consumption of the system [12].

Another approach for routing in WSNs is related to the TinyOS operating system, developed by the University of Berkeley. TinyOS does not use a complete addressing scheme like traditional IP-based ad-hoc networks [12], [13]. Traditional ad-hoc networks implement a routing protocol that constructs a Minimum Spanning Tree from an initial sink or access point [13]. Routing in WSNs can be categorized as: based on Medium Access Control (MAC), plane routing, hierarchical routing and geographical routing. There is also Geographical and Energy Aware Routing (GEAR) and Greedy Perimeter Stateless Routing (GPSR).

For this project, SunSpot wireless sensors were used for the tests. The SunSpot sensors use the ZigBee specification for wireless protocols, which use Ad hoc On-Demand Distance Vector Routing, that can be categorized as based on MAC.

In this section, we only mention the categories of routing protocols used in WSNs. An exhaustive and comparative description of routing protocols for WSNs is presented in [14], which was used as theoretical basis to formulate the tests and obtain the preliminary results that serve as validation for the present work.

IV. State-of-the-art

As previously mentioned, this article's proposal refers to an implementation of ANNs in a pervasive control system for pervasive/ubiquitous environments in conjunction with WSNs (composed by SunSpot wireless sensors for testing purposes). The system deals with concurrent data and adapts the environment's configuration to match its users' profile, according to the devices used to perform login onto the environment.

In [15], we can find several academic works that focus on the detection of devices in many application areas within the domain of pervasive systems and wireless sensor networks, as introduced by Mark Weiser. The work in [15] shows a system that implements several integrated systems that allow the monitoring of patients in their own home; however, it does not address the integration of different forms of communication within the same environment nor does it present mechanisms for handling user profiles. In [8], an extensive proposal that focus on elder users in ubiquitous environments is presented; the Internet is shown as a form of interaction but this work does not specify which languages, devices and forms of communication should be used. In [16], it was proposed a project that consisted on the creation of an artificial intelligence system capable of leading a robot through certain routes. The system was tested using a robot vehicle with freedom of horizontal movements. The environment was scanned using sensors where data were fed to the artificial intelligence system, and where an algorithm evaluated the situation and acted on the locomotion engines, making the robot avoid possible obstacles and follow the desired route.

Despite being an innovative proposal, a target audience and usage scenario are not defined, therefore not adapting to the proposed model. Fonseca [17] defines a solution that uses neural networks and RFID, partially achieving the desired results.

Percontrol presents all necessary features for pervasive/ubiquitous environment usage, considering different types of devices, and forms of communication. This work presents a significant improvement over previous versions in [4], and [5], while addressing all the primary objectives previously set. Table 1 presents the main characteristics and functionalities of works researched in the available literature.

V. Application Scenario

The application scenario shows the potential pervasive/ubiquitous computation for improving efficiency in workplaces. It also attempts to illustrate different possible perspectives one can have on a single pervasive scenario. Initial versions of Percontrol did not anticipate the use of WSNs or ANNs [6]; such versions only intended to automatize student attendance tasks in classrooms. Percontrol focuses on performing automatic detection of people that enters an environment.

This work proposes an extension of the work developed in [6], and [7], increasing the pervasive functionalities available in this user-tracking system, with the objective of increasing control over environmental conditions through user's mobile devices. Using SunSpot wireless sensors [14] and Arduino kits [7], Percontrol can sense and manage the temperature and luminosity of an environment; and by using ANNs, the system can also attempt to adjust the values of these environmental properties to fit user preferences and the number of people in the environment, turning it into an intelligent location.

The sequence diagram in Fig. 1 shows the primary interactions between all parts of the system, as well as the messages exchanged since a user is detected until the environment adapts to its preferences. When the application detects the entrance of a device in the environment, a Web service that manages the associations between users and devices is accessed. The device is identified through its BDA (Bluetooth Device Address), Wi-Fi or RFID. The application maintains a module called BlueID which holds a list of all devices that were ever detected. Each time the application verifies the devices currently present in the environment, it performs a comparison with the previously stored list; newly detected devices generate an "entry" event while missing devices are associated with an "exit" event.

