



Inxhinierimi softverik

Pjesa 8 – Dizajni: karakteristikat dhe metrikat

Prof. Asoc. Dr. Ermir Rogova

Karakterizimi i një dizajni të mirë

- Përveq që dizajni duhet të plotësojë kërkesat — egzistojnë dy karakteristika themelore:
 - - Konsistenca në të gjithë dizajnin:
 - UI i përbashkët
 - Dukja
 - Rrjedha logjike
 - Procesimi i gabimeve i përbashkët
 - Raportet e përbashkëta
 - Ndërfaqet sistemore të përbashkëta
 - Ndiurma (help) e përbashkët
 - I gjithë dizajni në nivel të njëjtë
 - Kompletimi i dizajnit
 - Të gjitha kërkesat janë marrë parasysh.
 - Të gjitha pjesët e dizajnit kanë shkuar deri në kompletim në të njëjtin nivel të thellësisë.

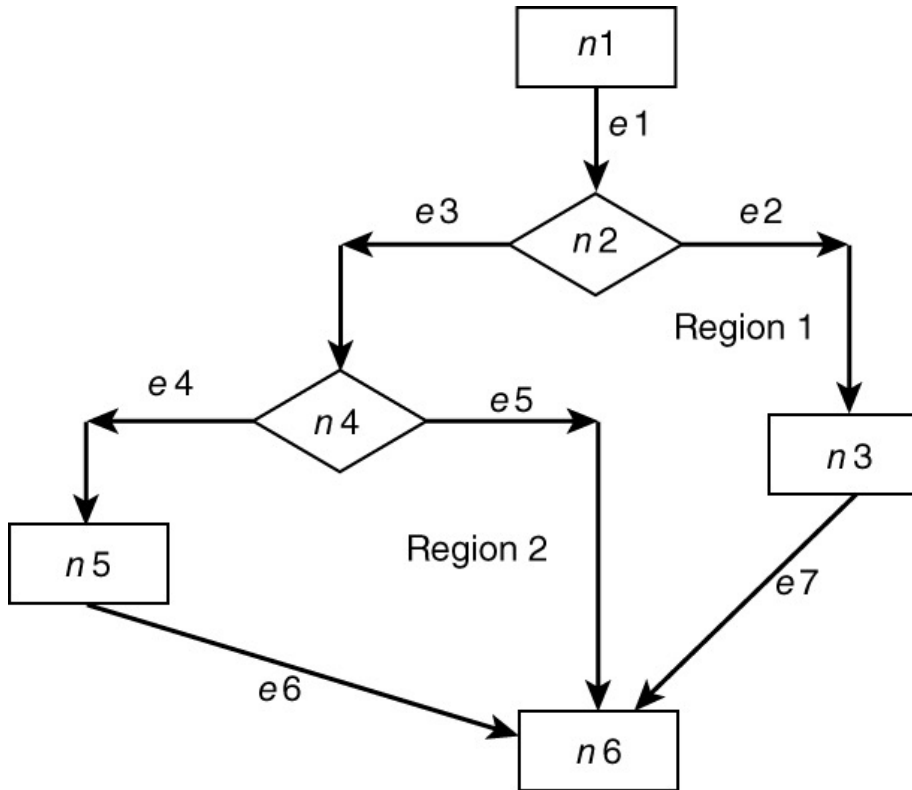
Kompleksiteti është i lidhur me dizainin e “Mirë/Keq”

- Disa karakterizime të kompleksitetit të dizajnit
 - Metrikat Halstead
 - Metrika e kompleksitetit ciklomatik e McCabe-it (ka përdorim të gjërë)
 - Metrika Henry-Kafura e rrjedhjes së Informacionit (fan-in/fan-out)
 - Metrika Card and Glass

