



YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

nesneye yönelik yazılım geliřtirm..

Yüksek Lisans Tezi

hüsniye ertemel

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

NESNEYE YÖNELİK YAZILIM GELİŞTİRMEDE
KALİTE ÖLÇÜTLERİNİN İNCELENMESİ

Bilgisayar Mühendisi Hüsniye Özlem ERTEMEL

FBE Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Prof. Dr. Oya KALIPSIZ

[Signature]

[Signature]

Doç. Dr. CAN ÖZTÜRKAN

İstanbul, 2009

[Signature]
Yrd. Doç. Dr. Yunus Emre Sel

İÇİNDEKİLER

	Sayfa
SİGMA LİSTESİ	v
İKİSALON LİSTESİ	viii
SEKİL LİSTESİ	x
CİZELER LİSTESİ	xiii
ÖNSÖZ	xv
ÖZET	xvii
ABSTRACT	xviii

1.	GİRİŞ	1
1.1.	Uygun	2
1.2.	Tez Organizasyonu	3
2.	YAZILIM UYUM ÖLÇÜLERİNİN KULLANIM AMAÇLARI	3
2.1.	Yazılım Kalitesinin Değerlendirilmesi	4
2.2.	Yazılım Kalitesinin ve Hata Eğiliminin Tespiti	5
2.3.	Yazılım Modülizasyonu	6
2.4.	Testler Kullanılarak Hata Potansiyelinin Belirlenmesi	6
3.	NY SİSTEMLERDE UYUMU ÖLÇMEK İÇİN KULLANILAN YAYINLAR	7
3.1.	Yapısal Uyum Ölçütleri	8
3.2.	Semantik Uyum Ölçütleri	9
3.3.	Ekspresif Uyum Ölçütleri	10
3.4.	Diğer Tabanlı Uyum Ölçütleri	11
3.5.	Yazılım Uyum Ölçütleri İçin Kullanılan Ölçütler	12
3.6.	Yazılım Uyum Ölçütleri İçin Kullanılan Ölçütlerin Yorumlanması	13
4.	NY SİSTEMLERDE KULLANILAN YAZILIM UYUM ÖLÇÜLERİ	14
4.1.	Yazılım Uyum Ölçütleri İçin Kullanılan Ölçütler	15
4.1.1.	Yazılım Uyum Ölçütleri İçin Kullanılan Ölçütler	16
4.1.2.	Yazılım Uyum Ölçütleri İçin Kullanılan Ölçütler	17
4.2.	Yazılım Uyum Ölçütleri İçin Kullanılan Ölçütler	18
4.3.	Yazılım Uyum Ölçütleri İçin Kullanılan Ölçütler	19

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ	v
KISALTMA LİSTESİ	viii
ŞEKİL LİSTESİ	x
ÇİZELGE LİSTESİ	xiii
ÖNSÖZ	xvi
ÖZET	xvii
ABSTRACT	xviii
1. GİRİŞ	1
1.1 Uyum	2
1.2 Tez Organizasyonu	3
2. YAZILIM UYUM ÖLÇÜTLERİNİN KULLANIM AMAÇLARI	5
2.1 Yazılım Kalitesinin Değerlendirilmesi	5
2.2 Yazılım Kalitesinin ve Hata Eğilimliliğinin Tahmini	5
2.3 Yazılım Modülleştirilmesi	6
2.4 Tekrar Kullanılabilir olan Bileşenlerin Belirlenmesi	6
3. NY SİSTEMLERDE UYUMU ÖLÇMEK İÇİN KULLANILAN YAKLAŞIMLAR	8
3.1 Yapısal Uyum Ölçütleri	8
3.2 Semantik Uyum Ölçütleri	8
3.3 Entropi Tabanlı Uyum Ölçütleri	9
3.4 Dilim Tabanlı Uyum Ölçütleri	9
3.5 Veri Madenciliğine dayalı Ölçütler	10
3.6 Bilgi Tabanlı Sistemler ve Dağıtık Sistemler gibi Özelleştirilmiş Yazılım Uygulamaları için Ölçütler	10
4. NY PROGRAMLAMADA YAPISAL BENZERLİK/UYUM ÖLÇÜTLERİ	12
4.1 Chidamber ve Kemerer Tarafından Önerilen Ölçütler	15
4.1.1 LCOM1	15
4.1.2 LCOM2	17
4.2 Hitz ve Montazeri Tarafından Önerilen Ölçütler	19
4.2.1 LCOM3	19
4.2.2 LCOM4	23
4.2.3 Co	27
4.3 Bieman ve Kang Tarafından Önerilen Ölçütler	29

4.3.1	TCC	30
4.3.2	LCC	33
4.4	Henderson-Sellors Tarafından Önerilen Ölçütler	34
4.4.1	LCOM5	34
4.5	Lee ve arkadaşları Tarafından Önerilen Ölçütler	37
4.5.1	ICH	37
4.6	Briand ve arkadaşları Tarafından Önerilen Ölçütler	42
4.6.1	RCI	42
4.6.2	Coh	44
4.7	Chae ve arkadaşları Tarafından Önerilen Ölçütler	46
4.7.1	CBMC	46
4.8	Bansiya ve arkadaşları Tarafından Önerilen Ölçütler	51
4.8.1	CAMC	51
4.9	Counsell ve arkadaşları tarafından Önerilen Ölçütler	54
4.9.1	NHD	54
4.10	Bonja ve arkadaşları Tarafından Önerilen Ölçütler	56
4.10.1	CC(X)	56
4.11	Zhou ve Xu Tarafından Önerilen Ölçütler	62
4.11.1	ICBMC	62
4.12	Jarallah ve arkadaşları Tarafından Önerilen Ölçütler	66
4.12.1	CCM	67
4.12.2	ECCM	68
4.13	Aman ve arkadaşları Tarafından Önerilen Ölçütler	69
4.13.1	OCC	70
4.13.2	PCC	74
4.14	Yang Tarafından Önerilen Ölçütler	77
4.14.1	OL _n	77
4.15	Badri Tarafından Önerilen Ölçütler	80
4.15.1	DC _D	81
4.15.2	DC _I	85
4.15.3	LCC _D	87
4.15.4	LCC _I	88
4.16	Gui Tarafından Önerilen Ölçütler	92
4.16.1	WTCoh ve ClassCoh	92
4.17	Ölçütlere Genel Bir Bakış	96
5.	NESNEYE YÖNELİK PROGRAMLAMADA YAPISAL BENZERLİK/UYUM ÖLÇÜTLERİ İÇİN SINIFLANDIRMA	99
5.1	Giriş	99
5.2	Sınıflandırma Kriterleri	99
5.3	Yapısal Benzerlik/Uyum Ölçütlerinin Sınıflandırılması	104
6.	YENİ ÖLÇÜT COMIAS	105
6.1	Metot Çağrılarının İncelenmesi	105
6.2	Üye Alan Kullanımlarının İncelenmesi	107
6.3	Metot Çağırımı ve Üye Alan Kullanımı	107
6.4	COMIAS Uyum Ölçütünün Tanımlanması	108
6.5	COMIAS Uyum Ölçütünün Örneklenmesi	110
6.6	Önerilen Ölçüt ve LCOM Ölçütlerinin Bir Örnek Üzerinde İncelenmesi	116
6.7	Teorik Geçerleme	119

6.8	Deneysel Geerleme	122
6.8.1	Kullanılan Program.....	122
6.8.2	Kalitesi llecek Yazılımın Seimi	123
6.8.3	lm Sonuları	124
6.8.4	Kalıtım Konusunun Değerlendirilmesi.....	129
6.8.4.1	Temel Sınıf “KokOzelDurumu” ve ilgili Tretilmiř Sınıfları.....	129
6.8.4.2	Temel Sınıf “TemelRaporlayici” ve ilgili Tretilmiř Sınıfları.....	136
6.8.4.3	Kalıtım ile ilgili lm Sonularının İncelenmesi.....	140
6.8.5	lmlerde İlgin Bulunan Sonuların İncelenmesi.....	141
6.9	Sınıflandırma	148
7.	SONU.....	149
KAYNAKLAR.....		151
EKLER		154
Ek 1 ltlerin sınıflandırılması		155
ZGEMİř.....		178

SİMGE LİSTESİ

a	Üye alan sayısı
ac	Üye alana doğrudan erişim ve <i>erişim metotları</i> ile olan dolaylı erişim
A	Parametre uyuşma matrisi
$Attr_pay(M_i, M_k)$	İki metot arasındaki üye alan paylaşımı ilişkisi
c_j	Parametre oluşum matrisi O 'daki j . kolondaki 1'lerin sayısı
C	Sınıf
Co	Connectivity
COM(m)	Metot m'nin güçlülüğü
COV(v)	Üye alan v'nin güçlülüğü
CS(G(c))	G(c) referans grafiğinin bütün kesme kümelerinin kümesi
DC _D	Metotlar arasındaki <i>doğrudan</i> ilişkileri hesaba katan uyum
DC _I	Metotlar arasındaki <i>doğrudan</i> ilişkilerin yanı sıra <i>dolaylı</i> ilişkileri de hesaba katan uyum
$e_{s,t}$	s ve t köşeleri arasındaki kenar
E	G referans grafiğinin kenarları
$F_c(G)$	G referans grafiğinin bağlılık çarpanı
$F_s(G)$	G referans grafiğinin yapı çarpanı
G	Referans grafiği
G _c	Referans grafiği
$G_w(V, E)$	Zayıf bağlantı grafiği, yönsüz bir grafik
$G_s(V, E)$	Güçlü bağlantı grafiği, yönlü bir grafik
G^i	G referans grafiğinin n çocuğundan biri
I	Üye alan
I _x	X sınıfının üye alanlarının kümesi
IM	Method invocation
IM_{M_i}	C sınıfının, M_i metodu tarafından <i>doğrudan</i> veya <i>dolaylı yoldan</i> çağrılan bütün metotlarının kümesi
İlişki(M_i, M_k)	M_i metodu ile M_k metodu arasındaki ilişki değeri
L	Parametre tip listesi
LCC _D	Metotlar arasındaki <i>doğrudan</i> ilişkileri hesaba katan uyum eksikliği
LCC _I	Metotlar arasındaki <i>doğrudan</i> ilişkilerin yanı sıra <i>dolaylı</i> ilişkileri de hesaba katan uyum eksikliği

m	Metot sayısı
M	Metot
$\text{Max}(\text{Metod_cag})$	Maksimum metot çağırımı ilişkisi
$\text{Max}(\text{Attr_pay})$	Maksimum üye alan paylaşımı ilişkisi
$\text{Max_İlişki}(X)$	Bir sınıftaki maksimum ilişki değeri
$\text{Metod_cag}(M_i, M_k)$	İki metot arasındaki metot çağırımı ilişkisi
$M_g(G)$	Yapışkan metotlar kümesi
$M_I(c)$	Yeni tanımlanmış veya kalıtım alınıp override edilmiş metotlar kümesi
$M_n(G)$	Normal metotlar kümesi
$M_{NEW}(c)$	Yeni tanımlanmış metotlar kümesi
$M_{OVR}(c)$	Override edilmiş metotlar kümesi
M_x	X sınıfının metotlarının kümesi
$\text{NC}(C)$	Sınıfın metotları arasındaki gerçek bağlantı sayısı
$\text{NCC}(C)$	Bağlantı grafiği G_c 'nin bağlı bileşenlerinin sayısı
$\text{NDC}(C)$	Bir sınıftaki doğrudan bağlantı sayısı
$\text{NIC}(C)$	Bir sınıftaki dolaylı bağlantı sayısı
$\text{NMP}(C)$	Sınıfın metotları arasındaki mümkün olan maximum bağlantı sayısı
$\text{NOIM}(C)$	Sınıftaki kalıtım alınmış metot sayısı
$\text{NORM}(C)$	Sınıftaki yeniden tanımlanan metot sayısı
$\text{NP}(C)$	Bir sınıftaki mümkün olan maximum doğrudan veya dolaylı bağlantı sayısı.
$\text{NPI}(m, m')$	m metodu tarafından m' metodunun polimorfik çağrılma sayısı
$N_m(G_c)$	G_c ve $\lambda(m_j, c)$ 'deki metotların sayısı
$N_v(G_c)$	G_c grafiğindeki üye alanların sayısı
O	Parametre oluşum matrisi
OL_n	Sınıfın n. iterasyondaki uyum değerini veren ölçüt
$\text{Par}(m')$	m' metodunun parametre kümesi
PUM	Public metotlar kümesi
re	üye alana veri okuyarak erişmek
r_i	Parametre oluşum matrisi O 'daki i . satırdaki 1'lerin sayısı
$R_w(m_i)$	$G_w(V, E)$ grafiğinde m_i metodu tarafından erişilebilen metotlar kümesi
$\text{Sim}(i, j)$	En büyük SimT değerine sahip yol
$\text{SimD}(i, j)$	M_i ve M_j metotlarının direk benzerliğinin ölçümü

$\text{SimT}(i, j, \pi)M_i$ ve M_j metotlarının π yolu boyunca dolaylı benzerliğinin ölçümü

SS Sınıf kümesi

Q cut set

$T(G)$ Yapı ağacı

Toplam_İlişki(X) Sınıfın toplam ilişki değeri

UA Attribute usage

UA_{M_i} C sınıfının, M_i metodu tarafından *doğrudan* veya *dolaylı yoldan* kullanılan üye alanların kümesi

V G referans grafiğinin köşeleri

wr üye alana veri yazarak erişmek

$|IV|_t$ İki metot tarafından kullanılan toplam üye alan sayısı

$|IV|_c$ İki metot tarafından kullanılan ortak üye alan sayısı

$\{IV_1\}$ M_1 metodu tarafından kullanılan üye alanlar kümesi

$\{IV_2\}$ M_2 metodu tarafından kullanılan üye alanlar kümesi

$\{V_i\}$ M_i metodu tarafından kullanılan üye alanların kümesi

σ Parametre oluşum matrisi O 'daki 1'lerin sayısı

$\sigma()$ Benzerlik - 2 şeyin özellikleri kümelerinin kesişimi

$\mu(A_j)$ A_j üye alanına referans eden metot sayısı

Π M_i 'den M_j metoduna giden bütün yolların kümesi

$\lambda(m_j, c)$ c sınıfındaki m_j metodu tarafından erişilebilen üye alanlar kümesi

$\rho(G_c)$ G_c ağacındaki her bir düğümün maximum sahip olabileceği çocuk sayısı

KISALTMA LİSTESİ

BCML	Byte Code Metric Library
BMB	Briand, Morasca and Basili
CAMC	Cohesion Among Methods of Classes
CBMC	Cohesion Based on Member Connectivity
CC	Class Cohesion
CCM	Class Connection Metric
CCQ	Class Comment Quality
CDC	Class Domain Complexity
ChF	Cohesion of Frame
CI	Cohesive Interactions
CIC	Class Interface Complexity
CIQ	Class Identifier Quality
ckjm	Chidamber and Kemerer Java Metrics
ClassCoh	Cohesion of the class
CM	Constructor Methods
COa	Class Overlap A
COb	Class Overlap B
Coh	Cohesion
COMIAS	Cohesion Method Invocation Attribute Sharing
DAS	Direct Attribute Sharing
DD	Data – Data
DDI	Data – Data Interaction
DM	Data – Method
DMI	Data – Method Interaction
DMMPP	Doğrudan Metot – Metot Parametre Paylaşımı
DMMR	Direct Method Method Referencing
ECCM	Enhanced Class Connection Metric
EM	Erişim Metodu
GRRN	General Regression
IAS	Indirect Attribute Sharing
ICBMC	Improved Cohesion Measure for Classes
ICH	Information-flow based cohesion
IMMR	Indirect Method Method Referencing

ISA	Identification of Subsystems based on Associations
KCI	Key Class Identification
KS	Kısmen Sağlanır
LCC	Loose Class Cohesion
LCOM	Lack of Cohesion of Methods
LORM	Logical Relatedness of Methods
MAR	Method Attribute Referencing
MCC	Most Cohesive Component
MIBAT	Method Interactions Based on Access Types
MPI	Method Parametre Inclusion
NPI	Metodun Parametre Sayısı
MS	Method Benzerlik
NHD	Normalised Hamming Distance
NY	Nesneye Yönelik
NYP	Nesneye Yönelik programlama
OCC	Optimistic Class Cohesion
OCDQa	Overall Class Documentation Quality A
OCDQb	Overall Class Documentation Quality B
PCC	Pessimistic Class Cohesion
RCI	Ratio of Cohesive Interactions
SCDE	Semantic Class Definition Entropy
SCIDE	Semantic Class Interface Definition Entropy
SZ	Sağlanmaz
TCC	Tight Class Cohesion
UZ	Uygulanamaz
WTCoh	Weighted Transitive Cohesion of the complete software system

ŞEKİL LİSTESİ

Şekil 4.1 Bir java program örneği	12
Şekil 4.2 Point sınıfının metot – üye alan ilişkisi grafiği	13
Şekil 4.3 Point sınıfının metot – metot ilişkisi grafiği.....	13
Şekil 4.4. Sınıf A ve Sınıf B metot – üye alan ve metot – metot ilişki örneği	14
Şekil 4.5 LCOM3 (1) şikkına göre metot-üye alan ilişkisi grafiği.....	20
Şekil 4.6 LCOM3 (1) şikkına göre G_x grafiği	21
Şekil 4.7 LCOM3 (2) şikkına göre metot-üye alan ilişkisi grafiği.....	21
Şekil 4.8 LCOM3 (2) şikkına göre G_x grafiği	21
Şekil 4.9 LCOM3 (3) şikkına göre metot-üye alan ilişkisi grafiği.....	22
Şekil 4.10 LCOM3 (3) şikkına göre G_x grafiği	22
Şekil 4.11 LCOM3 Sınıf A ve Sınıf B'ye ait G_x grafiği	22
Şekil 4.12 LCOM4 (1) şikkına göre metot-üye alan ilişkisi grafiği.....	24
Şekil 4.13 LCOM4 (1) şikkına göre metot çağırımı ilişkisi grafiği	24
Şekil 4.14 LCOM4 (1) şikkına göre G_x grafiği	24
Şekil 4.15 LCOM4 (2) şikkına göre metot-üye alan ilişkisi grafiği.....	25
Şekil 4.16 LCOM4 (2) şikkına göre metot çağırımı ilişkisi grafiği	25
Şekil 4.17 LCOM4 (2) şikkına göre G_x grafiği	25
Şekil 4.18 LCOM4 (3) şikkına göre metot - üye alan ilişkisi grafiği.....	25
Şekil 4.19 LCOM4 (3) şikkına göre metot çağırımı ilişkisi grafiği	26
Şekil 4.20 LCOM4 (3) şikkına göre G_x grafiği	26
Şekil 4.21 LCOM4 Sınıf A ve Sınıf B'ye ait G_x grafiği.....	26
Şekil 4.22 TCC (1) şikkına göre metot-üye alan ve metot çağırımı ilişkisi grafiği.....	31
Şekil 4.23 TCC (2) şikkına göre metot-üye alan ve metot çağırımı ilişkisi grafiği.....	31
Şekil 4.24 TCC (3) şikkına göre metot-üye alan ve metot çağırımı ilişkisi grafiği.....	32
Şekil 4.25 LCOM5 (1) şikkına göre metot-üye alan ilişkisi ve μ değerleri	35
Şekil 4.26 LCOM5 (2) şikkına göre metot-üye alan ilişkisi ve μ değerleri	35
Şekil 4.27 LCOM5 (3) şikkına göre metot-üye alan ilişkisi ve μ değerleri	36
Şekil 4.28 LCOM5 Sınıf A ve Sınıf B metot-üye alan ilişkisi ve μ değerleri.....	36
Şekil 4.29 CBMC (1) şikkına göre referans grafiği G.....	48
Şekil 4.30 CBMC (1) şikkına göre yapı ağacı.....	49
Şekil 4.31 CBMC Sınıf A ve Sınıf B referans grafikleri.....	49
Şekil 4.32 CBMC Sınıf A ve Sınıf B yapı ağaçları.....	50
Şekil 4.33 ICBMC (1) şikkına göre referans grafiği G	64

Şekil 4.34 ICBMC (1) şikkına göre yapı ağacı	65
Şekil 4.35 ICBMC Sınıf A ve Sınıf B referans grafikleri	65
Şekil 4.36 ICBMC Sınıf A ve Sınıf B için yapı ağaçları.....	66
Şekil 4.37 OCC (1) şikkına göre zayıf bağlantı grafiği.....	71
Şekil 4.38 OCC (2) şikkına göre zayıf bağlantı grafiği.....	72
Şekil 4.39 OCC (3) şikkına göre zayıf bağlantı grafiği.....	73
Şekil 4.40 PCC (1) şikkına göre güçlü bağlantı grafiği.....	75
Şekil 4.41 PCC (2) şikkına göre güçlü bağlantı grafiği.....	76
Şekil 4.42 PCC (3) şikkına göre güçlü bağlantı grafiği.....	76
Şekil 4.43 DC _D (1) şikkına göre G _D grafiği.....	83
Şekil 4.44 DC _D (2) şikkına göre G _D grafiği.....	83
Şekil 4.45 DC _D (3) şikkına göre G _D grafiği.....	84
Şekil 4.46 DC _D Sınıf A ya ve Sınıf B ye ait G _D	85
Şekil 4.47 DC _I (1) şikkına göre G _I grafiği.....	86
Şekil 4.48 DC _I (3) şikkına göre G _I grafiği.....	87
Şekil 4.49 (DC _D , DC _I , LCC _D , LCC _I) Örnek 3 Sınıf A ilişki grafiği.....	89
Şekil 4.50 (DC _D , DC _I , LCC _D , LCC _I) Örnek 3 Sınıf A'ya ait G _D	89
Şekil 4.51 (DC _D , DC _I , LCC _D , LCC _I) Örnek 3 Sınıf A'ya ait G _I	89
Şekil 4.52 (DC _D , DC _I , LCC _D , LCC _I) Örnek 3 Sınıf B ilişki grafiği.....	90
Şekil 4.53 (DC _D , DC _I , LCC _D , LCC _I) Örnek 3 Sınıf B'ye ait G _D	90
Şekil 4.54 (DC _D , DC _I , LCC _D , LCC _I) Örnek 3 Sınıf B'ye ait G _I	91
Şekil 6.1 Örnek bir doğrudan bağlantı grafiği G _{x1}	106
Şekil 6.2 Örnek bir dolaylı bağlantı grafiği G _{x2}	106
Şekil 6.3 Örnek bir metot çağırımı grafiği G _{xA}	106
Şekil 6.4 Örnek bir doğrudan üye alan kullanımı grafiği G _{xB}	107
Şekil 6.5 Örnek bir nihai ilişki grafiği G _x	107
Şekil 6.6 Maximum ilişki grafiği örnekleri	111
Şekil 6.7 Minimum ilişki grafiği örnekleri.....	112
Şekil 6.8 Dolaylı bağlantı değerlendirmesi için grafik örnekleri	113
Şekil 6.9 Etkileşim değerlendirmesi için grafik örnekleri.....	115
Şekil 6.10 LCOM ölçütleri ve COMIAS değerlendirmesi için grafik örnekleri	116
Şekil 6.11 LCOM3 ölçütü a ve b G _x grafikleri.....	117
Şekil 6.12 LCOM4 ölçütü a ve b G _x grafikleri.....	117
Şekil 6.13 Program arayüzü	123

Şekil 6.14 Sürümler için elde edilen gerçek verilerin grafiksel gösterimi	125
Şekil 6.15 Program sürümleri için elde edilen sonuçların grafiksel gösterimi.....	126
Şekil 6.16 Araçlar paketi için elde edilen sonuçların grafiksel gösterimi	126
Şekil 6.17 Yapı paketi için elde edilen sonuçların grafiksel gösterimi	127
Şekil 6.18 İşlemler paketi için elde edilen sonuçların grafiksel gösterimi.....	128
Şekil 6.19 İstatistik paketi için elde edilen sonuçların grafiksel gösterimi	128
Şekil 6.20 LCOM5 ölçütüne göre metot-üye alan ilişkisi ve $\mu(A_j)$ değerleri (0.6.4 sürüm İkiliKokOkuyucu sınıfı).....	142
Şekil 6.21 Doğrudan bağlantı grafiği G_{x1} değerleri (0.6.4 sürüm İkiliKokOkuyucu sınıfı) ..	142
Şekil 6.22 Doğrudan üye alan kullanımı grafiği G_{xB} değerleri (0.6.4 sürüm İkiliKokOkuyucu sınıfı)	143
Şekil 6.23 Nihai ilişki grafiği G_x değerleri (0.6.4 sürüm İkiliKokOkuyucu sınıfı).....	143
Şekil 6.24 LCOM5 ölçütüne göre metot-üye alan ilişkisi ve $\mu(A_j)$ değerleri değerleri (2.2.1 sürüm İkiliKokOkuyucu sınıfı).....	144
Şekil 6.25 Nihai ilişki grafiği G_x değerleri (2.2.1 sürüm İkiliKokOkuyucu sınıfı).....	144
Şekil 6.26 Nihai ilişki grafiği G_x değerleri (0.6.4 sürüm jarowinkler sınıfı).....	146

ÇİZELGE LİSTESİ

Çizelge 4.1 CAMC (1) şikkına göre metotlar ve parametre oluşum matrisi	52
Çizelge 4.2 CAMC (2) şikkına göre metotlar ve parametre oluşum matrisi	52
Çizelge 4.3 CAMC (3) şikkına göre metotlar ve parametre oluşum matrisi	53
Çizelge 4.4 NHD (1) şikkına göre parametre uyuşma matrisi	55
Çizelge 4.5 NHD (2) şikkına göre parametre uyuşma matrisi	55
Çizelge 4.6 NHD (3) şikkına göre parametre uyuşma matrisi	55
Çizelge 4.7 Uyum ölçütlerinin örneklerde verdiği sonuçlar.....	97
Çizelge 4.8 Uyum ölçütlerine genel bir bakış	98
Çizelge 5.1 Sınıflandırma kriterleri	99
Çizelge 5.2 Bağlantı tipleri.....	102
Çizelge 6.1 Metotlar arasında maksimum çağrım ilişkisi örneği	109
Çizelge 6.2 Metotlar arasında maximum üye alan paylaşımı ilişkisi örneği.....	110
Çizelge 6.3 Maximum ilişki grafiği 6.6.a için metot çağrım ve üye alan paylaşım durumu .	111
Çizelge 6.4 Maximum ilişki grafiği 6.6.b için metot çağrım ve üye alan paylaşım durumu .	111
Çizelge 6.5 Minimum ilişki grafiği 6.7.a için metot çağrım ve üye alan paylaşım durumu ..	112
Çizelge 6.6 Minimum ilişki grafiği 6.7.b için metot çağrım ve üye alan paylaşım durumu..	112
Çizelge 6.7 Dolaylı bağlantı hesaba katılmadan 6.8.a grafiği için metot çağrım ve üye alan paylaşım durumu.....	113
Çizelge 6.8 Dolaylı bağlantı hesaba katılmadan 6.8.b grafiği için metot çağrım ve üye alan paylaşım durumu.....	114
Çizelge 6.9 Dolaylı bağlantı hesaba katılırsa 6.8.b grafiği için metot çağrım ve üye alan paylaşım durumu.....	114
Çizelge 6.10 6.9.a grafiği için metot çağrım ve üye alan paylaşım durumu	115
Çizelge 6.11 6.9.b grafiği için metot çağrım ve üye alan paylaşım durumu	115
Çizelge 6.12 COMIAS ölçütünün Weyuker kriter kümesine göre değerlendirilmesi.....	122
Çizelge 6.13 Zemberek programının farklı sürümleri için gerçek veriler [AHMET AFŞİN AKIN]	124
Çizelge 6.14 Zemberek programının ilerleyen sürümleri için COMIAS ölçütü ile elde edilen sonuçlar	125
Çizelge 6.15 Araçlar paketi için COMIAS ölçüm sonuçları	126
Çizelge 6.16 Yapı paketi için COMIAS ölçüm sonuçları	127
Çizelge 6.17 İşlemler paketi için COMIAS ölçüm sonuçları.....	127
Çizelge 6.18 İstatistik paketi için COMIAS ölçüm sonuçları	128

Çizelge 6.19 UlamaOzelDurumu sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım yokken).....	131
Çizelge 6.20 YumusamaNK sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım yokken)	131
Çizelge 6.21 FiilKokBozulmasi sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım yokken)	132
Çizelge 6.22 UlamaOzelDurumu sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım varken).....	133
Çizelge 6.23 YumusamaNK sınıfı için metot çağrım ve üye alan paylaşım durumu (eklenen ilişkiler)	134
Çizelge 6.24 FiilKokBozulmasi sınıfı için metot çağrım ve üye alan paylaşım durumu (eklenen ilişkiler)	135
Çizelge 6.25 TemelRaporlayici sınıfı için metot çağrım ve üye alan paylaşım durumu.....	136
Çizelge 6.26 DosyaRaporlayici sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım yokken)	137
Çizelge 6.27 KonsolRaporlayici_yeni sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım yokken).....	138
Çizelge 6.28 DosyaRaporlayici sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım varken)	139
Çizelge 6.29 KonsolRaporlayici sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım varken).....	139
Çizelge 6.30 KonsolRaporlayici_yeni sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım varken).....	140
Çizelge 6.31 Temel sınıfı KokOzelDurumu olan türetilmiş sınıflar için elde edilen değerler	140
Çizelge 6.32 Temel sınıfı TemelRaporlayici olan türetilmiş sınıflar için elde edilen değerler	140
Çizelge 6.33 Metot çağrım ve üye alan paylaşım durumu değerleri (0.6.4 sürüm İkiliKokOkuyucu sınıfı).....	143
Çizelge 6.34 Metot çağrım ve üye alan paylaşım durumu değerleri (2.2.1 sürüm İkiliKokOkuyucu sınıfı).....	145
Çizelge 6.35 İkiliKokOkuyucu sınıfı ölçüm sonuçları.....	145
Çizelge 6.36 Metot çağrım ve üye alan paylaşım durumu değerleri (0.6.4 sürüm jarowinkler sınıfı)	147
Çizelge 6.37 $a = 0$ olan sınıflar için ölçüm sonuçları	148
Çizelge 6.38 COMIAS ölçütü sınıflandırması	148

Çizelge Ek1.1 LCOM1 ölçütü sınıflandırması.....	155
Çizelge Ek1.2 LCOM2 ölçütü sınıflandırması.....	156
Çizelge Ek1.3 LCOM3 ölçütü sınıflandırması.....	157
Çizelge Ek1.4 LCOM4 ölçütü sınıflandırması.....	158
Çizelge Ek1.5 Co ölçütü sınıflandırması.....	159
Çizelge Ek1.6 TCC ölçütü sınıflandırması.....	160
Çizelge Ek1.7 LCC ölçütü sınıflandırması.....	161
Çizelge Ek1.8 LCOM5 ölçütü sınıflandırması.....	162
Çizelge Ek1.9 ICH ölçütü sınıflandırması.....	163
Çizelge Ek1.10 RCI ölçütü sınıflandırması.....	164
Çizelge Ek1.11 Coh ölçütü sınıflandırması.....	165
Çizelge Ek1.12 CBMC ölçütü sınıflandırması.....	166
Çizelge Ek1.13 CAMC ölçütü sınıflandırması.....	167
Çizelge Ek1.14 NHD ölçütü sınıflandırması.....	168
Çizelge Ek1.15 CC(X) ölçütü sınıflandırması.....	169
Çizelge Ek1.16 ICBMC ölçütü sınıflandırması.....	170
Çizelge Ek1.17 CCM ölçütü sınıflandırması.....	171
Çizelge Ek1.18 ECCM ölçütü sınıflandırması.....	172
Çizelge Ek1.19 OCC ölçütü sınıflandırması.....	173
Çizelge Ek1.20 PCC ölçütü sınıflandırması.....	174
Çizelge Ek1.21 OL _n ölçütü sınıflandırması.....	175
Çizelge Ek1.22 DC _D , DC _i , LCC _D ve LCC _i ölçütleri sınıflandırması.....	176
Çizelge Ek1.23 WTCoh ve ClassCoh ölçütleri sınıflandırması.....	177

ÖNSÖZ

Tez çalışmalarımın gerçekleşmesinde çok büyük desteği olan tez danışmanım Sayın Prof. Dr. Oya KALIPSIZ'a ve Sayın Yrd. Doç. Dr. Yunus Emre SELÇUK'a, yüksek lisans programım süresince bana verdikleri eğitim ve ilgileri nedeniyle Yıldız Teknik Üniversitesi Elektrik-Elektronik Fakültesi Bilgisayar Mühendisliği bölümünde görevli olan tüm öğretim üyelerine ve asistanlarına teşekkür ederim.

Yüksek lisans eğitimim süresince bana hep destek olan ve sabır gösteren; gösterdikleri sevgi, destek ve özveriyle bana kuvvet veren eşime ve anneme en içten dileklerle teşekkür ederim.

Yüksek lisans eğitimim süresi içerisinde dünyaya gelen ve varlığıyla bana güç veren oğluma en içten dileklerle teşekkür ederim.

ÖZET

Yazılım kalitesi kavramı, halen belirsiz ve farklı kişilere farklı anlamlar ifade eden bir kavramdır. Geliştiricinin bakış açısıyla ya da içsel bakış açısı ile kalite, maliyet ve gecikmenin doğru tahminine, daha kolay teste, daha iyi bakıma giden yoldur (Baldassari vd., 2004). Kullanıcı tarafından ya da dışsal bakış açısı ile bakıldığında ise kullanım kolaylığı, estetik, anlaşılabilirlik vb. kavramlar ifade edebilmektedir.

Yazılım uyumu (cohesion), yazılımın içsel kalite özellikleri arasında önemli bir yere sahiptir. Nesneye Yönelik Programlama yaklaşımı için önemli bir kalite faktörü olan yazılım uyumu, bir modülün elemanlarının birbirine aitliğinin derecesi olarak tanımlanabilir. Bir sınıftaki metotların uyumlu olması tercih edilen durumdur. Yüksek uyum sayesinde yeniden kullanılabilirlik ve anlaşılabilirlik artar. Uyum eksikliği ise, sınıfların iki veya daha fazla alt sınıfa ayrılmaları gerekliliğini gösterir. Düşük uyum, karmaşıklığı arttırarak geliştirme sürecindeki hataların olma olasılığını arttırır.

Literatürde yazılım kalitesini değerlendirebilmek amacıyla yazılım kalite faktörlerini göz önüne alan ölçütler önerilmiştir. Yazılım kalitesinin ölçümü yazılımın birçok yönden değerlendirilmesi ve iyileştirilmesine olanak tanınması nedeniyle önem arz etmektedir.

Yazılım kalitesi faktörlerinden *uyum* konusunda, yazılımların uyumunu değerlendirebilmek amacıyla kullanım amaçları esas alınarak ve farklı yaklaşımlar kullanılarak pek çok ölçüt/ölçüt kümesi önerilmiştir. Ölçütlerin çoğunun deneysel olarak geçerlenmemesi ve halen standart kabul edilen ölçütlerin bulunmayışı bu alanda yeni çalışmalara ihtiyaç olduğunu göstermektedir.

Bu tez kapsamında literatürde bulunan mevcut yapısal uyum ölçütleri incelenmiş, örnekler yapılarak değerlendirilmiştir. Ölçütlerin incelenmesiyle yeni sınıflandırma kriterleri belirlenmiş ve her bir ölçütün belirlenmiş sınıflandırma kriterlerine göre bir sınıflandırması yapılmıştır.

Bu tez çalışmasında nesneye yönelik yazılımların yaşam döngüsünün her aşamasında kullanılabilecek bir *uyum* ölçütü olan COMIAS (Cohesion Method Attribute Sharing) tanıtılmakta, teorik geçerlemesi yapılmakta ve ilk deneysel geçerlemesine yer verilmektedir.

ABSTRACT

The software quality engineering notion covers many aspects, all along the software lifecycle. On the developer's side, quality is the way to costs and delays accurate estimation, easier testing and better maintainability. From the user's view or external point of view, the quality is the usability, esthetic, understandability etc.

Cohesion has an important place among the software internal quality factors. Cohesion, which is an important quality factor for Object Oriented Programming paradigm, can be described as the degree of connectivity among the elements of a single module. The cohesion of methods is desired as high cohesion increases reusability and understandability. The lack of cohesion implies the need of the class to be split into two or more classes. The low cohesion increases the complexity and thus causing the increase in the probability of occurrence of the errors during the development phase.

Metrics considering software quality factors have been proposed in the literature in order to evaluate the quality of the software. The measure of software quality is important because it gives opportunity to investigate software from a various side and its improvement

Using different approaches and for different aims, many metrics or metric sets have been proposed in order to evaluate the software cohesion. The lack of metrics' validation and the lack of metrics accepted as standard indicate that new studies are required.

The structural cohesion metrics in the literature have been investigated and evaluated by doing examples in this thesis scope. By the examination of metrics, the new classification criteria determined and the classification of each metric has been made according to these defined classification criteria.

In this thesis, COMIAS (Cohesion Method Attribute Sharing) which is a cohesion metric and can be used in every phase of object oriented software life cycle is introduced. The theoretical validation of COMIAS and its first empirical validation are also given in this study.

1. GİRİŞ

Günümüzde yazılım dünyasında gerçekleşen sistemlerde, yazılım kalitesi kavramı çok önemli rol oynamaktadır. Çünkü kaliteli bir yazılımın, müşteri devamlılığı gibi stratejik önem sağlamasından insan hayatını korumaya kadar giden etkisi ve sonucu vardır.

Son yıllarda yazılım kalitesi ölçütleri yazılım dünyasında çok önem kazanmaktadır. Ölçütler, yazılım geliştiricilere ve yöneticilere yazılımın birçok yönden değerlendirilmesine imkan tanır. Bu değerlendirmeler, iyi ve kaliteli bir yazılım geliştirme yolunda yardımcı rol oynarlar.

Ölçme eylemi, somut veya soyut bir varlığın sahip olduğu bir özelliğini, sayısal veya derecelendirilmiş bir veri olarak ifade etmektir. Ölçülecek özelliği ölçme yolu ve biçimine ölçüt, bir ölçüt kullanılarak yapılan ölçme eyleminin sonucunda elde edilen veriye ise ölçüm adı verilir. Ölçme eylemi işimize yarayacak, anlamlı sonuçlar elde etmek için yapılır. Amaç yazılımın kalitesini iyileştirmek olarak belirlendiği zaman, yazılımın kalitesi yazılım yaşam döngüsünün çeşitli aşamalarında ölçülmeli, elde edilen ölçümler karşılaştırılmalı ve değerlendirilmelidir.

Ölçme yapabilmek için önce bir veya birkaç ölçüt belirlenmelidir. Bu aşamada tanımlanması bile zor bir kavram olan yazılım kalitesi ile ilgili üzerinde anlaşmaya varılmış özellikler bulmanın zorluğu ile karşılaşılır. Yazılım yaşam döngüsünün erken aşamalarında ölçüme başlayabilmek için, içsel kalite özelliklerinin kullanılması gerekecektir. Bağlaşım (coupling) (Stevens vd., 1974) ve uyum (cohesion) (Bunge, 1977), bu amaçla kullanılan en temel özelliklerdir. Nesneye Yönelik Programlama (NYP) yaklaşımının doğru kullanımı ile oluşturulan yazılımlarda bağlaşımın düşük ve uyumun yüksek olması beklenir (Lieberherr vd., 1988). Literatürde bağlaşım ve uyum için birçok ölçüt bulunmaktadır ancak yazılım ile ilgili ölçütlerin deneysel açıdan doğrulanması kolay olmadığından bu yönde daha çok çalışma yapılmasına gerek vardır (Briand vd., 1999).

Ölçüm verilerini anlamlı bilgilere dönüştürmek için yapılması gereken yorumlama işlemi zor olabilir. Literatürde “intelligence barrier” (Kriz, 1988) olarak adlandırılan bu zorluk, matematiksel ve istatistiksel yöntemlerle aşılmaya çalışılabilir (Ebert ve Dumke, 2007), ancak bu yöntemlerin de kolay olduğu söylenemez.

Değinilen bütün zorluklara rağmen, yazılım geliştiricileri ve yöneticileri için yazılımın birçok yönden değerlendirilmesine ve iyileştirilmesine olanak tanımları nedeniyle, yazılım ölçütleri üzerinde çalışmalar etkin olarak sürmektedir. Bağlaşım ve uyumun ölçülmesi için önerilen ölçüt modelleri incelendiğinde, bağlaşım ölçümü açısından belirli bir olgunluğa erişilmesine

rağmen, uyum ölçümünde aynı düzeye ulaşamadığı değerlendirilmiştir. Bu nedenle bu tez çalışmasında COMIAS (Cohesion Method Invocation Attribute Sharing) adı verilen bir uyum ölçütü gerçekleştirilmiştir.

1.1 Uyum

Bunge 1977 yılında, 2 şeyin benzerliğini, benzerlik $\sigma()$, 2 şeyin özellikleri kümelerinin kesişimi şeklinde tanımlar (Chidamber ve Kemerer, 1994).

Tanım 1.1:

$$\sigma(X,Y) = p(x) \cap p(y) \quad (1.1)$$

Benzer ifadeyle, benzerlik, metotlar tarafından kullanılan üye alan kümelerinin kesişimi olarak tanımlanır.

Tanım 1.2:

$$\sigma(M1,M2) = \{I1\} \cap \{I2\} \quad (1.2)$$

$\sigma(M1, M2) = M1$ ve $M2$ metotlarının benzerlik derecesi,

$\{I1\} = M1$ metodu tarafından kullanılan üye alanların kümesi.

Örnek 1.1:

$\{I1\} = \{a,b,c,d,e\}$ ve $\{I2\} = \{a,b,e\}$. $\{I1\} \cap \{I2\}$ boş küme değil ve $\sigma(M1, M2) = \{a,b,e\}$.

Metotların benzerlik derecesi hem yazılım mühendisliğindeki geleneksel uyum düşüncesi (ilgili şeyleri bir arada tutma) hem de metotların ve üye alanların bir nesnede toplanması ile (encapsulation) ilişkilidir. Yüksek benzerlik derecesi, metotların yüksek benzerliği ve nesnenin yüksek depolanma derecesi anlamına gelir (Chidamber ve Kemerer, 1991).

Stevens ve arkadaşları ilk uyum tanımlamasını yapan ekiptir ve şöyle tanımlamışlardır:

Tanım 1.3:

Uyum, bir modülün elemanlarının birbirine aitliğinin derecesinin ölçümüdür.

Uyum eksikliği sınıfların bir veya daha fazla alt sınıfa ayrılmaları gerekliliğini gösterir.

Metotların birbirinden çok ayırık olmaları sınıfların tasarımında hata olduğunu gösterir. Düşük uyum karmaşıklığı ve dolayısıyla geliştirme sürecindeki hataların olma olasılığını artırır.

Uyum kalite faktörünün sağlanması açısından ölçüt çalışması adı altında kullanım amaçları esas alınarak ve farklı yaklaşımlar kullanılarak pek çok çalışma yapılmıştır. Bu yaklaşımlar altta yatan mekanizmaya göre farklılık gösterir.

Yazılım uyum ölçütlerinin kullanım amaçlarına şu örnekler verilebilir:

- Yazılım kalitesinin değerlendirilmesi (Bansiya ve Davis, 2002) (Briand vd., 2000)
- Yazılım kalitesinin ve hata eğilimliliğinin tahmini (El-Emam, 2002) (Quah ve Thwin, 2003)
- Yazılım modülleştirilmesi (Brito e Abreu ve Goulao, 2001) (Maletic ve Marcus, 2001)
- Tekrar kullanılabilir olan bileşenlerin belirlenmesi (Etzkorn ve Davis, 1997) (Lee vd., 2001)

“uyum” yazılım kalitesi faktörünü ölçmek için kullanılan bazı yaklaşımlar ise şunlardır:

- Yapısal uyum ölçütleri
- Semantik uyum ölçütleri (Etzkorn ve Delugach, 2000)
- Entropi tabanlı uyum ölçütleri (Allen vd., 2001)
- Slice-based ölçütler (Meyers ve Binkley, 2004)
- Veri Madenciliğine dayalı ölçütler (Montes de Oca ve Carver, 1998):
- Bilgi Tabanlı Sistemler (Kramer ve Kaindl, 2004) ve Dağıtık Sistemler (Cho vd., 1998) gibi özelleştirilmiş yazılım uygulamaları için ölçütler

Bu tez çalışmasında değinilecek olan yaklaşım Yapısal Uyum Ölçütleri'dir. Yapısal uyum ölçütleri alanında birçok çalışma yapılmıştır. Yapısal ölçütler temel olarak metotlar arasındaki sınıf değişken referanslaması ve paylaşımını göz önüne almaktadır. Sınıfın metotları arasındaki ilişkiler bu düşünce ile tanımlanır.

1.2 Tez Organizasyonu

Tezin geri kalan kısmı şu şekilde organize edilmektedir;

2. bölümde yazılım uyum ölçütlerinin kullanım amaçları anlatılmaktadır.

3. bölümde “uyum” yazılım kalitesi faktörünü ölçmek için kullanılan yaklaşımlar

anlatılmaktadır.

4. bölümde mevcut Yapısal Uyum Ölçütleri örneklerle birlikte incelenmekte ve elde edilen bulgular belirtilmektedir.

5. bölümde mevcut Yapısal Uyum Ölçütleri'nin sınıflandırılması yapılmaktadır.

6. bölümde yeni önerilen ölçüt anlatılmaktadır ve 7. bölümde elde edilen sonuç verilmektedir.

7.2. Yapısal Kuvvetlilik ve Hata Eğilimliliğinin Tahmini

Yapısal Kuvvetlilik ve Hata Eğilimliliğinin tahmini için yapılan bir çalışmada (Li-Ernang, 2002), NY ölçütleri incelenmektedir. Elde edilen sonuçlardan bazıları şunlardır:

- Çoğu ölçüt erken tasarım safhasında toplanabiliyor ve erken kalite yönünü sağlıyor. Yüksek değere sahip bağlam (coupling) ölçümlerine sahip sınıflar en iyi sonuçları verebiliyor gibi düşünceler varabiliyor.
- Eğer tarih verisi mevcutsa, sınıflar hata eğilimliliklerini ve erken kalite sonuçlarına göre değerlendirilmek için olmaktadır. Bunun sonucu olarak da hata eğilimliliği en yüksek sınıflar en iyi sonuçları verebiliyor gibi düşünceler çıkıyor.
- Yüksek hata eğilimliliği sahip sınıflar genelde düşük kaliteli malzeme ekipman kullanıyor ve bu ekipman için daha fazla test zamanı gerektiriyor.
- Yeni olan uyum ölçütleri hata eğilimliliği tahmininde iyi değillerdir.

Yapısal Kuvvetlilik ve Hata Eğilimliliğinin tahmini için yapılan diğer bir çalışmada (Quah ve Chan, 2003) NY ölçütleri kullanılarak yaygın sınıflar ağlarının (mental network) uygulanması incelenmektedir. Katma katmanlı yapılar bütün yapılabiliyor ya da sınıfların birbiriyle ilişkisi de diğer ölçütlerle birlikte değerlendirilerek hangi sınıfların en iyi sonuçlara ulaşabileceği araştırılmaktadır. Bu için bir sınıfların hata eğilimliliği için sınıfların birbiriyle ilişkisi değerlendirilmektedir. Bu çalışmada, 2 yapısal uyum ölçütü kullanılmaktadır. Ward ve General Regression (GRR) Kuvvetlilik ile ilgili ölçütler, kuvvetlilik ölçütleri olarak kullanılmaktadır. Bağlam ölçütleri ve diğer yapısal uyum ölçütleri

2. YAZILIM UYUM ÖLÇÜTLERİNİN KULLANIM AMAÇLARI

2.1 Yazılım Kalitesinin Değerlendirilmesi

Yazılım kalitesinin değerlendirilmesi için yapılan bir çalışmada (Bansiya ve Davis, 2002), Nesneye Yönelik (NY) tasarımlarda yüksek-seviye tasarım kalitesi özelliklerini değerlendirmek için iyileştirilmiş bir hiyerarşik model tanımlanır. Bu modelde sınıfların, nesnelerin ve ilişkilerinin yapısal ve davranışsal tasarım özellikleri NY tasarım ölçütleri kullanılarak değerlendirilmektedir.

Diğer bir çalışmada (Briand vd., 2000) amaç literatürdeki NY tasarım ölçütlerinin özellikle hata eğilimliği konusunda deneysel geçerliliğini gerçekleştirmek olmuştur. Elde edilen bir sonuç; literatürdeki çoğu ölçütün karşılaştırılabilir fikirlere ve hipotezlere dayalı olduğu ve bu nedenle gereksiz olabilecekleri görüşüdür. “uyum”, hata eğilimliği üzerinde anlamlı bir etkiye sahip gözüküyor olarak değerlendirilmiştir.

2.2 Yazılım Kalitesinin ve Hata Eğilimliliğinin Tahmini

Yazılım kalitesinin ve hata eğilimliliğinin tahmini için yapılan bir çalışmada (El-Emam, 2002), NY ölçütler incelenmektedir. Elde edilen sonuçlardan bazıları şunlardır;

- Çoğu ölçüt erken tasarım safhasında toplanabiliyor ve erken kalite yönetimi sağlıyor. Yüksek değere sahip bağlaşım (coupling) ölçütlerine sahip sınıflara en iyi insanları atayabilirsiniz gibi öneriler verebiliyor.
- Eğer tarih verisi mevcutsa, sınıfları hata eğilimliliklerine ve tahmin edilen sonuçlarına göre derecelendirmek iyi olacaktır. Bunun sonucu olarak da hata eğilimliliği en yüksek tahmin edilen sınıflara en iyi insanları atayabilirsiniz gibi öneriler çıkıyor.
- Yüksek hata eğilimliliğine sahip sınıflara geniş ve deneyimli inceleme ekipleri atamalı ve bu sınıflar için daha fazla test adımı geliştirmeli.
- Var olan uyum ölçütleri hata eğilimliliği tahmininde iyi değildirler.

Yazılım kalitesinin ve hata eğilimliliğinin tahmini için yapılan diğer bir çalışmada (Quah ve Thwin, 2003), NY ölçütler kullanılarak yapay sinir ağlarının (neural network) uygulanması incelenmektedir. Kalite tahmini yazılım bakım yapılabilirliğinin yanı sıra güvenilirliğinin tahminini de içerir. Güvenilirlik hata sayısı olarak bakım çabası ise sınıf başına değişen satır sayısı olarak ölçülebilir. İlki için bir sınıftaki hata sayısını ikincisi için sınıf başına düşen değişiklik sayısını tahmin etmek gerekir. Bu çalışmada, 2 yapay sinir ağ modeli kullanılmaktadır, Ward ve General Regression (GRRN). Kalıtım ile ilişkili ölçütler, karmaşıklık ölçütleri, uyum ölçütleri, bağlaşım ölçütleri ve bellek yerleştirme ölçütleri

bağımsız değişken olarak kullanılmaktadır. GRRN ağ modelinin Ward 'dan daha doğru tahminler yapabildiği görülmüştür.

2.3 Yazılım Modülleştirilmesi

Yazılım modülleştirilmesi için yapılan bir çalışmada (Brito e Abreu ve Goulao, 2001), daha iyi modülleştirme çözümlerini elde etmek için kümeleme analizi tekniklerini ve sınıf bağlaşımı (class-coupling) bilgisini kullanan bir yaklaşım önerilmektedir. Bu yaklaşım yapısal bağlaşımı göz önünde bulunduruyor fakat orijinal sistem modülleştirilmesinin arkasındaki semantiğin bir kısmını da kullanmaktadır. En iyi çözümü bulmak için birkaç kümeleme tekniği ve bağlaşım bilgisinin birkaç kombinasyonu denenmiştir. Buradaki yaklaşım hem tasarım safhasında hem de yeniden yapılamada (reengineering) yanlış modülleştirmeleri ve alternatif olanları belirlemek için kullanılabilir. Bu çalışmada geliştirilen araç, kalite anlamında yararlı bir destek sağlamaktadır.

Yazılım modülleştirilmesi için yapılan diğer bir çalışmada (Maletic ve Marcus, 2001), programların semantiğinin ve yapısal bilgisinin birlikte kullanımının, bakım ve yeniden yapılamada'daki (reengineering) işleri desteklemesi sorgulanmaktadır. Burada semantik, domain (problem ve geliştirme) özgü meseleleri, yapısal ise programın gerçek söz dizimsel (syntactic) yapısını temsil etmektedir. Önemli bir bilgi alma metodu olan "Latent Semantic Indexing", yazılım bileşenleri arasındaki semantik benzerliği tanımlamada kullanılmaktadır. Daha sonra Yazılım sistemi içindeki bileşenler benzerlik ölçülerine göre kümelenmektedir. Basit yapısal bilgi (ör: Dosya organizasyonu) daha sonra kümelerin ve dosyaların semantik uyumu açısından değerlendirilmesinde kullanılmaktadır. Yapılan deneyler sonucunda; kaynak kod dokümanlarının semantik benzerliğinin yazılım bakımı ve evrimi konularında değerli bilgiler sağladığı görülmüştür.

2.4 Tekrar Kullanılabilir olan Bileşenlerin Belirlenmesi

Tekrar kullanılabilir olan bileşenlerin belirlenmesi ile ilgili yapılan bir çalışmada (Etzkorn ve Davis, 1997), şu konulara değinilmektedir; NY kod kalıtım yoluyla fonksiyonel koddan daha fazla tekrar kullanılabilir yapıda olmasına rağmen çoğu NY kod tekrar kullanım mantığı ile yazılmamaktadır, bu da yararlı bileşenleri belirlemeyi zorlaştırmaktadır. Bu çalışmada, Patricia sistemi adıyla bir sistem geliştirilmiştir. Bu sistem, sezgisel bir yaklaşım kullanarak otomatik olarak yorumları ve belirteçleri anlayarak bu bileşenleri belirlemektedir.

