EECS 222: Embedded System Modeling Lecture 19

Rainer Dömer

doemer@uci.edu

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

Lecture 19: Overview

- Course Administration
  - Instructor evaluation
  - Final exam
- Unified Modeling Language (UML)
  - Overview
  - Example Diagrams
- EECS 222 Project Discussion
  - Assignment 8
  - Final technical report

Course Administration

- Final Course Evaluation
  - 9th through 10th week
  - May 22, 2017, through June 11, 2017, 11:45pm
  - Open until next Sunday night
  - Online via EEE evaluation application
- Evaluation of Course and Instructor
  - Voluntary
  - Anonymous
  - Very valuable!

➢ Please help to improve this class!

  - Please spend 5 minutes!

Course Administration

- Final Exam
  - Allocated time
    - Wednesday, June 14, 4:00-6:00pm
  - Location
    - Not applicable, we use electronic submission!
  - Format: Final Project Report
    - Submission script: `~eecs222/bin/turnin.sh`
    - Directory name: `hw8`
    - File names: `EECS222_Report.pdf` `Canny.sc` *or* `Canny.cpp`
  - Hard deadline!
    - June 14, 2017, 6pm

Unified Modeling Language (UML)

- Status
  - UML 2.0: Modeling Language in Software Engineering
  - standardized by OMG (Object Management Group) in 1997
  - standardized by ISO (Intl. Org. for Standardization) in 2005
- Goals
  - Raising the Level of Abstraction
  - Modeling of software applications
    - ➢ before coding!
  - Specification of software architecture
  - High-level description of software architecture to enable
    - scalability
    - security
    - robustness
    - maintenance
    - extendability
    - code reuse
  - Model Driven Architecture (MDA)

Unified Modeling Language (UML)

- What is UML?
  - 13 standard diagrams
    - Specification
    - Design
    - Documentation
  - Graphical representation of …
    - Software architecture
    - Software structure
    - Software behavior
    - Object relations
    - ...
    - ➢ Not executable!
  - Commercial tools available for …
    - Graphical capture
    - Editing
    - Code generation (template code)

Unified Modeling Language (UML)

- UML Standard Diagrams
  - Structure Diagrams
    - Class Diagram
    - Object Diagram
    - Component Diagram
    - Composite Structure Diagram
    - Package Diagram
    - Deployment Diagram
  - Behavior Diagrams
    - Use Case Diagram
    - Activity Diagram
    - State Machine Diagram
  - Interaction Diagrams
    - Sequence Diagram
    - Communication Diagram
    - Timing Diagram
    - Interaction Overview Diagram

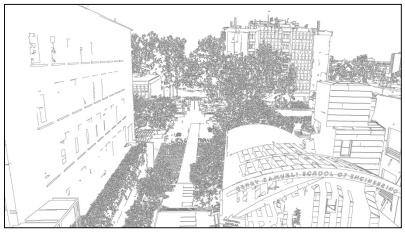
Unified Modeling Language (UML)

- UML Resources
  - Online Documents
    - Object Management Group (OMG)
      - `www.uml.org`
  - Online Tutorials
      - `https://www.tutorialspoint.com/uml/`
      - `http://www.sparxsystems.com/uml-tutorial.html`
  - Invited Talk at UCI in 2004
    - Dr. Wolfgang Mueller, C-LAB, Paderborn, Germany
      - `Lecture19_UML.pdf`

EECS 222 Project Discussion

- 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

Project: Homework Assignment 8

- Task: Throughput optimization of Canny Edge Decoder
  - Back-annotate more realistic delays and observe throughput
  - Maximize the pipeline throughput by using fixed-point arithmetic
- Steps
  1. Improve test bench with logging of frames per second (FPS)
  2. Estimate timing based on allocated CPU cores and ASICs
  3. Replace floating-point with fixed-point arithmetic in NMS block
  4. Back-annotate the improved NMS stage delay
  5. Final technical project report
- Deliverables
  - **`Canny.sc`** or **`Canny.cpp`** (choose one!)
  - **`EECS222_Report.pdf`** (in lieu of final exam)
- Due: By next week: June 14, 2017, 6pm (Thursday evening)

Project: Final Technical Report

- Final Technical Project Report
  - Title
    - *Specification and Modeling of a Canny Edge Detector for System-on-Chip Design*
  - Contents
    - "Story" of the course project
      - From downloading initial C reference code
      - Via describing and simulating in SpecC or SystemC SLDL
      - To modeling and optimization for SoC design
    - Use the results of Assignments 1, and 4 through 8
    - Conclude with a summary of the lessons learned
  - Length
    - About 12 pages (including title page, figures, and references)

Project: Outline of Final Technical Report

1. Title page
   - Project title, author, date, course number and title
   - Abstract
2. Introduction
   - System-level modeling and design
   - Essential concepts and coverage in SpecC/SystemC SLDL
3. Case study using the Canny Edge Detector application
   - Obtaining and studying the Canny application
   - Creating a simulatable model in SpecC/SystemC SLDL
   - Creating structural hierarchy with test bench
   - Pipelining and parallelization
   - Performance estimation and throughput optimization
4. Summary and Conclusion
   - Lessons Learned
   - Future work
5. References