

# Architecture Normalization for Component-Based Systems

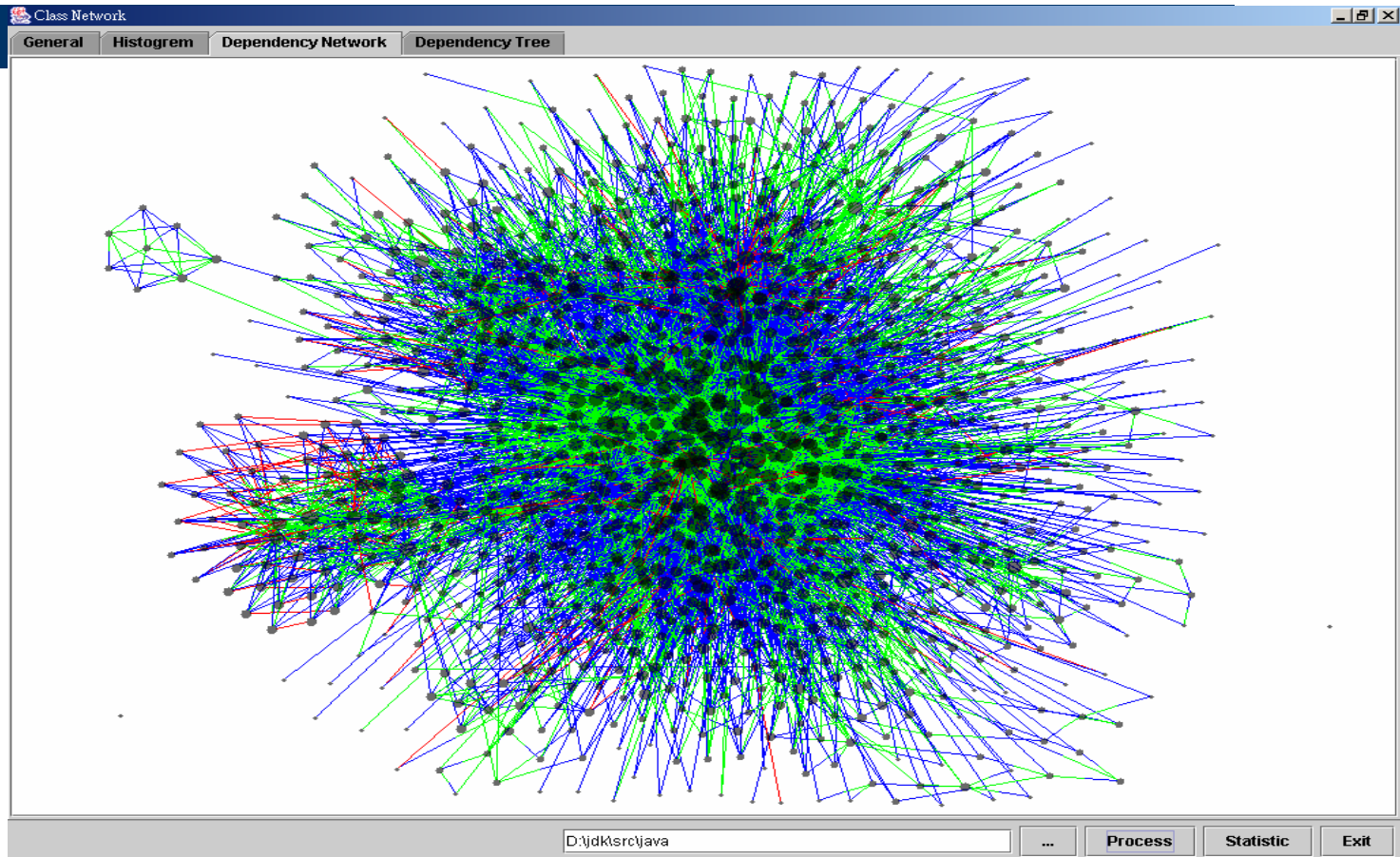
Lian Wen & Geoff Dromey  
Software Quality Institute



# A software system

- A set of components
- Plus the integration relationships
- Called a component integration network (CIN)
- The topological structure of a CIN is called component architecture

# Component Architecture can be very complex



The Component Dependency Network (CDN) of package “java” with 1172 “components”

