

# Software Architecture in Action

Flavio Oquendo, Jair C Leite, Thais Batista



# Chapter 11

## Designing Fault Tolerance in Software Architectures

# Learning outcomes of this chapter

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

# The structure of this chapter

- Introduction
- Fault Tolerance Causes and Effects
- Fault Tolerance Quality Attributes
- Fault Tolerance Tactics
- Applying Fault Tolerance Tactics
- Fault Tolerance Analysis
- Summary



# Introduction

# Fault tolerance

## Conceptual overview

- Fault tolerance is a quality to refer to the degree to which a product or system can maintain operational service even in the presence of faults
- An example of fault tolerance in the case of a temperature monitor system is to maintain its operational service when the controller fails

# Fault tolerance

## Expressing fault tolerance using software architecture concepts

- Fault tolerance as an architectural quality refers to the degree to which an architecture can maintain its operation even if components fails



# Fault Tolerance Causes and Effects



# Fault Tolerance Causes and Effects

## Expressing fault tolerance causes and effects

- We need to express the causes and the effects
  - Causes refer to identify the components that can fail
  - Effects refer to impact in the system implied by its architecture when one or more components fail

# Fault causes

## Concepts

- Components can fail due to several causes:
  - stop the operation (loss of service)
    - hardware crash or stop
    - software fail
  - Incorrect operation
    - hardware or software fail
- In the RTC system, we can identify components that can fail
  - a temperature sensor component
  - a presence sensor component
  - a room temperature controller component

# Fault effects

## Concepts – 1/2

- Effects are the consequences of the causes in the architecture
  - the impact on system operation
- We need to analyze fault at both component-level and architecture-level
  - at the component-level an internal fail can cause an error
    - If the error is recovered there is no failure
    - If it is not recovered and it is propagated out of the component causing a behavior that is not in conformance to the specification, it is a failure
  - Out of component means that a component stops to operate or it continues to operate but incorrectly

# Fault effects

## Concepts – 2/2

- Continued...
  - at the architecture-level a component failure can cause an error
    - If the error is recovered there is no failure
    - If it is not recovered and it is propagated out of the system causing a behavior that is not in conformance to the specification, it is a failure
    - Out of system means that a it stops to operate or it continues to operate but incorrectly

# Fault effects

## Example – 1/2

- In the RTC System ARCH3, the causes are
  - a temperature sensor fails
  - a presence sensor fails
  - the controller fails
- The effects in RTC System ARCH3 for those causes are
  - a temperature sensor fails
    - the system continues to operate but incorrectly
      - the monitor blocks waiting for data from the failed sensor
      - the controller does not change the room temperature anymore

# Fault effects

## Example – 2/2

- Continued...
  - a presence sensor fails
    - the system continues to operate but incorrectly
      - the system maintain the room temperature in 22 C even in the presence of a user
  - the controller fails
    - the system stops to operate



# Fault tolerance quality attributes

# Fault tolerance quality attributes

## Concept

- Fault tolerance quality attribute refers to a quality used to quantify how much the system tolerates fails
- The ripple effect in system operation
  - no impact in system services
    - the services continue to be provided in nominal mode
  - some impact in system services
    - the services continue to be provided but in a degraded mode
  - total impact in system services
    - the services stop to be provided

