SE2-DA326A
Ht-18
DSY16

Interactive House – Project Description

This document presents a project that should be developed at the course SE2-DA326A.
It is an example of a heterogeneous environment with several distributed communicating units. A term that also may be used is *Ubiquitous Computing*, or *UbiComp*, which may have The following meaning (for further information check out [1]):

***Ubiquitous computing** (**ubicomp**) is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities. In the course of ordinary activities, someone "using" ubiquitous computing engages many computational devices and systems simultaneously, and may not necessarily even be aware that they are doing so. This model is usually considered an advancement from the desktop paradigm.*

Another similar concept is *Internet of Things*, which may be described as ([7]): "*In computing, the term **Internet of Things** (also known as the **Internet of Objects**) refers to the networked interconnection of everyday objects. It is generally viewed as a self-configuring wireless network of sensors whose purpose would be to interconnect all things. … If all objects of daily life, from yoghurt to an airplane, are equipped with radio tags, they can be identified and managed by computers in the same way humans can. The next generation of Internet applications (IPv6 protocol) would be able to identify more objects than IPv4, which is currently in use. This system would therefore be able to instantaneously identify any kind of object. The Internet of objects should encode 50 to 100 trillion objects and follow the movement of those objects. Every human being is surrounded by 1,000 to 5,000 objects*"

The rest of the document includes following parts: Introduction, that will bring an overview of the context of the project, further descriptions on the project, project organization, etc., and finally References.

Introduction

The project is about doing experiments on a concept we call *Interactive House*. A similar concept is *Smart House* that also involves ambitions on intelligent activities (to study this, see more at, for instance [2]). Still, we primarily restrict our intension to study human controlled activities in first place.

The intention is to study how to communicate with and observe devices of a house. Devices may here be light, coffee machine, temperature, etc. Besides from direct communication we may also study built in "intelligence", so for instance sensors of the house may also be used to control devices of the house, or there may be a defined schedule, for instance, to turn on and off light at different points in time. Communication for controlling house devices may be performed through mobile phones, or e.g., from a web based GUI. All devices of the house, that should be controlled, should interact directly with a dedicated house server. Units, such as mobile phones or Web based application, communicates with the devices directly or through

the server. Furthermore, those units are informed by the server about the devices that are currently present for observation and /or control. Figure 1 below provides a first illustration of the context of the project.

