TTM3 Smart Heating System Report

Xiao Chen, xiaoch@stud.ntnu.no
Pavel Arteev, pavela@stud.ntnu.no

October 2013

1 Introduction

By the magnificent growth of using technology in the daily life, the concept of the smart housing system is raised these years. Since smart cell phone and other daily using device have more and more functions than its original function, the self-adaptive system is necessary service to control and manage other devices in the house.

Self-adaptive system in this report will be also dynamic component system. It is the system emphasis on context constraints that govern how components may be composed (without prescribing a particular structure), and compositional adaptation is bounded by the components.

The project of this report will be implemented in OSGi[6] framework. Since this project would demonstrate a dynamic component system scoped in smart heating system in the house, OSGi framework is a good choice here. The OSGi framework is a module system and service platform for the Java programming language that implements a complete and dynamic component model, something that does not exist in standalone Java/VM environments. Applications or components (coming in the form of bundles for deployment) can be remotely installed, started, stopped, updated, and uninstalled without requiring a reboot; management of Java packages/classes is specified in great detail. Application life cycle management (start, stop, install, etc.) is done via APIs that allow for remote downloading of management policies. The service registry allows bundles to detect the addition of new services, or the removal of services, and adapt accordingly.

There are several benefits point for OSGi framework using for Java project, since it improves some not advanced part of the Java programming language in self-adaptive system.

For Development time benefits[3]:

- Strict development time (and runtime) enforcement of module boundaries
- A service-oriented architecture that works for managing service dependencies between modules.
- Better ability to structure development teams the way you want to

- Faster team-based development
- Faster testing cycles
- Support for versioning as part of dependency management
- Less road-blocks

For Runtime benefits:

- Full information about the installed modules and their wiring is available at runtime - a level of insight operations teams have never had before.
- Isolate changes
- Share dependencies
- Use just the server facilities you need

However when we first time used OSGi framework, it is not so easy to use in this project which we will discuss in the conclusion section of this report for some feedback points.

This project will focus on implementation a self-adaptive system, smart heating system with four heater manager simulator and four heater device simulator and two temperature sensor simulator. The communication channel between heater device and temperature sensor is through heater manager. And all the communication will be based on Hydna[1], which is a hosted backend into which you can send data and have in instantly appear on other devices. More detail about the system design will be covered in the section 2.

2 System Design

Figure 1: Spheramid Gateway

3 Related Work

In the section 2, it explain the prototype system of this report. The concept is to use three bundles to communicate through local communication channel or remote communication channel to build the self-adaptive system to control the house heating system. There are several similar concept consuming product on the market, they will be discussed and compared with our self-adaptive heating system in this section.

3.1 Ninja Sphere

Ninja Sphere[2] is a kickstarter[7] project just currently founded on the kickstarter website. It is a next generation control of the environment with accurate in-home location data and a gesture control interface. Ninja Sphere learns about the user and user environment. It uses data from sensors and actuators to build a model that can inform user if something is out of place. It can monitor temperature,lighting,energy usage, user and even user pets' presence, and anything else user connect to their sphere. By using data from user's devices,environment and location, ninja sphere is able to advise you intelligently and give user control only when user need it. Like our prototype project, the devices connected with Ninja Sphere is 'dump' device, there are no directly communication between these 'dump' device, every device only care about their own responsible function. The developer of the Ninja Sphere has released the supported devices for the system to connect with in Figure2.

And according to Ninja Sphere system architecture in Figure 3, the main control component in the system is Spheramid Gateway1, it is the main service component to be used for other connected devices communicate with. In our

Figure 2: Ninja Sphere Current Supported Devices

project system, it will the same as heater binding to bind one heater device and one temperature sensor as a service block in the system. Although in the prototype of this report is only based on internet communication not like the Spheramid Gateway has Wifi, Bluetooth Classic & Low Energy, ZigBee and USB communication method, the basic concept is the same to be as a main communication bridge to cooperate with other component in the system.

For Ninja Sphere system, the user can user their smart phone or wearable smart device (like Google Glass or smart watch) to control and get notification from other daily using devices in the house. For example, user Adam left home and forget to turn off the heater, then the Ninja Sphere system will make the decision send the notification to Adam's smart device to ask user if they forgot to turn off the heater. Once Adam think he forgot to turn off the heater, then he can use the smart device to send turn off command to the system, after that the system will send the command to turn off the running heater in the house. And in current project of this report, the user case is almost the same except the notification and cooperate with smart device function is not implemented in current phase.

In the system of this report, the system has some configuration value from the user which can be changed by the user as well, then the other 'dump' component in the system will do the self-adaptive process when they reach the certain threshold value. The basic concept is the same as Ninja Sphere system.

3.2 HyttaMi

HyttaMi[5] service is a consuming product developed by DEFA AS[4]. Hyttami is a subscription based service including GSM subscription for product costs

