

Software Architecture in Action

Flavio Oquendo, Jair C Leite, Thais Batista



Chapter 10

Designing Scalability in Software Architectures

Learning outcomes of this chapter

- You will learn:
 - what is scalability as an architectural quality
 - what are the architectural causes and effects of scalability
 - tactics to improve scalability
 - a comparison technique to evaluate the scalability in alternative architectures

The structure of this chapter

- Introduction
- Scalability Causes and Effects
- Scalability Quality Attribute
- Scalability Tactics
- Applying the Scalability Tactics
- Scalability Analysis



Introduction

Scability

Conceptual overview

- Scalability is an quality to refer to the degree to which a product or system can effectively and efficiently be scaled for an increasing operational usage
- An example of scalability in the case of a temperature monitor system is to maintain its performance when the number of sensors increases in orders of magnitude
 - Increasing from 1 to 10 sensors, and then from 10 to 100, and then from 100 to 1000, and so on

Scalability

Expressing scalability using software architecture concepts

- Scalability as an architectural quality refers to the degree to which an architecture can effectively and efficiently be scaled for an increasing number of component usage of several orders of magnitude
 - the rise of the number of components can imply in the rise of the number of connectors and ports
- We need to express the causes and the effects
 - Causes refer to identify the component types to which their uses (number of instances) can increase in several orders of magnitude
 - Effects refer to impact on the effectiveness and efficiency of the system implied by its architecture when the number of component use increases



Scalability Causes and Effects

Scalability causes

- Component types can have their use (number of instances) significantly increased in the software architecture
 - connector types may have their use (number of instance) significantly increased as a consequence of the rise of component use
 - port types may have their use (number of instance) significantly increased as a consequence of the rise of component use
 - It is important to note that, if the architecture defines that a component type can have only one instance, we cannot increase the number of instances
- In the RTC system, we can increase the number of temperature sensors and presence sensors
 - We cannot increase the user interface and the controller component types as the architecture defines only one instance of each type.

Scalability effects

- Effects are the consequences of the causes in the architecture
 - the impact on the effectiveness and efficiency of the system
- According to the requirements of the RTC System
 - Effectiveness is to assure that the control-loop correctly adjust the room temperature
 - according to the user desired temperature, the current average temperature and the presence of a person
 - Efficiency refers to the time of the control-loop to adjust the temperature

Scalability effects

Example in RTC

- In the RTC System ARCH2, the causes are
 - we can increase the number of temperature sensors
- The effects in RTC System ARCH2 are
 - According to the cardinality, the minimum number of temperature sensor is 1
 - When the number of temperature sensor is increased by 10, the number of connectors is also multiplied by 10 and they are connected to the same port
 - Then, the time to read all sensors is multiplied by 10
 - As a consequence, the efficiency is divided by 10 (effect)
 - And the system is not effective if the efficiency is too low

Scalability quality attributes

- Scalability quality attribute refers to a quality used to quantify the effects
 - The ripple effect in system performance
 - Response time
 - Throughput
 - Resource consumption
 - As an example, if we increase the number of temperature sensors by ten, we expect not to impact the response time of the feedback control loop

Scalability Tactics

- Tactics are design decisions that influence quality attributes
 - Tactics are used to minimize effects
- Example of scalability tactic:
 - Introduce concurrency
 - To improve scalability we can introduce concurrence in order to keep performance
 - As an example, in the RTC System, we can introduce concurrency in the sensor monitor component to concurrently read the temperature from the sensors



