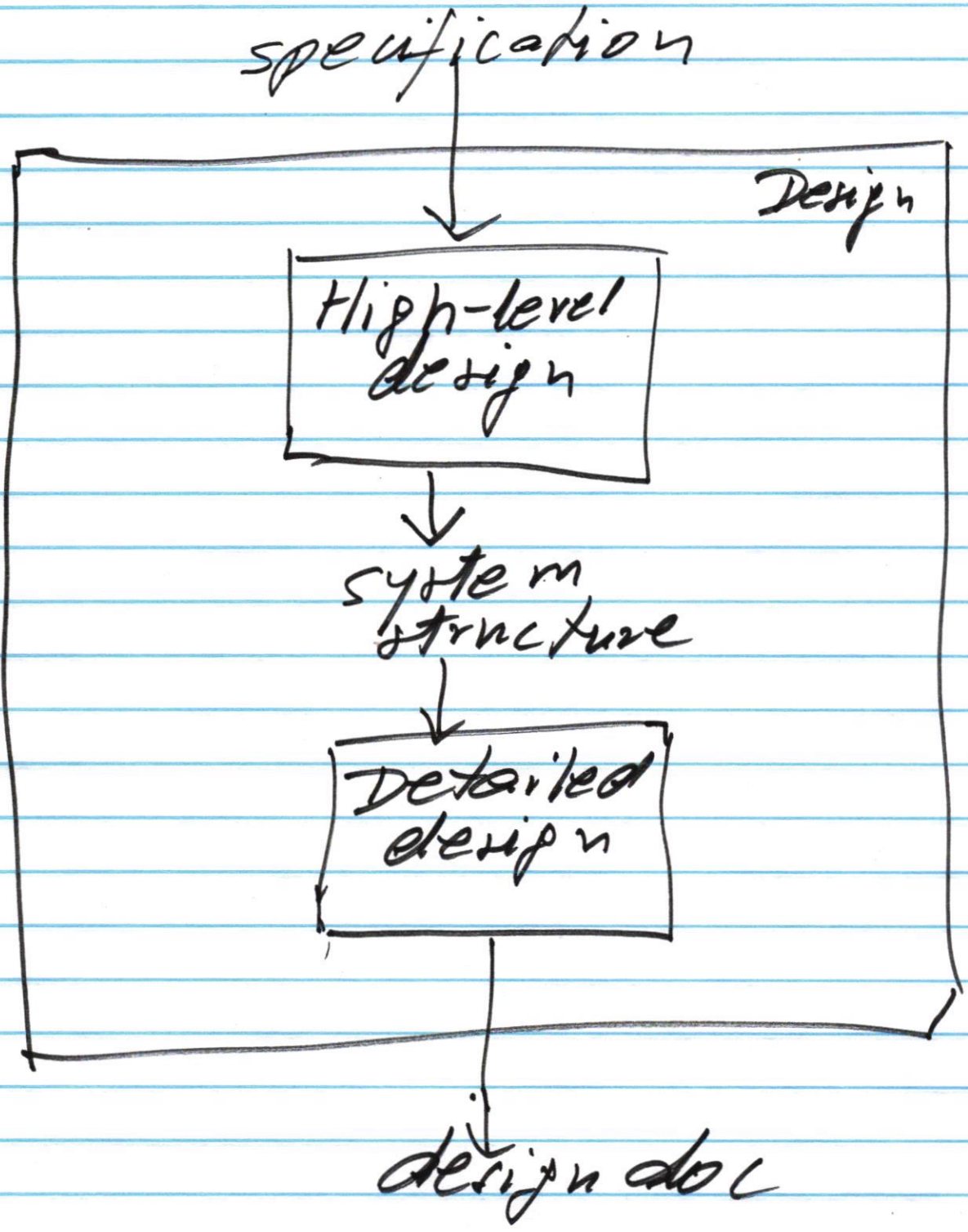
Design


specification
Design
High-level design
system structure
Detailed design
design doc

High level design

system structure

⇓

a set of components

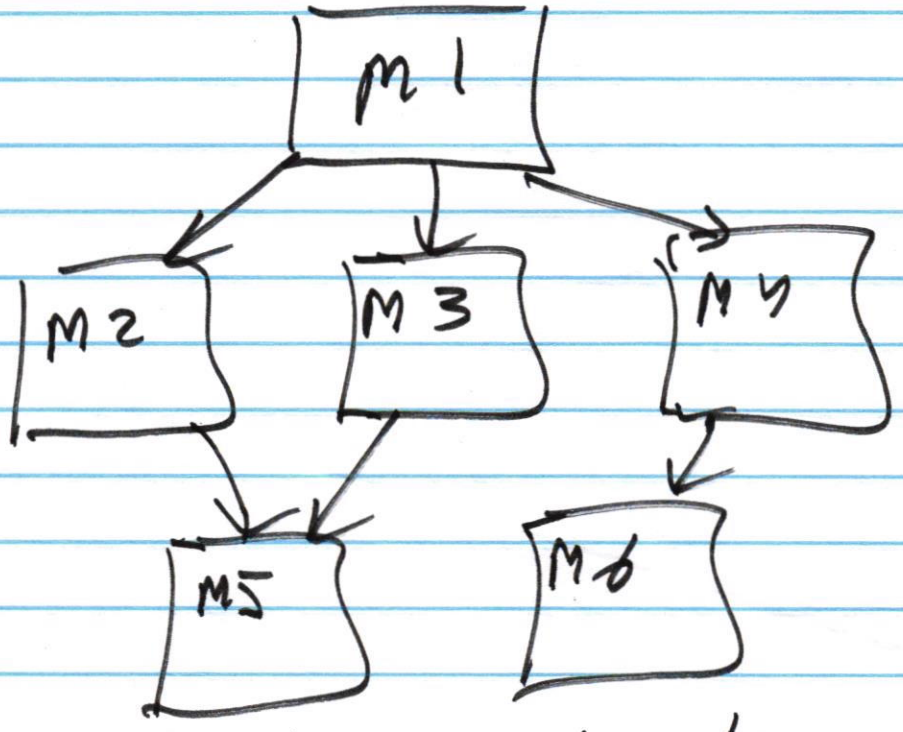
+

relationships between components

modular design

component: module
function/procedure.

relationship: call
relationship.



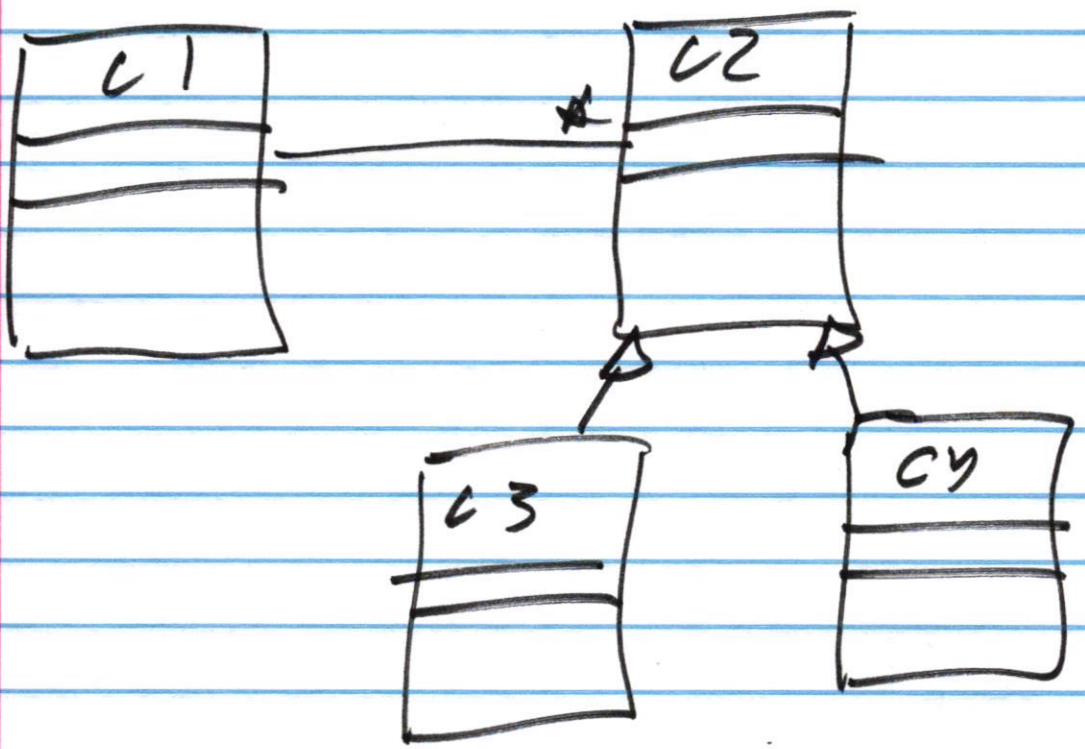
structure chart.

object-oriented design

component: class

relationships: inheritance
aggregation
association

class diagram



Large system

10,000 modules/classes

hard to understand and analyze the design

module/class: low level components

---

software architecture.

⇓

thigher level of abstraction

software architecture

a set of high-level components

+

relationships between them

---

high-level	relationships
Process	* communication
server/client	* protocol
layer	* pipe
filter	* . .
=	
—	
—	

Types of architecture.