Tekrar kullanılabilir olan bileşenlerin belirlenmesi ile ilgili yapılan diğer bir çalışmada (Lee vd., 2001), bileşen bağlaşımı, uyumu, bağlılığı, arayüzü, granüllülüğü ve mimariyi göz önünde bulunduran bileşen belirleme metodu ortaya konmuştur.

3.1 Yapısal Uyum Ölçütleri

Yapısal ölçütlerin arkasındaki düşünce, metodlar arasındaki sınıf değişken referanslarının paylaşımına göre değişmektedir. Çoğu yapısal ölçütler, sınıf metodları arasındaki ilişkileri bu düşünce ile tanımlar ve ölçer. Uygunluk, üye alanlarını veya sınıf değişkenlerini paylaşan metod çiftlerinin sayısına dayandığı görülmüştür. Yapısal ölçütlerin arasındaki farklar, metodlar için farklı türlerde, sınıfların tanımı ve soyut mekanizmasına dayanmaktadır.

Yapısal uyum ölçütleri ile ilgili yapılan çalışmalar Bölüm 4'de anlatılmaktadır.

3.2 İçeriksel Uyum Ölçütleri

Yapılan bir çalışmada (Eitzorn ve Dehagech, 2000) sınıf benzerliğini, sınıf tanımlı kümeleri benzerliğini, anlamlı sınıf belirlenmesini, sınıf arayüz karmaşıklığını, sınıf keskinliğini ve sınıfların dokümantasyon kalitesini ölçen manüskel tasarım ölçütleri önerilmiştir.

Sınıf benzerliği:

- Metodların Manüskel Benzerliği (Logical Relatedness of Methods (LORM)): sınıfın metodlarına karşılıklı benzerlikleri ölçer. Bu ölçümü toplamak için uzman sistemi kullanarak seçilen sınıflar uygulanmıştır.
- LORM, sınıfların toplam ilişki sayısı / sınıfları mümkün olan toplam ilişki sayısı şeklinde hesaplanır.
- LORM2, benzerlik ilişkisinde görünen kavramları inceliyerek fonksiyoneliğin diğer benzerliğini ölçer.
- LORM3, benzerlik benzerliği ölçer.

Sınıf tanımlı kümeleri karmaşıklığı

- Sınıf Tanımlı Kümeleri Karmaşıklığı (Class Domain Complexity (CDC))
- Semantik Sınıf Tanımlı Entropisi (Semantic Class Definition Entropy (SCDE))

Anlamlı sınıf belirlenme (Key Class Identification):

- KCI-0 veya 1 değeri alınır.

Sınıf arayüz karmaşıklığı

- Sınıf Arayüz Karmaşıklığı (Class Interface Complexity (LIC)): CDC'ye benzer fakat sınıfların arayüz metodları ile çalışır.
- Semantik sınıf Arayüz Karmaşıklığı (Semantic Class Interface Definition Entropy

3. NY SİSTEMLERDE UYUMU ÖLÇMEK İÇİN KULLANILAN YAKLAŞIMLAR

NY sistemlerde uyumu ölçmek için kullanılan farklı yaklaşımlar vardır. Bu yaklaşımlar altta yatan mekanizmaya göre şu şekilde ayrıştırılabilir (Poshyvanyk ve Marcus, 2005):

3.1 Yapısal Uyum Ölçütleri

Yapısal ölçütlerin arkasındaki düşünce, metotlar arasındaki sınıf değişken referanslaması ve paylaşımını göz önüne almaktır. Çoğu yapısal ölçütler, sınıfın metotları arasındaki ilişkileri bu düşünce ile tanımlar ve ölçer. Uyumun, üye alanlarını veya sınıf değişkenlerini paylaşan metot çiftlerinin sayısına dayandığı görülür. Yapısal ölçütlerin arasındaki farklar, metotlar arası ilişkilerin tanımına, sistem tanımına ve sayma mekanizmasına dayanmaktadır.

Yapısal uyum ölçütleri ile ilgili yapılan çalışmalar Bölüm 4’de anlatılmaktadır.

3.2 Semantik Uyum Ölçütleri

Yapılan bir çalışmada (Etzkorn ve Delugach, 2000) sınıf benzerliğini, sınıf tanım kümesi karmaşıklığını, anahtar sınıf belirlemesini, sınıf arayüz karmaşıklığını, sınıf kesişmesini ve sınıfın dokümantasyon kalitesini ölçen mantıksal tasarım ölçütleri önerilmiştir.

Sınıf benzerliği:

- Metotların Mantıksal İlişkililiği (Logical Relatedness of Methods (LORM)): sınıfın metotlarının kavramsal ilişkiliğini ölçer. Bu ölçümü toplamak için uzman sistem tarafından semantik analiz uygulanmalıdır.
- LORM, sınıftaki toplam ilişki sayısı / sınıftaki mümkün olan toplam ilişki sayısı şeklinde hesaplanır.
- LORM2, birden fazla fonksiyonda görünen kavramları inceleyerek fonksiyonelliğin üst üste binmesini ölçer.
- LORM3, kavram benzerliğini ölçer.

Sınıf tanım kümesi karmaşıklığı:

- Sınıf Tanım Kümesi Karmaşıklığı (Class Domain Complexity (CDC))
- Semantik Sınıf Tanım Entropisi (Semantic Class Definition Entropy (SCDE))

Anahtar sınıf belirleme (Key Class Identification):

- KCI: 0 veya 1 değerini alır

Sınıf arayüz karmaşıklığı:

- Sınıf Arayüz Karmaşıklığı (Class Interface Complexity (CIC)): CDC’ye benzer fakat sınıfın arayüz metotları üzerinde çalışır.
- Semantik Sınıf Arayüz Tanım Entropisi (Semantic Class Interface Definition Entropy)

(SCIDE)): Arayüz karmaşıklığını ölçer.

Sınıf üst üste binmesi (Class overlap):

Birden fazla sınıfta görünen kavramları inceleyerek 2 sınıf arasındaki fonksiyonelliğin üst üste binmesini ölçerler.

- Class Overlap A (COa)
- Class Overlap B (COb)

Sınıfın dokümantasyon kalitesi (Documentation):

Sınıfın belirteçlerini ve yorumlarını kullanarak ölçüm yaparlar.

- Overall Class Documentation Quality A (OCDQa)
- Overall Class Documentation Quality B (OCDQb)
- Class Comment Quality (CCQ)
- Class Identifier Quality (CIQ)

Bu ölçütlerin bir önceki bölümlerde anlatılan ve geneli sintatik olan ölçütlerden daha doğru sonuçlar verebildikleri görülmüştür (Etzkorn ve Delugach, 2000).

3.3 Entropi Tabanlı Uyum Ölçütleri

Yazılım mimaricileri ve tasarımcıları yazılımın soyut gösterimi için birçok grafik çeşidi kullanırlar. Örneğin “çağrı grafikleri (call graphs)” birçok yazılım geliştirici tarafından potansiyel “çağrı ilişkileri (calling relationships)”ni göstermek için kullanılır. Bilgi teorisi yaklaşımı, bir grafikteki kalıpları (patterns) saymaktan daha farklı bir yaklaşımdır. Ölçüt tanımları, grafiğin elemanlarını saymaktan ziyade ilişki kalıplarını göz önüne almaktadır. Her düğüme bağlı kenarların kalıbı rastgele bir sayı olarak modellenmektedir. Deneysel bir çalışmada (Allen vd., 2001) bir bilgisayar oyunu incelenmekte, kaynak kodu modül olarak bilinen bir fonksiyon çağrı grafiğinin bağlaşım ve uyumu hesaplanmaktadır. Sonuç olarak bu ölçütlerin standart sayı tabanlı ölçütlerle bezer sonuçlar verdiği görülmüştür.

3.4 Dilim Tabanlı Uyum Ölçütleri

Yazılımın yeniden yapılandırılması maliyetli bir iştir. Uyum ölçütleri, belirleme ve ölçme işinde yardım edecek potansiyele sahiptir. Böyle ölçütler de geniş çalışmalarda dilim tabanlı (slice-based) ölçütlerdir.

Mark Weiser (Meyers ve Binkley, 2004) program dilim tabanlı 5 ölçüt önermiştir. Bu ölçütlerin deneysel incelemeleri sonucu bazı katkılarının olduğu görülmüştür. Bunlar;

- Ölçütler, modül seviyesinde kullanılabildikleri için tersine mühendislik (reverse engineering) işinde direk modüle odaklanmayı sağlayarak yardımcı olur. Dilim tabanlı ölçütler için taban (baseline) değerler sağlanmıştır. Bu değerler, reengineering eforunun hedef değer dışındaki modüller üzerinde olması gerektiğini vurgular.
- Dilim tabanlı ölçütlerin programın yaşlandıkça bozulma miktarını belirleyebildikleri görülmüştür.
- Ölçütler, program seviyesinde kullanılabilmek için tersine mühendislik ile kodu iyileştirme işinde rehberlik ederler.

3.5 Veri Madenciliğine dayalı Ölçütler

Yapılan bir çalışmada (Montes de Oca ve Carver, 1998) veri uyumlu (cohesive) alt sistemleri belirlemek için veri madenciliğini kullanan ISA (Identification of Subsystems based on Associations) metodolojisi tanımlanmaktadır. Bu yaklaşımı kullanarak COBOL sistemler alt sistemlere ayrıştırılabilmektedir.

Bu çalışmada, veri madenciliğinin farklı tanecik seviyesinde uygulanabileceği ve framework olarak kullanılabilecek üç adımlı bir metot önerilmektedir. Önerilen adımlar şunlardır;

- Sistemin veri tabanı görünüşünü tanımla
- Veri madenciliğini gerçekleştir
- Sonuçları birleştir ve yorumla

Deneyimler göstermiş ki; veri madenciliği bir yazılım sisteminin mantıksal ayrıştırılmasının üretiminde kullanılabilir. Veri madenciliği, sistemin hiç bir konu ve geçmiş bilgisini bilmeden veri uyumlu alt sistemleri belirleyebilir. Sadece gerekli olan şey kaynak koddur. Veri madenciliği, büyük ve belgelenmemiş sistemlerin analizinde uygun bir yaklaşımdır.

3.6 Bilgi Tabanlı Sistemler ve Dağıtık Sistemler gibi Özelleştirilmiş Yazılım Uygulamaları için Ölçütler

Yapılan bir çalışmada (Kramer ve Kaindl, 2004) Bilgi tabanlı sistemler için uyum ölçütü, ChF (Cohesion of Frame) tanımlanmaktadır. Bu ölçüt bilgi tabanlarının birimselliği (modularity) ile ilgili problemleri adreslemede yardımcı olmuştur. Bilgi tabanlı sistemlerin geliştirici tarafından nicel ve nesnel kalite özelliklerinin sorgulanmasına ve sistemin geliştirilmesine yardımcı olabileceği düşünülmüştür.

İstemci/sunucu temelli ve dağıtık uygulamalarda nesne kalite kümelemesi, sistemin bütününe performansına karar vermede anahtar bir rol oynar. Bu nedenle, yüksek bağlaşıma sahip nesneler tek bir kümede gruplanmalı ki her bir küme yüksek bir uyuma sahip olsun. Sonuç olarak, nesneler arası mesaj trafiği yüksek oranda azaltılabilir. Çoğu ölçütler nesne

mesaj trafik yükünü ölçmek için uygun değildirler. Yapılan bir çalışmada (Cho vd., 1998), çeşitli sınıf ilişkileri üzerindeki ilgili ağırlıkları göz önüne alan ve nesneler arasındaki statik ve dinamik mesaj akışını tahminleyen yeni bir ölçüt kümesi önerilmektedir. Yapılan değerlendirmelerde bu ölçütlerin, dinamik ölçümü de yapabildiği görülmüştür.

Örnek 1 Java kaynak kodlu bir program örneğidir (Şekil 4.1).

```
public class Point
{
    private Boolean Visible;
    private int x, y;

    public Point(int initX, int initY)
    {
        this.x = initX;
        this.y = initY;
        this.Visible = false;
    }

    public int GetX()
    {
        return x;
    }

    public int GetY()
    {
        return y;
    }

    public void SetX(int x)
    {
        this.x = x;
    }

    public void SetY(int y)
    {
        this.y = y;
    }

    public void Show()
    {
        Visible = true;
        System.out.println("X : " + x + " Y : " + y);
    }

    public void Hide()
    {
        Visible = false;
    }

    public void Drag(int DragX, int DragY)
    {
        Show();
        int delta = 1;
        int counter = DragX / delta;
        while (counter > 0 || counter < -1)
        {
            MoveTo(x + delta, y + delta);
        }
    }

    public Boolean IsVisible()
    {
        return Visible;
    }

    public void MoveTo(int NewX, int NewY)
    {
        Hide();
        x = NewX;
        y = NewY;
        Show();
    }
}
```

Şekil 4.1 Bir Java program örneği

4. NY PROGRAMLAMADA YAPISAL BENZERLİK/UYUM ÖLÇÜTLERİ

Bu bölümde literatürde önerilen yapısal uyum ölçütleri anlatılmakta, Örnek 1 ve Örnek 2 üzerinde her bir ölçütün uygulaması yapılarak değerlendirilmeler verilmektedir.

Örnek 1 Java kaynak kodlu bir program örneğidir (Şekil 4.1).

```

public class Point
{
    private Boolean Visible;
    private int x, y;

    public Point(int InitX, int InitY){
        this.x = InitX;
        this.y = InitY;
        this.Visible = false;
    }
    public int GetX(){
        return x;
    }
    public int GetY(){
        return y;
    }
    public void SetX(int x){
        this.x = x;
    }
    public void SetY(int y){
        this.y = y;
    }
    public void Show(){
        Visible = true;
        System.out.println("X : " + x + " Y : " + y);
    }
    public void Hide(){
        Visible = false;
    }
    public void Drag(int DragBy){
        Show();
        int delta = 1;
        int counter = DragBy / delta;
        for (int i = 0; i < counter; ++i){
            MoveTo( x + delta, y + delta );
        }
    }
    public Boolean IsVisible(){
        return Visible;
    }
    public void MoveTo( int NewX, int NewY){
        Hide();
        x = NewX;
        y = NewY;
        Show();
    }
}

```

Şekil 4.1 Bir java program örneği

Şekil 1'deki kod örneğinde bulunan tüm öğeleri inceleyelim:

Sınıf: Point

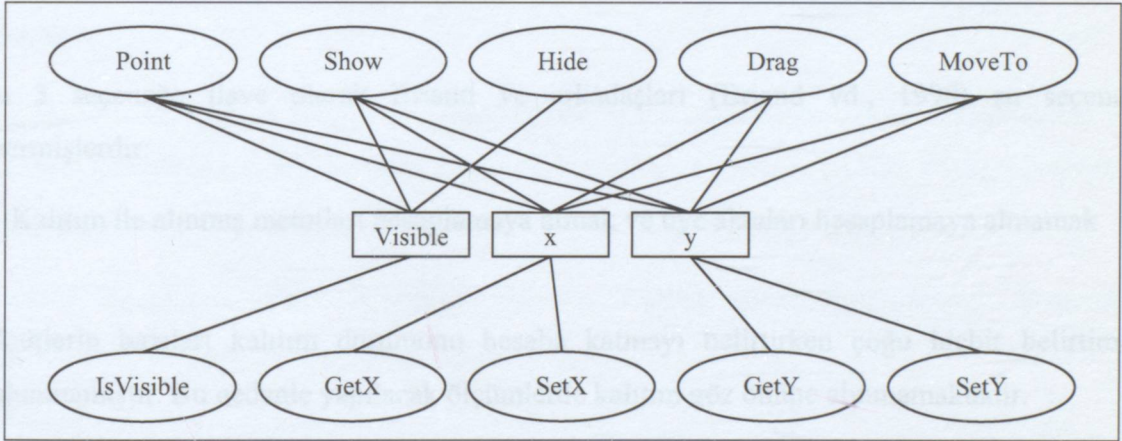
Constructor metot: Point()

Erişim metotları: IsVisible(),GetX(), GetY(), SetX(), SetY()

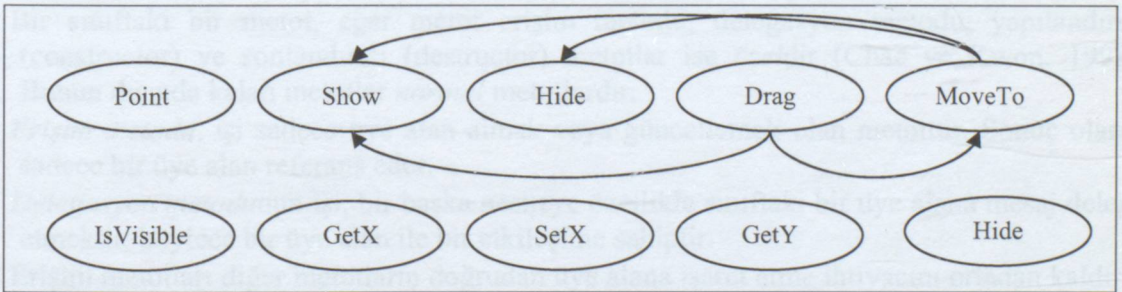
Normal metotlar: Show(), Hide(), Drag(), MoveTo()

Üye alanlar: Visible, x,y

Point sınıfının metotları ile eriştikleri üye alanların grafiksel gösterimi Şekil 4.2'de, metotların birbiriyle ilişkileri ise Şekil 4.3'de verilmektedir. Metot – üye alan ilişkisi gösteriminde metot ve eriştiği üye alan arasında bir çizgi oluşturulmuş, metot – metot ilişkisi gösteriminde ise çağıran metottan çağrılan metoda doğru yönlü bir çizgi oluşturulmuştur.

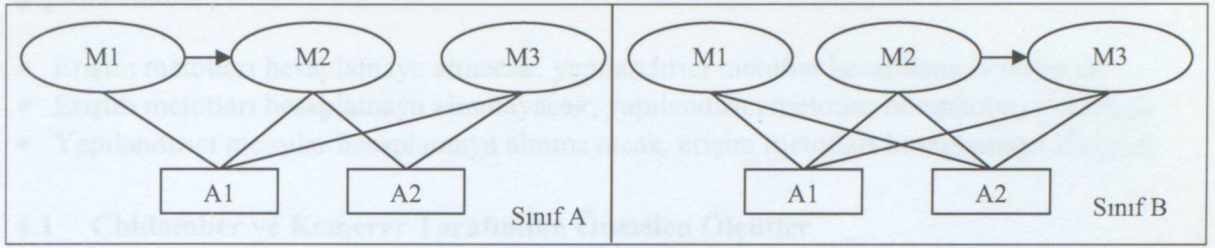


Şekil 4.2 Point sınıfının metot – üye alan ilişkisi grafiği



Şekil 4.3 Point sınıfının metot – metot ilişkisi grafiği

Örnek 2 ise iki adet sınıfın metot – üye alan ve metot – metot ilişkisi diyagramlarının bulunduğu bir örnektir (Şekil 4.4).



Şekil 4.4. Sınıf A ve Sınıf B metot – üye alan ve metot – metot ilişki örneği

Kalıtım ile alınmış metot ve üye alanları ölçüt hesaplamasına almak tamamen isteğe bağlı bir durumdur. Bu konu ile ilgili olarak Bieman ve Kang (Zhou vd., 2002) bir sınıfın uyumunun analizinde kullanılabilecek 3 seçenek önermişlerdir:

- Kalıtım ile alınmış metotları ve üye alanları hesaplamaya almamak
- Kalıtım ile alınmış metotları ve üye alanları hesaplamaya almak
- Kalıtım ile alınmış metotları hesaplamaya almamak ve üye alanları hesaplamaya almak

Bu 3 seçeneğe ilave olarak Briand ve arkadaşları (Briand vd., 1998) şu seçeneği önermişlerdir:

- Kalıtım ile alınmış metotları hesaplamaya almak ve üye alanları hesaplamaya almamak

Ölçütlerin bazıları kalıtım durumunu hesaba katmayı belirtirken çoğu hiçbir belirtimde bulunmamıştır. Bu nedenle yapılacak ölçümlerde kalıtım göz önüne alınmamaktadır.

Metotların yapıları analizi genelde etkilemektedir. Metot yapıları ile ilgili bilgiler şu şekildedir;

- Bir sınıftaki bir metot, eğer metot erişim metodu, delegasyon metodu, yapılandırıcı (constructor) ve sonlandırıcı (destructor) metotlar ise *özeldir* (Chae ve Kwon, 1998). Bunun dışında kalan metotlar *normal* metotlardır.
- *Erişim metodu*, işi sadece üye alan almak veya güncellemek olan metottur. Sonuç olarak sadece bir üye alan referans eder.
- *Delegasyon metodunun* işi, bir başka nesneye özellikle sınıftaki bir üye alana mesaj delege etmektir, böylece bir üye alan ile bir etkileşime sahiptir.
- Erişim metotları diğer metotların doğrudan üye alana işaret etme ihtiyacını ortadan kaldırıp kendi üzerlerinden eriştikleri için yapay olarak *uyumu* düşürürler, yapılandırıcılar çoğu üye alanın ilk değerlerini sağladıkları için yapay olarak *uyumu* artırırlar.

Bu bilgiler ışığında ölçütlerin Örnek 1 ve Örnek 2 uygulamaları yapılmaktadır. Metot tipinin *uyum*'a etkisini değerlendirebilmek amacıyla her bir ölçüt için 3 farklı yönden ölçüm

yapılmaktadır:

- Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak
- Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak
- Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

4.1 Chidamber ve Kemerer Tarafından Önerilen Ölçütler

4.1.1 LCOM1

LCOM1, Chidamber ve Kemerer tarafından 1991 yılında tanımlanan bir ölçüttür (Chidamber ve Kemerer, 1991).

LCOM1 (Lack of Cohesion of Methods) ölçütü, Bunge'in benzerlik ontolojisine (Bunge, 1977) dayanır. Uyum, bu ontolojiye göre "metotların benzerlik derecesi" şeklinde tanımlanmaktadır.

Tanım 4.1:

$M_1, M_2, \dots, M_n$ metoda sahip bir C sınıfı olsun. $\{I_i\} = M_i$ metodu tarafından kullanılan üye alanların kümesi olsun. Bu durumda n tane $\{I_1\}, \{I_2\} \dots, \{I_n\}$ küme olur. LCOM1, bir sınıftaki ortak üye alanı bulunmayan metot çifti sayısı olarak ölçülmektedir.

LCOM1 ölçütü ters uyum ölçümüdür. Yüksek LCOM1 değeri düşük uyumu gösterirken düşük LCOM1 değeri yüksek uyumu gösterir.

LCOM1 Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

M_1 : Point()	M_2 : Show()	M_3 : Hide()	M_4 : Drag()	M_5 : MoveTo()
I_1 : {Visible, x, y}	I_2 : {Visible, x, y}	I_3 : {Visible}	I_4 : {x, y}	I_5 : {x, y}
M_6 : IsVisible()	M_7 : GetX()	M_8 : SetX()	M_9 : GetY()	M_{10} : SetY()
I_6 : {Visible}	I_7 : {x}	I_8 : {x}	I_9 : {y}	I_{10} : {y}

Ortak üye alanı bulunmayan metot çiftleri: $\{M_3, M_4\}, \{M_3, M_5\}, \{M_3, M_7\}, \{M_3, M_8\}, \{M_3, M_9\}, \{M_3, M_{10}\}, \{M_4, M_6\}, \{M_5, M_6\}, \{M_6, M_7\}, \{M_6, M_8\}, \{M_6, M_9\}, \{M_6, M_{10}\}, \{M_7, M_9\}, \{M_7, M_{10}\}, \{M_8, M_9\}, \{M_9, M_{10}\}$

LCOM1 = 16

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

M₁: Point() M₂: Show() M₃: Hide() M₄: Drag() M₅: MoveTo()

I₁: {Visible, x, y} I₂: {Visible, x, y} I₃: {Visible} I₄: {x, y} I₅: {x, y}

Ortak üye alanı bulunmayan metot çiftleri: {M₃, M₄}, {M₃, M₅}

LCOM1 = 2

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

M₂: Show() M₃: Hide() M₄: Drag() M₅: MoveTo() M₆: IsVisible()

I₂: {Visible, x, y} I₃: {Visible} I₄: {x, y} I₅: {x, y} I₆: {Visible}

M₇: GetX() M₈: SetX() M₉: GetY() M₁₀: SetY()

I₇: {x} I₈: {x} I₉: {y} I₁₀: {y}

Ortak üye alanı bulunmayan metot çiftleri: {M₃, M₄}, {M₃, M₅}, {M₃, M₇}, {M₃, M₈}, {M₃, M₉}, {M₃, M₁₀}, {M₄, M₆}, {M₅, M₆}, {M₆, M₇}, {M₆, M₈}, {M₆, M₉}, {M₆, M₁₀}, {M₇, M₉}, {M₇, M₁₀}, {M₈, M₉}, {M₉, M₁₀}

LCOM1 = 16

Örnek 2:

Sınıf A için; LCOM1 = 0

Sınıf B için; LCOM1 = 0

Değerlendirme:

- LCOM1 ölçütünün minimum değeri "0" 'dır ve maximum değeri metot çifti sayısı ile değişmektedir. LCOM1 normalize edilmiş bir ölçüt değildir.
- Metot çağrılarını göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Sadece 1 metoda sahip sınıf için LCOM1 uygulanamamaktadır.
- Teorik olarak geçerlenmiştir.
- LCOM1 ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan LCOM1 ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak düşürdüğü gözlemlenirken yapılandırıcı metotların göz önüne alınmasının ölçüm sonuçlarında herhangi bir değişikliğe yol açmadığı görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü Sınıf B'de metotlar daha fazla ortak üye alan kullanılmaktadırlar. Ancak LCOM1 hesaplamasına bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize LCOM1 ölçütünün sezgilerle paralellik göstermediğini, iki tane farklı uyum özelliğine sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da LCOM1 ölçütünün *uyum* özelliğini çok da iyi ayırt edemediğini göstermektedir.

4.1.2 LCOM2

LCOM2, Chidamber ve Kemerer tarafından 1994 yılında tanımlanan bir ölçüttür (Chidamber ve Kemerer, 1994). LCOM1 ölçütünden yola çıkılarak önerilmiştir.

Tanım 4.2:

$M_1, M_2, \dots, M_n$ metoda sahip bir C sınıfı olsun. $\{I_i\} = M_i$ metodu tarafından kullanılan üye alanların kümesi olsun. Bu durumda n tane $\{I_1\}, \{I_2\} \dots, \{I_n\}$ küme olur. $P = \{(I_i, I_j) | I_i \cap I_j = \emptyset\}$ ve $Q = \{(I_i, I_j) | I_i \cap I_j \neq \emptyset\}$ olsun. Eğer bütün $\{I_1\}, \{I_2\} \dots, \{I_n\}$ kümeleri $= \emptyset$ ise $P = \emptyset$ olur. LCOM2 şu şekilde tanımlanmaktadır:

$$LCOM2 = \begin{cases} |P| - |Q|, & \text{if } |P| > |Q| \\ 0, & \text{if } |P| \leq |Q| \end{cases} \quad (4.1)$$

Diğer bir deyişle, LCOM2 bir sınıftaki ortak üye alanı bulunmayan metot çifti sayısından ortak üye alanı bulunan metot çifti sayısının çıkarılmasıyla ölçülmektedir.

LCOM2 ölçütü ters uyum ölçümüdür. Yüksek LCOM2 değeri düşük uyumu gösterirken düşük LCOM2 değeri yüksek uyumu gösterir.

LCOM2 Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

M_1 : Point()	M_2 : Show()	M_3 : Hide()	M_4 : Drag()	M_5 : MoveTo()
I_1 : {Visible, x, y}	I_2 : {Visible, x, y}	I_3 : {Visible}	I_4 : {x, y}	I_5 : {x, y}
M_6 : IsVisible()	M_7 : GetX()	M_8 : SetX()	M_9 : GetY()	M_{10} : SetY()
I_6 : {Visible}	I_7 : {x}	I_8 : {x}	I_9 : {y}	I_{10} : {y}

Ortak üye alanı bulunmayan metot çiftleri: $\{M_3, M_4\}, \{M_3, M_5\}, \{M_3, M_7\}, \{M_3, M_8\}, \{M_3, M_9\}, \{M_3, M_{10}\}, \{M_4, M_6\}, \{M_5, M_6\}, \{M_6, M_7\}, \{M_6, M_8\}, \{M_6, M_9\}, \{M_6, M_{10}\}, \{M_7, M_9\}, \{M_7, M_{10}\}, \{M_8, M_9\}, \{M_9, M_{10}\}$

$P = 16$

Ortak üye alanı bulunan metot çiftleri: $\{M_1, M_2\}, \{M_1, M_3\}, \{M_1, M_4\}, \{M_1, M_5\}, \{M_1, M_6\}, \{M_1, M_7\}, \{M_1, M_8\}, \{M_1, M_9\}, \{M_1, M_{10}\}, \{M_2, M_3\}, \{M_2, M_4\}, \{M_2, M_5\}, \{M_2, M_6\}, \{M_2, M_7\}, \{M_2, M_8\}, \{M_2, M_9\}, \{M_2, M_{10}\}, \{M_3, M_6\}, \{M_4, M_5\}, \{M_4, M_7\}, \{M_4, M_8\}, \{M_4, M_9\},$

$\{M_4, M_{10}\}, \{M_5, M_7\}, \{M_5, M_8\}, \{M_5, M_9\}, \{M_5, M_{10}\}, \{M_7, M_8\}, \{M_9, M_{10}\}$

$Q = 29$

$P < Q, LCOM2 = 0$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

$M_1: \text{Point}()$ $M_2: \text{Show}()$ $M_3: \text{Hide}()$ $M_4: \text{Drag}()$ $M_5: \text{MoveTo}()$

$I_1: \{\text{Visible}, x, y\}$ $I_2: \{\text{Visible}, x, y\}$ $I_3: \{\text{Visible}\}$ $I_4: \{x, y\}$ $I_5: \{x, y\}$

Ortak üye alanı bulunmayan metot çiftleri: $\{M_3, M_4\}, \{M_3, M_5\}$

$P = 2$

Ortak üye alanı bulunan metot çiftleri: $\{M_1, M_2\}, \{M_1, M_3\}, \{M_1, M_4\}, \{M_1, M_5\}, \{M_2, M_3\}, \{M_2, M_4\}, \{M_2, M_5\}, \{M_4, M_5\}$

$Q = 8$

$P < Q, LCOM2 = 0$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

$M_2: \text{Show}()$ $M_3: \text{Hide}()$ $M_4: \text{Drag}()$ $M_5: \text{MoveTo}()$ $M_6: \text{IsVisible}()$

$I_2: \{\text{Visible}, x, y\}$ $I_3: \{\text{Visible}\}$ $I_4: \{x, y\}$ $I_5: \{x, y\}$ $I_6: \{\text{Visible}\}$

$M_7: \text{GetX}()$ $M_8: \text{SetX}()$ $M_9: \text{GetY}()$ $M_{10}: \text{SetY}()$

$I_7: \{x\}$ $I_8: \{x\}$ $I_9: \{y\}$ $I_{10}: \{y\}$

Ortak üye alanı bulunmayan metot çiftleri: $\{M_3, M_4\}, \{M_3, M_5\}, \{M_3, M_7\}, \{M_3, M_8\}, \{M_3, M_9\}, \{M_3, M_{10}\}, \{M_4, M_6\}, \{M_5, M_6\}, \{M_6, M_7\}, \{M_6, M_8\}, \{M_6, M_9\}, \{M_6, M_{10}\}, \{M_7, M_9\}, \{M_7, M_{10}\}, \{M_8, M_9\}, \{M_9, M_{10}\}$

$P = 16$

Ortak üye alanı bulunan metot çiftleri: $\{M_2, M_3\}, \{M_2, M_4\}, \{M_2, M_5\}, \{M_2, M_6\}, \{M_2, M_7\}, \{M_2, M_8\}, \{M_2, M_9\}, \{M_2, M_{10}\}, \{M_3, M_6\}, \{M_4, M_5\}, \{M_4, M_7\}, \{M_4, M_8\}, \{M_4, M_9\}, \{M_4, M_{10}\}, \{M_5, M_7\}, \{M_5, M_8\}, \{M_5, M_9\}, \{M_5, M_{10}\}, \{M_7, M_8\}, \{M_9, M_{10}\}$

$Q = 20$

$P < Q, LCOM2 = 0$

Örnek 2:

Sınıf A için;

Ortak üye alanı bulunmayan metot çiftleri: $\{-\}$ ve $P = 0$

Ortak üye alanı bulunan metot çiftleri: $\{M_1, M_2\}$, $\{M_1, M_3\}$, $\{M_2, M_3\}$ ve $Q = 3$

$P < Q$, $LCOM2 = 0$

Sınıf B için;

Ortak üye alanı bulunmayan metot çiftleri: $\{-\}$ ve $P = 0$

Ortak üye alanı bulunan metot çiftleri: $\{M_1, M_2\}$, $\{M_1, M_3\}$, $\{M_2, M_3\}$ ve $Q = 3$

$P < Q$, $LCOM2 = 0$

Değerlendirme:

- LCOM2 ölçütünün minimum değeri "0" 'dır ve maximum değeri metot çifti sayısı ile değişmektedir. LCOM2 normalize edilmiş bir ölçüt değildir.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Sadece 1 metoda sahip sınıf için LCOM2 uygulanamamaktadır.
- Teorik olarak ve deneysel olarak (Briand vd., 1998) geçerlenmiştir.
- LCOM2 ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan LCOM2 ölçümü sonucunda, erişim metotlarının ve yapılandırıcı metotların göz önüne alınmasının *uyumu* etkilemediği görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü Sınıf B'de metotlar daha fazla ortak üye alan kullanılmaktadırlar. Ancak LCOM2 hesaplamasına bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize LCOM2 ölçütünün sezgilerle paralellik göstermediğini, iki tane farklı uyum özelliğine sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da LCOM2 ölçütünün *uyum* özelliğini çok da iyi ayırt edemediğini göstermektedir.

4.2 Hitz ve Montazeri Tarafından Önerilen Ölçütler**4.2.1 LCOM3**

Hitz ve Montazeri tarafından 1995 yılında tanımlanan ölçüttür (Hitz ve Montazeri, 1995). LCOM1 ölçütünün değişik bir varyasyonudur.

Tanım 4.3:

X bir sınıf olsun. I_x , X sınıfının üye alanlarının kümesi olsun. M_x , X sınıfının metotlarının

kümesi olsun. Yönsüz bir grafik $G_x(V, E)$ göz önüne alalım, uçları sınıfın metotları olsun.

$$V = M_x \quad (4.2)$$

$$E = \{ \langle m, n \rangle \in V \times V \mid \exists i \in I_x : (m \text{ accesses } i) \wedge (n \text{ accesses } i) \} \quad (4.3)$$

LCOM3, G_x grafiğinin bağlı bileşenlerinin sayısı olarak tanımlanmaktadır;

$$G_x(1 \leq \text{LCOM3} \leq |M_x|). \quad (4.4)$$

Diğer bir ifadeyle, Yönsüz bir grafik G göz önüne alalım, uçları sınıfın metotları olsun. Eğer metotlar en az bir ortak üye alan kullanıyorlarsa aralarında bir kenar olsun. Bu durumda LCOM3, G grafiğinin bağlı bileşenlerinin sayısı olarak tanımlanmaktadır.

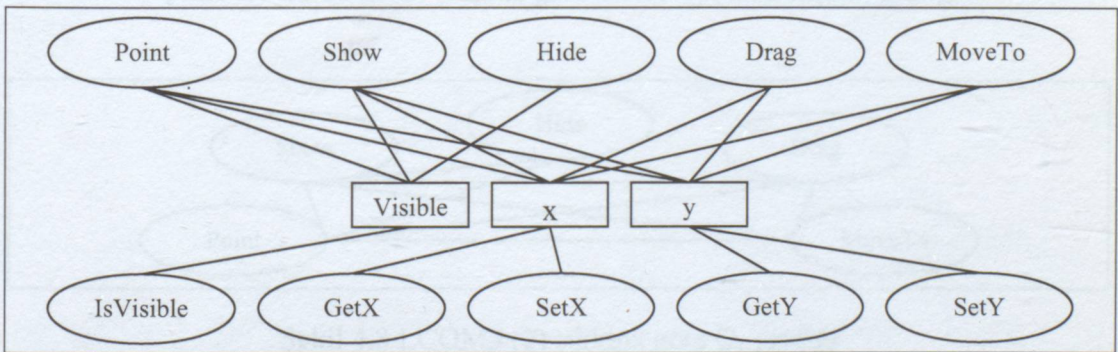
LCOM3 ölçütü ters uyum ölçümüdür. Yüksek LCOM3 değeri düşük uyumu gösterirken düşük LCOM3 değeri yüksek uyumu gösterir.

LCOM3 Ölçütü Hesaplaması:

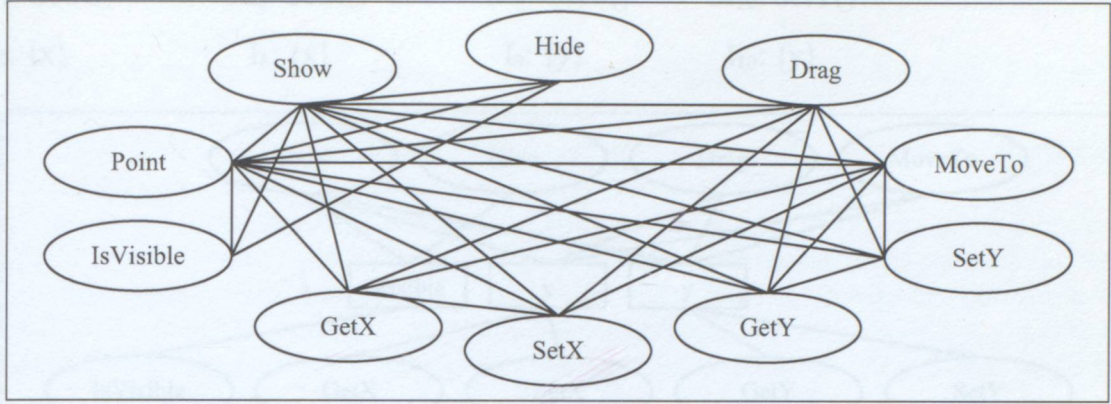
Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

M_1 : Point()	M_2 : Show()	M_3 : Hide()	M_4 : Drag()	M_5 : MoveTo()
I_1 : {Visible, x, y}	I_2 : {Visible, x, y}	I_3 : {Visible}	I_4 : {x, y}	I_5 : {x, y}
M_6 : IsVisible()	M_7 : GetX()	M_8 : SetX()	M_9 : GetY()	M_{10} : SetY()
I_6 : {Visible}	I_7 : {x}	I_8 : {x}	I_9 : {y}	I_{10} : {y}



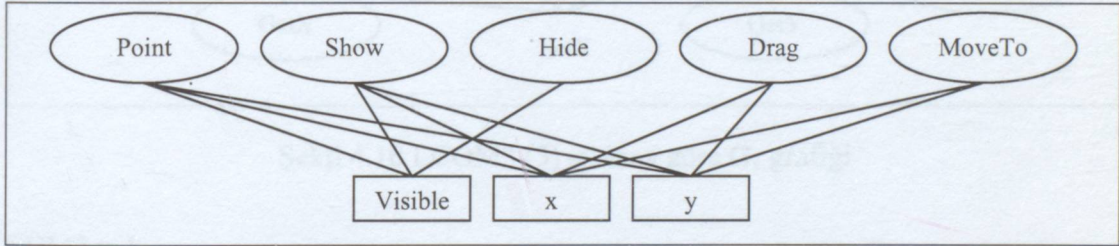
Şekil 4.5 LCOM3 (1) şikkına göre metot-üye alan ilişkisi grafiği

Şekil 4.6 LCOM3 (1) şikkına göre G_x grafiği

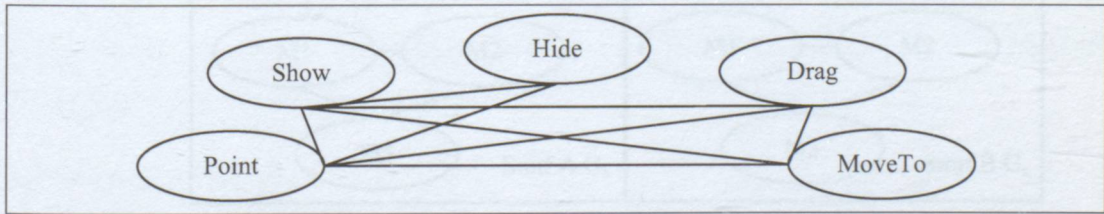
LCOM3 = 1

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

M_1 : Point() M_2 : Show() M_3 : Hide() M_4 : Drag() M_5 : MoveTo()
 I_1 : {Visible, x, y} I_2 : {Visible, x, y} I_3 : {Visible} I_4 : {x, y} I_5 : {x, y}



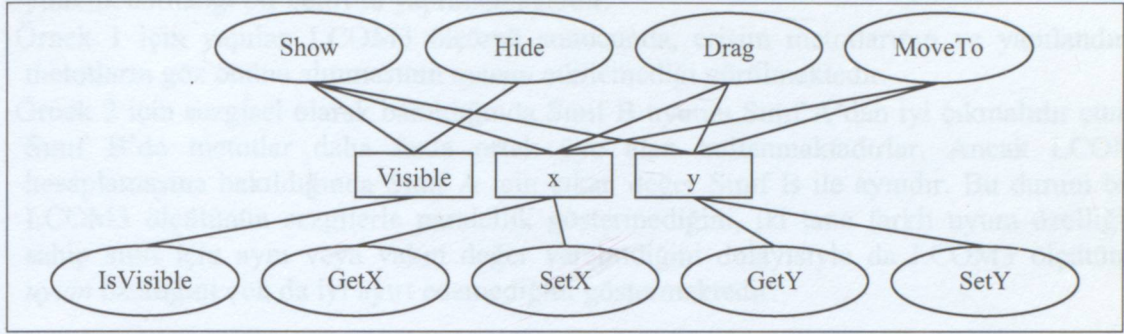
Şekil 4.7 LCOM3 (2) şikkına göre metot-üye alan ilişkisi grafiği

Şekil 4.8 LCOM3 (2) şikkına göre G_x grafiği

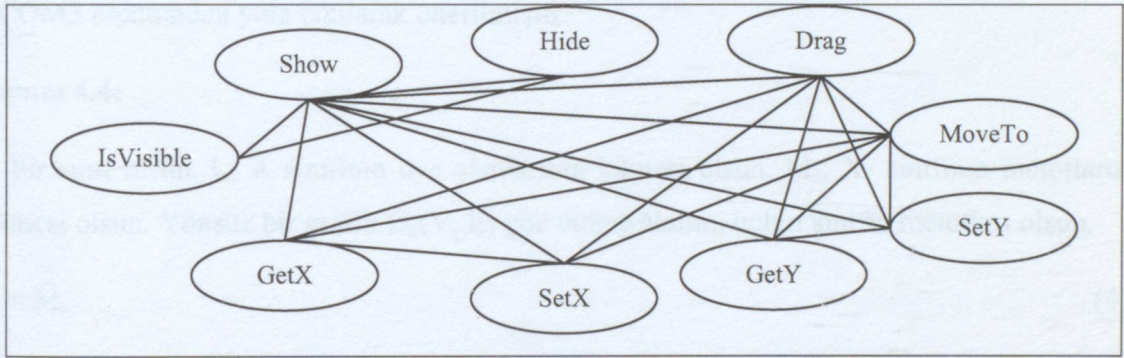
LCOM3 = 1

3) Yapılandırıcı metotları hesaplamaya alınmayacak, erişim metotlar hesaplamaya alınacak

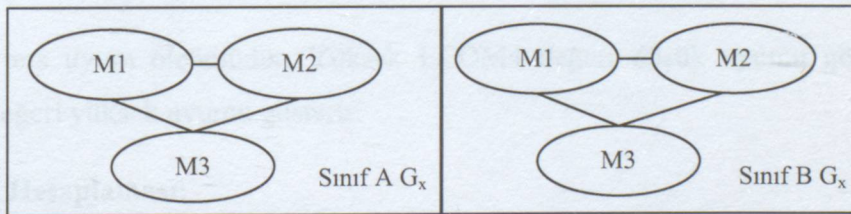
M_2 : Show() M_3 : Hide() M_4 : Drag() M_5 : MoveTo() M_6 : IsVisible()
 I_2 : {Visible, x, y} I_3 : {Visible} I_4 : {x, y} I_5 : {x, y} I_6 : {Visible}

M₇: GetX()M₈: SetX()M₉: GetY()M₁₀: SetY()I₇: {x}I₈: {x}I₉: {y}I₁₀: {y}

Şekil 4.9 LCOM3 (3) şikkına göre metot-üye alan ilişkisi grafiği

Şekil 4.10 LCOM3 (3) şikkına göre G_x grafiği

LCOM3 = 1

Örnek 2:Şekil 4.11 LCOM3 Sınıf A ve Sınıf B'ye ait G_x grafiği

Sınıf A için; LCOM3 = 1

Sınıf B için; LCOM3 = 1

Değerlendirme:

- LCOM3 ölçütünün minimum değeri "1" 'dir ve maximum değeri metot sayısı ile değişmektedir. LCOM3 normalize edilmiş bir ölçüt değildir.
- Metot çağrımları göz önüne alınmamaktadır.

- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Teorik ve deneysel olarak geçerlenmemiştir.
- LCOM3 ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan LCOM3 ölçümü sonucunda, erişim metotlarının ve yapılandırıcı metotların göz önüne alınmasının *uyumu* etkilemediği görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü Sınıf B'de metotlar daha fazla ortak üye alan kullanmaktadırlar. Ancak LCOM3 hesaplamasına bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize LCOM3 ölçütünün sezgilerle paralellik göstermediğini, iki tane farklı uyum özelliğine sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da LCOM3 ölçütünün *uyum* özelliğini çok da iyi ayırt edemediğini göstermektedir.

4.2.2 LCOM4

Hitz ve Montazeri tarafından 1995 yılında tanımlanan ölçüttür (Hitz ve Montazeri, 1995). LCOM3 ölçütünden yola çıkılarak önerilmiştir.

Tanım 4.4:

X bir sınıf olsun. I_x , X sınıfının üye alanlarının kümesi olsun. M_x , X sınıfının metotlarının kümesi olsun. Yönsüz bir grafik $G_x(V, E)$ göz önüne alalım, uçları sınıfın metotları olsun.

$$V = M_x \quad (4.5)$$

$$E = \{ \langle m, n \rangle \in V \times V \mid \exists i \in I_x : (m \text{ accesses } i) \wedge (n \text{ accesses } i) \vee (m \text{ calls } n) \vee (n \text{ calls } m) \} \quad (4.6)$$

LCOM4, G_x in bağlı bileşenlerinin sayısı olarak tanımlanmaktadır.

Diğer bir ifadeyle, LCOM3'e benzer. Yalnız burada G grafiğinde eğer metotlardan biri diğerini çağırıyorsa o zaman da aralarında bir kenar oluşturulmaktadır.

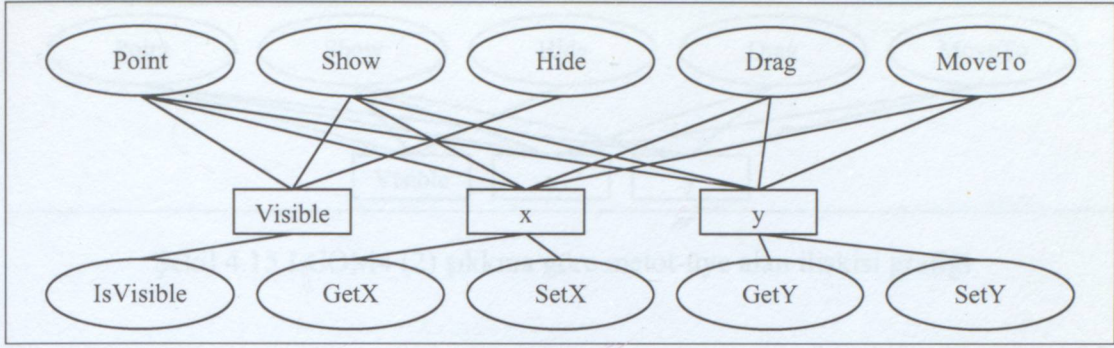
LCOM4 ölçütü ters uyum ölçümüdür. Yüksek LCOM4 değeri düşük uyumu gösterirken düşük LCOM4 değeri yüksek uyumu gösterir.