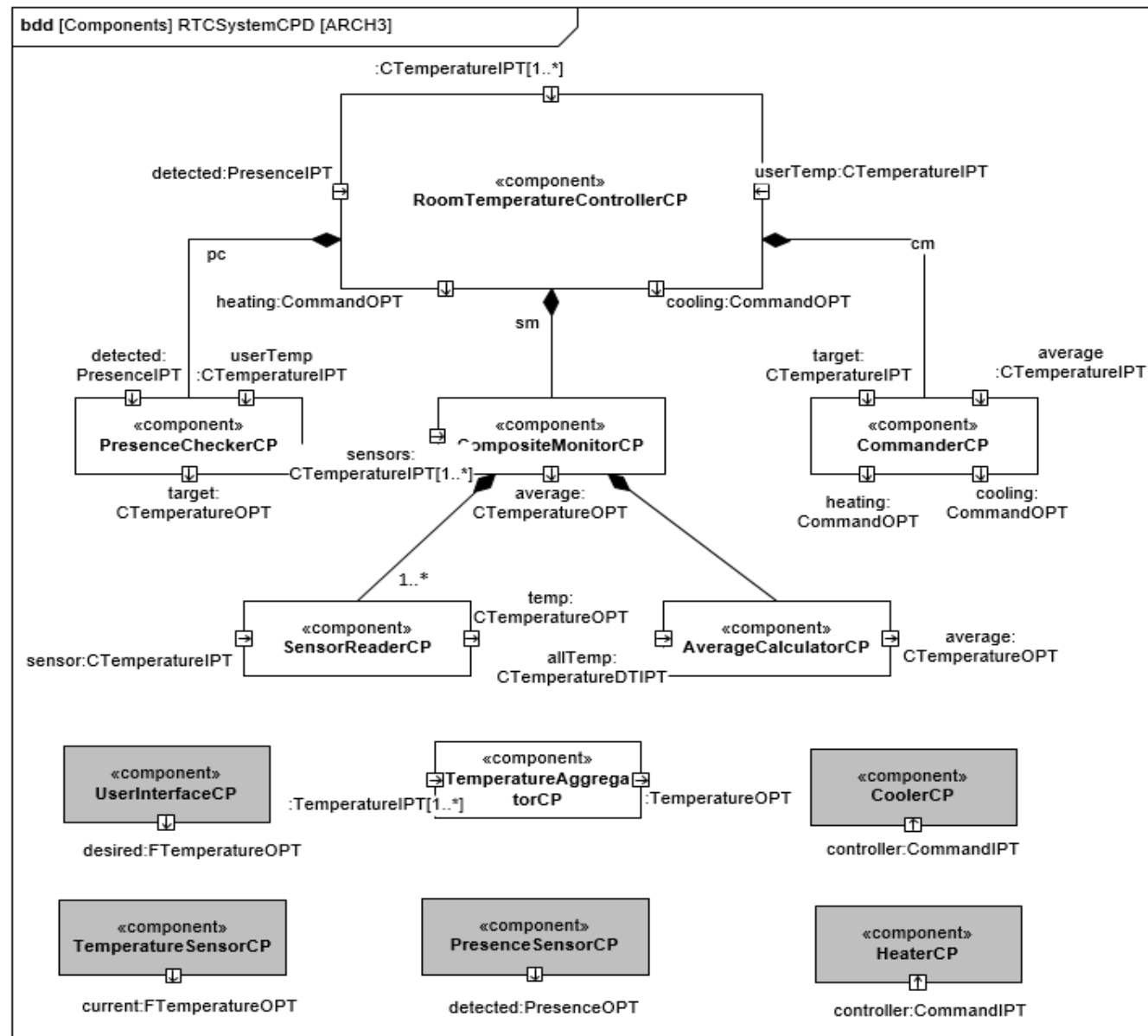


# Fault tolerance quality attributes

## Example

- As an example, in RTC System ARCH3 (see next slide)
  - if a temperature sensor fails
    - there is a total impact
      - the system continues to operate but it is blocked and the services are not provided anymore
  - if a presence sensor fails
    - there is total impact
      - the system continues to operate but it is blocked and the services are not provided anymore
  - if the controller fails
    - there is a total impact
      - the system stops to operate

# + RTC System – ARCH3





# Fault Tolerance Tactics

# Fault Tolerance Tactics

## Heartbeat

### ■ heartbeat

- a component emits a message periodically to notify it is operating
- as an example, in the RTC System the controller periodically sends a status to the connector informing that its status is up

# Fault Tolerance Tactics

## Introduce Redundancy

- introduce redundancy
  - to improve fault tolerance we can introduce component redundancy in order to keep system services
  - in the RTC System, we can introduce redundancy in the room temperature controller component to keep the system services in operation



# Applying Fault Tolerance Tactics

# Applying Fault Tolerance Tactics

## Designing a new architecture – ARCH4

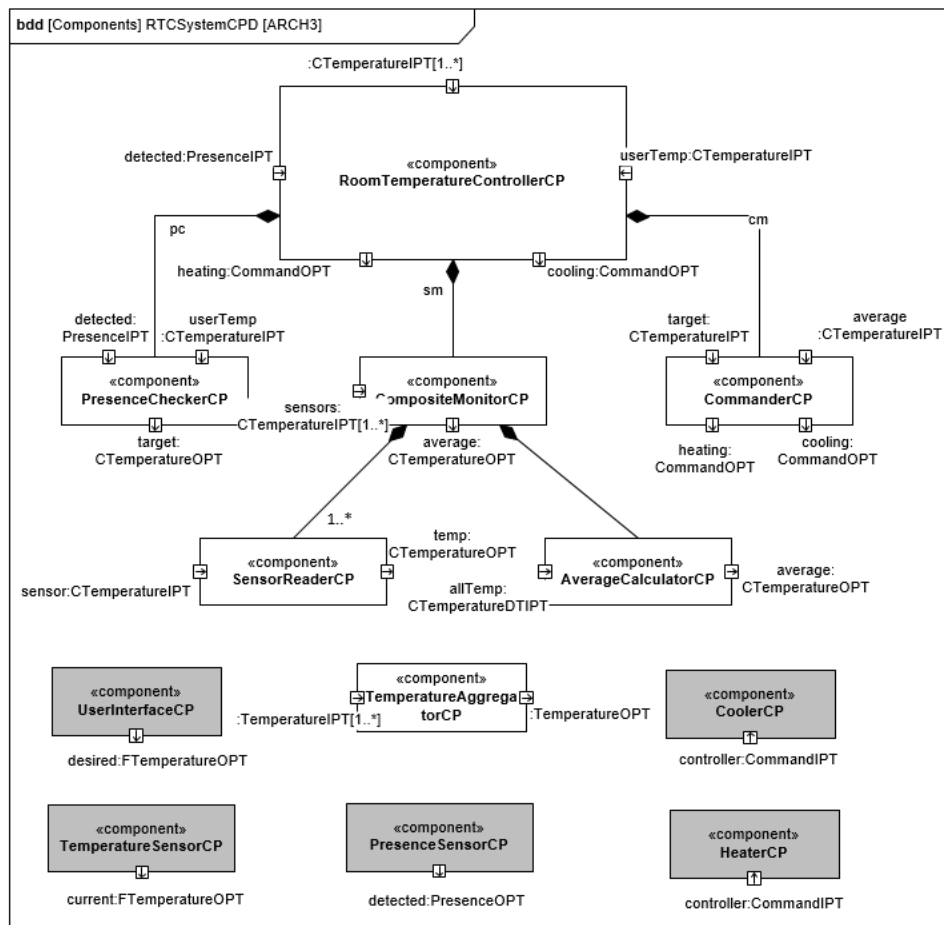
- To detect and recover a controller fail we design a new architecture of the RTC System
- In the new architecture ARCH4, we apply the *heartbeat* and the *introduce redundancy* tactics
  - there are two instances of the RoomTemperatureController, one is consider primary and the other is the secondary
  - a composite connector receives the controllers status and use that information to decide to which one it sends the temperature values
  - the RoomTemperatureControler has a composite port to send a message to the connectors and to receive the temperature value
- In the next slides we present the differences between the two architectures

