3. Modelling

Young W. Lim

2016-03-11 Thr

Young W. Lim

3. Modelling

2016-03-11 Thr 1 / 28

< 67 ▶



3

イロト イヨト イヨト イヨト

"Software Engineering for Embedded Systems...", R Oshana and M Kraeling, 2013

I, the copyright holder of this work, hereby publish it under the following licenses: GNU head Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled GNU Free Documentation License.

CC BY SA This file is licensed under the Creative Commons Attribution ShareAlike 3.0 Unported License. In short: you are free to share and make derivative works of the file under the conditions that you appropriately attribute it, and that you distribute it only under a license compatible with this one.

Young W. Lim

Image: A image: A

3

when & why system models?

- off-line simulation, initial validation
- basis in the subsequent steps
- reduce mistake, risk, and development time
- easy design evaluation and prediction
- iterative approach enhances peformance and reliability
- cost down (reusability, hardware independence)
- reduce error and overhead (automatic code generation)
- robust design
- time-to-market
- reduced development cost

- architecture modeling
- actor description
- function
- entity
- state
- input
- structure
- view

< 67 ▶

- simulation modeling
- system entity's behavior

Image: A image: A

- state char diagram
- hierarchy
- oncurrency
- communication
- Unified Modelling Language (1990)

э

- = behavioral description
 - state
 - transition
 - super state : history

A B A A B A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A

- simulation model
- VHDL entity
- abstraction
- interface

< 4 → <

- Architectural modeling
- Simulation modeling

Image: Image:

- design insight
- design and test
- design reuse
- test and debug
- fast and strong integration
- fast iteration of design and implementation

• simulation modeling performed at the initial stage

Image: A matrix

- software modeling
- visual aid
- behavioral description
- simulation aid
- concurrency offered
- UML

< 行い

- special situation, environment, problem
- which level of modeling is useful
- small vs large complex embedded systems
- UML, simulation may help or burdensom

- processor based simple embedded system
- FPGA equiped complex embedded system

э

- time sequence
- structural actor
- functional actor
- inter-actor relationship

- domain specific modeling language
- model of computation

< 47 ▶

graphical

textural

Ξ.

イロト イヨト イヨト イヨト

- graphical modeling language
- symbols
- connect lines
- diagram
- system constraint

æ

-textural modeling language

- standard key words
- parameter

< **1** → <

- graphic
- text
- architecture
- implemenation
- document
- simulation
- execution

< A > <

Ξ.

- document + architecture : UML diagram
- implementation + execution : C, C# ...
- hybrid model : state chart
 - high level architectural view- implementation + (simulation, execution) : data flow

э

- touch point
- abstraction
- simulation + implementation
- communication, IO

< 行い

- PID Controller
- time-based simulation
- state chart
- data flow

э

- software appliation design process
- design cycle
- x: time
- y: abstraction level
- design iteration
- reduce, bridge gap
- design team
- implementation team
- test team

- at the initial phase
- system requirements
- design issues can be handled

< 行い

- mission critical application DO-254B
- highly complex appliation / systems AGC, Jet, JSF, F-22 Raptor, Car Smart Factory
- large scale development communication, location, language, tools
- no prototype considered new product (non-existent)