Lecture 14 - The OO Paradigm - Cohesion and Coupling

Collin Roberts

October 31, 2023

Outline

Remarks

1. Chapter 7 does a poor job of explaining the object-oriented paradigm. So we will use more industry-standard definitions than the text does. Use our definitions instead of the text's, where they disagree.
2. All of Chapter 7 is "white-box", not "black-box". We cannot assess cohesion / coupling unless we can see all of the code.
3. However, soon (i.e. after discussing **encapsulation** and **information hiding**), we will see that it is a best practice **not** to expose all of the code to the outside world.

Key Quotation

When a large S/W product consists of a single monolithic block of code, maintenance is a nightmare.

A working definition for us

Definition 1

A **module** *is a lexically contiguous sequence of program statements, bounded by boundary elements and having an aggregate identifier.*

Remarks About Definition 1

1. Every function/procedure of the classical paradigm is a module.
2. In the OO paradigm, every class and every method within a class is a module.
    1. The main idea of OO is to keep data, and operations on that data, together.
    2. We need to be clear about the difference between the program statements that define the properties (a.k.a. attributes) of a class, and some **instantiation** of that class. Only an instantiation of a class can actually contain data.

What is a Module?

Definition 2

C/SD *is an acronym for* **composite/structured design**.

Remarks

1. The aim of C/SD is to apply common sense to make S/W product designs "make sense". (E.g. see Figures 7.1 to 7.3 in the text for designs that do, and do not, make sense.)
2. C/SD done well achieves **separation of concerns**.

Cohesion (§7.2)

Definition 3

Cohesion *of a module is the degree of interaction within that module.*

Remarks

1. The text defines many levels of cohesion. Do **not** memorize these!
2. For us, it will be enough to distinguish between **high** and **low** cohesion. high = good; low = bad.

Coupling (§7.3)

Definition 4

Coupling *of a pair of modules is the degree of interaction between the two modules.*

Remarks

1. The text defines many levels of coupling. Do **not** memorize these!
2. For us, it will be enough to distinguish between **loose** and **tight** coupling. loose = good; tight = bad.

Remarks on Assessing Cohesion and Coupling

1. Suppose that we are given two pairs of modules and it is our job to assess which pair's modules have
   1. high versus low cohesion, and
   2. loose versus tight coupling.
2. Because we have dispensed with the detailed levels of cohesion and coupling from the textbook, therefore making both judgments is **relative**, **not absolute**.

Remarks on Assessing Cohesion and Coupling

3. We can decide
    1. which pair's modules have higher cohesion than the modules of the other pair, and
    2. which pair's modules have looser coupling than the modules of the other pair.
4. In past offerings of CS 430, cohesion and coupling has caused some confusion. Keep our definitions, plus the above remarks in mind, and work out your comparisons carefully.

Cohesion / Coupling Example

Refer to the Examples document.

Results:

1. low cohesion, tight coupling (bad)
2. high cohesion, loose coupling (good)

Why Coupling is Important

1. Tight coupling means a higher probability of regression faults.
2. Suppose modules p and q are tightly coupled.
3. Then it is likely that making a change to p requires a change to q.
4. Making the change to q adds time, and hence cost, to the project (which would not be required with looser coupling).
5. Not making the change to q likely causes a fault later on.

Why Coupling is Important

6. The stronger the coupling with some other module, the more fault-prone a module is.
7. This in turn makes the module the more difficult and costly to maintain.
8. As mentioned above, our goal is high cohesion and loose coupling. The rest of Ch7 is about refining the techniques to achieve this goal. Ch14 of the text goes into more detail; unfortunately this will be beyond the scope of CS 430.
9. Also note that separation of concerns (in general terms) means high cohesion and loose coupling (in OO terms).