# Applying the Fault Tolerance Tactic

24

## The Component BDDs of ARCH3 and ARCH4

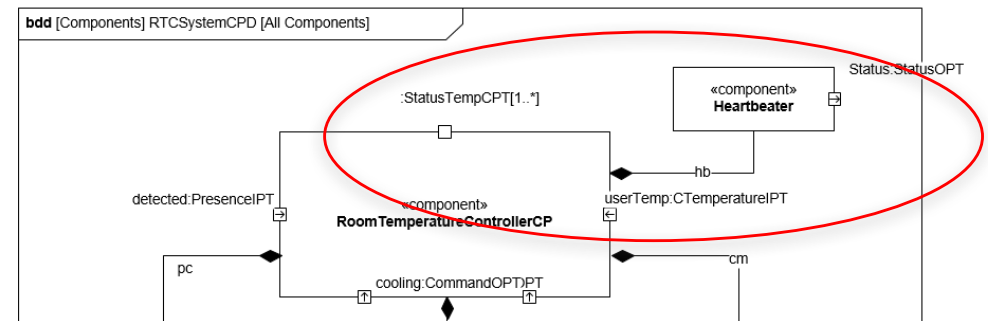
### ARCH3



### ARCH4

- ARCH4 introduces a composite port **StatusTempCP** and a **HeartbeaterCP**

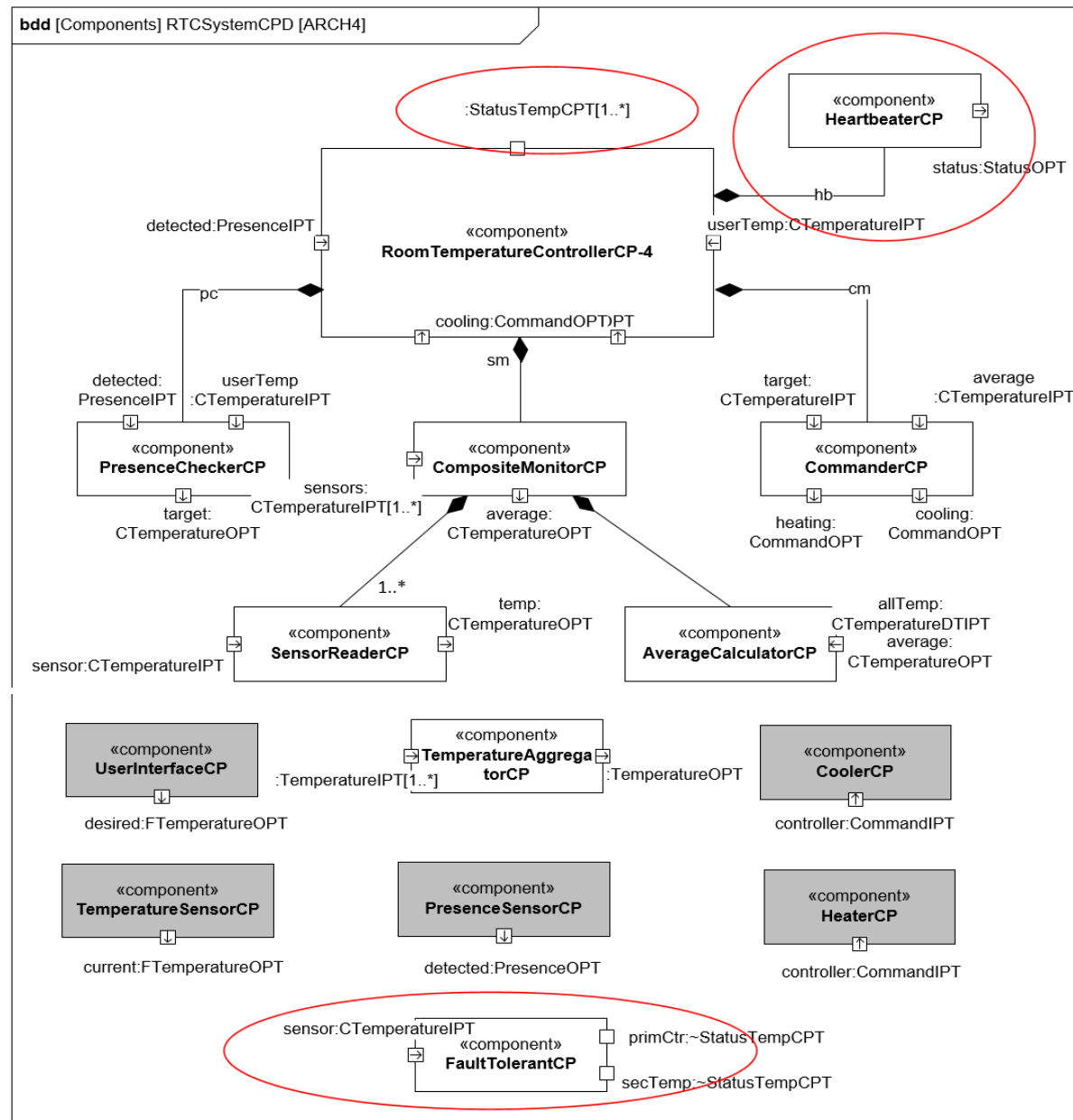
■ (complete in next slide)