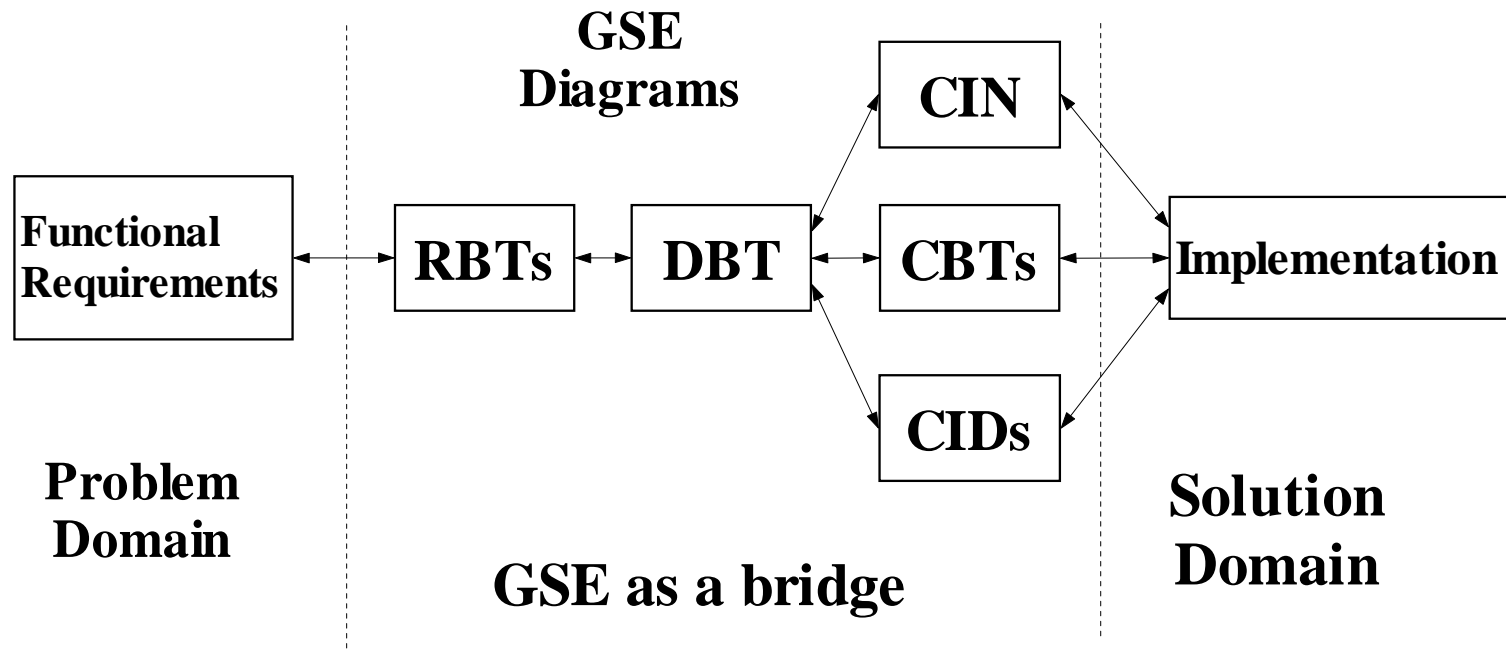


**Is Component  
Architecture determined  
by functional  
requirements???**

# Overview

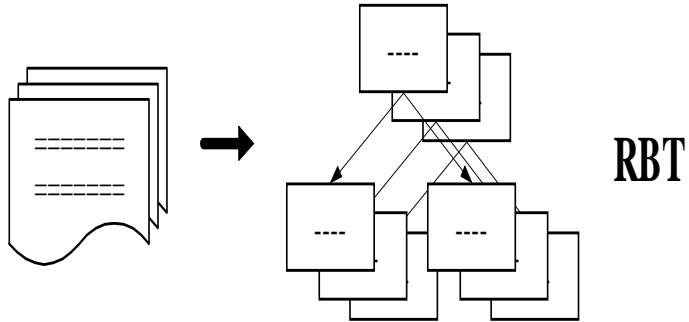
- Systematically change the original architecture of a component-based system to a desired target architecture.
- Keep the component architecture stable while the software system is evolving.
- Special advantages for software systems with tree-structured component architecture
- The work is based on the use of the Behavior Tree Representation and a proposed traceability model presented as SEFM'03 and SEFM'04

# GSE - Bridge Between the Two domains

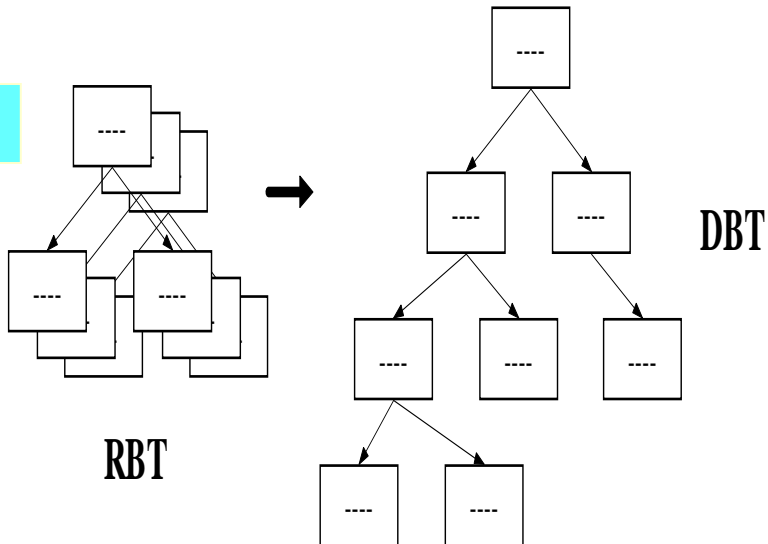


# GSE (Genetic Software Engineering)

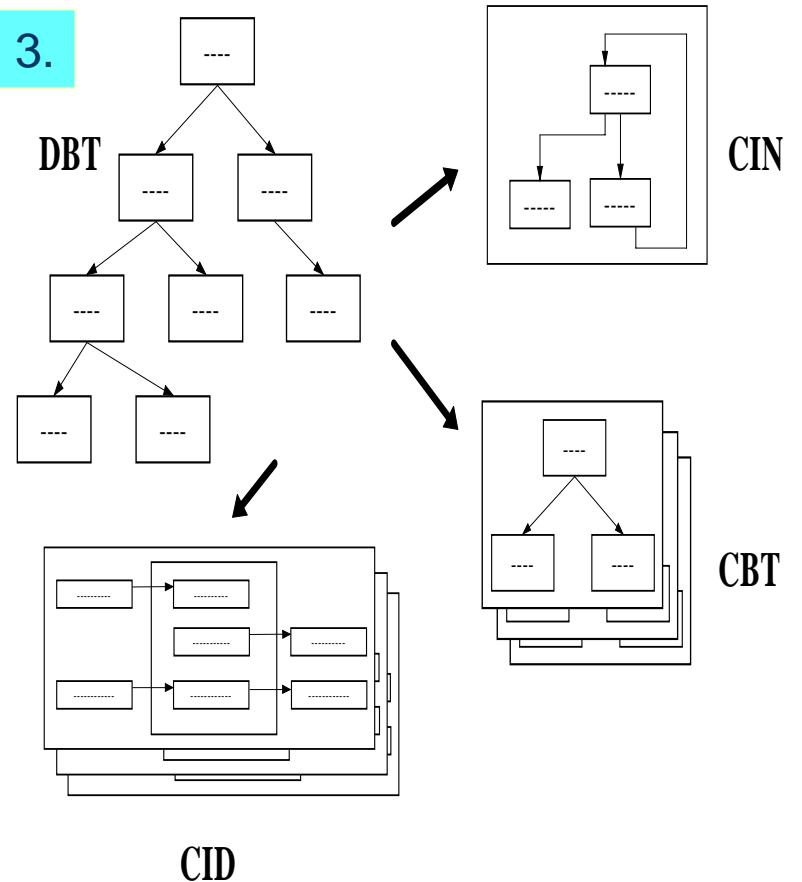
1.



2.



3.



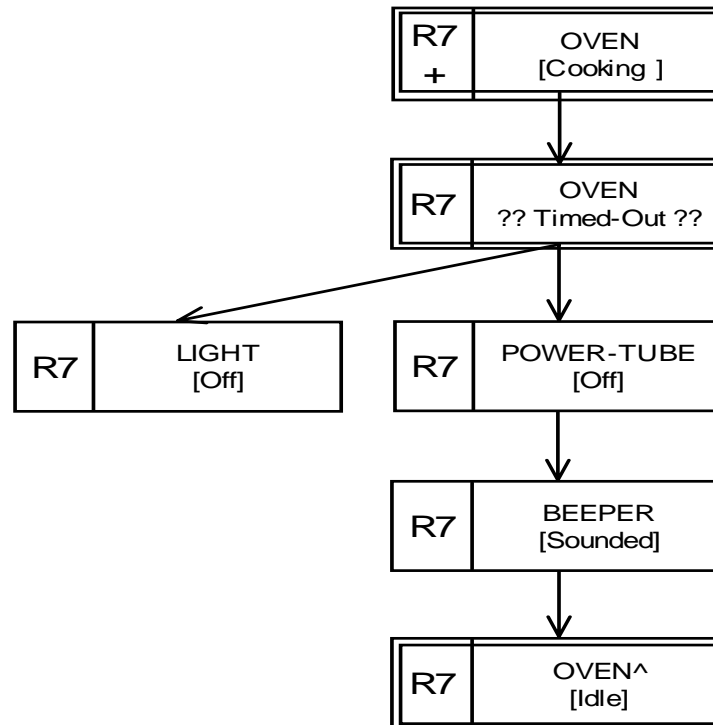
# Microwave Oven Study Case

- **R1.** There is a single control button available for the user of the oven. If the oven is idle with the door is closed and you push the button, the oven will start cooking (that is, energize the power-tube for one minute).
- **R2.** If the button is pushed while the oven is cooking it will cause the oven to cook for an extra minute.
- **R3.** Pushing the button when the door is open has no effect (because it is disabled).
- **R4.** Whenever the oven is cooking or the door is open the light in the oven will be on.
- **R5.** Opening the door stops the cooking.
- **R6.** Closing the door turns off the light. This is the normal idle state, prior to cooking when the user has placed food in the oven.
- **R7.** If the oven times-out the light and the power-tube are turned off and then a beeper emits a sound to indicate that the cooking is finished.

