

Exercise 1: Software Architecture Description of the HS07 System

Marjus Nielsen

Department of Computer Science, University of Aarhus

Aabogade 34, 8200 Århus N, Denmark

India

20064684

marjus@daimi.au.dk

6. feb. 2008

Abstract

The HS07 system implements a closed-loop control of the heating in a private home. It monitors thermometers in the home, and based on measurements HS07 adjusts radiators in the home. This report gives a software architecture description of an architectural prototype of the HS07 system. The techniques used for architectural description are taken from [Christensen et al., 2004].

1 Introduction

Figure 1 shows a schematic overview of HS07 in a home. The home may be accessed by the home owner from the outside through the HS07 gateway. The HS07 gateway also monitors and controls the home.

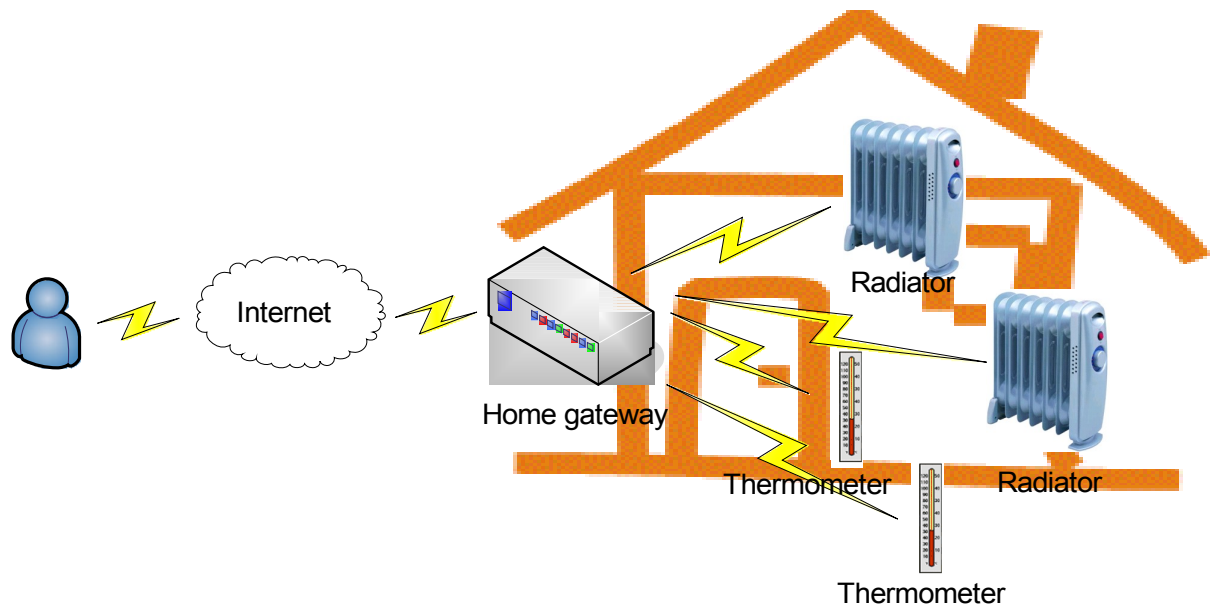


Figure 1 HS07 in a home

HS07 includes sensor and actuator hardware which runs on an embedded Java virtual machine with standard software.

2 Architectural requirements

For our purposes there is one main use case for the HS07 system:

Control Temperature: The gateway collects measurements from thermometers and reports this to radiators that then control the temperature.

The major driving quality attributes of the HS07 system are:

- *Performance.* HS07 should be performant so that a large number of thermometers and radiators may be part of the system.
- *Modifiability.* It must be possible to modify HS07 to include new types of sensors and actuators.
- *Simplicity.* Must be simple to add new thermometers and radiators.

