**ACS 560 Midterm I: Software Engineering (Fall 2011)**

***(Type answers and include question/only brief answers required/1-3 sentences each subtopic)***

1. **[10 pts] We have covered the architectural hierarchical decomposition process from your initial vision document. Describe:**
   1. **Application architecture**

The application architecture is a structured solution for the organization of the application. It shows the interaction between application packages, data stores, the user, and any external services. In this class, I utilized the .NET architecture for my project.

* 1. **Information architecture**

The information architecture is a representation of the software using functional requirements and design parameters. It is concisely displayed in the design matrix, which shows the relationships between the FRs and the DPs.

* 1. **Systems architecture**

The systems architecture is a representation of the software generated from the design structure matrix. It shows how the design parameters of the system are related to one another.

* 1. **UML architecture**

The UML architecture is a representation of the software using UML diagrams to show the system from different perspectives.

1. **[10 pts] In RUP, we are leveraging the axiomatic design process in the inception and elaboration phase. Describe the requirements engineering and elicitation process in more detail from initial gathering of VOC to CN to FR in axiomatic design, including risk concepts.**

The first step in the process is to define the customer needs (CNs). The CNs will lead to the definition of functional requirements (FRs). Next, design parameters (DPs) will be determined and mapped to the FRs. The FRs and DPs will continue to be decomposed to successively lower levels of detail. Once the FRs and DPs have been determined, risk analysis can be completed using tools such as Failure Mode and Effects Analysis (FMEA).

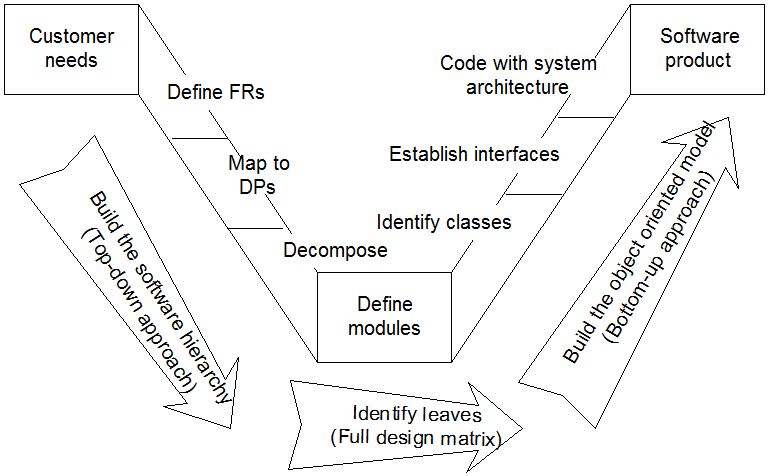
* 1. **Design matrix results in which UML diagram type?**

The design matrix will result in a UML class diagram, where the FRs correspond to objects, the DPs correspond to data, and the links between them will be operations.

* 1. **DSM results in which UML diagram type?**

The design structure matrix will result in a UML component diagram.

* 1. **What is the V-Model?**



The V-Model is a framework for design which is based on the idea that design cannot proceed without going back and forth between the functional and physical domain. It combines the advantages of the top-down and bottom-up approaches. It consists of three phases: decomposition, mapping, and integration.

* 1. **QFD**

QFD stands for Quality Function Deployment. It is a method to translate user needs and desires into design quality. In this class, we used the QFD tool called House of Quality (HOQ) that defines the relationship between customer needs and functional aspects of the design.

* 1. **FMEA**

FMEA stands for Failure Mode and Effects Analysis. It is a procedure for analyzing potential failures of a system by classifying severity and likelihood of the failures. It also helps to identify ways to mitigate the severity or reduce the likelihood of the failures.

1. **[5 pts] We have covered how to establish class architecture first as a foundation to detailed design with UML. When using Axiomatic Design process to develop classes, how do the following map to OO Design object elements [hint: see paper handout on “Object Oriented Design with Axiomatic Design” :**
   1. **FR**

An FR corresponds to an object.

* 1. **DP**

A DP corresponds to data or input for the object.

* 1. **FR/DP design matrix intersection**

The intersection of an FR and DP corresponds to an operation.