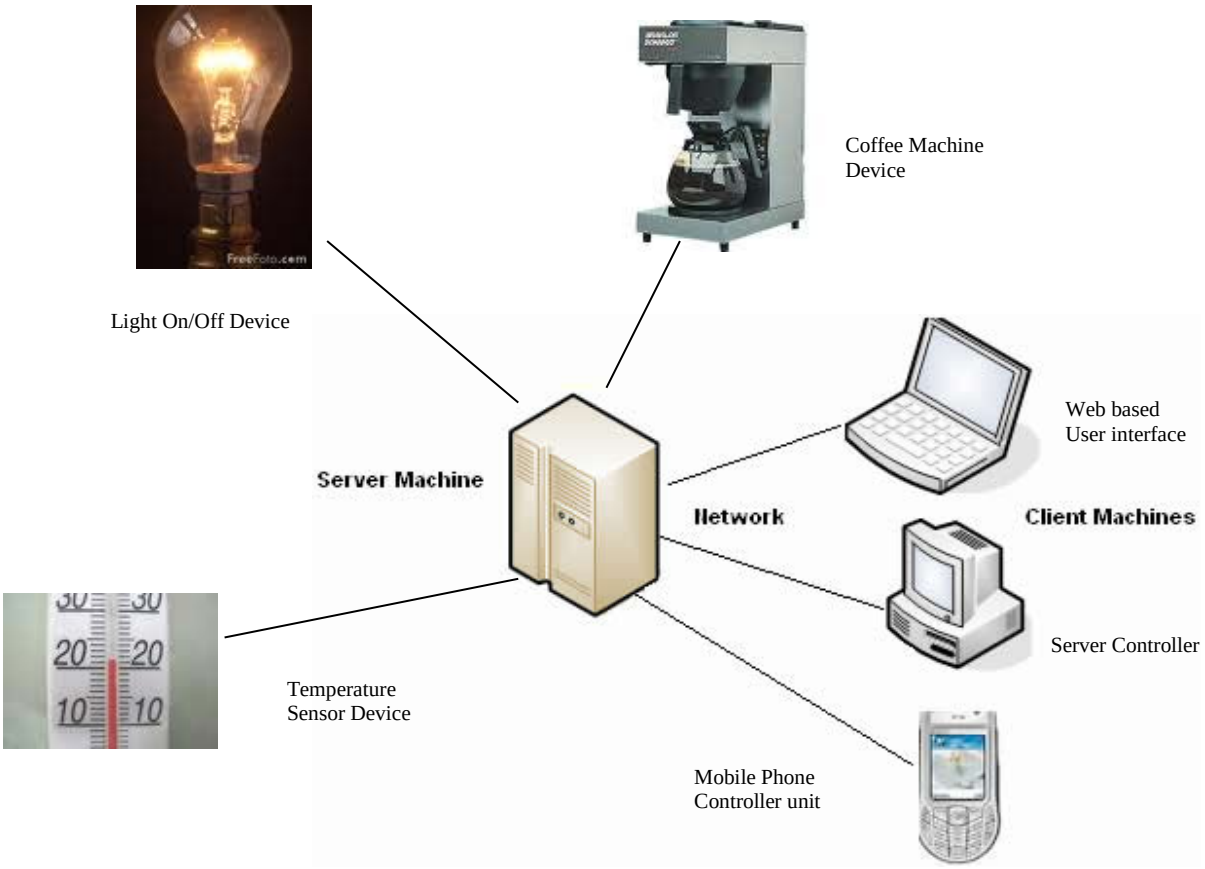

Figure 1, *Interactive house, with a house server, controllable devices (here a light, and a coffee machine), sensors (here a temperature giver), and communicating units (through web and mobile phone)*

In Figure 1, the Server Machine should be built in, in the house. It is here controlled through a specific Client Machine (probably there should be quite restrictive ways of controlling the server). Every device, controllable device or sensor device should be connected to the server to be able to communicate with those. House devices may then be observed and/or controlled through communicating units, such as a mobile phone, or a laptop with a web based user interface. If possible, to have a more uniform model of communication, both mobile phones and laptops (or other kinds of computer), may, for instance, carry web based user interfaces. Still, there may be reasons for having context dependent ways of communicating. Figure 2 shows a communication interface with a specific light.

Figure 2, *An example of an interaction interface towards a light at a mobile phone unit.*

Figure 2, illustrates a GUI-component at a mobile phone to turn on or off the light of a specific lamp. The mobile phone of Figure 2 seems to have interaction based on touch screens as that of an iPhone, or the newer examples of Android based phones. If a touch screen is not available, then, of course, the GUI-component would be slightly different. At a traditional computer interaction with the buttons would probably be done through a mouse utility. However, there may also here be alternatives.

Application domains

The system may be used by the general public. However, a further motivation arises in a perspective of health care ([6]) where persons with more or less high grade of disabilities may be supported by this kind of system to access the functionalities of their homes. That is, even though you are bounded to your bed or a wheelchair, etc., you may control the devices of your home. The system emphasizes on independency and self control.

Furthermore, in a health care domain, if the system should be integrated with communication and information modules, we would put it to an even higher level. For instance, state of health could be communicated in both a direction from a patient to nurses and medical doctors, and from those to a patient. However, at this course those kinds of functionalities are outside the scope of the project. Those discussions however, illuminate on some possible future uses of interactive or smart houses.

To summarize this sub section, we should strive after a context of health care, where we increase a person's independency, self control, and accessibility of functionality of the person's home. We here assume that the person has the ability to use a mobile phone. A web based user interface may also be used, inside, as well as outside the home, by a patient as well as by medical staff.

