[CS2310 Multimedia Software Engineering](http://people.cs.pitt.edu/%7Echang/231/231syl.html)

Final report

Anh Phan Nguyen

[apn15@pitt.edu](mailto:apn15@pitt.edu)

1. **Introduction**

In this T2 project, the local weather temperature data is used in combination with the room temperature from T1 project in order to give inferences about room context. The room context can be useful as other third parties may want to utilize this information for advertising or providing services. For example, many retailers who sell heater may want their ads reach home owner who are having problem with their heating devices. Health department may also want to visit places that remain cold during winter. In summer, retailer who sell fans or air conditioners may want to advertise their product to people who don’t have one installed in their room. The extension of this T2 project is that the contexts of these situations is inferred using two type of data mentioned above. Then the contexts are distributed to the retailers and other third parties by uploading it into a website.

In this project I assume there are 3 different type of context information:

* Room is too cold
* Room is too hot
* Room has a fire

And there are 4 type of third parties:

* Fire department (who will come when the house is on fire)
* Cold retailer (who want to sell/repair heaters)
* Hot retailer (who want to sell/repair fan, air conditioner)
* Health department (who need to visit people in cold places during winter)

Context from local weather temperature and room temperature can be divided into small cases. The table below list all possible combination of the two types of data. For example, the entry in which weather temperature < 0o and room temperature from 0-12o is filled with ‘Cold’. It means the room is currently very cold and the reason is because outdoor temperature drops too low.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Weather temperature** | | | | |
| Room temperature |  | < 0 | 0– 15 | 15 –30 | > 30 |
| < 0 | Cold | Cold | ? | ? |
| 0 – 12 | Cold | Cold | ? | ? |
| 12 – 18 | Cold | Cold | ? | ? |
| 18 – 30 | Normal | Normal | Normal | Normal |
| 30 - 50 | ? | ? | ? | Hot |
| > 50 | Fire | Fire | Fire | Fire |

Table 1: table of home context from two type of data, weather temperature and home temperature

The ‘?’ mark is used when no context can be inferred. For example, if room temperature is over 32o while out door temperature is 15o, perhaps the home owner turned the heater too high and thus, nothing useful can be infer from this.

1. **Program structure**
2. Components

Msg30

Msg1003

Msg1004

Input Processor

Temperature Sensor

Outdoor Component

Home Context Monitor

Uploader

Website

Msg38

Figure 2: diagram showing components of T2 project. Green boxes are data collectors, grey box is context processor, and blue box is the website that shares context to third parties.

Each box is a component of the SIS testbed except the Website (blue). The SIS server component is not showed here but in principle, all communications between components are routed through SIS server.

Two green components collect the input data. Temperature Sensor simulates the temperature sampling from the given file, with the interval of 3 seconds. Outdoor Component use the provided API from weathersource.com to collect real time forecast data of the area with zip code 15213 (Oakland – Pittsburgh). The data are then forwarded to the HomeContextMonitor component (grey) to infer the context.

The HomeContextMonitor component (grey) bases on the table 1 to calculate the home context. For example if HomeContextMonitor receives 135o from Temperature sensor and 25o from Outdoor component, it infers that the home is on fire and will create alert message ID = 38 with the alertType=Fire. This alert message will then be sent to Uploader component.

The uploader uses GET method to upload data to the website. For example, if Upload receive alert message ID = 38 from Homecontext monitor with 135o room temperature, 25o outdoor temperature, and alert type is Fire, then the URL is as following:

<http://lycapheden.com/npanh/update.php?status=fire&roomtemp=135&weathertemp=25&time=2014-12-16T02:43>

The website component (blue) resides at lycapheden.com/npanh. It has 3 separate log files: fire.txt, cold.txt and hot.txt. Each time an alert message is sent, a row will be added to the end of file based on the alert type. Each row of these files has 4 columns. First column is the status (fire or hot or cold), second column is room temperature, third column is outdoor temperature and last column is the time alert message generated. Given the example above, the row to be be added into fire.txt will be like this:

fire 125 25 2014-12-16T02:43

1. Messages.

Besides messages 20 and 23, there are 4 data messages is used in total, their ID are: 30, 38, 1003 and 1004. Messages with IDs 30 and 38 are reused with some modification, and messages with IDs 1003 and 1004 are newly created because new component is introduced into the system.

There are 2 new messages defined, Msg1003 and Msg1004 (see figure 2)

- Msg1003 has ID=1003, is used to send room temperature data from input processor to HomeContextMonitor. Format of the XML file is as following:

<Msg>

<Head>

<MsgID>**1003**</MsgID>

<Description>**TemperatureReading**</Description>

</Head>

<Body>

<Item>

<Key>**Temp**</Key>

<Value>**value**</Value>

</Item>

<Item>

<Key>**DateTime**</Key>

<Value>**value**</Value>

</Item>

</Body>

</Msg>

- Msg1004 has ID = 1004, is used to send weather temperature from Outdoor Component to HomeContextMonitor. Format of the XML file is as following

<Msg>

<Head>

<MsgID>**1004**</MsgID>

<Description>**WeatherReading**</Description>

</Head>

<Body>

<Item>

<Key>**Temp**</Key>

<Value>**value**</Value>

</Item>

<Item>

<Key>**DateTime**</Key>

<Value>**value**</Value>

</Item>

</Body>

</Msg>

- Msg38 has ID = 38, which is used to send alert message from HomeContextMonitor to Uploader. The key “DataStream” contain both room temperature and outdoor temperature, separated by the colons. The AlertType will have one of these three values: fire, hot, cold based on the context generated. Given the example above, the DataStream field and AlertType filed in XML will be generated as following:

<Item>

<Key>**DataStream**</Key>

<Value>**135:25**</Value>

</Item>

<Item>

<Key>**AlertType**</Key>

<Value>**value**</Value>

</Item>

1. **Implementation**

The components are designed and implement using SIS testbed provided. Detail of the design is stored in the file testProject1\_Spec.PRJ, which can be open using SISProjectCreator. All java files is implemented in Eclipse then port back to project folder.

The website is coded using PHP language. The main webpage is <http://www.lycapheden.com/npanh/>. To retrieve context information, go to webpage [lycapheden.com/npanh/getinfo.php?user=](http://www.lycapheden.com/npanh/getinfo.php?user=firedept). The ‘user’ parameter can be set to ‘firedept’, ‘hotretailer’ or ‘coldretailer’

1. **Conclusion**

In practice, third parties may be interested in contexts from many different homes, not just one. In this case, the diagram in figure 1 could be changed so that there are many separated green components, each set corresponds to one home. This scenario could be possible if in future, when ubiquitous computing makes sensors and intelligent devices become popular to home owners.

In addition, other types of sensors could be used to improve the context prediction of HomeContextMonitor component. For example, a sound sensor could be included to make sure there are people at home so that health department will not arrive an empty place that remain cold during winter.

Finally, when all the contexts is store in the website and shares among third parties. How this information can be effectively shared without violation of privacy. For example, should retailer really need to know the addresses of homes that need to buy heaters? Or there should be a way to deliver the ads to these homes without retailer knowing the addresses?