* Domain independent architecture.
  * architectural styles
  * " " patterns
  * - - -

* Domain specific architecture.
  (reference architecture)
  exploits specifics of the domain.

Software architectures

* layered architecture
* pipes and filters -"-
* model-driven -"-
* fault-tolerant -"-
* blackboard -"-
* client-server -"-
* architectural patterns

Traditional design methods

* modular design
* object-oriented design

modular design

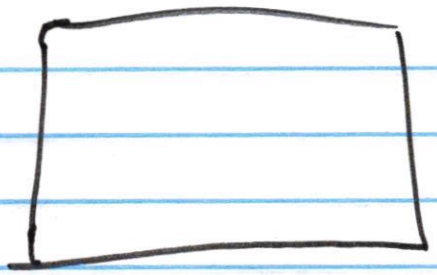
a set of modules

+

call relationships

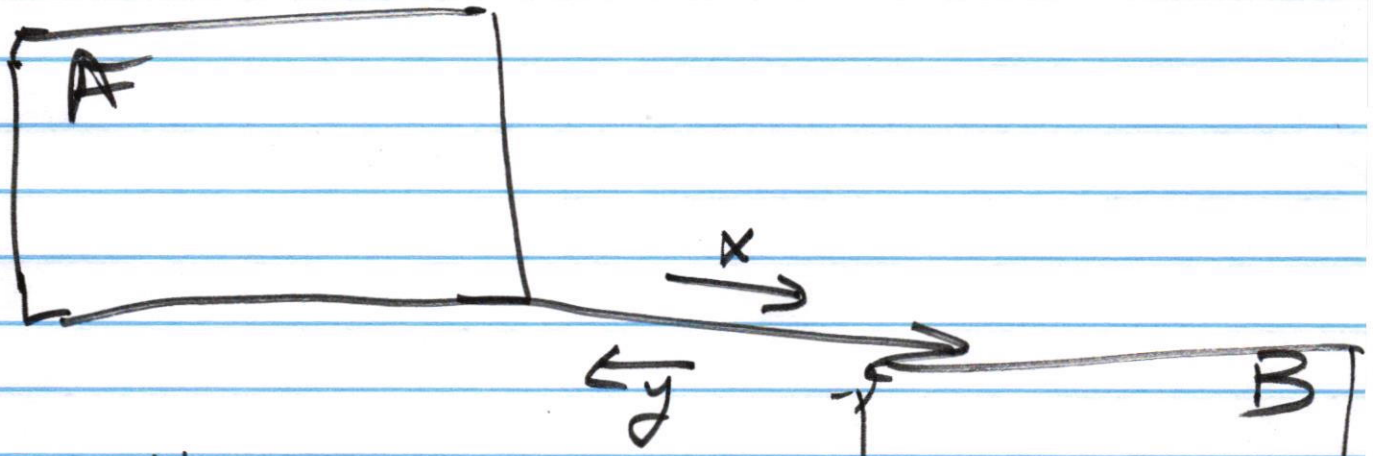
⇓

structure chart



module
(function / procedure)

call relationship.

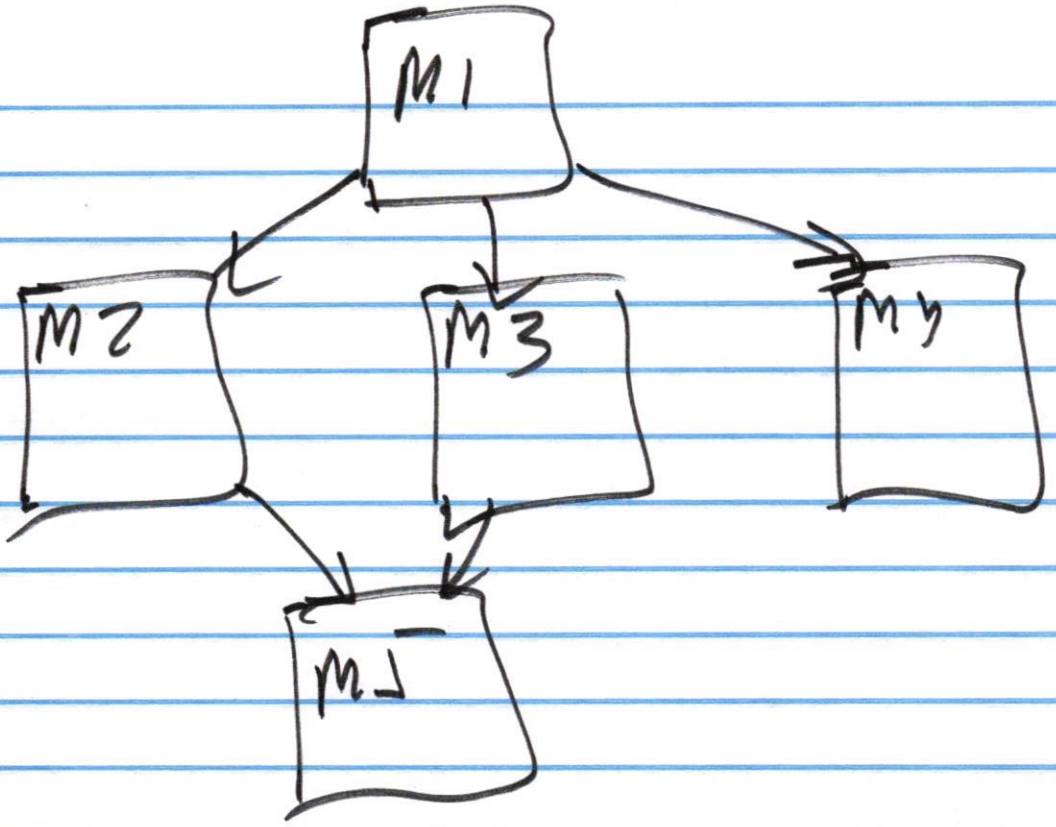


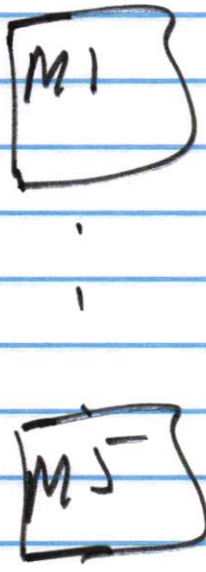
A calls B

A passes data x to B

B uses x to compute y.

B returns result y to A





specification responsibility.

* algorithms
* data structure.

Design evaluation criteria

* coupling.
* cohesion
* information hiding.

---

* maintenance
* performance

maintenance

making changes.

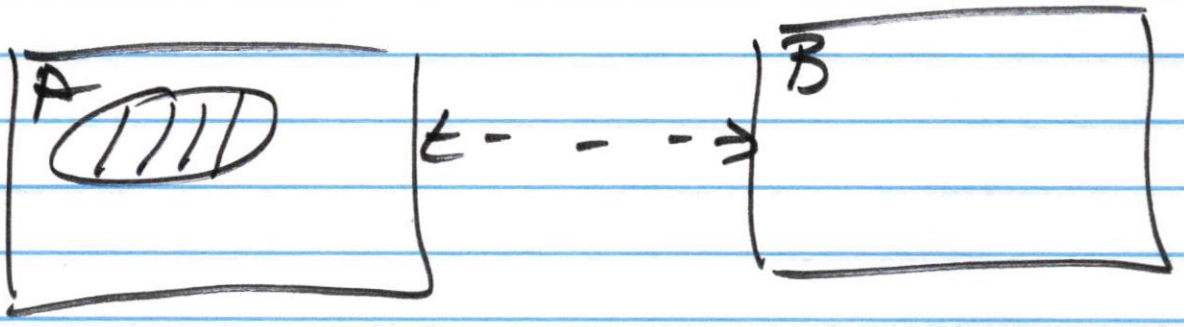
* requirements
* components.
* algorithms
* data structures.
* changing environment
* change hardware
* . .

coupling.

a degree of relationship between two components.

