EECS 222:
Embedded System Modeling
Lecture 12

Rainer Dömer

doemer@uci.edu

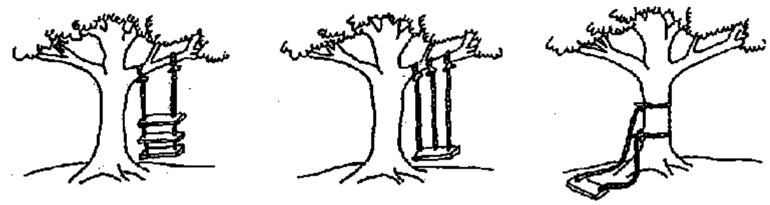
The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 12: Overview

- Embedded System Specification
  - Essential issues
  - Specification Modeling Guidelines
- Project Assignment 6
  - Structural Refinement of the DUT of the Canny Edge Detector

Essential Issues in Model Specification

- An Example ...



Proposed by the project team — Product specification — Product design by senior analyst

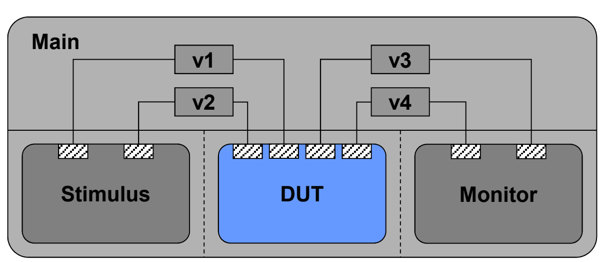
Product after implementation — Product after acceptance by user — What the user wanted

Source: unknown author

EECS222: Embedded System Modeling, Lecture 12 (c) 2019 R. Doemer 3

Specification Model

- Test bench
  - Main, Stimulus, Monitor
  - *Simulation only, no synthesis (no modeling restrictions)*
- DUT
  - Design under test
  - *Simulation and synthesis! (restricted by modeling guidelines!)*



EECS222: Embedded System Modeling, Lecture 12 (c) 2019 R. Doemer 4

Specification Modeling Guidelines

- Specification Model = "Golden" Reference Model
  - first functional model in the top-down design flow
  - all other models will be derived from and compared to this one
- High abstraction level
  - no implementation details
  - unrestricted exploration of design space
- Purely functional
  - fully executable for functional validation
  - no structural information
- No timing
  - exception: timing constraints
- Separation of communication and computation
  - channels and behaviors/modules

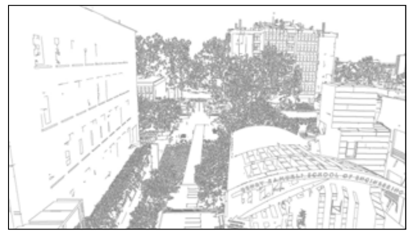
Specification Modeling Guidelines

- Computation: in Behaviors / Modules
  - Granularity: Leaf behaviors = smallest indivisible units
  - Hierarchy: Explicit execution order
    - Sequential, concurrent, pipelined, or FSM
  - Encapsulation: Localized variables, explicit port mappings
  - Concurrency: Potential parallelism explicitly specified
  - Time: Untimed (partial order only)
- Communication: in Channels
  - Communication: Standard channel library
  - Synchronization: Standard channel library
  - Dependencies: Data flow explicit in connectivity

EECS 222 Project

- Application Example: Canny Edge Detector
  - Embedded system model for image processing:
    Automatic Edge Detection in a Digital Video Camera

EngPlaza001.bmp



EngPlaza001_edges.pgm

  - Video taken by a drone hovering over UCI Engineering Plaza
    - Available on the server: **~eecs222/public/video/**
    - High resolution, 2704 by 1520 pixes
    - Video length 9 seconds, using 20 extracted frames for test bench model

EECS222: Embedded System Modeling, Lecture 12 (c) 2019 R. Doemer 7

Project Assignment 5

- Task: Structural Test Bench Model
  - Expected instance tree

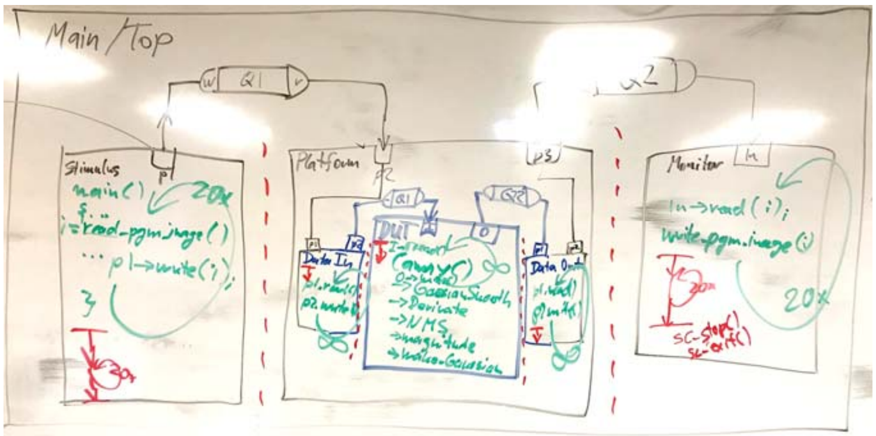
```
Main / Top
|------ Stimulus stimulus
|------ Platform platform
|         |------ DataIn din
|         |------ DUT canny
|         \------ DataOut dout
\------ Monitor monitor
```