1. **[5 pts] Describe the purpose of these tools/software used in your project:**
   1. **Acclaro DFSS**

Acclaro was used to perform Axiomatic Design, including capturing of Functional Requirements, Design Parameters, Design Matrix, Design Structure Matrix, and FMEA. It was also utilized to create the system architecture and QFD House of Quality.

* 1. **Visio**

Visio was used to create the application architecture for the project. While it was originally intended to also complete UML modeling in Visio, the StarUML tool was determined to be better suited for this task.

* 1. **Basecamp**

Basecamp was not used for the project due to being a one-person team.

* 1. **MS Project**

Microsoft Project was used to develop and maintain the schedule for the project in a Gantt Chart.

* 1. **MS PowerPoint**

Microsoft PowerPoint was used to develop and deliver presentations.

1. **[5 pts] Describe the standards applied in this project**
   1. **IEEE-830 SRS**

IEEE-830 is a standard which provides recommended practices for Software Requirements Specifications. It specifies an SRS is complete if it includes all significant requirements for the system, definitions of the software response to all input data in all situations, and definitions of all terms and units of measure. It provides a recommended document structure, as well as different organizational structures for the requirements section.

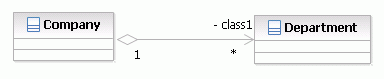
* 1. **IEEE-1058 PMP**

IEEE-1058 is a standard which provides a framework for writing software Project Management Plans. It specifies the format and contents of software project management plans, and identifies elements that should appear in the PMP.

* 1. **IEEE-1016 SDD**

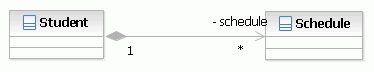
IEEE-1016 is a standard which provides recommended practices for Software Design Descriptions. It specifies the organizational structure for an SDD, including elements such as system architectural design, detailed description of components, and user interface design.

1. **[5 pts] In OO design, describe the concept [with symbol]:**
   1. **Aggregation**



Aggregation is a type of relationship depicting a “has a” relation. It shows a class is a part of another class, but the subordinate class will not be destroyed if its container is destroyed.

* 1. **Composition**

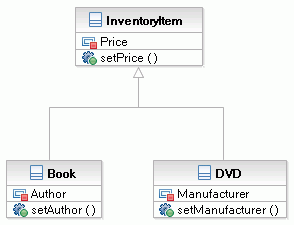


Composition is a type of relationship depicting a “owns a” relation. It shows a class is a part of another class, and if the container is destroyed, every instance that it contains is destroyed.

* 1. **Polymorphism**

Polymorphism is the ability to create a variable, function, or object that has more than one form. It means that some object or operation will behave differently in different contexts. An example is the overloading of the + (plus) operator.

* 1. **Inheritance**



Inheritance is a type of relationship depicting an “is a” relation. It shows a class is a specific type of another class, and as such, has the same attributes and operations of the inherited class.

* 1. **Blackbox**

http://upload.wikimedia.org/wikipedia/commons/thumb/f/f6/Blackbox.svg/320px-Blackbox.svg.png

A black box is a device, system, or object which can be viewed solely in terms of its input, output, and transfer characteristics without any knowledge of its internal workings.

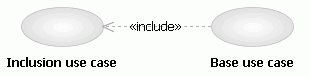
1. **[5 pts] In OO design, describe the concept [with symbol]:**
   1. **Public operations**

A public operation of a class is a method that can be accessed by all other classes. It is denoted by including a '+' before the name of the operation.

* 1. **Private operations**

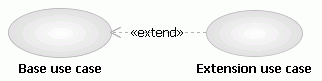
A private operation of a class is a method that can only be accessed by the class itself and is not visible to outside classes. It is denoted by including a '-' before the name of the operation.

* 1. **Inclusions**



An include relationship is a relationship in which one use case includes the functionality of another use case (the inclusion use case). The inclusion use case describes common functionality that is common to two or more use cases.

* 1. **Extensions**



An extend relationship is a relationship in which one use case (the extension use case) extends the behavior of another use case. The extension use case describes additional behavior that augments the behavior of the base use case.