Applying Scalability Tactics

Applying the Scalability Tactics

Designing a new architecture – ARCH3

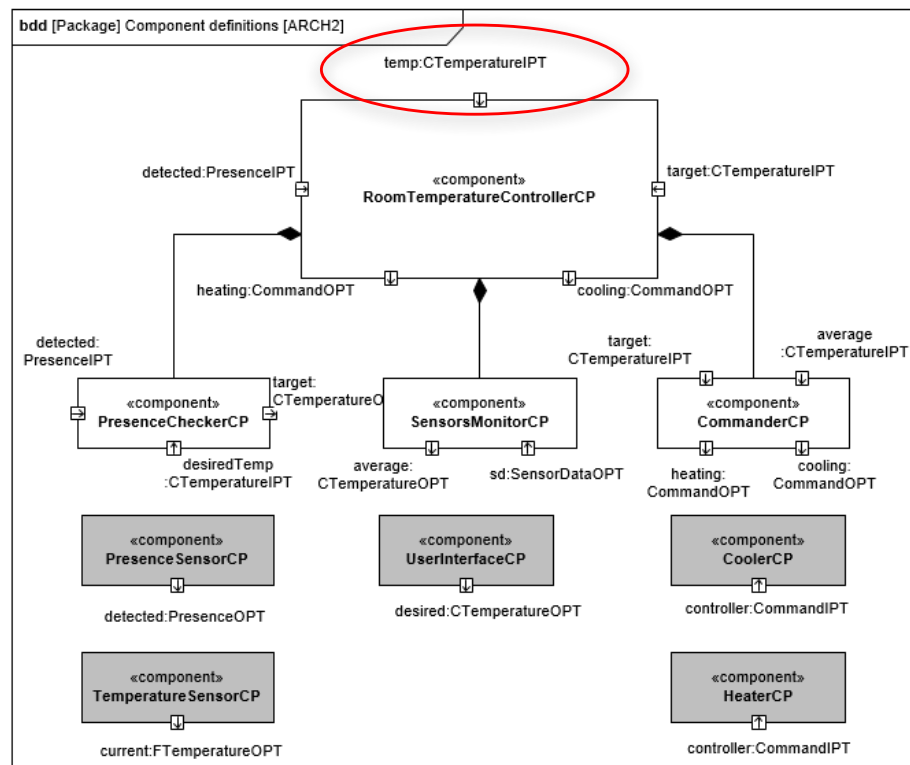
- To allow a significant rise of components (by order of magnitude) we design a new architecture of RTC System
- In the new architecture ARCH3, we apply the *introduce concurrence* tactic
 - we design a *CompositeMonitorCP* component that can read all sensors data concurrently
 - *CompositeMonitorCP* has one *SensorReadersCP* component for each *TemperatureSensorCP*
 - a n-ary connector (*AllTemperaturesCN*) joins all output from the *SensorReadersCP* components to the *AverageCalculatorCP* component
- In the next slides we present the differences between the two architectures

Applying the Scalability Tactics

The Component BDDs of ARCH2 and ARCH3

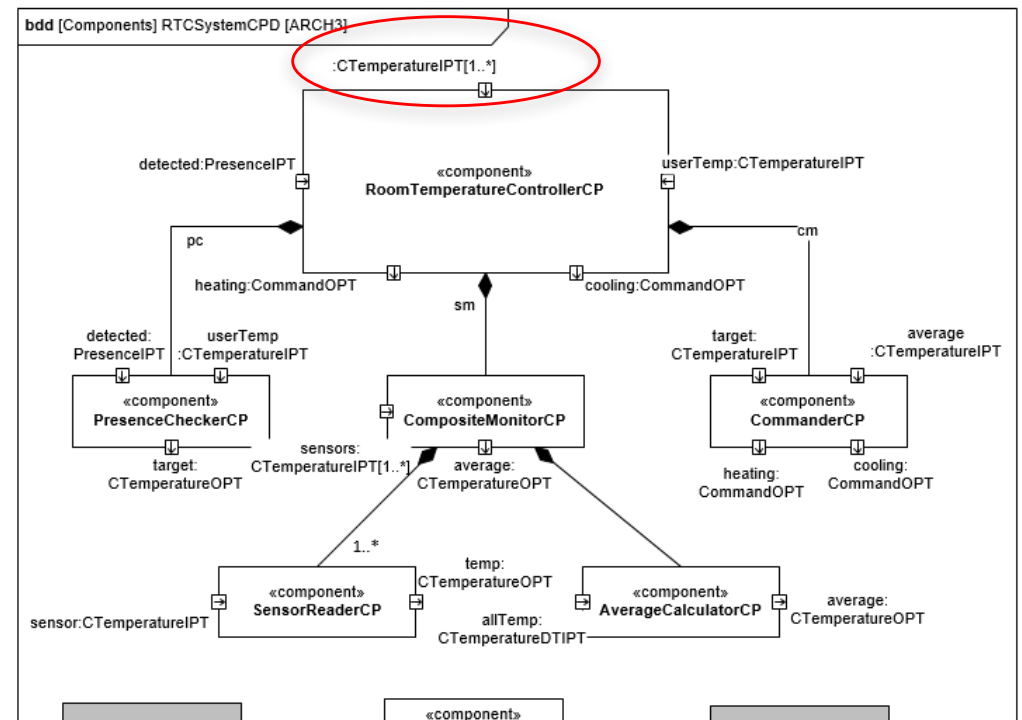
ARCH2

- ARCH2 has a simple component to read data from all sensors



ARCH3

- ARCH3 has a *CompositeMonitorCP* has a *SensorReadersCP* component for each *TemperatureSensorCP*

