System documentation of a heating control system

This document has been produced, within the context of software engineering courses at the University of Kaiserlslautern by Marco Heide, Wolfgang Wagenbichler, Andreas Schank, Angela Schmidt and Axel Seck, then reworked by Christiane Differding and Matthias Gutheil. It has been translated in english and adapted by Mario Negro Ponzi of Politecnico di Torino. It has subsequently undergone extensive rework by Vincenzo D'Elia, Alfredo Pironti and Massimiliano Schillaci of Politecnico di Torino.

V 4.0 October 2012

Abstract

A heating control system monitors and controls temperature in a building composed of several rooms. It is a good example of a system with software and non-software parts. This document contains the system requirements, the system design, the software requirements and the software design

Table of contents

Part 1 Problem description

The heating control system should be developed, to control heating in a house, optimizing energy efficiency and comfort. Below the idea of this system is described.

1.1 Problem

Up until today the temperature of every room is regulated by a thermostat in the heater in the room. A target temperature for each room can be set. An automatic management of the temperature is based only on heaters. A window also is useful to control temperature in a single room. But it should be opened or closed by someone. Water in heaters has a fixed temperature set by a boiler.
Problems in temperature management are:

1. Heater's temperature cannot be differently set basing on heating demand or time. There is no regulation linked with people presence
2. Thermal regulation is limited to heating. The use of windows to cool down is based on user intervention
3. There is no optimization of boiler temperature. The boiler's temperature is set to the maximum value regardless whether the temperature is too low or too high. That produces an energy loss.

The above control system is not satisfying both because of energy loss and because it is not comfortable.

1.2 Expected functionalities

The user wants to be able to individually set temperature for each room using a terminal. So he should be able to provide described parameters.
First it should be possible to set a temperature that could be maintained when there is someone in the room. In doing so the system should not react immediately when someone comes in. It should be provided a time span that indicates how long a person should stay in the room before the system takes control. The user should also be able to set the time span in which the desired temperature should be reached. The provided time span can be maintained heating up heaters for a small period of time over the desired temperature.
Also the user should be able to set a minimum temperature that is acceptable when there is no one in the room. In this case, too, a time span should be provided, after everyone has left the room, before the system takes control.
There should also be default values available for every parameter.
Windows, too, should be controlled by the system. If a window has been completely opened by a user, the system should maintain only the minimal temperature. If the window is open and it starts raining the system should put it in tilt position. More, the windows should be usable by the system to cool down when the outside temperature is cooler, by tilting them.

1.3 Environment characteristics

Below are described the characteristics of the environment inside which the system will operate.

1.3.1 House and rooms

The house is composed of a number of rooms.
The house also has:

- an outside temperature sensor
- a rain switch (closed if it rains)

Each room is so configured (as a result of the whole system concept):

- 1 window with two switches and two actuators. A switch gets opened when the window is completely open (open when the window is open / NC = normally closed). The other when is tilted (open when the window is tilted / NC = normally closed). When the window is completely open the tilt switch is closed. An actuator swings the window while the other tilts it.
- 1 heater (hot water) with a thermostat

- 1 presence switch (closed when someone is in the room)
- 1 room temperature sensor