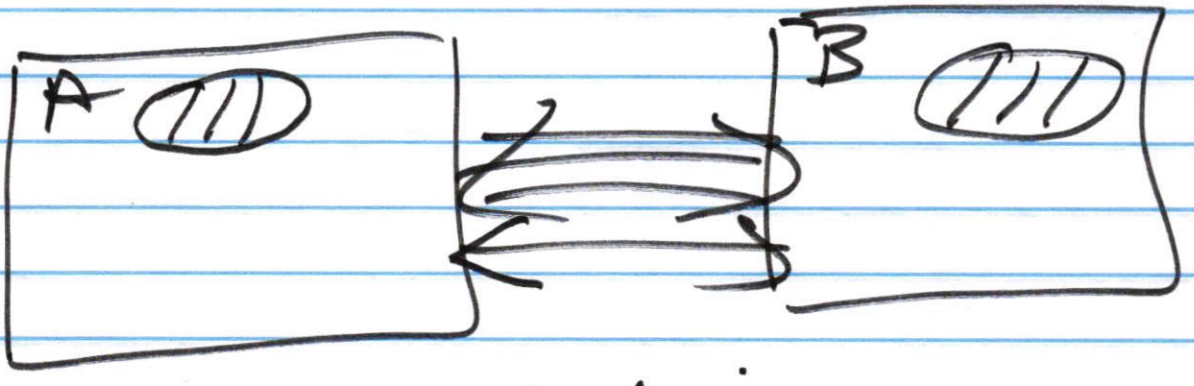
---

weak relationship. — good design



---

strong relationship.



bad design

coupling between modules

- common coupling . worst
- control —"—
- stamp —"—
- data —"—
- no —"—

} best

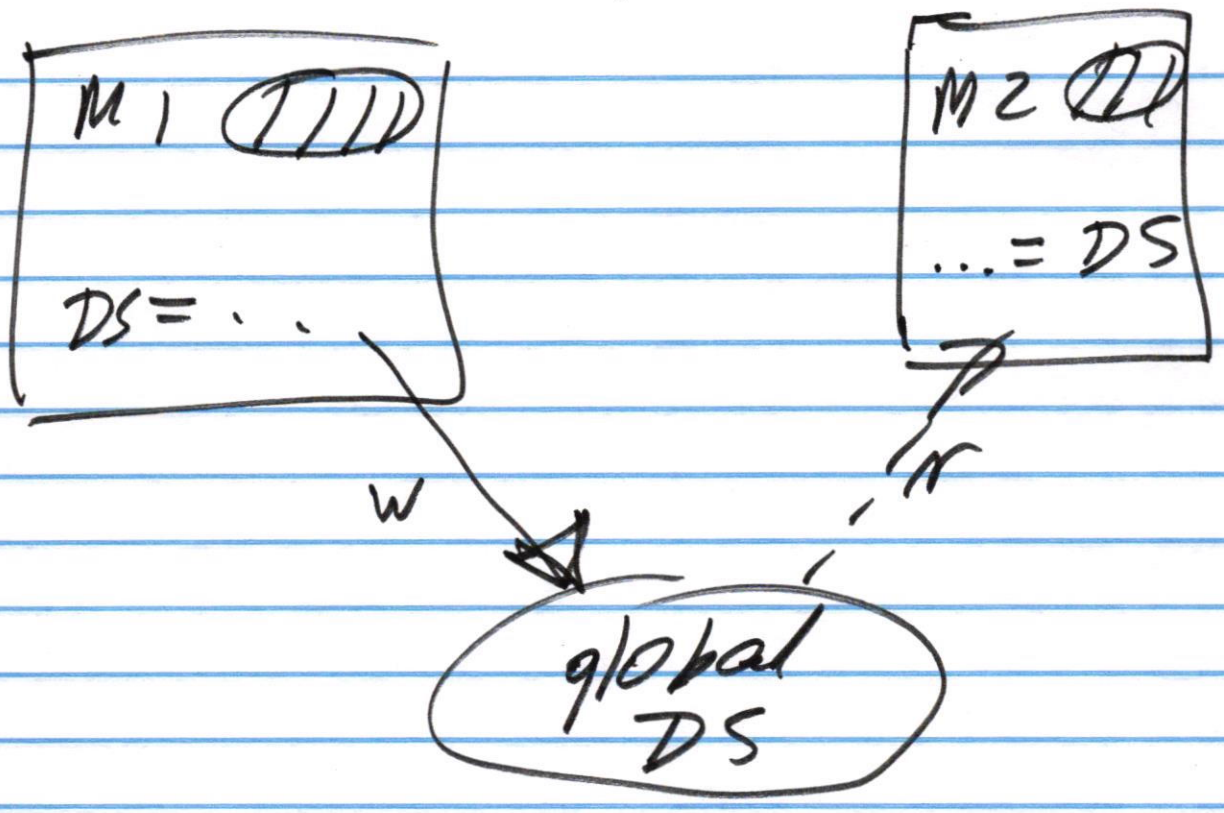
common coupling.

Module M1 and M2.

1. M1 and M2 access the same global data structure. DS

2. M1 writes to DS and M2 reads from DS.

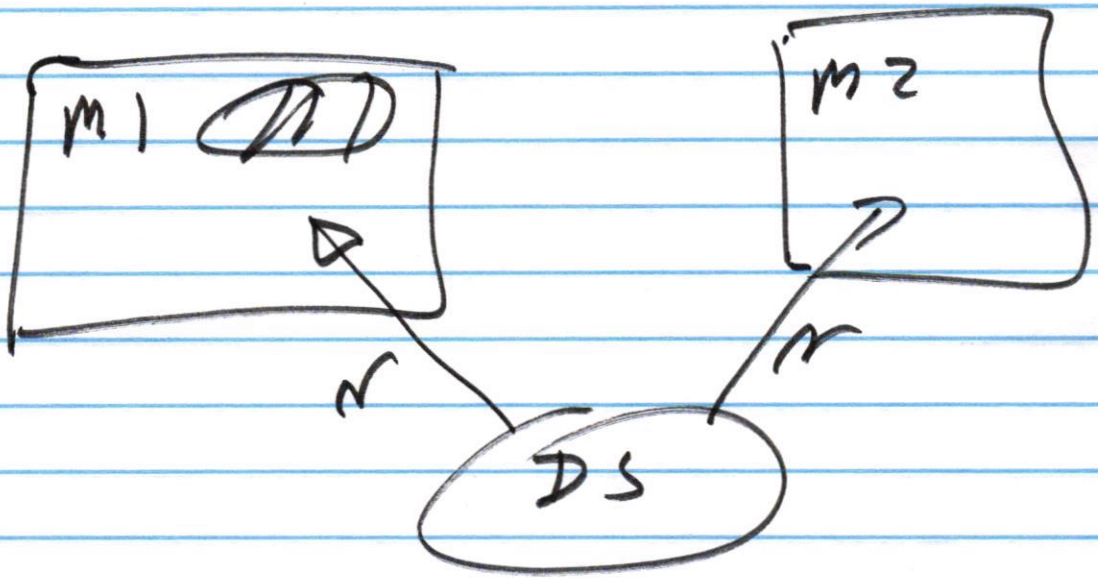
write-read relationship.