3 Architectural Description

3.1 Module Viewpoint

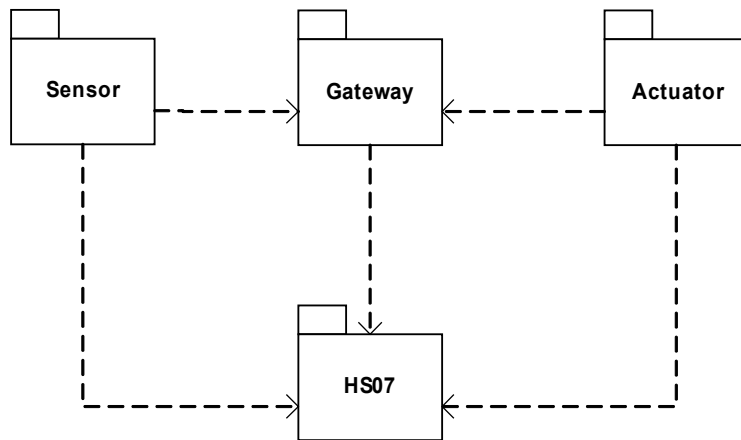


Figure 2 Package diagram.

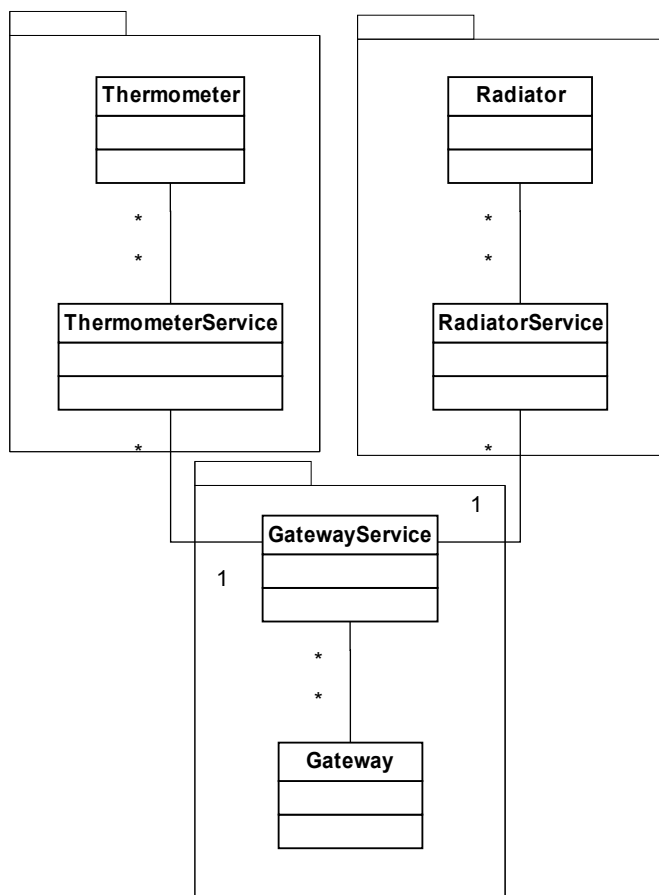


Figure 3 Class diagram. The classes shown on the diagram are associated via the classes in the HS07 package (not shown on the diagram).

