Lab 3: Architectural analysis and documentation

SE 464 Software Architecture and Design

0. Overview

The objective of this lab is for you to practice architectural analysis and documentation on an existing, successful software-intensive system. As a side effect, you will also have the opportunity to "learn form the masters."

The lab is to be performed by your Lab-3 Team. There are three deliverables: sign-up for the system to be analyzed (Section 1); report (Section 2); and an in-class presentation and the presentation slides (Section 3).

The key marking criteria are technical depth, technical correctness, and presentation clarity, rather than quantity.

The sign-up, together with a list of documentation resources, is to be completed on the course wiki. The report and the presentations slides are to be uploaded to the course wiki.

1. System selection (1 mark for sign-up; 1 mark for resource list)

Normally, your team is expected to pick a system from the following list of pre-approved systems: Apache Web Server, Android OS, Audacity (audio editor), Battle For Wesnoth (computer game), BerkeleyDB, Eclipse, Firefox, FreeRTOS, GDB (debugger), GIT (version control), GPSD (GPS Deamon), Hadoop (file system), Linux kernel, LLVM (compiler suite), MediaWiki, Mercurial (version control), MySQL, Open MPI, pulseaudio (audio server), Puppet (IT management tool), Selenium (test tool), Sendmail, Skype, Spring (web framework), Struts (server framework), The Glasgow Haskell Compiler, and VTK (visualization toolkit).

The sign-up is on a first-come, first-serve basis, and each team should analyze a different system. When you sign-up, you need to provide a list of resources you will use to complete this lab. The list does not need to be complete; the main purpose of this list is to ensure that you have a sufficient set of documentation available to you in order to complete this lab.

In special cases, your team may propose a different system, but the system would need to be approved by the course instructor in advance to the sign-up deadline.

The resources can be the system documentation, books, and articles. For example, some of the systems listed above have been described in the "The Architecture of Open-Source Applications" (http://www.aosabook.org/en/index.html). There are books for many of these systems (e.g., Apache Web Server and Linux kernel). Note that you may need to seek out multiple sources to complete this lab.

2. Report (36 marks)

Your report should cover the following topics.

2.1 System functionality (5 marks)
State the purpose of the system. Describe its functionality by listing 3-7 main user stories that it supports. For the user stories, use the following format:

As a <user role>, I want <goal/desire> so that <benefit>

2.2 Quality attributes and scenarios (5 marks)

Make a list of the key quality attributes that the system must satisfy. Document each attribute with one or more quality attribute analysis scenario (see the lecture material on quality attributes for examples).

2.3 Architectural views (10 marks)

Provide a list of architectural views that are relevant for the system and diagram or another adequate representation (e.g., a list or code fragments) of each view, along with a short description.

Examples of views that may be relevant are layered view, deployment view, module view (e.g., file or component packaging), behavioral view, and representation of specific mechanisms (e.g., the intent mechanism in Android or dependency injection in Spring).

2.4 Architectural analysis (10 marks)

For each quality attribute analysis scenario, briefly discuss which design tactics and architectural styles or patterns (not necessarily limited to those covered in the lectures) have been applied to address the attribute and relate the tactics, patterns or styles to the relevant architectural views.

2.5 What are the key weaknesses of the architecture? (5 marks)

Identify and briefly discuss the main weaknesses of the architecture. How could they be addressed?

2.6 References (1 mark)

List the references to the resources (books, papers, online articles, etc.) you used to prepare this report.

3. Presentation (15 marks)

The presentation should take 10 minutes. Your slides should briefly explain the purpose of the system (1 slide) and its key quality attributes (1 slide); show the main architectural views (2-3 slides) and how the quality attributes are addressed (1-2 slides); and identify the main weaknesses and how they could be addressed (1-2 slides). The numbers of slides are not strict limits, but recommendations.