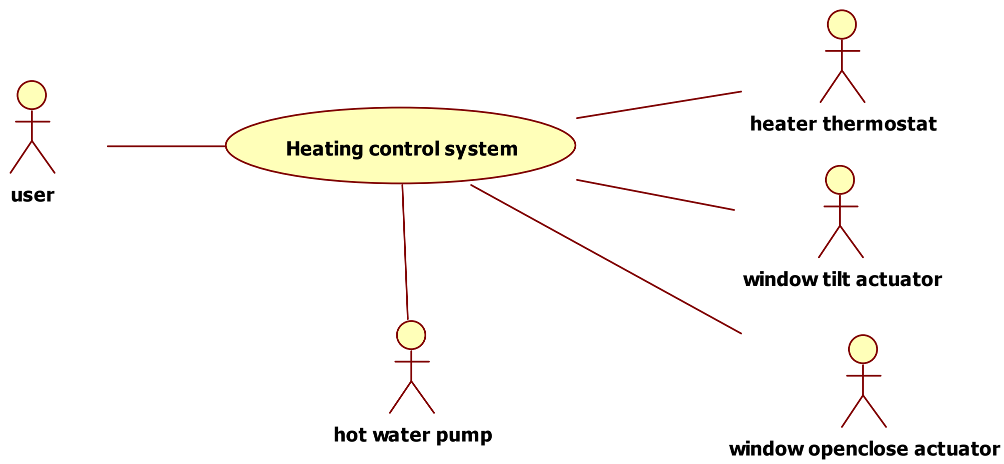
1.3.2 Boiler and water pump

Heat is produced by a centralized boiler, its control is an already deployed autonomous subsystem. The maximum temperature is fixed at 70 degrees. The house receives heat through a single pipe that brings hot water to the heaters. It can be assumed that there are no dispersions, so water is at 70 degrees. The heating control system controls the intake flow of hot water through a pump.

Part 2 System requirements

2.1 Context diagram and interfaces

The context diagram specifies what is meant by 'System' and what are its interfaces with the external world. See System design to know what is inside the system.



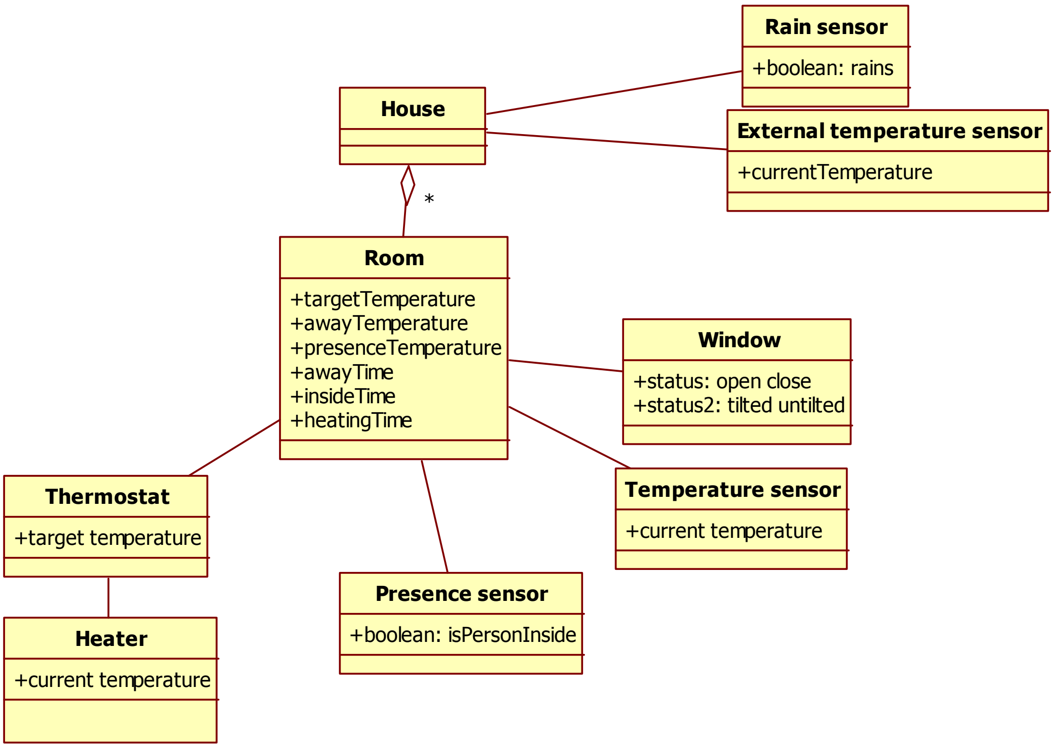
Interfaces

With object	Physical	Logical
User	Touch screen	Screens and buttons
Window tilt actuator	RS232	command protocol, see vendor tech spec main commands are read position, set position
Window open close actuator	RS232	command protocol, see vendor tech spec, main commands are read position, set position
Hot water pump	Bipolar wire, 12 V DC	close = pump ON, open = pump OFF
Heater thermostat	RS232	command protocol, see vendor tech spec, main commands are read temperature, set temperature

Remark that there is only one hot water pump, but one heater thermostat and window actuator per room.