LCOM4 Ölçütü Hesaplaması:

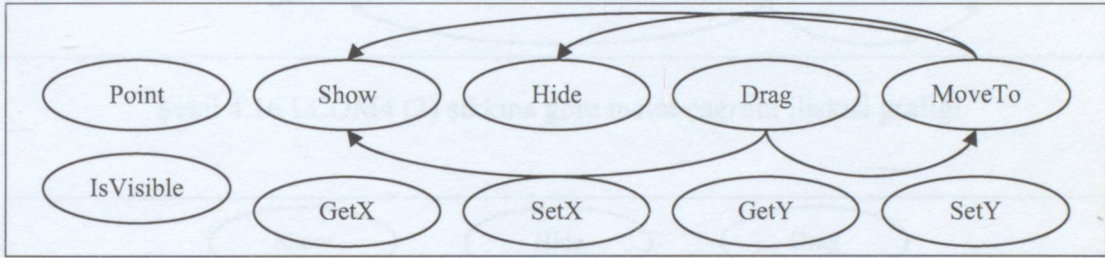
Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

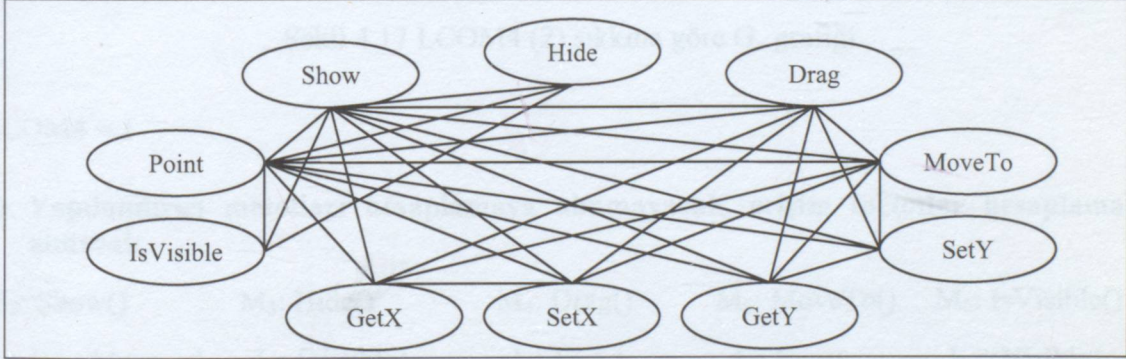
M_1 : Point()	M_2 : Show()	M_3 : Hide()	M_4 : Drag()	M_5 : MoveTo()
I_1 : {Visible, x, y}	I_2 : {Visible, x, y}	I_3 : {Visible}	I_4 : {x, y}	I_5 : {x, y}
M_6 : IsVisible()	M_7 : GetX()	M_8 : SetX()	M_9 : GetY()	M_{10} : SetY()
I_6 : {Visible}	I_7 : {x}	I_8 : {x}	I_9 : {y}	I_{10} : {y}



Şekil 4.12 LCOM4 (1) şikkına göre metot-üye alan ilişkisi grafiği



Şekil 4.13 LCOM4 (1) şikkına göre metot çağırımı ilişkisi grafiği



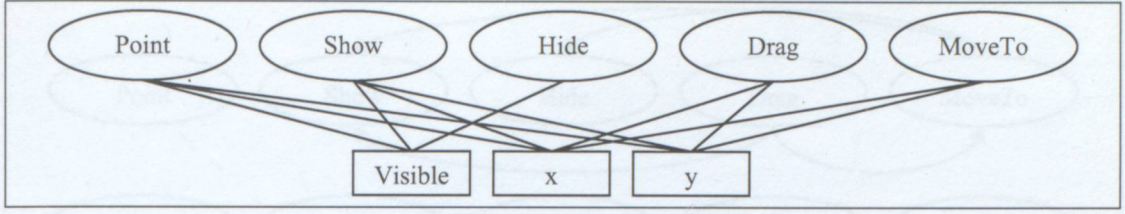
Şekil 4.14 LCOM4 (1) şikkına göre G_x grafiği

LCOM4 = 1

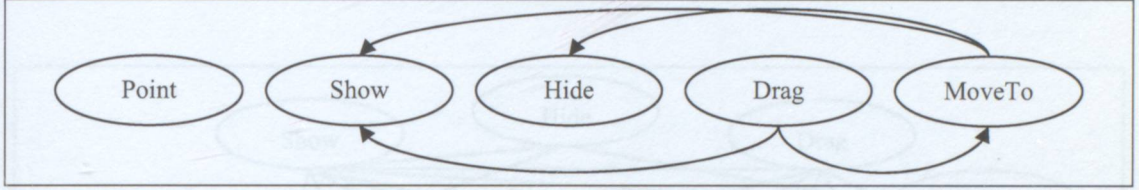
2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

M_1 : Point() M_2 : Show() M_3 : Hide() M_4 : Drag() M_5 : MoveTo()

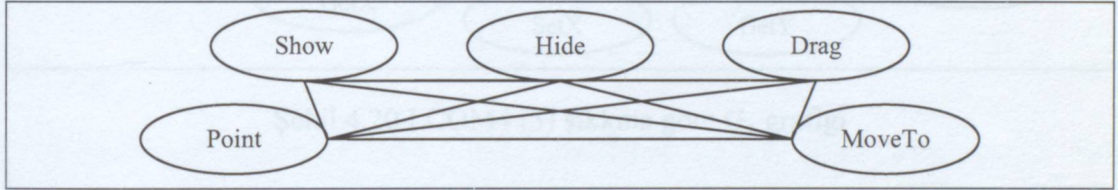
I_1 : {Visible, x, y} I_2 : {Visible, x, y} I_3 : {Visible} I_4 : {x, y} I_5 : {x, y}



Şekil 4.15 LCOM4 (2) şikkına göre metot-üye alan ilişkisi grafiği



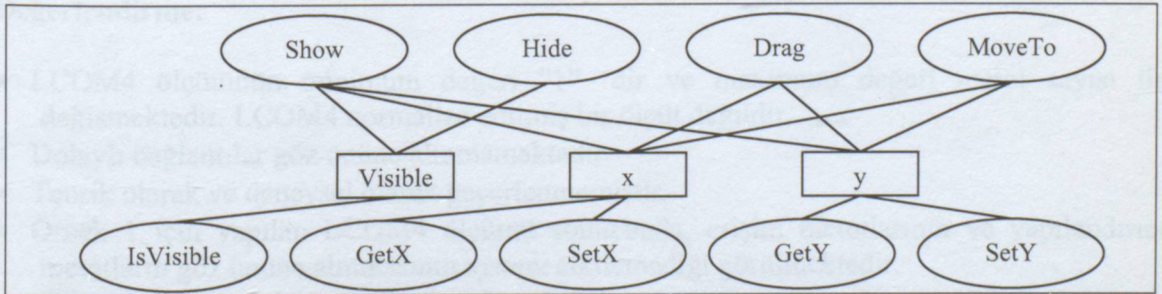
Şekil 4.16 LCOM4 (2) şikkına göre metot çağırımı ilişkisi grafiği

Şekil 4.17 LCOM4 (2) şikkına göre G_x grafiği

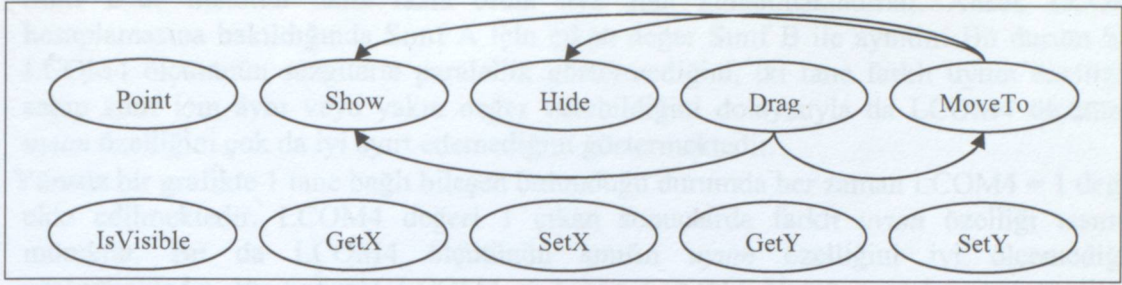
LCOM4 = 1

3) Yapılandırıcı metotları hesaplamaya alınmayacak, erişim metotlar hesaplamaya alınacak

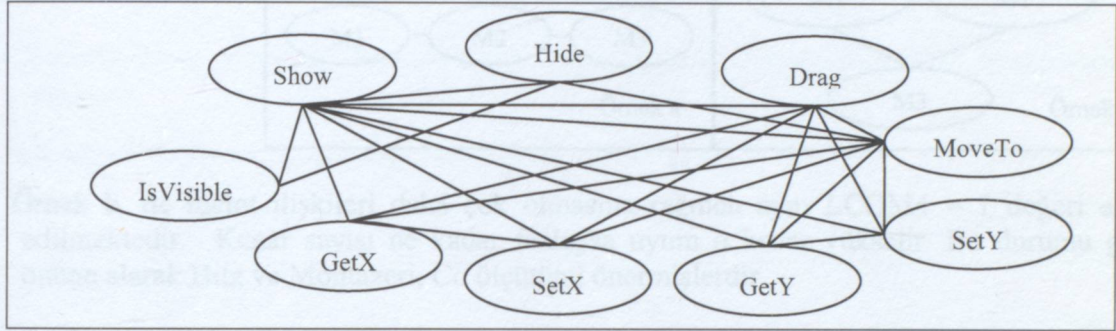
M ₂ : Show()	M ₃ : Hide()	M ₄ : Drag()	M ₅ : MoveTo()	M ₆ : IsVisible()
I ₂ : {Visible, x, y}	I ₃ : {Visible}	I ₄ : {x, y}	I ₅ : {x, y}	I ₆ : {Visible}
M ₇ : GetX()	M ₈ : SetX()	M ₉ : GetY()	M ₁₀ : SetY()	
I ₇ : {x}	I ₈ : {x}	I ₉ : {y}	I ₁₀ : {y}	



Şekil 4.18 LCOM4 (3) şikkına göre metot - üye alan ilişkisi grafiği



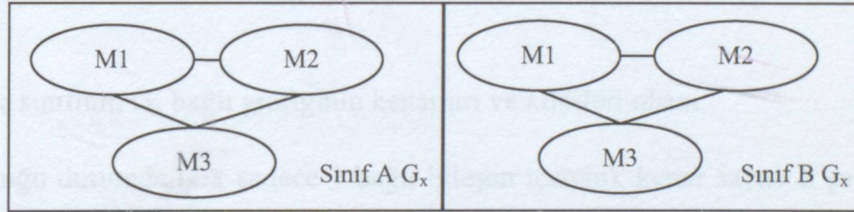
Şekil 4.19 LCOM4 (3) şikkına göre metot çağırımı ilişkisi grafiği



Şekil 4.20 LCOM4 (3) şikkına göre G_x grafiği

LCOM4 = 1

Örnek 2:



Şekil 4.21 LCOM4 Sınıf A ve Sınıf B'ye ait G_x grafiği

Sınıf A için; LCOM4 = 1,

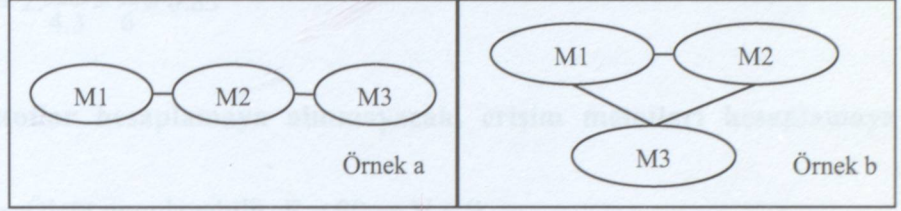
Sınıf B için; LCOM4 = 1

Değerlendirme:

- LCOM4 ölçütünün minimum değeri "1" 'dir ve maximum değeri metot sayısı ile değişmektedir. LCOM4 normalize edilmiş bir ölçüt değildir.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Teorik olarak ve deneysel olarak geçerlenmemiştir.
- Örnek 1 için yapılan LCOM4 ölçümü sonucunda, erişim metotlarının ve yapılandırıcı metotların göz önüne alınmasının *uyumu* etkilemediği görülmektedir.
- LCOM4 ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü

Sınıf B'de metotlar daha fazla ortak üye alan kullanmaktadırlar. Ancak LCOM4 hesaplamasına bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize LCOM4 ölçütünün sezgilerle paralellik göstermediğini, iki tane farklı uyum özelliğine sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da LCOM4 ölçütünün *uyum* özelliğini çok da iyi ayırt edemediğini göstermektedir.

- Yönsüz bir grafikte 1 tane bağlı bileşen bulunduğu durumda her zaman $LCOM4 = 1$ değeri elde edilmektedir. LCOM4 değeri 1 çıkan sonuçlarda farklı *uyum* özelliği taşımak mümkün. Bu da LCOM4 ölçütünün sınıfın *uyum* özelliğini iyi ölçemediğini göstermektedir. Bu nedenle $LCOM4 = 1$ için LCOM4 ölçütü gerçek *uyum* özelliğini yansıtmamaktadır. Şöyle ki: İki örneği ele alırsak;



- Örnek b. de metot ilişkileri daha çok olmasına rağmen aynı $LCOM4 = 1$ değeri elde edilmektedir. Kenar sayısı ne kadar fazlaysa uyum o kadar yükselir. Bu durumu göz önüne alarak Hitz ve Montazeri, Co ölçütünü önermişlerdir.

4.2.3 Co

Hitz ve Montazeri tarafından 1995 yılında tanımlanan ölçüttür (Hitz ve Montazeri, 1995). LCOM4 ölçütünün bir varyasyonudur. $LCOM4 = 1$ değerine sahip sınıfların uyum özelliğini ayırt etmek üzere bağlı bileşenin *kenar sayısı* hesaba katılarak Co ölçütü önerilmiştir.

Tanım 4.5:

E ve V sırasıyla c sınıfının G_x bağlı grafiğinin kenarları ve köşeleri olsun.

$LCOM4 = 1$ olduğu durumda (G_x sadece 1 bağlı bileşen içersin), kenar sayısı E şu değerler arasında değişir: $|V| - 1$ (minimum uyum) ve $|V| \cdot (|V| - 1)/2$ (maximum uyum). Co ölçütü bağlı bileşen 1 tane olduğu durumlarda uygulanabilen bir ölçüttür. Co ölçütü şu şekilde tanımlanmaktadır.

$$Co = 2 \cdot \frac{|E| - (|V| - 1)}{(|V| - 1)(|V| - 2)} \quad (4.7)$$

Co Ölçütü Hesaplaması:

Örnek 1:

- 1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak 1 adet bağlı bileşen var. Ölçüt uygulanabilir. $E = 31$ ve $V = 10$.

$$Co = 2 \cdot \frac{|31| - (|10| - 1)}{(|10| - 1) \cdot (|10| - 2)} = 2 \cdot \frac{22}{9 \cdot 8} = \frac{11}{18} = 0.61$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

1 adet bağlı bileşen var. Ölçüt uygulanabilir. E = 9 ve V = 5.

$$Co = 2 \cdot \frac{|9| - (|5| - 1)}{(|5| - 1) \cdot (|5| - 2)} = 2 \cdot \frac{5}{4 \cdot 3} = \frac{5}{6} = 0.83$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

1 adet bağlı bileşen var. Ölçüt uygulanabilir. E = 20 ve V = 9.

$$Co = 2 \cdot \frac{|20| - (|9| - 1)}{(|9| - 1) \cdot (|9| - 2)} = 2 \cdot \frac{12}{8 \cdot 7} = 0.42$$

Örnek 2:

Sınıf A için; 1 adet bağlı bileşen var. Ölçüt uygulanabilir. E = 3 ve V = 3.

$$Co = 2 \cdot \frac{|3| - (|3| - 1)}{(|3| - 1) \cdot (|3| - 2)} = 2 \cdot \frac{1}{2 \cdot 1} = 1$$

Sınıf B için; 1 adet bağlı bileşen var. Ölçüt uygulanabilir. E = 3 ve V = 3.

$$Co = 2 \cdot \frac{|3| - (|3| - 1)}{(|3| - 1) \cdot (|3| - 2)} = 2 \cdot \frac{1}{2 \cdot 1} = 1$$

Değerlendirme:

- Co ölçütü [0, 1] arasında değerler almakta, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Grafikte sadece 1 tane bağlı bileşen olduğu durumlarda uygulanabilmektedir.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Teorik olarak ve deneysel olarak geçirilmemiştir.
- Co ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan Co ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak düşürdüğü, yapılandırıcı metotların *uyumu* yapay olarak arttırdığı görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü Sınıf B'de metotlar daha fazla ortak üye alan kullanılmaktadırlar. Ancak, Co hesaplamasına

bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize Co ölçütünün sezgilerle paralellik göstermediğini, iki tane farklı uyum özelliğine sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da Co ölçütünün *uyum* özelliğini çok da iyi ayırt edemediğini göstermektedir.

4.3 Bieman ve Kang Tarafından Önerilen Ölçütler

Bieman ve Kang 1995 yılında Geniş C++ Sistemleri için mantıklı sayılabilecek ve Chidamber ve Kemerer ölçütleri temeline dayanan (Briand vd., 1998) iki yeni ölçüt önermiş ve bu ölçütleri sınıf uyumu ve tekrar kullanım için değerlendirmişlerdir (Bieman ve Kang, 1995). Çalışmaları sonucu görülmüştür ki; bir sınıf ne kadar sık kalıtım yoluyla tekrar kullanılıyorsa o kadar az uyuma sahiptir.

Bieman ve Kang'e göre;

Bir sınıfın bileşenleri sınıfta tanımlanan üye alanlar ve metotlar ve buna ilave kalıtım alınanlardır. Bir metot ve üye alanın ilişkili olması, üye alanın metot tarafından kullanılması anlamındadır. İki metodun ilişkili olması, her iki metodun da üye alanı kullanması anlamındadır. Sınıf uyumu, sınıftaki bağlı metotların sayısı olarak tanımlanabilir.

Burada metot çiftlerinin doğrudan veya dolaylı yoldan bağlanmalarına dayalı iki sınıf uyum ölçütü tanımlanmaktadır;

- TCC (Tight Class Cohesion) (doğrudan)
- LCC (Loose Class Cohesion) (doğrudan + dolaylı)

Bu iki ölçüt de "duyarlı" (Bir sınıftaki uyum değişikliğinin ölçüm sonucunu etkilemesi) olarak belirtilmişlerdir.

Bieman ve Kang'e göre metotlar 2 yolla bağlıdır (Bieman ve Kang, 1995);

- **MIV ilişkileri:** paylaşılan üye alanlar yoluyla olan metotlar arasındaki haberleşmeyi içerir.
- **Call ilişkileri:** bir metottan diğerine doğrudan veya dolaylı yoldan mesajların gönderilmesini içerir.

Yapılandırıcı'lar genellikle bir sınıftaki bütün üye alanlara erişir ve böylece üye alanlar bütün diğer metotlarla sanal (virtual) olarak paylaşır. Bu nedenle çalışmalarında yapılandırıcı metotları çıkarmışlardır. Sonlandırıcı metotları da kullanıma almamışlardır.

Bieman ve Kang ölçüt tanımlamalarından önce şu belirtileri yapmışlardır:

Eğer, *üye alan*, *M* metodunda veri belirtimi olarak görünüyorsa o zaman bu üye alan metot *M* tarafından *doğrudan kullanılıyor* demektir. *Üye alan M* metodu ile aynı sınıfta veya sınıfın ata

sınıfında tanımlanabilir.

Eğer *call graph*'ta M metodu M' metodunun öncesinde geliyorsa M' metodu M metodu tarafından *doğrudan çağrılıyor* demektir. Eğer *call graph*'ta M metodundan M' metoduna bir yol varsa M' metodu M metodu tarafından *dolaylı yoldan çağrılıyor* demektir.

Eğer, *üye alan*, M metodu tarafından doğrudan veya dolaylı yoldan çağrılan M' metodu tarafından doğrudan kullanılıyorsa o zaman bu *üye alan* metot M tarafından *dolaylı yoldan kullanılıyor* demektir.

Eğer iki metot arasında bir veya daha fazla ortak *üye alan* varsa bu iki metot *doğrudan ilişkilidir*. İki metot diğer doğrudan ilişkili metotlar üzerinden bağlıysa o zaman bu iki metot *dolaylı ilişkilidir*.

NP(C): bir sınıftaki mümkün olan maximum doğrudan veya dolaylı bağlantı sayısı. Eğer bir sınıfta N metot varsa o zaman $NP(C)$;

$$NP(C) = \frac{N * (N - 1)}{2} \quad (4.8)$$

NDC(C): doğrudan bağlantı sayısı, **NIC(C):** dolaylı bağlantı sayısı

4.3.1 TCC

Tanım 4.6:

Bir sınıftaki doğrudan bağlı metot çifti sayısının, $NDC(C)$, bir sınıftaki mümkün olan maximum doğrudan veya dolaylı bağlantı sayısına, $NP(C)$, oranı olarak bulunur..

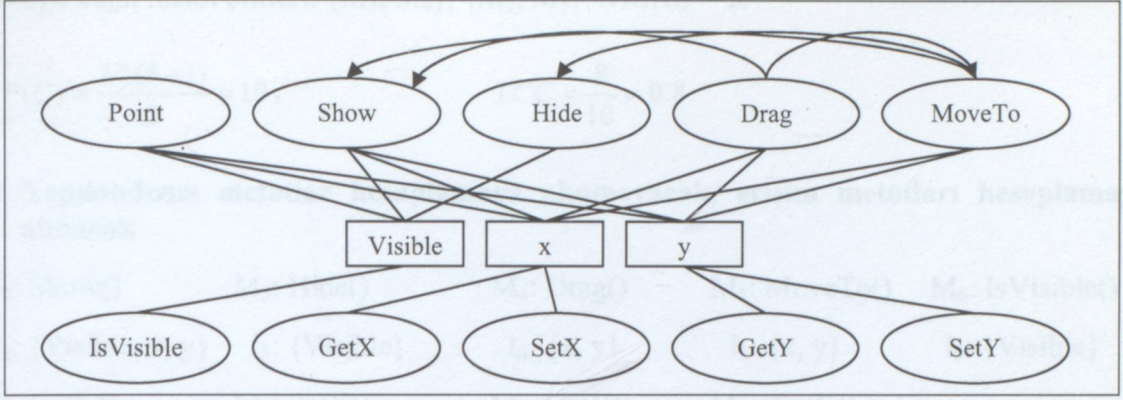
$$TCC = \frac{NDC(C)}{NP(C)} \quad (4.9)$$

TCC Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

M_1 : Point()	M_2 : Show()	M_3 : Hide()	M_4 : Drag()	M_5 : MoveTo()
I_1 : {Visible, x, y}	I_2 : {Visible, x, y}	I_3 : {Visible}	I_4 : {x, y}	I_5 : {x, y}
M_6 : IsVisible()	M_7 : GetX()	M_8 : SetX()	M_9 : GetY()	M_{10} : SetY()
I_6 : {Visible}	I_7 : {x}	I_8 : {x}	I_9 : {y}	I_{10} : {y}



Şekil 4.22 TCC (1) şikkına göre metot-üye alan ve metot çağırımı ilişkisi grafiği

Doğrudan bağlı metot çiftleri: $\{M_1, M_2\}$, $\{M_1, M_3\}$, $\{M_1, M_4\}$, $\{M_1, M_5\}$, $\{M_1, M_6\}$, $\{M_1, M_7\}$, $\{M_1, M_8\}$, $\{M_1, M_9\}$, $\{M_1, M_{10}\}$, $\{M_2, M_3\}$, $\{M_2, M_4\}$, $\{M_2, M_5\}$, $\{M_2, M_6\}$, $\{M_2, M_7\}$, $\{M_2, M_8\}$, $\{M_2, M_9\}$, $\{M_2, M_{10}\}$, $\{M_3, M_6\}$, $\{M_4, M_5\}$, $\{M_4, M_7\}$, $\{M_4, M_8\}$, $\{M_4, M_9\}$, $\{M_4, M_{10}\}$, $\{M_5, M_7\}$, $\{M_5, M_8\}$, $\{M_5, M_9\}$, $\{M_5, M_{10}\}$, $\{M_7, M_8\}$, $\{M_9, M_{10}\}$ ve $NDC(C) = 29$.

Dolaylı bağlı metot çiftleri: $\{M_3, M_4\}$, $\{M_3, M_5\}$, $\{M_3, M_7\}$, $\{M_3, M_8\}$, $\{M_3, M_9\}$, $\{M_3, M_{10}\}$, $\{M_4, M_6\}$, $\{M_5, M_6\}$, $\{M_6, M_7\}$, $\{M_6, M_8\}$, $\{M_6, M_9\}$, $\{M_6, M_{10}\}$, $\{M_7, M_9\}$, $\{M_7, M_{10}\}$, $\{M_8, M_9\}$, $\{M_8, M_{10}\}$ ve $NIC(C) = 16$.

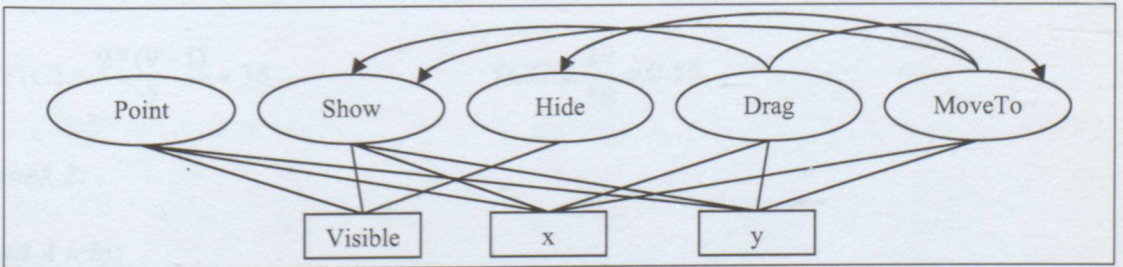
$$NP(C) = \frac{10 * (10 - 1)}{2} = 45,$$

$$TCC = \frac{29}{45} = 0.64$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

M_1 : Point() M_2 : Show() M_3 : Hide() M_4 : Drag() M_5 : MoveTo()

I_1 : {Visible, x, y} I_2 : {Visible, x, y} I_3 : {Visible} I_4 : {x, y} I_5 : {x, y}



Şekil 4.23 TCC (2) şikkına göre metot-üye alan ve metot çağırımı ilişkisi grafiği

Doğrudan bağlı metot çiftleri: $\{M_1, M_2\}$, $\{M_1, M_3\}$, $\{M_1, M_4\}$, $\{M_1, M_5\}$, $\{M_2, M_3\}$, $\{M_2, M_4\}$, $\{M_2, M_5\}$, $\{M_4, M_5\}$ ve $NDC(C) = 8$.

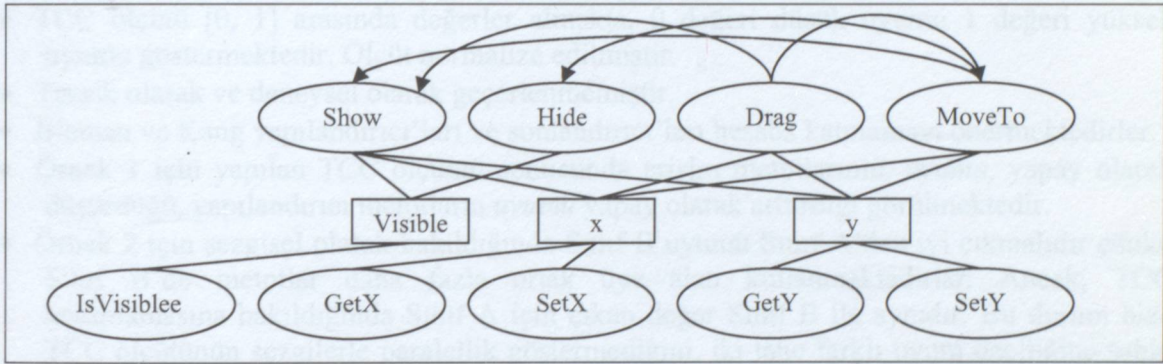
Dolaylı bağlı metot çiftleri: $\{M_3, M_4\}, \{M_3, M_5\}$. $NIC(C) = 2$.

$$NP(C) = \frac{5 * (5 - 1)}{2} = 10,$$

$$TCC = \frac{8}{10} = 0.8$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

M ₂ : Show()	M ₃ : Hide()	M ₄ : Drag()	M ₅ : MoveTo()	M ₆ : IsVisible()
I ₂ : {Visible, x, y}	I ₃ : {Visible}	I ₄ : {x, y}	I ₅ : {x, y}	I ₆ : {Visible}
M ₇ : GetX()	M ₈ : SetX()	M ₉ : GetY()	M ₁₀ : SetY()	
I ₇ : {x}	I ₈ : {x}	I ₉ : {y}	I ₁₀ : {y}	



Şekil 4.24 TCC (3) şikkına göre metot-üye alan ve metot çağırımı ilişkisi grafiği

Doğrudan bağlı metot çiftleri: $\{M_2, M_3\}, \{M_2, M_4\}, \{M_2, M_5\}, \{M_2, M_6\}, \{M_2, M_7\}, \{M_2, M_8\}, \{M_2, M_9\}, \{M_2, M_{10}\}, \{M_3, M_6\}, \{M_4, M_5\}, \{M_4, M_7\}, \{M_4, M_8\}, \{M_4, M_9\}, \{M_4, M_{10}\}, \{M_5, M_7\}, \{M_5, M_8\}, \{M_5, M_9\}, \{M_5, M_{10}\}, \{M_7, M_8\}, \{M_9, M_{10}\}$ ve $NDC(C) = 20$.

Dolaylı bağlı metot çiftleri: $\{M_3, M_4\}, \{M_3, M_5\}, \{M_3, M_7\}, \{M_3, M_8\}, \{M_3, M_9\}, \{M_3, M_{10}\}, \{M_4, M_6\}, \{M_5, M_6\}, \{M_6, M_7\}, \{M_6, M_8\}, \{M_6, M_9\}, \{M_6, M_{10}\}, \{M_7, M_9\}, \{M_7, M_{10}\}, \{M_8, M_9\}, \{M_8, M_{10}\}$ ve $NIC(C) = 16$.

$$NP(C) = \frac{9 * (9 - 1)}{2} = 36,$$

$$TCC = \frac{20}{36} = 0.55$$

Örnek 2:

Sınıf A için:

Doğrudan bağlı metot çiftleri: $\{M_1, M_2\}, \{M_1, M_3\}, \{M_2, M_3\}$ ve $NDC(C) = 3$.

Dolaylı bağlı metot çiftleri: - ve $NIC(C) = 0$.

$$NP(C) = \frac{3 * (3 - 1)}{2} = 3,$$

$$TCC = \frac{3}{3} = 1$$

Sınıf B için:

Doğrudan bağlı metot çiftleri: $\{M_1, M_2\}$, $\{M_1, M_3\}$, $\{M_2, M_3\}$ ve $NDC(C) = 3$.

Dolaylı bağlı metot çiftleri: - ve $NIC(C) = 0$.

$$NP(C) = \frac{3 * (3 - 1)}{2} = 3,$$

$$TCC = \frac{3}{3} = 1$$

Değerlendirme:

- TCC ölçütü $[0, 1]$ arasında değerler almakta, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Teorik olarak ve deneysel olarak geçerlenmemiştir.
- Bieman ve Kang yapılandırıcı'ları ve sonlandırıcı'ları hesaba katmamayı önermektedirler.
- Örnek 1 için yapılan TCC ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak düşürdüğü, yapılandırıcı metotların *uyumu* yapay olarak arttırdığı görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü Sınıf B'de metotlar daha fazla ortak üye alan kullanılmaktadırlar. Ancak, TCC hesaplamasına bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize TCC ölçütünün sezgilerle paralellik göstermediğini, iki tane farklı uyum özelliğine sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da TCC ölçütünün *uyum* özelliğini çok da iyi ayırt edemediğini göstermektedir.

4.3.2 LCC

Tanım 4.7:

Bir sınıftaki doğrudan bağlı metot çifti sayısı, $NDC(C)$, ve dolaylı bağlı metot çifti sayısının, $NIC(C)$, toplamının, bir sınıftaki mümkün olan maximum doğrudan veya dolaylı bağlantı sayısına, $NP(C)$, oranı olarak bulunur.

$$LCC = \frac{NDC(C) + NIC(C)}{NP(C)}$$

(4.10)

LCC Ölçütü Hesaplaması:

Örnek 1:

$NDC(C)$, $NIC(C)$, $NP(C)$ değerleri TCC ölçütünde hesaplanmıştır

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

$$NDC(C) = 29, \quad NIC(C) = 16, \quad NP(C) = 45,$$

$$LCC = \frac{29 + 16}{45} = 1$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

$$NDC(C) = 8, \quad NIC(C) = 2, \quad NP(C) = 10, \quad LCC = \frac{8+2}{10} = 1$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

$$NDC(C) = 20, \quad NIC(C) = 16, \quad NP(C) = 36, \quad LCC = \frac{20+16}{36} = 1$$

Örnek 2:

$$\text{Sınıf A için: } LCC = \frac{3+0}{3} = 1$$

$$\text{Sınıf B için: } LCC = \frac{3+0}{3} = 1$$

Değerlendirme:

- LCC ölçütü [0, 1] arasında değerler almakta, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Metot çağrımları, dolaylı yoldan üye alan kullanımına bakmak amacıyla kullanılmış, metotların birbirini çağırma anlamında metot ilişkisi göz önüne alınmamaktadır.
- Teorik olarak ve deneysel olarak geçerlenmemiştir.
- Bieman ve Kang yapılandırıcı'ları ve sonlandırıcı'ları hesaba katmamayı önermektedirler.
- Örnek 1 için yapılan LCC ölçümü sonucunda, erişim metotlarının ve yapılandırıcı metotların göz önüne alınmasının *uyumu* etkilemediği görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü Sınıf B'de metotlar daha fazla ortak üye alan kullanmaktadırlar. Ancak, LCC hesaplamasına bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize LCC ölçütünün sezgilerle paralellik göstermediğini, iki tane farklı uyum özelliğine sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da LCC ölçütünün *uyum* özelliğini çok da iyi ayırt edemediğini göstermektedir.

4.4 Henderson-Sellors Tarafından Önerilen Ölçütler

4.4.1 LCOM5

Henderson ve Sellors tarafından 1996 yılında tanımlanan ölçüttür (Briand vd., 1998b). Chidamber ve Kemerer'in LCOM ölçütünden esinlenerek önerilmiştir.

Tanım 4.8:

M_i metot kümesi ($i=1, \dots, m$) ve bu metotların eriştiği üye alan kümesi A_j ($j=1, \dots, a$) olsun. $\mu(A_j)$, A_j üye alanına referans eden metot sayısı olsun. LCOM5 şu şekilde tanımlanmaktadır:

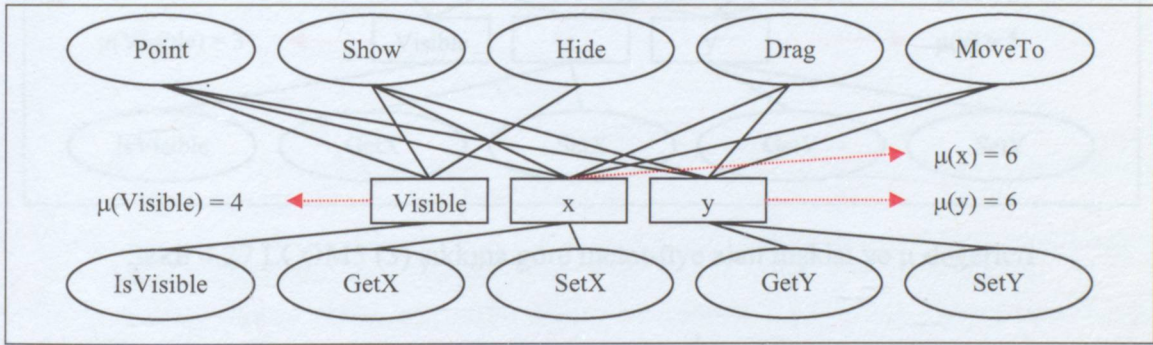
$$LCOM5 = \frac{\frac{1}{a} \sum_{j=1}^a \mu(A_j) - m}{1 - m} \quad (4.11)$$

LCOM5 Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

M ₁ : Point()	M ₂ : Show()	M ₃ : Hide()	M ₄ : Drag()	M ₅ : MoveTo()
I ₁ : {Visible, x, y}	I ₂ : {Visible, x, y}	I ₃ : {Visible}	I ₄ : {x, y}	I ₅ : {x, y}
M ₆ : IsVisible()	M ₇ : GetX()	M ₈ : SetX()	M ₉ : GetY()	M ₁₀ : SetY()
I ₆ : {Visible}	I ₇ : {x}	I ₈ : {x}	I ₉ : {y}	I ₁₀ : {y}

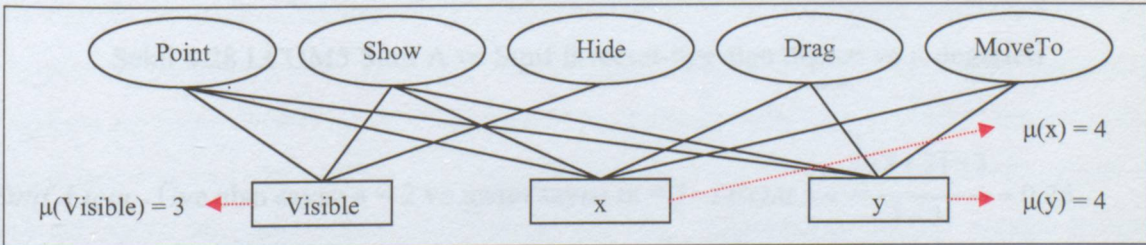


Şekil 4.25 LCOM5 (1) şikkına göre metot-üye alan ilişkisi ve μ değerleri

Üye alan sayısı $a = 3$ ve metot sayısı $m = 10$. $LCOM5 = \frac{\frac{1}{3}[4 + 6 + 6] - 10}{1 - 10} = 0.51$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

M ₁ : Point()	M ₂ : Show()	M ₃ : Hide()	M ₄ : Drag()	M ₅ : MoveTo()
I ₁ : {Visible, x, y}	I ₂ : {Visible, x, y}	I ₃ : {Visible}	I ₄ : {x, y}	I ₅ : {x, y}

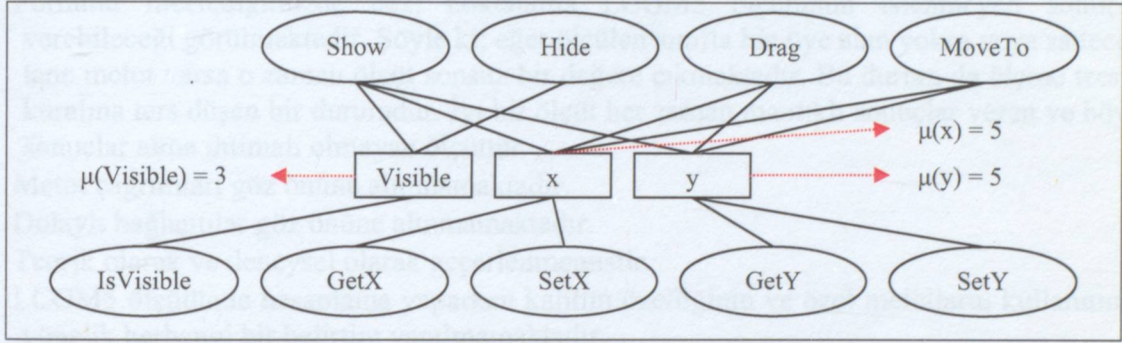


Şekil 4.26 LCOM5 (2) şikkına göre metot-üye alan ilişkisi ve μ değerleri

Üye alan sayısı $a = 3$ ve metod sayısı $m = 5$. $LCOM5 = \frac{\frac{1}{3}[3 + 4 + 4] - 5}{1 - 5} = 0.33$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

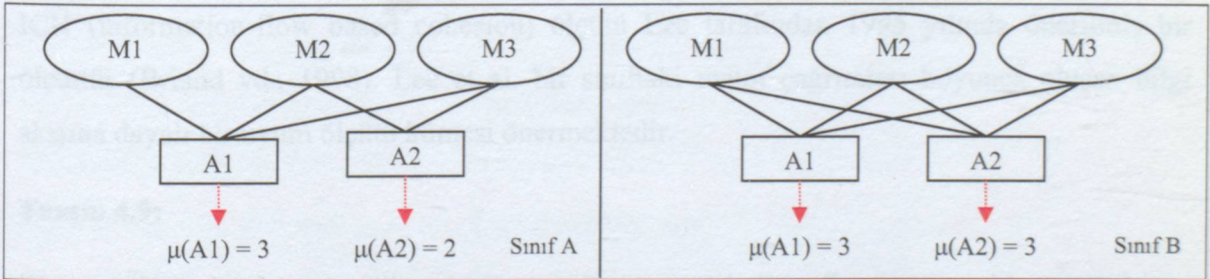
M ₂ : Show()	M ₃ : Hide()	M ₄ : Drag()	M ₅ : MoveTo()	M ₆ : IsVisible()
I ₂ : {Visible, x, y}	I ₃ : {Visible}	I ₄ : {x, y}	I ₅ : {x, y}	I ₆ : {Visible}
M ₇ : GetX()	M ₈ : SetX()	M ₉ : GetY()	M ₁₀ : SetY()	
I ₇ : {x}	I ₈ : {x}	I ₉ : {y}	I ₁₀ : {y}	



Şekil 4.27 LCOM5 (3) şıkına göre metod-üye alan ilişkisi ve μ değerleri

Üye alan sayısı $a = 3$ ve metod sayısı $m = 9$. $LCOM5 = \frac{\frac{1}{3}[3 + 5 + 5] - 9}{1 - 9} = 0.58$

Örnek 2:



Şekil 4.28 LCOM5 Sınıf A ve Sınıf B metod-üye alan ilişkisi ve μ değerleri

Sınıf A için; Üye alan sayısı $a = 2$ ve metod sayısı $m = 3$. $LCOM5 = \frac{\frac{1}{2}[3 + 2] - 3}{1 - 3} = 0.25$

$$\text{Sınıf B için; Üye alan sayısı } a = 2 \text{ ve metot sayısı } m = 3. \text{ LCOM5} = \frac{\frac{1}{2}[3+3]-3}{1-3} = 0$$

Değerlendirme:

- LCOM5 ölçütü, bir sınıfın her bir üye alanı en az 1 metot tarafından referans ediliyor varsayımı yapıldığı durumda 0 ve 1 arasında normalize edilmiş olmaktadır. Eğer bu varsayım çıkarılırsa bu durumda 0 ve 2 arasında normalize edilmiş olur. Halen bir normalize durum var fakat maximum değer 2 ancak $m = 2$ olduğunda geçerlidir. Bu durumu şu örnekle; LCOM5 hiçbir metot tarafından referans edilen üye alan olmadığı durumda ($\text{LCOM5} = m/m-1$) maximum değer 2 alıyor. $m > 2$ olduğu durumlarda maximum değer küçülüyor. Bu da bir anomali durumdur.
- Formülü incelediğimizde bazı noktalarda LCOM5 ölçütünün istenmeyen sonuçlar verebileceği görülmektedir. Şöyle ki; eğer ölçülen sınıfta hiç üye alan yoksa veya sadece 1 tane metot varsa o zaman ölçüt sonsuz bir değere çıkmaktadır. Bu durum da ölçme teorisi kuralına ters düşen bir durumdur. İyi bir ölçüt her zaman mantıklı sonuçlar veren ve böyle sonuçlar alma ihtimali olmayan ölçüttür.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Teorik olarak ve deneysel olarak geçerlenmemiştir.
- LCOM5 ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan LCOM5 ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak düşürdüğü, yapılandırıcı metotların *uyumu* yapay olarak arttırdığı görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan daha iyi çıkmalıdır. LCOM5 ölçütünün sezgilerle paralel sonuçlar verdiği görülmektedir.

4.5 Lee ve arkadaşları Tarafından Önerilen Ölçütler

4.5.1 ICH

ICH (information-flow based cohesion) ölçütü Lee tarafından 1995 yılında önerilmiş bir ölçüttür (Briand vd., 1998). Lee et al. bir sınıftaki metot çağrımları boyunca oluşan bilgi akışına dayalı bir uyum ölçütü kümesi önermektedir.

Tanım 4.9:

Bir c sınıfında implement edilen (yeni tanımlanan veya kalıtım ile alınıp yeniden tanımlanan “overriding”) metot m için, m metodunun uyumu c sınıfında gerçekleşen diğer metotların çağırılma sayısının çağırılan metotların parametre sayısı ile ağırlıklandırılması olarak ölçülür. Çağırılan metot ne kadar çok parametreye sahipse o kadar çok bilgi akışı vardır ve çağırılan ve çağırılan metot arasında o kadar çok güçlü bir bağ vardır demektir. Bir sınıfın uyumu, metotlarının uyumunun toplamı olarak ölçülür. Bir sınıf kümesinin uyumu ise kümedeki sınıfların uyumunun toplamı olarak ölçülür. Formül olarak bu ölçütler şu şekilde ifade edilir

(Briand vd., 1998).

$$\text{Metot } m \text{ için uyum ölçümü: } ICH(m) = \sum_{m' \in M_{NEW}(c) \cup M_{OVR}(c)} (1 + |Par(m')|) \cdot NPI(m, m') \quad (4.12)$$

$$\text{Sınıf } c \text{ için uyum ölçümü: } ICH(c) = \sum_{m \in M_I(c)} ICH(m) \quad (4.13)$$

$$\text{Sınıf kümesi } SS \text{ için uyum ölçümü: } ICH(SS) = \sum_{c \in SS} ICH(c) \quad (4.14)$$

$M_{NEW}(c)$: yeni tanımlanmış metotlar kümesi, $M_{OVR}(c)$: override edilmiş metotlar kümesi. $Par(m') = m'$ metodunun parametre kümesi, $NPI(m, m') = m$ metodu tarafından m' metodunun polimorfik çağrılma sayısı. $M_I(c) =$ yeni tanımlanmış veya kalıtım ile alınıp yeniden tanımlanmış (overriding) metotlar kümesi olarak tanımlanmaktadır.

ICH Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

M_1 : Point(): Çağrılan metotlar: {-}, Parametre sayısı: 2

M_2 : Show(): Çağrılan metotlar: {-}, Parametre sayısı: 0

M_3 : Hide(): Çağrılan metotlar: {-}, Parametre sayısı: 0

M_4 : Drag(): Çağrılan metotlar: {Show(), MoveTo()}, Parametre sayısı: 1

M_5 : MoveTo(): Çağrılan metotlar: {Show(), Hide()}, Parametre sayısı: 2

M_6 : IsVisible(): Çağrılan metotlar: {-}, Parametre sayısı: 0

M_7 : GetX(): Çağrılan metotlar: {-}, Parametre sayısı: 0

M_8 : SetX(): Çağrılan metotlar: {-}, Parametre sayısı: 1

M_9 : GetY(): Çağrılan metotlar: {-}, Parametre sayısı: 0

M_{10} : SetY(): Çağrılan metotlar: {-}, Parametre sayısı: 1

Point için uyum ölçümü: $ICH(Point) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

Show için uyum ölçümü: $ICH>Show) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

Hide için uyum ölçümü: $ICH(Hide) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

Drag için uyum ölçümü: $ICH(Drag) = [0 + ((1 + 0) \cdot 1) + 0 + ((1 + 2) \cdot 1) + 0 + 0 + 0 + 0 + 0] = 4$

MoveTo için uyum ölçümü:

$$ICH(MoveTo) = [0 + ((1+0).1) + ((1+0).1) + 0 + 0 + 0 + 0 + 0 + 0] = 2$$

IsVisible için uyum ölçümü: $ICH(IsVisible) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

GetX için uyum ölçümü: $ICH(GetX) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

SetX için uyum ölçümü: $ICH(SetX) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

GetY için uyum ölçümü: $ICH(GetY) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

SetY için uyum ölçümü: $ICH(SetY) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

Sınıf Point için uyum ölçümü:

$$ICH(Sınıf_Point) = ICH(Point) + ICH(Show) + ICH(Hide) + ICH(Drag) + ICH(MoveTo) + ICH(IsVisible) + ICH(GetX) + ICH(SetX) + ICH(GetY) + ICH(SetY)$$

$$ICH(Sınıf_Point) = 0 + 0 + 0 + 4 + 2 + 0 + 0 + 0 + 0 + 0 = 6$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

M_1 : *Point()*: Çağrılan metotlar: {-}, Parametre sayısı: 2

M_2 : *Show()*: Çağrılan metotlar: {-}, Parametre sayısı: 0

M_3 : *Hide()*: Çağrılan metotlar: {-}, Parametre sayısı: 0

M_4 : *Drag()*: Çağrılan metotlar: {*Show()*, *MoveTo()*}, Parametre sayısı: 1

M_5 : *MoveTo()*: Çağrılan metotlar: {*Show()*, *Hide()*}, Parametre sayısı: 2

Point için uyum ölçümü: $ICH(Point) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

Show için uyum ölçümü: $ICH(Show) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

Hide için uyum ölçümü: $ICH(Hide) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

Drag için uyum ölçümü: $ICH(Drag) = [0 + ((1+0).1) + 0 + ((1+2).1) + 0 + 0 + 0 + 0 + 0] = 4$

MoveTo için uyum ölçümü:

$$ICH(MoveTo) = [0 + ((1+0).1) + ((1+0).1) + 0 + 0 + 0 + 0 + 0 + 0] = 2$$

Sınıf Point için uyum ölçümü:

$$ICH(Sınıf_Point) = ICH(Point) + ICH(Show) + ICH(Hide) + ICH(Drag) + ICH(MoveTo)$$

$$ICH(Sınıf_Point) = 0 + 0 + 0 + 4 + 2 = 6$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

M₂: Show(): Çağrılan metotlar: {-}, Parametre sayısı: 0

M₃: Hide(): Çağrılan metotlar: {-}, Parametre sayısı: 0

M₄: Drag(): Çağrılan metotlar: {Show(), MoveTo()}, Parametre sayısı: 1

M₅: MoveTo(): Çağrılan metotlar: {Show(), Hide()}, Parametre sayısı: 2

M₆: IsVisible(): Çağrılan metotlar: {-}, Parametre sayısı: 0

M₇: GetX(): Çağrılan metotlar: {-}, Parametre sayısı: 0

M₈: SetX(): Çağrılan metotlar: {-}, Parametre sayısı: 1

M₉: GetY(): Çağrılan metotlar: {-}, Parametre sayısı: 0

M₁₀: SetY(): Çağrılan metotlar: {-}, Parametre sayısı: 1

Show için uyum ölçümü: $ICH(Show) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

Hide için uyum ölçümü: $ICH(Hide) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

Drag için uyum ölçümü: $ICH(Drag) = [0 + ((1 + 0).1) + 0 + ((1 + 2).1) + 0 + 0 + 0 + 0 + 0] = 4$

MoveTo için uyum ölçümü:

$$ICH(MoveTo) = [0 + ((1 + 0).1) + ((1 + 0).1) + 0 + 0 + 0 + 0 + 0 + 0] = 2$$

IsVisible için uyum ölçümü: $ICH(IsVisible) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

GetX için uyum ölçümü: $ICH(GetX) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

SetX için uyum ölçümü: $ICH(SetX) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

GetY için uyum ölçümü: $ICH(GetY) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

SetY için uyum ölçümü: $ICH(SetY) = (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$

Sınıf Point için uyum ölçümü:

$$ICH(Sınıf_Point) = ICH(Show) + ICH(Hide) + ICH(Drag) + ICH(MoveTo) + ICH(IsVisible) + ICH(GetX) + ICH(SetX) + ICH(GetY) + ICH(SetY)$$

$$ICH(Sınıf_Point) = 0 + 0 + 4 + 2 + 0 + 0 + 0 + 0 + 0 + 0 = 6$$

Örnek 2:

Sınıf A da; M1 metodunun M2 metodunu 1 kez çağırdığını ve M2 metodunun parametre sayısının 2 olduğunu varsayalım.

Sınıf B de; M2 metodunun M3 metodunu 1 kez çağırdığını ve M3 metodunun parametre sayısının 2 olduğunu varsayalım.

Sınıf A için;

M₁: Çağrılan metotlar: {M₂}, Parametre sayısı: 0

M₂: Çağrılan metotlar: {-}, Parametre sayısı: 2

M₃: Çağrılan metotlar: {-}, Parametre sayısı: 0

M₁ için uyum ölçümü: $ICH(M_1) = ((1 + 2) \cdot 1 + 0 = 2$

M₂ için uyum ölçümü: $ICH(M_2) = 0$

M₃ için uyum ölçümü: $ICH(M_3) = 0$

Sınıf A için uyum ölçümü:

$$ICH(SınıfA) = ICH(M_1) + ICH(M_2) + ICH(M_3) = 2 + 0 + 0 = 2$$

Sınıf B için;

M₁: Çağrılan metotlar: {-}, Parametre sayısı: 0

M₂: Çağrılan metotlar: {M₃}, Parametre sayısı: 0

M₃: Çağrılan metotlar: {-}, Parametre sayısı: 2

M₁ için uyum ölçümü: $ICH(M_1) = 0$

M₂ için uyum ölçümü: $ICH(M_2) = (0 + (1 + 2) \cdot 1) = 2$

M₃ için uyum ölçümü: $ICH(M_3) = 0$

Sınıf B için uyum ölçümü:

$$ICH(SınıfB) = ICH(M_1) + ICH(M_2) + ICH(M_3) = 0 + 2 + 0 = 2$$

Değerlendirme:

- ICH ölçütü normalize edilmemiştir.
- Üye alan kullanımı ve/veya paylaşımı göz önüne alınmamaktadır.

- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Teorik olarak geçerlenmiştir.
- ICH ölçütünde kalıtım özelliğinin hesaba katılacağı belirtilmiş ancak özel metotlarla ilgili herhangi bir şey belirtilmemiştir.
- Örnek 1 için yapılan ICH ölçümü sonucunda erişim metotlarının ve yapılandırıcı metotların göz önüne alınmasının ölçüm sonuçlarında herhangi bir değişikliğe yol açmadığı görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü Sınıf B'de metotlar daha fazla ortak üye alan kullanmaktadırlar. Ancak, ICH hesaplamasına bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize ICH ölçütünün sezgilerle paralellik göstermediğini, iki tane farklı uyum özelliğe sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da ICH ölçütünün *uyum* özelliğini çok da iyi ayırt edemediğini göstermektedir.
- ICH ölçütü, Briand ve arkadaşları tarafından yapılan çalışmalarda (Briand vd., 2000) deneysel olarak uyumu ölçmek için uygun bir ölçüt olmadığı anlamında geçerlenmiştir. ICH ölçütünün karmaşıklık ölçütü özellikleri taşıdığı yönünde görüşleri vardır. 2 açıdan diğer uyum ölçütlerinden farkı tespit edilmiştir. İlk olarak sadece metot çağrılarını ele almakta, sınıfın üye alanlarını hesaba katmamaktadır. İkinci olarak ICH teorik bir özelliğe sahiptir. ICH bir "katkı" görevi yapıyor. Eğer iki ilişkisiz fakat yüksek uyum özelliğe sahip sınıf tek bir sınıfta bir araya getirilirse, sınıfın uyumu her iki sınıfın ayrı uyum ölçümlerinin toplamı olacaktır. Bu da birleştirilmiş sınıfın her iki sınıftan daha yüksek uyuma sahip olmasına neden olacaktır. Bu sonuç beklenmeyen bir sonuçtur. Birleştirilmiş sınıfın bir objesi 2 ayrı anlam kavramlarını temsil edeceği için daha az uyum olmalıdır. Bu sonuç bize ICH ölçütünün pek de uyum ölçütü olmadığını göstermektedir.

4.6 Briand ve arkadaşları Tarafından Önerilen Ölçütler

4.6.1 RCI

Briand ve arkadaşları nesne tabanlı sistemler için bir uyum ölçüt kümesi önermişlerdir (Briand vd., 1997). 1997 yılında yaptıkları bir çalışmada bu ölçüt kümesini nesneye yönelik sistemlere adapte etmeye çalışmışlardır. Orijinal ölçütler sadece modül veya modüller kümesinin hiyerarşisi gibi yazılım parçalarına uygulanabilmektedir. Yeni ölçütleri sınıf seviyesine adapte olacak şekilde tanımlamakta fakat sınıf kümesini göz önüne almamaktadırlar.

Sınıf, bu ölçütler tanımlanırken veri tanımlamaları (data declarations) ve metotların bir koleksiyonu olarak görülmektedir. Veri tanımlamaları şunlardır:

- Yerel, public tip tanımlamalar
- Sınıfın kendisi
- Public üye alanlar (constant'ları da içerir)

Tanım 4.10: Veri-Veri etkileşimi (DD-interaction)

2 veri tanımlaması a ve b için, eğer a 'nın tanımlamasında veya kullanımında bir değişiklik olması b 'nin tanımlamasında veya kullanımında bir değişiklik ihtiyacına neden oluyorsa, a ve b arasında *Veri-Veri etkileşimi* var denir. *Veri-Veri etkileşimi* sadece 1 sınıfla sınırlı değildir. Farklı sınıfların üye alanları ve tip'leri arasında *Veri-Veri etkileşimi* olabilir. *Veri-Veri etkileşimi* ilişkisi geçişkendir. Yani, a ve b arasında *Veri-Veri etkileşimi* varsa ve b ve c arasında *Veri-Veri etkileşimi* varsa, a ve c arasında *Veri-Veri etkileşimi* vardır.

Örneğin;

- Eğer *public tip t* tanımlamasında bir başka *public tip t'* kullanıyorsa, o zaman t ve t' arasında *Veri-Veri etkileşimi* vardır.
- Eğer *public üye alan a* tanımlamasında *public tip t* kullanıyorsa, o zaman t ve a arasında *Veri-Veri etkileşimi* vardır.
- Eğer *public üye alan a* bir dizi ise ve tanımlamasında *public constant a'* kullanıyorsa, o zaman a ve a' arasında *Veri-Veri etkileşimi* vardır.