# Translate Requirements To Requirement Behavior Trees

## Requirement-7

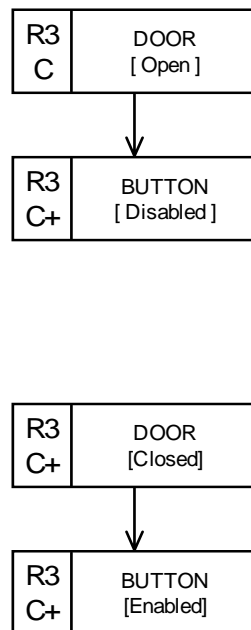
If the oven times-out the light and the power-tube are turned off and a beeper emits a sound to indicate that cooking has finished.



# More Examples

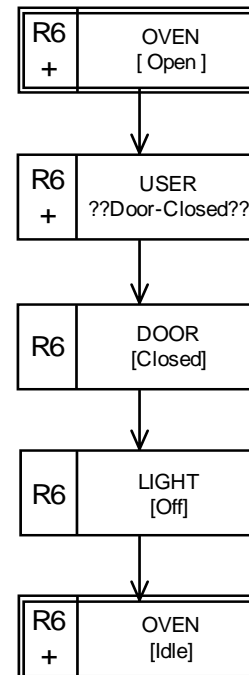
## Requirement-3

Pushing the button when the door is open has no effect (because the button is disabled)