Metrika Halstead

- Zhvilluar nga Maurice Halstead në vitet e 70^{ta} për të analizuar kompleksitetin e kodit burimor të programit.
- Përdor katër njësi themelore matjesh nga kodi:
 - $n1$ = numri i operatorëve të ndryshëm
 - $n2$ = numri i operandëve të ndryshëm
 - $N1$ = shuma e të gjitha paraqitjeve të $n1$
 - $N2$ = shuma e të gjitha paraqitjeve të $n2$
- Fjalori i programit: $n = n1 + n2$
- Gjatësia e programit: $N = N1 + N2$
- Nga këto ai definoi katër metrika:
 - Volumi: $V = N * (\log_2 n)$
 - Volumi potencial: $V^@ = (2 + n2^@) \log_2 (2+n2^@)$ $2^@$ numri i operandëve të përdorur nga funksioni
 - Niveli i implementimit të programit: $L = V^@ / V$
 - Mundi: $E = V / L$

Metrika McCabe



Cyclomatic complexity = $E - N + 2p$
where E = number of edges
 N = number of nodes
 p = number of connected components (usually 1)

So, for this control flow :
 $7 \text{ edges} - 6 \text{ nodes} + 2 = 3$

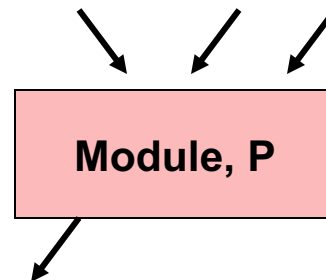
Cyclomatic complexity number can also be computed as follows:

- number of binary decision + 1
- number of closed regions + 1

- Metrika e kompleksitetit ciklomatik të **McCabe-it** bazohet në besimin që kualiteti i programit është i lidhur me kompleksitetin e “kontrollit të rrjedhës” së programit.

Henry-Kafura (Fan-In dhe Fan-Out)

- Metrika Henry-Kafura matë rrjedhën ndërmodulare, e cila përfshin:
 - Pasimin e parametrave
 - Qasjen e variablave globale
 - Inputet
 - Outputet
- Fan-in : numri hyrjeve intermodulare në një program
- Fan-out: numri i daljeve intermodulare nga një program



Module's complexity: $C_p = (\text{fan-in} \times \text{fan-out})^2$
for the "picture" above: $C_p = (3 \times 1)^2 = 9$

Card and Glass (Higher Level Complexity)

