"Starter kit for smart buildings"

Pervasive Computing : Selected Topics Project

Aebischer Nadia Department of Informatics
University of Fribourg
1700 Fribourg, Switzerland
nadia.aebischer@unifr.ch

Luyet Gil Department of Informatics
University of Fribourg
1700 Fribourg, Switzerland
gil.luyet@unifr.ch

ABSTRACT
The goal of this report is to describe an easy way to integrate a basic pervasive system into an already built house/building. The system should be cheap and easy enough to integrate and to operate so it could be used by the masses as a starter kit for smart buildings. The report will be based on the data analysis and the context of use of this system. Especially about the central unit managing the whole house.

1. INTRODUCTION
1.1 Smart building description
A smart building is by definition a building enhanced by the technology that is managing it. Generally the technology used is splitted into four parts. The sensors that collect data and allow to have an understanding on what is going on outside or within the house. The second part is the network that allows to spread the data collected by the sensors and to eventually activate the effectors if needed. The third part is the managing unit that allows to gather data and infer knowledge about the environment and thus either show the data to the user or activate effectors to modify the user's environment. The last part is the data visualisation that allows the user to interact with the data.

1.2 Aim of the project
The basic aim of the project is simple. We would like to produce a simple to use and to integrate system that would allows to enhance an already built building to a smart building. This system should be as cheap as possible and should physically alter as less as possible the house. This means it has to be based on cheap and already existing means and be a sort of "do-it-yourself" pack.

2. HOUSE DESCRIPTION
2.1 Context of use
We are going to consider a small building with a dozen of individual apartments. Each apartment would have 4 rooms each having 1 light and 1 windows with an electric sunblind.

In such a configuration it is usual to split the power supply into so call groups. Each group has his own electrical phase. For a standard 4 room apartment (1 kitchen, 3 rooms and 1 bathroom) we would have 3 or 4 group. One managing the kitchen, one managing the bathroom and 1 or 2 for the other rooms. The building has a central water heater, separated electric meters per apartment and an individual thermostat per apartment. A central electrical panel is available for the whole house. An indoor parking space in common with an electric door that is activated by a remote controller is also considered.

2.2 What could be automated
Sunblinds are opening or closing automatically by checking the sunlight and time. For example if it's sunny the sunblinds will be closing. The same way, at night, they should be closing if we don't want neighbour to take a look. The temperature of the apartment can also be automated. For example we can set the temperature we want to have in the apartment, and if the this temperature is below or above the default one, the system will change the thermostat accordingly. We can also imagine a way to infer if somebody is going to be there soon or not and raise more or less fast the overall temperature. This inference can be computed according to the presence of people in the house by reading the presence sensors. For example if somebody is never present during the weekends then the central unit should over the time learn that fact. The lights can also be automated by detecting the presence of people in a room. RFID tags linked with cars could be used to open the parking space doors. Of course, these automated behaviors can be stopped by the user if he doesn't want it. To do so, the user can access the central unit via a web interface and change the behavior the way he want.

3. A FEW WORDS ABOUT THE SENSORS ARCHITECTURE
3.1 Architecture of the sensors
The communication between the components are achieved by wifi or over the electrical network in case the wifi is not reachable. Communication over the electrical network can achieve a very fast connection speed between the components, up to 14 Mbits/s [1]. This means that theoretically pre processing of sensors data is not needed but for utility reasons we will preprocess the data. Each sensors is connected to a light computer, typically a Raspberry Pi could work but we could imagine an event less powerful one for

small tasks. This computer will preprocess the raw sensor data and play the role of the interface between the sensor and the Central Unit. This interface is needed in case of multiple types of sensors. By changing the sensors one could simply apply a patch to the small computer (even via the Central Unit) and resolve compatibility issues. This interface used here is a simple HTTP based communication via POST with a specific XML-based body very similar to the SOAP protocol.

```
POST address
  <MESSAGE>
    <TYPE>(alert|event|failure|do)</TYPE>
    <FROM>(sensor_id|central)</FROM>
    <CONTENT/>
  </MESSAGE>
```

Where the TYPE denotes an alert, an event or even a failure for example. The FROM and CONTENT parts deals respectively with the author of the message and the actual meaning of the message. A typical proximity sensor message would look like the following :

```
POST address_of_CentralUnit
  <MESSAGE>
    <TYPE>event</TYPE>
    <FROM>sensor_id2345</FROM>
    <CONTENT>distance:'2.34m'</CONTENT/>
  </MESSAGE>
```

A typical message send by the Central Unit to open a door would be:

```
POST address_of_door_effector
  <MESSAGE>
    <TYPE>do</TYPE>
    <FROM>central</FROM>
    <CONTENT>door[0]:'open'</CONTENT/>
  </MESSAGE>
```

3.2   Location
The location of the sensors depend of the type of sensor. Here is the list of the sensors we are going to consider

Luminosity sensor - outside, at each frontage of the house.

