Communicating and Thinking about Systems

1 Communicating about Systems

Human communication is the process of conveying an experience that can only be had directly through the consciousness from one person to another. When a person transmits information verbally, they project a complex and ineffable experience into a sequence of symbols. We interpret verbal information by reading at approximately 400 words per minute $\approx$ 40 characters per second $\approx$ 200 bits per second. If you look at what we retain from this, it turns out to be only a small number of bits per second[1]. What makes communication possible is that these streams of information are highly compressed. The person receiving the information associates the symbols with experiences that can be very complex. People who have a great deal of common experience can communicate much more efficiently about those common experiences. However, it is often the case that one person must convey an experience to another that has no direct knowledge of it. This is more challenging but can be achieved using metaphors, which serve as a relatable approximations of the notions to be transmitted. The distance between the metaphor and the subject being described can vary but is generally nonzero. Only tautological statements enjoy the property of having infinite precision. Everything else is an approximation to actuality, but approximations are extremely useful. History demonstrates that scientific laws that are held true at any point in time are often later shown to be approximations to deeper laws. Approximations that have enough precision to capture what is needed to accomplish the engineering tasks at hand are what we care about as engineers.

As a systems engineer, there are two areas where this issue will have a marked impact on you. The first is in dealing with other human beings. Your role is that of a conductor, interacting with customers, people from the business unit and engineers of all disciplines. Being aware of the level of abstraction is very important. A customer might not have an understanding of the technology and it will be your job to extract from what may mainly be a connotational communication, the appropriate message to give to the engineers. The other area where this will have a profound impact on you is that since you will be working at a high level with complex systems, you will need to work with approximate representations of engineering systems. It is not that precise descriptions of such systems are impossible, but that for certain tasks, it makes sense to use ideas that are course in granularity and omit certain details. Precision is verbose and this can make a model more difficult to create, read or modify.

1.1 Example Abstractions

Object-Oriented Modeling One ubiquitous systems model that was made popular through use in software is that of object-orientation. However, object-orientation has its limitations as a model for objects. Consider the example shown in Figure 1.

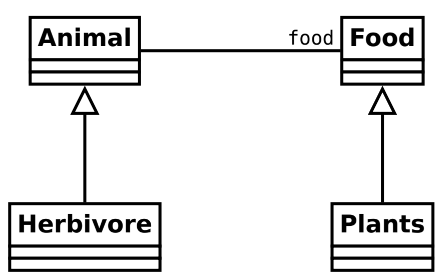

Figure 1. A simple object-oriented model. The hollow arrowhead indicates that the class at the source of the arrow, Herbivore, is a subclass of the destination of the arrow, Animal. The line connecting the Animal class to the Food class is an association. It indicates a relationship between the two classes.

[1] How many conversations or things that you read do you remember vebatim?

One important principle in object-oriented design and analysis is that of *substitutability*.

Liskov Substitution Principle: If an object S is a subtype of object T, then anywhere an object of type T is expected, an object of type S may be substituted transparently to the execution of the program.

The model in Figure 1 violates Liskov Substitution because while herbivores are animals, they may only eat plants. If an herbivore is substituted for an animal, it could be the case that the animal is assigned a non-plant food object. This is a commonly occurring problem in object-oriented systems with no known general solution. Such violations are often tolerated in practice because of the pragmatic nature of engineering. Abstractions work and are useful most of the time. It is simple to work around a problem like the above in code, but it is necessary to be aware of the when substitutability does and does not apply.

Unintended Consequences One way to view the performance of a company is by its short term profits. This is a useful abstraction and gives a way to focus efforts to improve the condition of a company. A possible decision that one could make in the interest of improving profits is downsizing. This could immediately improve profitability, but it may harm long term profitability by reducing productivity.

2 Exercises

1. Describe five abstractions. If they discard details, indicate the most important details that they discard.

2. Describe using a maximum of 300 words, as much as possible about one organ of the human body. You should describe the relationship between the organ and the rest of the body. Take the role of an engineer that is going to build such a system for a customer. Try to include the information given below.

   Use Cases: List the uses of the system that might interest the customer.

   Structure: Describe the most important structures of the system.

   Behavior: Describe the most important behaviors of the system.

   Behavior and Structure: Describe how the structure is linked to the behavior. This might already be done in the description of behaviors.

   Tradeoffs: For at least one feature of the system, describe either an alternative design or a range of variability. Discuss the pros and cons of the current design.

3. If the customer were to take your description at face value, where might he be misled if he was not aware of its level of granularity?