Tanım 4.11: Veri-Metot etkileşimi (DM-interaction)

Veri tanımlaması a ve metot m için, eğer a , m metodunun en az bir veri tanımlaması ile *Veri-Veri etkileşimi*'ne sahipse o zaman veri tanımlaması a ve metot m arasında *Veri-Metot etkileşimi* vardır. Metotların veri tanımlamaları, onların parametrelerini, geri dönüş tiplerini ve yerel değişkenlerini içerir.

Eğer bir c sınıfının bir m metodu c sınıfının a üye alanını kullanıyorsa veya erişimi varsa, m metodu ve c sınıfının a üye alanı arasında *Veri-Metot etkileşimi* vardır.

Veri tanımlamaları arasındaki bütün *Veri-Metot etkileşimi*'leri ve parametreleri ve geri dönüş tipleri içeren *Veri-Metot etkileşimi*'leri sınıf arayüzünden anlaşılabilir ve böylece geliştirme sürecinin erken safhasında mevcut olurlar.

Bu ifadelerle göre RCI (ratio of cohesive interactions) şöyle tanımlanmaktadır:

Tanım 4.12:

$$RCI(c) = \frac{|CI(c)|}{Max(c)} \quad (4.15)$$

$CI(c)$ (Cohesive Interactions): bütün Veri-Veri- ve Veri-Metot- etkileşimleri kümesi

$Max(c)$: sınıfın bütün mümkün olan Veri-Veri- ve Veri-Metot-etkileşimleri kümesi

Yani, verilen bir sınıf için RCI, gerçek etkileşimlerin mümkün olan tüm etkileşimlere oranıdır. RCI, 0 ve 1 arasında bir değer alır; 0 minimum uyumu, 1 maximum uyumu gösterir.

RCI Ölçütü Hesaplaması:

Örnek 1:

Bu örnekte veri tanımlamasına uyan veri bulunmadığından örnek uygulanamamaktadır. Örneğin; üye alanlar *private* tanımlanmıştır ve bu durum RCI ölçütünde tanımlanan veri tanımlamasına uymamaktadır.

Örnek 2:

$$\text{Sınıf A için; } CI(c) = 5, \quad \text{Max}(c) = 3 \times 2 = 6, \quad RCI(c) = \frac{5}{6}$$

$$\text{Sınıf B için; } CI(c) = 6, \quad \text{Max}(c) = 3 \times 2 = 6, \quad RCI(c) = \frac{6}{6} = 1$$

Değerlendirme:

- RCI ölçütü değeri 0 ve 1 arasında değişmekte ve 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Metot çağrıları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Teorik olarak ve deneysel olarak geçerlenmiştir.
- RCI ölçütünde hesaplama yaparken özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Briand ve arkadaşları analiz yaparken “kalıtım alınmış üye alanları analizden çıkarmak, kalıtım alınmış metotları analize almak” seçeneğini önermektedirler.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A’dan daha iyi çıkmalıdır. RCI ölçütünün sezgilerle paralel sonuçlar verdiği görülmektedir.
- Yapılan bir çalışmada (Briand vd., 1999) RCI ölçütünün nesne tabanlı programlamada hata-eğilimliğinin iyi bir tahmin edicisi olduğu raporlanmıştır.

4.6.2 Coh

Coh ölçütü Briand tarafından 1998 yılında önerilmiş bir ölçüttür (Briand vd., 1998). LCOM5 ölçütünün bir varyasyonudur.

Tanım 4.13:

Coh ölçütü LCOM5 ölçütüne benzer fakat burada düzgün bir şekilde normalize edilmiş durum vardır. Coh ölçütü, yüksek değerler için yüksek uyumu düşük değerler için düşük uyumu gösterir. Coh formül olarak şu şekilde verilmektedir:

$$Coh = \frac{\sum_{j=1}^a \mu(A_j)}{m.a} \quad (4.16)$$

Coh Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

Metot sayısı: $m = 10$, Üye alan sayısı: $a = 3$, $Coh = \frac{(4+6+6)}{10.3} = 0.53$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

Metot sayısı: $m = 5$, Üye alan sayısı: $a = 3$, $Coh = \frac{(3+4+4)}{5.3} = 0.73$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

Metot sayısı: $m = 9$, Üye alan sayısı: $a = 3$, $Coh = \frac{(3+5+5)}{9.3} = 0.48$

Örnek 2:

Sınıf A için; Metot sayısı: $m = 3$, Üye alan sayısı: $a = 2$, $Coh = \frac{(3+2)}{3.2} = 5/6 = 0.83$

Sınıf B için; Metot sayısı: $m = 3$, Üye alan sayısı: $a = 2$, $Coh = \frac{(3+3)}{3.2} = 1$

Değerlendirme:

- Coh ölçütü değeri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Teorik ve deneysel geçerlemesi yapılmamaktadır.
- Coh ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan Coh ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak düşürdüğü, yapılandırıcı metotların *uyumu* yapay olarak arttırdığı görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan daha iyi çıkmalıdır. Coh ölçütünün sezgilerle paralel sonuçlar verdiği görülmektedir.

4.7 Chae ve arkadaşları Tarafından Önerilen Ölçütler

4.7.1 CBMC

CBMC (Cohesion Based on Member Connectivity) ölçütü Chae ve arkadaşları tarafından 2000 yılında önerilmiştir (Chae vd., 2000). Var olan ölçütlerin şu iki ana problemi vurgulanmış ve CBMC ölçütü önerilmiştir:

- Çoğu ölçüt özel metotları hesaba katmıyor
- Çoğu ölçüt sınıfın üyeleri arasındaki etkileşim kalıplarını göz önüne almıyor, sadece metotlar tarafından referans edilen üye alan veya ortak üye alan kullanan metot çifti sayısını hesaplıyor.

Tanım 4.14:

Verilen bir c sınıfı için, metotlar ve üye alanlar arasındaki erişim ilişkilerini temsil eden $G(c)$ referans grafiği kullanılmaktadır. Bu grafik şu şekilde tanımlanmaktadır;

$$G(c) = (N(c), E(c)) \quad (4.17)$$

$$N(c) = M(c) \cup V(c) \quad (4.18)$$

$$E(c) = \{(m, v) \mid v \in V(c) \wedge m \in M(c), v \in R^*(m)\} \quad (4.19)$$

$M(c)$ ve $V(c)$, c sınıfının metotları ve üye alanları kümesini, $E(c)$, kenarları göstermektedir.

Sınıfın bütün üyeleri olası bütün etkileşimlere sahipse “most cohesive (çoğu uyumlu)” sınıf olabilir. Fakat, özel metotlar sadece sınıftaki bazı üye alanlarla etkileşerek görevlerini tamamlarlar. Bu nedenle, özel metotlar bütün üye alanlarla etkileşmek zorunda değildirler. Bu nedenle, “most cohesive (çoğu uyumlu)” sınıf, *normal metotları* bütün üye alanlarla etkileşen sınıf olarak tanımlanır.

Tanım 4.15: En Uyumlu Bileşen (Most Cohesive Component-MCC)

$M_n(G)$ kümesindeki her bir normal metot $V(c)$ kümesinin her elemanı ile etkileşiyorsa $G(c)(N, E)$ referans grafiği bir En Uyumlu Bileşen’dir.

$$E(c) = \{(m, v) \mid v \in V(G) \wedge m \in M_n(G)\} \quad (4.20)$$

Tanım 4.16: Bağlılık (Connectivity)

Bir sınıfın elemanları ne kadar güçlü bağlıysa o kadar uyumludur. Sınıfın üyeleri arasındaki

bağıllık uyumu etkiler.

G referans grafiğinin yapışkan metotlar (glue methods) denilen metotları, G grafiğinin alt grafiklere ayrılamaz en küçük metot kümesidir. $M_g(G)$ diye simgelenir.

G referans grafiğinin bağıllık çarpanı (connectivity factor) $F_c(G)$ üyeler arasındaki bağıllık derecesini temsil eder ve yapışkan metotların $|M_g(G)|$ sayısının normal metotların $M_n(G)$ sayısına oranı olarak tanımlanır.

$$F_c(G) = \frac{|M_g(G)|}{|M_n(G)|}, [0, 1] \quad (4.21)$$

Tanım 4.17: Yapı Ağacı (Structure Tree)

Sınıfın etkileşim modellerinin hiyerarşik yapısını incelemek için oluşturulur. Yapı ağacı, yapışkan metotlar çıkartılınca sınıfın nasıl bileşenlerine ayrılacağını tasvirler.

G referans grafiğinin $T(G)$ yapı ağacı, kök düğümü (root node) G olan $T = (N, E)$ bütün yaprakları MCC olan şu şekilde bir ağaçtır;

$$N = \{G^i \mid G^i \subseteq G\} \quad (4.22)$$

$$E = \left\{ \left(G^p, G^c \right) \mid G^p, G^c \text{ grafiğinden } M_g(G^p) \text{ kümesinin çıkarılmasıyla elde edilen alt referans grafiklerinden biri} \right\} \quad (4.23)$$

Bir referans grafiği için yapı ağacı, yapışkan metotların ve ilişkili etkileşimlerin referans grafiğinden çıkarılmasıyla elde edilen bileşenlerle oluşturulur. Bu bileşenlere ayırma işlemi, her bir bileşen MCC olana dek özyineli (recursive) olarak uygulanır. Bir referans grafiğinin bileşenlere ayrılması her zaman tekil değildir çünkü birkaç farklı yapışkan metotlar kümesi tarafından ayrılabilir. Böyle bir durumda en yüksek uyuma sahip çocuklarla sonuçlanan yapışkan metotlar kümesi seçilir.

Chae, *uyumu* hesaplarken bileşenlerin uyumluluğunu hesaba katmanın önemli olduğunu vurgulamıştır.

Tanım 4.18: Yapı Çarpanı (Structure Factor)

Bir referans grafiği G için, yapı çarpanı $F_s(G)$, sınıfın bileşenlerinin uyuma katkı derecelerini gösterir ve referans grafiğinin yapı ağacındaki çocuklarının ortalaması olarak şu şekilde tanımlanır.

$$F_s(G) = \frac{1}{n} \times \sum_{i=1}^n CBMC(G^i) \quad , [0, 1] \quad (4.24)$$

G^i : G grafiğinin n çocuğundan biri

$CBMC(G^i)$: G^i bileşeninin uyumu

Bu tanımlamalar kullanılarak CBMC ölçütü şu şekilde tanımlanmaktadır;

Tanım 4.19: CBMC

Bir C sınıfı için CBMC referans grafiğinin bağıllık çarpanı $F_c(G(C))$ 'ünün yapı çarpanı $F_s(G(C))$ ' ü ile ölçeklendirilmesi şeklinde tanımlanır.

$$CBMC(C) = F_c(G(C)) \times F_s(G(C)) \quad (4.25)$$

$$CBMC(C) = F_c(G(C)) \times \frac{1}{n} \sum_{i=1}^n CBMC(G^i) \quad (4.26)$$

Özel metot'ların uyuma etkisi yoktur diye düşünülmüşlerdir. CBMC özel metotlardan bağımsız olarak tanımlanmıştır.

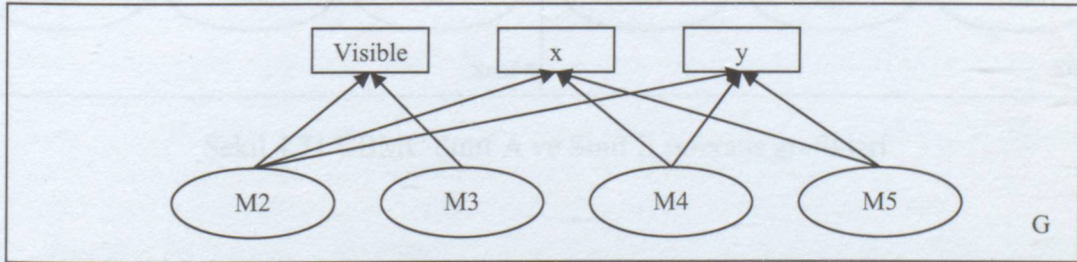
CBMC Ölçütü Hesaplaması:

Örnek 1:

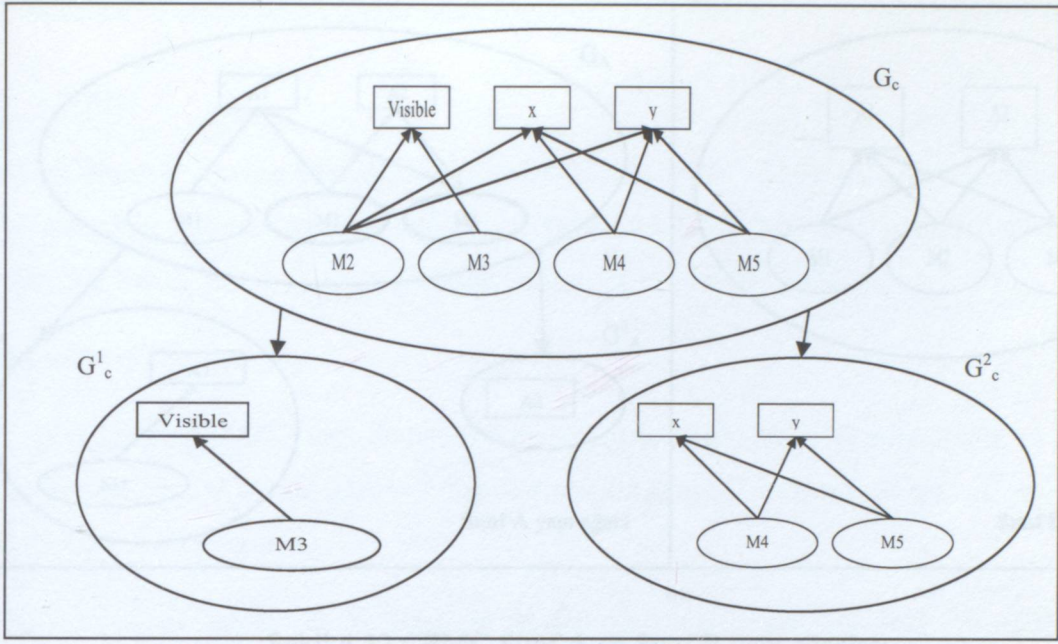
1) Erişim metotları ve yapılandırıcı metotlar hesaplamaya alınmayacak

M_2 : Show() M_3 : Hide() M_4 : Drag() M_5 : MoveTo()

I_2 : {Visible, x, y} I_3 : {Visible} I_4 : {x, y} I_5 : {x, y}

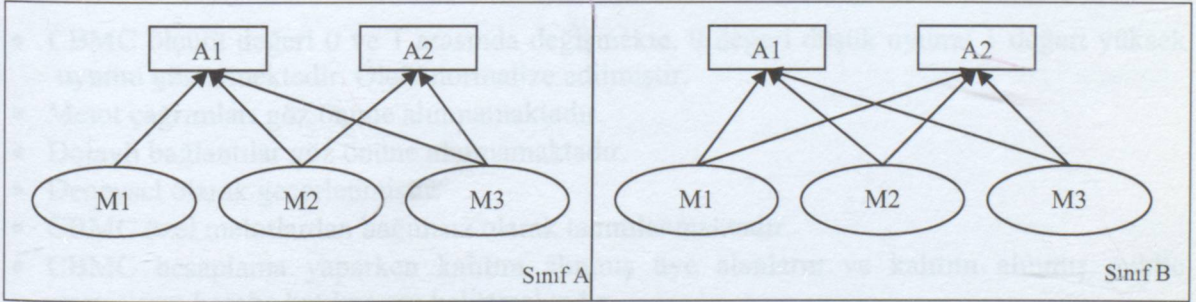


Şekil 4.29 CBMC (1) şikkına göre referans grafiği G

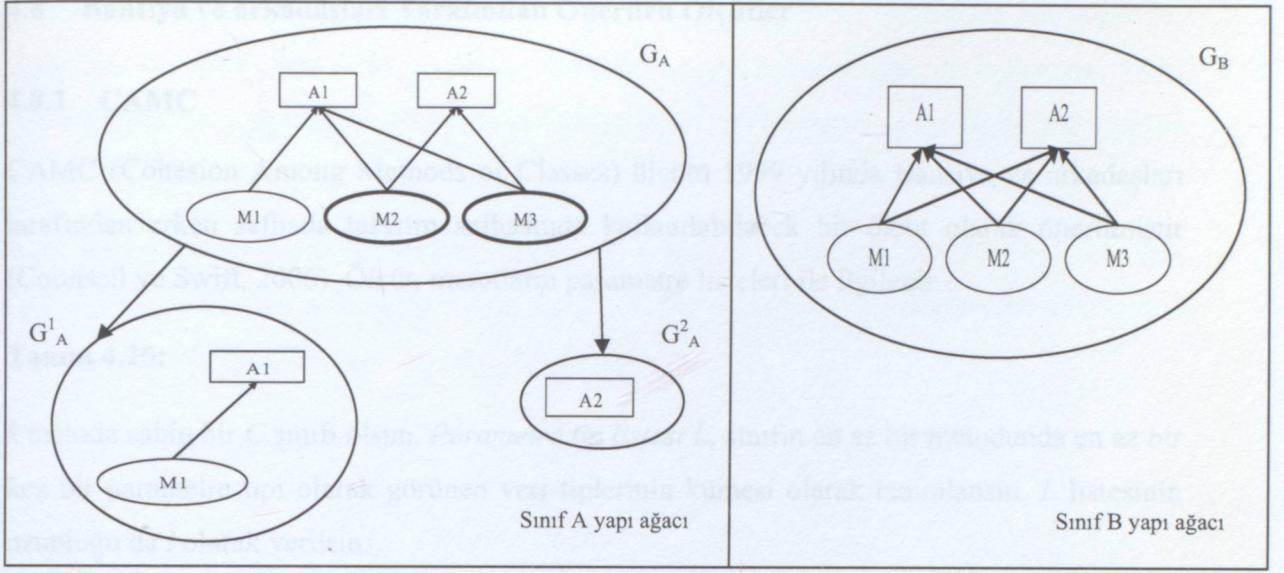


Şekil 4.30 CBMC (1) şıkkına göre yapı ağacı

$$CBMC(C) = F_c(G(C)) \times \frac{1}{2} [CBMC(G_c^1) + CBMC(G_c^2)], \quad CBMC(C) = \frac{1}{4} \times \frac{1}{2} [1 + 1] = 0.25$$

Örnek 2:

Şekil 4.31 CBMC Sınıf A ve Sınıf B referans grafikleri



Şekil 4.32 CBMC Sınıf A ve Sınıf B yapı ağaçları

$$CBMC(A) = F_C(G(A)) \times \frac{1}{2} \left[CBMC(G_A^1) + CBMC(G_B^2) \right]$$

$$CBMC(A) = \frac{2}{3} \times \frac{1}{2} [1 + 1] = 0.66, \quad CBMC(B) = 1$$

Değerlendirme:

- CBMC ölçütü değeri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Metot çağrıları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Deneysel olarak geçirilenmiştir.
- CBMC özel metotlardan bağımsız olarak tanımlanmaktadır.
- CBMC hesaplama yaparken kalıtım alınmış üye alanların ve kalıtım alınmış public metotların hesaba katılmasını belirtmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan daha iyi çıkmalıdır. CBMC ölçütünün sezgilerle paralel sonuçlar verdiği görülmektedir
- CBMC'nin diğer ölçütlerden farklı yaklaşımı şu sonuçları doğurmaktadır;
 - Bu ölçüt ile karmaşık etkileşimleri olan bir sınıf basit etkileşimleri olan bir sınıf ile aynı uyum ölçümüne sahip çıkabilmektedir. (g grafiğine bir etkileşim kenarı eklendiğinde eğer bileşenleri sayısı azalır ve yapı ağaçları yüksekliği değişmezse)
 - Bu ölçüt ile karmaşık etkileşimleri olan bir sınıf basit etkileşimleri olan bir sınıftan daha düşük bir uyum ölçümüne sahip çıkabilmektedir. (g grafiğine bir etkileşim kenar eklendiğinde eğer bileşenleri sayısı azalır ve yapı ağaçları yüksekliği artarsa)

4.8 Bansiya ve arkadaşları Tarafından Önerilen Ölçütler

4.8.1 CAMC

CAMC (Cohesion Among Methods of Classes) ölçütü 1999 yılında Bansiya ve arkadaşları tarafından erken safhada tasarım safhasında kullanılabilecek bir ölçüt olarak önerilmiştir (Counsell ve Swift, 2006). Ölçüt, metotların parametre listeleri ile ilgilenir.

Tanım 4.20:

k metoda sahip bir C sınıfı olsun. *Parametre tip listesi* L , sınıfın en az bir metodunda en az bir kez bir parametre tipi olarak görünen veri tiplerinin kümesi olarak tanımlansın. L listesinin uzunluğu da l olarak verilsin.

Parametre oluşum matrisi O , L listesinin üyeleri ile indekslenen kolonlara ve metotlarla indekslenen satırlara sahip olsun, $1 \leq i \leq k$, $1 \leq j \leq l$.

Parametre oluşum matrisi O , $k \times l$ matristir.

$$o_{ij} = \begin{cases} 1 & , j. \text{ veri tipi } i. \text{ metodda parametre olarak bulunuyorsa} \\ 0 & , \text{ diğer türlü} \end{cases} \quad (4.27)$$

Parametre oluşum matrisi O ' nun i . satırı *Parametre oluşum vektörü* olarak tanımlanıyor (i metodu için). *Parametre oluşum vektörü*, bit modellidir, veri tiplerinin i . metotta bulunup bulunmadığını gösterir. i . satırdaki 1'lerin sayısı r_i , j . kolondaki 1'lerin sayısı c_j ve *Parametre oluşum matrisi* O 'daki 1'lerin sayısı σ olarak simgeleniyor. Şöyle ki;

$$r_i = \sum_{j=1}^l o_{ij} \quad (4.28)$$

$$c_j = \sum_{i=1}^k o_{ij} \quad (4.29)$$

$$\sigma = \sum_{i=1}^k \sum_{j=1}^l o_{ij} \quad (4.30)$$

Yukarıdaki bilgiler doğrultusunda bir C sınıfı için CAMC ölçütü şu şekilde hesaplanır:

$$CAMC(C) = \frac{1}{k.l} \sum_{i=1}^k \sum_{j=1}^l o_{ij} = \frac{\sigma}{k.l} \quad (4.31)$$

Diğer bir ifadeyle, CAMC ölçütü, *Parametre oluşum matrisi* 'nin değerlerinin ortalamasıdır. Yani, her bir metot için farklı parametre tiplerinin sayısı hesaplanıyor ve toplanıyor. Daha sonra bu toplam, metot sayısı ve sınıftaki farklı parametre tipi sayısının toplamının çarpımına bölünüyor.

Eğer sınıfın metotları hiç parametre içermiyorsa o zaman CAMC ölçütünü hesaplamak için *self parametre* denilen parametre hesaba katılıyor. *Self parametre*, sınıfın her bir metodunun kapalı olarak içerdiği *sınıf tipi*'dir.

CAMC Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

Çizelge 4.1 CAMC (1) şikkına göre metotlar ve parametre oluşum matrisi

<i>Metotlar:</i>	<i>Parametre oluşum matrisi:</i>																						
Point(int InitX, int InitY); virtual void Show(); virtual void Hide (); virtual void Drag(int DragBy); void MoveTo(int NewX, int NewY); Boolean IsVisible(); Public int GetX(); Public int GetY(); Public void SetX(int x); Public void SetY(int y);	<table border="1"> <thead> <tr> <th>O</th><th>int</th></tr> </thead> <tbody> <tr><td>Point</td><td>1</td></tr> <tr><td>Show</td><td>0</td></tr> <tr><td>Hide</td><td>0</td></tr> <tr><td>Drag</td><td>1</td></tr> <tr><td>MoveTo</td><td>1</td></tr> <tr><td>IsVisible</td><td>0</td></tr> <tr><td>GetX</td><td>0</td></tr> <tr><td>GetY</td><td>0</td></tr> <tr><td>SetX</td><td>1</td></tr> <tr><td>SetY</td><td>1</td></tr> </tbody> </table>	O	int	Point	1	Show	0	Hide	0	Drag	1	MoveTo	1	IsVisible	0	GetX	0	GetY	0	SetX	1	SetY	1
O	int																						
Point	1																						
Show	0																						
Hide	0																						
Drag	1																						
MoveTo	1																						
IsVisible	0																						
GetX	0																						
GetY	0																						
SetX	1																						
SetY	1																						

$$k = 10 \text{ ve } l = 1 \text{ dir, } CAMC = \frac{1}{10.1} (1 + 0 + 0 + 1 + 1 + 0 + 0 + 0 + 1 + 1) = 0.5$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

Çizelge 4.2 CAMC (2) şikkına göre metotlar ve parametre oluşum matrisi

<i>Metotlar:</i>	<i>Parametre oluşum matrisi:</i>												
Point(int InitX, int InitY); virtual void Show(); virtual void Hide (); virtual void Drag(int DragBy); void MoveTo(int NewX, int NewY);	<table border="1"> <thead> <tr> <th>O</th><th>int</th></tr> </thead> <tbody> <tr><td>Point</td><td>1</td></tr> <tr><td>Show</td><td>0</td></tr> <tr><td>Hide</td><td>0</td></tr> <tr><td>Drag</td><td>1</td></tr> <tr><td>MoveTo</td><td>1</td></tr> </tbody> </table>	O	int	Point	1	Show	0	Hide	0	Drag	1	MoveTo	1
O	int												
Point	1												
Show	0												
Hide	0												
Drag	1												
MoveTo	1												

$k = 5$ ve $l = 1$ dir,

$$CAMC = \frac{1}{5.1} (1 + 0 + 0 + 1 + 1) = 0.6$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

Çizelge 4.3 CAMC (3) şikkına göre metotlar ve parametre oluşum matrisi

Metotlar:	Parametre oluşum matrisi:																				
virtual void Show(); virtual void Hide (); virtual void Drag(int DragBy); void MoveTo(int NewX, int NewY); Boolean IsVisible(); Public int GetX(); Public int GetY(); Public void SetX(int x); Public void SetY(int y);	<table border="1"> <thead> <tr> <th>O</th><th>int</th></tr> </thead> <tbody> <tr><td>Show</td><td>0</td></tr> <tr><td>Hide</td><td>0</td></tr> <tr><td>Drag</td><td>1</td></tr> <tr><td>MoveTo</td><td>1</td></tr> <tr><td>IsVisible</td><td>0</td></tr> <tr><td>GetX</td><td>0</td></tr> <tr><td>GetY</td><td>0</td></tr> <tr><td>SetX</td><td>1</td></tr> <tr><td>SetY</td><td>1</td></tr> </tbody> </table>	O	int	Show	0	Hide	0	Drag	1	MoveTo	1	IsVisible	0	GetX	0	GetY	0	SetX	1	SetY	1
O	int																				
Show	0																				
Hide	0																				
Drag	1																				
MoveTo	1																				
IsVisible	0																				
GetX	0																				
GetY	0																				
SetX	1																				
SetY	1																				

$k = 9$ ve $l = 1$ dir,

$$CAMC = \frac{1}{9.1} (0 + 0 + 0 + 1 + 1 + 0 + 0 + 0 + 1 + 1) = 0.44$$

Örnek 2:

Bu örnekte parametreler belirtilmediği için bu ölçüt uygulanamamaktadır.

Değerlendirme:

- CAMC ölçütü değeri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir. Bansiya ve arkadaşları 0.35 ve üzeri CAMC değerine sahip sınıfları uyumlu sınıf kabul etmektedirler.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Deneysel olarak geçirilenmiştir.
- CAMC ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan CAMC ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak düşürdüğü, yapılandırıcı metotların *uyumu* yapay olarak arttırdığı görülmektedir.
- CAMC ölçütünün formülü incelendiğinde bu ölçütün, Parametre oluşum matrisleri 'nde aynı sayıda 1 olan iki ayrı sınıfı ayırt edemediği görülmektedir.
- CAMC ölçütü küçük sınıfları yüksek uyuma sahip olarak hesaplamaktadır.
- 0.35 değerini eşik değeri kabul etmeleri hatalı. (eğer self parametre hesaba katılırsa)
- Sadece parametre tipleri ile ilgileniyor fakat metotlar içyapıları ile de birbiriyle ilişkili olabilirler.
- Sadece parametre tiplerinin benzerliği ile ilgileniyor onların sınıf içinde kullanımı ile ilgilenmiyor. İki parametre aynı tipte olabilir fakat ikisi farklı fonksiyonellik için kullanılıyor olabilir, ikisi benzer olarak göz önüne alınmamış olabilir.
- Sadece parametre tiplerinin benzerliğini hesaba katarak *uyumu* ölçmek gerçek sonuçları yansıtmayabilir.

- Yapılan bir çalışmada (Bonja ve Kidanmariam, 2006) sezgisel olarak beklenen sonuçları vermediği görülmektedir.

4.9 Counsell ve arkadaşları tarafından Önerilen Ölçütler

4.9.1 NHD

NHD (Normalised Hamming Distance) ölçütü Counsell ve arkadaşları tarafından 2002 yılında önerilmiş bir ölçüttür (Counsell ve Swift, 2006). Ölçüt, CAMC ölçütünde olduğu gibi metotların parametre listeleri ile ilgilenir. İkili matriste satırlar arası uyuşmayı ölçer. NHD ölçütü, bu anlamda CAMC ölçütünün bir alternatifi olarak kullanılabilir.

Tanım 4.21:

m_i ve m_j metodu arasındaki *Parametre uyuşması*, iki metodun *Parametre oluşum vektörü* 'ndeki eşit olan yerlerin sayısı olarak bulunmaktadır. *Parametre uyuşma matrisi* A ise $k-1$ boyutunda düşük üçgensel kare matristir, a_{ij} , m_i ve m_j metodu arasındaki *Parametre uyuşması* olarak tanımlanmaktadır, $1 \leq j < i \leq k$ ve 0 diğer türlü ise.

Bu bilgiler doğrultusunda bir C sınıfı için NHD ölçütü şöyle tanımlanmaktadır:

$$NHD(C) = \frac{1}{l \cdot \binom{k}{2}} \sum_{j=1}^{k-1} \sum_{i=j+1}^k a_{ij} = \frac{2}{l \cdot k(k-1)} \cdot \sum_{j=1}^{k-1} \sum_{i=j+1}^k a_{ij} \quad (4.32)$$

Diğer bir ifadeyle, NHD ölçütü, metotların bütün parametreler üzerindeki uyuşmazlıklarının toplamının 1'den çıkartılmasıyla da bulunabilir. Bu yöntemle bulunursa, *Parametre uyuşma matrisi* oluşturulmasına gerek yoktur.

CAMC ölçütünde karşılaşılan "*Parametre oluşum matrisleri* 'nde aynı sayıda 1 olan iki ayrı sınıfı ayırt edememe" problemi NHD için geçerli değildir. NHD ölçütü, *Parametre oluşum matrisleri* 'nde aynı sayıda 1 olan iki ayrı sınıfı ayırt edebilmektedir. Bu durum teorik olarak ispatlanmıştır.

Verilen sabit σ sayısında, NHD ölçütü için minimum ve maximum değerlerin elde edilebileceği *Parametre oluşum matrisleri* vardır.

NHD Ölçütü Hesaplaması:

Örnek 1:

CAMC ölçütünde parametre oluşum matrisleri oluşturulmuştu. Parametre uyuşma matrisleri

ise her şık için oluşturulmaktadır.

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

Çizelge 4.4 NHD (1) şikkına göre parametre uyuşma matrisi

A	Point	Show	Hide	Drag	MoveTo	IsVisible	GetX	GetY	SetX
Show	0								
Hide	0	0							
Drag	1	0	0						
MoveTo	1	0	0	1					
IsVisible	0	0	0	0	0				
GetX	0	0	0	0	0	0			
GetY	0	0	0	0	0	0	0		
SetX	1	0	0	1	1	0	0	0	
SetY	1	0	0	1	1	0	0	0	1
	4	0	0	3	2	0	0	0	1

k = 10 ve l = 1 dir,

$$NHD = \frac{1}{1.10.9} (4 + 0 + 0 + 3 + 2 + 0 + 0 + 0 + 1) = 0.11$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

Çizelge 4.5 NHD (2) şikkına göre parametre uyuşma matrisi

A	Point	Show	Hide	Drag
Show	0			
Hide	0	0		
Drag	1	0	0	
MoveTo	1	0	0	1
	2	0	0	1

k = 5 ve l = 1 dir,

$$NHD = \frac{1}{1.5.4} (2 + 0 + 0 + 1) = 0.15$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

Çizelge 4.6 NHD (3) şikkına göre parametre uyuşma matrisi

A	Show	Hide	Drag	MoveTo	IsVisible	GetX	GetY	SetX
Hide	0							
Drag	0	0						
MoveTo	0	0	1					
IsVisible	0	0	0	0				
GetX	0	0	0	0	0			
GetY	0	0	0	0	0	0		
SetX	0	0	1	1	0	0	0	
SetY	0	0	1	1	0	0	0	1
	0	0	3	2	0	0	0	1

$k = 9$ ve $l = 1$ dir,

$$NHD = \frac{1}{1.9.8} (0 + 0 + 3 + 2 + 0 + 0 + 0 + 1) = 0.08$$

Örnek 2:

Bu örnekte parametreler belirtilmediği için bu ölçüt uygulanamamaktadır.

Değerlendirme:

- NHD ölçütü değeri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir. Counsell ve arkadaşları 0.5 ve üzeri NHD değerine sahip sınıfları uyumlu sınıf kabul etmektedirler. Fakat bu öneri de güçlü bir şekilde savunulamamıştır.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Deneysel olarak geçerlenmiştir.
- NHD ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan CAMC ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak düşürdüğü, yapılandırıcı metotların *uyumu* yapay olarak arttırdığı görülmektedir.
- NHD ölçütü, *Parametre oluşum matrisleri* 'nde aynı sayıda 1 olan iki ayrı sınıfı ayırt edebilmektedir. Bu durum teorik olarak ispatlanmıştır.
- NHD ölçütünün, CAMC ölçütünden daha yüksek sonuçlar elde ettiği görülmektedir; NHD ölçütünün büyük sınıflarda yüksek uyumu bulabildiği buna karşın CAMC ölçütünün küçük sınıflarda yüksek uyumu bulabildiği görülmektedir.
- Sadece parametre tipleri ile ilgileniyor fakat metotlar içyapıları ile de birbiriyle ilişkili olabilirler.
- Sadece parametre tiplerinin benzerliği ile ilgileniyor onların sınıf içinde kullanımı ile ilgilenmiyor. İki parametre aynı tipte olabilir fakat ikisi farklı fonksiyonellik için kullanılıyor olabilir, ikisi benzer olarak göz önüne alınmamış olabilir.
- Sadece parametre tiplerinin benzerliğini hesaba katarak *uyumu* ölçmek gerçek sonuçları yansıtmayabilir.
- Bütün metotları da parametresiz olan bir sınıf için NHD ölçütü uygulanamamaktadır.

4.10 Bonja ve arkadaşları Tarafından Önerilen Ölçütler

4.10.1 CC(X)

CC(X) (Class Cohesion) ölçütü Bonja ve arkadaşları 2006 yılında bir sınıfın metotları arasındaki benzerliğe dayanan yeni bir ölçüt olarak önerilmiştir (Bonja ve Kidanmariam, 2006). Bu ölçütte bir sınıfın metotları arasındaki benzerlik mantığı olarak metotların üye alan kullanımında nasıl ilişkili oldukları hususu temel alınmaktadır. Ölçüt, temel olarak LCOM ölçütünün eksikliklerini adreslemektedir. Sadece iki metodun üye alan kümelerinin kesişiminin boş olmadığı durumun yerine, kesişimindeki üye alanların sayısı ile birleşimindeki üye alanların sayısının ilişkisi göz önüne alınmaktadır.

Tanım 4.22:

İki metot M_1 ve M_2 göz önüne alalım. $\{IV_1\}$ ve $\{IV_2\}$ de M_1 ve M_2 metotları tarafından kullanılan üye alanlar kümesi olsun. $|IV|_t = \{IV_1\} \cup \{IV_2\}$ kümelerinin miktarı (cardinality) olsun. $|IV|_c = \{IV_1\} \cap \{IV_2\}$ kümelerinin miktarı (cardinality) olsun. Yani, $|IV|_t$, iki metot tarafından kullanılan toplam üye alan sayısı, $|IV|_c$ de iki metot tarafından kullanılan ortak üye alan sayısı olsun. Metot benzerliği (MS) şu şekilde tanımlanmaktadır:

$$MS = \frac{|IV|_c}{|IV|_t} \quad (4.33)$$

Tanım 4.23:

n metoda sahip bir X sınıfı olsun. Sınıftaki metot çiftleri verildiğinde, $|IV|_t$, iki metot tarafından kullanılan toplam üye alan sayısı, $|IV|_c$ de iki metot tarafından kullanılan ortak üye alan sayısı olsun. Farklı kombinasyonları belirlemek için metot çiftlerini temsilen indeks olarak i kullanılıyor. n metoduna sahip bir sınıfta $\frac{n!}{2 \cdot (n-2)!}$ adet farklı kombinasyon vardır, i ,

1'den $\frac{n!}{2 \cdot (n-2)!}$ arasından değişir. Yukarıda belirtilen Metot benzerliği (MS) temel alınarak

$CC(X)$ ölçütü şu şekilde tanımlanmaktadır:

$$CC(X) = \frac{2(n-2)!}{n!} \sum_{i=1}^{\frac{n!}{2(n-2)!}} \frac{|IV|_c}{|IV|_t} i \quad (4.34)$$

Kısaca, $CC(X)$ ölçütüne dayanan sınıfın uyumu, sınıftaki bütün metotların kombinasyonu için metotlar arasındaki benzerliklerin ortalaması olarak düşünülebilir.

 $CC(X)$ Ölçütü Hesaplaması:**Örnek 1:****1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak**

M_1 : Point()	M_2 : Show()	M_3 : Hide()	M_4 : Drag()	M_5 : MoveTo()
I_1 : {Visible, x, y}	I_2 : {Visible, x, y}	I_3 : {Visible}	I_4 : {x, y}	I_5 : {x, y}
M_6 : IsVisible()	M_7 : GetX()	M_8 : SetX()	M_9 : GetY()	M_{10} : SetY()
I_6 : {Visible}	I_7 : {x}	I_8 : {x}	I_9 : {y}	I_{10} : {y}

M₁ metodu için metot çiftlerinin benzerliği:

$$MS(M_1, M_2) = 3/3 \quad MS(M_1, M_3) = 1/3 \quad MS(M_1, M_4) = 2/3 \quad MS(M_1, M_5) = 2/3$$

$$MS(M_1, M_6) = 1/3 \quad MS(M_1, M_7) = 1/3 \quad MS(M_1, M_8) = 1/3 \quad MS(M_1, M_9) = 1/3$$

$$MS(M_1, M_{10}) = 1/3$$

$$M_1 \text{ metodu için Toplam MS} = [1+1/3+2/3+2/3+1/3+1/3+1/3+1/3+1/3] = 13/3$$

M₂ metodu için metot çiftlerinin benzerliği:

$$MS(M_2, M_3) = 1/3 \quad MS(M_2, M_4) = 2/3 \quad MS(M_2, M_5) = 2/3 \quad MS(M_2, M_6) = 1/3$$

$$MS(M_2, M_7) = 1/3 \quad MS(M_2, M_8) = 1/3 \quad MS(M_2, M_9) = 1/3 \quad MS(M_2, M_{10}) = 1/3$$

$$M_2 \text{ metodu için Toplam MS} = [1/3+2/3+2/3+1/3+1/3+1/3+1/3+1/3] = 10/3$$

M₃ metodu için metot çiftlerinin benzerliği:

$$MS(M_3, M_4) = 0/3 \quad MS(M_3, M_5) = 0/3 \quad MS(M_3, M_6) = 1/3 \quad MS(M_3, M_7) = 0/3$$

$$MS(M_3, M_8) = 0/3 \quad MS(M_3, M_9) = 0/3 \quad MS(M_3, M_{10}) = 0/3$$

$$M_3 \text{ metodu için Toplam MS} = [0+0+1/3+0+0+0+0] = 1/3$$

M₄ metodu için metot çiftlerinin benzerliği:

$$MS(M_4, M_5) = 2/3 \quad MS(M_4, M_6) = 0/3 \quad MS(M_4, M_7) = 1/3 \quad MS(M_4, M_8) = 1/3$$

$$MS(M_4, M_9) = 1/3 \quad MS(M_4, M_{10}) = 1/3$$

$$M_4 \text{ metodu için Toplam MS} = [2/3+0+1/3+1/3+1/3+1/3] = 2$$

M₅ metodu için metot çiftlerinin benzerliği:

$$MS(M_5, M_6) = 0/3 \quad MS(M_5, M_7) = 1/3 \quad MS(M_5, M_8) = 1/3 \quad MS(M_5, M_9) = 1/3$$

$$MS(M_5, M_{10}) = 1/3$$

$$M_5 \text{ metodu için Toplam MS} = [0+1/3+1/3+1/3+1/3] = 4/3$$

M₆ metodu için metot çiftlerinin benzerliği:

$$MS(M_6, M_7) = 0/3 \quad MS(M_6, M_8) = 0/3 \quad MS(M_6, M_9) = 0/3 \quad MS(M_6, M_{10}) = 0/3$$

$$M_6 \text{ metodu için Toplam MS} = [0+0+0+0] = 0$$

M₇ metodu için metot çiftlerinin benzerliği:

$$MS(M_7, M_8) = 1/3 \quad MS(M_7, M_9) = 0/3 \quad MS(M_7, M_{10}) = 0/3$$

$$M_7 \text{ metodu için Toplam MS} = [1/3+0+0] = 1/3$$

M₈ metodu için metot çiftlerinin benzerliği:

$$MS(M_8, M_9) = 0/3 \quad MS(M_8, M_{10}) = 0/3$$

$$M_8 \text{ metodu için Toplam MS} = [0+0] = 0$$

M₉ metodu için metot çiftlerinin benzerliği:

$$MS(M_9, M_{10}) = 1/3$$

$$M_9 \text{ metodu için Toplam MS} = 1/3$$

$$CC(X) = \frac{2 \cdot (10-2)!}{10!} \left(\frac{13}{3} + \frac{10}{3} + \frac{1}{3} + 2 + \frac{4}{3} + 0 + \frac{1}{3} + 0 + \frac{1}{3} \right) = \frac{2 \cdot 8!}{10 \cdot 9 \cdot 8!} \left(\frac{36}{3} \right) = \frac{12}{45} = 0.26$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

$$M_1: \text{Point()} \quad M_2: \text{Show()} \quad M_3: \text{Hide()} \quad M_4: \text{Drag()} \quad M_5: \text{MoveTo()}$$

$$I_1: \{\text{Visible}, x, y\} \quad I_2: \{\text{Visible}, x, y\} \quad I_3: \{\text{Visible}\} \quad I_4: \{x, y\} \quad I_5: \{x, y\}$$

M₁ metodu için metot çiftlerinin benzerliği:

$$MS(M_1, M_2) = 3/3 \quad MS(M_1, M_3) = 1/3 \quad MS(M_1, M_4) = 2/3 \quad MS(M_1, M_5) = 2/3$$

$$M_1 \text{ metodu için Toplam MS} = [1+1/3+2/3+2/3] = 8/3$$

M₂ metodu için metot çiftlerinin benzerliği:

$$MS(M_2, M_3) = 1/3 \quad MS(M_2, M_4) = 2/3 \quad MS(M_2, M_5) = 2/3$$

$$M_2 \text{ metodu için Toplam MS} = [1/3+2/3+2/3] = 5/3$$

M₃ metodu için metot çiftlerinin benzerliği:

$$MS(M_3, M_4) = 0/3 \quad MS(M_3, M_5) = 0/3$$

$$M_3 \text{ metodu için Toplam MS} = 0$$

M₄ metodu için metot çiftlerinin benzerliği:

$$MS(M_4, M_5) = 2/3$$

M_4 metodu için Toplam MS = 2/3

$$CC(X) = \frac{2 \cdot (5-2)!}{5!} \left(\frac{8}{3} + \frac{5}{3} + 0 + \frac{2}{3} \right) = \frac{2 \cdot 3!}{5 \cdot 4 \cdot 3!} (5) = 0.5$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

M_2 : Show()	M_3 : Hide()	M_4 : Drag()	M_5 : MoveTo()	M_6 : IsVisible()
I_2 : {Visible, x, y}	I_3 : {Visible}	I_4 : {x, y}	I_5 : {x, y}	I_6 : {Visible}
M_7 : GetX()	M_8 : SetX()	M_9 : GetY()	M_{10} : SetY()	
I_7 : {x}	I_8 : {x}	I_9 : {y}	I_{10} : {y}	

M_2 metodu için metot çiftlerinin benzerliği:

$$MS(M_2, M_3) = 1/3 \quad MS(M_2, M_4) = 2/3 \quad MS(M_2, M_5) = 2/3 \quad MS(M_2, M_6) = 1/3$$

$$MS(M_2, M_7) = 1/3 \quad MS(M_2, M_8) = 1/3 \quad MS(M_2, M_9) = 1/3 \quad MS(M_2, M_{10}) = 1/3$$

$$M_2 \text{ metodu için Toplam MS} = [1/3 + 2/3 + 2/3 + 1/3 + 1/3 + 1/3 + 1/3 + 1/3] = 10/3$$

M_3 metodu için metot çiftlerinin benzerliği:

$$MS(M_3, M_4) = 0/3 \quad MS(M_3, M_5) = 0/3 \quad MS(M_3, M_6) = 1/3 \quad MS(M_3, M_7) = 0/3$$

$$MS(M_3, M_8) = 0/3 \quad MS(M_3, M_9) = 0/3 \quad MS(M_3, M_{10}) = 0/3$$

$$M_3 \text{ metodu için Toplam MS} = [0 + 0 + 1/3 + 0 + 0 + 0 + 0] = 1/3$$

M_4 metodu için metot çiftlerinin benzerliği:

$$MS(M_4, M_5) = 2/3 \quad MS(M_4, M_6) = 0/3 \quad MS(M_4, M_7) = 1/3 \quad MS(M_4, M_8) = 1/3$$

$$MS(M_4, M_9) = 1/3 \quad MS(M_4, M_{10}) = 1/3$$

$$M_4 \text{ metodu için Toplam MS} = [2/3 + 0 + 1/3 + 1/3 + 1/3 + 1/3] = 2$$

M_5 metodu için metot çiftlerinin benzerliği:

$$MS(M_5, M_6) = 0/3 \quad MS(M_5, M_7) = 1/3 \quad MS(M_5, M_8) = 1/3 \quad MS(M_5, M_9) = 1/3$$

$$MS(M_5, M_{10}) = 1/3$$

$$M_5 \text{ metodu için Toplam MS} = [0 + 1/3 + 1/3 + 1/3 + 1/3] = 4/3$$

M₆ metodu için metot çiftlerinin benzerliği:

$$MS(M_6, M_7) = 0/3 \quad MS(M_6, M_8) = 0/3 \quad MS(M_6, M_9) = 0/3 \quad MS(M_6, M_{10}) = 0/3$$

$$M_6 \text{ metodu için Toplam MS} = [0+0+0+0] = 0$$

M₇ metodu için metot çiftlerinin benzerliği:

$$MS(M_7, M_8) = 1/3 \quad MS(M_7, M_9) = 0/3 \quad MS(M_7, M_{10}) = 0/3$$

$$M_7 \text{ metodu için Toplam MS} = [1/3+0+0] = 1/3$$

M₈ metodu için metot çiftlerinin benzerliği:

$$MS(M_8, M_9) = 0/3 \quad MS(M_8, M_{10}) = 0/3$$

$$M_8 \text{ metodu için Toplam MS} = [0+0] = 0$$

M₉ metodu için metot çiftlerinin benzerliği:

$$MS(M_9, M_{10}) = 1/3$$

$$M_9 \text{ metodu için Toplam MS} = 1/3$$

$$CC(X) = \frac{2 \cdot (9-2)!}{9!} \left(\frac{10}{3} + \frac{1}{3} + 2 + \frac{4}{3} + 0 + \frac{1}{3} + 0 + \frac{1}{3} \right) = \frac{2 \cdot 7!}{9 \cdot 8 \cdot 7!} \left(\frac{23}{3} \right) = \frac{23}{108} = 0.21$$

Örnek 2:

Sınıf A için;

$$M1 = \{A_1\} \quad M2 = \{A_1, A_2\} \quad M3 = \{A_1, A_2\}$$

M1 metodu için metot çiftlerinin benzerliği:

$$MS(M1, M2) = 1/2 \quad MS(M1, M3) = 1/2$$

$$M1 \text{ metodu için Toplam MS} = [1/2+1/2] = 1$$

M2 metodu için metot çiftlerinin benzerliği:

$$MS(M2, M3) = 1/1$$

$$M2 \text{ metodu için Toplam MS} = 1$$

$$CC(X) = \frac{1}{3} \cdot (1+1) = \frac{2}{3} = 0.66$$

Sınıf B için;

$$M1 = \{A_1, A_2\} \quad M2 = \{A_1, A_2\} \quad M3 = \{A_1, A_2\}$$

M1 metodu için metot çiftlerinin benzerliği:

$$MS(M1, M2) = 1 \quad MS(M1, M3) = 1$$

$$M1 \text{ metodu için Toplam MS} = [1+1] = 2$$

M2 metodu için metot çiftlerinin benzerliği:

$$MS(M2, M3) = 1/1$$

$$M_2 \text{ metodu için Toplam MS} = 1$$

$$CC(X) = \frac{1}{3} \cdot (2+1) = 1$$

Değerlendirme:

- $CC(X)$ ölçütü değeri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Bonja ve arkadaşlarının ölçütü ilgili şu önerisi vardır: Sınıf 1 tane metoda sahipse veya hiç metodu yoksa ya da hiç üye alanı yoksa yüksek uyumlu düşünülebilir ve CC değeri olarak 1 atanır.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Weyuker özellik kümesi kullanılarak teorik olarak geçerlemesi yapılmıştır.
- Küçük bir uygulama kullanılarak deneysel geçerlemesi de yapılmıştır. Deneysel geçerlemede LCOM ve CAMC ölçütlerine karşı yeni önerilen $CC(X)$ ölçütünün daha güçlü olduğu sonucuna varılmıştır.
- $CC(X)$ ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan $CC(X)$ ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak düşürdüğü, yapılandırıcı metotların *uyumu* yapay olarak arttırdığı görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan daha iyi çıkmalıdır. $CC(X)$ ölçütünün sezgilerle paralel sonuçlar verdiği görülmektedir.

4.11 Zhou ve Xu Tarafından Önerilen Ölçütler

4.11.1 ICBMC

ICBMC (An Improved Cohesion Measure For Classes) ölçütü Zhou ve Xu tarafından 2002 yılında önerilmiştir (Zhou vd., 2002). CBMC'nin problemlerini adresleyen bir ölçüttür, referans grafiğinin etkileşim kalıbını daha iyi karakterize ettiği görülmüştür. Fakat bir sınıfın üyeleri arasındaki etkileşimlerin kalıplarını kısmen göz önünde bulundurur ve bazı

durumlarda sezgilerle uyuşmayan ölçüm sonuçlarına neden olabilir.

Tanım 4.24: Kesme kümesi (Cut Set)

ICBMC ölçütü referans grafiğini ayrıştırmak için kesme kümesi kullanır. $G(c)$ referans grafiği, $G(c) = (N(c), E(c))$, kullanılmaktadır. Bu grafik şu şekilde tanımlanmaktadır;

- Eğer $\rho(G_c) = 1$ ise, $G(c)$ referans grafiğinin bir Q cut seti aşağıdaki koşulları sağlayan ve kenar kümesinden oluşan kümedir.
- Eğer Q kümesindeki bütün kenarlar çıkarılırsa, G_c grafiği 2 bileşene ayrılabilir.
- Q en küçük kenar kümesidir.
- Eğer $\rho(G_c) > 1$ ise, $Q = \emptyset$

$CS(G(c))$, $G(c)$ referans grafiğinin bütün kesme kümelerinin kümesi olsun.

Tanım 4.25: Yapı ağacı (Structure Tree)

Sınıfın etkileşim modellerinin hiyerarşik yapısını incelemek için oluşturulur. Yapı ağacı, kesme kümeler çıkartılınca sınıfın nasıl bileşenlerine ayrılacağını tasvirler. G referans grafiğinin $T(G)$ yapı ağacı, kök düğümü (root node) G olan $T = (N, E)$ şu şekilde bir ağaçtır;

$$N = \{(G_c^i, Q(G_c^i)) \mid G_c^i \subseteq G_c\} \quad (4.35)$$

$$E = \left\{ \left((G_c^p, Q(G_c^p)), (G_c^s, Q(G_c^s)) \right) \mid G_c^s \in \overline{G_{cQ}^p} \right\} \quad (4.36)$$

$\rho(G_Q) = 2$ (yapı ağacı ikili ağaçtır).