When accessed, the webservice returns the username to the application, and also associated device resources and personal preferences through the HTTP protocol and an XML format message. The application also communicates with the SunSpot sensors to fetch the room temperature, luminosity, humidity or other environmental data that may be used at a later time. The following format was used to communicate with the sensor: messageID : sensorTypenn. Both messageID and sensorType are numerical values. The messageID field is used to associate sensor response with the respective BlueID request, an important step since communication is asynchronous.

The sensorType represents the type of data being sent (luminosity, humidity, etc); the "\n" character is used to mark the end of a message, while the ":" character is used as a data separator. The current environmental state is compared with user's environmental preferences in order to decide the needed changes to be done. After a decision is reached, the environment sends commands to the actuator controllers, connected through USB to an operating computer, to change the environmental characteristics (e.g., turning the A/C unit on and change the room temperature). The extension of Percontrol's functionalities translated into a more complex architecture shown in Fig. 2.

Initially, the prototype application and its respective transmitters were tested with a Windows operating system, an environment that benefited from the use of SunSpot sensors [6] and Arduino hardware [7]. There were many other advantages that led us to choose the Arduino boards, namely the embedded input/output ports, low cost and strong modularity. The main idea behind the use of Arduinos was to test their viability for middleware development in pervasive environments, not excluding the possibility of having these boards completely replace several individual sensors for an integrated, single-

 ISBN: 978-1-61208-289-9

TABLE I. COMPARISON BETWEEN RELATED RESEARCH

Support and Control Works	Scope	Devices Transmission
ARAUJO	ubiquitous computing	Not defined
CARVALHO	ubiquitous computing	Environmental sensors, measuring devices, electronic devices, Internet
RIBEIRO	Neural Networks	Sensor end-of-course and sonar
FONSECA	Neural Networks, pervasive systems	RFID
PERCONTROL	Neural Networks, systems, pervasive/ubiquitous	Wi-Fi/ RFID / Bluetooth / Wireless Sensor Networks

Support and Control Works	Public	Routing Protocol	Language
ARAUJO	Not defined	Not defined	Not defined
CARVALHO	Elderly people who require constant monitoring of their health	Not defined	Not defined
RIBEIRO	Not defined	Not defined	C/C++/C#, Java
FONSECA	Laboratory of the University	Not defined	Delphi
PERCONTROL	Educational and business environments	ZigBee	.Net

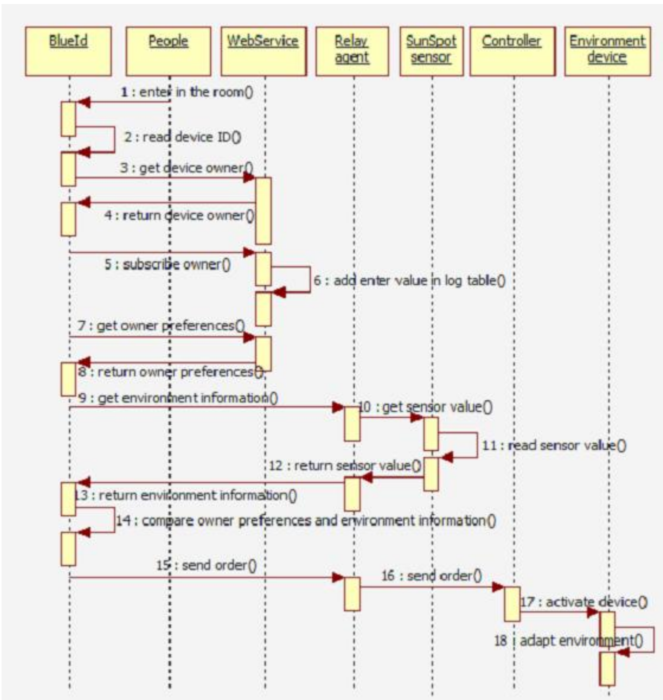

Fig. 1. Sequence Diagram

board solution connected to a computer. Fig. 3 illustrates the operation of Percontrol.

