

# Statecharts and Model-Driven Development

Dror Feitelson

Basic Seminar on Software Engineering  
Hebrew University  
2009

# The Lavi: 1980–1988



Small plane with smart avionic systems, based on software

# 1982 – enlist David Harel as a consultant

What happens when you push this button?

Book B page 389 clause 19.11.6.10 says...

But what if the radar is locked on a target?

Book A page 895 clause 6.12.3.7 says...

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

We don't know...



In the end, despite thousands of pages of detailed specifications, the behavior of the aircraft in some complex set of circumstances is determined by a decision made by a programmer

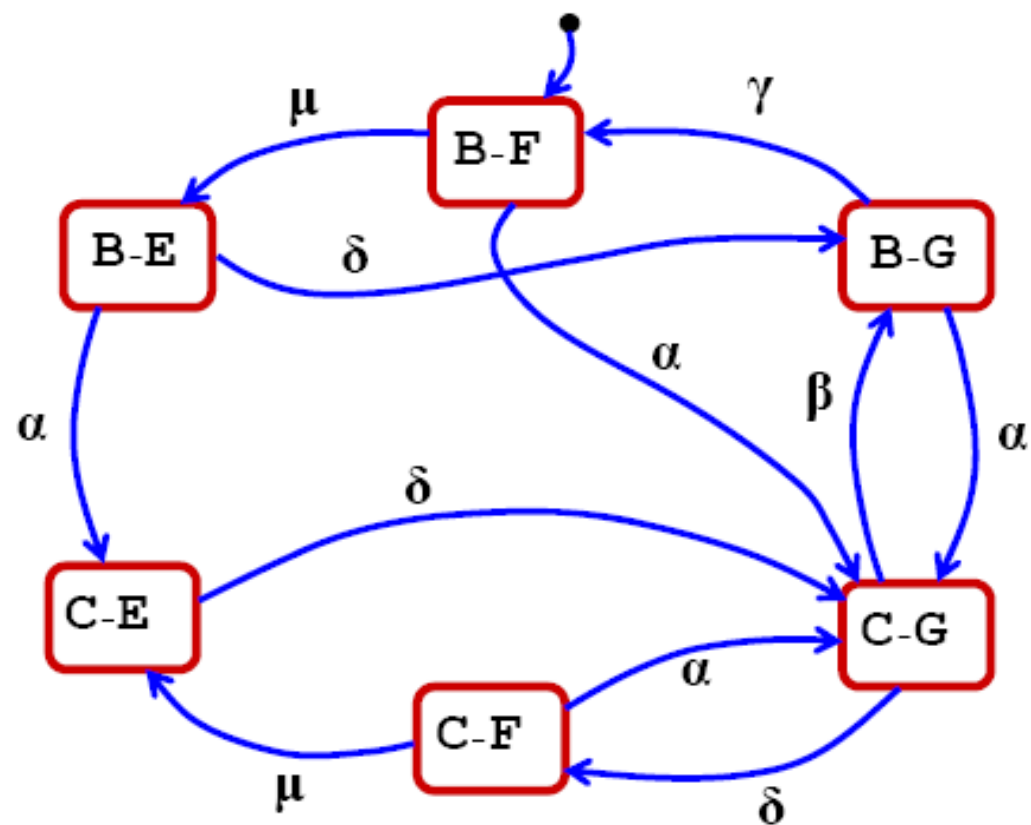
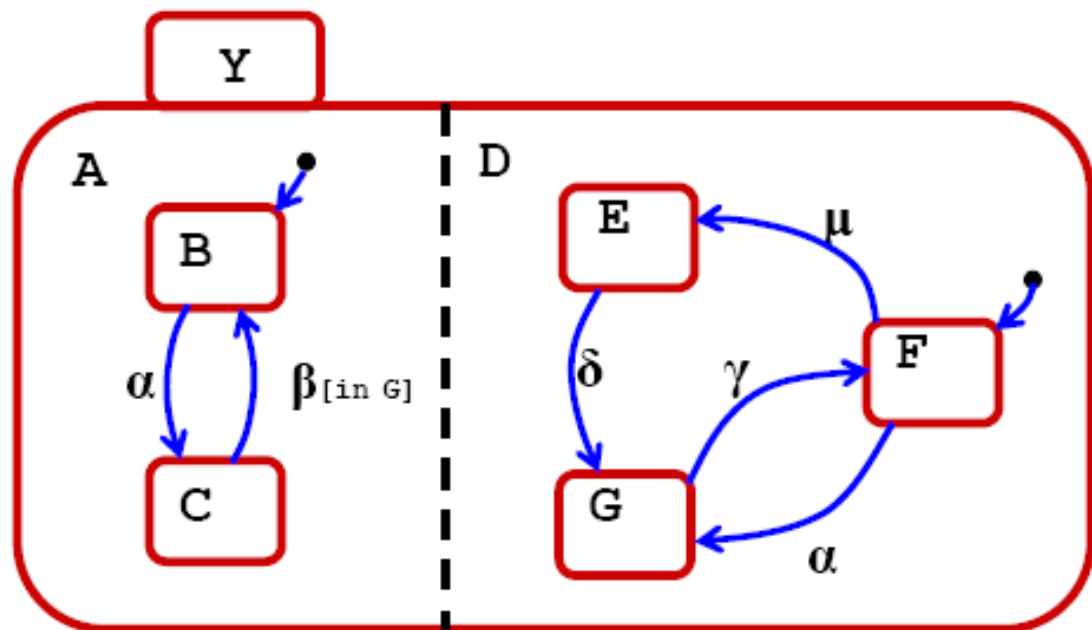
# This Is NOT Good