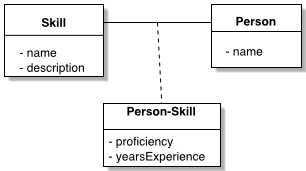
1. **[5 pts] You were given a handout on software project management by Fairley. Describe how the following concepts are used in your project.**
   1. **Architecture Decomposition View (ADV)**

The Architecture Decomposition View presents functionality in terms of manageable work pieces. It identifies modules and breaks them down into sub-modules. This type of view was used throughout the project to decompose the system into successively smaller modules.

* 1. **Work Breakdown Structure (WBS)**

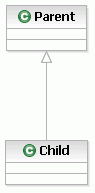
The Work Breakdown Structure is the deliverable oriented decomposition of the project. In this project, we used Microsoft Project to develop a schedule that is very similar to a work breakdown structure, as it identified the set of project deliverables.

1. **[5 pts] In Object oriented (OO) design, describe the concept [with symbol]:** 
   1. **Association**



An association is a relationship between two classifiers that describes the reason for the relationship and the rules that govern the relationship.

* 1. **Generalization**



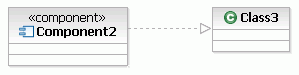
A generalization is a relationship in which one element is based on another element.

* 1. **Dependency**

Two class shapes, named Client and Supplier, are connected by a dashed line with an open arrow pointing from the Client to the Supplier.

A dependency is a relationship in which changes to one model element impact another model element.

* 1. **Realization**



A realization is a relationship between two model elements in which one model element realizes the behavior that the other model element specifies.

* 1. **Annotation**

http://www.tutorialspoint.com/images/uml_note.jpg

An annotation is a note that is added to UML diagrams to provide additional information. It is merely a comment to make the diagram more clear.

* 1. **Interface**

Interface shown as a rectangle with the keyword interface.

An interface is a classifier that declares a set of public features and obligations. It is a contract that any classifier realizing (or implementing) the interface must fulfill. The interface is an abstract class with no attributes and only abstract operations.

1. **[5 pts] Define each performance attribute:**
   1. **Efficiency**

Efficiency is the amount of computing resources and code required to perform the functions of the application.

* 1. **Flexibility**

Flexibility is the effort needed to modify operational programs.

* 1. **Integrity**

Integrity is ensuring data corruption is avoided.

* 1. **Security**

Security refers to protecting the software from accidental or malicious access, use, modification, destruction, or disclosure.

* 1. **Maintainability**

Maintainability is the ease with which the software can be maintained.

* 1. **Portability**

Portability refers to the effort required to port the software to another host machine or operating system.

* 1. **Reliability**

Reliability is the extent to which the program performs with the required precision.

* 1. **Usability**

Usability is the effort required to learn, operate, prepare input, and interpret output.

1. **[5 pts] Describe how you have used your concept map and basecamp tool to organize your work as a team using RUP as a guide.**

As indicated above, basecamp was not used. The concept map was used as a data repository for all deliverables of the project. It was organized according to the two phases of RUP that are completed in this class, the inception phase and the elaboration phase. Finally, color coding of elements in the Cmap was used in order to indicate the status of each task.

1. **[5 pts] You are working on SWEBOK reviews KA-1 through KA-11. Which three areas apply to your role in the project and why?**
   1. **Software Requirements**

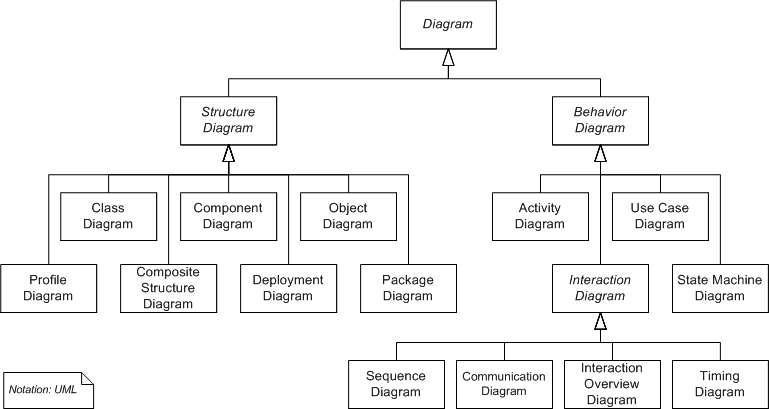
The Software Requirements KA is applicable as one of the key components of the project is the development of the Software Requirements Specification. My role has included elicitation and decomposition of the requirements.