From Fig. 3, it can be seen that a central controller is missing for allowing the exchange of profiles in the environment, according to user preferences. The main challenge here is controlling the number of parameters generated by the application while using WSNs and ANNs; the number of parameters increase with the amount of nodes and this means larger energetic and resource demands, as well as an more complex neural network processing, which may compromise the ANN's response time.

In the previous version of Percontrol [5], there was no control of the different ways for communicate and indentify users. As it is shown in Fig. 3, it is possible to visualize that the used can be identified through wireless communication protocols, and later this information is treated individually, based on parameters defined by the ANN. Said parameters can be definitions of user preferences as luminosity, temperature and the actual user indentification in the environment. In Fig. 3, there also can be seen the utilization of temperature

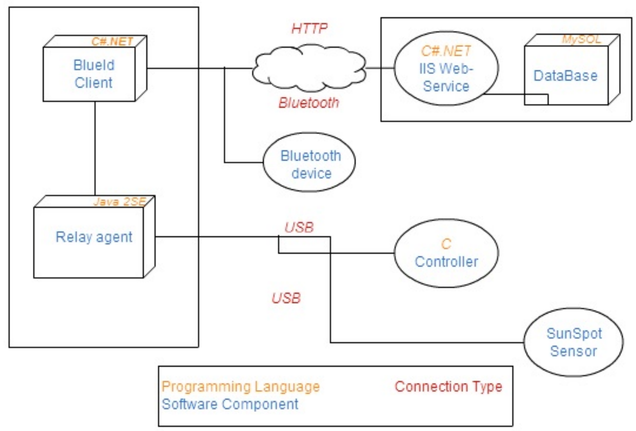

Fig. 2. System's Architecture

controller, which has access to the air conditioning system. It is possible also to update the location of several users in different environments, by sending messages to a REST web server that stores users and environment data and preferences.

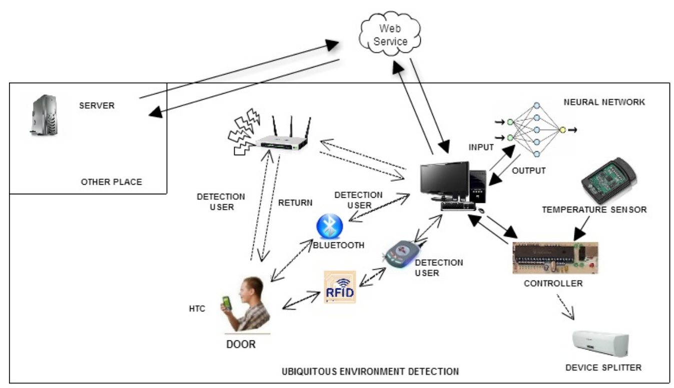

Fig. 3. Functioning of Percontrol

An efficient monitoring of the network performance is necessary to guarantee a good quality of service. An example of this can be observed in the amount of time necessary to obtain information regarding a monitored environment; if it takes too long to obtain environmental information, this information may lose its value from an application perspective. Management of performance may provide means for the application to define proper quality metrics. These may be influenced by node density, exposition time, amount of dissipated energy and other factors, as specified in the description of the study presented in section III.

A mechanism that evaluates the level of importance of information is necessary for the management of quality of service. For example, a sensor detecting a temperature of 20° C during spring is a normal occurrence, but a the measurement of 50° C under the same circumstances is an abnormal event, which would turn it into a relevant situation that would require extra attention; it would implicate an artificial intelligence mechanism that could compare the abnormal value with other measurted values by other sensors to see if the information is reliable and determine the proper course of action. Information that is of great importance to the normal function of the system should imply a greater effort for proper delivery. That is, energy consumed in communications should vary depending on the importance of the data.