# RTC System – ARCH4 BDD



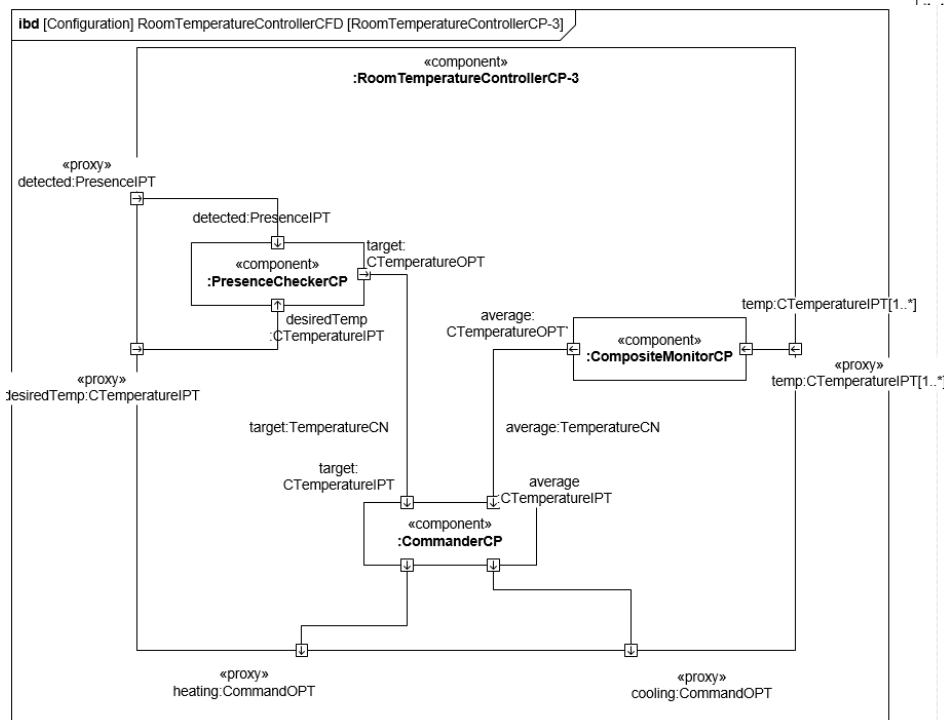
# Applying the Fault Tolerance Tactic

## The configuration of RoomTemperatureController

26

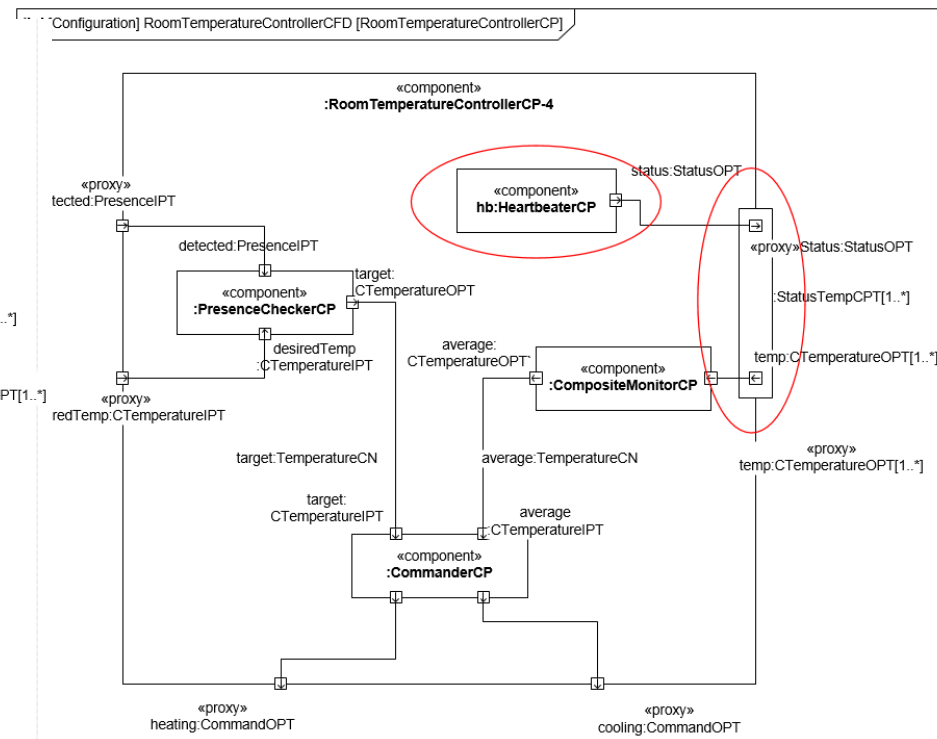
### ARCH3

- ARCH3 has no component to inform the system status



### ARCH4

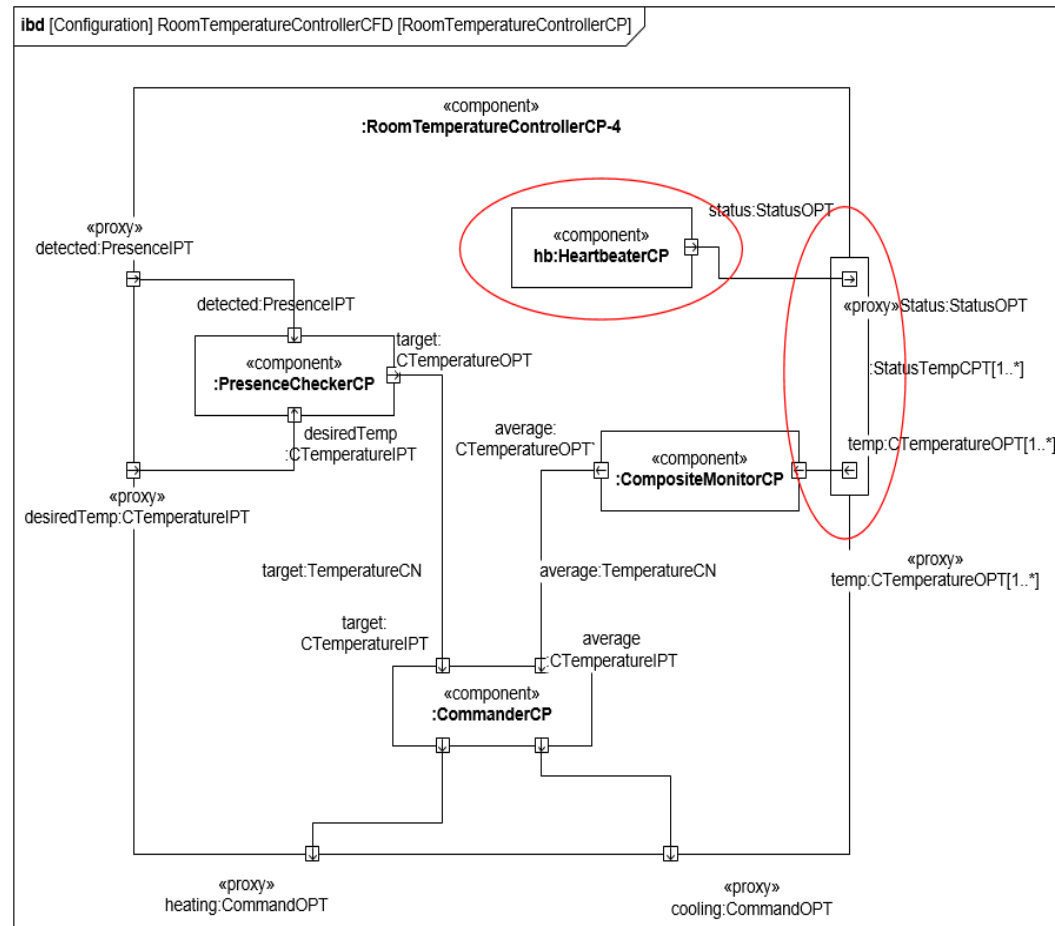
- In ARCH4 we introduce a HeartbeaterCP component and the StatusTempCPT composite port



## Heartbeater and StatusTempCPT

- The RoomTemperatureControllerCP has a HeartbeaterCP component that sends a status message informing it is up or down
- The StatusTempCPT composite port has an out port to send the status and an in port to receive the temperature