2.2 Glossary

Here we describe the key parts and concepts in the case considered. The class diagram below does NOT describe what is inside or outside the system.



A house has several rooms, a rain sensor (tells if it rains or not) and an external temperature sensor (reads temperature outside). A room has a heater (radiates heat from water to air) controlled through a thermostat (open closes a valve that controls water intake in the radiator, trying to keep a set target temperature), a window (can be open closed, or tilted), a presence sensor (tells if a person is inside or not), a temperature sensor (reads the current temperature in the room).

To control temperature in the room, a room defines

- Target temperature: desired temperature in the room
- Away temperature: what should be target temperature when no person is inside
- Presence temperature: what should be target temperature when a person is inside
- Away time: time to wait, after a person exits the room, before reducing target temperature to away temperature
- Inside time: time to wait, after a person enters the room, before rising target temperature to presence temperature
- Heating time: maximum time before which the room should get to target temperature after a person enters

2.3 Functional Requirements

UR-F	Requirement's description
Temp-UR-F 1	The user shall be able to set *PresenceTemp* in °C
Temp-UR-F 2	The user shall be able to set *AwayTemp* in °C
Temp-UR-F 3	The user shall be able to set a time span *HeatingTime* in minutes
Temp-UR-F 4	The user shall be able to set a time span *AwayTime* in minutes
Temp-UR-F 5	The user shall be able to set a time span *InsideTime* in minutes
Temp-UR-F 6	The temperature of Temp-UR-F 1 shall be maintained when there is someone in the room and the *InsideTime* from Temp-UR-F 5 has elapsed
Temp-UR-F 7	The temperature of Temp-UR-F 2 should be maintained in the room when the room is

	empty and *AwayTime* from Temp-UR-F 4 has elapsed
Temp-UR-F 8	The time span from Temp-UR-F 3 indicates after how many minutes the *PresenceTemp* from Temp-UR-F 1 after the entrance of a person in the room shall be reached.
Temp-UR-F 9	The time span from Temp-UR-F 4 indicates for how many minutes the *PresenceTemp* from Temp-UR-F 1 shall be maintained after the last person has left the room
Temp-UR-F 10	The time span from Temp-UR-F 5 indicates for how long shall someone stay in the room for the system to get in control
Temp-UR-F 11	The *HeatingTime* from Temp-UR-F 3 must be at least 5 minutes longer than the *InsideTime* from Temp-UR-F 5
Temp-UR-F 12	For each quantity from Temp-UR-F 1, Temp-UR-F 2, Temp-UR-F 3, Temp-UR-F 4, Temp-UR-F 5 default values shall exist.
Temp-UR-F 13	If a window gets opened the system maintains the room temperature at *AwayTemp*, whether or not there is someone in the room.
Temp-UR-F 14	If it starts raining with the window open, the system will set the window in a rain secure position.
Temp-UR-F 15	When the temperature of a room with someone inside exceeds that from Temp-UR-F 1 and the outer temperature is lower than the room temperature, the window in this room gets tilted
Temp-UR-F 16	The control of the temperature is admitted with an error of +/-1 °C
Temp-UR-F 17	The temperature of the boiler shall be set to the maximum required by any heater, not to waste energy.

Table 1

2.4 Non-functional requirements

The system should complain also with the following requirements.

UR-NF	Requirement's description
UR-NF 1	The system should be easily changeable.

Table 2

2.5 Inverse user requirements

Following situations should be avoided.

UR-Inv	Requirement's description
UR-Inv 1	When there is nobody in the room and the room is hotter than target temperature, the window does not get tilted to cool.
UR-Inv 2	If the user manually opens the window, the system shall not tilt it to the rain secure position.