Another relevant management aspect concerns the installation of ad-hoc networks in unknown areas, where the behavior of wireless communications can be highly unpredictable, with high error-rates and considerable delays, which may compromise the value of the information provided to the application. Performance management usually includes quality assurance, performance monitoring, control and analysis [2]. The QoS management process begins with the detection of performance degradation and ends when the source of the problem is ceased or removed. In between, the process has many intermediary stages of situation analysis [18]. Initially there were used only 2 Sunspot tests kits containing two wireless sensor nodes communicating with each base station connected to the computer via USB. With that comes the need to conduct a comparative study of routing protocols for use in different environments and with numbers over wireless sensors. To this end, several techniques exist to treat this problem and also allied service discovery, one of the most important, by the Service Location Protocol (SLP) [14], which basically consists of maintaining a directory that contains the services available to whom it is offering them to. However, it is necessary to study thoroughly the operation of routing protocols in order to verify the protocol that best fits the pervasive control system, it is not in the scope of this work the study of routing protocols. Therefore, for this work there were conducted only some tests to validate the survey and obtained results that demonstrate the feasibility of work and their implementation and use with Percontrol, contributing to the improvement of the system and data so that other researchers can use it.

VI. Example of use and preliminary results

One issue when having multiple users on the same system is the problem of concurrent data; e.g., the configuration of an air-conditioning unit may be influenced by every user that registers in this environment, since each user might have its own preferred temperature, and the temperature itself is general throughout the whole environment. In order to bypass this problem, the decision-making process for selecting the best "average" temperature must take into account the individual preferences from all users within the environment. A widely used solution [10][19][20] that has shown great results is the use of AI, in particular ANNs [9]. A neural network bases itself on real data that has occurred in the past and has been stored within the system for posterior access and use.

The main objective of this work is not the choice of proper protocols or AI tools, but the creation of novel help mechanisms for Percontrol. Our choice for an AI mechanism dwelt on neural networks, while communication mechanisms were TCP/IP and ZigBee. These choices are supported by published works in routing protocols [14], artificial intelligence [20], and comparison and use of neural networks [21] and [22].

The neural network loads the entire history of a device being handled within the environment, and uses its historical data as training, in order to identify decision patterns that were assumed in a recent past. Considering our air-conditioning example, these patterns include the temperature that each user wants for a certain environment and what temperature was actually used when all users were taken into consideration. This type of analysis is crucial for the network's decision-making process. Fig. 4 presents part of the source code used to define the desired and assumed temperatures. These values are fixed for testing, but in a real scenario they are fetched from a database or an archive.

```
// Initializes with
//   1 randon
//   3 perception neurons (entries)
//   6 hidden layer neurons
//   1 output neuron (output)
net.Initialize(1,3,6,1);
// Learning rate
net.LearningRate = 3;
// iterations
iterations = 10000;
// Train the network
net.Train(input, output, TrainingType.BackPropagation,
iterations);
```

Fig. 4. Part of the neural network's source code

The code shown in Fig. 4 is used to train the neural network. After training phase the next step is to test the network to determine if it is well-suited to solve the problem of finding the ideal temperature using past event data; in order to perform the testing, a graphical interface was developed. The interface receives the values for current data and returns the ideal temperature estimation, as shown in Fig. 5.

```
private void Treina()
{
        int iterations;
        double[][] input, output;
        #region Initialization

        // Instantiates the network
        net = new NeuralNet();

        // Assemble the historic Network for Train-there
        input = new double[5][];
        input[0] = new double[] { .30, .25, .27 }; // A very cold day
        input[1] = new double[] { .17, .18, .19 }; // A hot day
        input[2] = new double[] { .16, .20, .22 }; // A cold day
        input[3] = new double[] { .18, .19, .24 }; // A cold day
        input[4] = new double[] { .26, .28, .30 }; // A very cold day

        output = new double[5][];
        output[0] = new double[] { .26 }; // expected temperature
        output[1] = new double[] { .18 }; // expected temperature
        output[2] = new double[] { .20 }; // expected temperature
        output[3] = new double[] { .22 }; // expected temperature
        output[4] = new double[] { .27 }; // expected temperature

        #enderegion
```