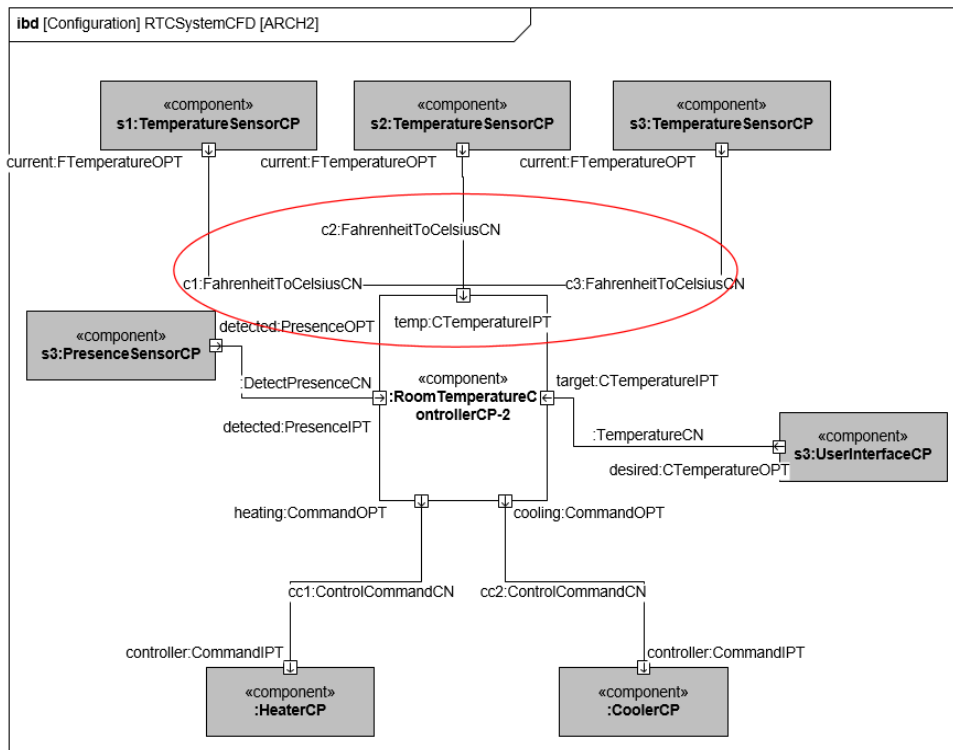


Applying the Scalability Tactics

Two different configurations of ARCH2 and ARCH3

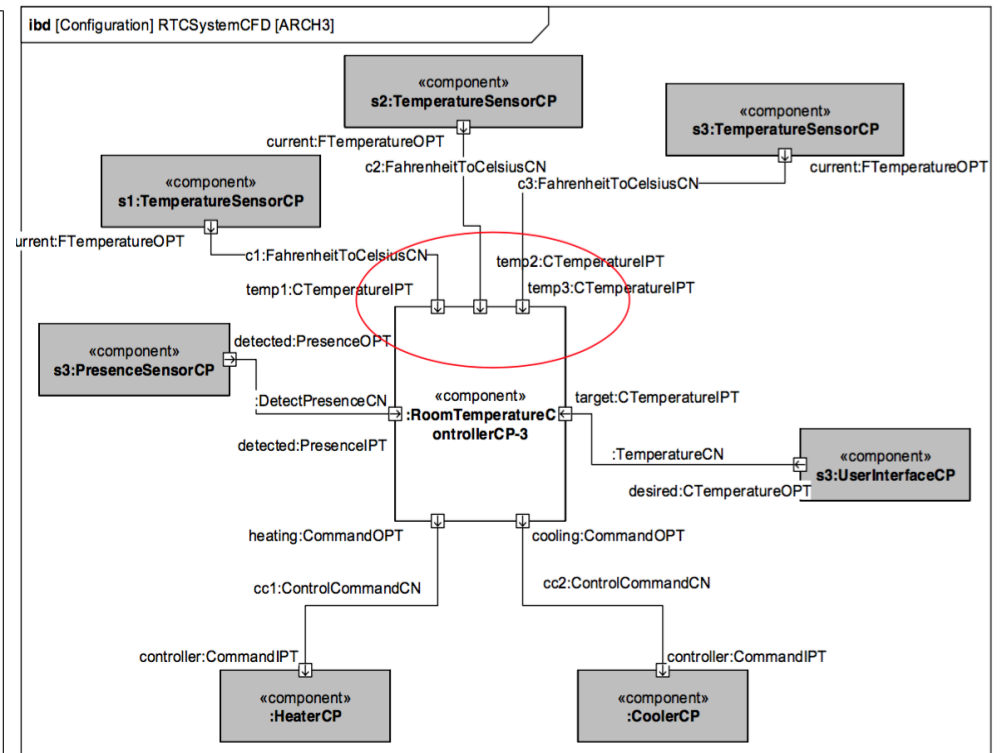
ARCH2

- In ARCH2, several connectors are linked to the *same* temp port



ARCH3

- In ARCH3, several connectors are linked to the *diferente* temp port (cardinality 1..n)