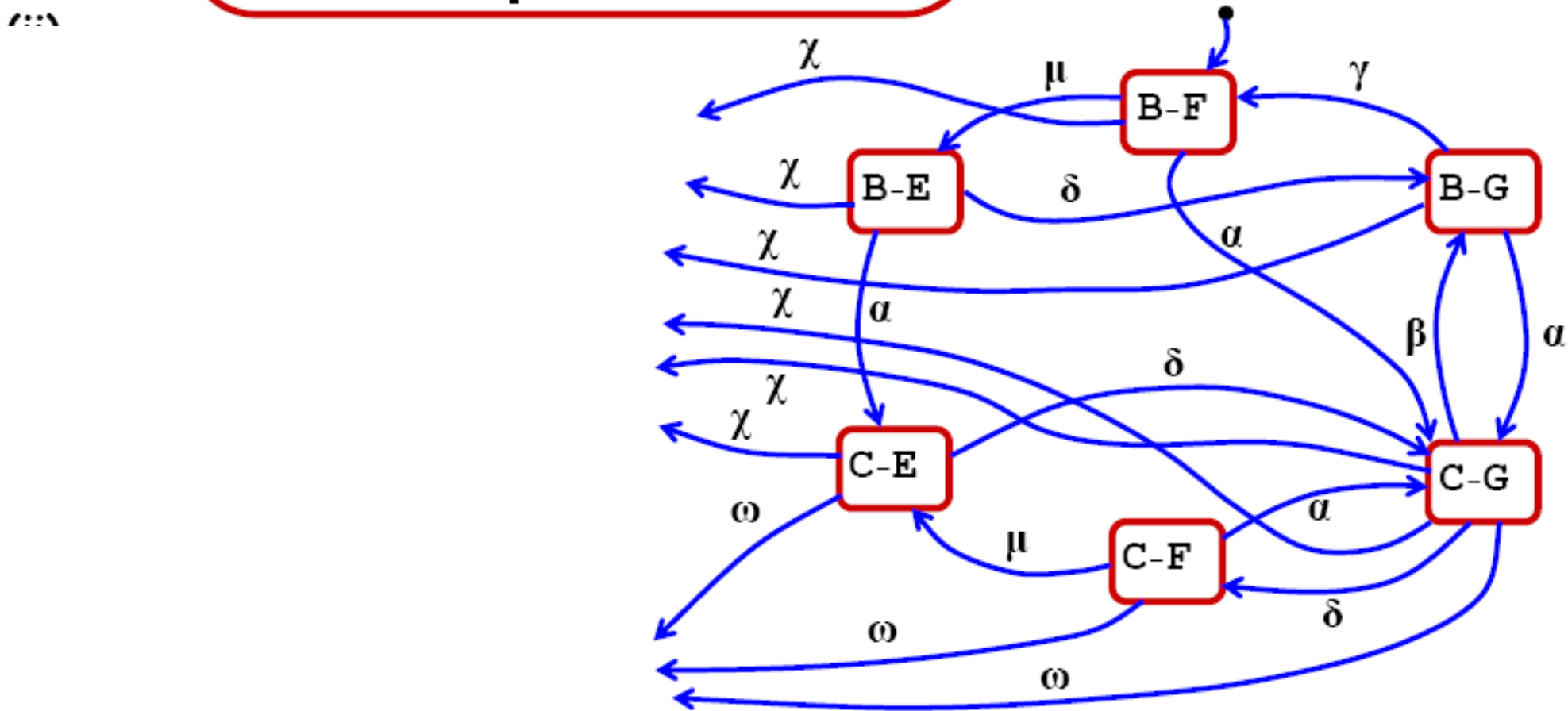
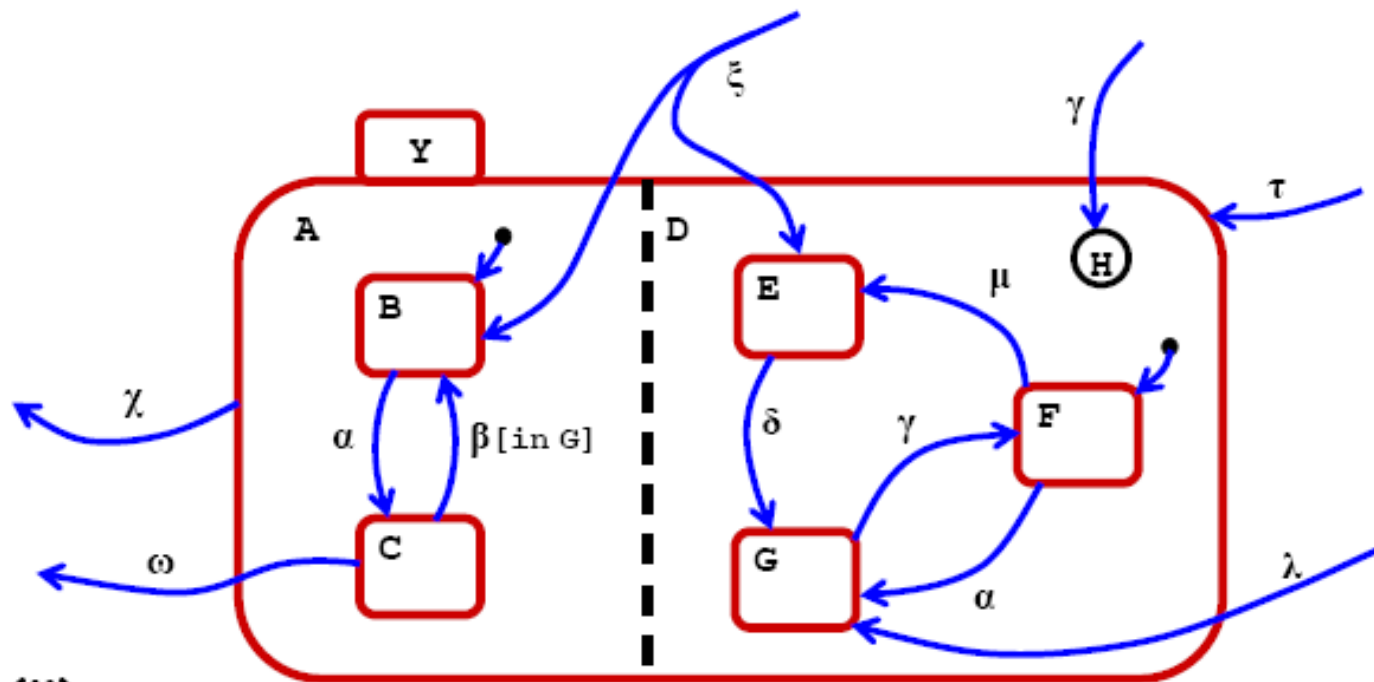
Need a way to describe specifications in a way that is

- 1)comprehensive – a full description of the system
- 2)understandable and intuitive to engineers and programmers
- 3)well defined with no inconsistencies and ambiguities

# Statecharts

- A language to organize, record, and present knowledge about the system behavior
  - The language is **visual**
  - But it is also rigorous, with well-defined semantics
  - Based on describing system states and transitions
  - Provide hierarchy (abstraction) and orthogonality
- The described model is directly executable
  - Essentially programming at a very high level of abstraction
  - Support for automatic code generation





# What's in a Model

- A model is a tool
  - The model is used to think about the design
  - The model serves as a specification for the system
  - The model describes the system and allows for theoretical analysis thereof
  - The model is part of the documentation
- The model *is* the system