## SysADL Notation



# Applying the Fault Tolerance Tactic

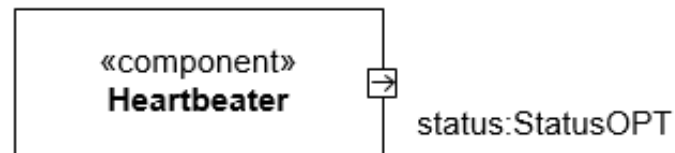
28

## The HeartbeaterCP component

### The HeartbeaterCP component

- The HeartbeaterCP component emits a message periodically to notify it is operating
  - it periodically sends a status message to inform that it is up

### Representation in SysADL

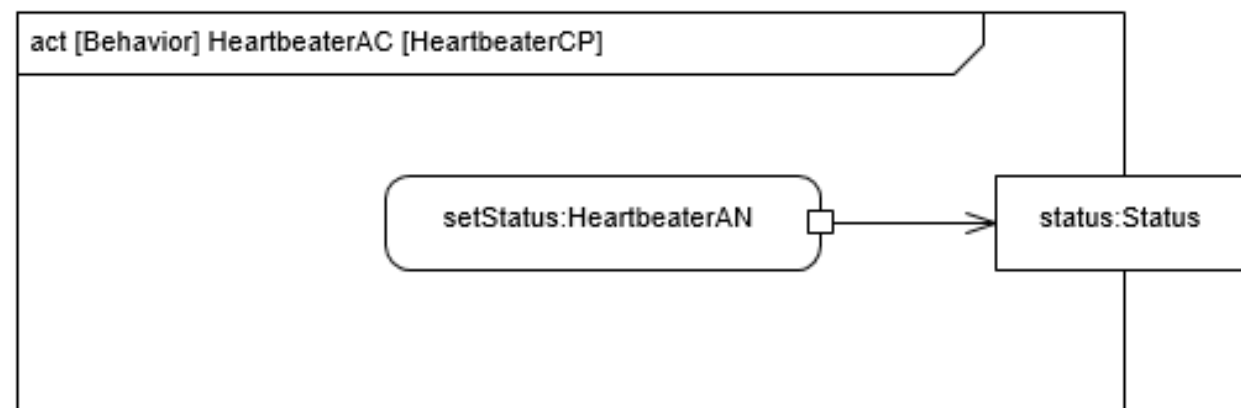


# Applying the Fault Tolerance Tactic

29

## The HeartbeaterCP component behavior

- The HeartbeaterAC activity has an action to sends a status message through its port