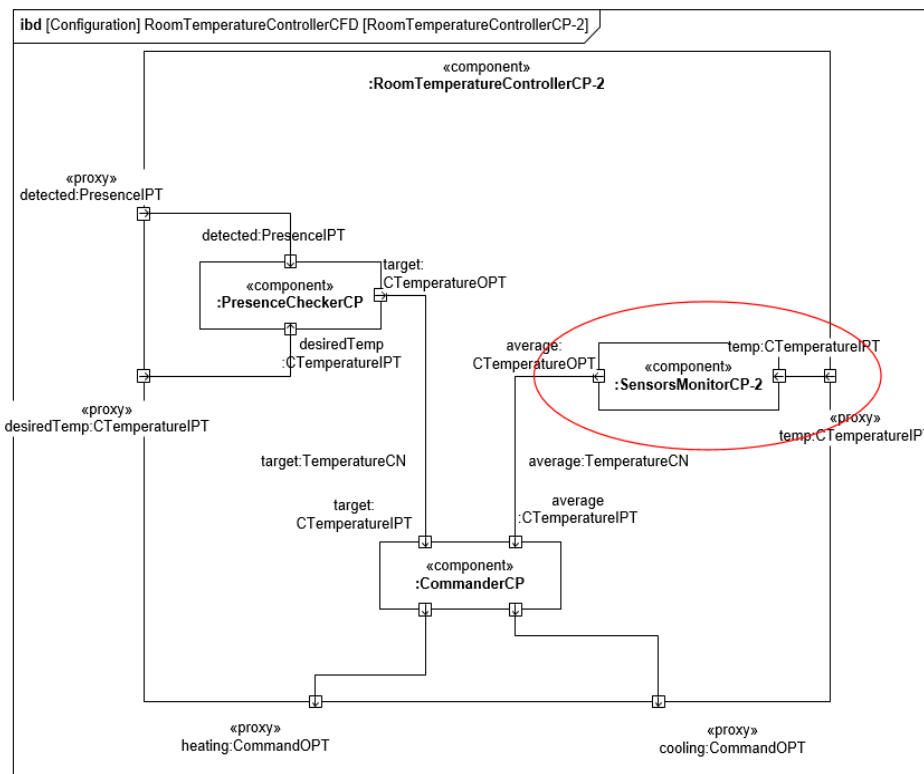


Applying the Scalability Tactics

Two different configurations of ARCH2 and ARCH3

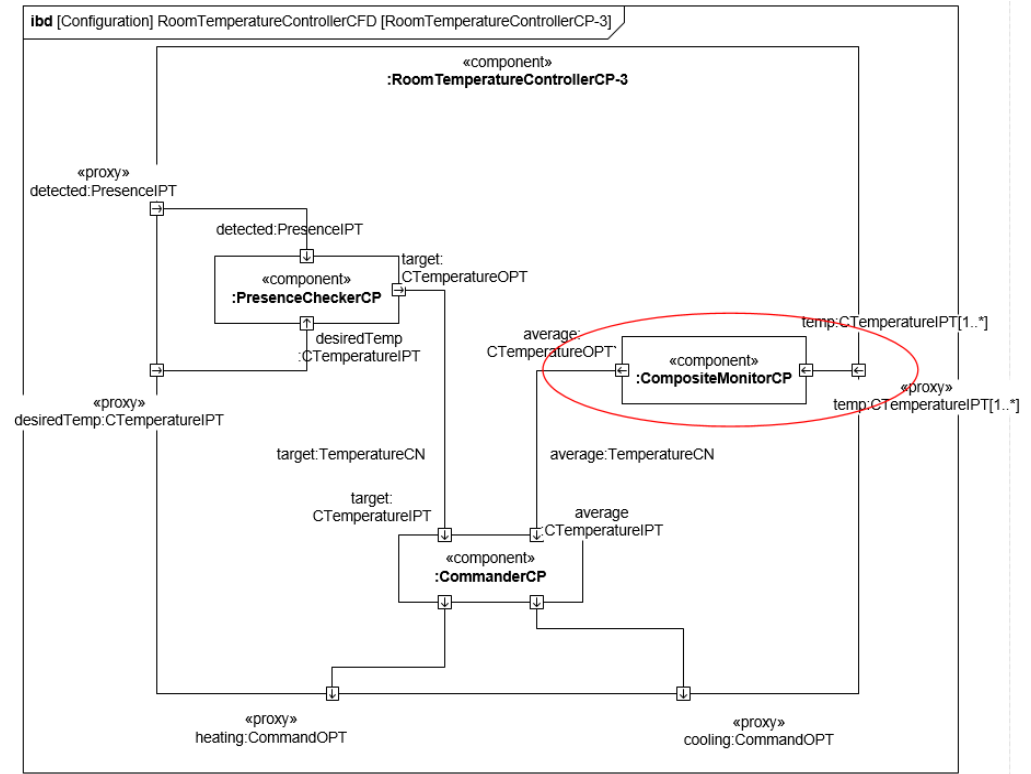
ARCH2 - RoomTemperatureControllerCP

- The *SensorMonitorCP* component reads the data from all sensor sequentially



ARCH3 - RoomTemperatureControllerCP

- The *CompositeMonitorCP* component that can read all sensors data concurrently



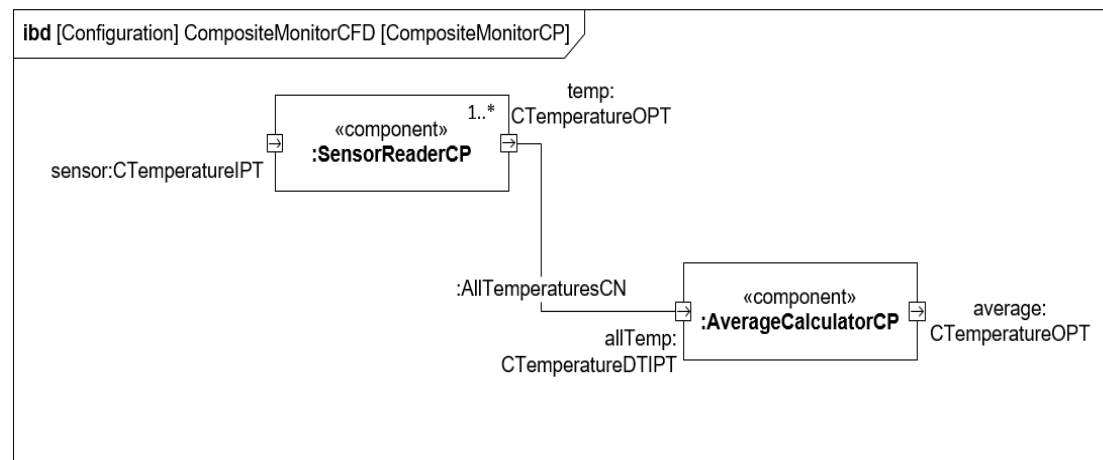
Applying the Scalability Tactics

The CompositeMonitorCP component in ARCH3

Differences between ARCH2 and ARCH3

- In ARCH2, the SensorMonitorCP is a simple component
 - It does not have a configuration
- In ARCH3, the CompositeMonitorCFD is a composite component
 - *CompositeMonitorCP* has one *SensorReadersCP* component for each *TemperatureSensorCP*
 - The *AllTemperaturesCN* (n-ary) connector joins all output from the *SensorReadersCP* components to the *AverageCalculatorCP* component

