Context Aware and Device Dependent Interaction in Smart Environments

Candidate: Emanuele Furci

Supervisors: Fulvio Corno, Dario Bonino, Luigi De Russis

Introduction

The thesis lies in the field of smart environment, a research area that embraces a variety of disciplines, including artificial intelligence, pervasive and mobile computing, robotics, middleware and agent-based software, sensor networks and multimedia computing. By integrating Information and Communication Technology (ICT) with automation systems, a smart environment is able to control, monitor and manage appliances, facilities and devices, increasing comfort, security and energy efficiency. Moreover, a smart environment can acquire and apply knowledge about the surrounding environment and its inhabitants to improve their experience in it. A typical smart environment is a smart home where an intelligent automation system is installed to support inhabitants' daily life.

Goal

The goal of the thesis is the creation of an intelligent notification system for smart homes, able to deliver messages regarding the environment to the most suitable end user device. The system takes into account the surrounding context, made up by information about users, devices and home state, to select the devices and send them generated messages. To achieve this goal, two steps have been performed:

- create a data modelling infrastructure able to represent information about users, devices, messages and smart home (Figure 1).
- develop a software to elaborate data model information and to deliver messages generated from the smart home to the appropriate end users device (Figure 2).

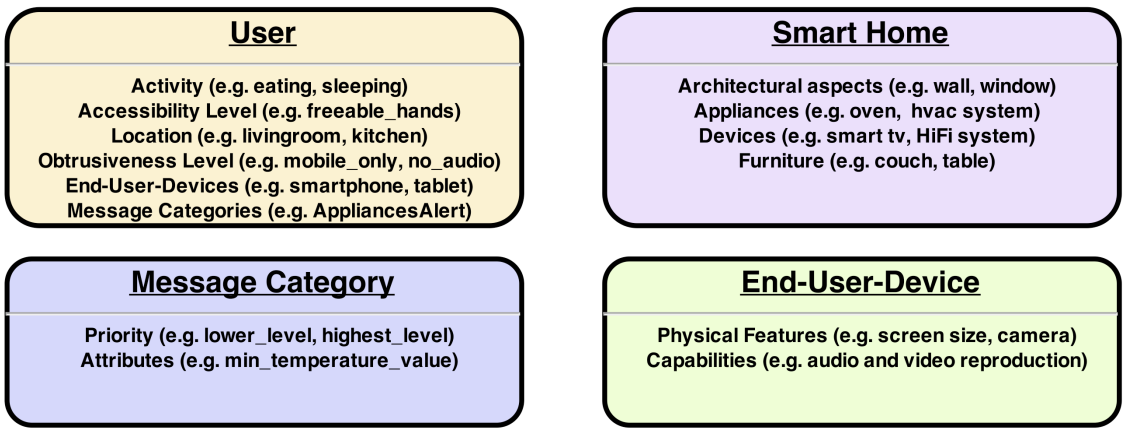

Figure 1: Main modelled information

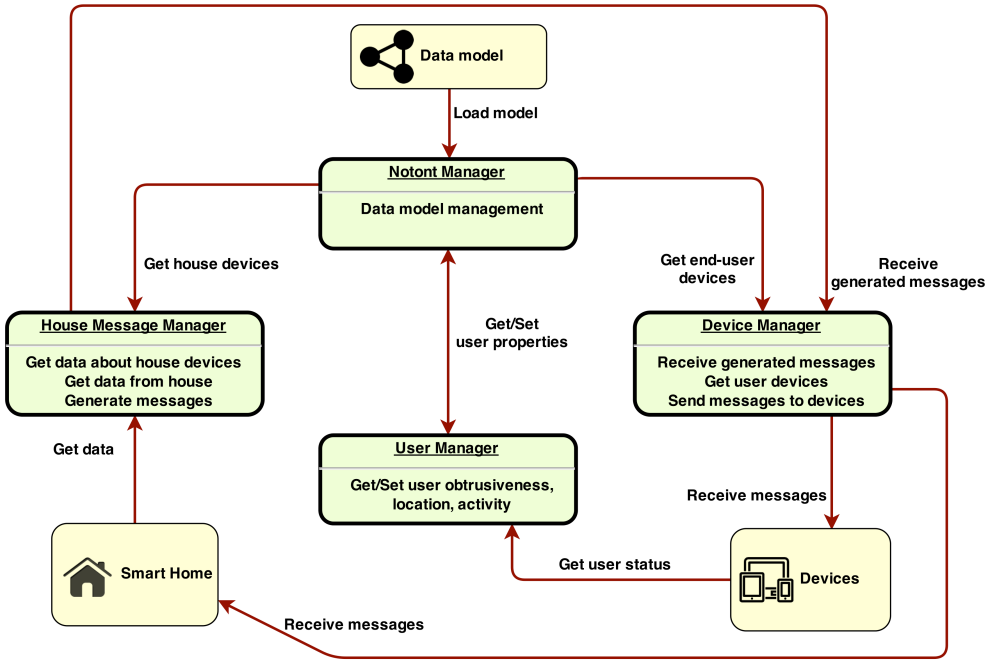

Figure 2: Software architecture for the notification system management

Data modelling and software implementation

The creation of the data model has a fundamental role in this work. Data is represented by means of an ontology[1], called Notont. At first, several existing ontologies for context management in smart environments were explored, but only some of them fit partly with the intended domain. Such ontologies were used as a base to create Notont. The ontology was refined setting the proper equivalence statements between classes and object properties of the different imported ones; it was completed, adding missing classes and properties (object and data properties). Four main entities were modelled: users, devices, smart home and message categories. Users are modelled by their current activity, location, obtrusiveness level[2], owned devices and interested message categories. Moreover, each activity is associated with an instance of Accessibility, so information about the user accessibility level[3] are modelled as well. Each device is described by means of its physical features such as screen size or type of microphone. Knowledge about such features gives information about how a message can be delivered (e.g. the presence of loud speakers means that a message may be delivered as a vocal message). The smart home is modelled in terms of its architectural aspects (rooms, walls, windows) and devices, such as plants and appliances. House devices are located in a particular place and are described in terms of functionalities and states.

Information is used to generate and deliver messages to user devices. Each message category implies the usage of the proper house appliances to generate that particular message. When a message is generated, the model is queried to get the end user devices to use for the delivery process. Using users information about their message preferences, it is possible to infer which

[1] an ontology is an explicit specification of conceptualization, a formal explicit description of concepts in a domain of discourse