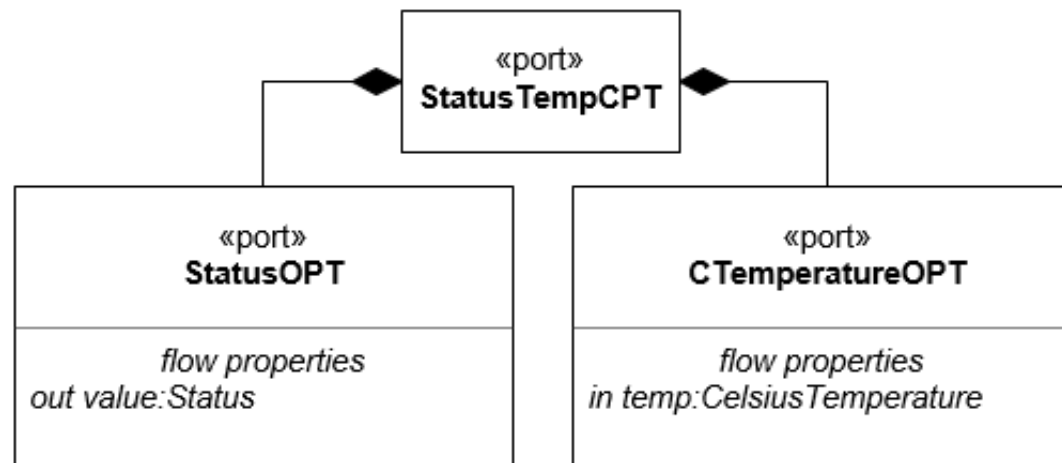
# Applying the Fault Tolerance Tactic

## The StatusTempCPT port in ARCH4

### ARCH3

- The StatusTempCPT port is composed of two ports
  - CTemperatureOPT is an input port to receive the temperature from the sensors
  - StatusOPT is an output port to send the status to the external connection

### ARCH4

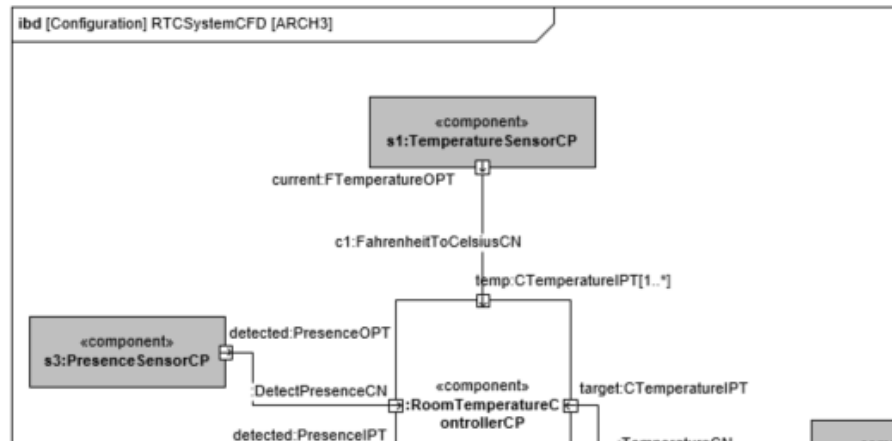


# Applying the Fault Tolerance Tactic

## Two different configurations of ARCH3 and ARCH4

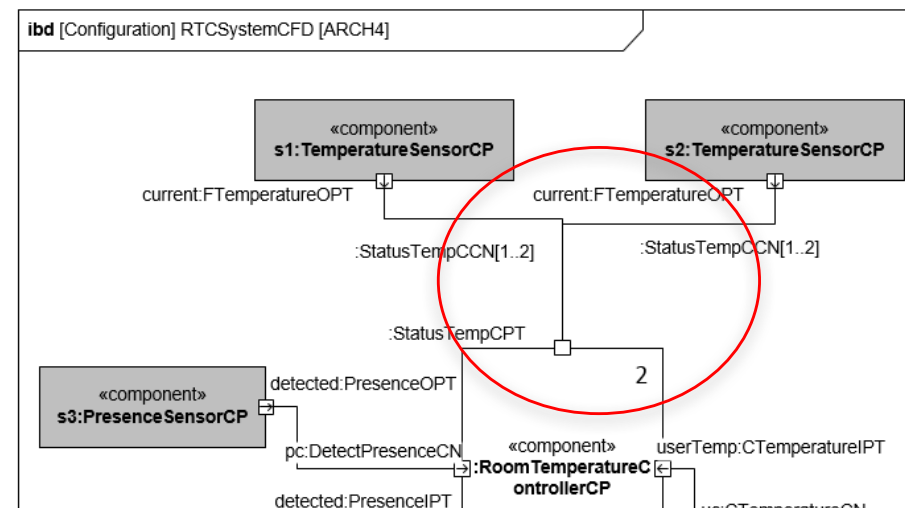
### ARCH3

- In ARCH3, each sensor connects to the RoomTemperatureControllerCP using the a simple connector that convert a temperature from Fahrenheit to Celsius