3.2 Component & Connector Viewpoint

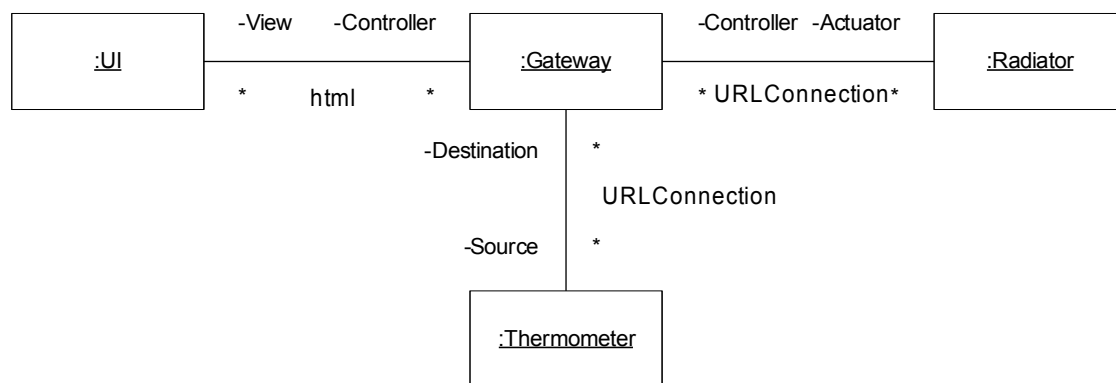


Figure 4 Component & Connector overview

3.3 Allocation Viewpoint

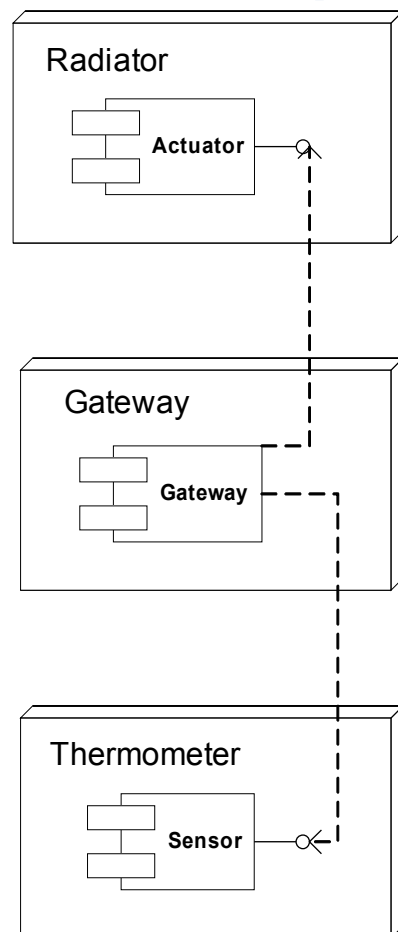


Figure 5 Deployment view. The Gateway reads the temperatures from the Thermometers and notifies all Radiators.

4 Discussion

The gateway is central to the system. That makes the system simple, because new components only have one gateway to connect to. But it is also a single-point-of-failure, so that if the gateway for some reason becomes defect, the whole system is defect. Hence, if stability is a priority, the system could be redesigned to have multiple collaborating gateways. The scalability of the HS07 would

benefit from multiple control units. But this would of course add complexity and it's doubtful if this kind of system would need the extra scalability.

Controllers in the provided java code are objects that don't conform to any interface and are neither derived from a base class. I would recommend having a Controller interface that the thermometer, radiator and gateway classes should implement, to add robustness and to be able to get a more type-safe system. This also adds to the quality attribute that the system should be modifiable, because with interfaces new types of controller could be added more easily and consistently.

A heating control system of this kind should have some notion of placement within the home, so that each room can be controlled independently. An average of the temperatures in each room would probably not be sufficient.

5 Questions

"For the architectural description above, discuss what (if anything) should be changed or added for it to comply with the IEEE recommended practice for architectural description"

IEEE specifies these elements to be in Architectural descriptions

Architectural documentation

This element contains information about the architectural document, like summary, change history, references and scope. This document has date, author and abstract but lacks glossary, change history, scope, status of the architectural description and glossary.

Stakeholders and concerns

Stakeholders are not identified for this system. To comply with IEEE users, acquirers, developers and maintainers of the system should be identified and their concerns registered.

Architectural viewpoints and views

This document uses the module, component & connector and the allocation viewpoints and corresponding views.

Consistency among architectural views

No consistency analysis has been made of the architectural description. This is required by IEEE.

Rationale

A rationale should be written for decisions made and alternative concepts should also be documented.

"Consider the definition of software architecture by [Perry and Wolf, 1992]. Discuss what the 'elements', 'form', and 'rationale' according to this definition would be for the HS07 system"

In [Perry and Wolf, 1992] there are 3 kinds of elements: processing elements, data elements and connecting elements. In our case the classes of the class diagram in figure 3 are processing elements,

and the data elements are few and simple. We have temperatures being processed and the radiators have a state (on or off). The ProtocolServlet and the Invoker classes are connecting elements.

The form consists of weighted properties and relationships. Properties and relationships are not clearly constrained in our description.

Like in the IEEE case, [Perry and Wolf, 1992] suggest stating the rationale for choices made when architecting the system.

6 References

[Christensen et al., 2004] Christensen, H., Corry, A., and Hansen, K. (2004). An approach to software architecture description using UML. Technical report, Computer Science Department, University of Aarhus.