Fig. 5. Training the ANN

After the neural network's training, we could identify the network's response time after a user enters the environment, as shown in Fig. 6.

Fig. 6 presents response times of the ANN for the cases with 1, 3 and 10 distinct users identified by the pervasive

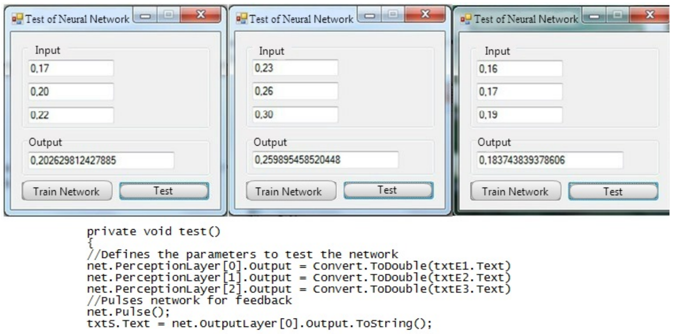

Fig. 6. Neural network's performance and response time

environment, where X axis represents the number of users and Y axis represents the elapsed time. With this prototype we obtained not very satisfactory results, for example, for a single user, the ANN took 3 seconds to process the information contained in the user's profile, returning an average temperature with a value equal to the one defined by the user (since it is just a single person). For three users, the neural network took 5 seconds to respond, and for ten users it took 13 seconds. In Fig. 7, a screenshot shows information on the users identified by the system, as well as on the devices associated with them.

To perform the identification of different environments we used an Arduino Duemilanove [7]. It is a microcontroller board based on ATmega328, with 14 digital input/output pins, 6 analog inputs, a 16MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The board contains the necessary assets to support the microcontroller and its use is as simple as connecting it to a computer with a USB cable or powering it with an AC-to-DC adapter or battery.

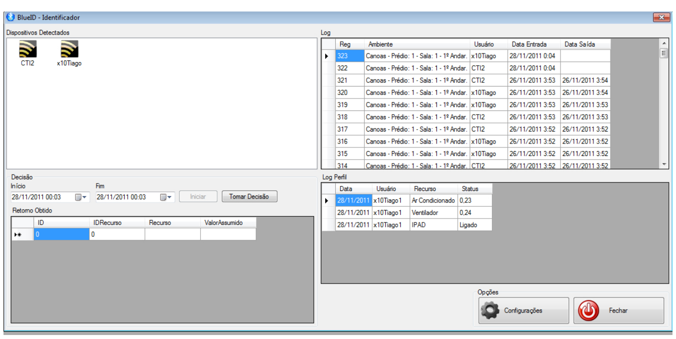

Fig. 7. User identification screen

With this board it was possible to detect devices via Bluetooth, Wi-Fi and, after being integrated with an appropriate card reader, RFID. The reader fetched the RFID a card's serial number that can be cross matched with the user's registration on the database. For this purpose, a RFID card reader model YHY502CTG was used in conjunction with the Arduino board. After obtaining the necessary application data and performing the necessary adjustments the system aws validated using a didactic MultiPIC development Kit, which possesses its own internal programmer. The didactic development Kit was connected to a stepper motor that simulated a ventilator. The stepper motor can be put in action with different speeds; initially we defined 3 different speeds that corresponded to 3 different profiles. Fig. 8 shows a picture of the assembled device.