Table 3

2.6 Design decisions

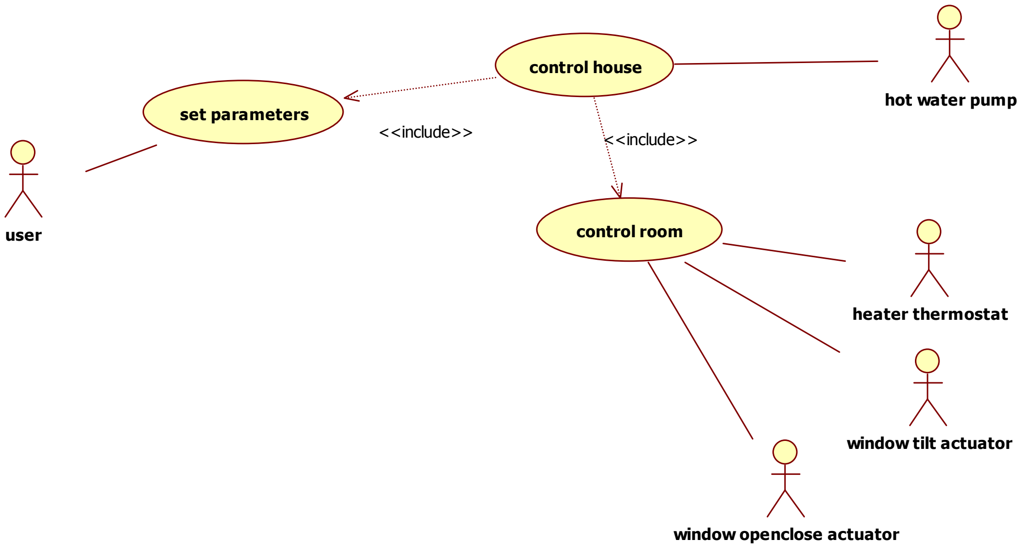
Following decisions have been taken.

DD	Requirement's description
DD 1	The system should be object oriented.

Table 4

2.7 Use case diagram

The main use cases are summarized in the following use case diagram.



Use Case: set parameters

The activating actor of this Use Case is the user, it deals with setting and verifying all user settable parameters. It is composed of:

- Scenario 1
- Scenario 8

Use Case: control room

It deals with controlling heater and windows in function of the target temperature set and the presence (or not) of a person in the room. It is composed of:

- Scenario 2
- Scenario 3
- Scenario 4
- Scenario 5
- Scenario 6
- Scenario 7
- Scenario 9
- Scenario 10
- Scenario 11

Use Case: control house

It deals with overviewing all rooms and interacting with the water pump. It is composed of

- Scenario 12
- Scenario 13

2.7.1 Scenarios for Use Cases

The following scenarios detail the use cases.

Other notes (to the following table):
(-) after a User Requirement indicates that a specific precondition (e.g. a time span elapsing) has not been met
(+) after a User Requirement indicates that a specific precondition has been met
Without any other specific note, in the following scenarios those following default values should be considered:
PresenceTemp: 20 °C
AwayTemp: 15 °C
InsideTime: 5 minutes
AwayTime: 15 minutes
HeatingTime: 10 minutes

