

SE4930 Lab 4 - Architectural Design

Dr. Walter Schilling

Due: 23:00 CDT January 7, 2013 23:59

1 Architectural design

“The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.

By ‘externally visible’ properties, we are referring to those assumptions other components can make of a component, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on. The intent of this definition is that a software architecture must abstract away some information from the system (otherwise there is no point looking at the architecture, we are simply viewing the entire system) and yet provide enough information to be a basis for analysis, decision making, and hence risk reduction.” - Bass, Clements, and Kazman. Software Architecture in Practice, Addison-Wesley 1997

“Architecture is the organizational structure of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components and subsystems.”
- UML 1.3

The two definitions above express the concept of a software architecture (and should be similar to what you learned in your Software Architecture course if you have already taken that course). In this lab, you will be developing an architecture for your system. In doing so, you will need to determine the overall structure for your system. To do this, you will need to focus on three things: components, connectors, and configurations. A component, in our terminology, represents a unit of computation or a data store of information. Components have state and functionality associated with them. A connection is an architectural element which models the interactions between components as well as the governing rules for those interactions. The configuration (topology) is a graph which describes the architectural structure of the system.

In lab, you will need to sketch an architectural diagram showing the topology of your proposed system and showing the relationship between the components your system will have. Examples (and more explanation) can be found online in this presentation sunset.usc.edu/~nenoc/cs477_2003/February11.ppt.

You may use whatever drawing tool you desire to show your architecture. Tools used by previous classes include EA and Visio. That being said, the starting point probably should be a whiteboard and marker or pencil and paper.

2 Data Access Matrix

In last week’s lab, you defined the assets that your system will hold. In specific, you looked at the confidentiality, integrity, and availability needs for each piece of asset. In this week’s lab, you will define an access control matrix for the data. An access control matrix shows the relationship between each user role that your system will have and the required ability to create, read, update, and delete the data.

In your case, you will start by listing each piece of data as a row in the table and defining each user role within your system as a column. Where a user requires access to a given data item, the appropriate abilities must be listed. If a user has no access to the data, then there will be no entry in the cell. But if there is any access to

User Roles									
Data	Application Administrator	Application server process Identity	Consumer Loan Agent	Database Administrator	Network Administrator	Remote Anonymous User	Remote User	Secure web server process identity	Website Administrator
user id									
user password									
agent id									
agent password									
personal data									
loan request data									
loan response data									
credit report									
unsecured web pages									
secured web pages									

- Item has a Use Case defined but no Conditions
- Item has a Use Case defined and Conditions
- A Use Case has not been defined
- Condition exists, but a Use Case has not been defined

Figure 3-3: Data access control matrix

the entry, then the data should be entered. An example of this material is provided below. Similar examples are available in chapter 13 and 14 of the operating systems textbook.

3 Time Analysis

Once the team members have completed their work, the spreadsheet shall be updated with effort and allocation.

4 Deliverables

Each team should submit through the course website a pdf with the following

1. A cover page with the project name, team members, date, course, and assignment information
2. The developed architectural diagram
3. A brief explanation (on paragraph or so) describing the purpose for each of the components within the system. Why does it exist and what will it do?
4. A data access matrix showing the assets and the users roles relative to to that data

Each team should also update and submit its spreadsheet with time and effort information included.