Fig. 8. Validation equipment

The software used for simulating a ventilator with 3 speeds and controlling the board and stepper motor was developed in C language using the development software from Microchip MTLab, and transferred to the microcontroller with the IC-Prog 1.06C, the software used to compile the source code onto the MultiPIC kit's processor.

On the performed tests the ANNs computed the average temperature from the user profiles and used current environmental information from the SunSpot sensors to correctly manage the ventilation system. From these tests we conclude that Percontrol managed the pervasive environment in a satisfactory manner and that the primary objectives of this research were met, although there is still much room for improvements.

VII. Conclusion and Future Work

The choice of a routing protocol should be based on the restrictions inherent to the observed phenomena, which defines the monitoring environment and minimum requirements for the sensoring hardware. The main purpose of the work and results herein presented are to be used as basis for future research work that improves the area of pervasive/ubiquitous computation, namely in the use of ANNs with low response times in environments with thousands of users.

The performed tests confirmed the viability of device detection with Wi-Fi, Bluetooth and RFID, an improvement over previous Percontrol versions. Nevertheless, there is still some latency in registering new devices on the system, which may be reduced by further adjustments of the parameters sent to the ANNs.

The main contribution of this work is an improvement of [5], using an artificial intelligence technique, as well as treating several users in environments with heterogeneous devices. Using ANNs made possible to automatically treat user profiles, which is a new feature of Percontrol. This work also represents a significant contribution since it covers different areas and technologies within pervasive computation. On top of several parameters and definitions still waiting implementation, future work needs to define a model and implement a mechanism for privacy control, for both users, their devices as well as for the pervasive environment itself.

VIII. ACKNOWLEDGMENTS

This work is associated with the UbiArch project - Ubiquitous Architecture for Context Management and Application Development of the Federal University of Rio Grande do Sul. We would like to thank the Conselho Nacional de Desenvolvimento Cientifico e Técnologico CNPq for its support. We would also like to thank Sun Microsystems for donating the SunSpot development kits, which allowed us to perform the necessary tests for our study. We kindly thank the students Sebastien Skorupski, Thiago Tomasel, Marceli Stein and Jeison Nunes da Silva, who contributed towards the completion of this work.

REFERENCES

[1] M. Weiser, "The computer for the twenty-first century," *Scientific American*, pp. 94–10, Sep 1991.

[2] L. Ruiz, J. Nogueira, and A. A. F. Loureiro, "Manna: a management architecture for wireless sensor networks," *Communications Magazine, IEEE*, vol. 41, no. 2, pp. 116–125, 2003.

[3] K. Kwon, M. Ha, T. Kim, S. H. Kim, and D. Kim, "The stateless point to point routing protocol based on shortcut tree routing algorithm for ip-wsn," in *Internet of Things (IOT), 2012 3rd International Conference on the*, 2012, pp. 167–174.

[4] A. G. M. Rossetto, C. O. Rolim, V. R. Q. Leithardt, M. A. R. Dantas, and C. F. R. Geyer, "An adaptive fault tolerance approach to enhance the execution of applications on multi-cluster grid configurations from mobile grid interfaces in wireless networks." *IJHPSA*, vol. 3, no. 4, pp. 202–215, 2011.

[5] V. Leithardt, C. Rolim, A. Rossetto, C. Geyer, M. Dantas, J. Silva, and D. Nunes, "Percontrol: A pervasive system for educational environments," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*, 30 2012, Feb. 2 2012, pp. 131–136.

[6] O. Labs, "Sun spot world," May 2013, last Accessed May 2013. [Online]. Available: http://www.sunspotworld.com

[7] A. BRASIL, "Arduino brasil," Apr. 2013, last Accessed April 2013. [Online]. Available: http://www.arduinobrasil.com/

[8] S. T. Carvalho, A. Copetti, and O. G. Loques, "Sistema de computação ubíqua na assistência domiciliar á saúde," *Journal of Health Informatics (JHI)*, vol. 3, no. 2, pp. 51–57, 2011.

