CSCI 306 Final Assignment - Gateway

200 Points

BE SURE TO READ THE INSTRUCTIONS BEFORE YOU BEGIN! These questions will not make sense unless you have read that document, as it contains additional requirements. Your answers should be all in a single document, either rtf, pdf or docx. Text for assignment must be in text readable format (i.e. do not submit text as graphics).

**General points for design, clarity, completeness, grammar, spelling, English, etc. – 20 pts**

1. Provide a brief explanation of the system you are modeling and its components. What are the main elements of the architecture and how do they work together? (20pts)

I decided to model the communication system model, a computer simulation of the Lunar Orbital Platform-Gateway (LOP-G) space station's communication subsystem. Communication between the space station and other entities, including Earth, other spacecraft, and lunar rovers, is made possible via this subsystem. Antennas, transceivers, modems, and protocols are among the elements of the communication system model.[[1]](#footnote-1)

The following are some of the main components.

CommunicationSubsystem Class: This is the main class that represents the communication subsystem. It contains objects of other classes such as Antenna, Transceiver, Modem, and Protocol.

Antenna Class: This class represents the antennas and contains methods for transmitting and receiving signals.

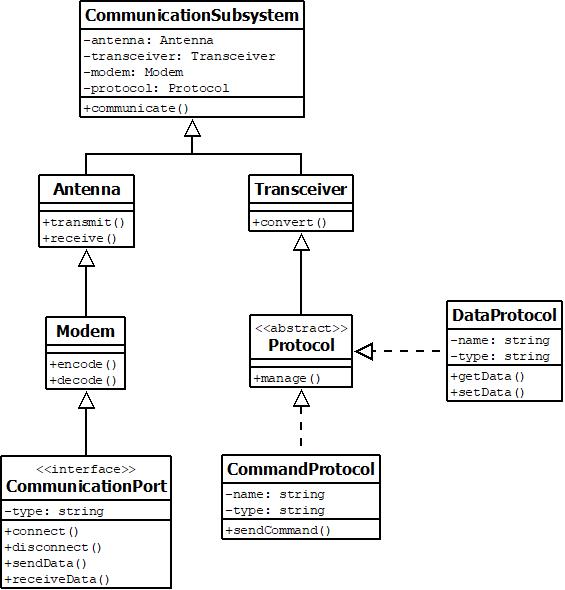
Transceiver Class: This class represents the transceivers and contains methods for converting signals between different frequencies and formats.

Modem Class: This class represents the modems and contains methods for encoding and decoding digital signals for transmission and reception.

Protocol Class: This class represents the communication protocols and contains methods for managing data transmission and reception between the space station and other entities.

The communication system model architecture works by using antennas to transmit and receive signals to and from other entities. To make sure that the signals are compatible with the communication protocols used by the space station and other entities, the transceivers transform the signals between various frequencies and formats. The digital signals are encoded and decoded by the modems for transmission and reception. To maintain dependable and effective communication, the protocols control data transmission and reception between the space station and other organizations.

2. Insert your UML diagram of your design below. Be sure that the diagram is readable! Break it up into a couple of pieces if needed. Because this is take-home, links and format of UML need to be correct. The following answers need to be consistent with the UML design. (40pts)



3. What are your abstract class(es)? Choose one and explain *why* your abstract class should be abstract, and why you need child classes. (20pts)

Here the abstract class is Protocol which is inherited by DataProtocol and CommandProtocol. The reason Protocol is an abstract class is because it provides a common interface and implementation for its subclasses, it’s not useful to create objects of type Protocol on its own. Instead, Protocol is meant to be extended by its child classes, which provide more specific functionality. Using an abstract class allows for code reuse and ensures that objects are created with a consistent set of methods and behaviors, while still allowing for customization and specialization in the child classes.

