CS 446 / ECE 452 / CS 646 – Spring 2013
Course Project Part1 – Software Architecture

1. Overview

For the term project, you are asked to architect, design, and implement a complex software-intensive system. The project will be completed in four stages, where in the first stage you will model the software architecture of the system, in the second stage you will provide detailed design models, in the third stage you will implement the system and present your project to the instructor or the TAs, and in the fourth stage you will reverse engineer the as-implemented architecture of your system. A major part of your project grade will come from your implementation – **so you should start coding early**.

It is preferred that your project falls into one of the following broad categories, but projects outside these categories will be considered as well.

Category A: Software for a mobile device, such as an application (app) for Android, iOS, or Blackberry platforms. The application has to provide unique functionality, different from similar applications already on the market. The application can be written against an emulator of the mobile platform.

Category B: Software embedded into a hardware platform, such as software that would operate on-board GPS device in an automobile. The application needs to interface with the simulator of the chosen embedded system, if interface with the actual hardware is unavailable.

Category C: Web-based application, such as an eCommerce application that provides services to online users. Note however that script-based implementations, primarily using PHP or Java Script for instance, will not be accepted.

Category D: Desktop application, such as finance, accounting, or other special-purpose software including video games.

Sample topics from the previous terms:

- Personal Finance Manager – A web based application to provide integrated management of bills from different companies.
- University Room Scheduler – Web based application for reserving rooms.
- Iron Cook – A device-based cook book with recipes and ingredients.

- ECU Control Wiper Blades – Hardware circuit design for controlling wiper blades. The low-level software provides control features along with fail-safe control mechanisms.
- Knight Vengeance – Blackberry-based video game with level editing.

The key requirements for each project are that

(1) It should be something that excites your team, e.g., something that you expect to do in your job post graduation;

(2) It exhibits sufficient technical complexity, such as (a) it uses networking for communication, (b) it interfaces with a DBMS, and (c) it is implemented as a multi-threaded application;

(3) The scope of the project should assume at least 5 hours per week for each person of the development/design time.

(4) You are allowed to use existing frameworks, libraries, or even third-party software as components in your system, but the majority of code needs to be your own work.

2. Software Architecture Design

Before you start your architectural design, you are to specify the corresponding UML use case models for your system. You are then to iterate through each use case and identify software components and connectors needed to support the underlying functionality.

Once you have identified software components and connectors, you are to specify them in detail. As part of this specification, you are to select a corresponding architectural style (e.g., layered design) or a combination of styles that best fits the subsystem decomposition. You should discuss at least two plausible architectural solutions with your group mates, and select one that best fits your design goals. Write down the justification for selecting one style or a combination of styles.

Your architecture should be easy to understand. You should focus on specifying simple interfaces, and moderate interactions among your subsystems. You should also focus on higher-level design goals such as performance and modifiability, instead of lower-level concepts such as classes, variables and control flow.

You are encouraged to use UML diagrams to specify your architecture, but you are allowed to use other notational schemas (e.g., data flow diagrams) where appropriate.

3. **Submission Requirements**
   - Deliverables: a printed Software Architecture Description Document (based on IEEE ISO/IEC 42010:2007) that includes:

     0. **Abstract**
     1. **Introduction**
        Scope, Purpose, Intended Users
     2. **Software Architecture Overview**
        a. System Architecture Context
           Describe the overall system architecture, i.e., how the software interacts with its environment
        b. Software Architecture Representation
           Illustrate and describe your software architecture decomposition
           Show the structural design that you are proposing to implement, with descriptions of each major component and connector, and their interactions
     3. **Architectural Goals and Constraints**
        a. Explain the architectural rationale for creating different components and connectors, including justification of the style selection
        b. Analyse and discuss architectural elements affecting performance, maintainability, and other related quality aspects of the system
     4. **Use Case View**
        UML use case models and use case descriptions
     5. **Behavioural View**
        Specify the information and control flow within the system, e.g., using UML sequence, state machine, or communication diagrams
        Explain each presented view
     6. **Viewpoint #3**
        Specify a concern, such as concurrency or run-time performance, that is crucial to your architectural design, and model it as a viewpoint
        Explain each presented view
     7. **Deployment View**
        UML deployment models depicting how the software components will be deployed onto hardware and networking nodes

   - **Please limit your report to 20 pages maximum (strict limit).**

- Deliverables: A printed report submitted in class that complies with the project specifications above.

- The deadline for this deliverable is Mon (Jun 3rd) by 8:55am.
- No late submissions will be accepted.