A further system description

The system will be mainly divided into the following significant sub parts:

- ***Server***: controls the main communication between parts, also involves a database with information, such as, authorized users of the system, and devices/units attached to the system
- ***Communicating and observing units***: to interact with the system for controlling and observation reasons. There may be different such units, such as:
    - Computer with a web based interface (or perhaps a more common application).
    - Mobile phone: interaction may be based through touch screen based GUIs, or with alternative techniques that are current today, such as, speech recognition, and gestures (at screen or in the air).
- ***Devices***: those are the parts that actually are controlled and observed, as shown in Figure 1. In the project we may have to regard the following:
    - In some cases there may be true physical components to control, such as light. No matter what, we may probably not interact with the device directly, but rather through, for instance, a micro controller that in turn interacts with the server.
    - In other cases we have to simulate devices, such as a coffee machine, or perhaps a temperature giver, through an application at a specifically dedicated computer. An example on this may also be a media player for music that may run on a computer but still act as a controllable device of the interactive house.
- ***Communication protocol***: there are two significant aspects on this:
    - First, we have the common interaction between parts, meaning, for instance, that a light is put on from a mobile phone
    - Second, to support a scalable system with new devices that may be plugged into the system, we need the devices to hold their own user interfaces (UIs). A device not only registers itself at the server when they are plugged into the system, it should also communicate that UI to the server, so that the UI may be further distributed to the communicating units. For instance, the light of Figure 2, holds a software component corresponding to its UI (as shown in Figure 2). That component will be distributed to mobile phones of authorized users, through the server. Once being uploaded at a communicating unit, that UI is used for controlling and observing the corresponding device.

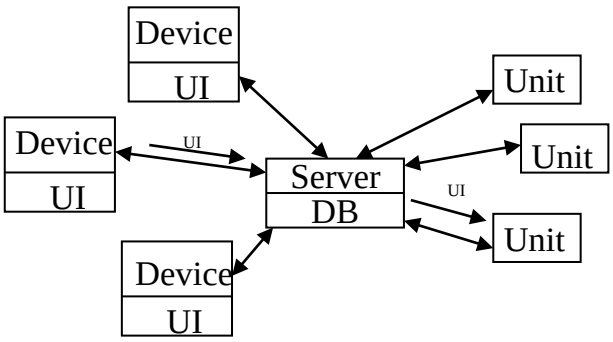

Figure 3, *A high level architecture with a server, including a database, controllable and observable devices, and communicating units. From a device, its UI is sent to the server, and uploaded from a unit.*

Project organization

A project group to implement an interactive house will be divided into sub groups to fulfil the sub tasks of the project, as described previously, more precisely:

- Server
  - Server + Database
- Units
  - Mobile phone
  - Computer
- Device
  - Simulation of a house, with light, alarm, temperature
  - Simulation of other more specific devices, such as, a media/music player, and a coffee machine
- Communication protocol

Each sub task (see the list above) should be implemented by a sub group of about two to three developers. Furthermore, there should be one person that acts as a project manager, assisted by one specific requirements manager. The Project Manager and Requirements Manager will also be included in groups. This means that about fifteen developers are required to fulfil the task as a whole.

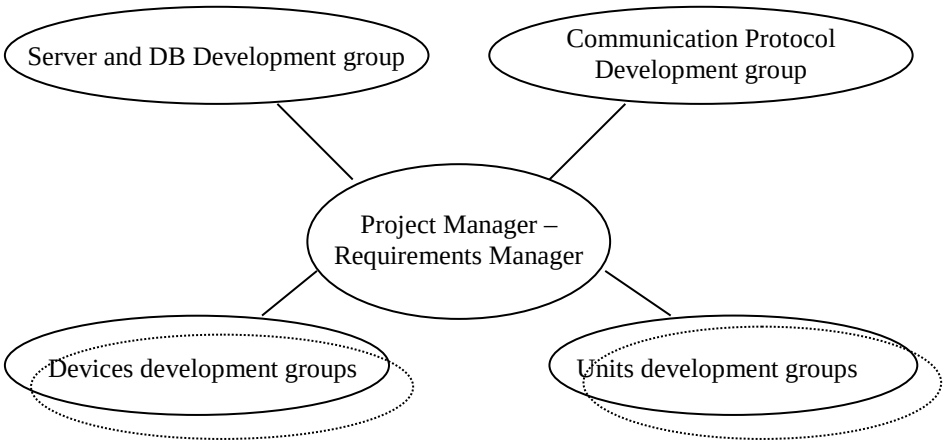

Figure 4, *Project organization.*

Figure 4, illustrates the project organization with respect to sub groups. Furthermore, Figure 4 illuminates on the projects control by, in first place its project manager (*PM*), but also the by requirements manager (*RM*). Moreover, the project groups should answer to a group of costumers/super project managers (that is, the teachers).