### ARCH4

- In ARCH4, we can have up to two instances of the RoomTemperatureControllerCP
- Each sensor is connected using a composite connector – StatusTempCCN – to additionally allow the sensor to receive the status of the HeaterCP

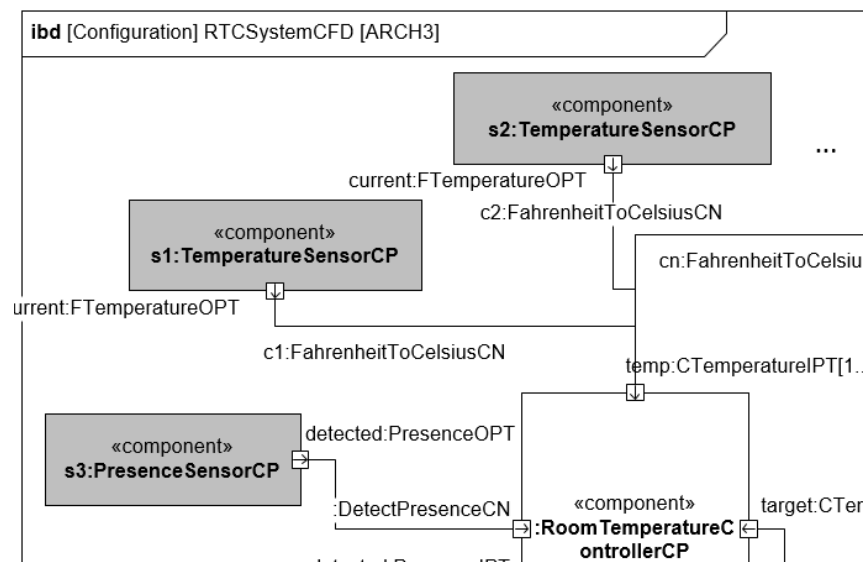


# Applying the Fault Tolerance Tactic

## Two different configurations of ARCH3 and ARCH4

### ARCH3

- In ARCH3, each sensor connects to the RoomTemperatureControllerCP using the a simple connector that convert a temperature from Fahrenheit to Celsius



### ARCH4

- In ARCH4, we can have up to two instances of the RoomTemperatureControllerCP
- Each sensor is connected using a composite connector – StatusTempCCN – to additionally allow the sensor to receive the status of the HeartbeaterCP
- (see next slide)





# Applying the Fault Tolerance Tactic

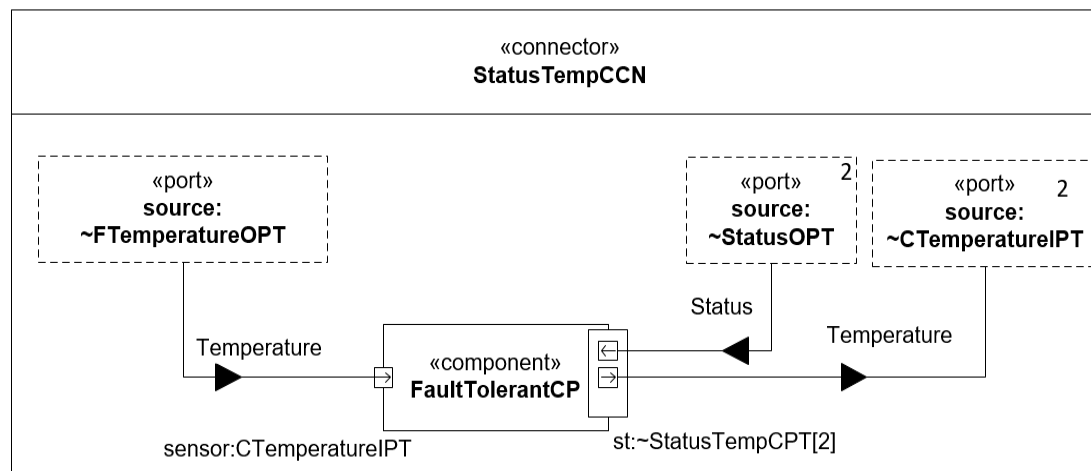
## The StatusTempCCN connector definition in ARCH4

### ARCH3

- ARCH3 has a FahrenheitToCelsiusCN connector between a TemperatureSensorCP and the RoomTemperatureControllerCP component

