A FRAMEWORK FOR USING REAL DATA WITH DISTRIBUTED LOW COST SENSORS

4 AUTHORS:

N. Dias
Polytechnic Institute of Porto

Daniela Campos
Polytechnic Institute of Porto

A. Dias
Instituto Superior de Engenharia do Porto

Hugo Almeida Ferreira
Polytechnic Institute of Porto

A FRAMEWORK FOR USING REAL DATA WITH DISTRIBUTED LOW COST SENSORS

N. Dias, D. Campos, A. Dias, H. Ferreira

School of Management and Industrial Studies - Porto Polytechnic (PORTUGAL)
ndias@eu.ipp.pt, danielacampos@eu.ipp.pt, adias@eu.ipp.pt, hugo.ferreira@eu.ipp.pt

Abstract

Currently, due to the widespread use of computers and the internet, students are trading libraries for the World Wide Web and laboratories with simulation programs. In most courses, simulators are made available to students and can be used to proof theoretical results or to test a developing hardware/product.

Although this is an interesting solution: low cost, easy and fast way to perform some courses work, it has indeed major disadvantages. As everything is currently being done with/in a computer, the students are loosing the "feel" of the real values of the magnitudes.

For instance in engineering studies, and mainly in the first years, students need to learn electronics, algorithmic, mathematics and physics. All of these areas can use numerical analysis software, simulation software or spreadsheets and in the majority of the cases data used is either simulated or random numbers, but real data could be used instead. For example, if a course uses numerical analysis software and needs a dataset, the students can learn to manipulate arrays. Also, when using the spreadsheets to build graphics, instead of using a random table, students could use a real dataset based, for instance, in the room temperature and its variation across the day.

In this work we present a framework which uses a simple interface allowing it to be used by different courses where the computers are the teaching/learning process in order to give a more realistic feeling to students by using real data.

A framework is proposed based on a set of low cost sensors for different physical magnitudes, e.g. temperature, light, wind speed, which are connected to a central server, that the students have access with an Ethernet protocol or are connected directly to the student computer/laptop. These sensors use the communication ports available such as: serial ports, parallel ports, Ethernet or Universal Serial Bus (USB).

Since a central server is used, the students are encouraged to use sensor values results in their different courses and consequently in different types of software such as: numerical analysis tools, spreadsheets or simply inside any programming language when a dataset is needed. In order to do this, small pieces of hardware were developed containing at least one sensor using different types of computer communication.

As long as the sensors are attached in a server connected to the internet, these tools can also be shared between different schools. This allows sensors that aren't available in a determined school to be used by getting the values from other places that are sharing them.

Another remark is that students in the more advanced years and (theoretically) more know how, can use the courses that have some affinities with electronic development to build new sensor pieces and expand the framework further.

The final solution provided is very interesting, low cost, simple to develop, allowing flexibility of resources by using the same materials in several courses bringing real world data into the students computer works.

Keywords: sensor networks, teaching tools

1 INTRODUCTION

The widespread of computers gave rise to a growing attention of many available software tools by professors and students, placing the computer as an important tool in education.

The engineering courses are no exception, there is no engineering course that can dispense the use of a computer nowadays. Although this can be very useful and time saving, using only computers *as is* could lead to a simple virtualization of everything. Currently, a lot of experiences of using computer simulators [1],[2],[3] and software for virtualizing [4],[5],[6] almost everything have achieved great success and can improve classes (speaking in terms of both the students and the professor point of view). When teaching/learning engineering, where mainly the students are asked to solve problems, using completely virtual environments can lead students to loose touch with reality and its pitfalls.

Although the focus of this work was engineering students, it can easily be extended to other students.

Engineering students must always have a notion of reality: which king of magnitudes some physical system has, what type of precision the values have or how long does a communications between two systems takes. Nowadays, when students need to some practical work for statistics, they usually use random generated values obtained with a numeric analysis tool or, when studying databases, in most cases they simulate a theoretical example presented by the professor. On top of all this, during Programming course study the students are faced with a machine (computer) which has some connectors such as USB, Parallel port, Serial port, etc. which are kept unknown for the majority of the students.

Our goal was to introduce a framework that will allow the teaching/learning process to be more in touch with reality. Therefore, by using real data the theoretical examples become real problems and students can relate to how their courses can be used in their future life.

Our main concern was to introduce low cost sensors in the computer studies cycle by developing simple pieces of hardware that are connected into a computer communication port. With limited funding, only a low cost solution can grow in future years, and the idea is to have some of the hardware being developed by the students in more advanced years.