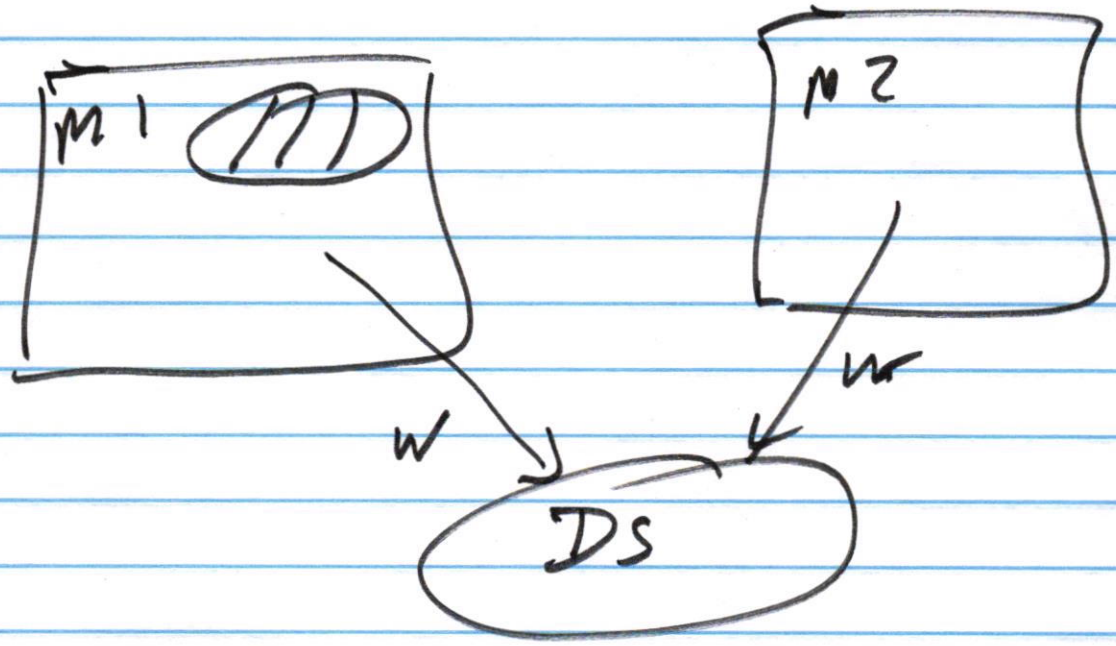
write-read relationship

No common coupling

read - read



write - write

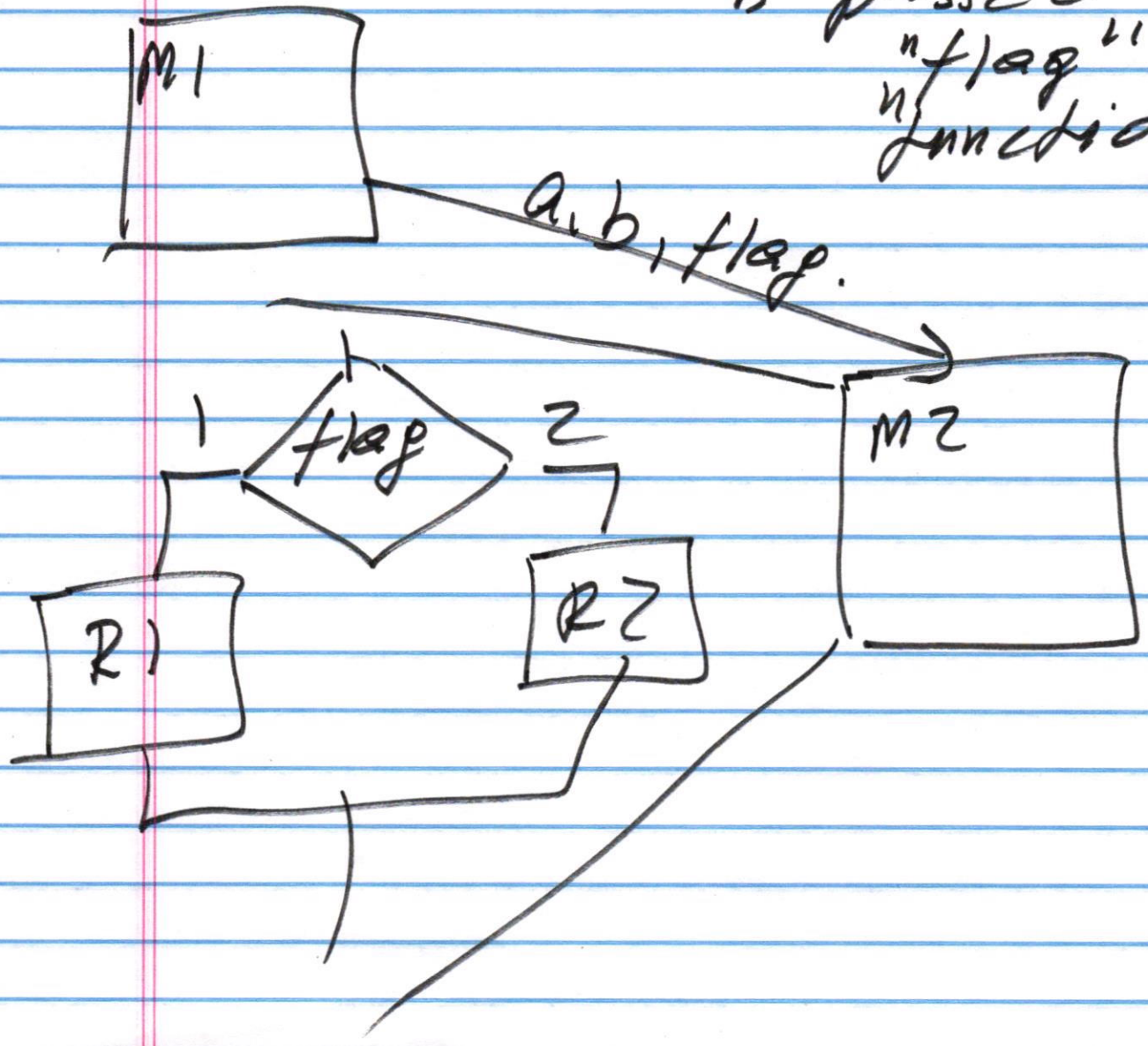


control coupling.

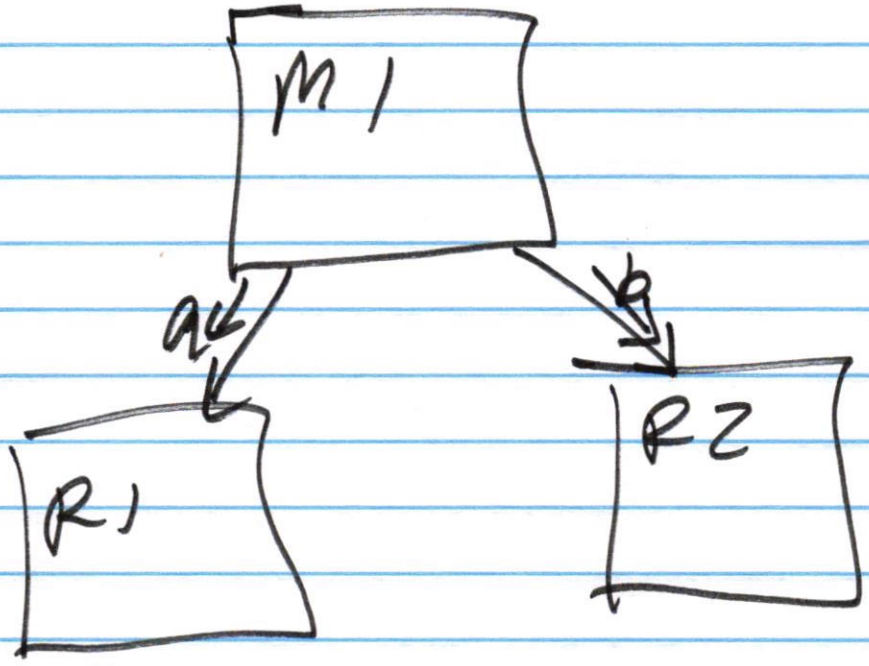
Module M1 and M2.

M1 call M2:

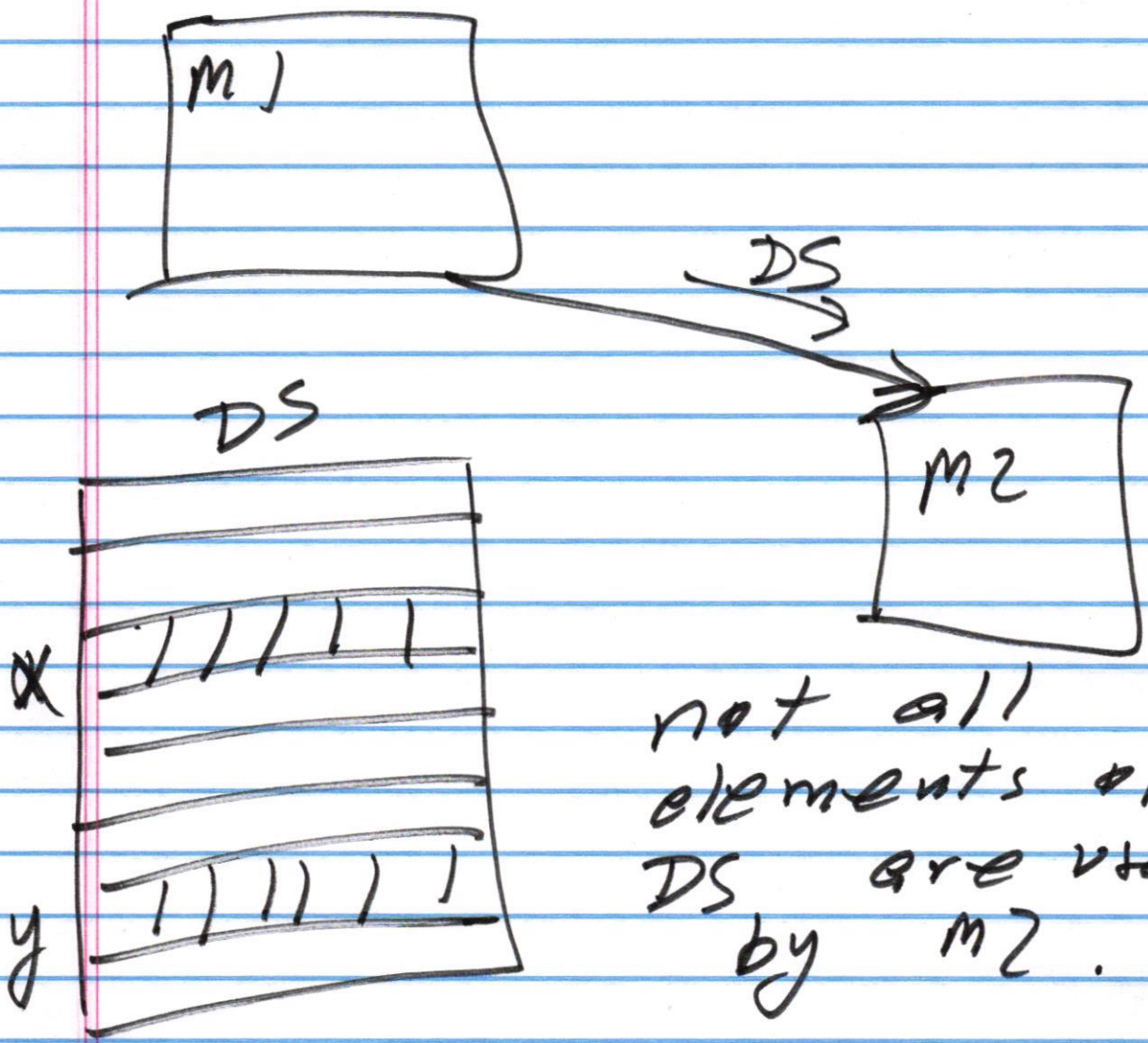
a control "element" is passed "flag" "function code"



