

MEASURING STRUCTURED DESIGN QUALITY

Didi Rosiyadi¹, Nana Suryana²

Information System Field, Research Center for Informatics,
Indonesian Institute of Sciences, Jl. Cicitu Bandung
E-mail: ¹rosiyadi@informatika.lipi.go.id, ²surya@informatika.lipi.go.id

ABSTRACT

Structured Design is the development of a blueprint of a computer system solution to a problem that has the same components and interrelationships among the components as the original problem. Structured Design offers a set of strategies for developing a design solution from a well-defined statement of a problem and offers a set of objective and empirically justified criteria for evaluating the quality of a given design solution with respect to the problem to be solved. Structured design is not the structure chart any more than good sculpture is a chisel. The chisel is a tool for chipping that must be used with dexterity to achieve a good result. Similarly, a structure chart is simply a tool for showing a picture of the modules in a system and their relationships to one another. One of the fundamental principles of structured design is that a large system should be partitioned into manageable modules. However, it is vital that this partitioning should be carried out in such a way that the modules are as independent as possible, this is the criterion of coupling.

Keywords: Design, Structured Design, Coupling

1. Description of Coupling

One way of measuring structured design quality is coupling, the degree of interdependence between two modules. Our objective is to minimize coupling, that is, to make modules as independent as possible. Low coupling between modules indicates a well partitioned system and can be attained in one of the three ways are by eliminating unnecessary relationships, by reducing the number of necessary relationships and by easing the tightness of necessary relationships.

2. The Principles of Coupling

Reducing the coupling between modules means, in effect, reducing the complexity of the connection(s) between the modules as much as possible. To paraphrase Einstein, we seek to meek such connections as simple as the application will allow, and no simpler. The principles by which to reduce coupling are the following:

- Create narrow (as opposed to broad) connections.
- Create direct (as opposed to indirect) connections.
- Create local (as opposed to remote) connections.
- Create obvious (as opposed to obscure) connections.
- Create flexible (as opposed to rigid) connections.

3. The Categories of Coupling

The categories of coupling are:

- Normal Coupling, Two modules, A and B, are normally coupled if (1) A calls B, (2) B returns to A, and (3) all information passed between them is by means of parameters presented with the call itself. This, of course, is the situation depicted on a normal structure chart, as shown in Figure 1 and 2.

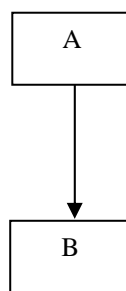


Figure 1

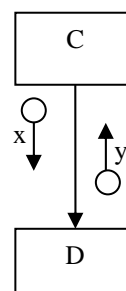


Figure 2

Figure 1: A and B Normally Coupled but having Nothing to say to each other.

Figure 2: C and D Normally coupled and communicating Data X and Y.

The categories of Normal coupling are:

a.1 Data Coupling is two modules are data coupled if they communicate by parameters, each parameter being an elementary piece of data. Data coupling is the necessary communication of data between modules. Since modules must communicate, data coupling is unavoidable and is quite harmless as long as its kept to a minimum. For example in figure 3:

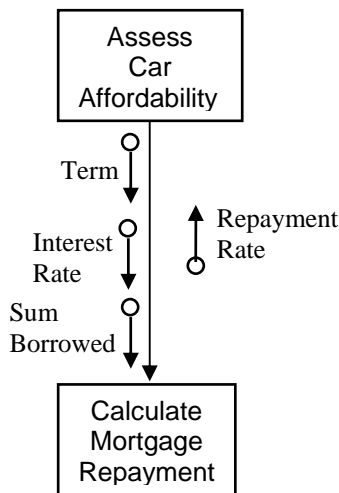


Figure 3. Data Coupling

a.2 Stamp Coupling, two normally coupled modules are stamp coupled if one passes to the other a composite piece of data, that is, a piece of data with meaningful internal structure. Example in Figure 4:

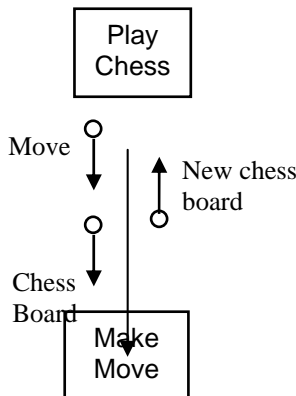


Figure 4. Stamp Coupling

a.3 Control Coupling, Two modules are control coupled if one passes to the other a piece of information intended to control the internal logic of the other. Figure 5 shows two control-coupled modules.