This work keeps the computer in the teaching/learning process by extending its potential in a simple way and giving a more realistic feeling instead of a uniquely virtual one.

The paper is organized as follows: Section 2 describes the system architecture and Section 3 presents the experimental setup. Section 4 provides some conclusions and Section 5 discusses future work.

2 SYSTEM ARCHITECTURE

2.1 Overview

The Figure 1 presents our system's general architecture. Mainly, there is a central server that can have several types of sensor units connected. The values of these sensors are accessed by a socket communication from a student computer (remote access). In some cases, the sensor units can be connected directly to the student computer (locally connected) when the objective is to learn how to interact with the different type of computer communication ports.

There are other systems that already store this type of information and even work in real time such as PacHube [7]. This system is solely a huge database that can be used by anyone but does not put the emphasis in neither the hardware nor the learning mechanisms that are inherent to our framework.

Even though, in the case of remote access, students do not have access to the hardware and not necessarily need to know about how the data is acquired, when using real data instead of random generated values also implies that the example used in the exercise/assignment will be adjusted to the data and not a theoretical example made up by the professor. This has advantages and pitfalls, it gets the students (especially in engineering) interest but it is not a controlled example and the data can give unexpected results.

Also, it can reinforce students the idea of the usefulness of the courses syllabus when using it in real practical examples.

One other aspect is that it can interconnect the courses and give a sense of interdisciplinary which is so often desired and difficult to obtain. This can be further explored by having the students develop sensor hardware in more advanced electronic courses.

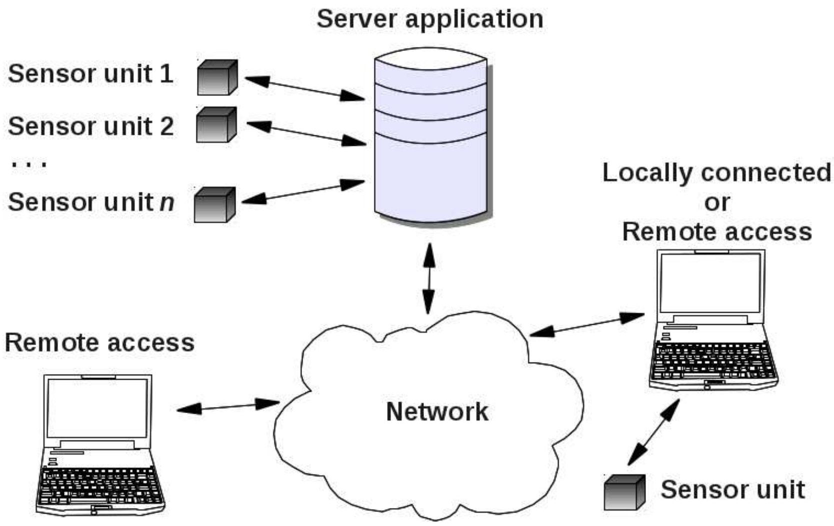

Figure 1: System Architecture

2.2 Sensor Units

Sensor precision is not our main objective, therefore the sensors used are low cost and easy to find on any electronic components store, such as Light Dependent Resistors (LDR) and LM25 temperature sensors. Besides, the sensors' interface to the computers should be very easy to do. We must keep in mind that sensors cannot be connected directly to the communication ports so, more hardware is needed.

The hardware for the sensor units can have small differences from sensor to sensor but in its essence remains equal.

The Figure 2 represents a block diagram of the sensor reading hardware. Currently, we are using several types of microcontrollers. The main differences in the schematic are related to the sensor output.

The sensors used have the following outputs:

- I2C – Inter-Integrated Circuit;
- RS232 – Serial Communication;
- SPI – Serial Programming Interface;
- Analog – A/D converter.

The sensors that have analog output are usually connected directly to an Analog/Digital Converter (ADC) passing trough a signal conditioning circuit. The interface to the computer is normally done with the serial interface directly or using a serial to USB converter. The parallel connection is normally used when the students have the sensor units connected directly in their computers. Although the name refers to a single sensor, each Unit Sensor can have more than one sensor attached.