  - Communication via standard channels
    - SystemC: **sc_fifo<IMAGE>** based on class **IMAGE**
    - SpecC: **c_img_queue** based on typedef **img**
  - ➢ Pay attention to stack sizes!

EECS222: Embedded System Modeling, Lecture 12 (c) 2019 R. Doemer 8

Project Assignment 5

- Structural Test Bench for the Canny Edge Detector
  - Discussion on whiteboard: Top-level structure, platform for DUT



EECS222: Embedded System Modeling, Lecture 12 (c) 2019 R. Doemer 9

Project Assignment 6

- Task: Hierarchical DUT of the Canny Edge Detector
  - Refine the structural hierarchy of the DUT block
  - Refine the structural hierarchy of the Gaussian Smooth block
- Steps
  1. Refine the DUT structure
     - Gaussian Smooth, Derivative, …, Apply Hysteresis
  2. Refine the Gaussian Smooth structure
     - Receive Image, Gaussian Kernel, BlurX, BlurY
  3. Visualize the structural hierarchy of the model
- Deliverables
  - **`Canny.sc`** or **`Canny.cpp`** (choose one!)
  - **`Canny.txt`**
- Due: February 20, 2019, 6pm

EECS222: Embedded System Modeling, Lecture 12 (c) 2019 R. Doemer 10

Project Assignment 6

- Step 1: Refined hierarchy of the DUT block
  - Expected instance tree

```
Platform platform
|------ DataIn din
|------ DUT canny
|       |------ Gaussian_Smooth gaussian_smooth
|       |------ Derivative_X_Y derivative_x_y
|       |------ Magnitude_X_Y magnitude_x_y
|       |------ Non_Max_Supp non_max_supp
|       \------ Apply_Hysteresis apply_hysteresis
\------ DataOut dout
```

EECS222: Embedded System Modeling, Lecture 12 (c) 2019 R. Doemer 11

Project Assignment 6

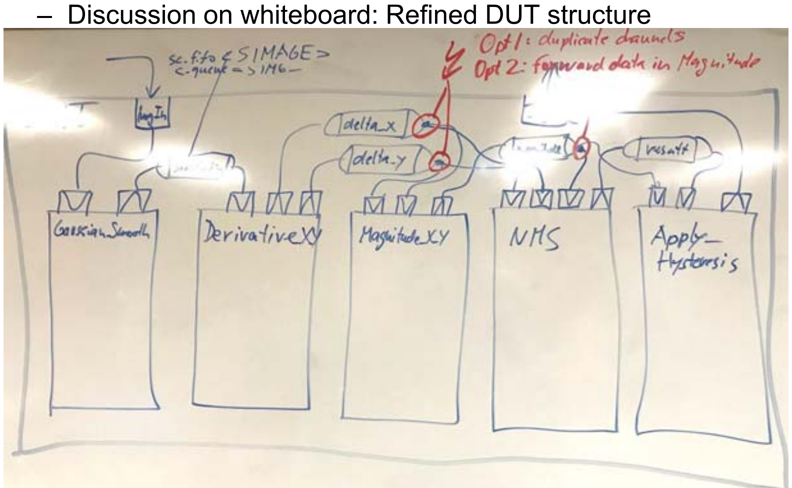
- Step 2: Refined hierarchy of the Gaussian Smooth
  - Expected instance tree

```
DUT canny
|------ Gaussian_Smooth gaussian_smooth
|       |------ Receive_Image receive
|       |------ Gaussian_Kernel gauss
|       |------ BlurX blurX
|       \------ BlurY blurY
|------ Derivative_X_Y derivative_x_y
|------ Magnitude_X_Y magnitude_x_y
|------ Non_Max_Supp non_max_supp
\------ Apply_Hysteresis apply_hysteresis
```

EECS222: Embedded System Modeling, Lecture 12 (c) 2019 R. Doemer 12

Project Assignment 6

- Structural model of the DUT of the Canny Edge Detector
  - Discussion on whiteboard: Refined DUT structure



EECS222: Embedded System Modeling, Lecture 12 (c) 2019 R. Doemer 13