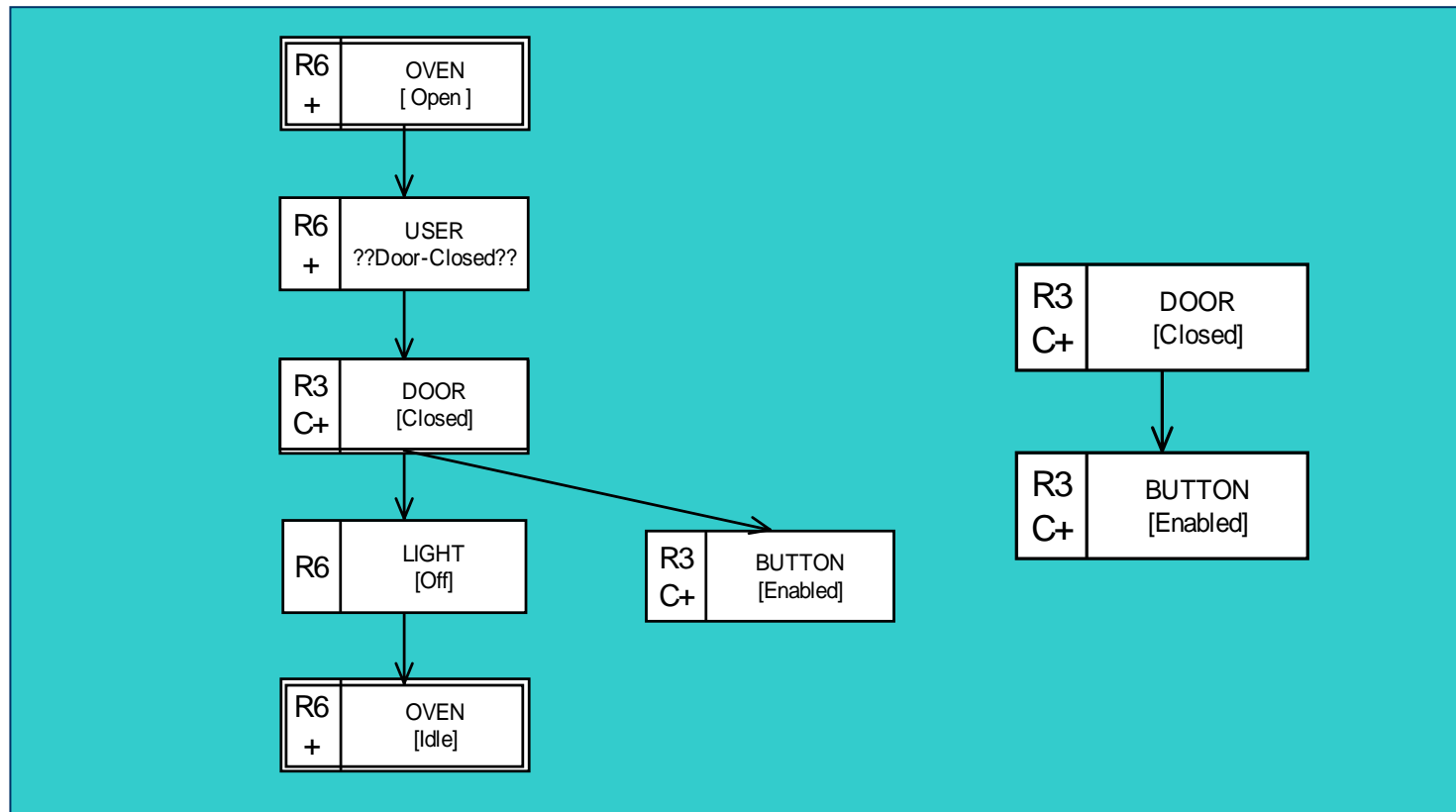


## Requirement-6

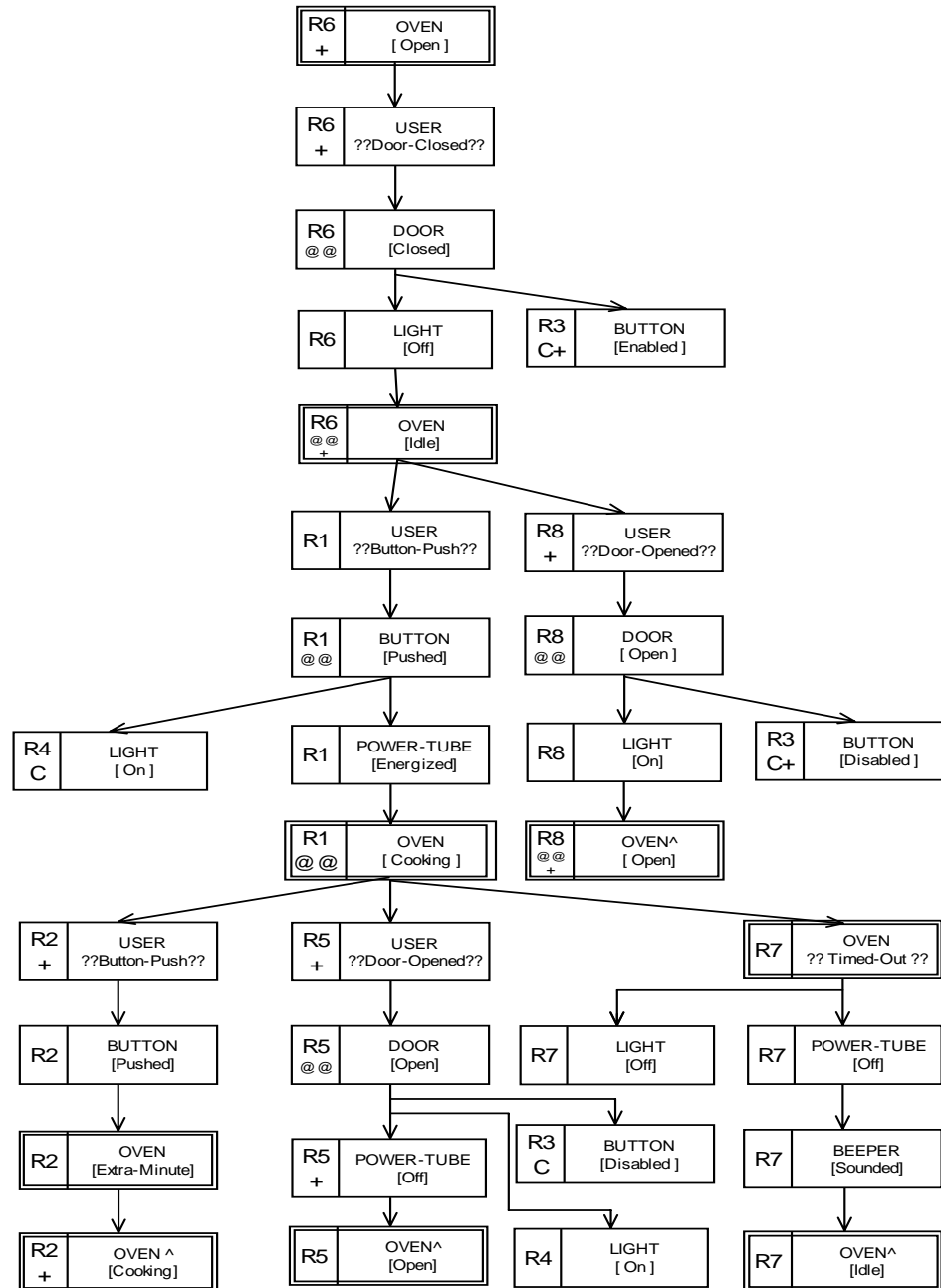
Closing the door turns off the light. This is the normal idle state prior to cooking when the user has placed the food in the oven.



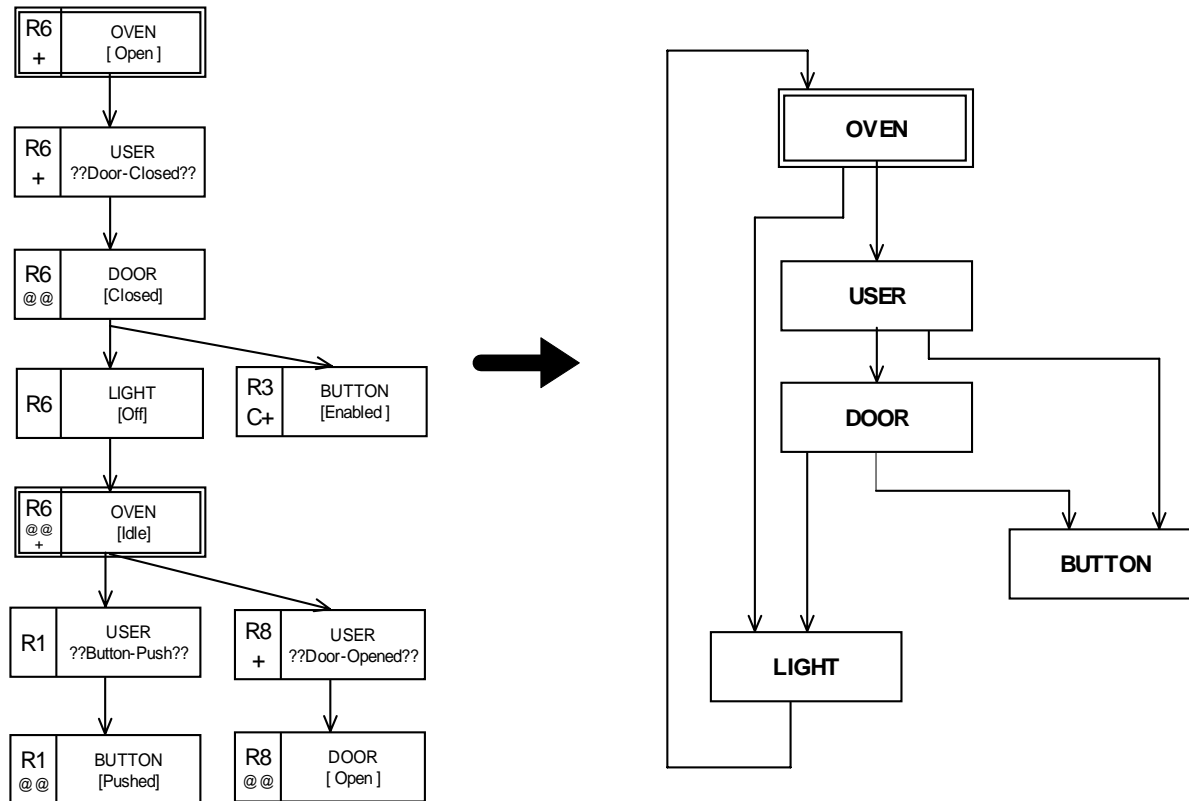
# Integrate Behavior Trees



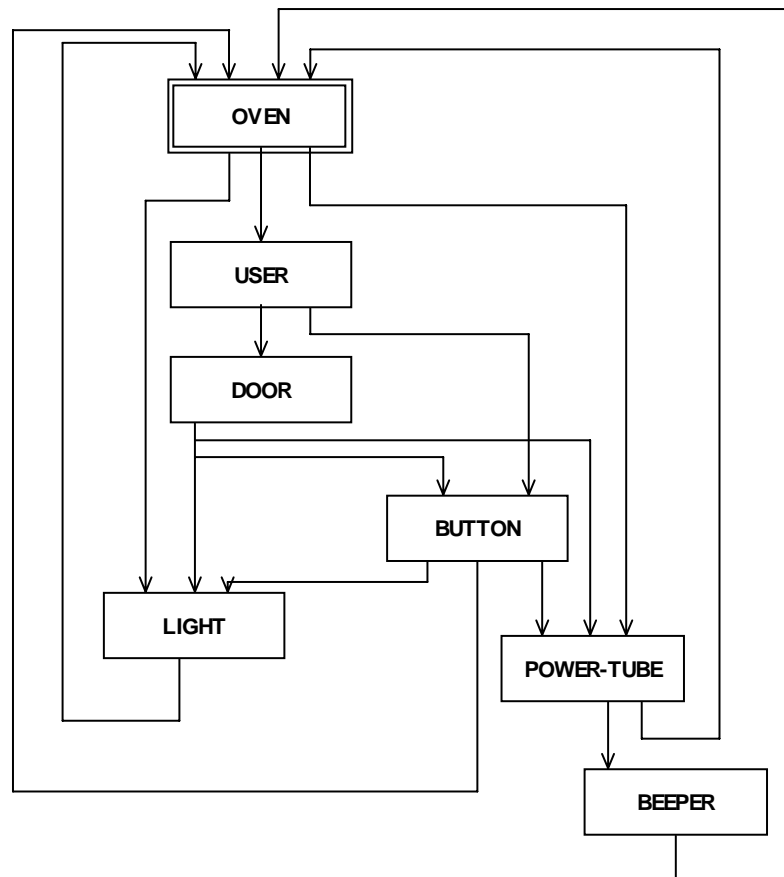
# DBT (Design Behavior Tree)