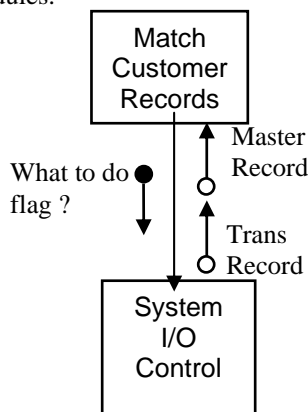


Figure 5. Control Coupling

b. Common Coupling, two modules are common coupled if they refer to the same global data area. In figure 6 FIND PART NAME and REMOVE STOCK are common coupled because they both refer to the same global area, which contains (probably among other data) PARTS TABLE and ERROR FLAGS.

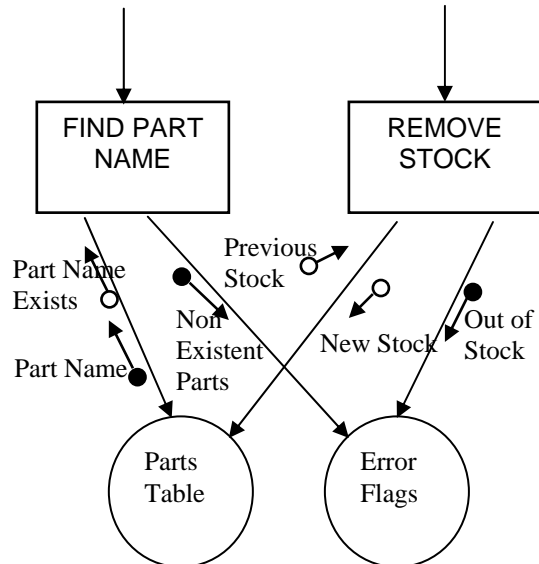


Figure 6. Common Coupling

c. Content Coupling, two modules are content coupled if:

- one module changes a statement in another (Lisp was famous for this ability)
- one module references or alters data contained inside another module
- one module branches into another module

Examples

- Type 2 can be done in, say, Pascal by using the language scope rules to do non-local referencing... read and write the values of variable in one procedure from another one (assuming a procedure is considered a module)
- A worse example of type 2 is if you use a language like Simula in which you can create objects (data and procedures to operate on that data, in a module) but then you violate the object boundaries by directly reading and writing the variable in the object. Later OO languages (C++, Eiffel) prevent this, but earlier ones do not.
- Type3 can be done in Pascal... branch from one procedure to a label in another... using scope rules.

4. Determining Coupling Type

Two modules may be coupled in more than one way. In that case, their coupling is defined by the worst coupling type they exhibit. For example, if two modules are stamp and common coupled, they are characterized as common coupled. They are still

stamp coupled, but that is outweighed by their common coupling.

A good rule of thumb for designing the way a module gets and returns its information is to imagine it as a library module. How would this module be easiest to understand? How would this module be most usable to other people in the shop?

Another way in which the coupling of a design can be evaluated is to suppose that each module will be coded by a different programmer. How independently can the programmers work? What may change that may cause changes to multiple modules? Is there any way in which the effects of the change could be isolated to one module? Answering these questions will determine which user changes are likely to require modifications to a large number of modules.

5. Summary

Coupling is the measure of the interdependence between two modules. In a good design, coupling is kept to a minimum by: eliminating unnecessary relationships between modules; reducing the number of necessary relationships; and easing the tightness of necessary relationships.

In sequence from best to worst, there are three broad categories of coupling between modules: normal coupling (coupling by parameters); global or common coupling (coupling via a globally accessible data area); and content or pathological coupling (coupling by direct connection between the internals of modules). Again sequence from best to worst, normal coupling may be: data coupling (coupling by elementary parameters); stamp coupling (coupling by composite parameters); or control coupling (coupling by parameters used explicitly to control the behavior of the recipient module). The most flexible and maintainable designs are typically those with the simplest interfaces between modules: the types of interfaces that you might design for standard library modules.

6. Advise

Expand later research is required of measuring object oriented design and agent topics.

References

- [1] Bennett S, McRobb, Farmer, 2002, *Object Oriented Analysis and Design*, McGraw-Hill International Editions, New York.
- [2] Computer Science Department, *Coupling and Cohesion*, 2005, <http://www.cs.unc.edu/~stotts/145/coupling.htm>
- [3] Dewitz, S. D., 1996, *System Analysis and Design and the Transition to Objects*, McGraw-Hill International Editions, New York.
- [4] Kowal, J. A., 1985, *Analyzing System*, Meilir Page-Jones, Seattle.

- [5] Page-Jones, M., 1980, *The Practical Guide to Structured Systems Design*, Yourdon Press Series, London.
- [6] Rumbaugh J, 1991, *Object Oriented Modeling and Design*, Prentice Hall, New Jersey.
- [7] San Diego State University, 2001, *Module of Coupling and Cohesion*, <http://www.eli.sdsu.edu/courses/spring01/cs635/notes/module/module.html#Heading4>
- [8] Yourdan, E., 1989, *Modern Structured Analysis*, Yourdan Press Series, London.

