

# **Deliverable #3: Phase #2 Model**

## **Due: Monday, November 23, 2:30 PM**

### **System Design Document**

1. Introduction
2. Subsystem decomposition
  - 2.1 Phase #1 prototype decomposition
  - 2.2 System decomposition
  - 2.3 Design evolution
3. Design strategies
  - 3.1 Hardware/software mapping
  - 3.2 Persistent data management
  - 3.3 Design patterns
4. Subsystem services
5. Class interfaces

### **Contents**

#### **All sections:**

- Where applicable, every section must have **traceability**

#### **Introduction:**

- An overview of the project, mainly based on the project description, and an overview of the document

#### **Phase #1 Prototype Decomposition:**

- A description of every logical subsystem implemented in the Phase #1 prototype
- Class diagrams and packages must show which classes are included in every logical subsystem
- UML component diagrams must depict all logical subsystems and the dependencies between them

#### **System Decomposition:**

- A description of every subsystem required for the entire *cuPID* system
- Class diagrams and packages must show which classes are included in every subsystem
- UML component diagrams must depict all subsystems and the dependencies between them
- Design must promote high cohesion and loose coupling

#### **Design Evolution:**

- Explain the design differences between the decomposition for the Phase #1 prototype and the decomposition for the entire system
- Discuss design choices made for decomposing the entire system, and why they are superior to the ones made for Phase #1
- Discuss the evolution of your design from Phase #1 to the next phase

#### **Hardware/Software Mapping:**

- A description of strategy (architectural style) for mapping subsystems to components, and components to nodes
  - reasoning must be justified
- All subsystems for the entire system must be depicted as assigned to a node, using UML deployment diagrams
- Subsystems must be grouped into runtime components, where applicable

### **Persistent Data Management:**

- A description of strategy for storing persistent data and how the data is organized in storage to avoid duplication
  - reasoning must be justified
- A description of all objects in persistent storage

### **Design Patterns:**

- A description of the use of established design patterns
  - reasoning must be justified
- Failure to use a design pattern where warranted will be penalized
- Incorrect use of a design pattern will also be penalized

### **Subsystem Services:**

- A description of services offered by each subsystem in the entire system
- A description of operations within each service, including the class to which each operation belongs
- Services provided by each subsystem must be depicted using the ball-and-socket notation in UML component diagrams

### **Class interfaces:**

- A description of each class involved in providing operations for each service
- UML class diagrams must be provided for each of these classes
- Each class must specify its attributes and the operations involved in the service
  - attributes and operations must indicate visibility
  - attributes must indicate data type
  - each operation must indicate return type and parameters with their types

## **Grading**

### **Breakdown:**

Introduction	5%
Subsystem decomposition:	40%
Design strategies:	20%
Subsystem services:	15%
Class interfaces:	20%

### **Criteria:**

- Completeness: all the material is there, as compared to other teams; completeness covers both breadth and depth; e.g. it is insufficient to only present high-level subsystems, they must be accompanied by detailed ones
- Correctness: all the material is presented accurately, with appropriate diagrams and in the appropriate format; superfluous, unnecessary material is not considered to be correct
- Traceability: every part of every model (subsystem, service, class, etc.) should be traced to each other, using a numbering scheme set up in the Deliverable #1 requirements

## **Format**

The content of this deliverable will be discussed in class. A soft copy of the document, in PDF format, must be submitted on [cuLearn](#) on or before Monday, November 23 at 2:30 PM sharp. The submitted copy must be typed and legible, and it must look as professional as if it was being submitted to real client. All diagrams and tables must be introduced and explained in the text. Documents that do not conform to these specifications will not be marked.