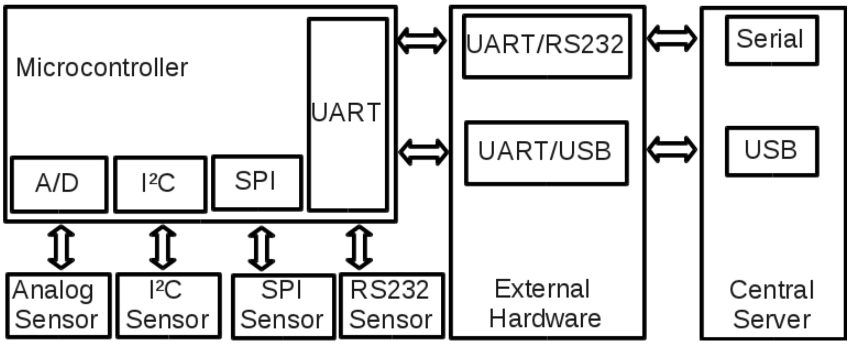

Figure 2: Generic hardware block diagram

Currently the hardware implementation used for the sensor units is a small Arduino [8] board based on the AVR architecture [9]. Tests were made with three types of sensors: temperature, humidity and carbon monoxide connected to the microcontroller ADC which respectively sends the data via RS232 communication port (serial communication). We are planning to use another microcontroller architecture, namely ARM [10] in order to provide a different test bed for the students. In order to keep the framework running, the sensor type and the microcontroller architecture are not important.

2.3 Central server

The central server is a machine running a Linux Operating system. On top of this operating system, a daemon was developed and is responsible for reading the sensor values and write them into a MySQL database. Figure 3 represents the data sensor flow from the hardware to the users.

In order to access the sensor data by Hyper Text Transfer Protocol (HTTP). A web page, based on Hypertext Preprocessor (PHP), was developed. This page (see Figure 4) displays all the sensors connected to the server, the last result from each sensor reading and has the option for download the sensor data into a file.