ARCH3



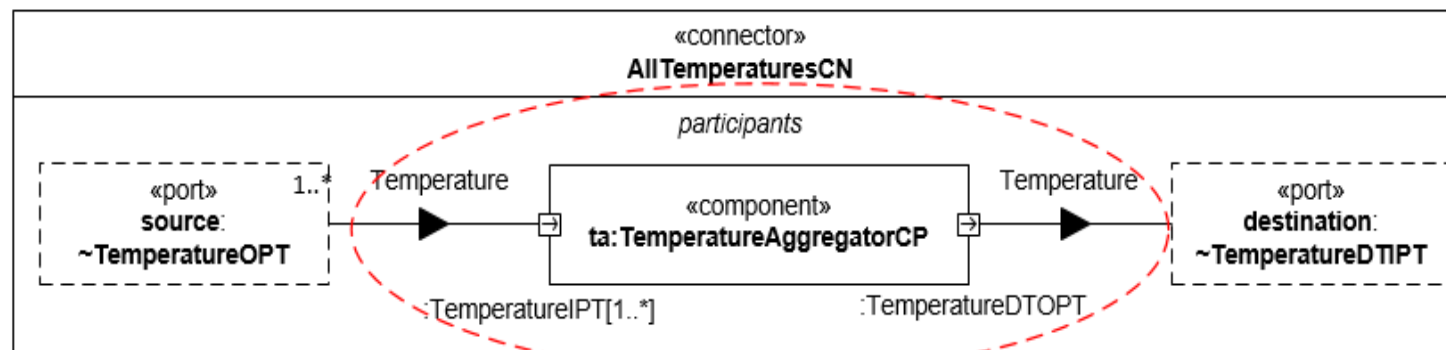
Applying the Scalability Tactics

The AllTemperaturesCN connector in ARCH3

ARCH2

- The configuration of the AllTemperaturesCN connector is described below.
 - The configuration includes a component to aggregate the data from the ports that are linked to the sensors
 - They TemperatureAggregatorCP connector provide a data structure with all data through the TemperatureDToPT port

Example in SysADL





Scalability Analysis

Scalability example

RTC System – Scalability requirements

- We use an AND/OR tree to depict the requirements (R), causes (C) and effects (E)
- RTC Scalability requirements
 - R1 – The system must allow the rise of the temperature sensors of a defined type in an order of magnitude (multiply by 10) to improve accuracy

Scalability example

RTC System – causes

- RTC Scalability requirements
 - R1 – The system must allow to increase the number of temperature sensors to improve accuracy while maintaining response time
 - C1 – Increase the temperature sensors of a defined type in an order of magnitude (multiply by 10) to improve accuracy

Scalability example – 1/2

Analysing the ripple effect

- We analyse the response time on two different architectures of the RTC system considering the same cause
 - the modification is to improve temperature measures (Requirement R1) by increasing the temperature sensors of the same type by an order of magnitude to the RTC system (Cause C1)
 - first, we analyse the ARCH2 architecture
 - this architecture has one multiplex port in the controller to receive data from many sensors
 - If we multiply the number of sensors by 10, the response time is proportionally increased by 10

Scalability example – 1/2

Analysing the ripple effect

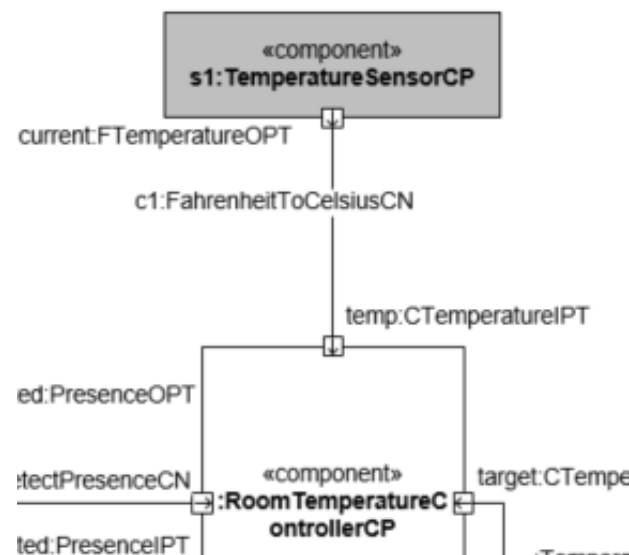
- then, we analyse the ARCH3 architecture
 - this architecture has a composite monitor concurrently that reads all temperature
 - therefore, if we multiply the number of sensors by 10, the response time do not increase
- According to the aforementioned analysis, we can conclude that ARCH2 is not scalable and ARCH3 is scalable