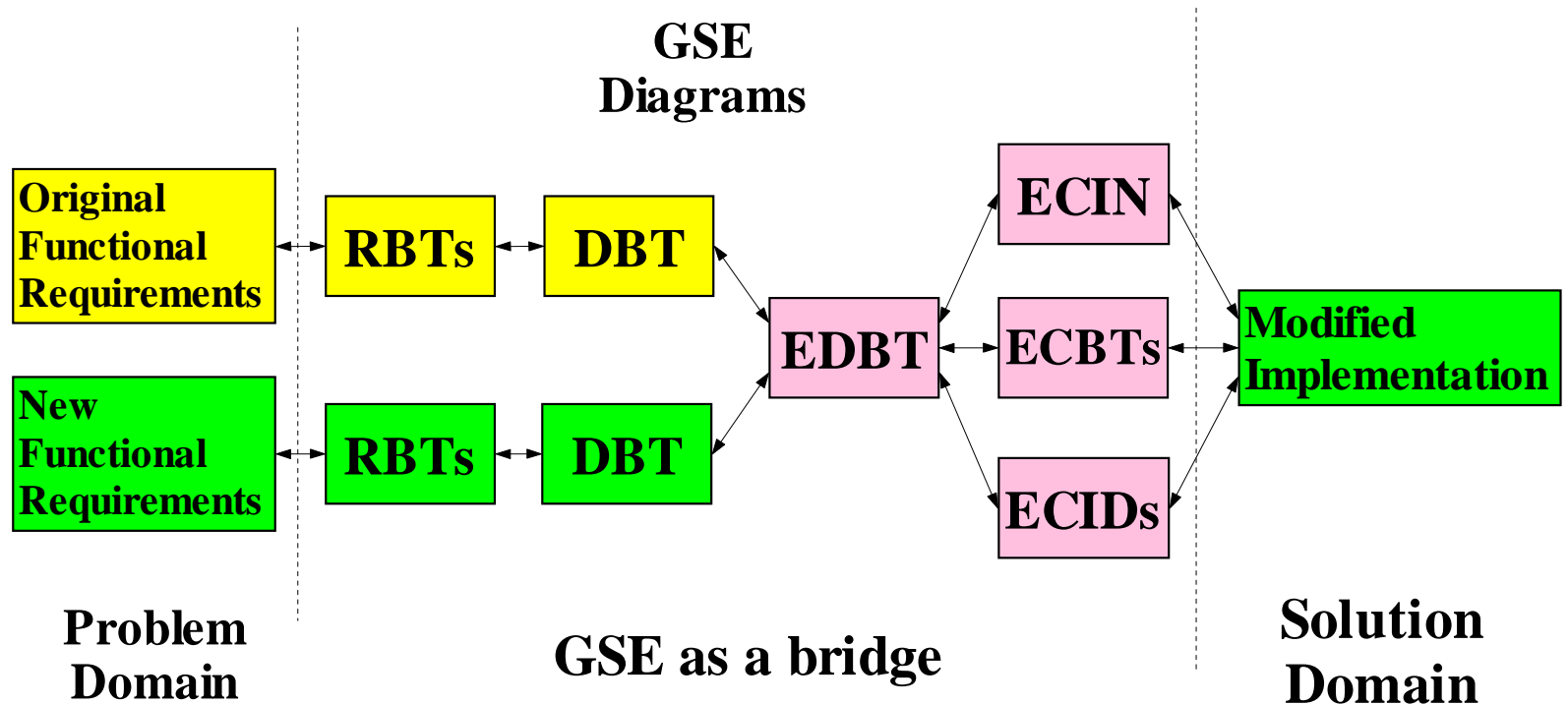
# From DBT to Component Interaction Network (CIN)



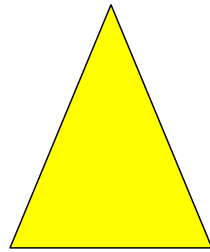
# The Completed CIN



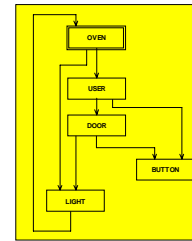
# Our Proposed Traceability Model



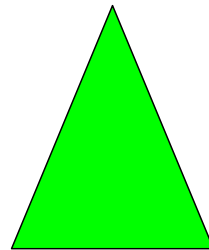
# Modify the DBT to get a desired CIN



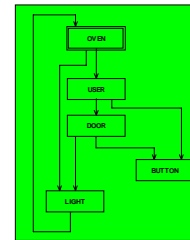
Original DBT



Associated CIN



Modified DBT

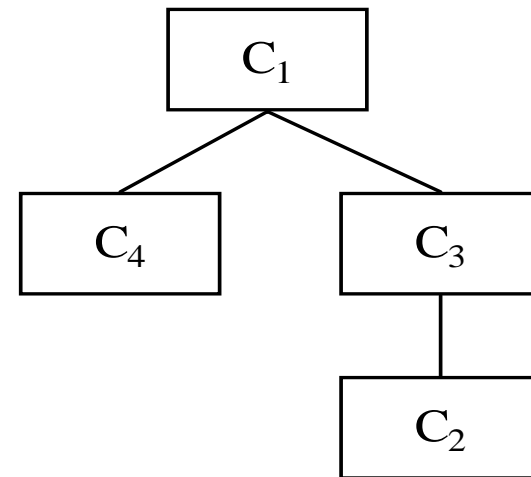
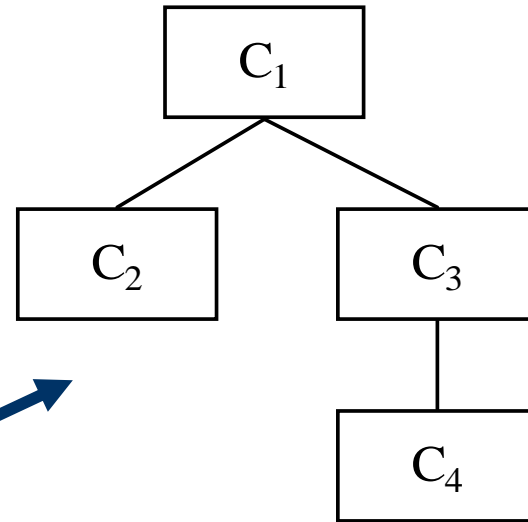
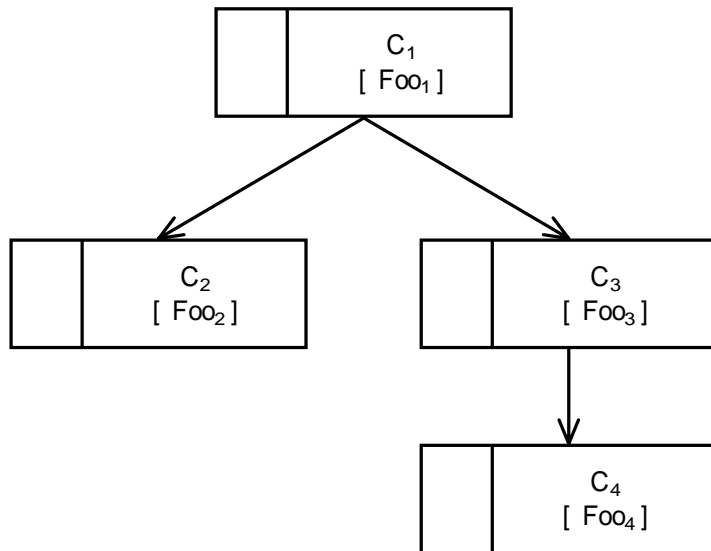