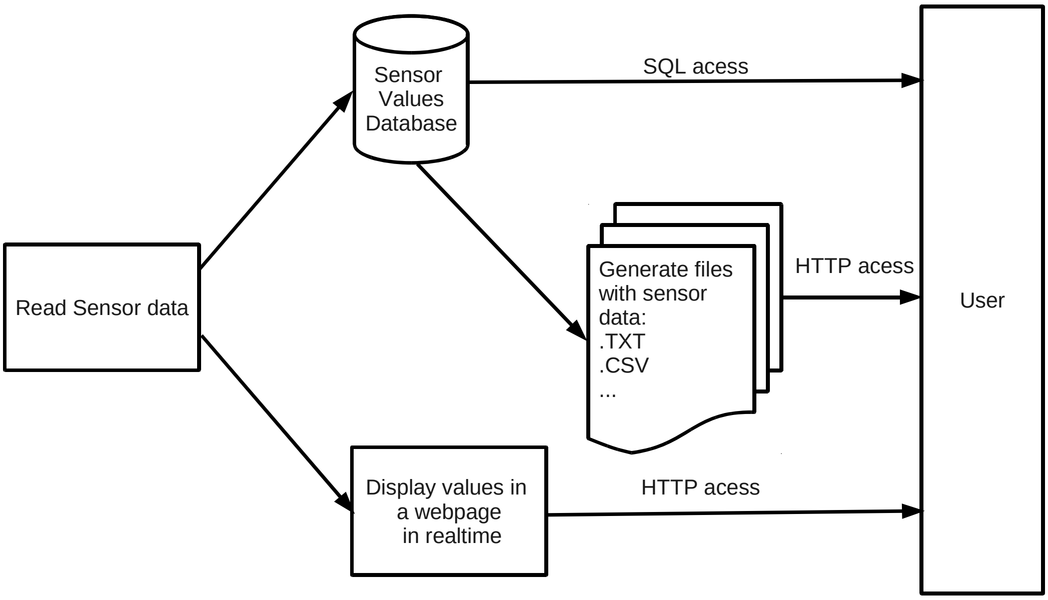

Figure 3: Sensor data flow from hardware to user

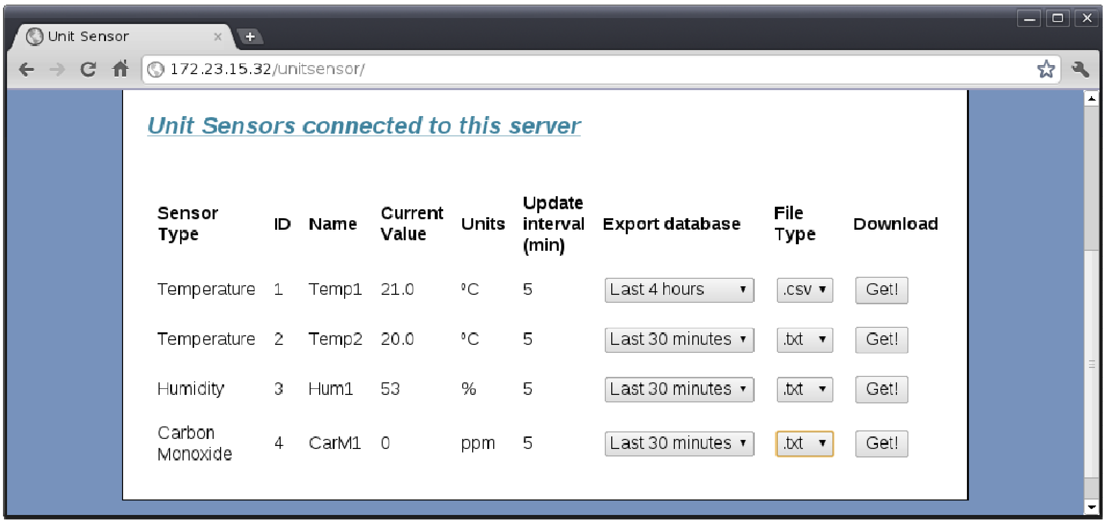

Figure 4: Screenshot from the Unit Sensor web page

3 PRACTICAL CASES

We present three cases that exemplify some of the potential of the framework.

The first case explains how a numerical analysis tool can be used in order to study some dataset. In the second case the sensors database is used in programming course problem and in the last case we use the sensor data to learn the basis of a typical spreadsheet.

3.1 Case I - Numerical analysis tool

Using datasets of any kind inside numerical analysis tools are extremely important in several engineering courses. It is needed in statistical analysis, numerical analysis and physics among others. Although, sometimes, the dataset must be “perfect” in order to work, which can be achieved by using the 'random' command and specifying, for instance, that the values must be zero mean Gaussian white noise.

With the exception of these cases, all that is need is a dataset, so either by using a 'random' command or any other solution is just fine.

With the sensor database that is provided by this system we intend to replace those random data with real measured data. This can be done by simply download the data files directly from the sensor unit web server page or, if the type of software is largely used, a toolbox can be used to read directly from the database.

3.2 Case II - Database connection

One of the typical topics of programming courses is databases. When teaching databases it is important to learn about: how to connect to a database, the use of filters, search options and so on.

The sensor data can be used in the practical classes by the students to achieve the majority of topics.

Moreover during assignments development this database can be used to fulfil the students’ needs in several ways. For instance, they can use the available information and integrate a decision layer in the code by using state conditions. These decision layers can be:

- room temperature control;
- luminosity control;
- Heating, ventilation, and air conditioning (HVAC) system.

More examples can be added, the main question here is that the available information in the database allows the students to evaluate and define state conditions off-line or in real-time with such database information. Figure 5 represents the two tables that we use to save the sensor values and that can be used directly by the students. In order too keep things working the users have only permissions for reading the database.

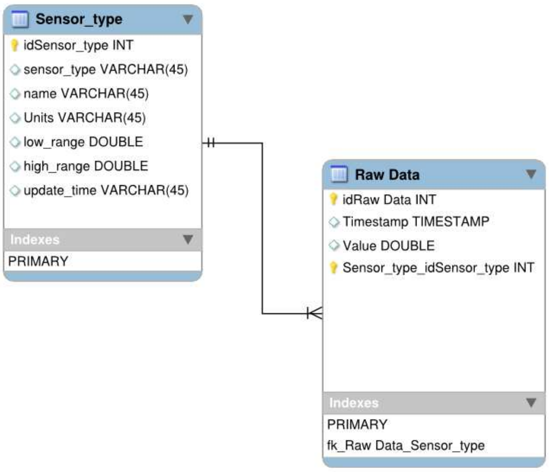

Figure 5: Sensor data normalized database

3.3 Case III - Spreadsheet use

In courses where the use of spreadsheets is taught, it is a normal procedure to generate random values in order to fill the cells that will be used to apply formulas, generate graphics and so on.

This framework doesn't teach how to use the spreadsheet, but helps students by bringing real data values to the cells so that the learning process can have meaning.