Tanım 4.26: Bağlılık çarpanı (Connectivity Factor)

Bir kesme kümesindeki kenar sayısının bütün olası etkileşim kenar sayısına oranıdır:

$$F_c(G) = \frac{|Q|}{|N_m(G(c))| \times |N_v(G(c))|} \quad (4.37)$$

Tanım 4.27: Yapı ağacı çarpanı (Structure Factor)

Bir sınıfın referansının 2 bileşeninin ortalamasıdır:

$$F_s(G) = \frac{1}{2} \times [ICBMC(G_{cQ}^1) + ICBMC(G_{cQ}^2)] \quad (4.38)$$

Tanım 4.28:

Bir C sınıfı için ICBMC şu şekilde tanımlanır:

$$ICBMC(G_c) = \text{Max} \left\{ \frac{|M_g|}{|N_m(G(c))| \times |N_v(G(c))|} \times \frac{1}{2} \sum_{i=1}^2 ICBMC(G_{M_g^i}) \mid Q \in CS(G_c) \right\} \quad (4.39)$$

$$ICBMC(G_c) = F_c(G) \times F_s(G) \quad (4.40)$$

$$ICBMC(G_c) = \frac{|Q|}{|N_m(G(c))| \times |N_v(G(c))|} \times \frac{1}{2} \times [ICBMC(G_{cQ^1}) + ICBMC(G_{cQ^2})] \quad (4.41)$$

ICBMC Ölçütü Hesaplaması:

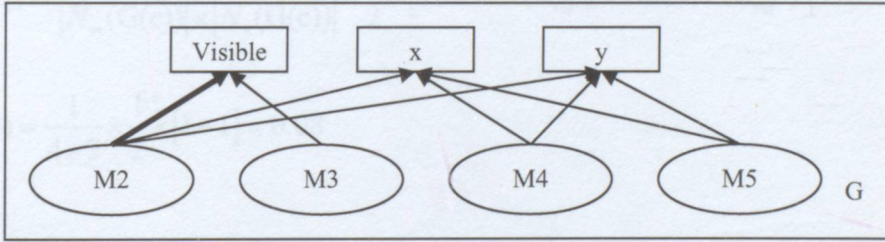
Örnek 1:

CBMC ölçütü, özel metotların göz önüne alınmamasını vurgulamıştı. ICBMC ölçütü de CBMC'nin eksikliğine yönelik önerildiği için bu örnek aşağıdaki koşula göre çözülmektedir.

1) Erişim metotları ve yapılandırıcı metotlar hesaplamaya alınmayacak

M₂: Show() M₃: Hide() M₄: Drag() M₅: MoveTo()

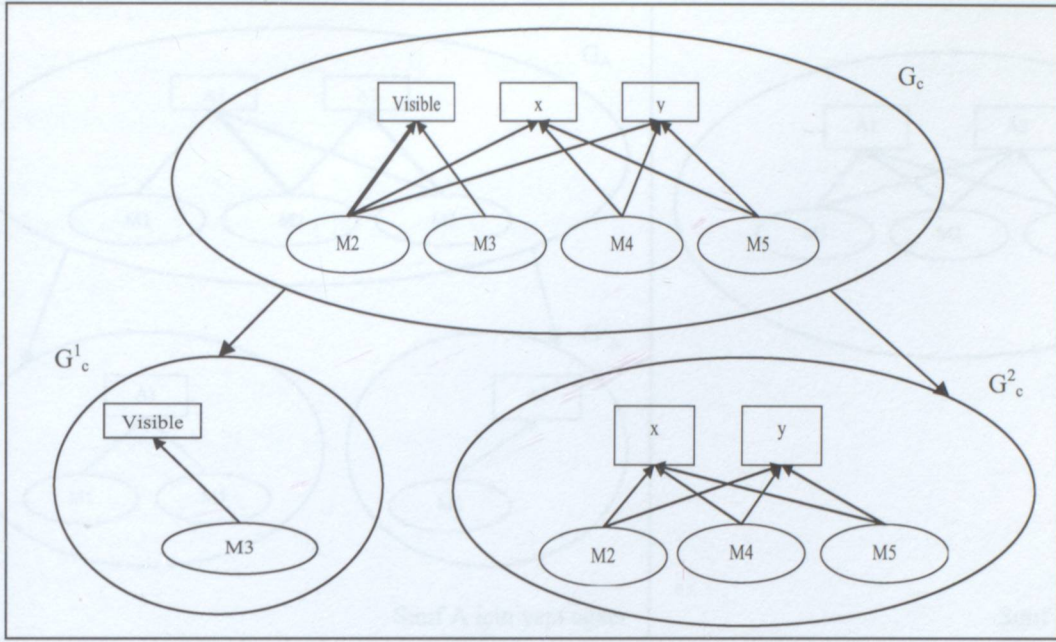
I₂: {Visible, x, y} I₃: {Visible} I₄: {x, y} I₅: {x, y}



Şekil 4.33 ICBMC (1) şikkına göre referans grafiği G

$$CS(G) = \{ \{ \langle M2, \text{Visible} \rangle \} \}$$

Q1 = {< M2, Visible>} kesme kümesine karşılık gelen yapı ağacı Şekil 4.34'de verilmektedir.

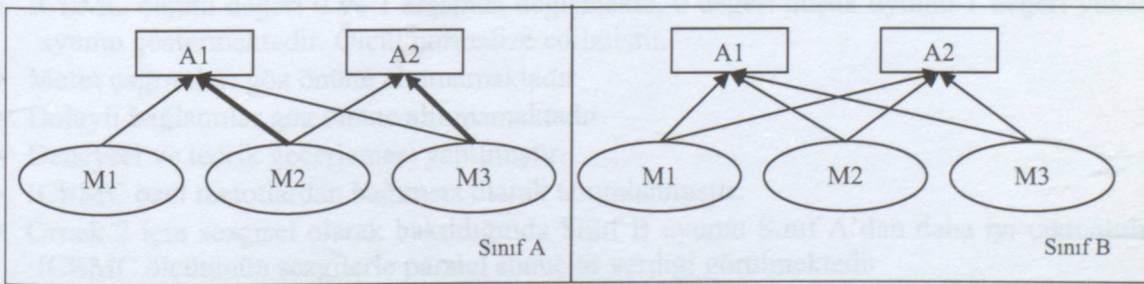


Şekil 4.34 ICBMC (1) şikkına göre yapı ağacı

$$ICBMC(G_c) = \frac{|Q|}{|N_m(G(c))| \times |N_v(G(c))|} \times \frac{1}{2} \times [ICBMC(G_{cQ}^1) + ICBMC(G_{cQ}^2)]$$

$$ICBMC(G) = \frac{1}{4 \times 3} \times \frac{1}{2} \times [1 + 1] = 0.08$$

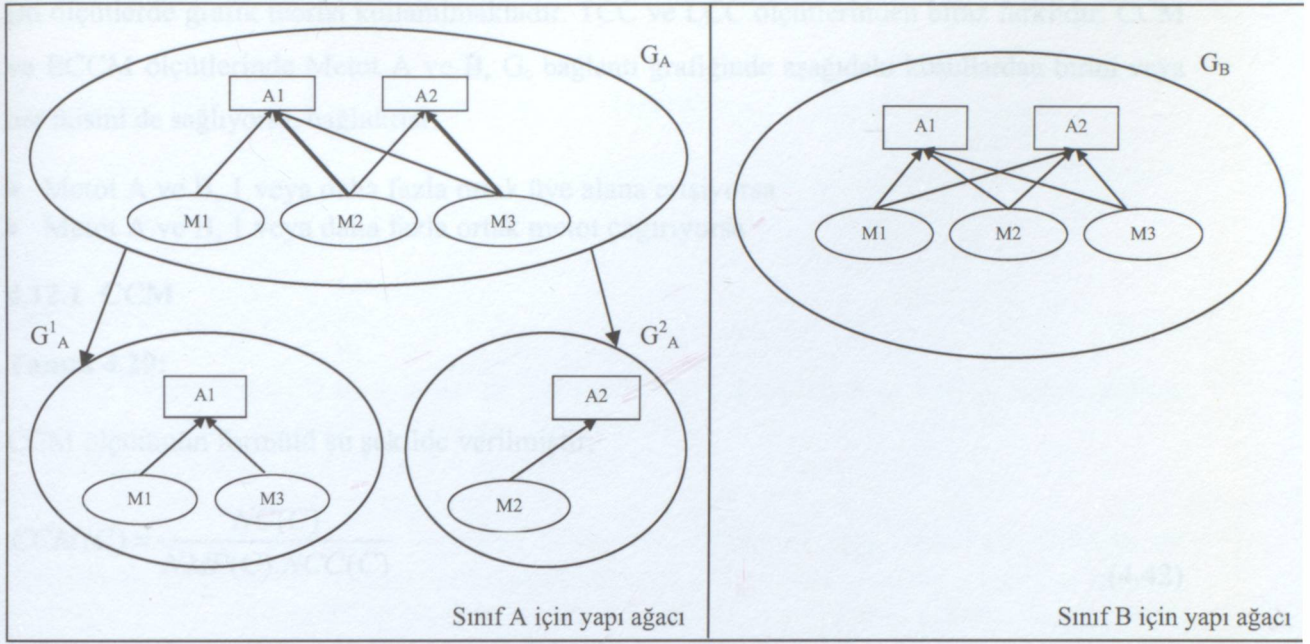
Örnek 2:



Şekil 4.35 ICBMC Sınıf A ve Sınıf B referans grafikleri

$$CS(G_A) = \{ \{ \langle M2, A1 \rangle, \langle M3, A1 \rangle \}, \{ \langle M2, A2 \rangle, \langle M3, A1 \rangle \}, \{ \langle M2, A1 \rangle, \langle M3, A2 \rangle \} \}$$

$Q3 = \{ \langle M2, A1 \rangle, \langle M3, A2 \rangle \}$ kesme kümesini kullanarak çözelim. $Q3$ kesme kümesine karşılık gelen yapı ağacı Şekil 4.36'da verilmektedir.



Şekil 4.36 ICBMC Sınıf A ve Sınıf B için yapı ağaçları

$$ICBMC(G_c) = \frac{|Q|}{|N_m(G(c))| \times |N_v(G(c))|} \times \frac{1}{2} \times [ICBMC(G_{cQ}^1) + ICBMC(G_{cQ}^2)]$$

$$ICBMC(G_A) = \frac{2}{3 \times 2} \times \frac{1}{2} \times [1 + 1] = 0.33, \quad ICBMC(G_B) = 1$$

Değerlendirme:

- ICBMC ölçütü değeri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Deneysel ve teorik geçerlemesi yapılmıştır.
- ICBMC özel metotlardan bağımsız olarak tanımlanmıştır.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan daha iyi çıkmalıdır. ICBMC ölçütünün sezgilerle paralel sonuçlar verdiği görülmektedir.
- Kalıtım konusunda herhangi bir belirtim yapılmamıştır.
- Bazı durumlarda sezgilerle uyuşmayan ölçüm sonuçlarına neden olabildiği ölçüt tanımında belirtilmiş fakat Örnek 2 için sezgilerle benzer sonuçlar verdiği görülmektedir.

4.12 Jarallah ve arkadaşları Tarafından Önerilen Ölçütler

Jarallah et al. kalıtım hiyerarşisinin izlediği bazı 4 tasarım kuralını değerlendirdiği 2001 yılında yaptıkları çalışmada (Jarallah vd., 2001), 2 uyum ölçütü önermişlerdir; CCM (Class Connection Metric) ve ECCM (Enhanced Class Connection Metric)

Bu ölçütlerde grafik teorisi kullanılmaktadır. TCC ve LCC ölçütlerinden biraz farklıdır: CCM ve ECCM ölçütlerinde Metot A ve B, G_c bağlantı grafiğinde aşağıdaki koşullardan birini veya her ikisini de sağlıyorsa, bağlıdır:

- Metot A ve B, 1 veya daha fazla ortak üye alana erişiyorsa
- Metot A ve B, 1 veya daha fazla ortak metot çağırıyor

4.12.1 CCM

Tanım 4.29:

CCM ölçütünün formülü şu şekilde verilmiştir;

$$CCM(C) = \frac{NC(C)}{NMP(C).NCC(C)} \quad (4.42)$$

$NC(C)$: sınıfın metotları arasındaki gerçek bağlantı sayısı

$NMP(C)$: sınıfın metotları arasındaki mümkün olan maximum bağlantı sayısı

$NCC(C)$: Bağlantı grafiği G_c 'nin bağlı bileşenlerinin sayısı

CCM Ölçütü Hesaplaması:

LCOM4 ölçütündeki G_x grafiği bu ölçütte istenen grafikdir. LCOM4'de de üye alan paylaşımı ve metot çağırımı göz önüne alınıyor ve iki metot arasındaki bağlantı buna göre belirleniyordu.

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

$$NC(C) = 31, \quad NCC(C) = 1, \quad NMP(C) = \frac{10 * (10 - 1)}{2} = 45, \quad CCM(C) = \frac{31}{45.1} = 0.68$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

$$NC(C) = 9, \quad NCC(C) = 1, \quad NMP(C) = \frac{5 * (5 - 1)}{2} = 10, \quad CCM(C) = \frac{9}{10.1} = 0.9$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

$$NC(C) = 20, \quad NCC(C) = 1, \quad NMP(C) = \frac{9 * (9 - 1)}{2} = 36, \quad CCM(C) = \frac{9}{36.1} = 0.25$$

Örnek 2:

$$\text{Sınıf A için; } NC(A) = 3, \quad NCC(A) = 1, \quad NMP(A) = 1, \quad CCM(A) = \frac{3}{3.1} = 1$$

$$\text{Sınıf B için; } NC(B) = 3, \quad NCC(B) = 1, \quad NMP(B) = 1, \quad CCM(B) = \frac{3}{3.1} = 1$$

Değerlendirme:

- CCM ölçütü değeri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Teorik olarak geçerlemesi yapılmıştır.
- CCM ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan CCM ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak düşürdüğü, yapılandırıcı metotların *uyumu* yapay olarak arttırdığı görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü Sınıf B'de metotlar daha fazla ortak üye alan kullanmaktadırlar. Ancak, CCM hesaplamasına bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize CCM ölçütünün sezgilerle paralellik göstermediğini, iki tane farklı *uyum* özelliğine sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da CCM ölçütünün uyum özelliğini çok da iyi ayırt edemediğini göstermektedir.

4.12.2 ECCM**Tanım 4.30:**

ECCM ölçütünün formülü şu şekilde verilmektedir:

$$ECCM(C) = \frac{NC(C)}{NMP(C).NCC(C)} \times (1 - \text{PenaltyFactor}(C)), \quad (4.43)$$

basitleştirilmiş olarak,

$$ECCM(C) = CCM(C) \times (1 - \text{PenaltyFactor}(C)) \quad (4.44)$$

$$\text{PenaltyFactor}(C) = \frac{NORM(C)}{NOIM(C)} \quad (4.45)$$

$NORM(C)$: yeniden tanımlanan metot sayısı

$NOIM(C)$: kalıtım alınmış metot sayısı

ECCM Ölçütü Hesaplaması:

Örnek 1 ve Örnek 2'de kalıtım ile ilgili herhangi bir belirtim yapılmadığından her şık için;

ECCM = CCM'dir

Değerlendirme:

- ECCM ölçütü değeri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Teorik olarak geçerlemesi yapılmıştır.
- ECCM ölçütünde hesaplama yaparken Kalıtım alınmış metotların ve yeniden tanımlanmış metotların kullanımından sakınılması gerektiği önerilmektedir.
- Sezgilerle paralellik göstermeyebilmektedir. 2 tane farklı *uyum* özelliğine sahip sınıf aynı veya yakın değer alabilmektedir.

4.13 Aman ve arkadaşları Tarafından Önerilen Ölçütler

Aman ve arkadaşları (Aman vd., 2002) 2001 yılında sadece bir sınıfın bileşenleri arasındaki bağlantıyı değil bağlı bileşenlerin büyüklüğünü ve metot bağlantı güçlülüğünü de göz önüne alarak iki uyum ölçütü önermişlerdir;

- OCC (Optimistic Class Cohesion)
- PCC (Pessimistic Class Cohesion)

Tanım 4.31:

Verilen bir sınıf için M metotların kümesi, A da üye alanların kümesi olsun. Her hangi bir $m \in M$, $a \in A$ için, aşağıdaki önermeler tanımlanmaktadır:

$$ac(m, a) \stackrel{def}{\Leftrightarrow} \exists m' \in M \text{ s.t. } [(m S^* m') \wedge (m' \text{ accesses } a)], \quad (4.46)$$

$$wr(m, a) \stackrel{def}{\Leftrightarrow} \exists m' \in M \text{ s.t. } [(m S^* m') \wedge (m' \text{ writes data on } a)], \quad (4.47)$$

$$re(m, a) \stackrel{def}{\Leftrightarrow} \exists m' \in M \text{ s.t. } [(m S^* m') \wedge (m' \text{ reads data from } a)], \quad (4.48)$$

ac önermesi, sadece üye alana doğrudan erişimi değil *erişim metotları* ile olan dolaylı erişimi de göz önüne almaktadır. wr ve re önermeleri, üye alana erişirken veri-yazma ve veri-okuma'ya odaklanmaktadır.

Tanım 4.32: Zayıf bağlantı grafiği (weak connection graph)

Verilen bir sınıf için M metotların kümesi, A da üye alanların kümesi olsun. Zayıf bağlantı grafiği, yönsüz bir grafik $G_w(V, E)$,

$$V = M, \quad (4.49)$$

$$E = \{\{u, v\} \in M \times M \mid \exists a \in A \text{ s.t. } (ac(u, a) \wedge ac(v, a))\} \quad (4.50)$$

olarak tanımlanmaktadır. Zayıf bağlantı grafiği, üye alan paylaşımını temsil etmektedir. Zayıf bağlantı grafiği, üye alan erişimlerin yönlerinde insensitive özelliktedir, yani bu grafik metot bağlantılarına iyimser (optimistic) bakar.

Tanım 4.33: Güçlü bağlantı grafiği (strong connection graph)

Verilen bir sınıf için M metotların kümesi, A da üye alanların kümesi olsun. Güçlü bağlantı grafiği, yönlü bir grafik $G_w(V, E)$,

$$V = M, \quad (4.51)$$

$$E = \{\{u, v\} \in M \times M \mid \exists a \in A \text{ s.t. } (wr(u, a) \wedge re(v, a))\} \quad (4.52)$$

olarak tanımlanmaktadır. Eğer bir metot bir üye alana atama yapıyorsa ve diğer bir metot da aynı üye alandan veri okuyorsa, bu metotlar arasında bağlı bir ilişki vardır denir. Bu tarz ilişki, Güçlü bağlantı grafiği tarafından belirtilir. Güçlü bağlantı grafiği, üye alan erişimlerin yönlerinde sensitive özelliktedir, yani bu grafik metot bağlantılarına kötümser bakar.

4.13.1 OCC

Verilen bir sınıf C için M metotların kümesi, A da üye alanların kümesi olsun. Zayıf bağlantı grafiği tanımında verildiği gibi Yönsüz bir grafik $G_w(V, E)$, $V=M$ ve E göz önüne alalım, $n=|M|$ olsun.

Her bir metot $m_i \in M$ ($i = 1, \dots, n$) için, $R_w(m_i)$, $G_w(V, E)$ grafiğinde m_i metodu tarafından erişilebilen metotlar kümesi olsun:

$$R_w(m_i) = \left\{ m_j \in M \mid \begin{array}{l} \exists m_{k_1}, \dots, m_{k_p} \in M \\ \text{s.t. } \{m_{k_s}, m_{k_{s+1}}\} \in E \ (s=1, \dots, p-1), m_i=m_{k_1}, m_j=m_{k_p}, i \neq j \end{array} \right\} \quad (4.53)$$

C sınıfı için OCC şu şekilde tanımlanmaktadır;

Tanım 4.34:

$$OCC(C) = \begin{cases} \max_{i=1, \dots, n} \left[\frac{|R_w(m_i)|}{n-1} \right], & (n > 1), \\ 0, & (n = 1). \end{cases} \quad (4.54)$$

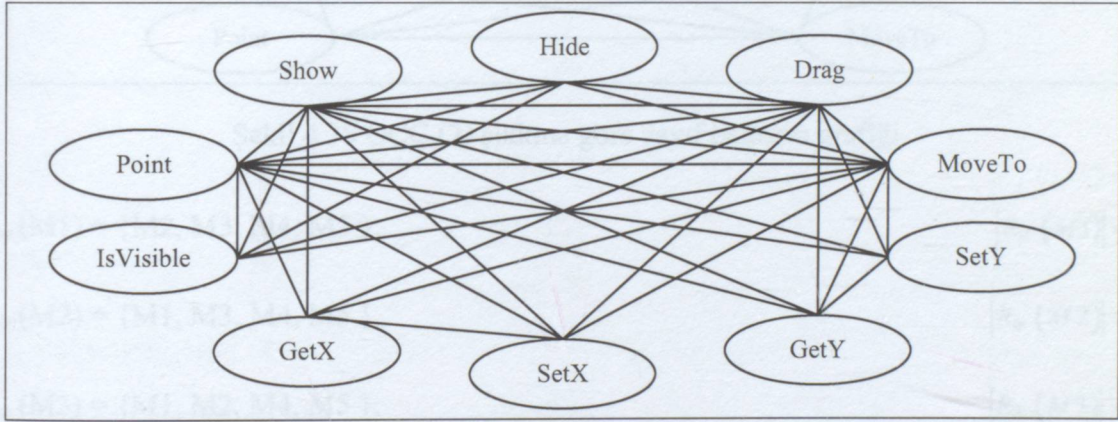
$\frac{|R_w(m_i)|}{n-1}$, Zayıf bağlantı grafiğinde m_i metodu tarafından erişilebilen metotların yüzdesini gösterir (-1 metodun kendisinin çıkarılmış olduğunu gösterir). OCC, o yüzdelerin maximum olanının değeridir.

OCC Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

M ₁ : Point()	M ₂ : Show()	M ₃ : Hide()	M ₄ : Drag()	M ₅ : MoveTo()
I ₁ : {Visible, x, y}	I ₂ : {Visible, x, y}	I ₃ : {Visible}	I ₄ : {x, y}	I ₅ : {x, y}
M ₆ : IsVisible()	M ₇ : GetX()	M ₈ : SetX()	M ₉ : GetY()	M ₁₀ : SetY()
I ₆ : {Visible}	I ₇ : {x}	I ₈ : {x}	I ₉ : {y}	I ₁₀ : {y}



Şekil 4.37 OCC (1) şikkına göre zayıf bağlantı grafiği

$$R_w(M_1) = \{M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}\},$$

$$|R_w(M_1)| = 9$$

$$R_w(M_2) = \{M_1, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}\},$$

$$|R_w(M_2)| = 9$$

$$R_w(M_3) = \{M_1, M_2, M_4, M_5, M_6\},$$

$$|R_w(M_3)| = 5$$

$$R_w(M_4) = \{M_1, M_2, M_3, M_5, M_7, M_8, M_9, M_{10}\},$$

$$|R_w(M_4)| = 8$$

$$R_w(M_5) = \{M_1, M_2, M_3, M_4, M_7, M_8, M_9, M_{10}\},$$

$$|R_w(M_5)| = 8$$

$$R_w(M_6) = \{M_1, M_2, M_3, M_4, M_5\},$$

$$|R_w(M_6)| = 5$$

$$R_w(M_7) = \{M_1, M_2, M_4, M_5, M_8\},$$

$$|R_w(M_7)| = 5$$

$$R_w(M8) = \{M1, M2, M4, M5, M7\},$$

$$|R_w(M8)| = 5$$

$$R_w(M9) = \{M1, M2, M4, M5, M10\},$$

$$|R_w(M9)| = 5$$

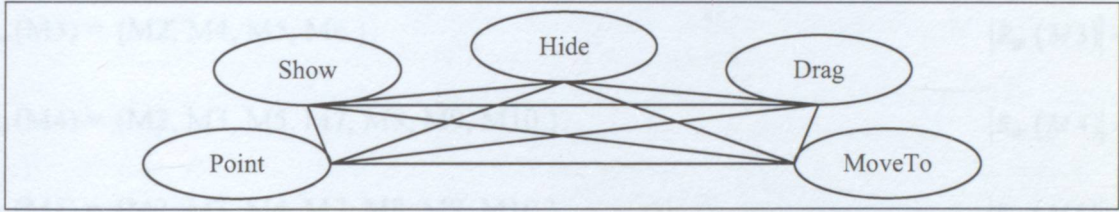
$$R_w(M10) = \{M1, M2, M4, M5, M9\},$$

$$|R_w(M10)| = 5$$

$$OCC = \max [9/(10-1), 5/(10-1), 8/(10-1)] = 1$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

M₁: Point() M₂: Show() M₃: Hide() M₄: Drag() M₅: MoveTo()
 I₁: {Visible, x, y} I₂: {Visible, x, y} I₃: {Visible} I₄: {x, y} I₅: {x, y}



Şekil 4.38 OCC (2) şikkına göre zayıf bağlantı grafiği

$$R_w(M1) = \{M2, M3, M4, M5\},$$

$$|R_w(M1)| = 4$$

$$R_w(M2) = \{M1, M3, M4, M5\},$$

$$|R_w(M2)| = 4$$

$$R_w(M3) = \{M1, M2, M4, M5\},$$

$$|R_w(M3)| = 4$$

$$R_w(M4) = \{M1, M2, M3, M5\},$$

$$|R_w(M4)| = 4$$

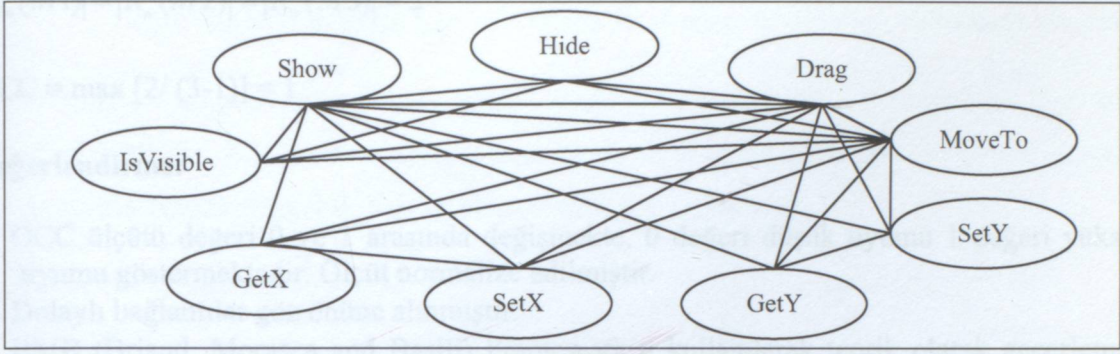
$$R_w(M5) = \{M1, M2, M3, M4\},$$

$$|R_w(M5)| = 4$$

$$OCC = \max [4/(5-1), 2/(5-1), 3/(5-1)] = 1$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

M₂: Show() M₃: Hide() M₄: Drag() M₅: MoveTo() M₆: IsVisible()
 I₂: {Visible, x, y} I₃: {Visible} I₄: {x, y} I₅: {x, y} I₆: {Visible}
 M₇: GetX() M₈: SetX() M₉: GetY() M₁₀: SetY()
 I₇: {x} I₈: {x} I₉: {y} I₁₀: {y}



Şekil 4.39 OCC (3) şikkına göre zayıf bağlantı grafiği

$$R_w(M2) = \{M3, M4, M5, M6, M7, M8, M9, M10\}, \quad |R_w(M2)| = 8$$

$$R_w(M3) = \{M2, M4, M5, M6\}, \quad |R_w(M3)| = 4$$

$$R_w(M4) = \{M2, M3, M5, M7, M8, M9, M10\}, \quad |R_w(M4)| = 7$$

$$R_w(M5) = \{M2, M3, M4, M7, M8, M9, M10\}, \quad |R_w(M5)| = 7$$

$$R_w(M6) = \{M2, M3, M4, M5\}, \quad |R_w(M6)| = 4$$

$$R_w(M7) = \{M2, M4, M5, M8\}, \quad |R_w(M7)| = 4$$

$$R_w(M8) = \{M2, M4, M5, M7\}, \quad |R_w(M8)| = 4$$

$$R_w(M9) = \{M2, M4, M5, M10\}, \quad |R_w(M9)| = 4$$

$$R_w(M10) = \{M2, M4, M5, M9\}, \quad |R_w(M10)| = 4$$

$$OCC = \max [8/(9-1), 4/(9-1), 7/(9-1)] = 1$$

Örnek 2:

Sınıf A ve B için;

$$R_w(M1) = \{M2, M3\}, \quad |R_w(M1)| = 2$$

$$R_w(M2) = \{M1, M3\}, \quad |R_w(M2)| = 2$$

$$R_w(M3) = \{M1, M2\}, \quad |R_w(M3)| = 2$$

$$|R_w(M1)| = |R_w(M2)| = |R_w(M3)| = 2$$

$$OCC = \max [2 / (3-1)] = 1$$

Değerlendirme:

- OCC ölçütü değeri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Dolaylı bağlantılar göz önüne alınmıştır.
- BMB (Briand, Morasca and Basili) Framework'ü kullanılarak teorik olarak geçerlemesi yapılmıştır.
- OCC ölçütünde hesaplama yaparken özel metotlardan erişim metotlarının göz önüne alınması belirtilmektedir.
- OCC ölçütünde hesaplama yaparken kalıtım özelliğinin kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan OCC ölçümü sonucunda erişim metotlarının ve yapılandırıcı metotların kullanımının *uyumu* etkilemediği görülmektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü Sınıf B'de metotlar daha fazla ortak üye alan kullanılmaktadırlar. Ancak, OCC hesaplamasına bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize OCC ölçütünün sezgilerle paralellik göstermediğini, iki tane farklı uyum özelliğine sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da OCC ölçütünün *uyum* özelliğini çok da iyi ayırt edemediğini göstermektedir.

4.13.2 PCC

Verilen bir sınıf C için M metotların kümesi, A da üye alanların kümesi olsun. Güçlü bağlantı grafiği tanımında verildiği gibi bir grafik $G_s(V, E)$, $V=M$ ve E göz önüne alalım, $n=|M|$ olsun.

Her bir metot $m_i \in M$ ($i = 1, \dots, n$) için, $R_s(m_i)$, $G_s(V, E)$ grafiğinde m_i metodu tarafından erişilebilen metotlar kümesi olsun:

$$R_s(m_i) = \left\{ m_j \in M \mid \exists m_{k_1}, \dots, m_{k_p} \in M \right. \\ \left. \text{s.t. } \{m_{k_s}, m_{k_{s+1}}\} \in E \ (s=1, \dots, p-1), m_i=m_{k_1}, m_j=m_{k_p}, i \neq j \right\} \quad (4.55)$$

C sınıfı için PCC şu şekilde tanımlanmaktadır:

Tanım 4.35:

$$PCC(C) = \begin{cases} \max_{i=1, \dots, n} \left[\frac{|R_s(m_i)|}{n-1} \right], & (n > 1), \\ 0, & (n = 1). \end{cases} \quad (4.56)$$

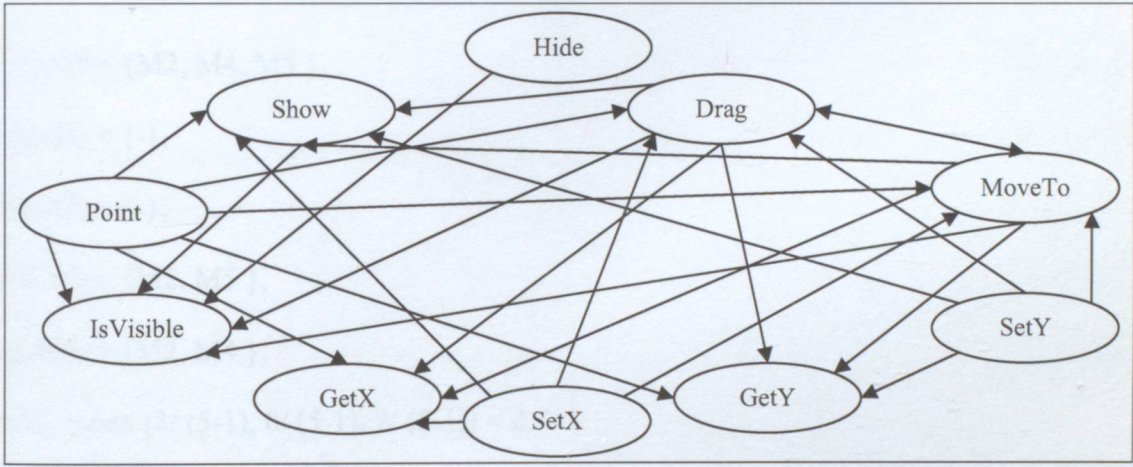
$\frac{|R_s(m_i)|}{n-1}$, Güçlü bağlantı grafiğindeki m_i metodu tarafından erişilebilen metotların yüzdesini

gösterir (-1 metodun kendisinin çıkarılmış olduğunu gösterir). PCC, o yüzdelerin maximum olanının değeridir. PCC, sınıfın metotlarının bağlı ilişkilerinin maximum boyutunu gösterir. Bu da sınıfın yüksek cohesive olan kısmının maximum boyutudur.

PCC Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak



Şekil 4.40 PCC (1) şikkına göre güçlü bağlantı grafiği

$$R_s(M1) = \{M2, M3, M4, M5, M6, M7, M9\},$$

$$|R_s(M1)| = 7$$

$$R_s(M2) = \{M6\},$$

$$|R_s(M2)| = 1$$

$$R_s(M3) = \{M6\},$$

$$|R_s(M3)| = 1$$

$$R_s(M4) = \{M2, M3, M5, M6, M7, M9\},$$

$$|R_s(M4)| = 6$$

$$R_s(M5) = \{M2, M3, M4, M6, M7, M9\},$$

$$|R_s(M5)| = 6$$

$$R_s(M6) = \{-\},$$

$$|R_s(M6)| = 0$$

$$R_s(M7) = \{-\},$$

$$|R_s(M7)| = 0$$

$$R_s(M8) = \{M2, M3, M4, M5, M6, M7, M9\},$$

$$|R_s(M8)| = 7$$

$$R_s(M9) = \{-\},$$

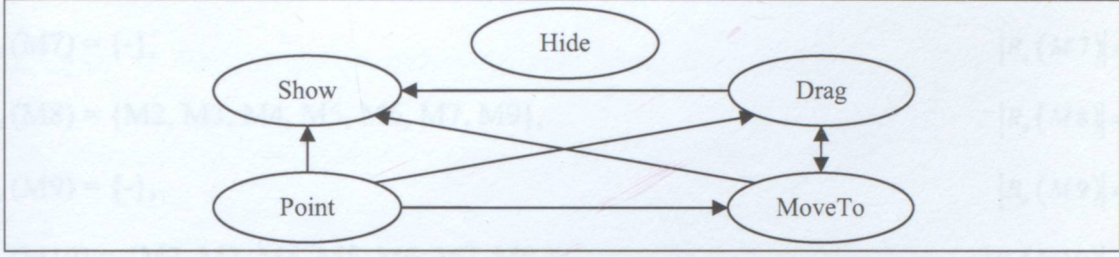
$$|R_s(M9)| = 0$$

$$R_s(M10) = \{M2, M3, M4, M5, M6, M7, M9\},$$

$$|R_s(M10)| = 7$$

$$PCC = \max [7/ (10-1), 1/ (10-1), 6/ (10-1)] = 0.77$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak



Şekil 4.41 PCC (2) şikkına göre güçlü bağlantı grafiği

$$R_s(M1) = \{M2, M4, M5\},$$

$$|R_s(M1)| = 3$$

$$R_s(M2) = \{-\},$$

$$|R_s(M2)| = 0$$

$$R_s(M3) = \{-\},$$

$$|R_s(M3)| = 0$$

$$R_s(M4) = \{M2, M5\},$$

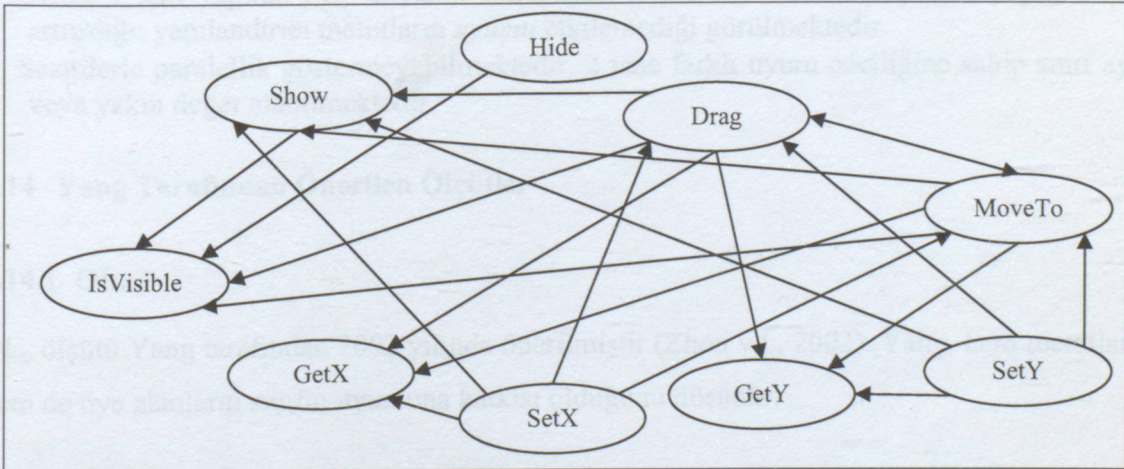
$$|R_s(M4)| = 2$$

$$R_s(M5) = \{M2, M4\},$$

$$|R_s(M5)| = 2$$

$$PCC = \max [3/ (5-1), 0/ (5-1), 2/ (5-1)] = 0.75$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak



Şekil 4.42 PCC (3) şikkına göre güçlü bağlantı grafiği

$$R_s(M2) = \{M6\},$$

$$|R_s(M2)| = 1$$

$$R_s(M3) = \{M6\},$$

$$|R_s(M3)| = 1$$

$R_s(M4) = \{M2, M3, M5, M6, M7, M9\},$	$ R_s(M4) = 6$
$R_s(M5) = \{M2, M3, M4, M6, M7, M9\},$	$ R_s(M5) = 6$
$R_s(M6) = \{-\},$	$ R_s(M6) = 0$
$R_s(M7) = \{-\},$	$ R_s(M7) = 0$
$R_s(M8) = \{M2, M3, M4, M5, M6, M7, M9\},$	$ R_s(M8) = 7$
$R_s(M9) = \{-\},$	$ R_s(M9) = 0$
$R_s(M10) = \{M2, M3, M4, M5, M6, M7, M9\},$	$ R_s(M10) = 7$
$PCC = \max [1/ (10-1), 6/ (10-1), 7/(10-1)] = 0.77$	

Örnek 2:

Bu örnekte metotların üye alanlara ne amaçla eriştiği belirtilmediği için bu ölçüt uygulanamamaktadır.

Değerlendirme:

- PCC ölçütü değeri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Dolaylı bağlantılar göz önüne alınmıştır.
- BMB Framework'ü kullanılarak teorik geçerlemesi yapılmıştır.
- PCC ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 1 için yapılan PCC ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak arttırdığı, yapılandırıcı metotların *uyumu* etkilemediği görülmektedir.
- Sezgilerle paralellik göstermeyebilmektedir. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabilmektedir.

4.14 Yang Tarafından Önerilen Ölçütler

4.14.1 OL_n

OL_n ölçütü Yang tarafından 2002 yılında önerilmiştir (Zhou vd., 2002). Yang, hem metotların hem de üye alanların sınıfın uyumuna katkısı olduğunu düşünür.

Tanım 4.36:

COM(m) ve COV(v) metot m'nin ve üye alan v'nin güçlülüğünü temsil etsin. Verilen bir sınıf c için Yang ilk olarak G_c referans grafiğini oluşturmaktadır. Eğer G_c ayrıksa uyuma 0 atamakta, değilse aşağıdaki adımları kullanarak sınıfın uyumunu hesaplamaktadır:

- 1) İterasyon sayma değişkenine 0 atıyor, ör: $count = 0$;
- 2) Her bir metot m_i için başlangıç güçlülüğü olarak 1 atıyor, ör: $COM(m_i) = 1$;
- 3) Her bir üye alan v_i için $COV(v_i) = \frac{1}{N_m(G_c)} \sum_{j=1}^l COM(m_j)$ olsun.

$N_m(G_c)$, G_c ve $\lambda(m_j, c)$ 'deki metotların sayısını temsil etsin.

- 4) Her bir metot m_i için $COM(m_i) = \frac{1}{N_v(G_c)} \sum_{k=1}^m COV(v_k)$ olsun.

$N_v(G_c)$, G_c grafiğindeki üye alanların sayısını temsil etsin ve $v_k \in \lambda(m_i, c)$

- 5) $count = count + 1$, eğer $count < n$ ise (3). adıma git.
- 6) Sınıfın uyumunu aşağıdaki formülü kullanarak hesapla:

$$OL_n(c) = \frac{1}{N_v(G)} \sum_{r=1}^{N_v(G)} COV(v_r) \quad (4.57)$$

OL_n Ölçütü Hesaplaması:

Örnek 1:

CBMC ölçütünde çizilen referans grafikler kullanılmaktadır.

1) Erişim metotları ve yapılandırıcı metotlar hesaplamaya alınmayacak

- 1) $count = 0$ ve $n = 2$ alınsın
- 2) $COM(m_2) = 1$ $COM(m_3) = 1$ $COM(m_4) = 1$ $COM(m_5) = 1$,
- 3) $COV(Visible) = \frac{1}{4}(1+1) = 0.5$ $COV(x) = \frac{1}{4}(1+1+1) = 0.75$
 $COV(y) = \frac{1}{4}(1+1+1) = 0.75$
- 4) $COM(m_2) = \frac{1}{3}(0.5 + 0.75 + 0.75) = 0.66$ $COM(m_3) = \frac{1}{3}(0.5) = 0.16$
 $COM(m_4) = \frac{1}{3}(0.75 + 0.75) = 0.5$ $COM(m_5) = \frac{1}{3}(0.75 + 0.75) = 0.5$
- 5) $count = 1$, eğer $count < 2$ ise (3). adıma git.

$$5.3) COV(Visible) = \frac{1}{4}(0.66 + 0.16) = 0.82$$

$$COV(x) = \frac{1}{4}(0.66 + 0.5 + 0.5) = 0.41$$

$$COV(y) = \frac{1}{4}(0.66 + 0.5 + 0.5) = 0.41$$

$$5.4) COM(m_2) = \frac{1}{3}(0.82 + 0.41 + 0.41) = 0.54 \quad COM(m_3) = \frac{1}{3}(0.82) = 0.27$$

$$COM(m_4) = \frac{1}{3}(0.41 + 0.41) = 0.27 \quad COM(m_5) = \frac{1}{3}(0.41 + 0.41) = 0.27$$

$$5.5) count = 2$$

$$6) OL_n(c) = \frac{1}{3}(COV(Visible) + COV(x) + COV(y)) = \frac{1}{3}(0.82 + 0.41 + 0.41) = 0.54$$

Örnek 2:

CBMC ölçütünde çizilen referans grafikler kullanılmaktadır.

Sınıf A için OL_n :

$$1) count = 0 \text{ ve } n=2 \text{ alınsın}$$

$$2) COM(m_1)=1 \quad COM(m_2)=1 \quad COM(m_3)=1$$

$$3) COV(a_1) = \frac{1}{3}(1+1+1) = 1 \quad COV(a_2) = \frac{1}{3}(1+1) = 0.66$$

$$4) COM(m_1) = \frac{1}{2} = 0.5 \quad COM(m_2) = \frac{1}{2}(1+1) = 1 \quad COM(m_3) = \frac{1}{2}(1+1) = 1$$

$$5) count = 1, \text{ eğer } count < 2 \text{ ise (3). adıma git.}$$

$$5.3) COV(a_1) = \frac{1}{3}(0.5 + 1 + 1) = 0.83 \quad COV(a_2) = \frac{1}{3}(1 + 1) = 0.66$$

$$5.4) COM(m_1) = \frac{1}{2}(0.83) = 0.42 \quad COM(m_2) = \frac{1}{2}(0.83 + 1) = 0.915 \quad COM(m_3) = \frac{1}{2}(0.83 + 1) = 0.915$$

$$5.5) count = 2$$

$$6) OL_n(c) = \frac{1}{2}(COV(a_1) + COV(a_2)) = \frac{1}{2}(0.83 + 0.66) = 0.745$$

Sınıf B için OL_n :

$$1) count = 0 \text{ ve } n=2 \text{ alınsın}$$

$$2) COM(m_1)=1 \quad COM(m_2)=1 \quad COM(m_3)=1$$

$$3) COV(a_1) = \frac{1}{3}(1+1+1) = 1 \quad COV(a_2) = \frac{1}{3}(1+1) = 0.66$$

$$4) COM(m_1) = \frac{1}{2}(1+1) = 1 \quad COM(m_2) = \frac{1}{2}(1+1) = 1 \quad COM(m_3) = \frac{1}{2}(1+1) = 1$$

5) $count = 1$, eğer $count < 2$ ise (3). adıma git.

$$5.3) COV(a_1) = \frac{1}{3}(1+1+1) = 1$$

$$COV(a_2) = \frac{1}{3}(1+1+1) = 1$$

$$5.4) COM(m_1) = \frac{1}{2}(1+1) = 1$$

$$COM(m_2) = \frac{1}{2}(1+1) = 1$$

$$COM(m_3) = \frac{1}{2}(1+1) = 1$$

$$5.5) count = 2$$

$$6) OL_n(c) = \frac{1}{2}(COV(a_1) + COV(a_2)) = \frac{1}{2}(1+1) = 1$$

Değerlendirme:

- OL_n ölçütü 0 ile 1 arasında değişen değerler almakta, 0 düşük uyumu 1 yüksek uyumu göstermektedir. Ölçüt normalize edilmiştir.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.
- Briand Framework'ü kullanılarak teorik geçerlemesi yapılmıştır. Briand Framework'ünün 4 özelliğini de sağladığı görülmüştür.
- OL_n ölçütünde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan daha iyi çıkmalıdır. OL_n ölçütünün sezgilerle paralel sonuçlar verdiği görülmektedir. Her ne kadar bu örnek için sezgilerle paralel sonuçlar verdiği görülse de bazı araştırmacıların yaptıkları çalışmalar (Zhou vd., 2002) sonucu şu görüş vardır; OL_n ölçütü pratikçilere uygun bir n iterasyon faktörü seçmeleri için rehberlik etmez ve böylece uyum ölçümü pratikte zordur. İkinci olarak kendi kendine çelişen sonuçlar verebilir.

4.15 Badri Tarafından Önerilen Ölçütler

Linda Badri ve Mourad Badri 2004 yılında 4 adet ölçüt önermişlerdir (Badri ve Badri, 2003). Çoğu ölçütün sadece üye alan kullanımını benimsediği ve bu nedenle bu ölçütlerin uyumlu metot çiftlerini yakalamada yetersiz olduğu görüşünden yola çıkarak, metotlar arasındaki etkileşimi (metot çağırımı şeklinde) de göz önüne alarak 4 ölçüt (DC_D , LCC_D , DC_I , LCC_I) önermişlerdir. Temel düşünceleri, hiç ortak üye alan kullanmayan metotların da birbirleriyle ilişkili olabileceği düşüncesidir.

Tanım 4.37: Üye alan kullanımı kriteri

Bir C sınıfı olsun. $A = \{A_1, A_2, \dots, A_n\}$ üye alanlar kümesi ve $M = \{M_1, M_2, \dots, M_n\}$ metotlar kümesi olsun.

UA_{M_i} , M_i metodu tarafından *doğrudan* veya *dolaylı yoldan* kullanılan üye alanların kümesi

olsun. n tane $UA_{M_1}, UA_{M_2}, \dots, UA_{M_n}$ kümesi vardır.

Bir üye alan, eğer M_i metodunun metot tanımında yer alıyorsa, metot M_i tarafından *doğrudan* kullanılıyor demektir.

Bir üye alan, eğer M_i metodu tarafından *doğrudan* veya *dolaylı yoldan* çağrılan M_j metodu tarafından *doğrudan* kullanılıyorsa, metot M_i tarafından *dolaylı yoldan* kullanılıyor demektir.

İki metot M_i ve M_j , eğer $UA_{M_i} \cap UA_{M_j} \neq \emptyset$ ise, *UA ilişkisi* tarafından *doğrudan* ilişkilidir.

Bu ifade, iki metot tarafından en az 1 üye alanın *doğrudan* veya *dolaylı yoldan* paylaşıldığını gösterir.

Tanım 4.38: Metot çağırımı kriteri:

Bir C sınıfı olsun. $M = \{M_1, M_2, \dots, M_n\}$ metotlar kümesi olsun.

IM_{M_i} , C sınıfının, M_i metodu tarafından *doğrudan* veya *dolaylı yoldan* çağrılan bütün metotlarının kümesi olsun. n tane $IM_{M_1}, IM_{M_2}, \dots, IM_{M_n}$ kümesi vardır.

Bir metot M_j , eğer M_i metodunun metot tanımında yer alıyorsa, metot M_i tarafından *doğrudan* kullanılıyor demektir.

Bir metot M_j , eğer M_i metodu tarafından *doğrudan* veya *dolaylı yoldan* çağrılan C sınıfının diğer bir metodu tarafından *doğrudan* çağrılıyorsa, metot M_i tarafından *dolaylı yoldan* kullanılıyor demektir.

İki metot M_i ve M_j , eğer $IM_{M_i} \cap IM_{M_j} \neq \emptyset$ ise, *IM ilişkisi* tarafından *doğrudan* ilişkilidir.

M_i ve M_j , eğer $M_j \in IM_{M_i}$ veya $M_i \in IM_{M_j}$ ilişkisi varsa *doğrudan* ilişkilidir.