Project meetings will be performed, on one hand, internally by the project group, lead by the group's *PM*, and on the other hand, with the costumers/super project managers. At those formal meetings with costumers/super project managers all of the group participants should

be present. There may also be less formal meetings where the project's *PM* and *RM* represents the project. That is, those meetings should not involve all of the group members.

Formal Project Meetings and Artefacts

The project should be driven by an iterative and incremental process approach, such as Rational Unified Process, RUP (still, only a very limited amount of the artefacts recommended by RUP will be used). For more information on iterative and incremental processes, and RUP, see e.g., [3], or [4]. Still, other process models may be partially used by the sub groups, such as a more agile model, e.g., eXtreme Programming.

The formal meetings will be held each second week (or each week, if that should be preferable). At the formal meetings, the *PM* and the *RM*, should be ready to answer for the project, through discussing the results so far, but furthermore, also through prototype systems, and artefacts. The following list of artefacts will be used, and each subgroup should produce that:

- Vision document, explains the project Vision
- Requirement document, system functional requirements
- Supplementary requirements, non-functional requirements, or quality attributes
- Project risks, for the process, not especially the product but may include this
- Design document, Software Architecture and Design, Use UML diagrams for both analysis and design
    - Use case diagram, should probably be very close to requirements doc
    - Class diagrams
    - Interaction sequence diagrams
    - State diagrams
- V&V and Test, use specific document for this.

More detailed on those artefacts may be found through [5]. The content and intension of such artefacts will moreover be studied further as a part of the course.

More on what to do

To support the project there will be several lectures and labs concerning technical details. The course schedule outlines the lectures and labs that will play this role. However, a significant amount of work will be expected from the course participants themselves to be able to develop themselves far enough to make the techniques useful in the context of the project.

Since it is required that some specific techniques are used in developing the project the first part of the course will probably mainly concern design. When entering a later phase of the course more core implementation work may be done. Still, early prototypes may be developed as soon as possible, without the major production code being implemented.

Since the project is developed in a group of quite many involved persons, it is probably only possible to develop the task of one group with only the assumption that the functionality of other groups' sub projects is correct. The integration tests will later on reveal presence of errors when software components are integrated. To handle integration each sub group should

strive after simulating the environment, and only assume that other parts work. This has then to be based on an interface, or protocol, that the corresponding groups have negotiated about.

The course schedule also points out project meeting appointments between the representatives of each project group and the teachers. At those meetings the above mentioned artefacts should be discussed. The artefacts should consider the project as a whole. Furthermore, project groups should have presentations for about one hour each. According to the course schedule there should be two such presentation occasions. The first one should bring a first state report, while the second one should concern the final project.

A final report shall be delivered to the course leaders at the point of the project presentation. This report should include all final artefacts for the project as a whole. The design should also elaborate on a significant amount of so called *Design Patterns* that the design may reveal.

References

1. http://en.wikipedia.org/wiki/Ubiquitous_computing
2. http://architecture.about.com/od/buildyourhous1/g/smarthouse.htm
3. Sommerville Ian, *Software Engineering*, Addison Wesley (792 s), 9th edition (2010), ISBN-10: 0137035152, ISBN-13: 978-0137035151
4. Material used in previous courses, such as, Working in Projects – Project Based Development, or others
5. Templates and examples of documents that should be used as process artefacts, available at the course moodle page.
6. http://en.wikipedia.org/wiki/Health_care
7. http://en.wikipedia.org/wiki/Internet_of_Things