Let's use a very simple example by using a sensor unit with temperature readings. In order to learn how to use a spreadsheet, the work that the students have to implement is to plot the temperature for the last 4 hours and achieve the respective median value.

After exporting the values from the server where the sensor units are attached, the students just have to use the spreadsheet to obtain the temperature plot such as represented in the Figure 6.

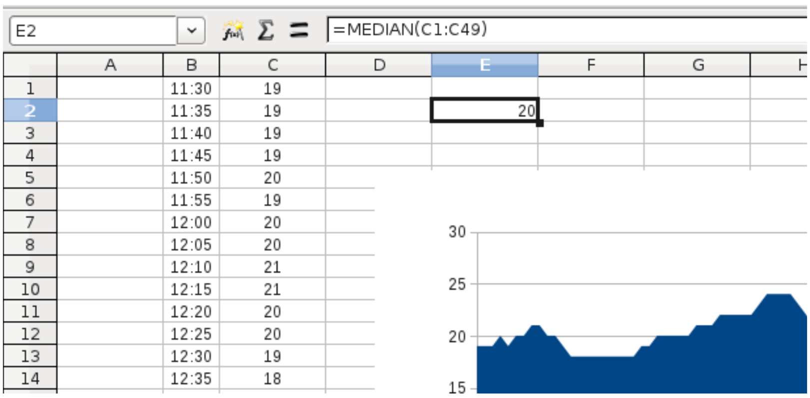

Figure 6: Spreadsheet use example

Although the temperature value doesn't bring anything new, at least it's variation during the last four hours does. For instance, if this sensor was plugged in a server inside the same classroom as the students are, they can immediately understand how the room temperature as ranged before and after the computers were tuned on, by knowing the exact amplitude vale. This feeling is real and could be achieved without any major effort.

4 CONCLUSIONS

A generic architecture was used to allow the students to use real data in computer based work. This allows professor to replace random generated data with real data and give assignments with real examples instead of theoretical ones. Students find their work more interested and their courses more related to their future work environment.

The framework proposed has several types of communications data output in order to respond to the different requirements of the several courses.

This is a work in progress and the impact of the final solution has not been fully measured yet, however, the use of sensor units has already been used with promising results.

5 FUTURE WORK

Students in more advance years can further participate, in the scope of electronic courses, by developing themselves other unit sensors that can be used within this framework.

The web page displaying the sensor information should be improved based on the user needs.

It is also planned to study a possible implementation that extends this work to actuators (LED, motors, etc) instead of just sensors.

A final remark to our main goal, which is to have a set of servers with different unit sensors running at different schools, having a large community with different types of sensor, each school can be used by others to remotely access data, for instance, from sensors that they don't have in their own.

6 ACKNOWLEDGEMENTS

The authors would like to thank the School of Management and Industrial Studies and Porto Polytechnic. This work is sponsored by ESEIG and IPP.

REFERENCES

[1] Dias A., Dias N., Campos D. and Ferreira H. (2010). Autonomous Guided Vehicles Applied to Industrial Engineering and Management Studies. Proc. RiE 2010 – International Conference on Robotics in Education, Bratislava, Slovakia, September 2010.

[2] Dias A., Dias N., Campos D. and Ferreira H. (2010). A Learning Environment for Robotic Production Line Simulation. ECEL 2010 – 9th European Conference on e-learning, Porto, Portugal, November 2010.

[3] Schaf F.M., Muller D., Bruns F.W., Pereira C.E., Erbe H.-H.(2009). Collaborative learning and engineering workspaces. Annual Reviews in Control 33, pp. 246–252.

[4] Lau H.Y.K., Mak K.L. (2004).The virtual company: a re-configurable open shell for problem-based learning in industrial engineering. Computers & Industrial Engineering 47, pp. 289–312.

[5] Potkonjak V., Vukobratović M., Jovanović K. and Medenica M. (2010). Virtual Mechatronic/Robotic laboratory – A step further in distance learning. Computers & Education, 55, Issue 2, pp. 465-475, September 2010.

[6] Siddharth S., Woo A., Olson T., Liu J. and Zhao F. (2005). A Spreadsheet Toolkit for Streaming Sensor Data. SenSys'05, San Diego, California, USA, November 2–4, 2005.

[7] PacHube [online] http://www.pachube.com

[8] Arduino [online] http://www.arduino.cc/

[9] AVRl [online] http://www.atmel.com/products/avr/

[10] ARM [online] http://www.arm.com/products/processors/index.php