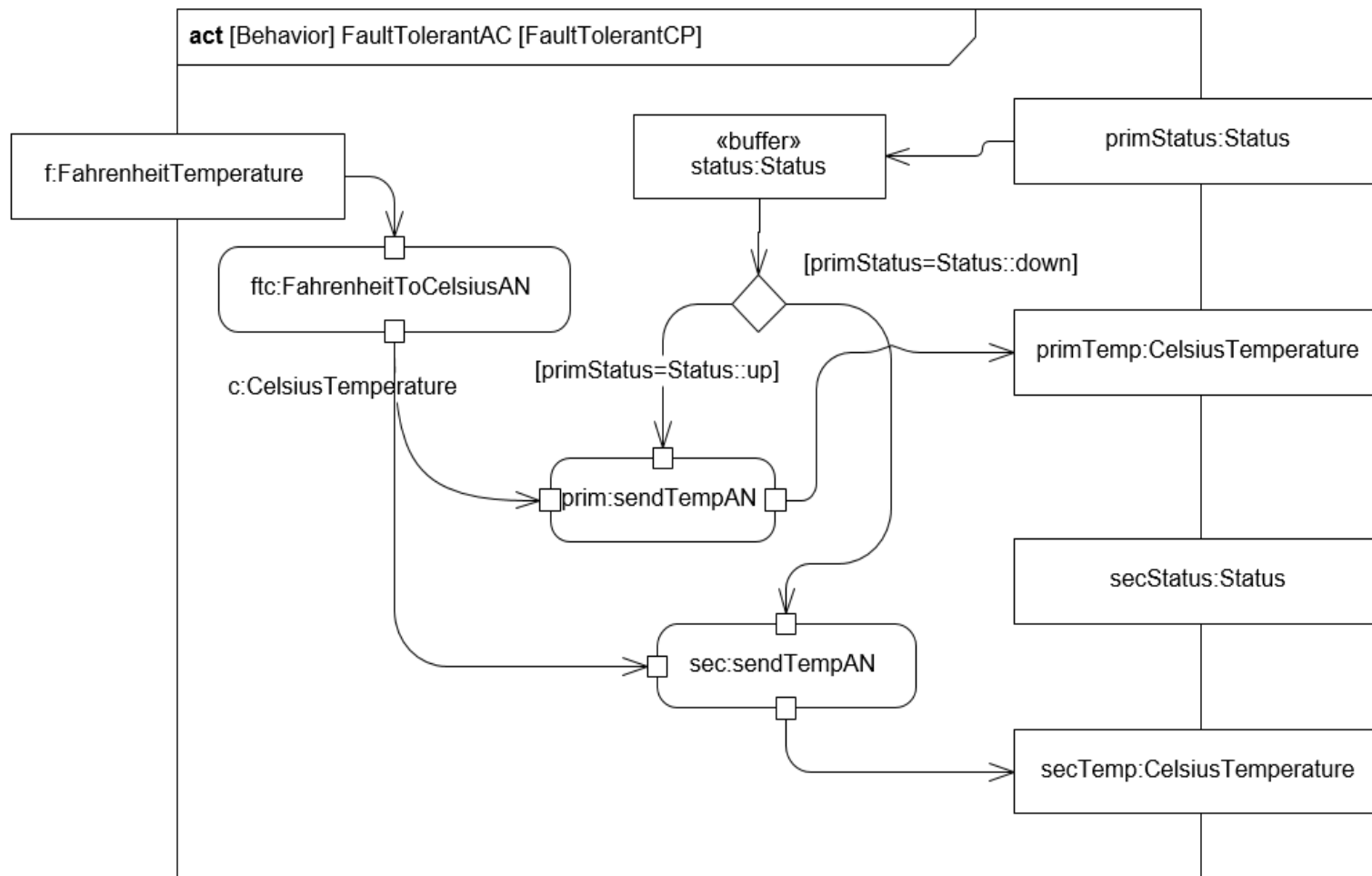
### ARCH4

- ARCH4 has a more complex connector that includes a FaultTolerantCP component
- it receives a signal from the RoomTemperatureController component that informs its status (up or down)



# Applying the Fault Tolerance Tactic

## The FaultTolerantAC behavior in ARCH4





# Fault tolerance Analysis

# Fault tolerance example

## RTC System – Fault tolerance requirements

- We use an AND/OR tree to depict the requirements (R), causes (C) and effects (E)
- RTC Fault tolerance requirements
  - R1 – The system must tolerate the failure of the controller

# Fault tolerance example

## RTC System – causes

- RTC Fault tolerance requirements
  - R1 – The system must tolerate the failure of the controller
    - C1 – The controller failure

# Fault tolerance example

## Analysing the ripple effect – 1/2

- We analyse the fault tolerance on two different architectures of the RTC system considering the same cause
  - the tolerance to controller failure (Requirement R1)  
studying the ripple effect of this failure (Cause C1)
  - first, we analyse the ARCH3 architecture
    - in this architecture the failure of the controller causes the failure of the whole system

# Fault tolerance example

## Analysing the ripple effect – 2/2

- then, we analyse the ARCH4 architecture
  - in this architecture the failure of the controller does not cause the failure of the whole system
    - the architecture has two instances of a controller. When the first fail, the system continues to operate using the second instance
- According to the aforementioned analysis, considering the failure, we can conclude that ARCH3 is not fault tolerant and ARCH4 is fault tolerant



# Summary

- In this chapter you learnt
  - the fault tolerance concept
  - the architectural causes and effects of fault tolerance
  - tactics to improve fault tolerance
  - a comparison technique to evaluate the fault tolerance in alternative architectures
- You learnt how to
  - express fault tolerance in software architecture using a cause-effect relationship
  - compare two alternative architectures to evaluate their fault tolerance by analysing the ripple effect (impact) in system operation

# For Further Reading

- Bass, L.; Clements, P.; Kazman, R. Software Architectures in Practice, 2nd ed. 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)