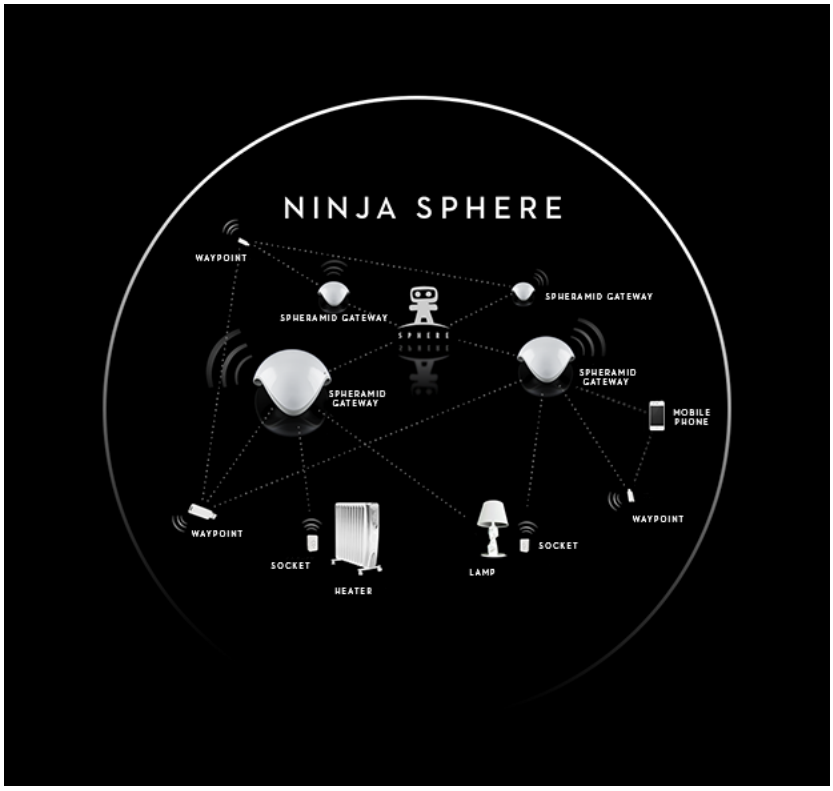

Figure 3: Ninja Sphere System Architecture

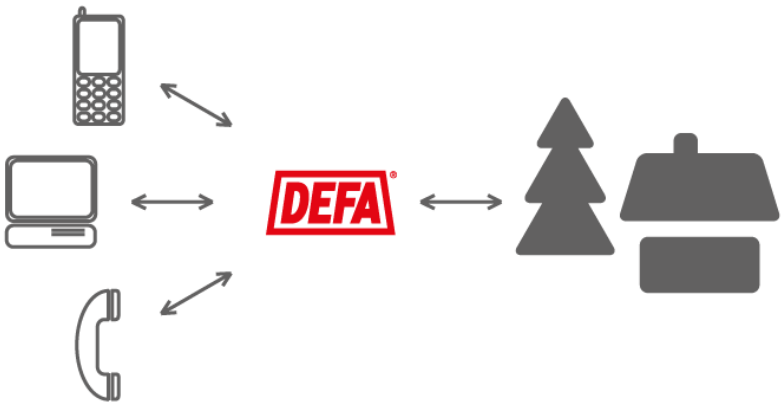

Figure 4: HyttaMi System Architecture

associated with the use of the service , and it provides the easiest and most effective way for remote control and monitoring of electricity, heating and alarm in the cabin.

HyttaMi gives the user a safe and secure solution with full control of heating and alarms in the house . It is very easy to control and monitor both heat , temperature and alarm via the internet , telephone and sms.

The main function HyttaMi has are:

- Turn on or off the heat using a text message
- Check the status of the cabin using a text message
- Turn on or off the heat using the internet
- Check the status of the cabin using the internet
- Turn on or off the heat using a normal phone

According to the system architecture in Figure 4, the working process is also a self-adaptive and user controlling system. Although HyttaMi provides more services than heating control system, other services compare to heating control system has the same basic working logic which is also the same as the prototype system in this report.

The normal user case of HyttaMi could be like, user Eva wants to go to her cabin this weekend, but she still needs to work in the weekday, so originally there is no way for her to make the heating system working in her cabin unless she goes there. But now by using Hyttami, she can just use her phone to send some sms message to the correct number, then the system will turn on the heating system and wait her coming to enjoy the cabin weekend. This working process is also implement in the system of this report. Once user binds the heater device and temperature sensor together, it initially start the heating logic, then

the temperature sensor would send current temperature value to the control component, according to certain threshold user set before, the heating would be working to reach the temperature threshold. Moreover since the heater binder, heater device and temperature sensor are not communicate directly, they all communicate with the Hydna service, then each bundle can be initiated and allocated resources in different places. So the remote control process is also possible in the current prototype system.

3.3 Feedback from Related Work

After researching about the related work in the consumer market, it is quite interesting to notice that there are quite a few product have the same working process and concept as the system of this report. And since the development of the smart devices, more and more the system provide the service which mainly communicated with smart device component in the reality.

Because of the time limitation of the system in this report, the potential function of this kind system which are communicated with smart devices has not been implemented yet. But the work flow to implement this important function will not be so hard in the future work of this system since currently all the communication are based on the internet and the main advantage of the smart device is to keep the user on the internet as long as possible.

It is really promising feature if the cooperation of smart devices are done for this project.

4 Conclusion

References

[1] Hydna AB. *Hydna*. 2013. URL: https://www.hydna.com/ (visited on 12/01/2013).

[2] Ninja Blocks. *NINJA SPHERE: Next Generation Control of Your Environment*. 2013. URL: http://www.kickstarter.com/projects/ninja/ninja-sphere-next-generation-control-of-your-envir?ref=users (visited on 12/01/2013).

[3] ADRIAN COLYER. *Why should I care about OSGi anyway?* 2008. URL: http://spring.io/blog/2008/05/15/why-should-i-care-about-osgi-anyway/ (visited on 12/01/2013).

[4] DEFA. *DEFA offical website*. 2013. URL: http://www.defa.com/no/fp/ (visited on 12/01/2013).

[5] DEFA. *HyttaMi offical website*. 2013. URL: http://www.hyttami.no/om_hyttami (visited on 12/01/2013).

[6] Wikipedia. *OSGI*. 2013. URL: http://en.wikipedia.org/wiki/OSGi (visited on 12/01/2013).

[7] Wikipedia. *Wikipedia–Kickstarter*. 2013. URL: http://en.wikipedia.org/wiki/Kickstarter (visited on 12/01/2013).