Desired CIN

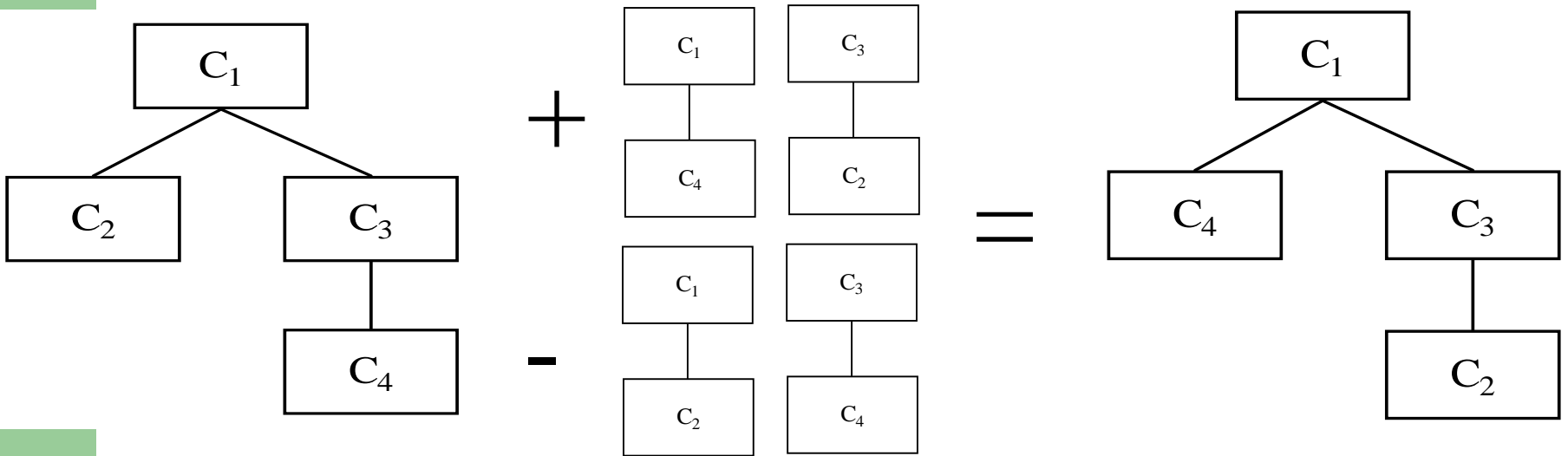
# How to manipulate the DBT

- Adjust the order of nodes
- Insert nodes
- An important property of a CIN, a CIN is a connected network
- Treat a CIN as an unidirectional graph

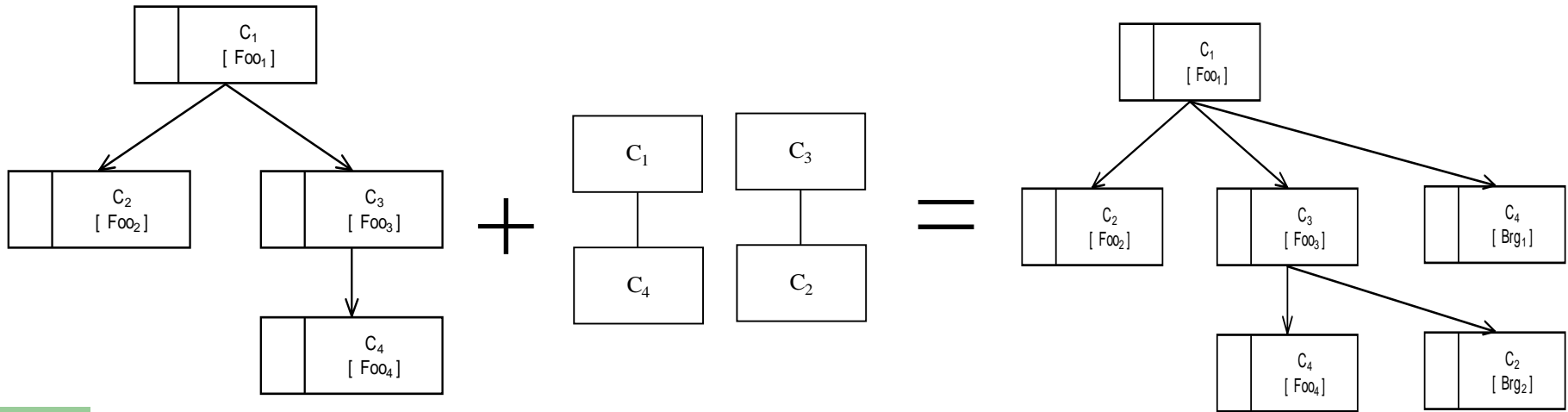
# A simple example



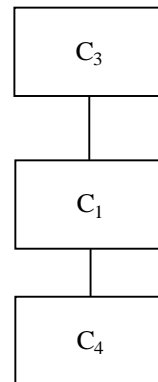
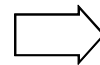
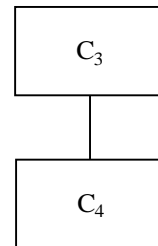
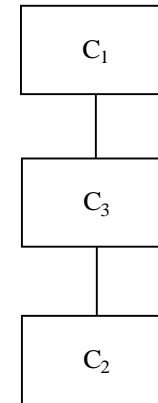
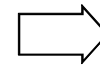
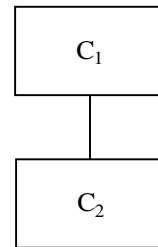
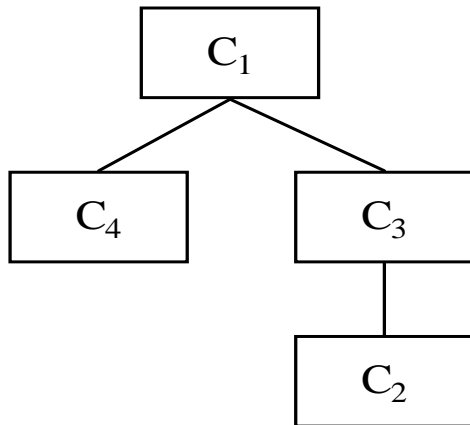
# Operations