* 1. **Software Design**

The Software Design KA is also relevant as the Elaboration Phase of RUP is chiefly concerned with design of the software application. My role has included design of the application architecture as well as design of the application as a whole using UML modeling.

* 1. **Software Engineering Management**

The Software Engineering Management KA applies as one of my main roles for this project has been acting as project manager. This role includes development of the Project Management Plan and the project schedule. After these were developed, the focus was on ensuring the project follows the process and remains on schedule.

1. **[10 pts] Other than class and component diagrams that all teams are required to use, select 3 UML behavior diagram types that your project could use and why?** 

In addition to class and component diagrams, my project could also use activity diagrams, use case diagrams, and sequence diagrams. Use case diagrams will clearly show how the application will be used by different user groups. These diagrams can help ensure that all important use cases are being addressed. Sequence diagrams will be used to show the sequence of operations and flow of control internal to the application. Activity diagrams will be used to illustrate algorithms used by the application, specifically the algorithm for analyzing microblog data and detecting emergency situations.

1. **[5 pts] We reviewed UML and related concepts for automation in software systems engineering.**
   1. **What is the relationship between UML and SysML?**

SysML is a specialized usage or profile of UML. SysML was developed by a consortium of organizations in order to mitigate UML’s software bias.

* 1. **What is executable UML?**

Executable UML is a profile of UML that can be used to define the structural and behavioral semantics of the system. This allows for validation of the behavioral specification of the system. It also acts as an implementation of the system.

1. **[5 pts] Describe how this course has helped you organize your team**
   1. **Management**

Several documents were completed describing how the project would be managed, including the Project Management Plan (PMP) and the project schedule. These documents were used to decide and document the processes and schedule used to manage the project.

* 1. **Architecture**

Multiple different architectural views were used to design the organization of the system. The first architecture used was the application architecture, followed by the information, system, and UML architectures. Each view progressively helped to refine the design for the project.

* 1. **Detail design**

Multiple activities have been undertaken to facilitate detailed design. First, Acclaro was used to perform axiomatic design, resulting in the design matrix and design structure matrix. These matrices were then used to launch into UML modeling to further define the system. Finally, the Software Design Description based on IEEE -1016 was used to document the design of the project.

* 1. **Documentation**

All aspects of the project, from management to requirements to design were documented using established standards and templates. The documentation was reviewed by the professor and other classmates, and updated to correct deficiencies. The Cmap acted as a repository for all versions of the documentation.

1. **[5 pts] Cyberphysical systems: Describe** 
   1. **What is it?**

Cyberphysical systems (CPS) integrate computation, communication, and control into physical systems. According to the NSF, CPS “refers to the tight conjoining of and coordination between computational and physical resources.”

* 1. **Why is it important?**

CPS is the primary area where disruptive technologies emerge. Advances in CPS will produce systems which are quicker, more precise, highly efficient, and able to function in dangerous or inaccessible environments. They can be applied to a wide range of domains, including transportation, defense, energy, health, agriculture, and infrastructure.

1. **[5 pts] Software Engineering: Identify a topic of interest in**
   1. **Past – Internet**

One area of software engineering history that I find fascinating is the rise of the internet in the 1990s. (Of course, the mechanics of the internet were around long before the 90s, but it didn’t come to prominence until then.) It was a major disruptive technology that has completely changed the way we approach the world. Now, nearly everything we do is online.

* 1. **Present – Software Defined Radio**

In the past, communications equipment was largely hardware, with software acting only as the mechanism for control. Now, advances in digital electronics technology have allowed many functions typically left to hardware (mixing, filtering, modulation, demodulation, etc.) to be implemented in software. The is important because it allows radios to adapt to new protocols without specialized hardware, allowing devices like cellphones to utilize the latest protocols with only a change in software.

* 1. **Future – Data Mining**

Data Mining is a relatively new area of computer science. It focuses on discovering patterns from large data sets. It has important implications for understanding our world and utilizing the vast amounts of data that are produced and collected on a daily basis. It has applications in many areas, including business, geo-spatial mining, and surveillance. It also faces many challenges, such as privacy concerns and ethics.

**Bonus: [10 pts] List up to 10 aspects of this course you enjoyed/learned from the most?**