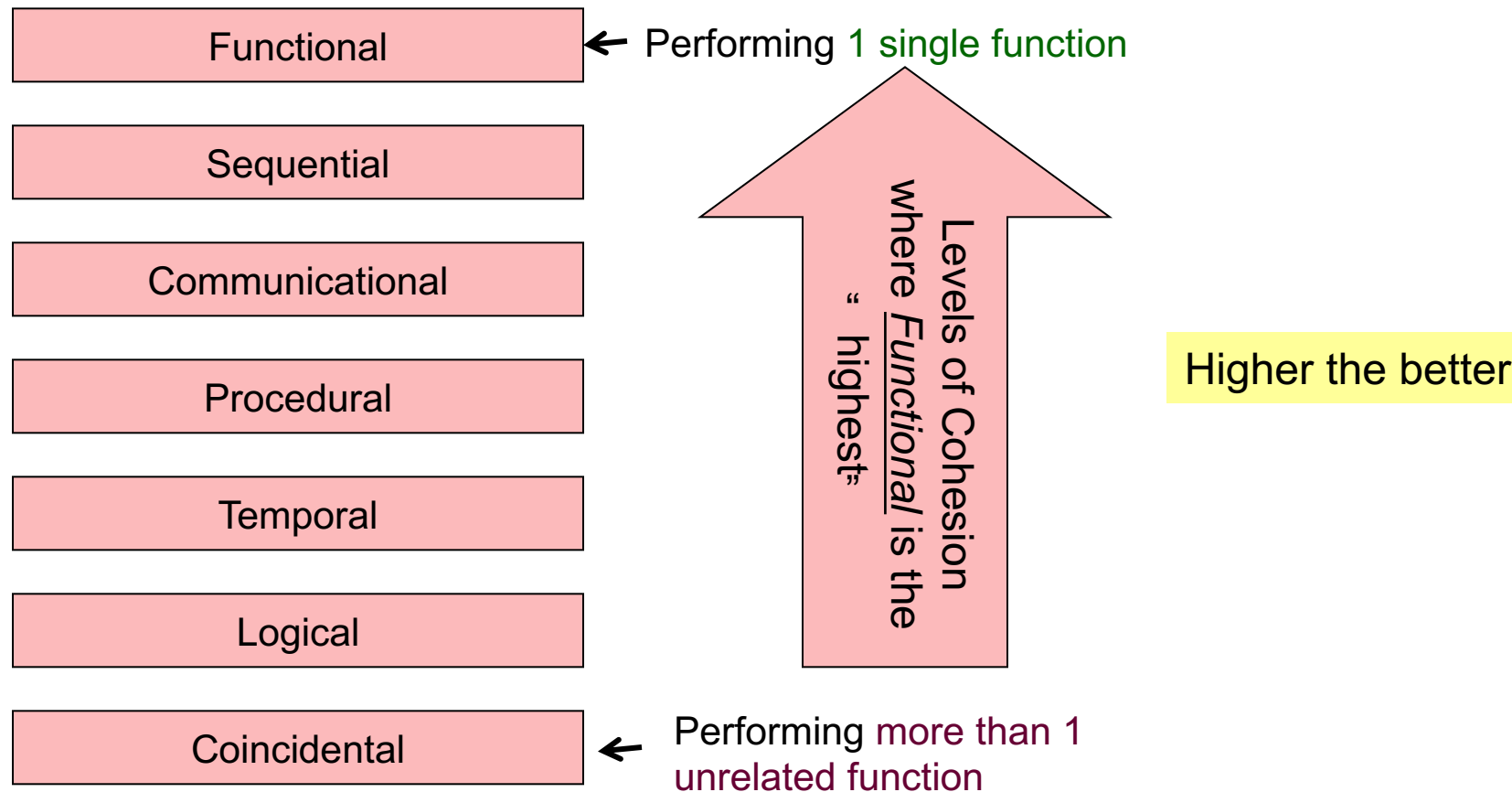
- Card and Glass përdor të njëjtin koncept si fan-in dhe fan-out për të përshkruar kompleksitetin e dizajnit:
 - Kompleksiteti struktural i modulit x
 - $S_x = (\text{fan-out})^2$
 - Kompleksiteti i të dhënave
 - $D_x = P_x / (\text{fan-out} + 1)$, ku P_x është numri i variablave që pasohen drejt dhe prej modulit
 - Kompleksiteti i Sistemit
 - $C_x = S_x + D_x$

Më shumë për atributet për dizajn të mirë

- Lehtë për tu:
 - Kuptuar
 - Ndryshuar
 - Ripërdorur
 - Testuar
 - Integruar
 - Koduar
- Shumë prej këtyre mund ti bëjmë nëse kemi parasysh:
 - Kohezionin (cohesion)
 - Kouplingun (coupling)

Kohezioni

- **Kohesioni** i një njësie, moduli, objekti apo komponenti adreson atributin e “shkallës së lidhjes” brenda asaj njësie, moduli, objekti apo komponenti.



Matja e kohezionit të programit

- Bieman dhe Ott kanë dhënë njësi matëse të kohezionit të programit duke përdorur këto koncepte:
 - data token është çfardo paraqitje e variablës apo konstatës një program.
 - slice brenda një programi është koleksioni i të gjitha deklaratave të cilat ndikojnë në vlerën e një variable specifike me interes.
 - data slice është koleksioni i të gjithë data tokenëve në slice të cilët do të ndikojnë në vlerën e një variable specifike me interes.
 - Glue tokens janë data tokenë në program që gjenden në më shumë se një data slice.
 - Super glue tokens janë data tokenë në program që gjenden në secilin data slice të programit.