better design

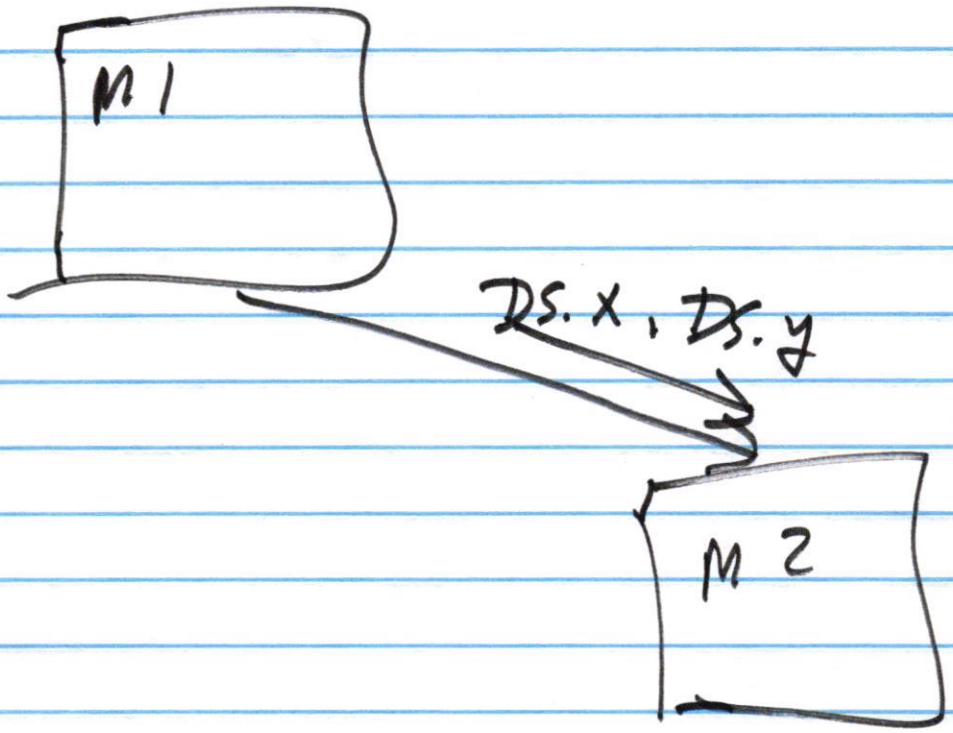


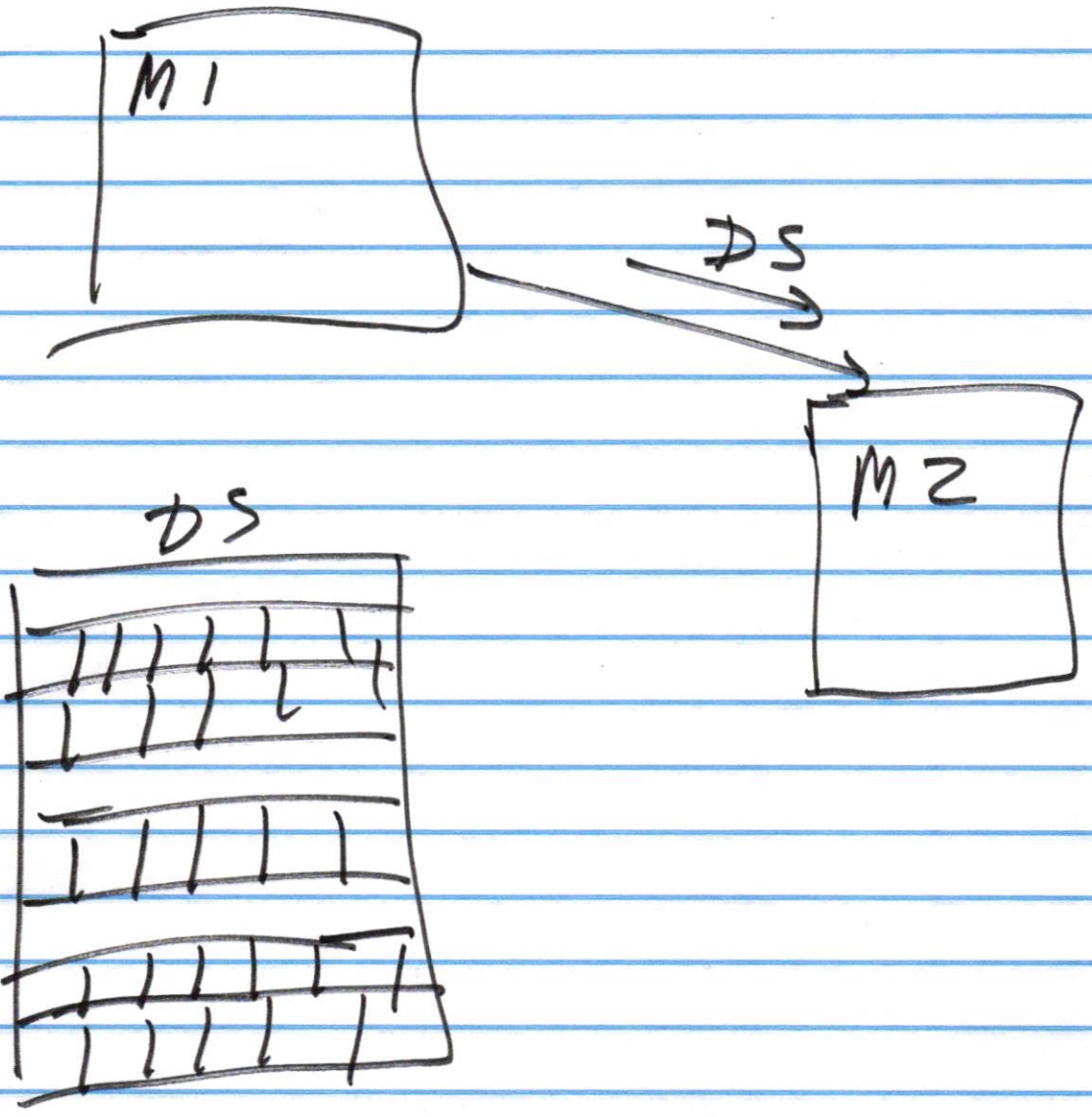
stamp coupling.



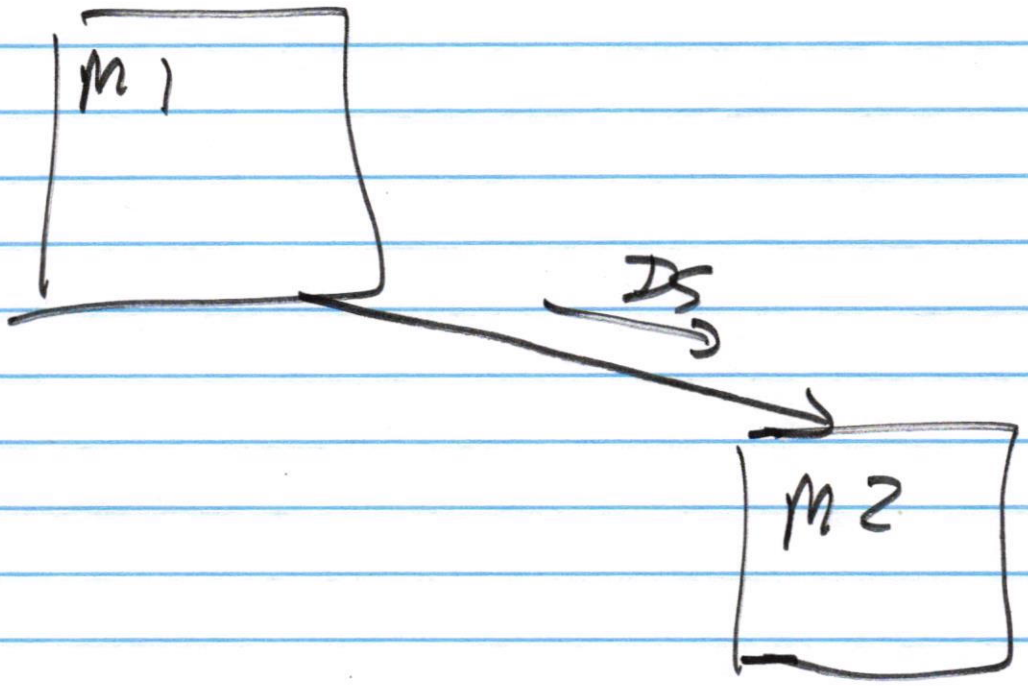
not all elements of DS are used by m2.

better design




M1
DS
M2
DS

Data coupling.



all elements of DS are used by m2

Grader system

R1: read correct answers

R2: read student -ll-

R3: compute scores.