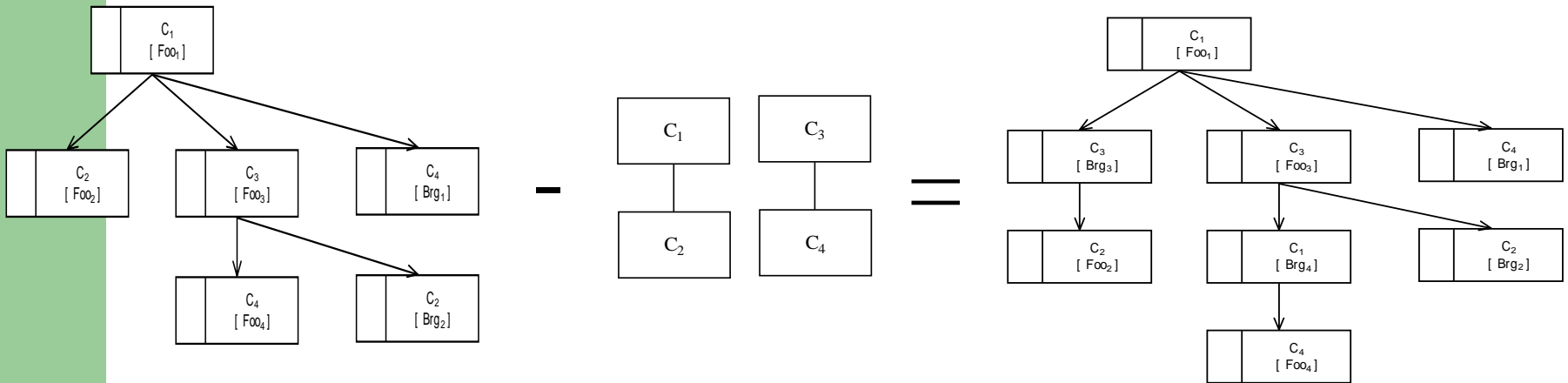
# Adding Missing Links



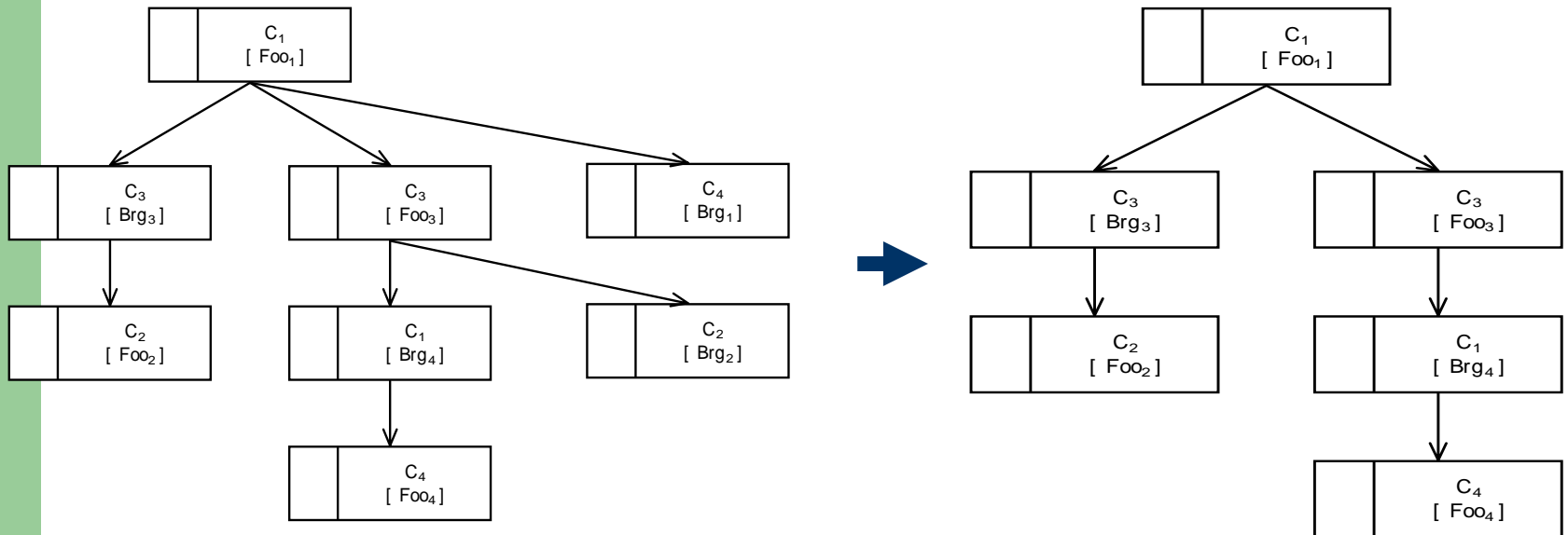
# Removing Extra Links 1



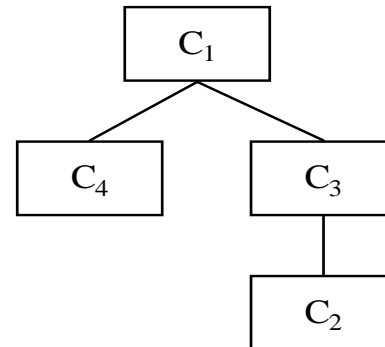
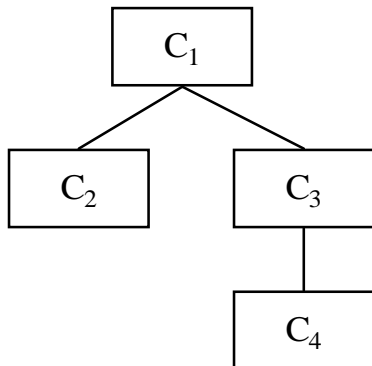
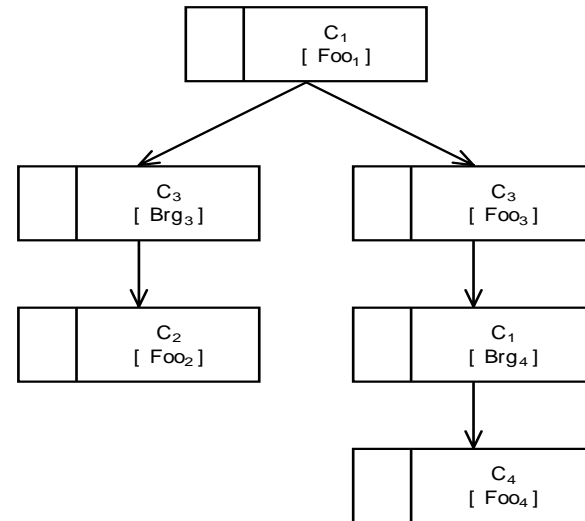
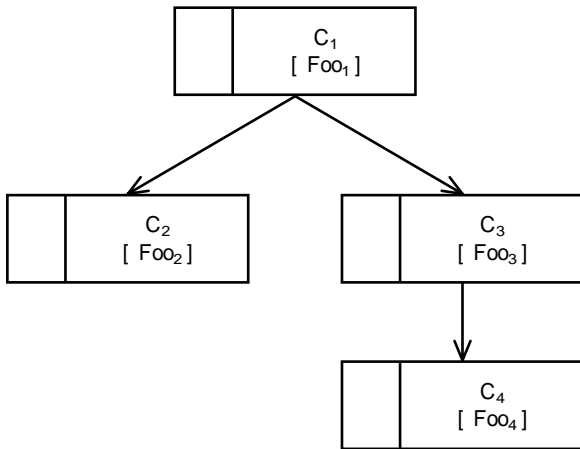
# Removing Extra Links 2