[2] type of user availability in receiving messages on personal or house devices such as smartphone or HiFi system (e.g. obtrusiveness_mobileOnly to indicate that user doesn't want to receive messages on house devices)
[3] type of interaction between user and devices still available while performing an activity (e.g. accessibility_freeableHands for an activity like eating. User is performing an activity but can still use its hands)

one is interested in receiving that message. Then, for each interested user, information about the current activity and obtrusiveness level are used to infer if the user can actually receive the message. At last, combining accessibility level, obtrusiveness level and available devices, it is possible to infer the most suitable user device.

The ontology management is performed by the developed software whose modules are shown in Figure 2. Such software aim at test the ontology in a test case scenario. Notont Manager takes care of the data model management (insert, update, delete and query the model) by means of the OWL API, a framework for ontologies management in Java. The query process required an external reasoner[4] and, for this work, HermiT[5] was used (an example of query is in Figure 3). The House Message Manager encompasses the gathering of information from the smart home. For this work, the e-Lite lab was used, in which a smart environment is recreated. This system is managed by Dog[6], a gateway developed by the e-Lite research team for smart homes management. The House Message Manager interfaces Dog to gather data and, from them, generates messages. The User Manager takes care of updating user state, such as activities, locations, etc.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX locont: <http://webmind.dico.unimi.it/CARE/locont.owl#>

SELECT ?activity ?activityType  WHERE {

?person foaf:firstName "Giulia"^^xsd:string.
?person locont:hasCurrentActivity ?Activity.

?activity rdf:type ?activityType.
?activityType rdfs:subClassOf* locont:Activity.
}
```

Figure 3: SPARQL query: get the activity and its type from a person called "Giulia"

For example, the smart home can signal, through the flooding sensors, a flood in the cellar. Such information is used to generate a message belonging to the proper category (Plants Failure). Messages are sent to the Device Manager that, by means of Notont Manager, queries the model to get end-user devices (one for each user) and send them the incoming message.

Conclusions and future works

Starting from Notont, a test case scenario has been created, inserting information about users, devices, house appliances and information about the message types. The scenario was used to test both the ontology and the software.

Preliminar results are promising: the system can generate messages starting from the incoming data retrieved from Dog and deliver them to end user devices with respect to the context. An improvement of the ontology could be the removal of useless classes to make the ontology lighter and reduce the overall time to infer upon it. The software provides functionalities of message generation, device inferring and message delivery. The datamodel management, due to the usage of the reasoner, heavily impact on software performances. Starting from a message generated by the House Message Manager, the system infers the best end user devices in a mean time of 3.1 s[7] (sd 2.2 s). The overall delivery process, from the message generation to its reception on the end user devices, takes a mean time of 5.7 s (sd 3.3 s). Finally, message generation may be improved to provide a more wide range of message categories.

[4] software able to infer and extrapolate implicit information from an ontology

[5] http://hermit-reasoner.com

[6] http://dog-gateway.github.io

[7] calculated on 20 messages of 3 different types on a Dual-core CPU based computer with a clock frequency of 2.26GHz and 4GB RAM