4.15.1 DC_D

Tanım 4.39: Metotlar arasındaki *doğrudan* ilişki

İki *public* metot M_i ve M_j , birçok yoldan *doğrudan* bağlı olabilirler;

- En az 1 üye alanı paylaşıyorlarsa (*UA ilişkisi*)
- En az sınıfın bir diğer metodu ile etkileşim yapıyorsa (*IA ilişkisi*)
- Veya her iki durumda varsa

Bir C sınıfı ve $PUM = \{M_1, M_2, \dots, M_n\}$ *public* metotlar kümesi olsun. Köşeleri C sınıfının *public* metotları olan ve eğer karşılıklı metotlar *doğrudan* ilişkili ise aralarında bir kenar olan

yönsüz bir grafik G_D göz önüne alalım. E_D grafikteki kenar sayısı olsun. *Public* metotlarının *doğrudan* ilişkisine dayanan Sınıf C 'nin *uyum* derecesi şu şekilde tanımlanmaktadır:

$$DC_D = \frac{|E_D|}{[n*(n-1)/2]} \in [0, 1]. \quad (4.58)$$

DC_D Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

M_1 : Point(): Çağrılan metotlar: {-}, I_1 : {Visible, x, y}

M_2 : Show(): Çağrılan metotlar: {-}, I_2 : {Visible, x, y}

M_3 : Hide(): Çağrılan metotlar: {-}, I_3 : {Visible}

M_4 : Drag(): Çağrılan metotlar: {Show(), MoveTo()}, I_4 : {x, y}

M_5 : MoveTo(): Çağrılan metotlar: {Show(), Hide()}, I_5 : {x, y}

M_6 : IsVisible(): Çağrılan metotlar: {-}, I_6 : {Visible}

M_7 : GetX(): Çağrılan metotlar: {-}, I_7 : {x}

M_8 : SetX(): Çağrılan metotlar: {-}, I_8 : {x}

M_9 : GetY(): Çağrılan metotlar: {-}, I_9 : {y}

M_{10} : SetY(): Çağrılan metotlar: {-}, I_{10} : {y}

$$UA_{M_1} = \{\text{Visible, x, y}\}$$

$$UA_{M_2} = \{\text{Visible, x, y}\}$$

$$UA_{M_3} = \{\text{Visible}\}$$

$$UA_{M_4} = \{\text{Visible, x, y}\}$$

$$UA_{M_5} = \{\text{Visible, x, y}\}$$

$$UA_{M_6} = \{\text{Visible}\}$$

$$UA_{M_7} = \{x\}$$

$$UA_{M_8} = \{x\}$$

$$UA_{M_9} = \{y\}$$

$$UA_{M_{10}} = \{y\}$$

$$IM_{M_1} = \{\}$$

$$IM_{M_2} = \{\}$$

$$IM_{M_3} = \{\}$$

$$IM_{M_4} = \{\text{Show(), MoveTo(), Hide()}\}$$

$$IM_{M_5} = \{\text{Show(), Hide()}\}$$

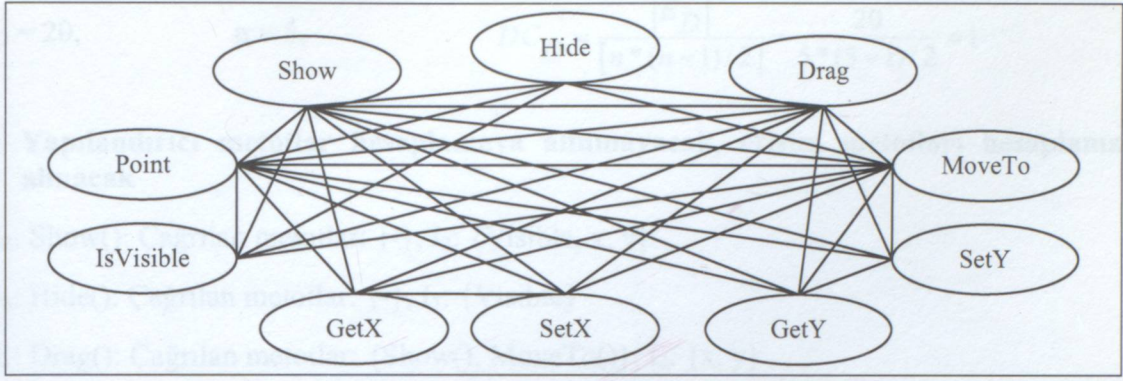
$$IM_{M_6} = \{\}$$

$$IM_{M_7} = \{\}$$

$$IM_{M_8} = \{\}$$

$$IM_{M_9} = \{\}$$

$$IM_{M_{10}} = \{\}$$

Şekil 4.43 DC_D (1) şikkına göre G_D grafiği

$$E_D = 33, \quad n = 10, \quad DC_D = \frac{|E_D|}{[n * (n-1) / 2]} = \frac{33}{10 * (10-1) / 2} = 0.73$$

2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

M₁: Point(): Çağrılan metotlar: {-}, I₁: {Visible, x, y}

M₂: Show(): Çağrılan metotlar: {-}, I₂: {Visible, x, y}

M₃: Hide(): Çağrılan metotlar: {-}, I₃: {Visible}

M₄: Drag(): Çağrılan metotlar: {Show(), MoveTo()}, I₄: {x, y}

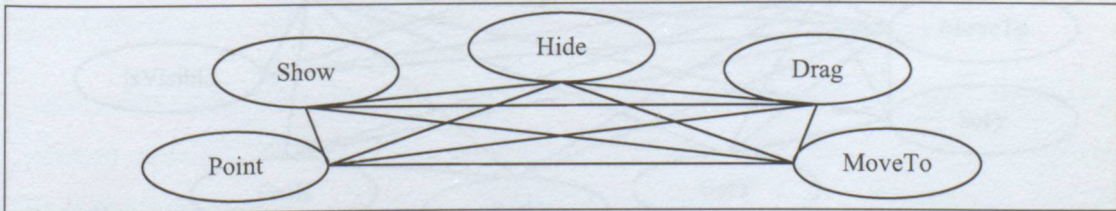
M₅: MoveTo(): Çağrılan metotlar: {Show(), Hide()}, I₅: {x, y}

$$UA_{M_1} = \{Visible, x, y\} \quad UA_{M_2} = \{Visible, x, y\} \quad UA_{M_3} = \{Visible\}$$

$$UA_{M_4} = \{Visible, x, y\} \quad UA_{M_5} = \{Visible, x, y\}$$

$$IM_{M_1} = \{\} \quad IM_{M_2} = \{\} \quad IM_{M_3} = \{\}$$

$$IM_{M_4} = \{Show(), MoveTo(), Hide()\} \quad IM_{M_5} = \{Show(), Hide()\}$$

Şekil 4.44 DC_D (2) şikkına göre G_D grafiği

$$E_D = 20,$$

$$n = 5,$$

$$DC_D = \frac{|E_D|}{[n*(n-1)/2]} = \frac{20}{5*(5-1)/2} = 1$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

M₂: Show(): Çağrılan metotlar: {-}, I₂: {Visible, x, y}

M₃: Hide(): Çağrılan metotlar: {-}, I₃: {Visible}

M₄: Drag(): Çağrılan metotlar: {Show(), MoveTo()}, I₄: {x, y}

M₅: MoveTo(): Çağrılan metotlar: {Show(), Hide()}, I₅: {x, y}

M₆: IsVisible(): Çağrılan metotlar: {-}, I₆: {Visible}

M₇: GetX(): Çağrılan metotlar: {-}, I₇: {x}

M₈: SetX(): Çağrılan metotlar: {-}, I₈: {x}

M₉: GetY(): Çağrılan metotlar: {-}, I₉: {y}

M₁₀: SetY(): Çağrılan metotlar: {-}, I₁₀: {y}

$$UA_{M_2} = \{Visible, x, y\}$$

$$UA_{M_3} = \{Visible\}$$

$$UA_{M_4} = \{Visible, x, y\}$$

$$UA_{M_5} = \{Visible, x, y\}$$

$$UA_{M_6} = \{Visible\}$$

$$UA_{M_7} = \{x\}$$

$$UA_{M_8} = \{x\}$$

$$UA_{M_9} = \{y\}$$

$$UA_{M_{10}} = \{y\}$$

$$IM_{M_2} = \{\}$$

$$IM_{M_3} = \{\}$$

$$IM_{M_4} = \{Show(), MoveTo(), Hide()\}$$

$$IM_{M_5} = \{Show(), Hide()\}$$

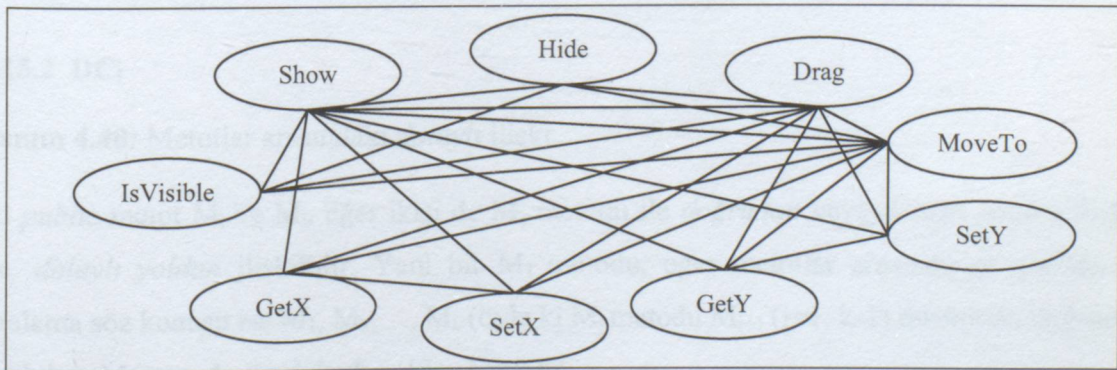
$$IM_{M_6} = \{\}$$

$$IM_{M_7} = \{\}$$

$$IM_{M_8} = \{\}$$

$$IM_{M_9} = \{\}$$

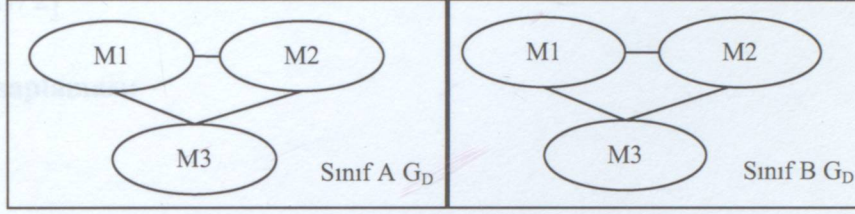
$$IM_{M_{10}} = \{\}$$



Şekil 4.45 DC_D (3) şikkına göre G_D grafiği

$$E_D = 24, \quad n = 9, \quad DC_D = \frac{|E_D|}{[n*(n-1)/2]} = \frac{24}{9*(9-1)/2} = 0.53$$

Örnek 2:



Şekil 4.46 DC_D Sınıf A ya ve Sınıf B ye ait G_D

Sınıf A için;

$$\begin{aligned} UA_{M_1} &= \{A_1, A_2\} & UA_{M_2} &= \{A_1, A_2\} & UA_{M_3} &= \{A_1, A_2\} \\ IM_{M_1} &= \{M_2\} & IM_{M_2} &= \{\} & IM_{M_3} &= \{\} \end{aligned}$$

$$E_D = 3, \quad n = 3, \quad DC_D = \frac{|E_D|}{[n*(n-1)/2]} = \frac{3}{3*(3-1)/2} = 1$$

Sınıf B için;

$$\begin{aligned} UA_{M_1} &= \{A_1, A_2\} & UA_{M_2} &= \{A_1, A_2\} & UA_{M_3} &= \{A_1, A_2\} \\ IM_{M_1} &= \{\} & IM_{M_2} &= \{M_3\} & IM_{M_3} &= \{\} \end{aligned}$$

$$E_D = 3, \quad n = 3, \quad DC_D = \frac{|E_D|}{[n*(n-1)/2]} = \frac{3}{3*(3-1)/2} = 1$$

4.15.2 DC_I

Tanım 4.40: Metotlar arasındaki *dolaylı* ilişki

İki *public* metot M_i ve M_j , eğer ikisi de M_k metodu ile *doğrudan* veya *dolaylı yoldan* ilişkili ise, *dolaylı yoldan* ilişkilidir. Yani bir M_1 metodu, eğer metotlar arasında şu şekilde bir sıralama söz konusu ise $M_1, M_2, \dots, M_k$ (öyle ki M_i metodu M_{i+1} ($i=1, k-1$) metoduna *doğrudan* bağlıdır), M_k metodu ile *dolaylı yoldan* bağlıdır.

Köşeleri C sınıfının *public* metotları olan ve eğer karşılıklı metotlar *doğrudan* veya *dolaylı yoldan* ilişkili ise aralarında bir kenar olan yönsüz bir grafik G_1 göz önüne alalım. E_1

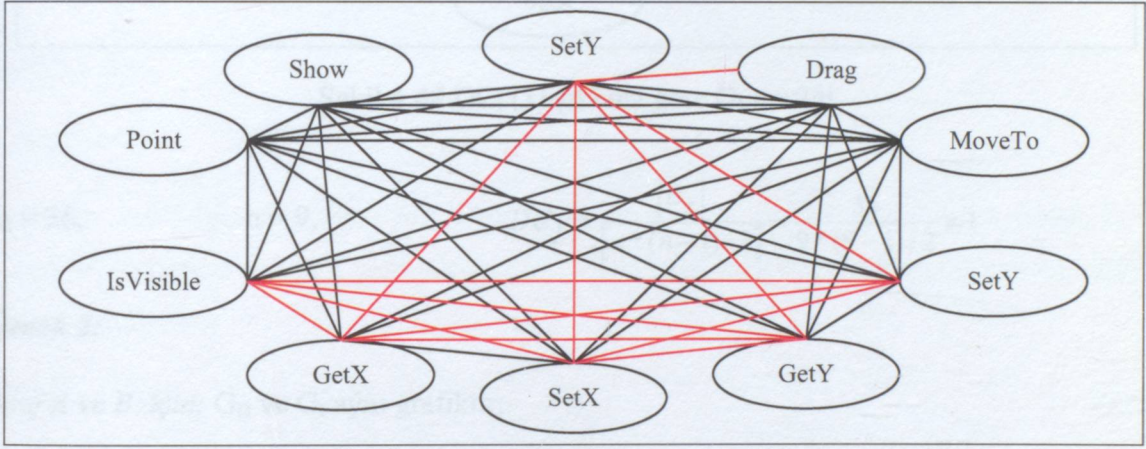
grafikteki kenar sayısı olsun. *Public* metodlarının *doğrudan* veya *dolaylı yoldan* ilişkisine dayanan Sınıf C 'nin *uyum* derecesi şu şekilde tanımlanmaktadır:

$$DC_I = \frac{|E_I|}{[n*(n-1)/2]} \in [0, 1]. \quad (4.59)$$

DC_I Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metodları hesaplamaya alınacak, yapılandırıcı metodlar hesaplamaya alınacak



Şekil 4.47 DC_I (1) şikkına göre G_I grafiği

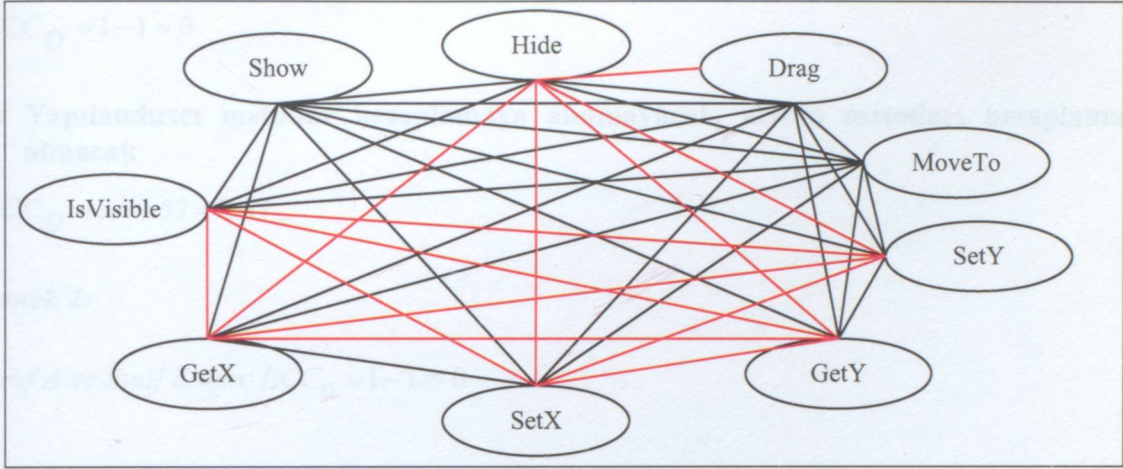
$$E_I = 45, \quad n = 10, \quad DC_I = \frac{|E_I|}{[n*(n-1)/2]} = \frac{45}{10*(10-1)/2} = 1$$

2) Erişim metodları hesaplamaya alınmayacak, yapılandırıcı metodlar hesaplamaya alınacak

Bu şık için G_D ve G_I aynı grafiktir.

$$E_I = 20, \quad n = 5, \quad DC_I = \frac{|E_I|}{[n*(n-1)/2]} = \frac{20}{5*(5-1)/2} = 1$$

3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak



Şekil 4.48 DC_I (3) şikkına göre G_I grafiği

$$E_I = 36, \quad n = 9, \quad DC_I = \frac{|E_I|}{[n*(n-1)/2]} = \frac{36}{9*(9-1)/2} = 1$$

Örnek 2:

Sınıf A ve B için; G_D ve G_I aynı grafiğdir;

$$E_I = 3, \quad n = 3, \quad DC_I = \frac{|E_I|}{[n*(n-1)/2]} = \frac{3}{3*(3-1)/2} = 1$$

4.15.3 LCC_D

Tanım 4.41:

Metotlar arasındaki *doğrudan* ilişkileri hesaba katan LCC_D (Lack of Cohesion in the Class) ölçütü, DC_D ölçütünün tersi olarak düşünülerek şu şekilde tanımlanmaktadır:

$$LCC_D = 1 - DC_D \quad \in [0, 1]. \quad (4.60)$$

LCC_D Ölçütü Hesaplaması:

Örnek 1:

1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

$$LCC_D = 1 - 0.73 = 0.27$$

- 2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

$$LCC_D = 1 - 1 = 0$$

- 3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

$$LCC_D = 1 - 0.53 = 0.47$$

Örnek 2:

Sınıf A ve Sınıf B için; $LCC_D = 1 - 1 = 0$

4.15.4 LCC_I

Tanım 4.42:

Metotlar arasındaki *doğrudan* ilişkilerin yanı sıra *dolaylı* ilişkileri de hesaba katan LCC_I (Lack of Cohesion in the Class) ölçütü, DC_I ölçütünün tersi olarak düşünülerek şu şekilde tanımlanmaktadır:

$$LCC_I = 1 - DC_I \quad \in [0, 1]. \quad (4.61)$$

LCC_I Ölçütü Hesaplaması:

Örnek 1:

- 1) Erişim metotları hesaplamaya alınacak, yapılandırıcı metotlar hesaplamaya alınacak

$$LCC_I = 1 - 1 = 0$$

- 2) Erişim metotları hesaplamaya alınmayacak, yapılandırıcı metotlar hesaplamaya alınacak

$$LCC_I = 1 - 1 = 0$$

- 3) Yapılandırıcı metotlar hesaplamaya alınmayacak, erişim metotları hesaplamaya alınacak

$$LCC_I = 1 - 1 = 0$$

Örnek 2:

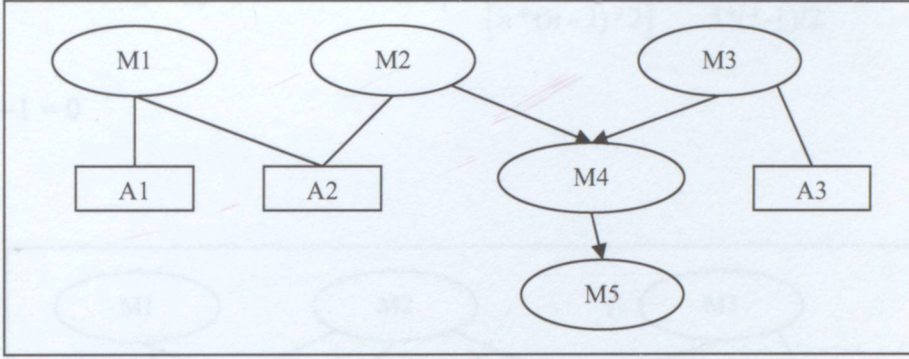
Sınıf A ve B için; $LCC_I = 1 - 1 = 0$

Örnek 1 ve Örnek 2 (DC_D , DC_I , LCC_D , LCC_I) için hesaplanan değerlerinden ölçütleri

anlamak için yeterli olmadıklarını görüyoruz. Bu durum tamamen örneklerin yapısıyla ilgili bir durumdur. Ölçütlerin farkını görebilmek için aşağıda başka bir örnek verilmektedir.

Örnek 3:

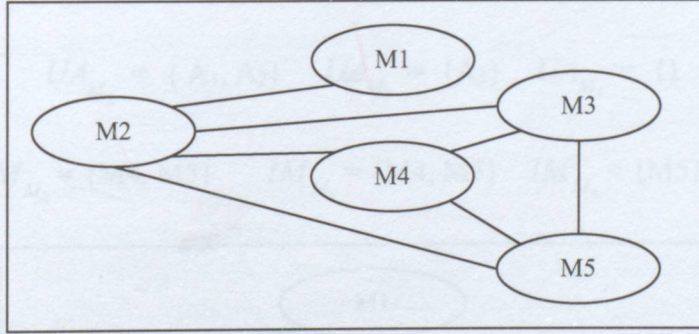
Sınıf A:



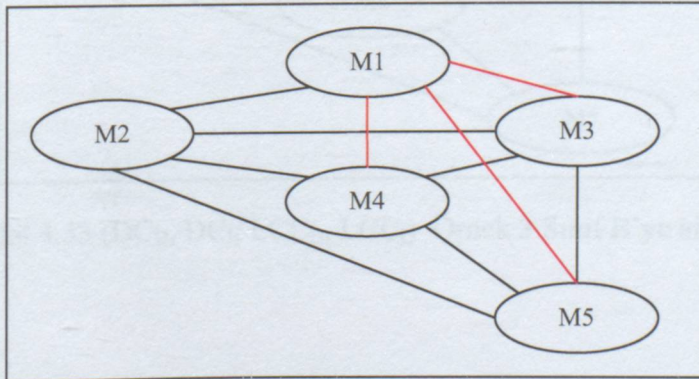
Şekil 4.49 (DC_D , DC_I , LCC_D , LCC_I) Örnek 3 Sınıf A ilişkisi grafiği

$$UA_{M_1} = \{A_1, A_2\} \quad UA_{M_2} = \{A_2\} \quad UA_{M_3} = \{A_3\} \quad UA_{M_4} = \{\} \quad UA_{M_5} = \{\}$$

$$IM_{M_1} = \{\} \quad IM_{M_2} = \{M_4, M_5\} \quad IM_{M_3} = \{M_4, M_5\} \quad IM_{M_4} = \{M_5\} \quad IM_{M_5} = \{\}$$



Şekil 4.50 (DC_D , DC_I , LCC_D , LCC_I) Örnek 3 Sınıf A'ya ait G_D



Şekil 4.51 (DC_D , DC_I , LCC_D , LCC_I) Örnek 3 Sınıf A'ya ait G_I

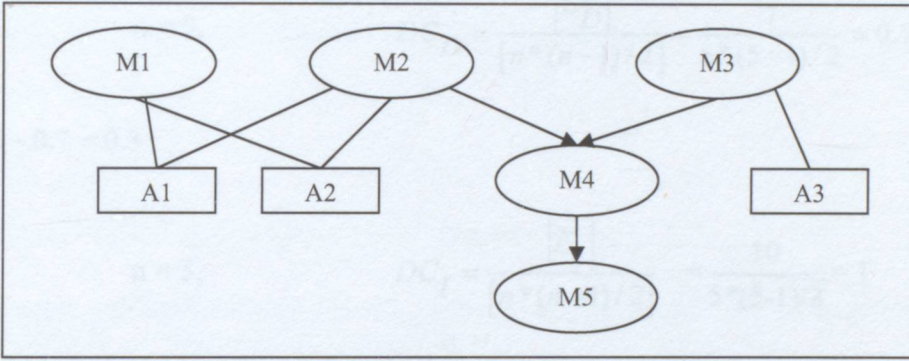
$$E_D = 7, \quad n = 5, \quad DC_D = \frac{|E_D|}{[n*(n-1)/2]} = \frac{7}{5*(5-1)/2} = 0.7$$

$$LCC_D = 1 - 0.7 = 0.3$$

$$E_I = 10, \quad n = 5, \quad DC_I = \frac{|E_I|}{[n*(n-1)/2]} = \frac{10}{5*(5-1)/2} = 1$$

$$LCC_I = 1 - 1 = 0$$

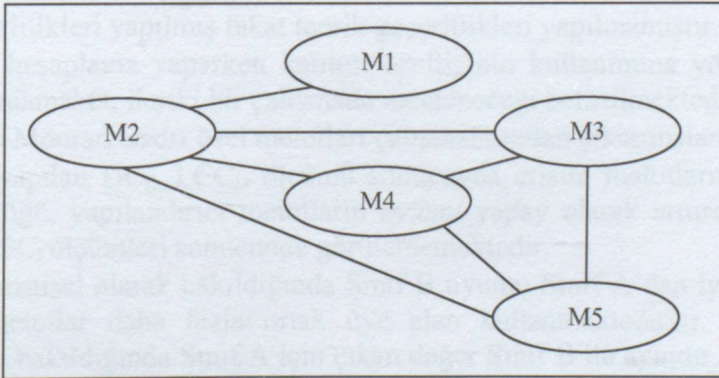
Sınıf B:



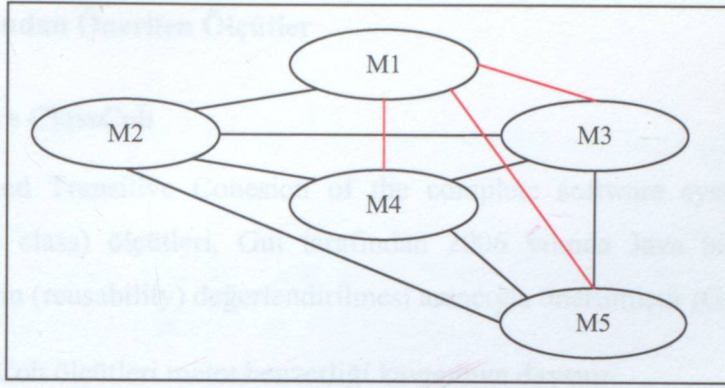
Şekil 4.52 (DC_D , DC_I , LCC_D , LCC_I) Örnek 3 Sınıf B ilişki grafiği

$$UA_{M_1} = \{A_1, A_2\} \quad UA_{M_2} = \{A_1, A_2\} \quad UA_{M_3} = \{A_3\} \quad UA_{M_4} = \{\} \quad UA_{M_5} = \{\}$$

$$IM_{M_1} = \{\} \quad IM_{M_2} = \{M_4, M_5\} \quad IM_{M_3} = \{M_4, M_5\} \quad IM_{M_4} = \{M_5\} \quad IM_{M_5} = \{\}$$



Şekil 4.53 (DC_D , DC_I , LCC_D , LCC_I) Örnek 3 Sınıf B'ye ait G_D



Şekil 4.54 (DC_D , DC_I , LCC_D , LCC_I) Örnek 3 Sınıf B'ye ait G_I

$$E_D = 7, \quad n = 5, \quad DC_D = \frac{|E_D|}{[n*(n-1)/2]} = \frac{7}{5*(5-1)/2} = 0.7$$

$$LCC_D = 1 - 0.7 = 0.3$$

$$E_I = 10, \quad n = 5, \quad DC_I = \frac{|E_I|}{[n*(n-1)/2]} = \frac{10}{5*(5-1)/2} = 1$$

$$LCC_I = 1 - 1 = 0$$

Değerlendirme (DC_D , DC_I , LCC_D , LCC_I):

- Ölçüt değerleri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçütler normalize edilmiştir.
- Metot çağrımları göz önüne alınmaktadır.
- Dolaylı bağlantılar göz önüne alınmaktadır.
- Deneysel geçerlilikleri yapılmış fakat teorik geçerlilikleri yapılmamıştır.
- Bu ölçütlerde hesaplama yaparken kalıtım özelliğinin kullanımına yönelik herhangi bir belirtim yapılmamakta, ileriki bir çalışmada inceleneceği belirtilmektedir.
- Linda Badri ve Mourad Badri özel metotları çalışmalarından çıkarmışlardır.
- Örnek 1 için yapılan DC_D , LCC_D ölçümü sonucunda erişim metotlarının, *uyumu*, yapay olarak düşürdüğü, yapılandırıcı metotların *uyumu* yapay olarak artırdığı görülürken bu durum DC_I , LCC_I ölçümleri sonucunda görülememektedir.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan iyi çıkmalıdır çünkü Sınıf B'de metotlar daha fazla ortak üye alan kullanmaktadırlar. Ancak, ölçütlerin hesaplamasına bakıldığında Sınıf A için çıkan değer Sınıf B ile aynıdır. Bu durum bize bu ölçütlerin sezgilerle paralellik göstermediğini, iki tane farklı uyum özelliğine sahip sınıf için aynı veya yakın değer verebildiğini dolayısıyla da bu ölçütlerin *uyum* özelliğini çok da iyi ayırt edemediğini göstermektedir.

4.16 Gui Tarafından Önerilen Ölçütler

4.16.1 WTCoh ve ClassCoh

WTCoh (Weighted Transitive Cohesion of the complete software system) ve ClassCoh (cohesion of the class) ölçütleri, Gui tarafından 2006 yılında Java bileşenlerinin tekrar kullanılabilirliğinin (reusability) değerlendirilmesi amacıyla önerilmiştir (Gui ve Scott, 2006).

WTCoh ve ClassCoh ölçütleri metot benzerliği kavramına dayanır.

WTCoh ve ClassCoh ölçütleri 2 açıdan diğer çoğu ölçütten farklılık gösterir:

- Metotlar arasında benzerlik derecesinin sayısal bir ölçümünü kullanmaktadırlar
- Transitive'dir (Geçişkendir). Dolaylı benzerliği içermektedirler.

Tanım 4.43:

Metotlar, eriştikleri üye alanlar kümesi çakışırsa, benzerler.

Grafik yaklaşımı kullanılmaktadır. Sınıfın metotları köşeler olarak öngörülmektedir. $M_1, M_2, \dots, M_m$ metoda sahip bir C sınıfı olsun. $\{V_i\} = M_i$ metodu tarafından kullanılan üye alanların kümesi olsun. Ancak ve ancak $V_i \cap V_j$ boş küme değilse M_i 'den M_j 'ye bir kenar vardır. Grafiğin bir kenarı metotların benzerliğini, en az bir ortak üye alan kullandıklarını gösterir. $SimD(i, j)$, M_i ve M_j metotlarının direk benzerliğinin ölçümü olarak şöyle tanımlanmaktadır:

$$SimD(i, j) = \frac{|V_i \cap V_j|}{|V_i \cup V_j|}, \quad i \neq j \quad (4.62)$$

Bu ölçüm dolaylı benzerlikleri de içerecek şekilde düzenlenebilir. İki metot arasındaki yoldan sağlanan benzerlik güçlülüğü o yolda bulunan kenarların $SimD$ değerlerinin çarpımı olarak hesaplanır. $SimT(i, j, \pi)$, M_i ve M_j metotlarının π yolu boyunca dolaylı benzerliğinin ölçümü olarak şöyle tanımlanmaktadır:

$$SimT(i, j, \pi) = \prod_{e_{s,t} \in \pi} SimD(s, t) = \prod_{e_{s,t} \in \pi} \frac{|V_s \cap V_t|}{|V_s \cup V_t|} \quad (4.63)$$

$e_{s,t}$: s ve t köşeleri arasındaki kenarı simgeler.

İki metodun benzerliğini tanımlamak için en büyük $SimT$ değerine sahip yol seçilir, $Sim(i, j)$.

$$\text{Sim}(i, j) = \text{SimT}(i, j, \pi_{\max}) \quad (4.64)$$

$\pi_{\max}(i, j) = \arg \max_{\pi \in \Pi} \text{SimT}(i, j, \pi)$ ve $\prod$, M_i 'den M_j metoduna giden bütün yolların kümesidir.

Sınıfın uyumunu ölçmek için bu ölçüt şöyle kullanılmaktadır; bütün metot çiftlerinin benzerliğinin toplamının, toplam çift sayısına bölümü, ClassCoh.

$$\text{ClassCoh} = \frac{\sum_{i,j=1}^m \text{Sim}(i, j)}{m^2 - m}, \quad m: \text{bir sınıftaki metot sayısı} \quad (4.65)$$

Komple bir yazılım sisteminin ağırlıklandırılmış transitive uyumu WTCoh, sistemin bütün sınıflarının ortalamasıdır ve şöyle bulunur:

$$\text{WTCoh} = \frac{\sum_{j=1}^n \text{ClassCoh}_j}{n}, \quad n: \text{bir sistemdeki sınıf sayısı} \quad (4.66)$$

WTCoh ve ClassCoh Ölçütleri Hesaplaması:

Örnek 1:

Bu örnek için WTCoh ve ClassCoh ölçütleri hesaplaması çok uzun matematiksel işlemler gerektireceği ve uzun bir alan kaplayacağı için uygulanmamaktadır. Örnek 2'de ölçütlerin hesaplama mantığı işlemleri ile birlikte gösterilmektedir.

Örnek 2:

Sınıf A için;

M1:

M2:

M3:

V1: {A1}

V2: {A1, A2}

V3: {A1, A2}

$$\text{SimD}(1, 2) = \frac{1}{2}$$

$$\text{SimD}(1, 3) = \frac{1}{2}$$

$$\text{SimD}(2, 3) = \frac{2}{2} = 1$$

$$\text{SimT}(1, 2, \pi_1) = \text{SimD}(1, 2) = \frac{1}{2}$$

$$\text{SimT}(1, 2, \pi_2) = \text{SimD}(1, 3) \cdot \text{Sim}(3, 2) = \frac{1}{2} \cdot 1 = \frac{1}{2}$$

$$\text{SimT}(1, 2, \pi_1) = \text{SimT}(1, 2, \pi_2)$$

O nedenle $\text{SimT}(1, 2, \pi_{\max})$ değeri yerine ikisinden biri alınabilir.

$$\text{Sim}(1,2) = \text{SimT}(1, 2, \pi_{\max}) = 1/2$$

$$\text{SimT}(1, 3, \pi_1) = \text{SimD}(1,3) = \frac{1}{2} \quad \text{SimT}(1, 3, \pi_2) = \text{SimD}(1,2) \cdot \text{Sim}(2,3) = \frac{1}{2} \cdot 1 = \frac{1}{2}$$

$$\text{SimT}(1, 3, \pi_1) = \text{SimT}(1, 3, \pi_2)$$

O nedenle $\text{SimT}(1, 3, \pi_{\max})$ değeri yerine ikisinden biri alınabilir.

$$\text{Sim}(1,3) = \text{SimT}(1, 3, \pi_{\max}) = 1/2$$

$$\text{SimT}(2, 3, \pi_1) = \text{SimD}(2,3) = 1 \quad \text{SimT}(2, 3, \pi_2) = \text{SimD}(2,1) \cdot \text{Sim}(1,3) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

$$\text{SimT}(2, 3, \pi_1) > \text{SimT}(2, 3, \pi_2)$$

O nedenle $\text{SimT}(1, 3, \pi_{\max})$ değeri yerine $\text{SimT}(2, 3, \pi_1)$ alınmaktadır.

$$\text{Sim}(2,3) = \text{SimT}(2, 3, \pi_1) = 1$$

$$\text{ClassCoh} = \frac{\text{Sim}(1,2), \text{Sim}(1,3), \text{Sim}(2,3)}{3^2 - 3} = \frac{\frac{1}{2} \cdot \frac{1}{2} \cdot 1}{6} = \frac{1}{24} = 0.04$$

Sınıf B için;

M1:

M2:

M3:

V1: {A1, A2}

V2: {A1, A2}

V3: {A1, A2}

$$\text{SimD}(1,2) = \frac{2}{2} = 1 \quad \text{SimD}(1,3) = \frac{2}{2} = 1 \quad \text{SimD}(2,3) = \frac{2}{2} = 1$$

$$\text{SimT}(1, 2, \pi_1) = \text{SimD}(1,2) = 1$$

$$\text{SimT}(1, 2, \pi_2) = \text{SimD}(1,3) \cdot \text{Sim}(3,2) = 1 \cdot 1 = 1$$

$$\text{SimT}(1, 2, \pi_1) = \text{SimT}(1, 2, \pi_2)$$

O nedenle $\text{SimT}(1, 2, \pi_{\max})$ değeri yerine ikisinden biri alınabilir.

$$\text{Sim}(1,2) = \text{SimT}(1, 2, \pi_{\max}) = 1$$

$$SimT(1, 3, \pi_1) = SimD(1, 3) = 1$$

$$SimT(1, 3, \pi_2) = SimD(1, 2) \cdot Sim(2, 3) = 1 \cdot 1 = 1$$

$$SimT(1, 3, \pi_1) = SimT(1, 3, \pi_2)$$

O nedenle $SimT(1, 3, \pi_{\max})$ değeri yerine ikisinden biri alınabilir.

$$Sim(1, 3) = SimT(1, 3, \pi_{\max}) = 1$$

$$SimT(2, 3, \pi_1) = SimD(2, 3) = 1$$

$$SimT(2, 3, \pi_2) = SimD(2, 1) \cdot Sim(1, 3) = 1 \cdot 1 = 1$$

$$SimT(2, 3, \pi_1) = SimT(2, 3, \pi_2)$$

O nedenle $SimT(2, 3, \pi_{\max})$ değeri yerine ikisinden biri alınabilir.

$$Sim(2, 3) = SimT(2, 3, \pi_{\max}) = 1$$

$$ClassCoh = \frac{Sim(1, 2), Sim(1, 3), Sim(2, 3)}{3^2 - 3} = \frac{1 \cdot 1 \cdot 1}{6} = \frac{1}{6} = 0.16$$

Sınıf A ve Sınıf B'yi aynı sistemde bulunuyor şeklinde düşünülürse sistemin uyumu şu şekilde bulunur:

$$WTCoh = \frac{\sum_{j=1}^n ClassCoh_j}{n} = \frac{\frac{1}{24} + \frac{1}{6}}{2} = \frac{5}{48} = 0.10$$

Değerlendirme (ClassCoh ve WTCoh):

- Ölçüt değerleri 0 ve 1 arasında değişmekte, 0 değeri düşük uyumu 1 değeri yüksek uyumu göstermektedir. Ölçütler normalize edilmiştir.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmaktadır.
- Deneysel geçerlilikleri yapılmış fakat teorik geçerlilikleri yapılmamıştır. WTCoh ve ClassCoh ölçütlerinin deneysel olarak geçerlenmesinde kullanılan diğer uyum ölçütlerinden daha iyi sonuçlar verdiği görülmüş ve tekrar kullanılabilirliği daha iyi tahmin edici oldukları belirlenmiştir.
- Bir sistemin uyumu hesaplanırken sınıflar birbirinden bağımsız düşünülmekte ve WTCoh ölçütü doğal olarak sistemin uyumunu düşürmektedir. Bu da beklenen sonuçtur.
- Bu ölçütlerde hesaplama yaparken kalıtım özelliğinin ve özel metotların kullanımına yönelik herhangi bir belirtim yapılmamaktadır.
- Örnek 2 için sezgisel olarak bakıldığında Sınıf B uyumu Sınıf A'dan daha iyi çıkmalıdır. ClassCoh ölçütünün sezgilerle paralel sonuçlar verdiği görülmektedir.

4.17 Ölçütlere Genel Bir Bakış

Çizelge 4.7’de yapısal ölçütlerin Örnek 1 ve Örnek 2 için yapılan hesaplamalarından elde edilen sonuçlar ve bu sonuçlardan yola çıkarak yapılan yorumlar verilmektedir. Yorumlarda şu iki husus değerlendirilmektedir:

- Erişim metotlarının uyumu yapay olarak düşürdüğü, yapılandırıcı metotların arttırdığı gerçeğinden yola çıkarak, ölçütün bu gerçeği yansıtıp yansıtmadığı
- Ölçütün sezgisel olarak beklediğimiz sonuçlara paralel sonuçlar verip vermediği

Etkileşim grafikleri incelenip sezgisel olarak bakıldığında Örnek 2 Sınıf A uyumu Örnek 2 Sınıf B uyumu’ndan daha iyi çıkmalıdır.

Çizelge 4.7’de ifade edilen kısaltmaların anlamları aşağıda verilmektedir:

EM(+): Erişim metotları hesaplamaya katılmıştır

CM(+): Constructor (Yapılandırıcı) metotları hesaplamaya katılmıştır

EM(-): Erişim metotları hesaplamaya katılmamıştır

CM(-): Constructor metotları hesaplamaya katılmamıştır

UZ: Ölçüt uygulanamaz

Y1: EM’nin uyuma etkisi (↑: uyumu arttırıyor, ↓: uyumu düşürüyor, -: etkisi yok)

Y2: CM’nin uyuma etkisi (↑: uyumu arttırıyor, ↓: uyumu düşürüyor, -: etkisi yok)

Y3: Sınıf A ve B arasındaki farkı ayırt etme (+: farkı ayırt edebilmekte, -: farkı ayırt edememekte)

Çizelge 4.7 Uyum ölçütlerinin örneklerde verdiği sonuçlar

No	Ölçüt	Örnek 1			Örnek 2	Örnek 2	Yorum		
		EM(+), CM (+)	EM(-), CM(+)	EM(+), CM(-)	Sınıf A	Sınıf B	Y1	Y2	Y3
1	LCOM1	16	2	16	0	0	↓	-	-
2	LCOM2	0	0	0	0	0	-	-	-
3	LCOM3	1	1	1	1	1	-	-	-
4	LCOM4	1	1	1	1	1	-	-	-
5	Co	0.61	0.83	0.42	1	1	↓	↑	-
6	TCC	0.64	0.8	0.55	1	1	↓	↑	-
7	LCC	1	1	1	1	1	-	-	-
8	LCOM5	0.51	0.33	0.58	0.25	0	↓	↑	+
9	ICH	6	6	6	2	2	-	-	-
10	RCI	-	-	-	0.83	1	...	...	+
11	Coh	0.53	0.73	0.48	0.83	1	↓	↑	+
12	CBMC	EM (-),	-	-	0.66	1	...	...	+
13	CAMC	0.5	0.6	0.44	UZ	UZ	↓	↑	UZ
14	NHD	0.11	0.15	0.08	UZ	UZ	↓	↑	UZ
15	CC(X)	0.26	0.5	0.21	0.66	1	↓	↑	+
16	ICBMC	EM (-),	-	-	0.33	1	...	...	+
17	CCM	0.68	0.9	0.25	1	1	↓	↑	-
18	ECCM	0.68	0.9	0.25	1	1	↓	↑	-
19	OCC	1	1	1	1	1	-	-	-
20	PCC	0.77	0.75	0.77	UZ	UZ	-	-	-
21	OL _n	EM (-),	-	-	0.745	1	...	...	+
22	DC _D	0.73	1	0.53	1	1	↓	↑	-
23	DC _I	1	1	1	1	1	-	-	-
24	LCC _D	0.27	0	0.47	0	0	↓	↑	-
25	LCC _I	0	0	0	0	0	-	-	-
26	ClassCoh	-	-	-	0.04	0.16	...	...	+
27	WTCoh	-	-	-	0.10		...	...	UZ

Çizelge 4.8’de ölçütlerin genel bir değerlendirilmesi verilmektedir. Çizelge 4.8’de ifade edilen kısaltmaların anlamları aşağıda verilmektedir:

Normalizasyon: Normalizasyon yapılma durumu (+: yapılmıştır, -: yapılmamıştır, $\frac{1}{2}$: kısmen yapılmıştır)

T: Teorik geçerleme durumu (+: yapılmıştır, -: yapılmamıştır)

D: Deneysel geçerleme durumu (+: yapılmıştır, -: yapılmamıştır, !: olumsuz anlamda geçerlenmiştir)

Çizelge 4.8 Uyum ölçütlerine genel bir bakış

Ölçüt	Uyum Kriteri	Normalizasyon	Ters/Düz	Geçerleme	
				T	D
LCOM1	Üye alan paylaşımı	-	Ters	+	-
LCOM2	Üye alan paylaşımı	-	Ters	+	+
LCOM3	Üye alan paylaşımı	-	Ters	-	-
LCOM4	Üye alan paylaşımı ve metot çağırımı	-	Ters	-	-
Co	Üye alan paylaşımı ve metot çağırımı	+	Düz	-	-
TCC	Üye alan paylaşımı	+	Düz	-	-
LCC	Üye alan paylaşımı	+	Düz	-	-
LCOM5	Üye alan kullanımı	⊥	Ters	-	-
ICH	Metot çağırımı	-	Düz	+	!
RCI	Tip ve üye alan kullanımı	+	Düz	+	+
Coh	Üye alan kullanımı	+	Düz	-	-
CBMC	Üye alan paylaşımı, metot çağırımı ve metot paterni	+	Düz	-	+
CAMC	Parametre tip kesişimi	+	Düz	-	+
NHD	Parametre tip kesişimi	+	Düz	-	+
CC(X)	Üye alan kullanımı	+	Düz	+	+
ICBMC	Üye alan paylaşımı, metot çağırımı ve metot paterni	+	Düz	-	-
CCM	Üye alan paylaşımı ve metot çağırımı	+	Düz	+	-
ECCM	Attribute paylaşımı ve metot çağırımı	+	Düz	+	-
OCC	Üye alan paylaşımı ve metot çağırımı	+	Düz	+	-
PCC	Üye alan paylaşımı ve metot çağırımı	+	Düz	+	-
OL _n	Üye alan kullanımı	+	Düz	+	-
DC _D	Üye alan kullanımı ve metot çağırımı	+	Düz	-	+
DC _I	Üye alan kullanımı ve metot çağırımı	+	Düz	-	+
LCC _D	Üye alan kullanımı ve metot çağırımı	+	Düz	-	+
LCC _I	Üye alan kullanımı ve metot çağırımı	+	Düz	-	+
ClassCoh	Üye alan paylaşımı	+	Düz	-	+
WTCoh	Üye alan paylaşımı	+	Düz	-	+

5. NESNEYE YÖNELİK PROGRAMLAMADA YAPISAL BENZERLİK/UYUM ÖLÇÜTLERİ İÇİN SINIFLANDIRMA

5.1 Giriş

Sınıflandırma, aynı amaç için öngörülen birçok özelliğin farkını daha net görebilmemiz açısından yararlı bir uygulamadır. Sınıflandırmanın amacı, ölçütleri birbirleri ile karşılaştırabilmek ve değerlendirebilmektir. Bu amaçla bölüm 5.2’de, mevcut yapısal uyum ölçütlerinin incelenmesiyle elde edilen ve tez yazarı tarafından yeni oluşturulan sınıflandırma kriterleri tanımlanmakta, Bölüm 5.3’de mevcut yapısal uyum ölçütleri, tanımlanan sınıflandırma kriterlerine göre sınıflandırılmaktadır.

5.2 Sınıflandırma Kriterleri

Sınıflandırma kriterleri Çizelge 5.1’de verilmektedir. Sınıflandırma kriterleri, Briand ve arkadaşlarının çalışması (Briand vd., 1997) ve bir master tezi çalışmasının (Adam, 2005) modifiye edilmiş şeklidir. Tez yazarı tarafından belirlenen kriterler “Yeni” olarak ifade edilmektedir.

Çizelge 5.1 Sınıflandırma kriterleri

No	Sınıflandırma kriteri	Kaynak
1	Altta yatan yaklaşım	(Adam, 2005)
2	Tanesellik	(Adam, 2005)
3	Mevcudiyet	(Briand vd., 1997)
4	Sağlamlık/Geçerlilik	(Adam, 2005)
5	Duyarlılık	(Adam, 2005)
6	Normalizasyon	(Adam, 2005)
7	Geçerleme	(Briand vd., 1997)
8	Yorumlama	(Adam, 2005)
9	Bağlantı tipi	(Adam, 2005)
10	Özel metotlar	(Briand vd., 1997)
11	Kalıtım	(Briand vd., 1997)
12	Tanesellik esnekliği	Yeni
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Yeni
14	Hesaplama Kolaylığı	Yeni
15	Üye alan/metot varlığı	Yeni
16	Dolaylı bağlantı varlığı	Yeni

Çizelge 5.1’de verilen kriterlerin açıklamaları şu şekildedir:

Altta yatan yaklaşım

Bir kavramı daha iyi anlamak için, çalışmanın dayandığı altta yatan kuralı anlamak önemlidir. Altta yatan kural, orijinal fikrin nereden alındığını bilmekle elde edilebilir.

Tanesellik

Ölçütün taneciklik seviyesi demektir. Ölçütün sistemin hangi parçasında ölçüm yaptığını gösterir; NYP için bu parçalar metot, sınıf ve pakettir.

Mevcudiyet

Ölçütün yazılım mühendisliği geliştirme sürecinin hangi safhalarında kullanılıp hangilerinde kullanılmadığını gösterir. Bazı ölçütler sadece kodlama tamamlandıktan sonra kullanılabilirler ve bu ölçütler gerçekleştirme safhasında yer alırlar. Bazı ölçütler ise tasarım safhasının sonunda kullanılabilirler.

Sağlamlık/Geçerlilik

Ölçütün bir yazılım sisteminin modülünün benzerlik/uyumunu ne kadar yakalayabildiği, ölçüt hesaplamasında bir anlam belirsizliği olup olmadığı gibi sınamalar ile ölçütün doğruluğuna karar verir. Muğlâklık ise, ölçütün sezgisel olarak farklı uyum değerine sahip sınıflar için aynı değeri vermesi gibi durumlarda oluşur.

Duyarlılık

Ölçüt değişikliklere ne kadar duyarlı olduğunu gösterir. “Bir modüldeki/sınıftaki değişiklik, ölçümü nasıl etkiler?”, “Değişiklik, ölçüt sonucu üzerinde olumlu mu olumsuz mu etkiye sahiptir?” gibi sorulara yanıt aranır.

Normalizasyon

Normalizasyon, ölçütün sonucunun normalize edilip edilmediğini değerlendirir; [0, 1] aralığında değerler alan ölçüt, normalizedir.

Geçerleme

“Ölçüt geçerlenmiş mi geçerlenmemiş mi?”, “Eğer geçerlendiyse teorik mi yoksa deneysel mi?”, “Eğer geçerlenmediyse geçerleme süreci nasıl karmaşıktır; geçerleme yapmak için neye ihtiyaç vardır ve ölçütün geçerlenebilmesi için araştırmacılar bir yol göstermişler mi?” gibi

Çizelge 5.2 Bağlantı tipleri

No	Eleman 1	Eleman 2	Tanım	İsim	Ölçütler	Kaynak
1	c sınıfının m metodu	c sınıfının a üye alanı	m, üye alan a'yı referans eder	MAR	LCOM5, Coh, ICBMC, CBMC, OLn	(Briand vd., 1997)
2	c sınıfının m metodu	c sınıfının m' metodu	m, m' metodunu doğrudan çağırır	DMMR	LCOM4, Co, CCM, ECCM, OCC, PCC, ICH, DC _D , DC _I , LCC _D , LCC _I , COMIAS	(Briand vd., 1997)
3	c sınıfının m metodu	c sınıfının m' metodu	m, m' metodunu birbirini doğrudan çağırarak metotlar yoluyla dolaylı yoldan çağırır	IMMR	DC _D , DC _I , LCC _D , LCC _I , COMIAS	(Adam, 2005)
4	c sınıfının m metodu	c sınıfının m' metodu, m ≠ m'	m ve m' c sınıfının bir üye alanı a'yı doğrudan referans eder	DAS	LCOM1, LCOM2, LCOM3, LCOM4, Co, TCC, CCM, ECCM, OCC, PCC, LCC, CC(X), OLn, DC _D , DC _I , LCC _D , LCC _I , WTCoh, ClassCoh, COMIAS	(Briand vd., 1997)
5	c sınıfının m metodu	c sınıfının m' metodu, m ≠ m'	m ve m' c sınıfının bir üye alanı a'yı dolaylı yoldan referans eder	IAS	LCC, DC _D , DC _I , LCC _D , LCC _I , WTCoh, ClassCoh, COMIAS	(Briand vd., 1997)
6	c sınıfının m metodu	c sınıfının m' metodu	m metodu c sınıfının üye alanı a'ya yazar ve m' metodu okur	MIBAT	PCC	(Adam, 2005)
7	c sınıfında veri tanımlaması	c sınıfında veri tanımlaması	Veri-Veri etkileşimi	DDI	RCI	(Briand vd., 1997)
8	c sınıfının m metodu	c sınıfında veri tanımlaması	Veri-Metot etkileşimi	DMI	RCI	(Briand vd., 1997)
9	c sınıfının m metodu	c sınıfının en az 1 metodunda bulunan parametre p	m metodunda p parametresinin bulunup bulunmaması	MPI	CAMC	Yeni
10	c sınıfının m metodu	c sınıfının m' metodu	m metodu tarafından çağrılan m' metodunun parametre sayısı	NPI	ICH	Yeni
11	c sınıfının m metodu	c sınıfının m' metodu, m ≠ m'	m ve m' c sınıfında en az bir metotta bulunan bir parametre p'yi doğrudan içerir	DMMPP	NHD	Yeni

MAR: metot – üye alan referanslaması

DMMR: Doğrudan metot-metot referanslaması

IMMR: Dolaylı yoldan metot-metot referanslaması

DAS: Doğrudan üye alan paylaşımı

IAS: Dolaylı yoldan üye alan paylaşımı

DDI: Veri-Veri etkileşimi

DMI: Veri-Metot etkileşimi

MPI: Metodun parametre içermesi

NPI: Metodun parametre sayısı

MIBAT: Erişim tiplerine dayanan metot etkileşimleri

DMMPP: Doğrudan metot – metot parametre paylaşımı

Özel metotlar

Bir sınıftaki bir metot eğer metot erişim metodu, delegasyon metodu, yapılandırıcı (constructor) ya da sonlandırıcı (destructor) metot ise *özel*dir (Chae ve Kwon, 1998). Bunun dışında kalan metotlar *normal* metotlardır. Bu metot tanımlamaları Bölüm 4’de verilmektedir. Bu kritere göre, ölçüt tanımında özel metotların etkisinin göz önüne alınıp alınmadığı konusu değerlendirilmektedir.

Kalıtım

Ölçüt önerilirken kalıtım özelliğinin etkisinin göz önüne alınıp alınmadığı konusu değerlendirilmektedir. Kalıtım ile alınmış metot ve üye alanların uyuma etkisi vardır.

Tanesellik Esnekliği

Ölçütün metot seviyesinden sınıf seviyesine ya da sınıf seviyesinden paket seviyesine yükseltip indirgenebilmesini sorgular. Diğer bir deyişle, ölçütün tanımlandığı tanesellik seviyesi dışında diğer sistem bileşenleri içinde de kullanılabiliyor kullanılamamasına bakılır.

Kalite faktörüne etkisi (ispat edilen ya da olması düşünülen)

Bazı ölçüt tanımlamalarında ölçütün hata eğilimliliği, üretkenlik çabası, bakım amaçlı değişiklik miktarı gibi kalite faktörlerini belirlemede kullanılabileceği belirtilmiştir. Bu kriterler, ölçütün orijinal tanımında, ‘Ölçütün herhangi bir kalite faktörüne etkisi ispatlanmış mı ya da etkisi olduğu düşünülmüş mü?’ sorusunun cevabını aramaktadır.

Hesaplama Kolaylığı

Ölçüt hesaplaması formülde veri toplama ve veriyi kullanmayı içerir. Hesaplama kolaylığı veri toplama ve verileri formülde kullanma kolaylığını sorgular. Formül basit ya da karmaşık matematiksel ifadeler içerebilir. Hesaplama kolaylığı, ölçütün formülündeki karmaşık matematiksel ifadeyi sorgular. Veri toplama kolaylığı, formüle konacak veri *direk bakılarak görülebiliyor ve formüle konulabiliyorsa* basit, *grafik çizerek elde ediliyorsa* zor diye nitelendirilebilir. Formülde kullanma kolaylığı, formül sadece *dört işlem* içeriyorsa basit, *dört işlem harici matematiksel ifadeler* içeriyorsa zor diye nitelendirilebilir.

Üye alan metot varlığı

Bir sınıf üye alanlar ve üye metotlardan oluşur. Bu iki önemli elemandan birisinin ölçüt tanımında göz ardı edilmesinin, gerçek uyumun yakalanamamasına neden olacağı düşünülmektedir. Bir ölçütte üye alan, kullanım ve paylaşım yönünden, metot ise çağrım yönünden göz önüne alınmalıdır. Metot çağrımı ile kastedilen; metotların birbirini çağırma anlamındaki metot ilişkisinin göz önüne alınıp alınmadığıdır. Bu kriter, ölçütün bu varlıkları bu açıdan göz önüne alıp almadığını sorgular.

Dolaylı bağlantı varlığı

Çoğu ölçüt tanımında dolaylı ilişkiler göz önüne alınmamaktadır. Dolaylı bağlantılar göz ardı edildiğinde gerçek uyumun yakalanamayacağı ve elde edilen ölçüt sonucunun gerçek uyumu yansıtmayacağı düşünülmektedir. Dolaylı bağlantılar, dolaylı metot çağrımları ve metot çağrımları yoluyla olan dolaylı üye alan kullanımı/paylaşımı şeklinde düşünülebilir. Bu kriter, ölçütün dolaylı bağlantıları göz önüne alıp almadığını sorgular.