# Pruning the Unnecessary Nodes



# Comparing



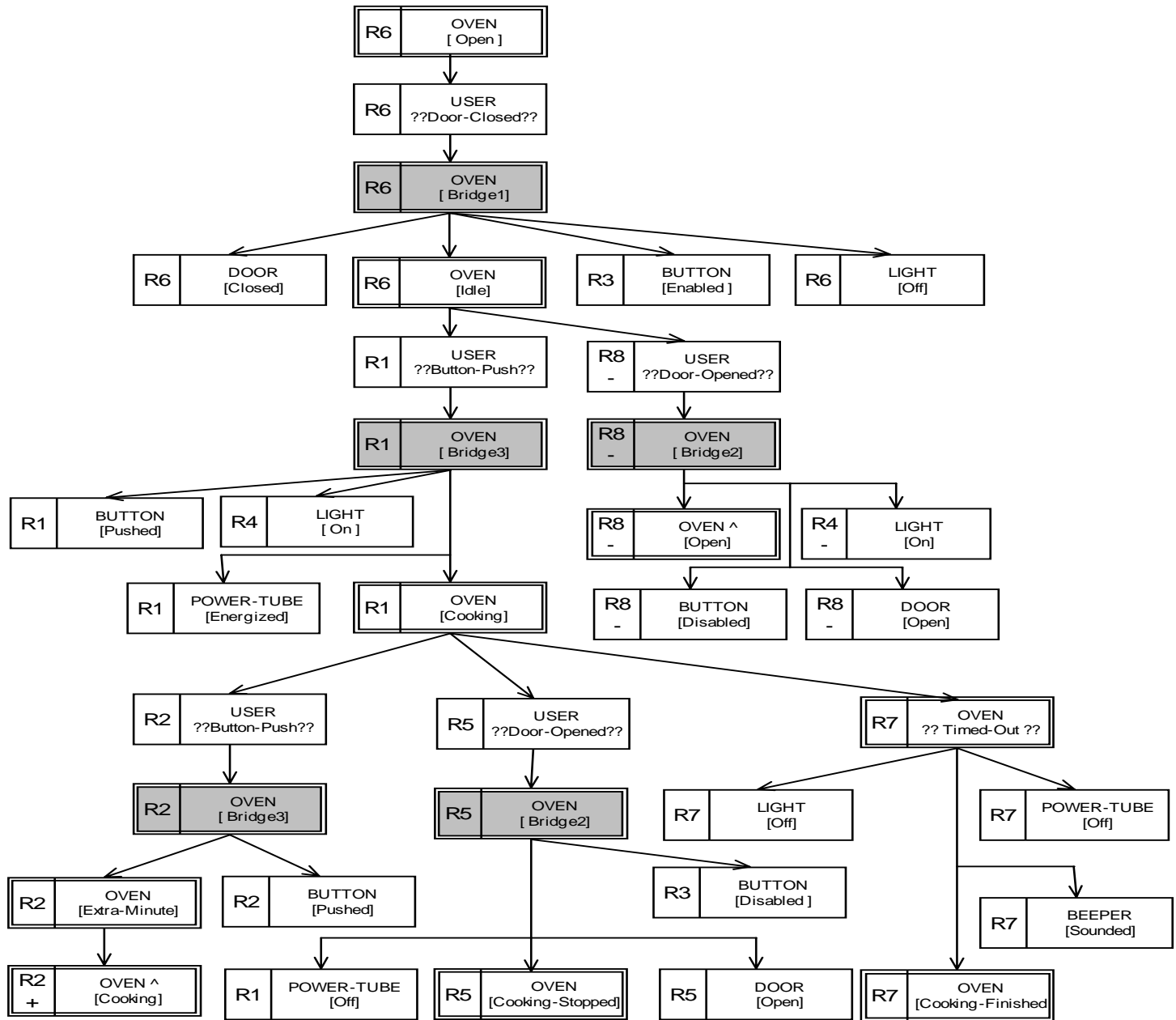
# Architecture Normalization

- ***Definition:*** A DBT is called a **normalized DBT** if the associated CIN is a tree. A software system with a normalized DBT is called **normalized software system**.

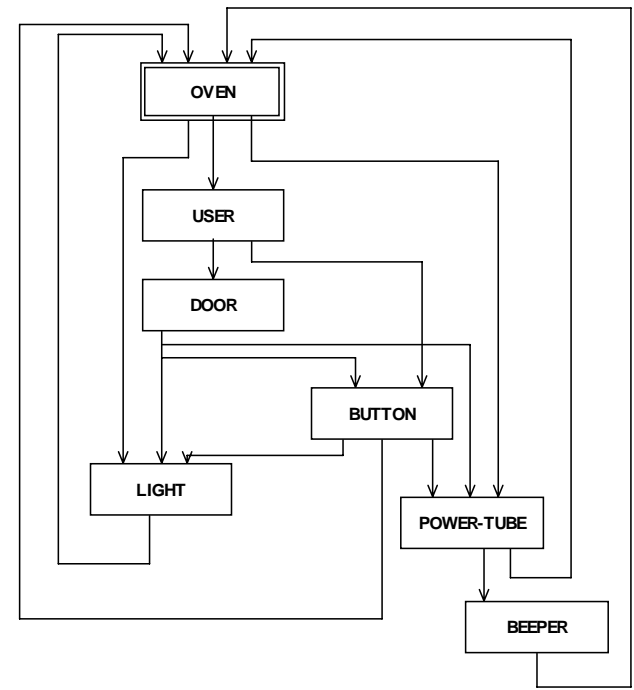
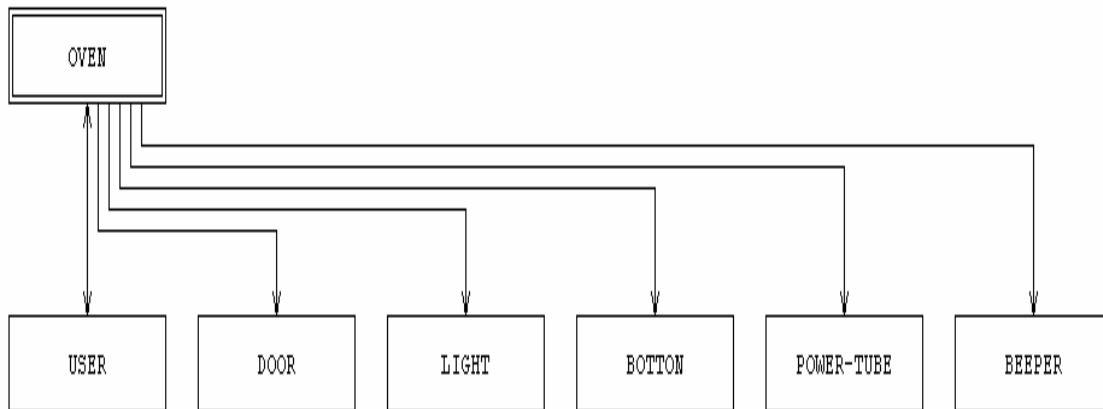
# Advantages of Tree-Structured Architecture

- **Simplicity**
  - Low coupling
  - All connections are local
  - Unique path
- **Scalability**
  - Self similarity
  - Integration and Decomposition
  - Social and natural structure

# Normalized DBT



# The Tree-Structured CIN



# Summary

- Change the software architecture to a desired target architecture.
- Keep the component architecture stable while the software system is evolving.
- Special advantages for software systems with tree-structured component architecture
  - Simplicity and Scalability

**The End**

*Thank You*