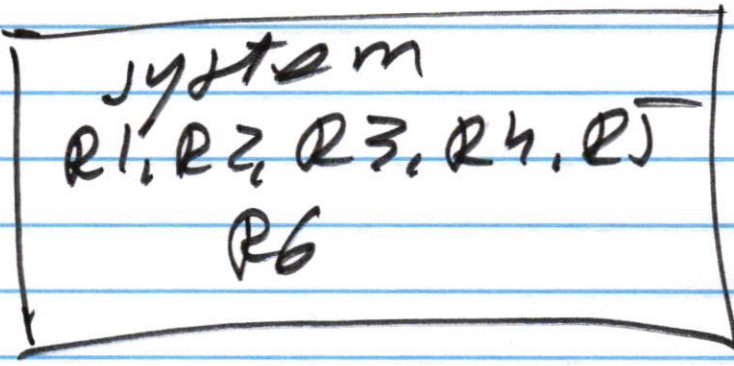
R4: compute statistics (mean)

R5: report test scores

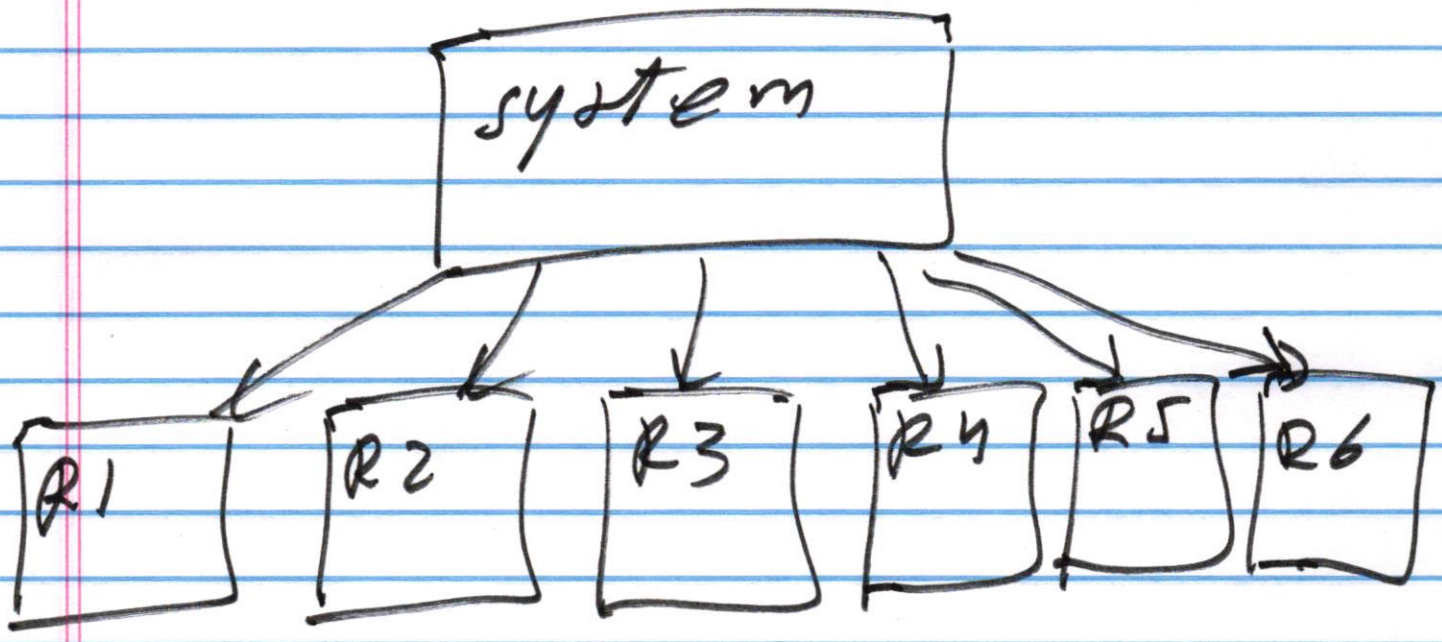
R6: report statistics

Possible design

Design #1



Design #2



data structures

CA: correct answers

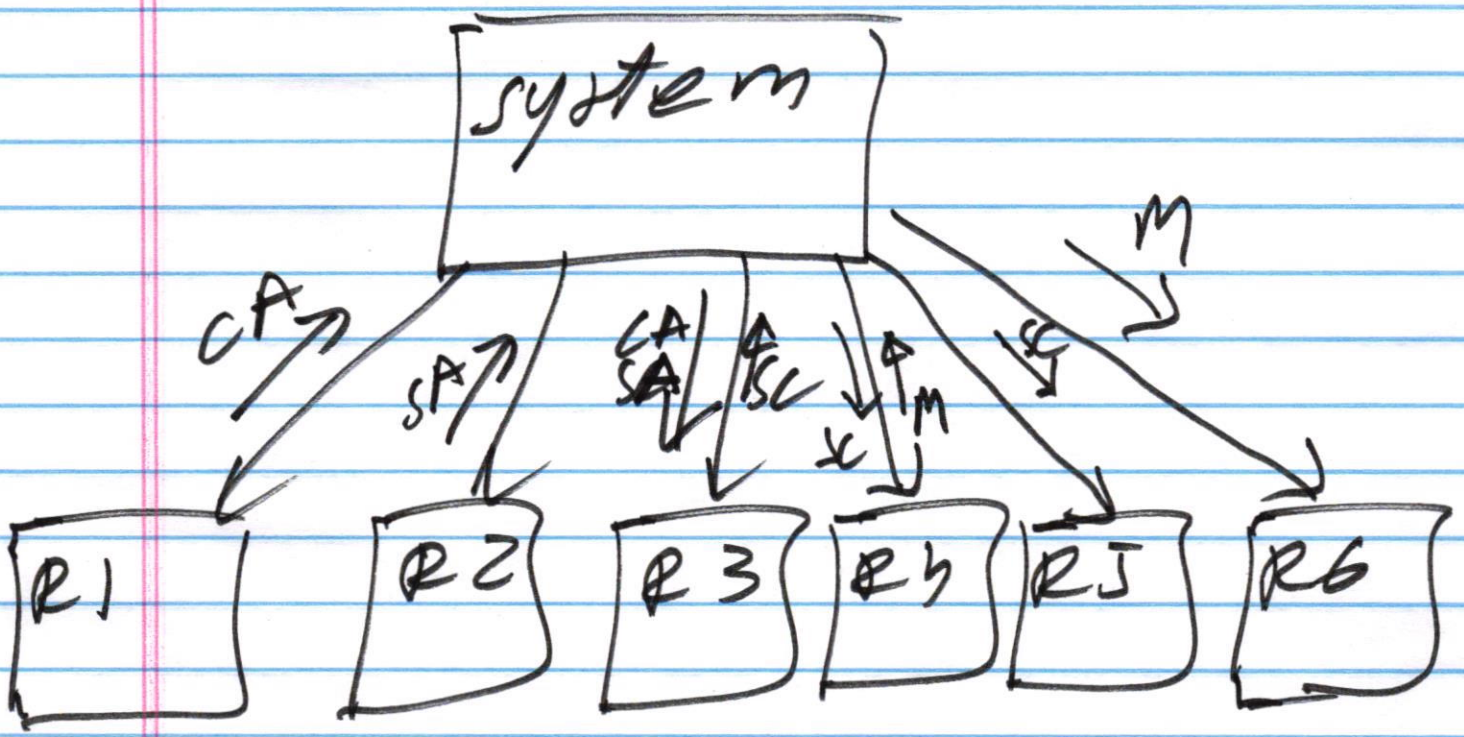
SA: student -"-

SC: scores

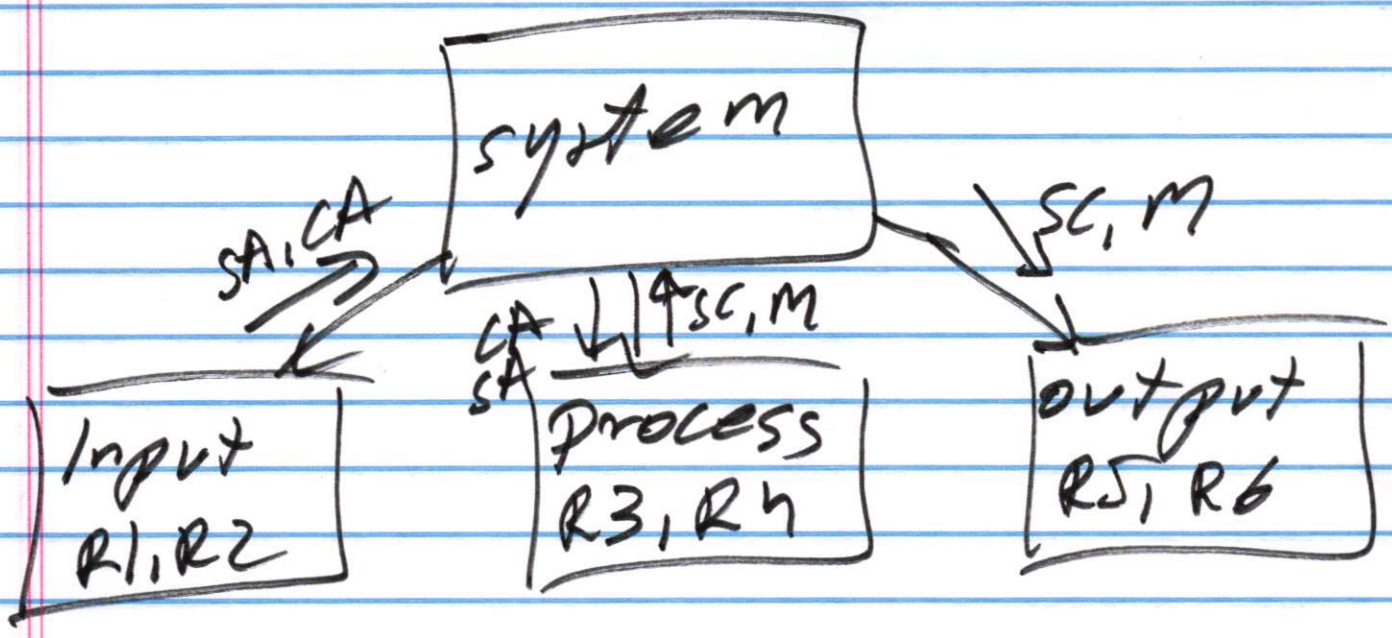
M: mean

All data structures are global

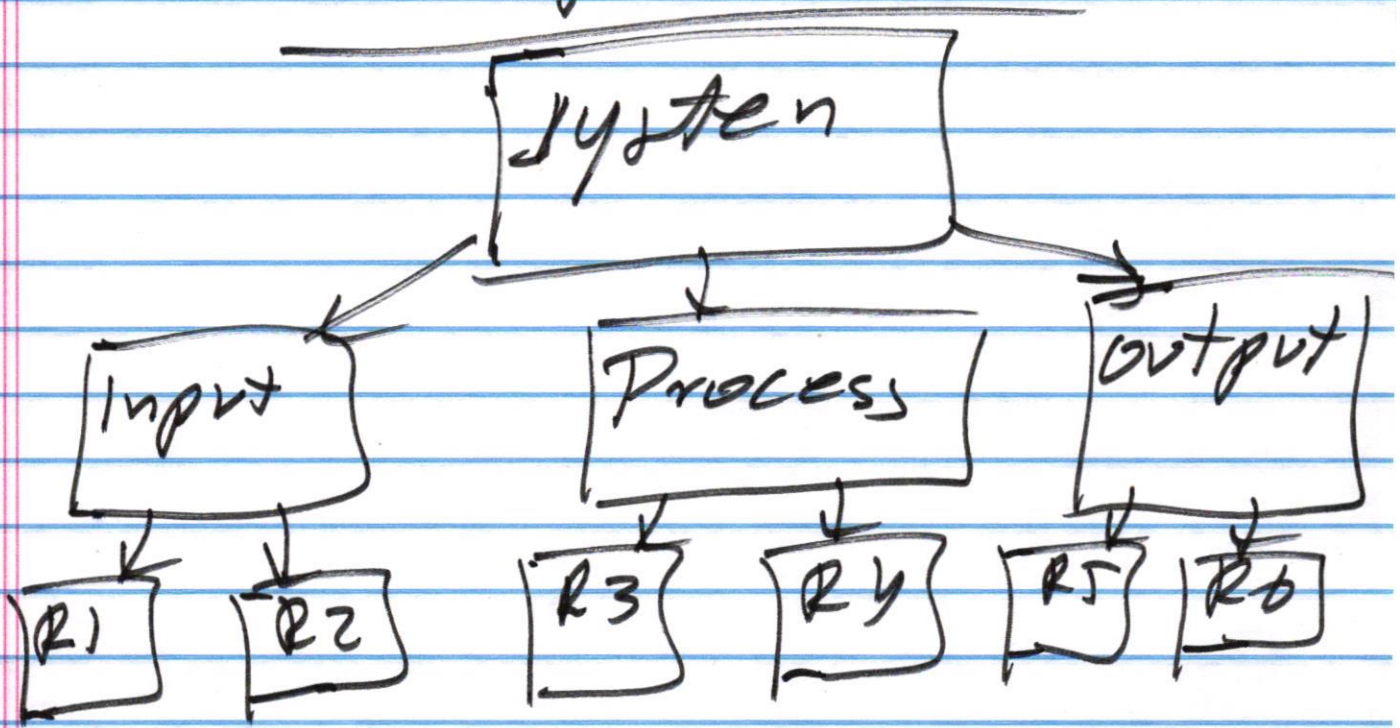
Design #3



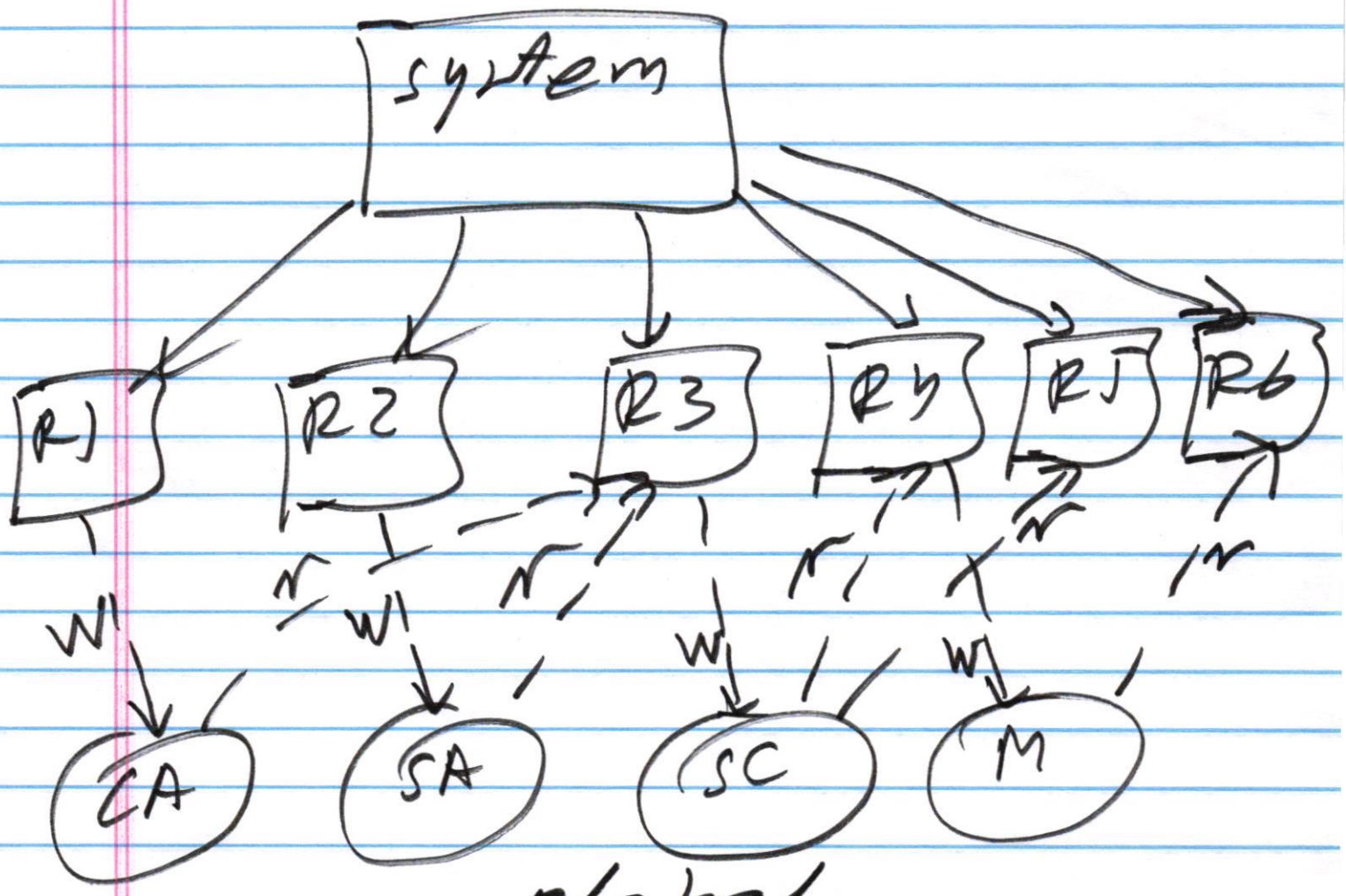
Design #4



Design #1



Design#2



global