Scalability Analysis

Comparing the time for reading sensors in ARCH2

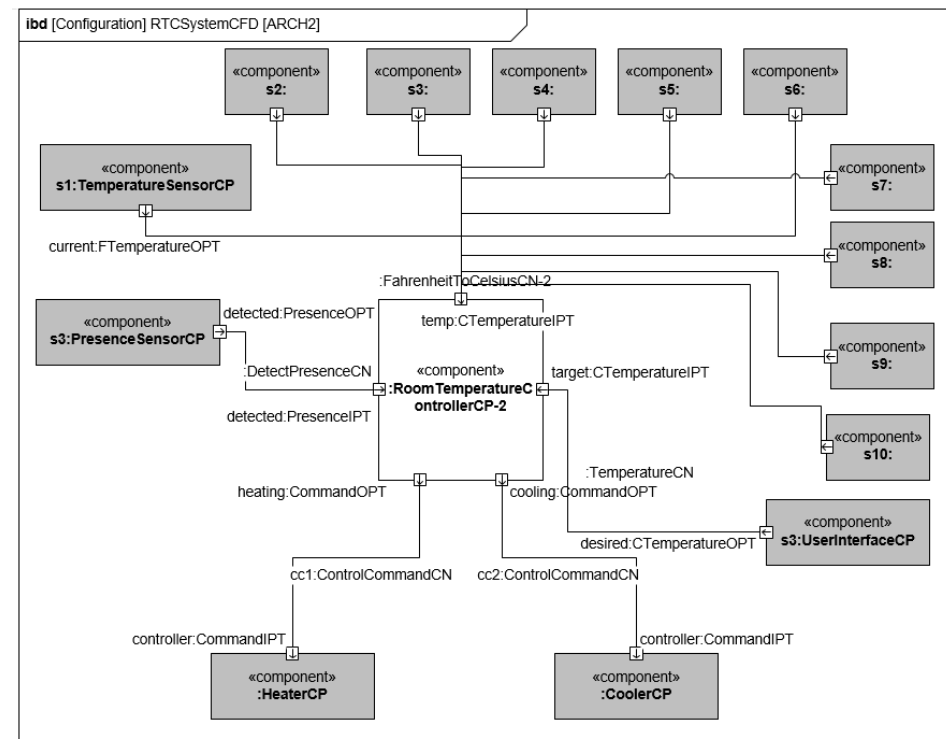
Reading one sensor

- In ARCH2, to read one sensor, one unit of time is necessary



Reading ten sensors

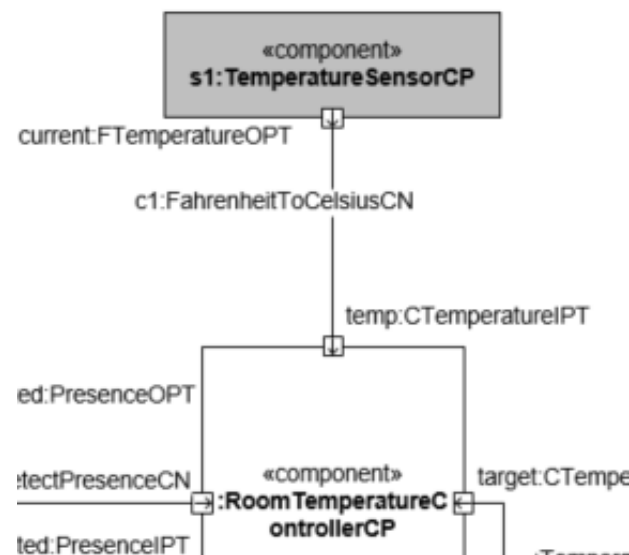
- If we have ten sensors, the monitor reads one temperature at a time, so it needs ten units of time