5.3 Yapısal Benzerlik/Uyum Ölçütlerinin Sınıflandırılması

Bölüm 5.2’de verilen sınıflandırma kriterlerine göre Bölüm 4’de incelenen ölçütlerin sınıflandırması çizelgeler halinde Ek 1’de verilmektedir. Sınıflandırma sonucu elde edilen gözlemler şunlardır:

- Çoğu ölçüt Bunge’ın “benzerlik” ontolojisine dayanan Chidamber ve Kemerer’in ölçüt çalışmalarından esinlenerek önerilmiştir.
- Çoğu ölçüt sınıf seviyesinde uyumu ölçmek amacıyla önerilmiştir.
- Çoğu ölçüt tasarım safhasında uyumu değerlendirmek üzere önerilmiştir.
- Bazı ölçütler normalize edilmemiştir.
- Bazı ölçütlerin teorik ve/veya deneysel geçerlemesi yapılmamaktadır.
- Çoğu ölçütün nasıl yorumlanması gerektiği ölçüt tanımlamalarında verilmemiştir.
- Çoğu ölçüt çalışmalarında kalıtım özelliğini ve özel metotların kullanımını göz önüne almamıştır.
- Çoğu ölçüt önerilirken herhangi bir kalite faktörüne etkisi yönünden düşünülerek önerilmemiştir.
- Kimi üye alan kullanımını, kimi üye alan paylaşımını, kimi üye metotların çağrımını, kimi parametre tip kesişimini, kimi parametre tip kullanımını kimi de bunlardan bazılarının bileşimini esas almıştır.
- Çoğu ölçüt dolaylı bağlantıları hesaba katmamıştır.

Bu tez çalışmasında yukarıdaki gözlemlerden yola çıkarak temelde LCOM ölçütlerinin eksikliklerini adresleyen yeni bir ölçüt önerilmektedir. Yeni ölçütün tanımı ve yapılan çalışmalar bir sonraki bölümde verilmektedir.

6. YENİ ÖLÇÜT COMIAS

COMIAS (Cohesion Method Invocation and Attribute Sharing) uyum ölçütü gerçekleştirken aşağıdaki hususlar dikkate alınmaktadır:

- Bir sınıf üye alanlar ve üye metotlardan oluşur. Bu iki önemli elemandan birisinin ölçüt tanımında göz ardı edilmesinin, gerçek uyumun yakalanamamasına neden olacağı düşünülmektedir.
- Bir ölçütün normalize edilmiş olması, bir başka deyişle 0 ile 1 arasında değişen bir değer üretmesi tercih edilir. Normalize edilmiş ölçütlerin yorumlanması daha kolay olacaktır. Buna rağmen literatürde normalize olmayan birçok ölçüt bulunmaktadır.
- Çoğu ölçüt tanımında dolaylı metot ilişkileri göz önüne alınmamaktadır. Dolaylı bağlantılar göz ardı edildiğinde gerçek uyumun yakalanamayacağı düşünülmektedir.
- İyi tanımlanmış bir ölçüt mutlaka sezgilerle paralel sonuçlar vermeli, matematiksel olarak belirsiz ve tanımsız sonuçlar vermemelidir.

Bir sınıfın uyumu COMIAS ölçütü kullanılarak hesaplanırken üç temel adım yürütülmektedir. Birinci adımda üye metotların arasındaki ilişki metot çağırımı yönünden incelenmekte ve ilk grafik (G_{xA}) elde edilmektedir. İkinci adımda ise üye metotlar, üye alanların kullanımı yönünden incelenmekte ve ikinci grafik (G_{xB}) elde edilmektedir. Üçüncü adımda ise ilk iki adımda elde edilen grafikler tek bir grafikte (G_x) birleştirilmektedir. Bu bölümün geri kalanında söz konusu üç adım ayrıntılı olarak incelenecektir.

COMIAS ölçütünün tanımlanması, “4. Ulusal Yazılım Mühendisliği Sempozyumu”nda sunulmak üzere kabul alan bildiride (Ertemel vd., 2009a) ve “13. World Scientific and Engineering Academy and Society International Conference on Computers” konferansında sunulmak üzere kabul alan diğer bir bildiride (Ertemel vd., 2009b) ele alınmıştır.

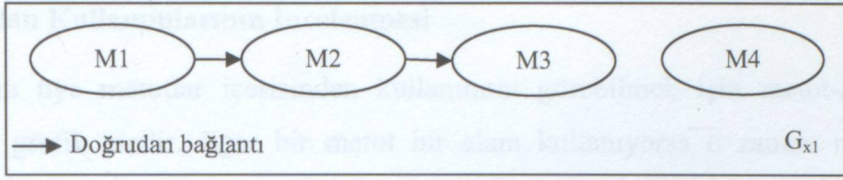
6.1 Metot Çağrılarının İncelenmesi

Metot çağrımları incelenirken, ilk olarak metotlar arasındaki doğrudan bağlantı ilişkileri incelenir. Eğer bir metot diğerini çağırıyorsa o zaman aralarında çağırıcı taraftan çağrılan tarafa doğru yönlü bir çizgi oluşturulur. Oluşturulan grafik G_{x1} aşağıdaki şekilde ifade edilebilir:

Yönlü bir grafik $G_{x1}(V, E)$ ve $V = M_x$ ise

$$E = \{ \langle m, n \rangle \in V \times V \mid (m \text{ calls } n) \} \quad (6.1)$$

Örnek bir G_{x1} grafiği Şekil 6.1’de görülebilir.

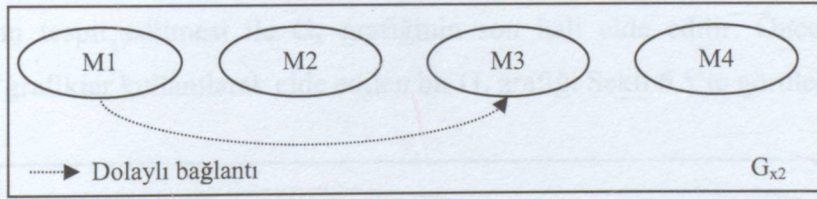
Şekil 6.1 Örnek bir doğrudan bağlantı grafiği G_{x1}

Dolaylı bağlantıları bulabilmek için ise G_{x1} grafiğindeki yönlü çizgilerden yararlanılır. Şekil 6.1 incelendiğinde M1 metodunun M2 metodunu çağırdığı, M2 metodunun da M3 metodunu çağırdığı görülebilir. Dolayısıyla, M1 metodu dolaylı yoldan M3 metodunu çağırılmaktadır. M1 metodundan M3 metoda doğru yönlü bir çizgi oluşturulur. Bu şekilde oluşturulan dolaylı bağlantı grafiği olan G_{x2} aşağıdaki gibi ifade edilebilir:

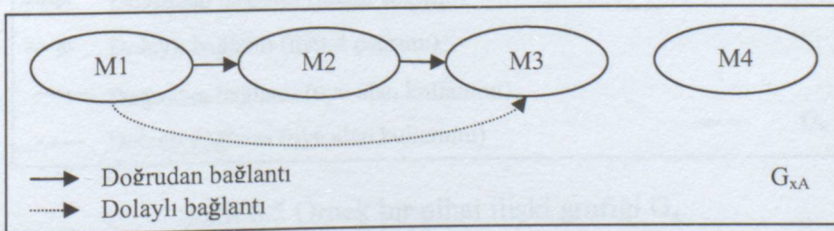
Yönlü bir grafik $G_{x2}(V, E)$ ve $V = M_x$ ise

$$E = \left\{ \langle m, l \rangle \in V \times V \mid \left(\langle m, n \rangle \in G_{x1} \right) \wedge \left(\langle n, l \rangle \in G_{x1} \right) \right\} \quad (6.2)$$

Örnek G_{x1} grafiğinden elde edilen G_{x2} grafiği Şekil 6.2'de görülebilir.

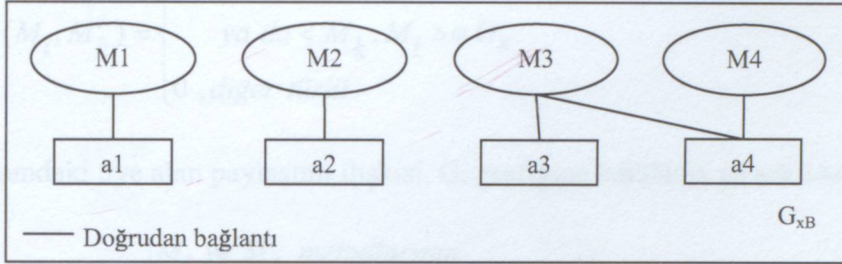
Şekil 6.2 Örnek bir dolaylı bağlantı grafiği G_{x2}

Metot çağırımlarının incelenmesinin son adımı olarak, önceki adımlarda elde edilmiş olan G_{x1} ve G_{x2} grafikleri tek bir grafikte G_{xA} grafiğinde birleştirilerek, doğrudan ve dolaylı yoldan bağlantılı olan tüm metotlar gösterilmiş olunur. Örnek G_{x1} ve G_{x2} grafiklerinden elde edilen G_{xA} grafiği Şekil 6.3'te görülebilir.

Şekil 6.3 Örnek bir metot çağırımı grafiği G_{xA}

6.2 Üye Alan Kullanımlarının İncelenmesi

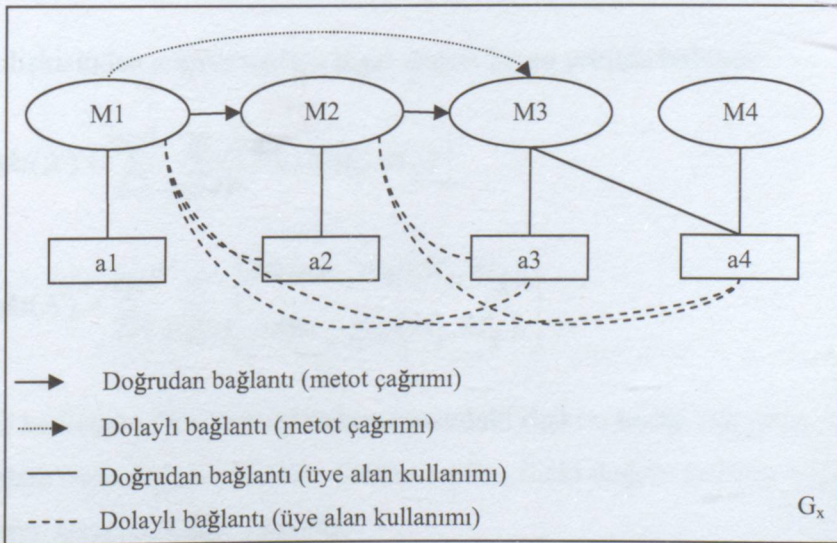
Üye alanların üye metotlar içerisinde kullanımı görebilmek için metot-alan ilişkisini gösteren bir grafik çizilir. Eğer bir metot bir alanı kullanıyorsa o zaman metot ve alan arasında bir çizgi oluşturulur. Şekil 6.4'te bu şekilde elde edilen örnek bir G_{XB} grafiği görülebilir.



Şekil 6.4 Örnek bir doğrudan üye alan kullanımı grafiği G_{XB}

6.3 Metot Çağırımı ve Üye Alan Kullanımı

Önceki adımlarda oluşturulan G_{XA} ve G_{XB} grafikleri bu adımda tek bir grafikte birleştirilerek G_X grafiği elde edilir. Dolaylı metot çağırımlarının incelenmesi ve dolaylı üye alan kullanımlarının tespit edilmesi ile G_X grafiğinin son hali elde edilir. Önceki bölümlerde verilen örnek grafikler kullanılarak elde edilen bir G_X grafiği Şekil 6.5'te görülebilir.



Şekil 6.5 Örnek bir nihai ilişki grafiği G_X

6.4 COMIAS Uyum Ölçütünün Tanımlanması

X bir sınıf, $I_x = X$ sınıfının üye alanlarının kümesi ve $M_x = X$ sınıfının metotlarının kümesi olsun. Yönlü bir grafik $G_x(V, E)$ göz önüne alındığında, iki metot arasındaki metot çağırımı ilişkisi şu şekilde tanımlanır:

$$Metod_cag(M_i, M_k) = \begin{cases} 1, & \text{eğer } < M_i, M_k > \in G_x \\ & \text{ya da } < M_k, M_i > \in G_x \\ 0, & \text{diğer türlü} \end{cases} \quad (6.3)$$

İki metot arasındaki üye alan paylaşımı ilişkisi, G_x grafiğine bakılarak şu şekilde bulunur:

$$Attr_pay(M_i, M_k) = n \left| \begin{array}{l} M_i \text{ ve } M_k \text{ metodlarının} \\ \text{paylastıkları attribute sayısı} \end{array} \right| \quad (6.4)$$

İki metot arasındaki ilişki, metot çağırımı ve üye alan paylaşımlarının toplamı olarak şu şekilde tanımlanır:

$İlişki(M_i, M_k) = M_i$ metodu ile M_k metodu arasındaki ilişki değeri, $M_i \in M_x$ ve $M_k \in M_x$, $i \neq k$ olarak tanımlansın. Bu durumda;

$$İlişki(M_i, M_k) = Metod_cag(M_i, M_k) + Attr_pay(M_i, M_k) \quad (6.5)$$

olarak tanımlanır.

İki metodun ilişkisinden sınıfın toplam ilişki değeri ise şu şekilde bulunur:

$$Toplam_İlişki(X) = \sum_{i=1}^{m-1} \sum_{k=i+1}^m [İlişki(M_i, M_k)] \quad (6.6)$$

$$Toplam_İlişki(X) = \sum_{i=1}^{m-1} \sum_{k=i+1}^m [Metod_cag(M_i, M_k) + Attr_pay(M_i, M_k)] \quad (6.7)$$

$İlişki(M_i, M_k)$ ne kadar yüksekse, iki metot arasındaki ilişki o kadar yüksek değerdedir denilir ve sınıfın toplam ilişki değeri yükselir. Sınıfın toplam ilişki değeri $Toplam_İlişki(X)$ ne kadar yüksekse sınıfın uyumu o kadar yükselir.

COMIAS'ın normalize bir ölçüt olması için bir sınıftaki maksimum ilişki değerini bulup bu sayıya yukarıdaki $Toplam_İlişki(X)$ değerini oranlamak gerekmektedir. Bir sınıftaki maksimum ilişki değeri $Max_İlişki(X)$, maksimum metot çağırımı ilişkisi ve maksimum üye alan paylaşımı ilişkisinin toplamı ile bulunur.

$$Max_İlişki(X) = Max(Metod_cag) + Max(Attr_pay) \quad (6.8)$$

Bir sınıftaki maksimum metot çağırımı ilişkisi şu şekilde tanımlansın:

$Max (Metod_cag) =$ Her (M_i, M_k) ilişkisinde bir çağırım vardır demektir ve bu da mümkün olan tüm (M_i, M_k) ilişki sayısı ile bulunur:

$$M_i \in M_x \text{ ve } M_k \in M_x, i \neq k. \quad (6.9)$$

X sınıfının m tane metodu varsa mümkün olan tüm (M_i, M_k) ilişki sayısı $m.(m-1)/2$ olacaktır. Her ilişkide bir çağırım olması durumu, maksimum metot çağırımı durumunu verecektir. Maksimum metot çağırımı şu şekilde ifade edilir:

$$Max(Metod_cag) = \frac{m.(m-1)}{2} \times 1 \quad (6.10)$$

Çizelge 6.1'de 4 adet metodu olan bir sınıf için bu durum örneklenerek açıklanmıştır. 4 adet metodu olan bir sınıf için tüm ilişkiler ve maksimum metot çağırımı ilişkisi bu çizelgede verilmektedir; her metot çifti için aralarında bir çağırım vardır.

Çizelge 6.1 Metotlar arasında maksimum çağırım ilişkisi örneği

	Metot çağırım durumu		Metot çağırım durumu
(M_1, M_2)	1	(M_2, M_3)	1
(M_1, M_3)	1	(M_2, M_4)	1
(M_1, M_4)	1	(M_3, M_4)	1
Max (Metod_cag)	$6 = 6 * 1$		

Aynı şekilde bir sınıftaki maksimum üye alan paylaşımı ilişkisi şu şekilde tanımlanır:

$Max (Attr_pay) =$ Her (M_i, M_k) ilişkisinde sınıfta bulunan tüm üye alanlar paylaşılıyor demektir:

$$M_i \in M_x \text{ ve } M_k \in M_x, i \neq k. \quad (6.11)$$

X sınıfının a tane üye alanı varsa, maksimum üye alan paylaşımı, üye alan sayısının mümkün olan tüm (M_i, M_k) ilişki sayısı $m.(m-1)/2$ ile çarpımı kadardır. Maksimum üye alan paylaşımı şu şekilde ifade edilir:

$$Max(Attr_pay) = \frac{m.(m-1)}{2} \times a \quad (6.12)$$

Çizelge 6.2’de 4 adet metodu ve 4 adet üye alanı olan bir sınıf için maksimum üye alan paylaşımı durumu örneklenerek açıklanmıştır. Her metot çifti, var olan tüm üye alanları paylaşıyor demektir.

Çizelge 6.2 Metotlar arasında maximum üye alan paylaşımı ilişkisi örneği

	Üye alan paylaşım durumu		Üye alan paylaşım durumu
(M ₁ , M ₂)	4	(M ₂ , M ₃)	4
(M ₁ , M ₃)	4	(M ₂ , M ₄)	4
(M ₁ , M ₄)	4	(M ₃ , M ₄)	4
Max(Attr_pay)	24 = 6 * 4		

Ölçütü normalize edebilmek için gereken Max_İlişki(X) değeri aşağıdaki gibi bulunur:

$$Max_İlişki(X) = \left(\frac{m.(m-1)}{2} \times 1 \right) + \left(\frac{m.(m-1)}{2} \times a \right) \quad (6.13)$$

Yukarıdaki tanımlamalardan yola çıkarak, önerilen ölçüt aşağıdaki gibi formüle edilir:

$$COMIAS = \frac{Toplam_İlişki(X)}{Max_İlişki(X)} \quad (6.14)$$

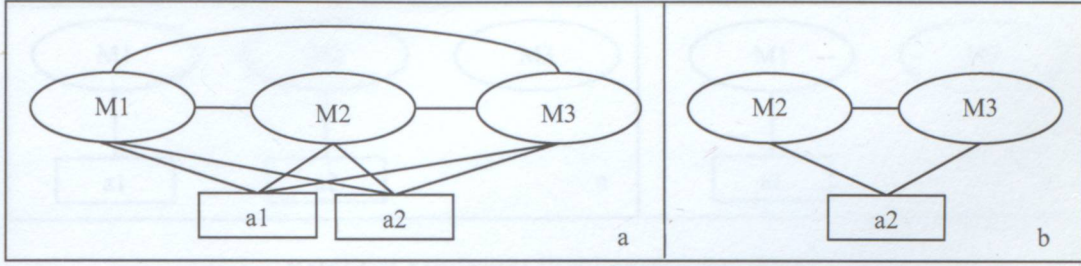
COMIAS ölçütü [0, 1] arasında normalize edilmiştir. 0 düşük uyumu, 1 yüksek uyumu göstermektedir.

6.5 COMIAS Uyum Ölçütünün Örneklenmesi

Bu bölümde, verilen hipotezler doğrultusunda yapılan örnekler ile COMIAS ölçütünün verdiği sonuçlar değerlendirilmektedir. Bu hipotezler, “13. World Scientific and Engineering Academy and Society International Conference on Computers” konferansında sunulmak üzere kabul alan bildiride (Ertemel vd., 2009b) ele alınmıştır.

HİPOTEZ 1:

“Maximum ve minimum değerler sezgisel olarak aynı elde etmeyi beklediğimiz örneklerde aynı çıkmalıdır”

Maximum ilişki:

Şekil 6.6 Maximum ilişki grafiği örnekleri

Şekil 6.6.a ve 6.6.b grafikleri maximum ilişki grafiklerine örnektir. Her iki grafik için ölçüt maximum uyum değeri olan “1” sonucunu vermelidir.

6.6.a grafiği için ölçüt hesaplaması:

$$m = 3, a = 2$$

Çizelge 6.3 Maximum ilişki grafiği 6.6.a için metot çağrım ve üye alan paylaşım durumu

	Metot çağrım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	1	2
(M ₁ , M ₃)	1	2
(M ₂ , M ₃)	1	2

$$COMIAS = \frac{[(1+2) + (1+2) + (1+2)]}{\left(\frac{3 \times (3-1)}{2} \times 1\right) + \left(\frac{3 \times (3-1)}{2} \times 2\right)} = \frac{9}{9} = 1$$

6.6.b grafiği için ölçüt hesaplaması:

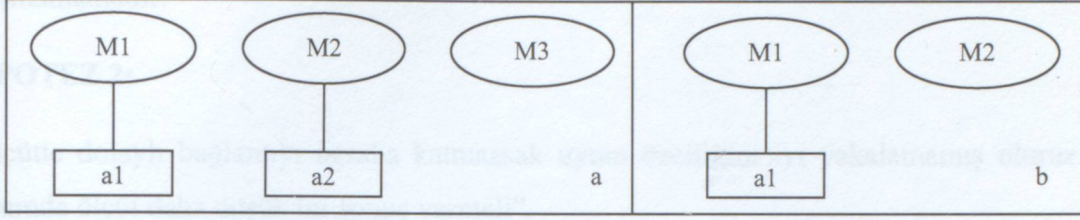
$$m = 2, a = 1$$

Çizelge 6.4 Maximum ilişki grafiği 6.6.b için metot çağrım ve üye alan paylaşım durumu

	Metot çağrım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	1	1

$$COMIAS = \frac{[(1+1)]}{\left(\frac{2 \times (2-1)}{2} \times 1\right) + \left(\frac{2 \times (2-1)}{2} \times 1\right)} = \frac{2}{2} = 1$$

Her iki grafik için ölçüt maximum uyum değeri olan “1” sonucunu vermektedir.

Minimum ilişki:

Şekil 6.7 Minimum ilişki grafiği örnekleri

Şekil 6.7.a ve 6.7.b grafikleri minimum ilişki grafiklerine örnektir. Her iki grafik için ölçüt minimum uyum değeri olan “0” sonucunu vermelidir.

6.7.a grafiği için ölçüt hesaplaması:

$$m = 3, a = 2$$

Çizelge 6.5 Minimum ilişki grafiği 6.7.a için metot çağrım ve üye alan paylaşım durumu

	Metot çağrım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	0	0
(M ₁ , M ₃)	0	0
(M ₂ , M ₃)	0	0

$$COMIAS = \frac{[(0+0) + (0+0) + (0+0)]}{\left(\frac{3 \times (3-1)}{2} \times 1\right) + \left(\frac{3 \times (3-1)}{2} \times 2\right)} = \frac{0}{9} = 0$$

6.7.b grafiği için ölçüt hesaplaması:

$$m = 2, a = 1$$

Çizelge 6.6 Minimum ilişki grafiği 6.7.b için metot çağrım ve üye alan paylaşım durumu

	Metot çağrım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	0	0

$$COMIAS = \frac{[(0+0)]}{\left(\frac{2 \times (2-1)}{2} \times 1\right) + \left(\frac{2 \times (2-1)}{2} \times 1\right)} = \frac{0}{2} = 0$$

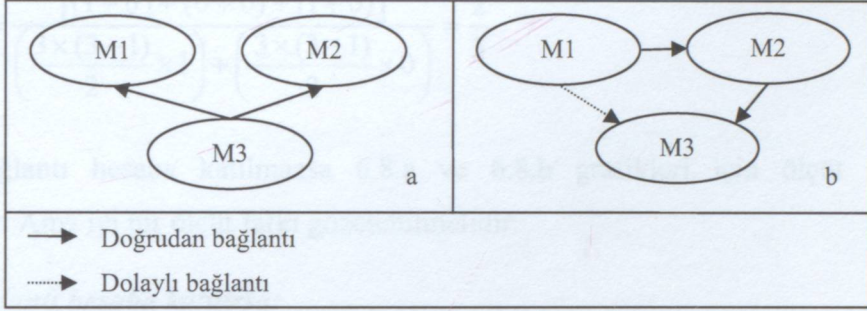
Her iki grafik için ölçüt minimum uyum değeri olan “0” sonucunu vermektedir.

Görülüyor ki, ölçüt sezgisel olarak uyum özelliği aynı beklenen sınıflar için aynı sonucu

vermektedir. Aynı zamanda, sonuçlarda herhangi bir belirsizlik ya da tanımsız bir durum oluşmamaktadır.

HİPOTEZ 2:

“Ölçütte dolaylı bağlantıyı hesaba katmazsak uyum özelliğini iyi yakalamamış oluruz. Bu durumda ölçüt daha düşük bir sonuç vermeli”.



Şekil 6.8 Dolaylı bağlantı değerlendirmesi için grafik örnekleri

Şekil 6.8 grafiğine sezgisel olarak bakıldığında 6.8.b grafiğinin uyum değeri 6.8.a grafiğinden daha iyi çıkmalıdır. Fakat dolaylı bağlantı hesaba katılmazsa şu sonuç elde edilmektedir:

Dolaylı bağlantı hesaba katılmazsa:

Dolaylı bağlantı hesaba katılmazsa 6.8.a grafiği için ölçüt hesaplaması:

$$m = 3, a = 0$$

Çizelge 6.7 Dolaylı bağlantı hesaba katılmadan 6.8.a grafiği için metot çağrım ve üye alan paylaşım durumu

	Metot çağrım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	0	0
(M ₁ , M ₃)	1	0
(M ₂ , M ₃)	1	0

$$COMIAS = \frac{[(0+0) + (1+0) + (1+0)]}{\left(\frac{3 \times (3-1)}{2} \times 1\right) + \left(\frac{3 \times (3-1)}{2} \times 0\right)} = \frac{2}{3}$$

Dolaylı bağlantı hesaba katılmazsa 6.8.b grafiği için ölçüt hesaplaması:

$$m = 3, a = 0$$

Çizelge 6.8 Dolaylı bağlantı hesaba katılmadan 6.8.b grafiği için metot çağrım ve üye alan paylaşım durumu

	Metot çağrım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	1	0
(M ₁ , M ₃)	0	0
(M ₂ , M ₃)	1	0

$$COMIAS = \frac{[(1+0) + (0+0) + (1+0)]}{\left(\frac{3 \times (3-1)}{2} \times 1\right) + \left(\frac{3 \times (3-1)}{2} \times 0\right)} = \frac{2}{3}$$

Dolaylı bağlantı hesaba katılmazsa 6.8.a ve 6.8.b grafikleri için ölçüt aynı sonucu vermektedir. Ama iyi bir ölçüt farkı gözetebilmelidir.

Dolaylı bağlantı hesaba katılırsa:

Dolaylı bağlantı hesaba katılırsa 6.8.a grafiği için ölçüt hesaplaması değişmiyor çünkü dolaylı bağlantı yok.

Dolaylı bağlantı hesaba katılırsa 6.8.b grafiği için ölçüt hesaplaması:

Çizelge 6.9 Dolaylı bağlantı hesaba katılırsa 6.8.b grafiği için metot çağrım ve üye alan paylaşım durumu

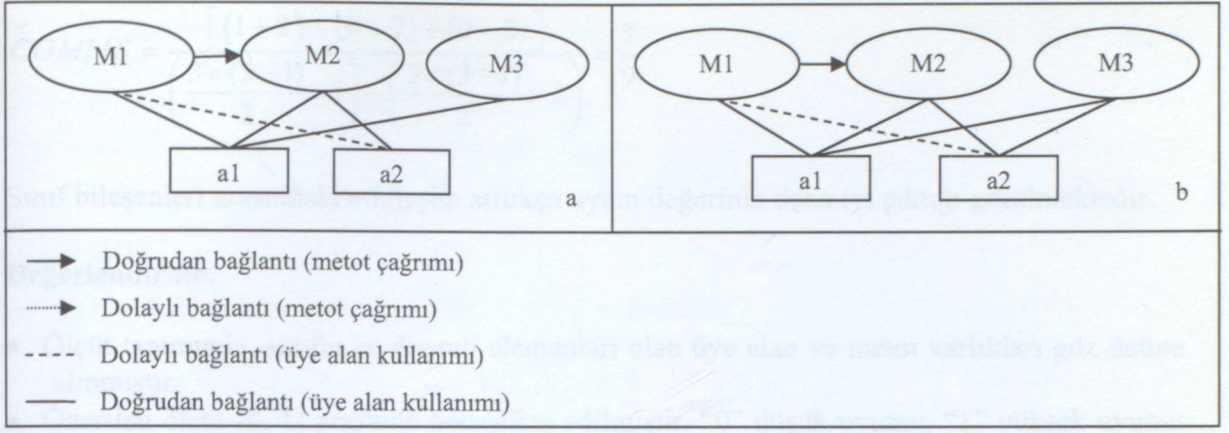
	Metot çağrım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	1	0
(M ₁ , M ₃)	1	0
(M ₂ , M ₃)	1	0

$$COMIAS = \frac{[(1+0) + (1+0) + (1+0)]}{\left(\frac{3 \times (3-1)}{2} \times 1\right) + \left(\frac{3 \times (3-1)}{2} \times 0\right)} = \frac{3}{3} = 1$$

Dolaylı bağlantı hesaba katılırsa sezgilerle paralel sonuç alınmaktadır. 6.8.b grafiğinin uyum değeri daha iyi çıktı. Dolaylı bağlantıları hesaba katmanın önemi bu örnekle görülmektedir. Dolaylı bağlantıları görebilmenin tek yolu da yönlü grafik çizmektir.

HİPOTEZ 3:

“Sınıf bileşenleri arasındaki etkileşim arttıkça uyum değeri daha iyi çıkmalı”.



Şekil 6.9 Etkileşim değerlendirme için grafik örnekleri

Şeki 6.9 grafiğine sezgisel olarak bakıldığında 6.9.b grafiğinin uyum değeri 6.9.a grafiğinden daha iyi çıkmalıdır.

6.9.a grafiği için ölçüt hesaplaması:

$$m = 3, a = 2$$

Çizelge 6.10 6.9.a grafiği için metot çağırımı ve üye alan paylaşım durumu

	Metot çağırımı durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	1	2
(M ₁ , M ₃)	0	1
(M ₂ , M ₃)	0	1

$$COMIAS = \frac{[(1+2) + (0+1) + (0+1)]}{\left(\frac{3 \times (3-1)}{2} \times 1\right) + \left(\frac{3 \times (3-1)}{2} \times 2\right)} = \frac{5}{9}$$

6.9.b grafiği için ölçüt hesaplaması:

$$m = 3, a = 2$$

Çizelge 6.11 6.9.b grafiği için metot çağırımı ve üye alan paylaşım durumu

	Metot çağırımı durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	1	2
(M ₁ , M ₃)	0	2
(M ₂ , M ₃)	0	2

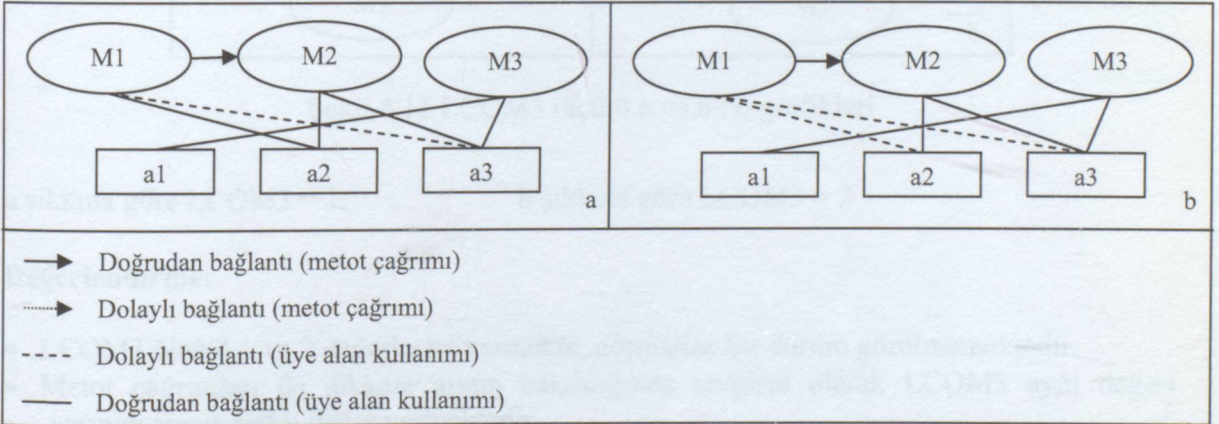
$$COMIAS = \frac{[(1+2) + (0+2) + (0+2)]}{\left(\frac{3 \times (3-1)}{2} \times 1\right) + \left(\frac{3 \times (3-1)}{2} \times 2\right)} = \frac{7}{9}$$

Sınıf bileşenleri arasındaki etkileşim arttıkça uyum değerinin daha iyi çıktığı görülmektedir.

Değerlendirme:

- Ölçüt tanımında, sınıfın en önemli elemanları olan üye alan ve metot varlıkları göz önüne alınmıştır.
- Önerilen ölçüt $[0, 1]$ arasında normalize edilmiştir. “0” düşük uyumu, “1” yüksek uyumu göstermektedir.
- Sezgilerle paralellik göstermektedir. 2 tane farklı uyum özelliğine sahip sınıf farklı değer almaktadır.
- Metot çağrımları göz önüne alınmıştır.
- Dolaylı bağlantılar göz önüne alınmıştır.
- Ölçüt, matematiksel olarak belirsiz ve tanımsız sonuçlar vermemektedir.
- Kalıtım özelliği göz önüne alınmamıştır.
- Özel metotlardan erişim metotları *uyumu* yapay olarak düşürürken yapılandırıcı metotlar *uyumu* yapay olarak arttırmaktadırlar. Literatürdeki ölçütlerin incelendiği 4. bölümde birçok ölçüt bu yönde sonuç vermiştir. COMIAS ölçütünde bu gerçek bilinmekle beraber herhangi bir analiz yönlendirmesi yapılmamaktadır.

6.6 Önerilen Ölçüt ve LCOM Ölçütlerinin Bir Örnek Üzerinde İncelenmesi



Şekil 6.10 LCOM ölçütleri ve COMIAS değerlendirmesi için grafik örnekleri

1) LCOM1

a şıkkına göre LCOM1: $\{I_1\} = \{a2\}$, $\{I_2\} = \{a2, a3\}$, $\{I_3\} = \{a1, a3\}$, LCOM1 = 1

b şıkkına göre LCOM1: $\{I_1\} = \{-\}$, $\{I_2\} = \{a2, a3\}$, $\{I_3\} = \{a1, a3\}$, LCOM1 = 2

Değerlendirme:

- LCOM1 ölçütü 1 ve 2 değerlerini vermekte, normalize bir durum görülmemektedir.

- Metot çağrımları da dikkate alınıp bakıldığında sezgisel olarak LCOM1 aynı değeri vermeli ancak farklı değer vermektedir.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.

2) LCOM2

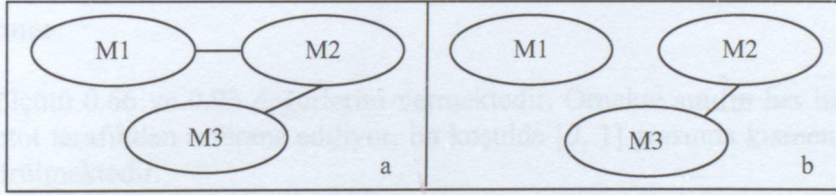
a şıkkına göre LCOM2: $P = 1, Q = 2, LCOM2 = P < Q = 0$

b şıkkına göre LCOM2: $P = 2, Q = 1, LCOM2 = P - Q = 2 - 1 = 1$

Değerlendirme:

- LCOM2 ölçütü 0 ve 1 değerlerini vermekte, normalize bir durum görülmemektedir.
- Metot çağrımları da dikkate alınıp bakıldığında sezgisel olarak LCOM2 aynı değeri vermeli ancak farklı değer vermektedir.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.

3) LCOM3



Şekil 6.11 LCOM3 ölçütü a ve b G_x grafikleri

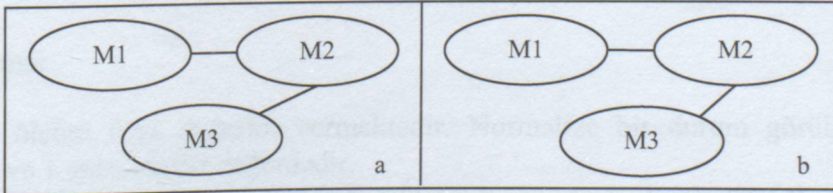
a şıkkına göre LCOM3 = 1,

b şıkkına göre LCOM3 = 2

Değerlendirme:

- LCOM3 ölçütü 1 ve 2 değerlerini vermekte, normalize bir durum görülmemektedir.
- Metot çağrımları da dikkate alınıp bakıldığında sezgisel olarak LCOM3 aynı değeri vermeli ancak farklı değer vermektedir.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.

4) LCOM4



Şekil 6.12 LCOM4 ölçütü a ve b G_x grafikleri

a şıkkına göre LCOM4 = 1,

b şıkkına göre LCOM4 = 1

Değerlendirme:

- LCOM4 ölçütü 1 değerini vermekte, normalize bir durum görülmemektedir.
- Metot çağrımları da dikkate alınıp bakıldığında sezgisel olarak LCOM4 aynı değeri vermeli. LCOM4 aynı değeri vermiştir.
- Metot çağrımları göz önüne alınmaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.

5) LCOM5

$$a \text{ şıkkına göre } LCOM5: m = 3, a = 3, LCOM5 = \frac{\frac{1}{3}(1+2+2) - 3}{1-3} = 0.66$$

$$b \text{ şıkkına göre } LCOM5: m = 3, a = 3, LCOM5 = \frac{\frac{1}{3}(1+1+2) - 3}{1-3} = 0.83$$

Değerlendirme:

- LCOM5 ölçütü 0.66 ve 0.93 değerlerini vermektedir. Örnekte sınıfın her bir üye alanı en az bir metot tarafından referans ediliyor, bu koşulda [0, 1] arasında kısmen normalize bir durum görülmektedir.
- Metot çağrımları da dikkate alınıp bakıldığında sezgisel olarak LCOM5 aynı değeri vermeli ancak farklı değer vermektedir.
- Metot çağrımları göz önüne alınmamaktadır.
- Dolaylı bağlantılar göz önüne alınmamaktadır.

6) COMIAS

$$a \text{ şıkkına göre } COMIAS: COMIAS = \frac{[(1+2) + (0+1)] + [(0+1)]}{\left(\frac{3 \times (3-1)}{2}\right) \times (3+1)} = \frac{5}{12} = 0.41$$

$$b \text{ şıkkına göre } COMIAS: COMIAS = \frac{[(1+2) + (0+1)] + [(0+1)]}{\left(\frac{3 \times (3-1)}{2}\right) \times (3+1)} = \frac{5}{12} = 0.41$$

Değerlendirme:

- COMIAS ölçütü 0.41 değerini vermektedir. Normalize bir durum görülmekte, ölçüm sonucu 0 ve 1 arasında bir değerdedir.
- Metot çağrımları da dikkate alınıp bakıldığında sezgisel olarak COMIAS aynı değeri vermeli. COMIAS aynı değeri vermektedir.
- Metot çağrımları göz önüne alınmaktadır.

- Dolaylı bağlantılar göz önüne alınmaktadır.

LCOM1, LCOM2, LCOM3 ve LCOM5 ölçütlerinin sonucuna göre b şikkının uyum değeri daha düşük çıkmıştır. Oysa a ve b şıklarının uyum değeri aynı çıkmalıdır. Bu ölçütler metot çağrılarını hesaba katmadıkları için uyum özelliğini iyi yakalayamayabiliyorlar.

Metot çağrımını hesaba kattığı için LCOM4 ölçütü a ve b şıkları için aynı sonucu vermiştir fakat LCOM4 ölçütünü normalizasyonu olmadığı için yorumlamak kolay değildir. Zaten Yönsüz bir grafikte 1 tane bağlı bileşen bulunduğu durumda her zaman $LCOM4 = 1$ değeri elde edilmektedir (Bknz. Bölüm 4.2.2), (Hitz ve Montazeri, 1995).

COMIAS ölçütünde ise sezgilerle paralel olarak a ve b şıkları için elde edilen değerler aynıdır. COMIAS ölçütü için normalizasyondan dolayı 0'ın düşük uyumu 1'in yüksek uyumu ifade ettiği bilindiğinden, çıkan 0.41 değeri a ve b şıklarında verilen örnek grafiklerin pek iyi bir uyum özelliğine sahip olmadıklarını göstermektedir.

6.7 Teorik Geçerleme

Literatürde ölçütleri teorik olarak değerlendirmek üzere önerilen bir takım kriter kümeleri sunulmuştur. Bu kriter kümeleri, mevcut ölçütleri geçerlilik yönünde değerlendirme imkanı sağlarken bir yandan da yeni önerilecek ölçütlerin geçerli bir ölçüt olması yönünde nasıl tanımlanmaları gerektiği konusunda bir kılavuz özelliği gösterirler. Kriter kümeleri, mevcut ölçüt kümelerinin güçlülük ve zayıflıklarını ortaya çıkararak uygun ölçütü seçmeye yardımcı olurlar.

Weyuker (Weyuker, 1988) tarafından 1988 yılında karmaşıklık ölçütlerinin değerlendirilmesine yönelik bir kriter kümesi önerilmiştir. Karmaşıklık ölçütlerinin bu kriter kümesindeki özellikleri sağlaması gerektiğine inanılmıştır. Weyuker kriter kümesi uyum ölçütleri öneren kimselerce de teorik geçerleme yapmak amacıyla kullanılmıştır. Önerilen ölçütün, Weyuker kriter kümesi kullanılarak teorik geçerlemesi yapılmaktadır.

Özellik 1- Non-Coarseness (Aynı değeri vermeme):

Bu özellik, ölçütün bütün sınıflar için aynı değeri vermemesi gerektiğini belirtir. COMIAS bu özelliği sağlamaktadır çünkü ölçüt, çoğu sınıfta farklı oluşturulacak olan sınıfın metotları arasındaki metot ve üye alan ilişkisine dayanmaktadır.

Özellik 2-Granularity (Sonluluk):

Bu özellik “c, pozitif bir sayı olsun. c ölçüt değerine sahip sonlu sayıda sınıf vardır” demektir. Bir sınıf sonlu sayıda üye alan ve metoda sahip olabilir. Dahası, NY bir program sonlu sayıda sınıf içerebilir. Bu nedenle özellik 2 de, COMIAS ölçütü tarafından sağlanmaktadır.

Özellik 3-Non-Uniqueness (Tekil olmama):

Bu özellik, bir ölçütün tekil bir sonuç üretmemesini vurgular. Maximum uyuma sahip iki sınıfı ölçmek, tıpkı minimum uyuma sahip iki sınıfı ölçmek ya da mümkün olan maximum metot çağırımı ve üye alan paylaşımı ilişki sayısının yarısına sahip iki sınıfı ölçmek gibi aynı sonucu verecektir. Sonuç olarak özellik 3, COMIAS ölçütü tarafından sağlanmaktadır.

Özellik 4 – Implementation (Design) Details are important (Tasarım detayları önemlidir):

Bu özellik, ölçütün, farklı tasarıma fakat aynı fonksiyonelliğe sahip herhangi iki sınıf için farklı sonuçlar vermesini bekler:

$$(\exists P)(\exists Q)(P = Q \& |P| \neq |Q|). \quad (6.15)$$

COMIAS ölçütü bu özelliği sağlar çünkü farklı tasarlanmış fakat aynı hizmeti sağlayan iki farklı sınıf oluşturmak mümkündür. Bu şekilde farklı tasarlanarak oluşturulmuş sınıflar farklı sayıda metot, üye alan ve etkileşim sayılarına sahip olacaktır. Bu farklılıklar COMIAS ölçütü hesaplamasında farklı sayıların kullanılmasına yol açacak ve farklı sonuçlar üretilecektir.

Özellik 5 – Monotonicity (Süreklilik):

Bu özellik, programın bileşenlerinin programdan daha karmaşık olmaması gerektiğini vurgular. Weyuker karmaşıklık ölçütünün sürekli artan bir fonksiyon olmasını önerir:

$$(\forall P)(\forall Q)(|P| \leq |P;Q| \vee |Q| \leq |P;Q|). \quad (6.16)$$

Bu özellik uyum ölçütlerine uygulanamamaktadır çünkü iki sınıfın birleşimi uyumu çoğu zaman düşürecektir. Gerçekte, bir sınıf için düşük uyum, bu sınıfın daha iyi bir tasarım için iki veya daha fazla alt sınıfa ayrılması gerektiğini önerir. Tek istisna bu iki sınıfın ‘özellik kıskançlığı’ (feature envy) olarak adlandırılabilir bir kod kusurunu (code smell) taşımasıdır (Fowler vd., 1999). Bu istenmeyen bir durum olan yüksek bağlaşıma sahip bir tasarıma götürecek yanlış bir programlama pratiğidir. Bu mantıkla, tez yazarı uyum ölçütlerinin sürekli azalan bir yapıda olmalarının tercih edilen durum olduğunu vurgular. COMIAS ölçütünün de

bu kapsamda sürekli azalması beklenir: 6.13 eşitliğinde verilen Max_İlişki hesaplaması bu durumda artacaktır ve 6.14 eşitliğine yerleştirildiğinde COMIAS ölçüm sonucunun azalmasına neden olacaktır.

Özellik 6 – None Equivalence of Interaction (Etkileşimin eşitsizliği):

Bu özellik, bir programın aynı ölçüme sahip bir veya iki farklı programla birleşiminin birleşimde farklı etkilere sahip olması gerektiğini vurgular (Eşitlik 6.17). COMIAS bu özelliğe uymaktadır, çünkü 6.17 eşitliğindeki $\langle P, R \rangle$ ve $\langle Q, R \rangle$ sınıf çiftleri arasındaki bağlaşım farklıdır. Diğer bir ifadeyle, P ve Q sınıfları aynı uyuma sahip olsa da, R sınıfı ile etkileşimleri (bağlaşımları) çoğu durumda farklı olacaktır. Bu fark COMIAS ölçütünün farklı sonuçlar üretmesine neden olacaktır.

$$\begin{aligned} a. & (\exists P)(\exists Q)(\exists R)(|P| = |Q| \& |P; R| \neq |Q; R|) \\ b. & (\exists P)(\exists Q)(\exists R)(|P| = |Q| \& |R; P| \neq |R; Q|) \end{aligned} \quad (6.17)$$

Özellik 7 – Permutation is Significant (Sıralama anlamlıdır):

Bu özellik bir ölçütün bir programın kod satırlarının sıralamasına duyarlı olması gerektiğini belirtir. Fakat bir sınıfın uyumu, üye alanların, metotların ve metot içerisindeki kod satırlarının sıralamasından bağımsızdır. Bu nedenle, COMIAS dahil bütün uyum ölçütleri bu özelliği sağlamaz. Bu mantıkla, bu özelliğin uyum ölçütleri için anlamlı olmadığı tartışılabilir.

Özellik 8 – Renaming Property (Yeniden adlandırma özelliği):

Bu özellik bir ölçütün, biri diğerinin yeniden adlandırılmış hali olan iki sınıf için aynı sonucu üretmesi gerektiğini söyler. Bu özellik, bir sınıfın tüm üye alanlarının, metotlarının ve sınıfın kendisinin yeniden adlandırılmasının sınıfın uyumunu değiştirmeyeceği konusu ile bağdaşır. COMIAS dahil bütün uyum ölçütleri sınıfın isimlerini değil, üyelerini kullanır.

Özellik 9 – Interaction Increases Complexity (Reduces Cohesion) (Etkileşim karmaşıklığı artırır (Uyumu düşürür)):

Bu özellik iki program içeriğinin birleştirilmesi ile oluşturulan bir programın karmaşıklığının iki programdan da daha fazla olması gerektiğini vurgular:

$$(\exists P)(\exists Q)(|P| + |Q| < |P; Q|). \quad (6.18)$$

Bu özellik Özellik 5'e benzer; benzer tartışmalar burada da geçerlidir. Normal durumlarda, iki sınıfın birleşimi uyumu azaltır, dolayısıyla bu tezin yazarı COMIAS dahil tüm uyum

ölçütlerinin bu özelliği 6.19 eşitliğinde olduğu gibi ters yönde sağlaması gerektiğini belirtir.

$$(\exists P)(\exists Q)(|P| + |Q| > |P; Q|). \quad (6.19)$$

Çizelge 6.12’de COMIAS ölçütünün Weyuker kriter kümesine göre analitik olarak değerlendirmesinin sonucu verilmektedir.

Çizelge 6.12 COMIAS ölçütünün Weyuker kriter kümesine göre değerlendirilmesi

Ö1	Ö2	Ö3	Ö4	Ö5	Ö6	Ö7	Ö8	Ö9
S	S	S	S	KS	S	U/SZ	S	KS

S: Sağlanır SZ: Sağlanmaz KS: Kısmen Sağlanır U: Uygulanmaz

6.8 Deneyisel Geçerleme

Bu bölümde, deneyisel geçerlemede kullanılan program ve kalitesi ölçülecek yazılımın seçimi anlatılmakta, yapılan ölçüm sonuçları ile miras konusunun ve ölçümlerde ilginç bulunan sonuçların bir değerlendirmesi verilmektedir.

6.8.1 Kullanılan Program

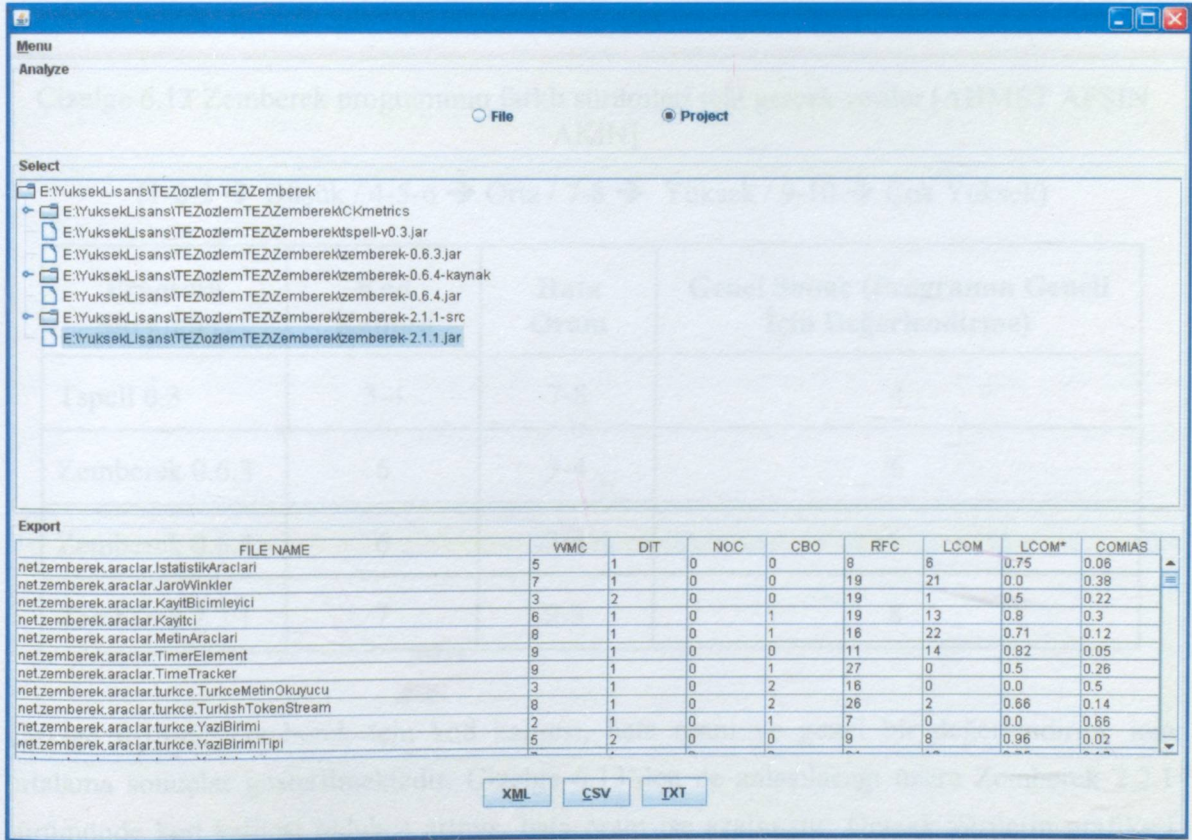
Chidamber ve Kemerer’in nesneye yönelik ölçütlerini hesaplayan çeşitli programlar bulunmaktadır. Bunların bir kısmı ücretli, bir kısmı ise ücretsizdir. Ücretli alternatiflerde, Essential Metrics, JStyle, JHawk ve Analyst4j gibi programlar göze çarpmaktadır. JMetric, Byte Code Metric Library (BCML) ve ckjm (Chidamber and Kemerer Java Metrics) ise ücretsiz olanlardır.

Deneyisel geçerlemede kullanılacak yazılım daha önce “Nesneye Yönelik Programlamada Kalite Ölçütlerinin Hesaplanması” (NilSeri, 2008) adlı ara proje kapsamında açık kaynak, Java programlama dili ile geliştirilmiş ckjm tabanlı yazılım kalite ölçüm programıdır.

Ckjm, derlenmiş Java dosyalarının byte kodunu işlemiden geçirerek Chidamber ve Kemerer nesneye dayalı ölçütlerini ölçer. Program her sınıf için Chidamber ve Kemerer tarafından teklif edilen altı ölçütü de hesaplar.

“Nesneye Yönelik Programlamada Kalite Ölçütlerinin Hesaplanması” (NilSeri, 2008) adlı ara proje kapsamında ckjm kodlarının, işlevini kullanıcı dostu bir şekilde yerine getirebilmesi için bir arayüz oluşturulmuştur. Arayüz geliştirmenin yanı sıra LCOM5 ölçütü kodlaması eklenerek hesaplanması sağlanmıştır.