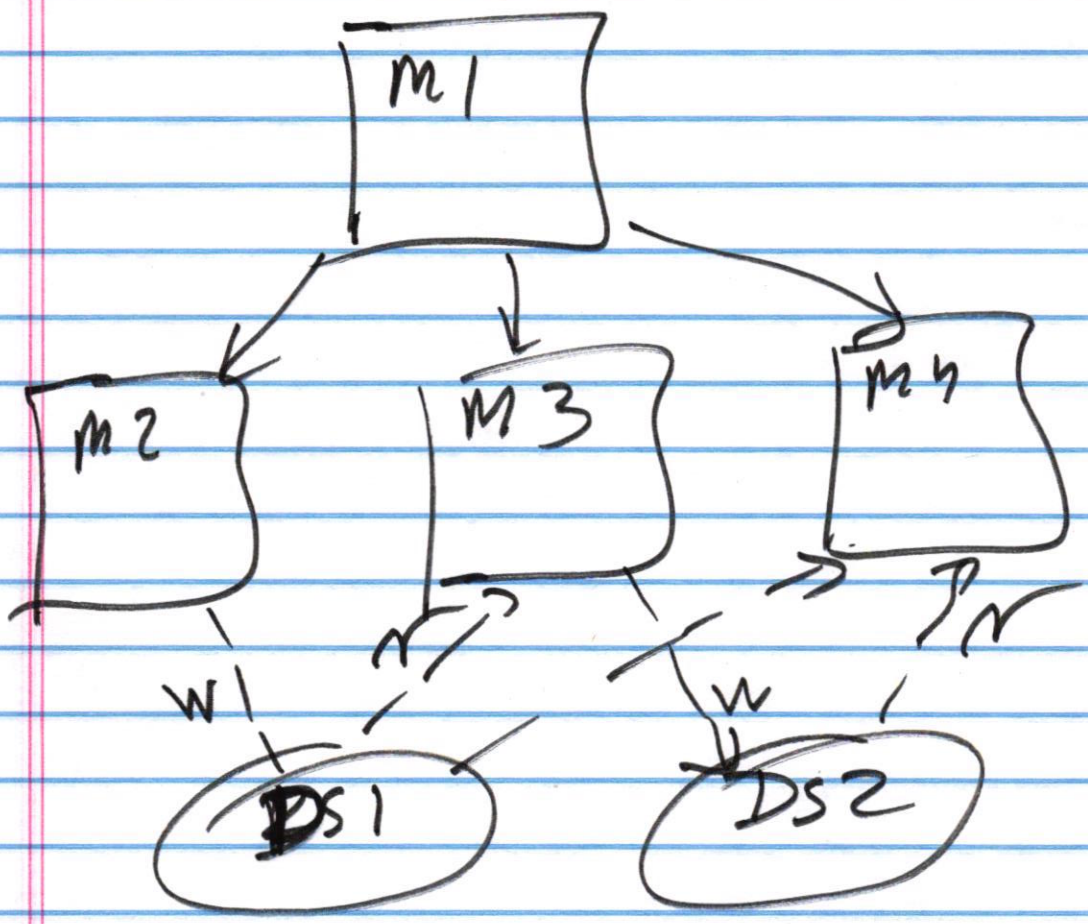
common couplings

R1 → R3
R2 → R3
R3 → R4
R3 → R5
R4 → R6

} common couplings.

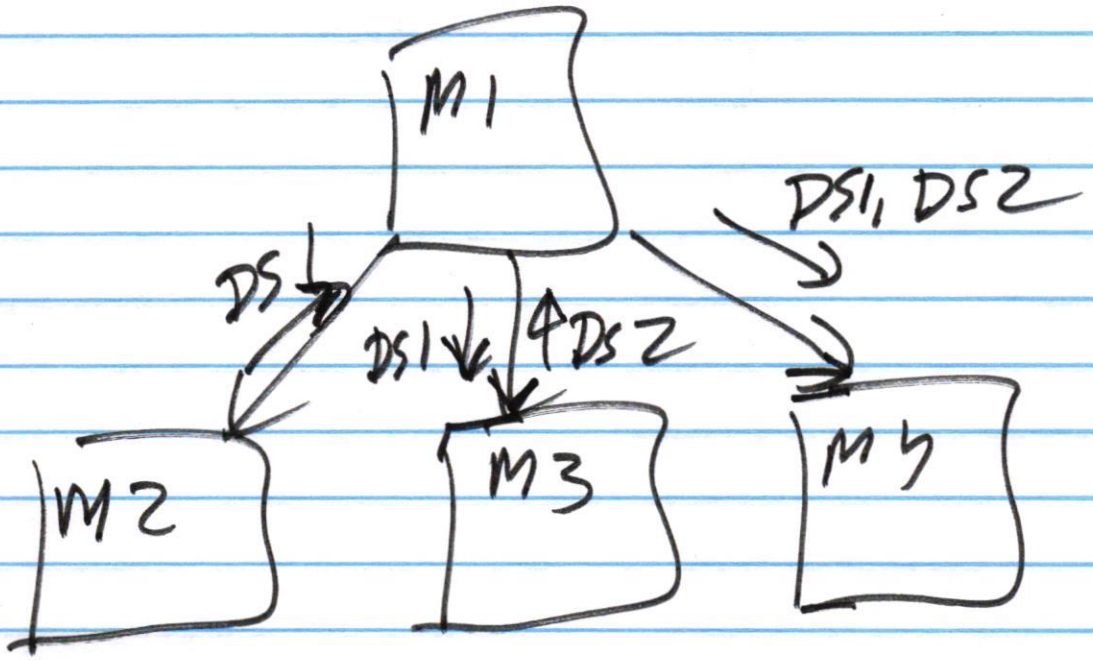
bad design

maintainability vs performance



maintainance: bad

performance: good

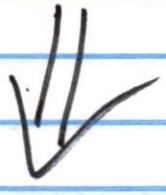


maintaince: good.

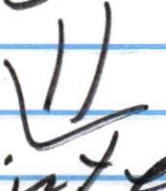
performance: bad

other types of coupling.

* compiler coupling.
  source code may be tied to some specific of a compiler

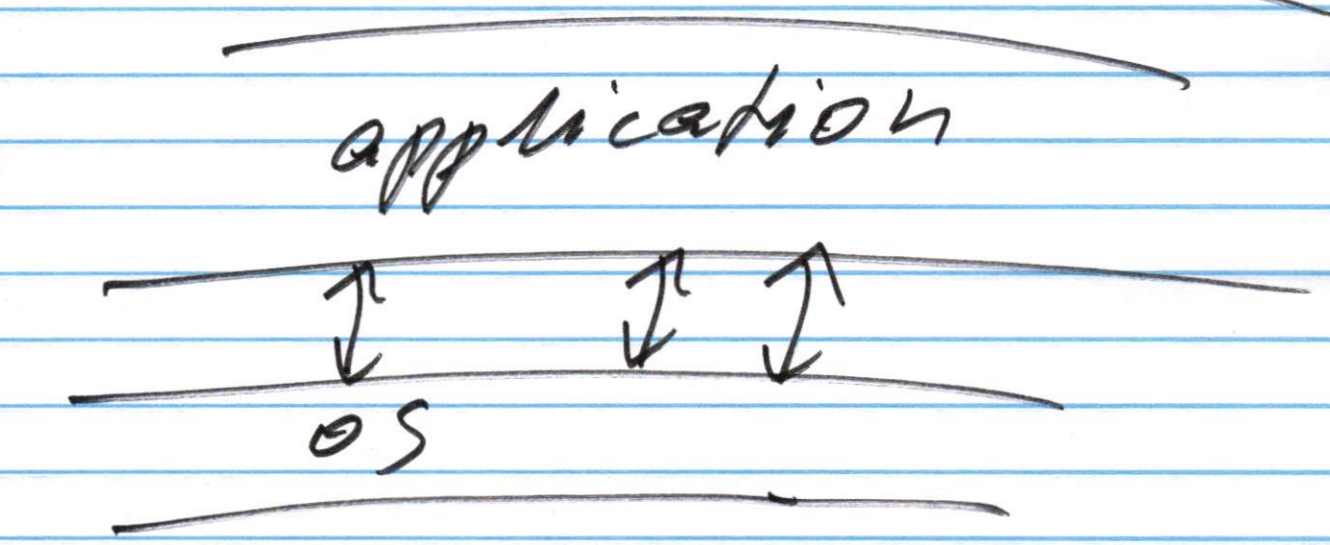


when a new version of a compiler is used



maintenance