Matja e kohezionit të programit përmes dy metrikave:

- weak functional cohesion = (# of glue tokens) / (total # of data tokens)
- strong functional cohesion = (# of super glue tokens) / (total # of data tokens)

Shembull i metrikave në Pseudo-Code

Finding the maximum and minimum values procedure

```
MinMax (z, n)
integer end, min, max, i;
end = n,
max = z[0];
min = z[0];
For (i = 0, i = < end; i++){
    if z[i] > max then max = z[i];
    if z[i] > min then min = z[i];
}
return max, min;
```

Data Tokens:	Slice max:	Slice min:	Glue Tokens:	Superglue:
z1	z1	z1	z1	z1
n1	n1	n1	n1	n1
end1	end1	end1	end1	end1
min1	max1	min1	l1	l1
max1	l1	l1	end2	end2
l1	end2	end2	n2	n2
end2	n2	n2	l2	l2
n2	max2	min2	03	03
max2	z2	z3	l3	l3
z2	01	02	end3	end3
01	l2	l2	l4 (11)	l4 (11)
min2	03	03		
z3	l3	l3		
02	end3	end3		
l2	l4	l4		
03	z4	z6		
l3	l5	l7		
end3	max3	min3		
l4	max4	min4		
z4	z5	z7		
l5	l6	l8		
max3	max5 (22)	min5 (22)		
max4				
z5				
l6				
z6				
l7				
min3				
min4				
z7				
l8				
max5				
min5 (33)				

Shembull i metrikave në Pseudo-Code

- Për gjetjen e min dhe max, glue tokens janë të njëjtë sikur super glue tokens.
 - Super glue tokens = 11
 - Glue tokens = 11
- Data slice për min dhe data slice për max del të jetë e njëjtë, 22.
- Numri total i data tokens është 33.

The cohesion metrics for the example of min-max are:

weak functional cohesion = $11 / 33 = 1/3$

strong functional cohesion = $11 / 33 = 1/3$

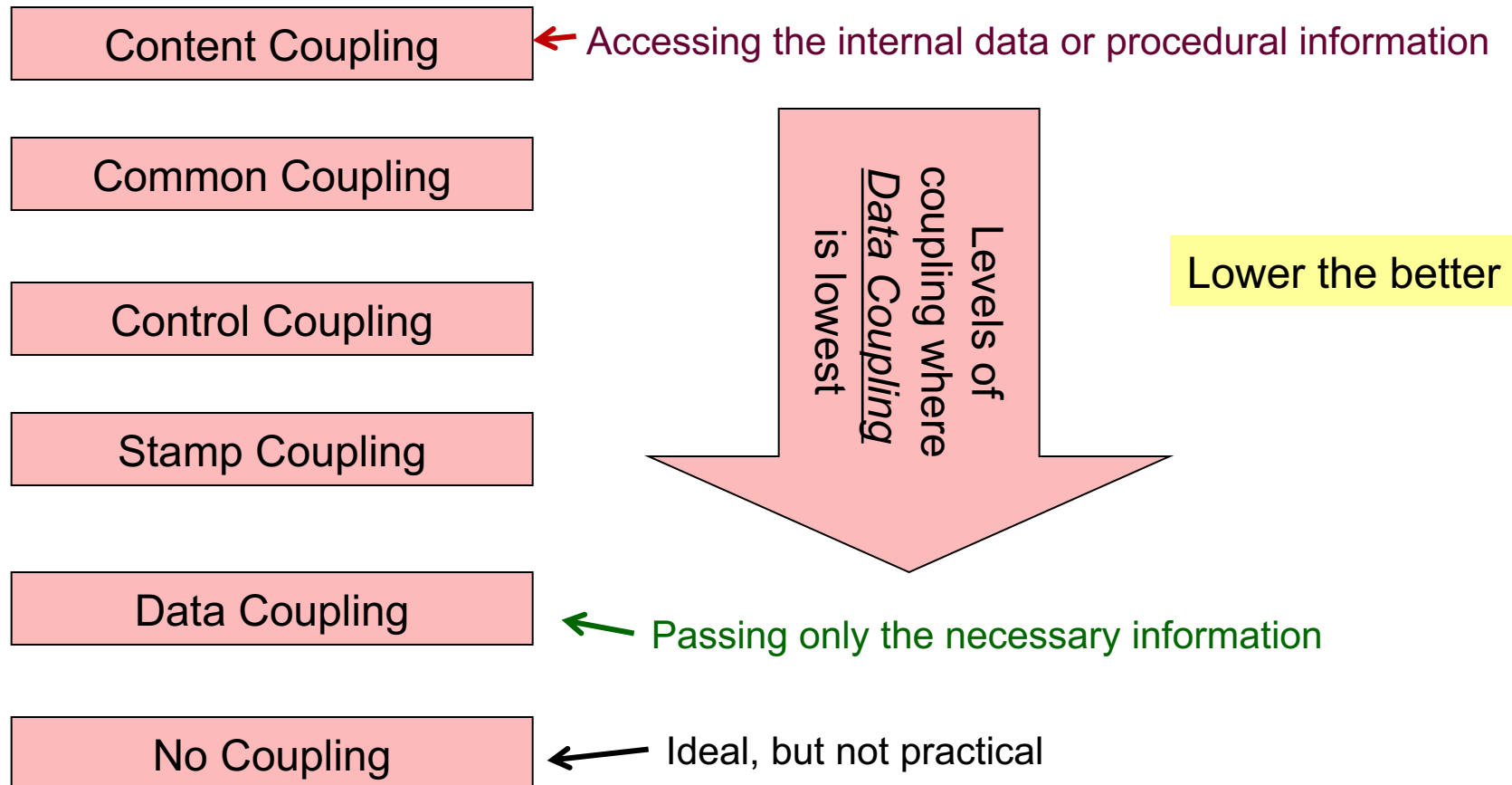
If we had only computed one function (e.g., max), then:

weak functional cohesion = $22 / 22 = 1$

strong functional cohesion = $22 / 22 = 1$

Kouplingu

- Kouplingu adreson atributin e “shkallës së ndërvvarshmërisë” ndërmjet njësive, moduleve apo komponenteve softverike.



Origjina e Ligjit Demeter

- Një udhëzues dizajni për sistemet OO i cili u paraqit nga projekti Demeter System në:
 - Northeastern University në vitet 80-ta
 - Projekt për Programimin “Aspect-Oriented”
- Adreson qështjen e kouplingut të dizajnit përmes vendosjes së kufizimeve në dërgimin e mesazheve ndërmjet objekteve

Ligji Demeter

- Një objekt duhet të **dërgojë mesazhe vetëm** këtyre llojeve të objekteve:
 - *Vetë objektit*
 - *Atributeve të objektit (instance variables)*
 - *Parametrave të metodave në objekt*
 - *Çfardo objekti të krijuar nga metoda në objekt*
 - *Çfardo objekti të kthyer nga thirrja e bërë njëjës nga metodave të objektit*
 - *Çfardo objekti në cilindo koleksion nga kategoritë e mësipërme*

User Interface

- Tri rregullat e arta të Mandel-it për dizajnin e user interface (UI)
 - *Jepi shfrytëzuesit kontrollën.*
 - *Redukto ngarkesën e memorjes së shfrytëzuesit*
 - *Konsistenca.*
- Shneiderman dhe Plaisant (tetë rregullat për dizajn)
 - *Konsistenca*
 - *Short cuts për shfrytëzuesit e shpeshtë (apo me eksperiencë).*
 - *Feedback informativ.*
 - *Dialogjet duhet të kenë përfundim.*
 - *Preventiva dhe trajtimi i gabimeve.*
 - *Kthimi i lehtë i veprimeve (“undo”).*
 - *Kontroll i brendshëm.*
 - *Reduktim i memorjes afat shkurtë.*

Prototipi UI dhe “Testimi”

- Prototipet e dizajnit UI:
 - Low fidelity (me kartonë)
 - High fidelity (me vegla për “story board”)
- Testime në përdorshmëri dhe analiza statistikore
 - Numri i personave që mund të përfundojnë një detyrë brenda një kohe të caktuar
 - Koha e nevojshme për të kryer detyra të ndryshme
 - Numri sa herë janë nevojitur funksionet “help”
 - Numri sa herë është përdorur “redo” dhe ku
 - Numri sa herë janë përdorur “short cuts”

Pyetje???