[9] C. O. Rolim, A. G. Rossetto, V. R. Leithardt, and C. F. Geyer, "Analysis of a hybrid neural network as underlying mechanism for a situation prediction engine," *Journal of Applied Computing Research*, vol. 2, no. 1, pp. 22–31, 2012.

[10] R. Silva, V. R. Leithardt, J. S. Silva, C. Geyer, J. Rodrigues, and F. Boavida, "A comparison of approaches to node and service discovery in 6lowpan wireless sensor networks," in *Proceedings of the 5th ACM symposium on QoS and security for wireless and mobile networks*. ACM, Oct 2009, pp. 44–49.

[11] C. H. P. Augusto, C. B. Carvalho, M. W. R. da Silva, and J. F. de Rezende, "Reuse: A combined routing and link scheduling mechanism for wireless mesh networks," *Computer Communications*, vol. 34, no. 18, pp. 2207–2216, 2011.

[12] S.-H. Lee, D.-W. Kum, K.-W. Lee, and Y.-Z. Cho, "Energy-efficient data-aware routing protocol for wireless sensor networks," in *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2007. UBICOMM '07. International Conference on*, 2007, pp. 101–106.

[13] A. T. A. Gomes, A. Ziviani, L. Lima, and M. Endler, "Dichotomy: A resource discovery and scheduling protocol for multihop ad hoc mobile grids," in *Cluster Computing and the Grid, 2007. CCGRID 2007. Seventh IEEE International Symposium on*, 2007, pp. 719–724.

[14] W. Khan, N. Saad, and M. Aalsalem, "An overview of evaluation metrics for routing protocols in wireless sensor networks," in *Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on*, vol. 2, 2012, pp. 588–593.

[15] R. B. ARAUJO, "Computação ubíqua: Princípios, tecnologias e desafios. departamento de computação – universidade federal de s. carlos (ufscar)," Sep. 2003, last Accessed September 2013. [Online]. Available: http://im.ufba.br/pub/MAT570FG/LivroseArtigos/045_AraujoRB.pdf

[16] D. R. d. S. Ribeiro, K. L. d. S. Machado, and L. C. A. Perereira, "Rede neural perceptron embarcada em robótica," *Engenharia de Computacão em Revista*, p. 1, 2008.

[17] L. O. Fonseca, "Sistema de localização rfid de usuários visando à racionalização de energia em ambientes inteligentes," *Universidade de Brasília*, p. 1, Feb 2011.

[18] V. Pereira, J. Silva, J. Granjal, R. Silva, E. Monteiro, and Q. Pan, "A taxonomy of wireless sensor networks with qos," in *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*, Feb 2011, pp. 1 –4.

[19] A. de Pádua Braga, *Redes neurais artificiais: teoria e aplicações*. LTC, 2007. [Online]. Available: http://books.google.com.br/books?id=R-p1GwAACAAJ

[20] H. Qiao, J. Peng, Z.-B. Xu, and B. Zhang, "A reference model approach to stability analysis of neural networks," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 33, no. 6, pp. 925–936, 2003.

[21] R. Silva, J. S. Silva, C. Geyer, V. Leithard, and F. Boavida, "Use of gps and 6lowpan in mobile multi-sink wireless sensor networks - issues and feasibility," in *8th International Information and telecommunication Technologies Symposium (I2TS)*. IEEE R9 Latim America, 2009, pp. 154–160.

[22] C. O. Rolim, F. Schubert, A. G. Rossetto, V. R. Leithardt, C. F. Geyer, and C. B. Westphall, "Comparison of a multi output adaptative neuro-fuzzy inference system (manfis) and multi layer perceptron (mlp) in cloud computing provisioning," *X Workshop de Computação em Grade e Aplicações - WCGA*, 2012.