Presence sensors - around the lights and the doors.

Temperature sensors - inside/outside.

Energy consumption sensor - after the electric meter.

Water consumption sensor - before the water inlet.

Water temperature sensor - after the water heater.

RFID readers - for cars around the garage doors.

We also consider effectors, the list of them is given there.

Sunblinds effectors - to open and close sunblinds.

Light effectors - to switch the lights on and off.

Thermostat effector - to change the house temperature.

Automatic garage door opener - to be linked with the RFID reader.

This part is not really important for the scope we choose to aim our report but this should be at least mentioned because it list which sensors we are going to consider and receive data from to form a context and also on which effectors the Central Unit can send message to modify the environment.

4.   CENTRAL UNIT
The Central Unit is the main part of this architecture and the scope of this report. This Central Unit depends on several topics as its own architecture, the sensors data coming in, the effectors it has to interact with its environment and of cause its standard interface with the user.

4.1   Architecture of central unit
The architecture itself has no specific requirement. We can choose whatever we want, here we might focus on a standard Linux distribution as Ubuntu without any graphical environment as the only interaction we have should be via a web interface discussed later. Furthermore the computation needed is not too high as the preprocessing of the sensor is done on the sensor side. The computer should also be running 24/7 and thus the computer has to have a low consumption. A light computer as a Raspberry Pi might be enough.

4.2   Sensor fusion
The sensors fusion is performed to produce a context. A context is the actual state of an entity such as a room or a person. In this report we need to know exactly which kind of contexts we are going to consider to know exactly which sensor data has to be aggregated to produce the actual state of a room for example. Here are the contexts we are going to conside

The lightning state of the room
   Dark or lighted ?

The occupation of the room
   Is the room empty or not. When does the room is usually empty ?

The actual weather state inside and outside
   (temperature, lightning and precipitations). Is the temperature cold/warm, based on a threshold. Is the temperature cooler or warmer outside ? Does it rain/snow outside ? What is the actual lightning ?

Which is the part of the day ?
   Look at the actual time => infers moning / evening / midnight / ...

Is a car waiting at the garage doors ?
   Look at the RFID reader. Is the car allowed to pass ?

What is the actual consumption of the whole building ?

May be used to display to the webinterface.

[INSERT HERE WHY THOSE CONTEXTS]
As this list is now fully established we have to on which sensors to use to analyse the context wanted.

- The lightning state of the room: Lightning sensors + sunblinds status + outside lightning.

- The occupation of the room: Presence sensors.

- The actual weather state: Regional meteo status available over the internet.

- Is a car waiting at the parking doors: RFID reader near the garage doors.

- What is the actual consumption of the whole building: Energy consumption sensor and water consumption sensor

Admit now that each context has been computer via the sensor data. We must now describe how an actual context combined with new sensors data coming in will react. This can be done via listening all the possibilities.

Do activate lightning **if** somebody enters the room $\wedge$ the room is dark.

Do deactivate lightning **if** The room becomes empty $\wedge$ the room is lightened.

Do reduce heating **if** room is usually empty for a quite long period $\wedge$ nobody is inside.

Do reduce heating **if** outside temperature $>$ comfort temperature + threshold

Do augment heating **if** temperature $<$ comfort temperature + threshold.

Do open doors **if** car is at the parking doors $\wedge$ door closed.

Do close the doors **if** timer for doors is over $\wedge$ no car at the parking doors.

Do close sunblinds **if** lightning outside $<$ threshold $\wedge$ part of the day!='day'.

Do open sunblinds **if** sunblinds closed $\wedge$ part of the day='day' $\wedge$ someone enters the room.

4.3 Effectors
Automation of control. Inference for temperature calculation by knowing is somebody usually will come to the house.

4.4 Web-based manual controller
for mobile + desktop + over Internet stats presentations failure presentations notifications (SMS, e-mail, Facebook, invasive smartphone application) what must be stopped what must be switched on/off automatically crash of computer (?) : keep some switches ?

5. CONCLUSION
6. ACKNOWLEDGMENTS