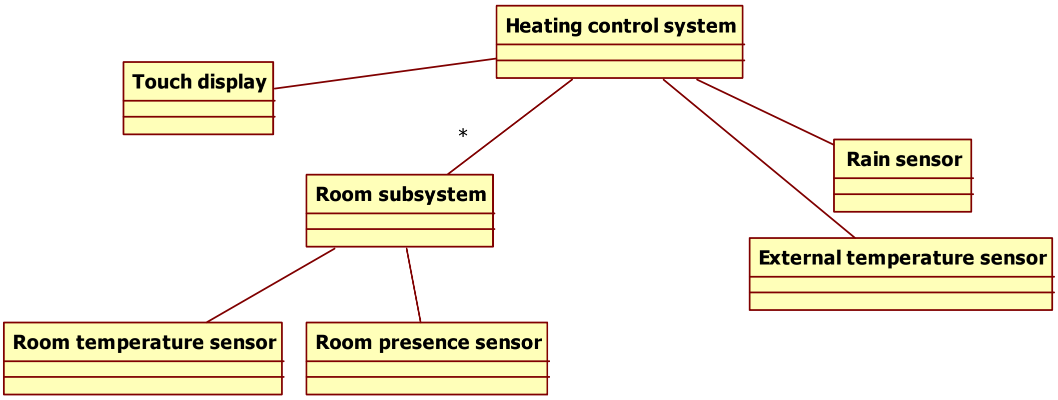
#	Step	Scenario's description	User requirement
Scenario 1	*User changed HeatingTime, InsideTime and PresenceTemp.*		
	1.	**Presence switch**: closed **Outside Temperature**: 21 ºC **Window**: closed **Rain switch**: open **Target temperature**: 19 ºC (PresenceTemp) **Room temperature**: 18 ºC	
	2.	The user sets InsideTime to 5 minutes	Temp-UR-F5
	3.	The user sets HeatingTime to 9 minutes	Temp-UR-F3
	4.	The system ignores new time set, because it is just 4 minutes over InsideTime and throws an error message	Temp-UR-F11
	5.	The user sets HeatingTime to 10 minutes	Temp-UR-F3
	6.	The user sets AwayTemp to 16 ºC	Temp-UR-F2
	7.	The user sets AwayTime to 10 minutes	Temp-UR-F4
	8.	The user sets PresenceTemp to 21 ºC	Temp-UR-F1
	9.	The target temperature is set at 21 ºC	
Scenario 2	*The user leaves the room before the system activates*		
	1.	**Presence switch**: open **Outside temperature**: 21 ºC **Window**: closed **Rain sensor**: open **Target temperature**: 15 ºC (AwayTemp) **Room temperature**: 18 ºC	
	2.	A user enters in the room	
	3.	The user leaves the room before the InsideTime has elapsed	Temp-UR-F10 (-)
	4.	The target temperature does not change	Temp-UR-F7
Scenario 3	*The user leaves the room for a long time, affecting system behavior*		
	1.	**Presence switch**: closed **Outside temperature**: 16 ºC **Window**: closed **Rain sensor**: open **Target temperature**: 20 ºC (PresenceTemp) **Room temperature**: 19 ºC	
	2.	The user leaves the room, which becomes empty	
	3.	After AwayTime the target temperature is set to AwayTemp.	Temp-UR-F7

	4.	The window does not get opened because there is no one inside the room.	UR-Inv 1
Scenario 4	*There is only one user in the room. This user leaves for a short time the room so that it does not have any effect on the system behavior*		
	1.	**Presence switch**: closed **Outside temperature**: 19 ºC **Window**: closed **Rain sensor**: open **Target temperature**: 20 ºC (PresenceTemp) **Room temperature**: 19 ºC	
	2.	The use leaves the room, which becomes empty	
	3.	The user comes back in the room before AwayTime has elapsed.	Temp-UR-F7 (-)
	4.	The temperature does not change	Temp-UR-F6
Scenario 5	*The window is open and a user is present. Target temperature is set to AwayTemp*		
	1.	**Presence switch**: closed **Outside temperature**: 19 ºC **Window**: closed **Rain sensor**: open **Target temperature**: 20 ºC (PresenceTemp) **Room temperature**: 20 ºC	
	2.	The user opens the window	
	3.	Target temperature is set to AwayTemp	Temp-UR-F13
Scenario 6	*It rains with the window open*		
	1.	**Presence switch**: closed **Outside temperature**: 17 ºC **Window**: completely open **Rain sensor**: open **Target temperature**: 15 ºC (AwayTemp) **Room temperature**: 18 ºC	
	2.	It starts raining	
	3.	The system closes the window	Temp-UR-F14
	4.	Target temperature is set to PresenceTemp	Temp-UR-F6
	5.	The user opens again the window	
	6.	Since the window has been opened manually, it doesn't get closed by the system	Temp-UR-F14 UR-Inv2
	7.	Target temperature is set to AwayTemp	Temp-UR-F13
Scenario 7	*The window gets controlled by the system to cool*		
	1.	**Presence switch**: closed **Outside temperature**: 17 ºC **Window**: closed **Rain sensor**: open **Target temperature**: 20 ºC (PresenceTemp) **Room temperature**: 21 ºC	
	2.	The room temperature raise to 22 °C and so is more that 1 °C over target temperature.	Temp-UR-F16
	3.	The system controls the window to cool down	Temp-UR-F15
Scenario 8	*Scenario for Use Case setValues* *The user sets all values to default*		

