

Module Cohesion

Module Cohesion

Module Cohesion

Types of Cohesion

Best

- Functional
- Informational (in sec 7.8, not chap 6)
- Sequential
- Communicational
- Procedural
- Temporal
- Logical
- Coincidental

Worst

Module Cohesion

Functional Cohesion

All parts of the module contribute to just *one function*.
Everything needed for the function is in the module.
Nothing else is in the module

Ex's: Compute Square Root
Sort Array
Compile Source File
Read Next Record

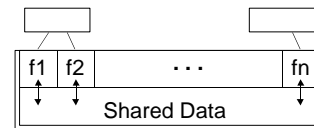
In math:
 $y = f(x)$

Note: a function *must* be complete,
it *may* be simple or complex.

Module Cohesion

Informational Cohesion

Multiple functions on shared data.



Only f_1, \dots, f_n have access to the shared data.
Similar to a class in object oriented programming

Note: f_1, \dots, f_n are functionally cohesive.

Module Cohesion

Example of Informational Cohesion

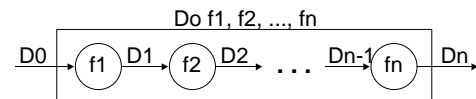
In the Symbol Table Module:

- Initialize Symbol Table
- Find Table Symbol
- Create Table Entry
- Update Table Entry

Module Cohesion

Sequential Cohesion

Multiple functions that relay data sequentially.
Do not constitute a complete function



Module Cohesion

Module Cohesion

Example Sequential Cohesion

Read Valid Record

Sequentially Cohesive Functionally Cohesive

Module Cohesion

Communicational Cohesion

Multiple functions on shared data.

Do f1, f2, ..., fn

Ex: Sort and Plot Array
Find Customer Address and Credit History

Problem: Not General and Reusable

Module Cohesion

Procedural Cohesion

Multiple functions on *different* data in *sequence*.

Do f1, f2, ..., fn

Order is *important*. Functions often weakly connected

Ex: Write Record and Get Next Transaction
Update Screen and Get Next Operation

Problem: Not General and Reusable

Module Cohesion

Temporal Cohesion

Multiple functions done *at same time*.

Do f1, f2, ..., fn

Order is *unimportant*. Functions often weakly connected

Ex: Close Files and Print Summary
Open Files and Initialize Variables

Problem: Not General and Reusable

Module Cohesion

Logical Cohesion

Multiple functions *selected by module user*.

Ex: DoFunction(7, xCoord, yCoord, dummy)
DoFunction(3, dummy, isPressed, dummy)

Meaning of arguments depend on function used.

Ex: Microsoft Mouse Driver (assembly)
AH = function number
BX = depends on function number
CX = depends on function number
DX = depends on function number

Module Cohesion

Logical Cohesion Typical Module Structure

```

if ( function == 1 )
{ ... }
else if ( function == 2 )
{ ... }
...

else if ( function == n )
{ ... }
    
```

Problem: Functions *share code and data*

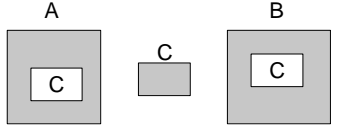
Thus: Functions *highly* intertwined,
Very hard to change.

Module Cohesion

Module Cohesion

Coincidental Cohesion

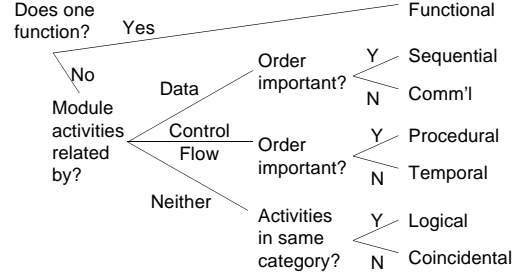
Parts of a module together *merely by coincidence*.
 Ex: To save mem, cut module C from A and B.



Problem: change in A involving C fails in B.

Module Cohesion

Module Cohesion Decision Tree



Module Cohesion

Exercises

AFTIN_1 After input, add control items and verify totals.
Sequential, if the added control items are the totals being verified.

AFTIN_2 After input, write proof tape, add control items and verify totals.
Temporal, since write tape merely done at same *time* as other steps, and *could* be done *before* or *after* them.

GENREPT Produce either a sales report, a project status report, or a customer transaction report.
Logical, since function given by an argument.

Module Cohesion

Exercises

SYNCH Check space vehicle guidance parameters for syntactic correctness.
Functional, since complexity is irrelevant

OUTTRAN Print transaction and copy it to tape.
Communicational (also Temporal), since both apply to *whole module*, the stronger cohesion is used.

UPCREDOUT Update current credit record and write it to disk.
Sequential.

Module Cohesion

Exercises

START_IT Open files, get first transaction and first master records, and print page headings.
Temporal, since the activities are done at *same time*

NEWTRAN Update record to file and get next transaction.
Procedural, since execution *order is important*.

CIRCDISP Make circuit diagram from an electrical connection matrix.
Functional, since complexity is irrelevant.

Module Cohesion

The End

Module Cohesion