operating system coupling.



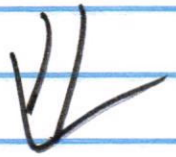
application takes advantage of some specifics of OS

↓

very strong coupling.

Hardware coupling.

taking advantage
of some specifics
of hardware



very strong coupling.

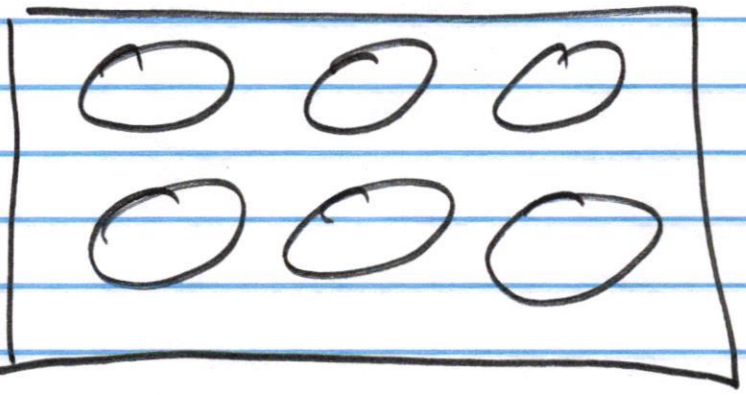
cohesion

a measure of relationships between internal elements of a component.

element: functional element responsibility.

no relationship
weak —"—

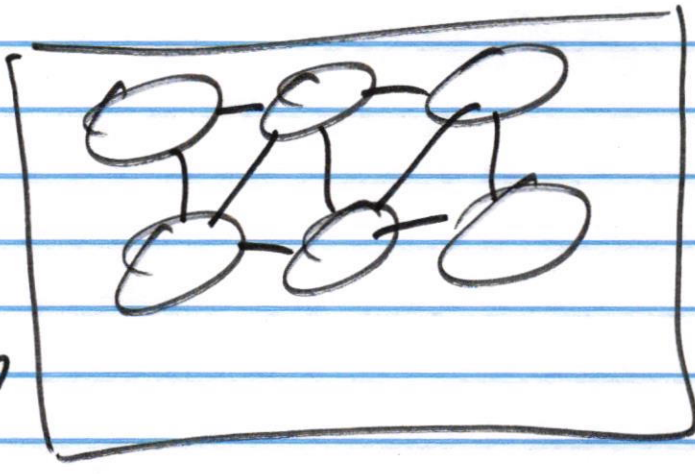
bad design



---

strong relationships

good design



grader system