	1.	The user sets PresenceTemp to default	Temp-UR-F12
	2.	The user sets AwayTemp to default	
	3.	The user sets InsideTime to default	
	4.	The user sets AwayTime to default	
Scenario 9	*Target Temperature of a room is reached*		
	1.	**Presence switch**: closed **Outside temperature**: 14 ºC **Window**: closed **Rain sensor**: open **Target temperature**: 20 ºC (PresenceTemp) **Room temperature**: 18 ºC	
	2.	Room temperature reach 19 °C	
	3.	Stop temperature has been reached, since room temperature is just 1 °C below target temperature.	Temp-UR-F6 Temp-UR-F16
	4.	Required new boiler temperature is computed	Temp-UR-F18
Scenario 10	*The room must be heated up*		
	1.	**Presence switch**: closed **Outside temperature**: 23 ºC **Window**: closed **Rain sensor**: open **Target temperature**: 20 ºC (PresenceTemp) **Room temperature**: 19 ºC	
	2.	Room temperature drops to 18 °C	
	3.	Heating temperature gets calculated because room temperature has dropped below 1 °C under target temperature.	Temp-UR-F6 Temp-UR-F8 Temp-UR-F16
	4.	Heater temperature is set to the necessary value	
	5.	Required new boiler temperature is computed	Temp-UR-F18
Scenario 11	*Room temperature is above target temperature*		
	1.	**Presence switch**: closed **Outside temperature**: 23 ºC **Window**: closed **Rain sensor**: open **Target temperature**: 20 ºC (PresenceTemp) **Room temperature**: 21 ºC	
	2.	Room temperature raises to 22 °C	
	3.	As the temperature of the room is more than 1 °C over target temperature, heater temperature is set to 0 °C	
	4.	Required new boiler temperature is computed	Temp-UR-F18
	5.	The window is not opened to cool down room temperature.	Temp-UR-F15(-)
Scenario 12	*At least one room requires heat*		
	1.	All rooms have thermostat closed	
	2.	Room 2 opens the thermostat	
	3.	Water pump is set to ON	
Scenario 13	*One more room requires heat*		
	1.	Room 1 and 3 have thermostat open	
	2.	Room 2 opens the thermostat	
	3.	No changes to water pump	

Part 3 System design

This UML class diagrams shows the main components of the system, as defined by the system designers. Remark that the class 'Heating control system' corresponds to the bubble 'Heating control system' in the context diagram. See the context diagram for interfaces of the system with the external world.
A key choice is on having one interaction point between the user and the system. (Another, more expensive, option is to have an interaction point in each room).
Another choice is to have only one computer (as above, another option is to have a computer in each room).
Another choice is wired connections (instead of wireless).



Components

Heating control system	Dedicated computer with touch screen, Cpu, rom, ram, AD converters, DA converters; type of cpu and size of memory to be defined, as compromise between performance and cost At least 12 AD converters Runs Android OS.
Room presence sensor	Infrared sensor, battery powered, analog output (3V)
Rain sensor	Self powered (rechargeable battery and solar cell), analog output (0V dry, 10V rain)
External temperature sensor, room temperature sensor	Self powered (rechargeable battery and solar cell), analog output (0V, 3.5V inverse prop to temperature)

Internal Connections

From	to	type
Temperature sensor (external, room)	Computer (AD converter)	Bipolar electrical wire
Rain sensor	Computer	Bipolar electrical wire
Room presence sensor	Computer	Bipolar electrical wire

\
External connections
See section 2.1 for connection to be managed with external devices.