4. What are your interface(s)? Choose one and justify *why* it makes sense to include that interface. Do NOT just grab a function from one of your classes and throw it into an interface. Your design and explanation should clearly show when an interface is appropriate compared to an abstract class. (20pts)

Here CommunicationPort is the interface. It makes sense to include this interface because it defines a set of methods that any communication port should implement to connect to and communicate over a network. We can guarantee that different kinds of communication ports can be used interchangeably in the CommunicationSubsystem as long as they implement the identical methods provided in the CommunicationPort interface by defining the interface independently of the classes that implement it.

5. Explain how your design adheres to the Open-Closed Principle (OCP). **Be specific**. (20pts)

The OCP adheres to this model in several ways. The use of abstract classes and interfaces allows for new functionality to be added through inheritance rather than modification. For example, if a new type of communication protocol needed to be added, a new subclass of Protocol could be created without modifying the existing CommunicationSubsystem code. The CommunicationSubsystem class is designed to be modular and extensible. Each component of the communication system is built as a different class, and they all communicate using the CommunicationPort interface. This makes it possible to introduce new subsystems without changing old code, provided that they adhere to the CommunicationPort interface. The CommunicationSubsystem class uses dependency injection to provide flexibility and extensibility. Each subsystem is created and injected into the CommunicationSubsystem object at runtime, allowing for different combinations of subsystems to be used without modifying the CommunicationSubsystem code.

6. Explain how your design adheres to the Interface Segregation Principle (ISP). **Be specific** (e.g., you might describe a naïve version of your design that would violate ISP, to show that yours does not violate ISP). (20pts)

The ISP adheres to this model in several ways. The CommunicationPort interface is designed to be specific to the needs of the communication subsystem, and it only includes methods that are relevant to the task of communicating over a network. By doing this, clients that use the CommunicationSubsystem class are not forced to depend on methods they do not use. The Protocol abstract class provides a common implementation for its subclasses, but it does not include any unnecessary methods that would violate the ISP. The use of interfaces and abstract classes allows for the creation of specific interfaces that are tailored to the needs of the clients that use them. Examples include methods in the DataProtocol interface that are special to the needs of the DataSubsystem and methods in the CommandProtocol interface that are specific to the needs of the CommandSubsystem.

7. Explain how someone might extend your program in a way that violates Liskov Substitution Principle (LSP). **Be specific** (i.e., give a concrete example). (20pts)

In this design, one way that LSP could be violated is If a subclass of CommunicationSubsystem does not implement all the methods of its parent class or alters the behavior of its parent class in a way that violates the parent class's assumptions. When the WirelessCommunicationSubsystem is used in place of the CommunicationSubsystem, the program may behave incorrectly. For instance, if someone were to create a WirelessCommunicationSubsystem subclass that does not implement the sendData method of the CommunicationSubsystem or if it modifies the behavior of the sendData method in a way that violates the assumptions of the parent class.

8. Include code for one failing JUnit test method. Include a brief comment about what you are testing. Remember that TDD is a design process. This method should test some planned functionality (e.g., test that a car slows down if another car’s brake lights are showing), and NOT just a setter/getter. (20pts)

public class TestCommunicationSubsystem {

*@*Test

public void testSendData() {

CommunicationSubsystem commSubsystem = new CommunicationSubsystem();

byte[] data = new byte[] {0x01, 0x02, 0x03};

commSubsystem.sendData(data);

byte[] receivedData = commSubsystem.receiveData();

assertArrayEquals(data, receivedData); // this test should pass

}

}

This test is checking whether the sendData and receiveData methods of the CommunicationSubsystem class are working correctly. It sends some data using the sendData method and then receives it using the receiveData method. Finally, it uses the assertArrayEquals function to determine whether the received data matches the original data. Since the sendData and receivedata methods of the CommunicationSubsystem class are currently only placeholders and unimplemented, this test will fail. Therefore, the receiveData method will return null, and the test will fail with a NullPointerException.

1. <https://encyclopedia.pub/entry/31274> [↑](#footnote-ref-1)