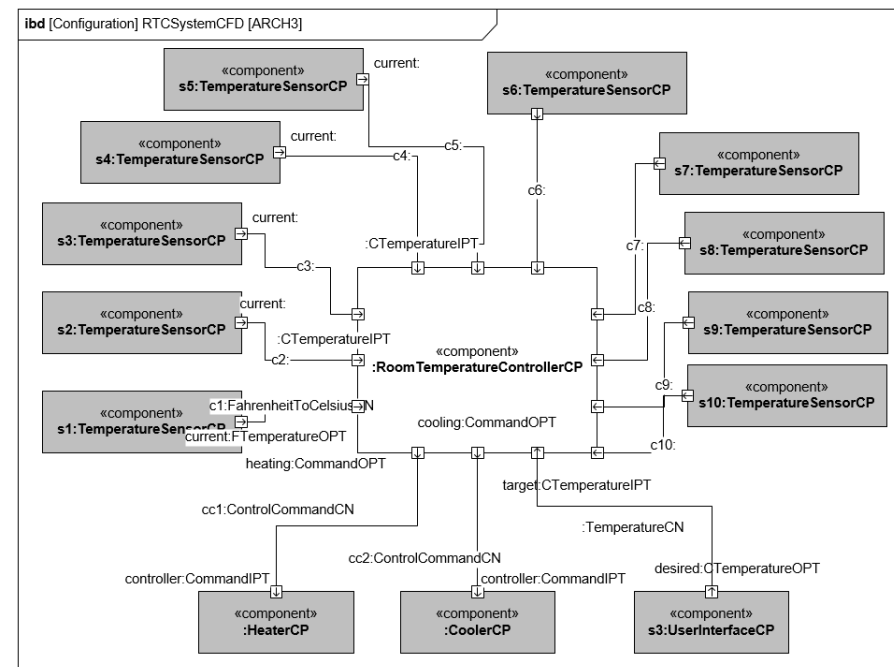
Scalability Analysis

Comparing the time for reading sensors in ARCH3

- In ARCH3, to read one sensor, one unit of time is necessary

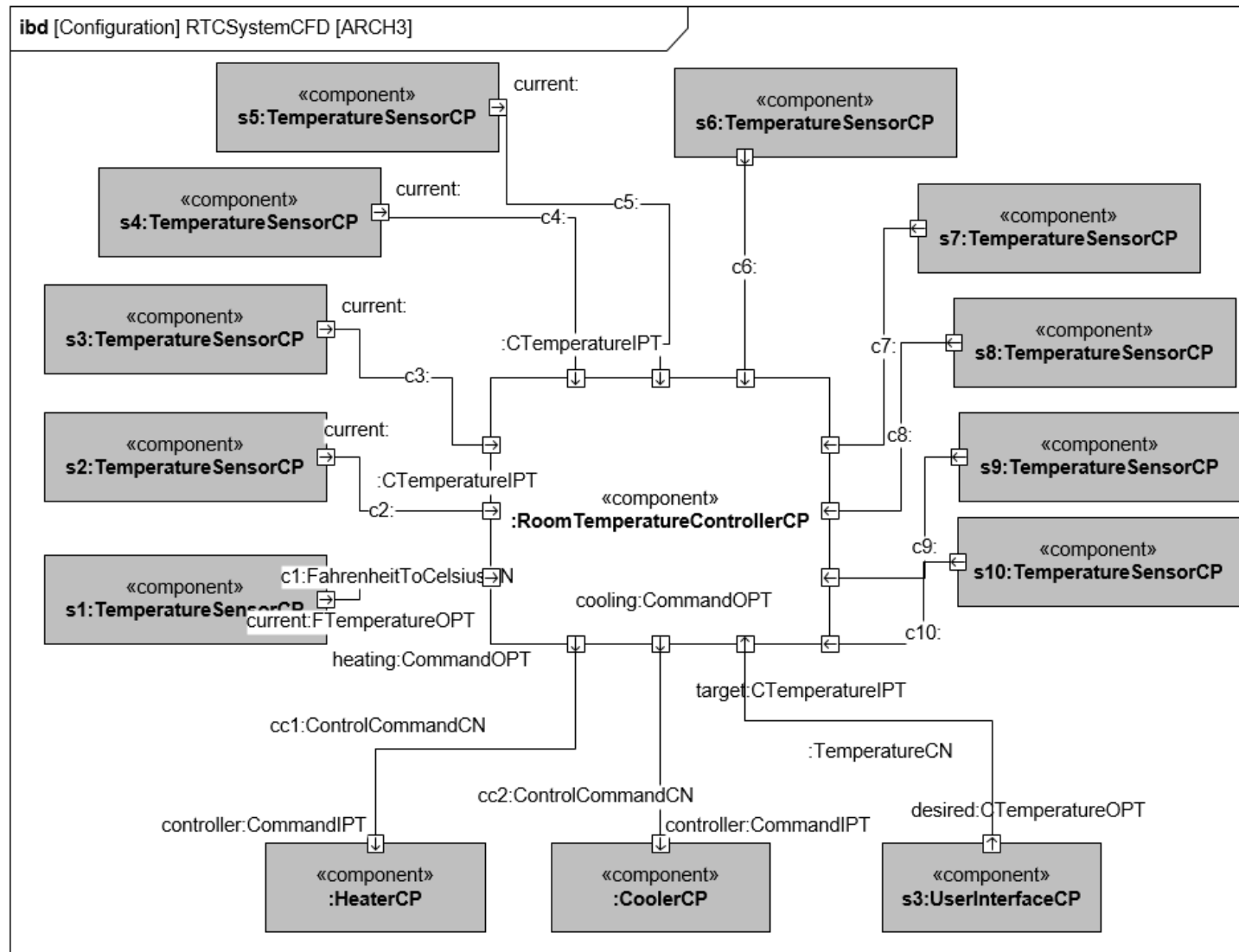


- If we have ten sensors, the monitor reads all temperature at the same time, so it reads all them in one unit of time



(see bigger in next slide)

+ IBD ARCH3



Summary

- In this chapter you learnt
 - the scalability concept
 - the architectural causes and effects of scalability
 - tactics to improve scalability
 - a comparison technique to evaluate the scalability in alternative architectures
- You learnt how to
 - express scalability in software architecture using a cause-effect relationship
 - compare two alternative architectures to evaluate their scalability by analysing the ripple effect of the response time quality attribute

For Further Reading

- Bass, L., Clements, P., Kazman, R.: *Software Architectures in Practice*, 2nd edn. Addison Wesley, Reading (2003)
- Clements, P., Bachmann, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R.: *Documenting Software Architecture: Views and Beyond*. SEI Series in Software Engineering (2003)