COMIAS ölçütünün deneysel geçerlemesini yapabilmek amacıyla “Nesneye Yönelik Programlamada Kalite Ölçütlerinin Hesaplanması” (NilSeri, 2008) adlı ara proje çalışması kullanılmaktadır. Ara proje çalışmasına COMIAS ölçütünün hesaplanması için gerekli olan kodlama eklentileri yapılmıştır. Mevcut çalışmada ölçülen ölçütler için var olan kodlama COMIAS ölçütünün hesaplanması için yeterli değildi. COMIAS ölçütü dolaylı bağlantıları ve metot çağrılarını içerdiğinden ilave eklentiler yapma ihtiyacı doğdu. Gerekli olan tüm kodlama eklentileri sağlanarak aynı yazılımdan COMIAS ölçütünün de hesaplanması sağlanmıştır. Oluşturulan programın arayüzünün gösterimi Şekil 6.13’de verilmektedir. Ölçütlerin hesaplama sonuçları arayüzden görülebilmektedir.



Şekil 6.13 Program arayüzü

6.8.2 Kalitesi Ölçülecek Yazılımın Seçimi

COMIAS uyum ölçütünün deneysel olarak geçerlenmesi amacıyla Zemberek adı verilen açık kaynak kodlu projenin çeşitli sürümleri kullanılmaktadır. Zemberek, açık kaynak kodlu Türkçe doğal dil işleme kütüphanesi ve OpenOffice eklentisidir. Zemberek'in Java'da geliştirilmiş olması, kalite açısından aralarında karşılaştırma yapılabilecek farklı sürümlerinin bulunması ve nesne tabanlı olması, bu programın seçilmesinde etkili olmuştur.

6.8.3 Ölçüm Sonuçları

Zemberek ile İlgili Gerçek Veriler

COMIAS tarafından üretilen sonuçların gerçek dünya verileriyle karşılaştırılabilmesi için kullanıcı görüşlerine başvurulmuştur. Bu amaçla Zemberek programı geliştirici grubundan olan Ahmet Afşin AKIN ile görüşülmüştür ve şu bilgiler elde edilmiştir:

- Tspell (Zemberek programının eski adı) 0.3 daha çok bir deneme sürümüdür. Sistem sadece belirli işlemleri yüksek sayılmayacak bir başarı ile yapabilmektedir.
- Zemberek 0.6.3 ve 0.6.4 arasında aslında çok fazla fark bulunmamaktadır. Çünkü asıl zıplama Zemberek 0.5 ile Zemberek 0.6 arasında gerçekleşmiştir.
- Zemberek 2.2.1 ile kod olgunluğa ulaşmıştır ve çalışmalar daha çok alt yapıda bazı düzenlemeler ile hata giderimi şeklinde olmuştur.

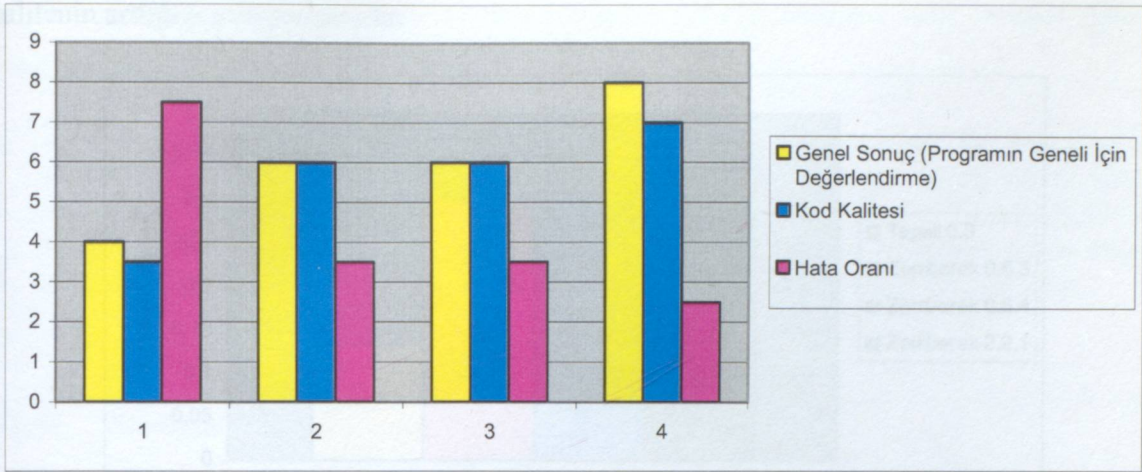
Çizelge 6.13 Zemberek programının farklı sürümleri için gerçek veriler [AHMET AFŞİN AKIN]

(1-2-3 → Düşük / 4-5-6 → Orta / 7-8 → Yüksek / 9-10 → Çok Yüksek)

Program Sürümleri	Kod Kalitesi	Hata Oranı	Genel Sonuç (Programın Genelî İçin Değerlendirme)
Tspell 0.3	3-4	7-8	4
Zemberek 0.6.3	6	3-4	6
Zemberek 0.6.4	6	3-4	6
Zemberek 2.2.1	7	2-3	8

Çizelge 6.13'de Zemberek için kod kalitesi, hata oranı ve genel bir değerlendirme için ortalama sonuçlar gösterilmektedir. Çizelge 6.13'den de anlaşılacağı üzere Zemberek 2.2.1 sürümünde kod kalitesi oldukça artmış, hata oranı ise azalmıştır. Gerçek verilerin grafiksel gösterimi ise Şekil 6.14'de verilmektedir.

	Tspell 0.3	Zemberek 0.6.3	Zemberek 0.6.4	Zemberek 2.2.1
Kod Kalitesi	3.5	6.0	6.0	7.0
Hata Oranı	7.5	3.5	3.5	2.5
Genel Sonuç	4.0	6.0	6.0	8.0



Şekil 6.14 Sürümler için elde edilen gerçek verilerin grafiksel gösterimi

Bu bilgilere dayanarak, Zemberek programı için sürüm sayıları arttıkça genel olarak programın kalitesinin arttığı söylenebilir. COMIAS ölçütünün de bu görüşe paralel sonuçlar vermesi beklenmektedir.

Zemberek Programı Ölçüm Sonuçları

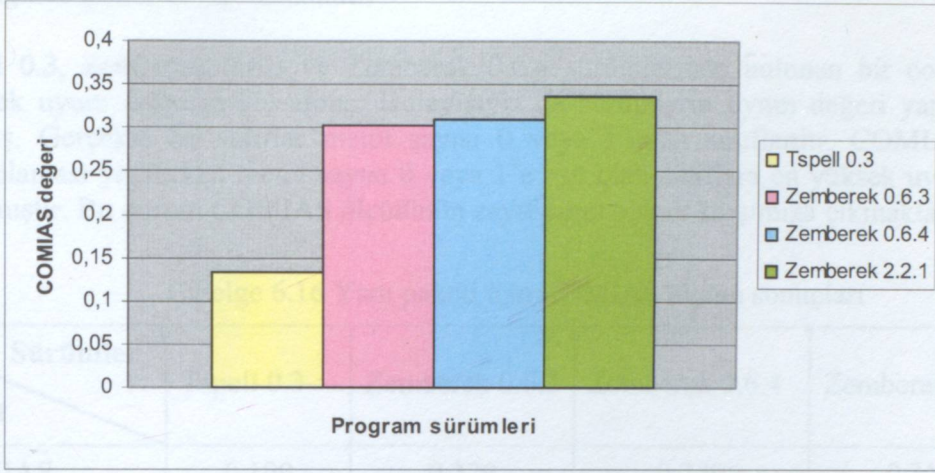
Kalite ölçüm programında Zemberek programının farklı sürümleri kullanılarak elde edilen sonuçlar Çizelge 6.14' de verilmektedir. Kalite ölçüm programında seçilen sürümün her sınıfı için elde edilen sonuçların aritmetik ortalaması alınarak elde edilen değer, sürümün seçilen ölçüt için genel değeri olarak alınmıştır. Deneysel geçerleme sonuçları, “4. Ulusal Yazılım Mühendisliği Sempozyumu”nda sunulmak üzere kabul alan bildiride (Ertemel vd., 2009a) ve “13. World Scientific and Engineering Academy and Society International Conference on Computers” konferansında sunulmak üzere kabul alan diğer bir bildiride (Ertemel vd., 2009b) ele alınmıştır.

Çizelge 6.14 Zemberek programının ilerleyen sürümleri için COMIAS ölçütü ile elde edilen sonuçlar

Sürümler Ölçüt	Tspell 0.3	Zemberek 0.6.3	Zemberek 0.6.4	Zemberek 2.2.1
COMIAS	0,133	0,303	0,310	0,337

Çizelge 6.14 incelendiğinde COMIAS ölçütünün Zemberek ile ilgili gerçek verilerle paralel sonuçlar verdiği görülmektedir. COMIAS ölçütü ölçüm sonuçları sürümlerde uyumun ve

kalitenin arttığını göstermektedir.



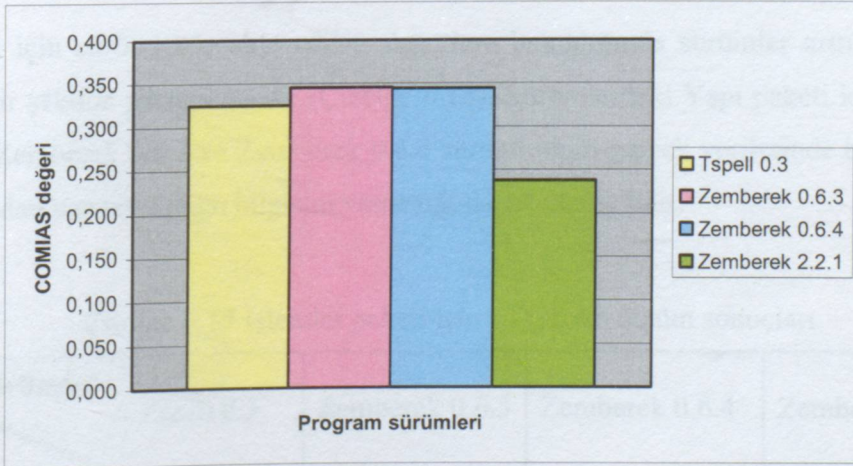
Şekil 6.15 Program sürümleri için elde edilen sonuçların grafiksel gösterimi

Paket Seviyesinde Ölçüm Sonuçları

Bu bölümde Zemberek yazılımının seçilen sürümlerinin hepsinde bulunan ortak paketler için ölçüm yapılmıştır (Çizelge 6.15).

Çizelge 6.15 Araçlar paketi için COMIAS ölçüm sonuçları

Sürümler Ölçüt	Tspell 0.3	Zemberek 0.6.3	Zemberek 0.6.4	Zemberek 2.2.1
COMIAS	0,325	0,345	0,344	0,238



Şekil 6.16 Araçlar paketi için elde edilen sonuçların grafiksel gösterimi

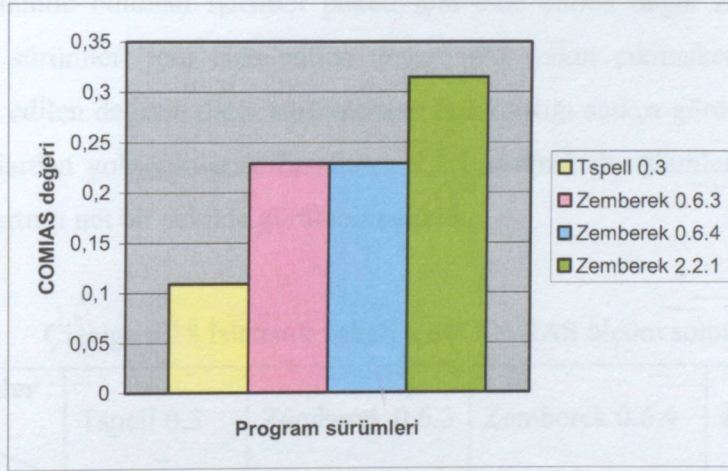
Araçlar paketi için elde edilen değerlere bakıldığında sürümler arttıkça kalitenin arttığı

görülememektedir (Çizelge 6.15). Ölçüm sonuçları pakette bulunan sınıf bazında detaylı incelendiğinde şu tespit yapılmaktadır:

- Tspell 0.3, Zemberek 0.6.3 ve Zemberek 0.6.4 sürümlerinde bulunan bir çok sınıf en yüksek uyum değerini (1) almış. Dolayısıyla da sürümlerin uyum değeri yapay olarak artmış. Gerçekte bu sınıflar metot sayısı 0 veya 1 olan sınıflardır. COMIAS ölçütü hesaplaması yapılırken metot sayısı 0 veya 1'e eşit olan sınıflara en yüksek uyum değeri atanmıştır. Bu durum COMIAS ölçütünün zayıf yönü olarak karşımıza çıkmaktadır.

Çizelge 6.16 Yapı paketi için COMIAS ölçüm sonuçları

Sürümler Ölçüt	Tspell 0.3	Zemberek 0.6.3	Zemberek 0.6.4	Zemberek 2.2.1
COMIAS	0,109	0,229	0,229	0,315

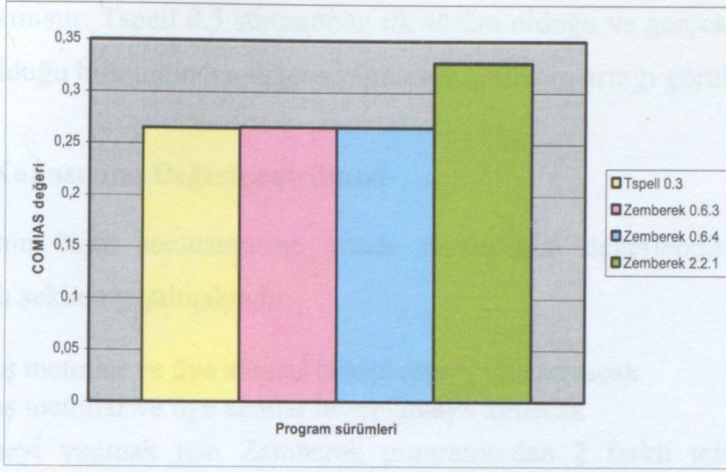


Şekil 6.17 Yapı paketi için elde edilen sonuçların grafiksel gösterimi

Yapı paketi için sürümlerde elde edilen değerlere bakıldığında sürümler arttıkça kalitenin arttığı net bir şekilde görülmektedir (Çizelge 6.16). Sürümlerdeki Yapı paketi için elde edilen değerlerin, Zemberek 0.6.3 ve Zemberek 0.6.4 sürümlerinin gerçek verilerinde bu sürümlerin kalite açısından benzer olduğu bilgisini yansıttığı da görülmektedir.

Çizelge 6.17 İşlemler paketi için COMIAS ölçüm sonuçları

Sürümler Ölçüt	Tspell 0.3	Zemberek 0.6.3	Zemberek 0.6.4	Zemberek 2.2.1
COMIAS	0,265	0,266	0,266	0,329

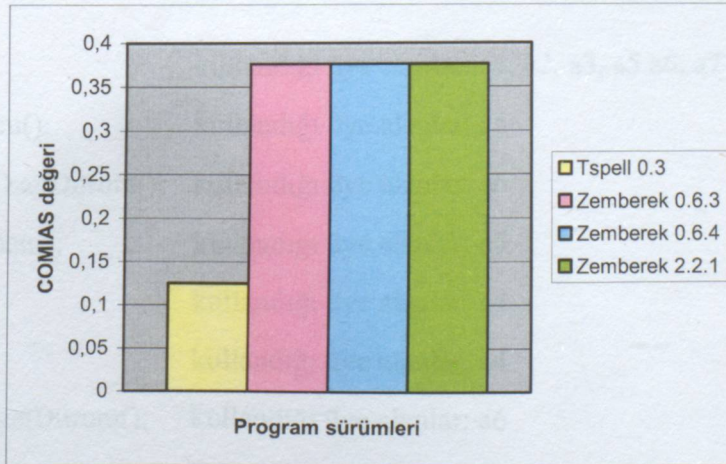


Şekil 6.18 İşlemler paketi için elde edilen sonuçların grafiksel gösterimi

Tspell 0.3 sürümünde bulunan İşlemler paketi için elde edilen değer Zemberek 0.6.3 ve Zemberek 0.6.4 sürümleri için elde edilen değere çok yakın çıkmışken Zemberek 2.2.1 sürümü için elde edilen değer diğer sürümlerden farklılaştığı açıkça görülmektedir (Çizelge 6.17). Bu sonuçlardan yola çıkılarak Zemberek 2.2.1 sürümünde işlemler paketi için genel olarak kalitenin arttığı net bir şekilde görülebilmektedir.

Çizelge 6.18 İstatistik paketi için COMIAS ölçüm sonuçları

Sürümler	Tspell 0.3	Zemberek 0.6.3	Zemberek 0.6.4	Zemberek 2.2.1
Ölçüt				
COMIAS	0,125	0,376	0,376	0,376



Şekil 6.19 İstatistik paketi için elde edilen sonuçların grafiksel gösterimi

Tspell 0.3 sürümünde bulunan İstatistik paketi için elde edilen değer diğer sürümlerden

oldukça farklı çıkmıştır. Tspell 0.3 sürümünün ilk sürüm olduğu ve gerçek verilerle de kalite açısından zayıf olduğu bilindiğinden diğer sürümlerde kalitenin arttığı görülebilmektedir.

6.8.4 Kalıtım Konusunun Değerlendirilmesi

Kalıtım özelliğinin ölçüt sonucunu ne yönde etkilediğini değerlendirebilmek amacıyla ölçümler iki farklı şekilde yapılmaktadır:

- Kalıtım alınmış metotlar ve üye alanlar hesaplamaya alınmayacak
- Kalıtım alınmış metotlar ve üye alanlar hesaplamaya alınacak

Bu değerlendirmeyi yapmak için Zemberek programından 2 farklı temel sınıf ve ilgili türetilmiş sınıflar ele alınmaktadır.

6.8.4.1 Temel Sınıf “KokOzelDurumu” ve ilgili Türetilmiş Sınıfları

Zemberek 0.6.4 sürümündeki sınıflardan temel sınıf olarak *KokOzelDurumu* sınıfı, türetilmiş sınıf olarak *UlamaOzelDurumu*, *YumusamaNK* ve *FiilKokBozulmasi* sınıfları kullanılmaktadır. *KokOzelDurumu* sınıfının uyum değeri COMIAS ölçütü ile 0.01 olarak hesaplanmıştır.

KokOzelDurumu sınıfı üyeleri:

Üye alanlar:

a1: gelebilecekEkler,	a2: sesliEkOzelDurumu	a3: ekEtkiler,	a4: isim
a5: yapıBozucu,	a6: secimlikOzelDurum	a7: ekKisitleyici,	a8: indeks

Metotlar:

m1: init();	kullandığı üye alanlar: a1, a2, a3, a5 a6, a7
m2: isYapiBozucu();	kullandığı üye alanlar: a5
m3: setSecimlikOzelDurum();	kullandığı üye alanlar: a6
m4: setYapiBozucu();	kullandığı üye alanlar: a5
m5: getIsim();	kullandığı üye alanlar: a4
m6: setIsim();	kullandığı üye alanlar: a4
m7: isSecimlikOzelDurum();	kullandığı üye alanlar: a6
m8: isSesliEkOzelDurumu();	kullandığı üye alanlar: a2
m9: setSesliEkOzelDurumu();	kullandığı üye alanlar: a2
m10: getGelebilecekEkler();	kullandığı üye alanlar: a1

m11: setGelebilecekEkler(); kullandığı üye alanlar: a1
 m12: abstract uygula(); kullandığı üye alanlar: -
 m13: uygula(); kullandığı üye alanlar: -
 m14: dusabilirMi(); kullandığı üye alanlar: a1 ve a2
 m15: ozelEkEkle(); kullandığı üye alanlar: a1
 m16: isEkKisitleyici(); kullandığı üye alanlar: a7
 m17: getIndeks(); kullandığı üye alanlar: a8
 m18: setIndeks(); kullandığı üye alanlar: a8
 m19: equals(); kullandığı üye alanlar: a8
 m20: hashCode(); kullandığı üye alanlar: a8

UlamaOzelDurumu sınıfı üyeleri:

Üye alanlar: t_a1: ulanacak

Metotlar:

t_m1: init(); kullandığı üye alanlar: t_a1, a5 çağırdığı metotlar: m1
 t_m2: uygula(); kullandığı üye alanlar: t_a1

YumusamaNK sınıfı üyeleri:

Üye alanlar: t_a1: NK

Metotlar:

t_m1: clinit(); kullandığı üye alanlar: t_a1
 t_m2: init(); kullandığı üye alanlar: a2, a5 çağırdığı metotlar: m1
 t_m3: uygula(); kullandığı üye alanlar: t_a1

FiilKokBozulmasi sınıfı üyeleri:

Üye alanlar: t_a1: ozelKelimeler

Metotlar:

t_m1: clinit(); kullandığı üye alanlar: t_a1 çağırdığı metotlar: t_m2
 t_m2: ozelKelimeBelirle(); kullandığı üye alanlar: t_a1
 t_m3: FiilKokBozulmasi(); kullandığı üye alanlar: - çağırdığı metotlar: m1
 t_m4: uygula(); kullandığı üye alanlar: t_a1

1) Kalıtım alınmış metotlar ve üye alanlar hesaplamaya alınmayacak

UlamaOzelDurumu sınıfı:

- Metot sayısı: $m = 2$, Üye alan sayısı: $a = 1$
- Metot çağırımı: Metotlar arasında metot çağırımı yok.
- Üye alan kullanımı:
 - t_m1 metodu t_a1 üye alanını kullanıyor.
 - t_m2 metodu t_a1 üye alanını kullanıyor.

Çizelge 6.19 *UlamaOzelDurumu* sınıfı için metot çağırım ve üye alan paylaşım durumu (kalıtım yokken)

	Metot çağırım durumu	Üye alan paylaşım durumu
(t_m1, t_m2)	0	1

$$COMIAS = \frac{[(0+1)]}{\left(\frac{2 \times (2-1)}{2} \times 1\right) + \left(\frac{2 \times (2-1)}{2} \times 1\right)} = \frac{1}{2} = 0.5$$

YumusamaNK sınıfı:

- Metot sayısı: $m = 3$, Üye alan sayısı $a = 1$
- Metot çağırımı: Metotlar arasında metot çağırımı yok.
- Üye alan kullanımı:
 - t_m1 metodu t_a1 üye alanını kullanıyor.
 - t_m3 metodu t_a1 üye alanını kullanıyor.

Çizelge 6.20 *YumusamaNK* sınıfı için metot çağırım ve üye alan paylaşım durumu (kalıtım yokken)

	Metot çağırım durumu	Üye alan paylaşım durumu
(t_m1, t_m2)	0	0
(t_m1, t_m3)	0	1
(t_m2, t_m3)	0	0

$$COMIAS = \frac{[(0+0) + (0+1) + (0+0)]}{\left(\frac{3 \times (3-1)}{2} \times 1\right) + \left(\frac{3 \times (3-1)}{2} \times 1\right)} = \frac{1}{6} = 0.16$$

FiilKokBozulmasi sınıfı:

- Metot sayısı: $m = 4$, Üye alan sayısı $a = 1$
- Metot çağırımı: Metotlar arasında metot çağırımı yok.
- Üye alan kullanımı:
 - t_m1 metodu t_a1 üye alanını kullanıyor.
 - t_m2 metodu t_a1 üye alanını kullanıyor.
 - t_m4 metodu t_a1 üye alanını kullanıyor.

Çizelge 6.21 FiilKokBozulmasi sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım yokken)

	Metot çağrım durumu	Üye alan paylaşım durumu		Metot çağrım durumu	Üye alan paylaşım durumu
(t_m1, t_m2)	1	1	(t_m2, t_m3)	0	0
(t_m1, t_m3)	0	0	(t_m2, t_m4)	0	1
(t_m1, t_m4)	0	1	(t_m3, t_m4)	0	0

$$COMIAS = \frac{[(1+1) + (0+0) + (0+1) + (0+0) + (0+1) + (0+0)]}{\left(\frac{4 \times (4-1)}{2} \times 1\right) + \left(\frac{4 \times (4-1)}{2} \times 1\right)} = \frac{4}{12} = 0.33$$

2) Kalıtım alınmış metotlar ve üye alanlar hesaplamaya alınacak

UlamaOzelDurumu sınıfı:

- Temel sınıfın m12: abstract uygula() metodu t_m2 metodu tarafından override edilmiştir. Bu nedenle temel sınıfın m12 metodu hariç diğer metotları ve t_m1, t_m2 metotları hesaplamaya alınacaktır.
 - $m = 19$ (temel sınıftan) + 2 (türetilmiş sınıftan) = 21
- Üye alan sayısı: temel sınıfın tüm üye alanları ve t_a1 hesaplamaya alınacaktır.
 - $a = 8$ (temel sınıftan) + 1 (türetilmiş sınıftan) = 9
- Metot çağırımı:
 - t_m1 metodu m1 metodunu çağırıyor
- Üye alan kullanımı:
 - t_m1 metodu m1 metodunu çağırdığından kullandığı üye alanlar: t_a1, a5, a1, a2, a3, a6, a7.
 - Temel sınıfın metotlarının kullandığı üye alanlar yukarıda belirtilmiştir.

Çizelge 6.22'de belirtilmeyen ilişkiler (Metot çağrım durumu, Üye alan paylaşım durumu) = (0, 0) olan ilişkilerdir.

Çizelge 6.22 UlumaOzelDurumu sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım varken)

	Metot çağrım durumu	Üye alan paylaşım durumu		Metot çağrım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	0	1	(M ₉ , M ₁₄)	0	1
(M ₁ , M ₃)	0	1	(M ₉ , t_m1)	0	1
(M ₁ , M ₄)	0	1	(M ₁₀ , M ₁₁)	0	1
(M ₁ , M ₇)	0	1	(M ₁₀ , M ₁₃)	0	1
(M ₁ , M ₈)	0	1	(M ₁₀ , M ₁₄)	0	1
(M ₁ , M ₉)	0	1	(M ₁₀ , M ₁₅)	0	1
(M ₁ , M ₁₀)	0	1	(M ₁₀ , t_m1)	0	1
(M ₁ , M ₁₁)	0	1	(M ₁₁ , M ₁₄)	0	1
(M ₁ , M ₁₄)	0	2	(M ₁₁ , M ₁₅)	0	1
(M ₁ , M ₁₅)	0	1	(M ₁₁ , t_m1)	0	1
(M ₁ , M ₁₆)	0	1	(M ₁₄ , M ₁₅)	0	1
(M ₁ , t_m1)	1	6	(M ₁₄ , t_m1)	0	2
(M ₂ , M ₄)	0	1	(M ₁₅ , t_m1)	0	1
(M ₂ , t_m1)	0	1	(M _{t_m2} , t_m1)	0	1
(M ₃ , t_m1)	0	1	(M ₁₆ , t_m1)	0	1
(M ₄ , t_m1)	0	1	(M ₁₇ , M ₁₈)	0	1
(M ₅ , M ₆)	0	1	(M ₁₇ , M ₁₉)	0	1
(M ₇ , t_m1)	0	1	(M ₁₇ , M ₂₀)	0	1
(M ₈ , M ₉)	0	1	(M ₁₈ , M ₁₉)	0	1
(M ₈ , M ₁₄)	0	1	(M ₁₈ , M ₂₀)	0	1
(M ₈ , t_m1)	0	1	(M ₁₉ , M ₂₀)	0	1

m1 metodu ilişkileri toplamı: 19

m2 metodu ilişkileri toplamı: 2

m3 metodu ilişkileri toplamı: 2

m4 metodu ilişkileri toplamı: 1

m5 metodu ilişkileri toplamı: 1

m6 metodu ilişkileri toplamı: 0

m7 metodu ilişkileri toplamı: 1

m8 metodu ilişkileri toplamı: 3

m9 metodu ilişkileri toplamı: 2

m10 metodu ilişkileri toplamı: 5

m11 metodu ilişkileri toplamı: 3

m_{t_m2} metodu ilişkileri toplamı: 1

m13 metodu ilişkileri toplamı: 0

m14 metodu ilişkileri toplamı: 3

m15 metodu ilişkileri toplamı: 1

m16 metodu ilişkileri toplamı: 1

m17 metodu ilişkileri toplamı: 3

m18 metodu ilişkileri toplamı: 2

m19 metodu ilişkileri toplamı: 1

m20 metodu ilişkileri toplamı: 0

$$COMIAS = \frac{\left[(19 + 2 + 2 + 1 + 1 + 0 + 1 + 3 + 2 + 5 + 3 + 1 + 0 + 3 + 1 + 1 + 3 + 2 + 1 + 0) \right]}{\left(\frac{21 \times (21 - 1)}{2} \times 1 \right) + \left(\frac{21 \times (21 - 1)}{2} \times 9 \right)} = \frac{51}{2100} = 0.02$$

YumusamaNK sınıfı:

- Temel sınıfın m12: abstract uygula() metodu t_m3 metodu tarafından override edilmiştir. Bu nedenle temel sınıfın m12 metodu hariç diğer metotları ve t_m1, t_m2, t_m3 metotları hesaplamaya alınacaktır.
 - $m = 19$ (temel sınıftan) + 3 (türetilmiş sınıftan) = 22
- Üye alan sayısı: temel sınıfın tüm üye alanları ve t_a1 hesaplamaya alınacaktır.
 - $a = 8$ (temel sınıftan) + 1 (türetilmiş sınıftan) = 9
- Metot çağırımı:
 - t_m2 metodu m1 metodunu çağırıyor
- Üye alan kullanımı:
 - t_m2 metodu m1 metodunu çağırdığından kullandığı üye alanlar: a5, a1, a2, a3, a6, a7.
 - Temel sınıfın metotlarının kullandığı üye alanlar yukarıda belirtilmiştir.

“Çizelge 6.22 UlamaOzelDurumu sınıfı için metot çağırım ve üye alan paylaşım durumu” çizelgesinden UlamaOzelDurumu sınıfındaki t_m1 ve t_m2 metotları ile olan ilişkiler çıkarılarak ve YumusamaNK sınıfındaki ilişkilerin eklenmesiyle (Çizelge 6.23) ölçüt hesaplanmaktadır. Çizelge 6.23’de belirtilmeyen ilişkiler (Metot çağırım durumu, Üye alan paylaşım durumu) = (0, 0) olan ilişkilerdir.

Çizelge 6.23 YumusamaNK sınıfı için metot çağırım ve üye alan paylaşım durumu (eklenen ilişkiler)

	Metot çağırım durumu	Üye alan paylaşım durumu		Metot çağırım durumu	Üye alan paylaşım durumu
(M ₁ , t_m2)	1	6	(M ₁₀ , t_m2)	0	1
(M ₂ , t_m2)	0	1	(M ₁₁ , t_m2)	0	1
(M ₃ , t_m2)	0	1	(t_m3, t_m1)	0	1
(M ₄ , t_m2)	0	1	(M ₁₄ , t_m2)	0	2
(M ₇ , t_m2)	0	1	(M ₁₅ , t_m2)	0	1
(M ₈ , t_m2)	0	1	(M ₁₆ , t_m2)	0	1
(M ₉ , t_m2)	0	1			

m1 metodu ilişkileri toplamı: 19

m2 metodu ilişkileri toplamı: 2

m3 metodu ilişkileri toplamı: 2

m4 metodu ilişkileri toplamı: 1

m5 metodu ilişkileri toplamı: 1

m6 metodu ilişkileri toplamı: 0

m7 metodu ilişkileri toplamı: 1

m8 metodu ilişkileri toplamı: 3

m_t_m3 metodu ilişkileri toplamı: 1

m13 metodu ilişkileri toplamı: 0

m14 metodu ilişkileri toplamı: 3

m15 metodu ilişkileri toplamı: 1

m16 metodu ilişkileri toplamı: 1

m17 metodu ilişkileri toplamı: 3

m18 metodu ilişkileri toplamı: 2

m19 metodu ilişkileri toplamı: 1

m9 metodu ilişkileri toplamı: 2

m20 metodu ilişkileri toplamı: 0

m10 metodu ilişkileri toplamı: 5

m_{t_m1} metodu ilişkileri toplamı: 0

m11 metodu ilişkileri toplamı: 3

$$COMIAS = \frac{[(19 + 2 + 2 + 1 + 1 + 0 + 1 + 3 + 2 + 5 + 3 + 1 + 0 + 3 + 1 + 1 + 3 + 2 + 1 + 0 + 0)]}{\left(\frac{22 \times (22 - 1)}{2} \times 1\right) + \left(\frac{22 \times (22 - 1)}{2} \times 9\right)} = \frac{51}{2310} = 0.02$$

FiilKokBozulmasi sınıfı:

- Temel sınıfın m12: abstract uygula() metodu t_{m4} metodu tarafından override edilmiştir. Bu nedenle temel sınıfın m12 metodu hariç diğer metotları ve t_{m1}, t_{m2}, t_{m3} ve t_{m4} metotları hesaplama alınacaktır.
 - m = 19 (temel sınıftan) + 4 (türetilmiş sınıftan) = 23
- Üye alan sayısı: temel sınıfın tüm üye alanları ve t_{a1} hesaplama alınacaktır.
 - a = 8 (temel sınıftan) + 1 (türetilmiş sınıftan) = 9
- Metot çağırımı:
 - t_{m3} metodu m1 metodunu çağırıyor.
- Üye alan kullanımı:
 - t_{m3} metodu m1 metodunu çağırdığından kullandığı üye alanlar: a5, a1, a2, a3, a6, a7.
 - Temel sınıfın metotlarının kullandığı üye alanlar yukarıda belirtilmiştir.

“Çizelge 6.22 UlamaOzelDurumu sınıfı için metot çağırım ve üye alan paylaşım durumu” çizelgesinden UlamaOzelDurumu sınıfındaki t_{m1} ve t_{m2} metotları ile olan ilişkiler çıkarılarak ve FiilKokBozulmasi sınıfındaki ilişkilerin eklenmesiyle (Çizelge 6.24) ölçüt hesaplanmaktadır. Çizelge 6.24’de belirtilmeyen ilişkiler (Metot çağırım durumu, Üye alan paylaşım durumu) = (0, 0) olan ilişkilerdir.

Çizelge 6.24 FiilKokBozulmasi sınıfı için metot çağırım ve üye alan paylaşım durumu (eklenen ilişkiler)

	Metot çağırım durumu	Üye alan paylaşım durumu		Metot çağırım durumu	Üye alan paylaşım durumu
(M ₁ , t _{m3})	1	6	(M ₁₁ , t _{m3})	0	1
(M ₂ , t _{m3})	0	1	(t _{m4} , t _{m1})	0	1
(M ₃ , t _{m3})	0	1	(t _{m4} , t _{m2})	0	1
(M ₄ , t _{m3})	0	1	(t _{m4} , t _{m3})	0	1
(M ₇ , t _{m3})	0	1	(M ₁₄ , t _{m3})	0	2
(M ₈ , t _{m3})	0	1	(M ₁₅ , t _{m3})	0	1
(M ₉ , t _{m3})	0	1	(M ₁₆ , t _{m2})	0	1
(M ₁₀ , t _{m3})	0	1	(t _{m1} , t _{m2})	1	1

m1 metodu ilişkileri toplamı: 19

m_{t_m4} metodu ilişkileri toplamı: 3

m2 metodu ilişkileri toplamı: 2

m13 metodu ilişkileri toplamı: 0

m3 metodu ilişkileri toplamı: 2

m14 metodu ilişkileri toplamı: 3

m4 metodu ilişkileri toplamı: 1

m15 metodu ilişkileri toplamı: 1

m5 metodu ilişkileri toplamı: 1

m16 metodu ilişkileri toplamı: 1

m6 metodu ilişkileri toplamı: 0

m17 metodu ilişkileri toplamı: 3

m7 metodu ilişkileri toplamı: 1

m18 metodu ilişkileri toplamı: 2

m8 metodu ilişkileri toplamı: 3

m19 metodu ilişkileri toplamı: 1

m9 metodu ilişkileri toplamı: 2

m20 metodu ilişkileri toplamı: 0

m10 metodu ilişkileri toplamı: 5

m_{t_m1} metodu ilişkileri toplamı: 2

m11 metodu ilişkileri toplamı: 3

m_{t_m2} metodu ilişkileri toplamı: 0

$$COMIAS = \frac{[(19+2+2+1+1+0+1+3+2+5+3+3+0+3+1+1+3+2+1+0+2+0)]}{\left(\frac{23 \times (23-1)}{2} \times 1\right) + \left(\frac{23 \times (23-1)}{2} \times 9\right)} = \frac{55}{2530} = 0.02$$

6.8.4.2 Temel Sınıf “TemelRaporlayici” ve ilgili Türetilmiş Sınıfları

Zemberek 0.6.4 sürümündeki sınıflardan temel sınıf olarak *TemelRaporlayici* sınıfı, türetilmiş sınıf olarak *DosyaRaporlayici*, *KonsolRaporlayici* ve *KonsolRaporlayici_yeni* sınıfları kullanılmaktadır.

TemelRaporlayici sınıfı üyeleri:

Üye alanlar: b_al: istatistikler

Metotlar:

b_m1: init();

kullandığı üye alanlar: b_al

b_m2: raporla();

kullandığı üye alanlar: b_al

COMIAS hesaplaması:

Dolayısıyla a = 1 ve m = 2.

Çizelge 6.25 TemelRaporlayici sınıfı için metot çağrım ve üye alan paylaşım durumu

	Metot çağrım durumu	Üye alan paylaşım durumu
(b_m1, b_m2)	0	1

$$COMIAS = \frac{[0+1]}{\left(\frac{2 \times (2-1)}{2} \times 1\right) + \left(\frac{2 \times (2-1)}{2} \times 1\right)} = \frac{1}{2} = 0.5$$

DosyaRaporlayici sınıfı üyeleri:**Üye alanlar:** t_a1: fos**Metotlar:**

t_m1: DosyaRaporlayici(); kullandığı üye alanlar: t_a1

t_m2: raporla(); kullandığı üye alanlar: t_a1

KonsolRaporlayici sınıfı üyeleri:**Üye alanlar:** -**Metotlar:**

t_m1: init(); kullandığı üye alanlar: -

t_m2: raporla(); kullandığı üye alanlar: -

KonsolRaporlayici_yeni sınıfı üyeleri:

Bu sınıf 0'dan farklı ölçüt değeri vermesi amacıyla KonsolRaporlayici sınıfının modifiye edilmiş halidir.

Üye alanlar: t_a1 ve t_a2**Metotlar:**

t_m1: init(); kullandığı üye alanlar: t_a1

t_m2: raporla(); kullandığı üye alanlar: t_a1 ve t_a2

1) Kalıtım alınmış metotlar ve üye alanlar hesaplamaya alınmayacak*DosyaRaporlayici sınıfı:*

- Metot sayısı: $m = 2$, Üye alan sayısı: $a = 1$
- Metot çağırımı: Metotlar arasında metot çağırımı yok.
- Üye alan kullanımı:
 - t_m1 ve t_m2 metotları t_a1 üye alanını kullanıyor.

Çizelge 6.26 DosyaRaporlayici sınıfı için metot çağırımı ve üye alan paylaşım durumu (kalıtım yokken)

	Metot çağırımı durumu	Üye alan paylaşım durumu
(t_m1, t_m2)	0	1

$$COMIAS = \frac{[(0+1)]}{\left(\frac{2 \times (2-1)}{2} \times 1\right) + \left(\frac{2 \times (2-1)}{2} \times 1\right)} = \frac{1}{2} = 0.5$$

KonsolRaporlayici sınıfı:

- Üye alanı olmadığı ve metot çağırımı da bulunmadığı için $COMIAS = 0$ 'dır.

KonsolRaporlayici_yeni sınıfı:

- Metot sayısı: $m = 2$, Üye alan sayısı: $a = 2$
- Metot çağırımı: Metotlar arasında metot çağırımı yok.
- Üye alan kullanımı:
 - t_m1 metodu t_a1 üye alanını kullanıyor.
 - t_m2 metodu t_a1 ve t_a2 üye alanını kullanıyor.

Çizelge 6.27 *KonsolRaporlayici_yeni* sınıfı için metot çağırımı ve üye alan paylaşımı durumu (kalıtım yokken)

	Metot çağırımı durumu	Üye alan paylaşımı durumu
(t_m1, t_m2)	0	1

$$COMIAS = \frac{[(0+1)]}{\left(\frac{2 \times (2-1)}{2} \times 1\right) + \left(\frac{2 \times (2-1)}{2} \times 2\right)} = \frac{1}{3} = 0.33$$

2) Kalıtım alınmış metotlar ve üye alanlar hesaplamaya alınacak

DosyaRaporlayici sınıfı:

- Temel sınıfın b_m2 : raporla() metodu t_m2 metodu tarafından override edilmiştir. Bu nedenle temel sınıfın b_m2 metodu hariç diğer metodu b_m1 ve t_m1, t_m2 metotları hesaplamaya alınacaktır.
 - $m = 1$ (temel sınıftan) + 2 (türetilmiş sınıftan) = 3
- Temel sınıfın üye alanı b_a1 ve türetilmiş sınıfın üye alanı t_a1 hesaplamaya alınacaktır.
 - $a = 1$ (temel sınıftan) + 1 (türetilmiş sınıftan) = 2
- Metot çağırımı:
 - t_m1 metodu b_m1 metodunu çağırıyor.
- Üye alan kullanımı:
 - b_m1 metodu b_a1 üye alanını kullanıyor.
 - t_m1 metodu b_m1 metodunu çağırdığından kullandığı üye alanlar: t_a1, b_a1 .
 - t_m2 metodu t_a1 üye alanını kullanıyor.

Çizelge 6.28 DosyaRaporlayici sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım varken)

	Metot çağrım durumu	Üye alan paylaşım durumu
(t_m1, t_m2)	0	1
(t_m1, b_m1)	1	1
(t_m2, b_m1)	0	0

$$COMIAS = \frac{[(1+2)]}{\left(\frac{3 \times (3-1)}{2} \times 1\right) + \left(\frac{3 \times (3-1)}{2} \times 2\right)} = \frac{3}{9} = 0.33$$

KonsolRaporlayici sınıfı:

- Temel sınıfın b_m2: raporla() metodu t_m2 metodu tarafından override edilmiştir. Bu nedenle temel sınıfın b_m2 metodu hariç diğer metodu b_m1 ve t_m1, t_m2 metotları hesaplamaya alınacaktır.
 - m = 1 (temel sınıftan) + 2 (türetilmiş sınıftan) = 3
- Temel sınıfın üye alanı b_a1 hesaplamaya alınacaktır.
 - a = 1 (temel sınıftan)
- Metot çağırımı:
 - t_m1 metodu b_m1 metodunu çağırıyor.
- Üye alan kullanımı:
 - b_m1 metodu b_a1 üye alanını kullanıyor.
 - t_m1 metodu b_m1 metodunu çağırdığından kullandığı üye alanlar: b_a1.

Çizelge 6.29 KonsolRaporlayici sınıfı için metot çağrım ve üye alan paylaşım durumu (kalıtım varken)

	Metot çağrım durumu	Üye alan paylaşım durumu
(t_m1, t_m2)	0	0
(t_m1, b_m1)	1	1
(t_m2, b_m1)	0	0

$$COMIAS = \frac{[(2)]}{\left(\frac{3 \times (3-1)}{2} \times 1\right) + \left(\frac{3 \times (3-1)}{2} \times 1\right)} = \frac{2}{6} = 0.33$$

KonsolRaporlayici_yeni sınıfı:

- Temel sınıfın b_m2: raporla() metodu t_m2 metodu tarafından override edilmiştir. Bu nedenle temel sınıfın b_m2 metodu hariç diğer metodu b_m1 ve t_m1, t_m2 metotları hesaplamaya alınacaktır.
 - m = 1 (temel sınıftan) + 2 (türetilmiş sınıftan) = 3
- Temel sınıfın üye alanı b_a1 ve türetilmiş sınıfın üye alanı t_a1 ve t_a2 hesaplamaya alınacaktır.
 - a = 1 (temel sınıftan) + 2 (türetilmiş sınıftan) = 3

- Metot çağırımı:
 - t_m1 metodu b_m1 metodunu çağırıyor.
- Üye alan kullanımı:
 - b_m1 metodu b_a1 üye alanını kullanıyor.
 - t_m1 metodu b_m1 metodunu çağırdığından kullandığı üye alanlar: b_a1 ve t_a1.
 - t_m2 metodunun kullandığı üye alanlar: t_a1 ve t_a2.

Çizelge 6.30 KonsolRaporlayici_yeni sınıfı için metot çağırım ve üye alan paylaşım durumu (kalıtım varken)

	Metot çağırım durumu	Üye alan paylaşım durumu
(t_m1, t_m2)	0	1
(t_m1, b_m1)	1	1
(t_m2, b_m1)	0	0

$$COMIAS = \frac{\left[\begin{matrix} (3) \\ \left(\frac{3 \times (3-1)}{2} \times 1 \right) + \left(\frac{3 \times (3-1)}{2} \times 3 \right) \end{matrix} \right]}{12} = \frac{3}{12} = 0.25$$

6.8.4.3 Kalıtım ile ilgili Ölçüm Sonuçlarının İncelenmesi

- Temel sınıfı KokOzelDurumu sınıf (**uyum değeri: 0.01**) olan türetilmiş sınıflar için elde edilen değerler:

Çizelge 6.31 Temel sınıfı KokOzelDurumu olan türetilmiş sınıflar için elde edilen değerler

	Kalıtım göz önüne alınmamışken	Kalıtım göz önüne alınmışken
KokOzelDurumu	0.01	-
UlamaOzelDurumu	0.5	0.02
YumusamaNK	0.16	0.02
FiiilKokBozulmasi	0.33	0.02

- Temel sınıfı TemelRaporlayici sınıf (**uyum değeri: 0.5**) olan türetilmiş sınıflar için elde edilen değerler:

Çizelge 6.32 Temel sınıfı TemelRaporlayici olan türetilmiş sınıflar için elde edilen değerler

	Kalıtım göz önüne alınmamışken	Kalıtım göz önüne alınmışken
TemelRaporlayici	0.5	-
DosyaRaporlayici	0.5	0.33
KonsolRaporlayici	0	0.33
KonsolRaporlayici_yeni	0.33	0.25

Kalıtım ile ilgili elde edilen görüşler:

- Genel olarak kalıtım alınmış metot ve üye alanları hesaplamaya almak uyumun düşük çıkmasına neden olmaktadır.
- Eğer türetilmiş sınıfın metotları, çoğunlukla temel sınıfın üye alanlarını kullanarak işlem yapıyor ve aralarında metot çağrım durumu yok ise kalıtım göz önüne alınmadığı durumda türetilmiş sınıfın uyumu düşük çıkar. Kalıtım göz önüne alındığı durumda temel sınıfın üye alanları türetilmiş sınıfın üye alanları olarak kabul edileceği için üye alan paylaşım durumu sayısı artacak dolayısıyla da uyum daha yüksek çıkacaktır.
- Eğer türetilmiş sınıfın tüm metotları temel sınıfın tüm metotlarını yeniden tanımlıyor (override) ise metot sayısı her iki durumda da değişmeyecektir ve eğer türetilmiş sınıfın metotları temel sınıfın üye alanlarını kullanıyor ise kalıtım göz önüne alındığı durumda elde edilen uyum değeri daha yüksek olacaktır.
- Eğer türetilmiş sınıfın her hangi bir üye alanı yoksa ve metotlar arasında bir çağrım yoksa (**uyum 0 ise**) o zaman kalıtımı hesaba katmak uyumu **arttıracaktır** (eğer türetilmiş sınıfın metotları temel sınıfın üye alanlarını kullanıyor ise), çünkü hali hazırdaki uyum zaten minimumdur. Ancak, eğer türetilmiş sınıfın hesaplanan uyum değeri 0'dan farklı ise o zaman kalıtımı hesaba katmak uyumu bir nebze de olsa **düşürmektedir**.
- Temel sınıfın uyum değeri türetilmiş sınıf ile **aynı** ise kalıtımı hesaba katınca türetilmiş sınıfın uyumu **düşmektedir**.
- Temel sınıfın uyum değeri türetilmiş sınıfinkinden **düşük veya yüksek** ise kalıtımı hesaba katınca türetilmiş sınıfın uyumu **düşmektedir**.
- Bu "**düşürme**" durumu temel sınıfın yüksek uyumlu veya düşük uyumlu olmasından **bağımsız** gelişen bir sonuçtur.
- Kalıtım özelliğini analize almak isteğe bağlı bir durumdur.

6.8.5 Ölçümlerde İlginç Bulunan Sonuçların İncelenmesi

Bu bölümde, Zemberek programın deneysel geçerlemesi esnasında LCOM5 ölçütünün yanlış sonuçlar verdiği ve/veya hesaplanamadığı sınıflar incelenmektedir. İki ayrı bölüm halinde bu durum verilmektedir.

1. LCOM5 yanlış sonuçlar verebiliyor
2. $a = 0$ olan sınıflar için LCOM5 ölçütü hesaplanamıyor ve yanlış sonuçlar verebiliyor

1) LCOM5 yanlış sonuçlar verebiliyor

LCOM5 ölçütünün yanlış sonuçlar verebildiği **İkiliKokOkuyucu** sınıfı üzerinde gösterilmektedir:

a) 0.6.4 sürüm İkiliKokOkuyucu sınıfı:

Üye alanlar: `dis` ve `ozelDurumlar`

Metotlar: `IkiliKokOkuyucu`, `hepsiniOku`, `oku`, `main`

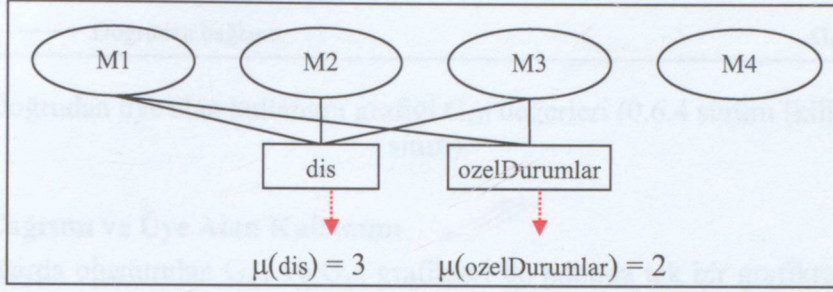
M_1 : `IkiliKokOkuyucu()`, I_1 : {`ozelDurumlar`, `dis`}, Çağrılan metotlar: {-}

M_2 : `hepsiniOku()`, I_2 : {`dis`}, Çağrılan metotlar: {`oku()`}

M₃: oku(), I₃: {ozelDurumlar, dis}, Çağrılan metotlar: {-}

M₄: main(), I₄: {-}, Çağrılan metotlar: {IkiliKokOkuyucu()}

LCOM5 ölçütüne göre uyum hesaplaması:



Şekil 6.20 LCOM5 ölçütüne göre metot-üye alan ilişkisi ve $\mu(A_j)$ değerleri (0.6.4 sürüm İkiliKokOkuyucu sınıfı)

Toplam metot sayısı: $m = 4$,

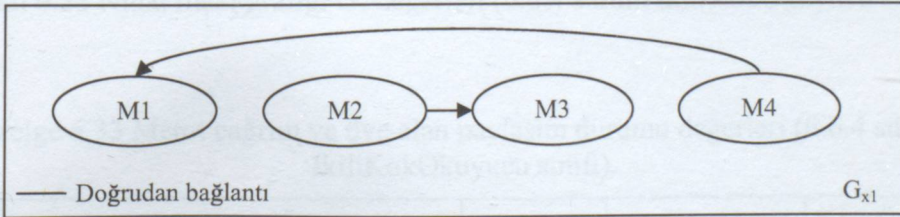
Toplam üye alan sayısı: $a = 2$

$$LCOM5 = \frac{\frac{1}{2}(3+2) - 4}{1-4} = \frac{-3/2}{-3} = 0.5$$

COMIAS ölçütüne göre uyum hesaplaması:

i) Metot Çağrılarının İncelenmesi

Metot çağrıları incelenirken, ilk olarak metotlar arasındaki doğrudan bağlantı ilişkileri incelenir. Eğer bir metot diğerini çağırıyorsa o zaman aralarında çağırıcı taraftan çağrılan tarafa doğru yönlü bir çizgi oluşturulur. Doğrudan bağlantı G_{x1} grafiği Şekil 6.21'de görülebilir.

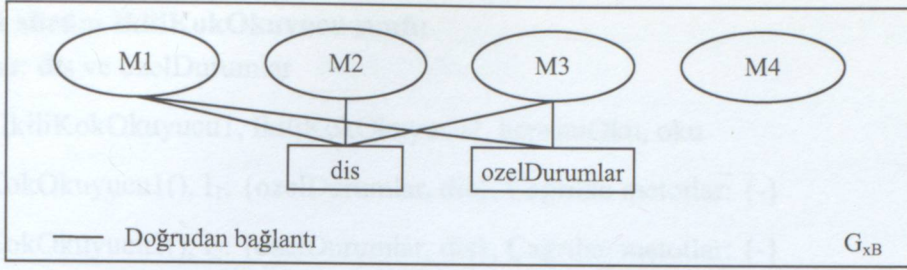


Şekil 6.21 Doğrudan bağlantı grafiği G_{x1} değerleri (0.6.4 sürüm İkiliKokOkuyucu sınıfı)

Dolaylı bağlantı yok. O nedenle G_{x1} grafiği, metot çağrı grafiği olan G_{xA} grafiğidir.

ii) Üye Alan Kullanımlarının İncelenmesi

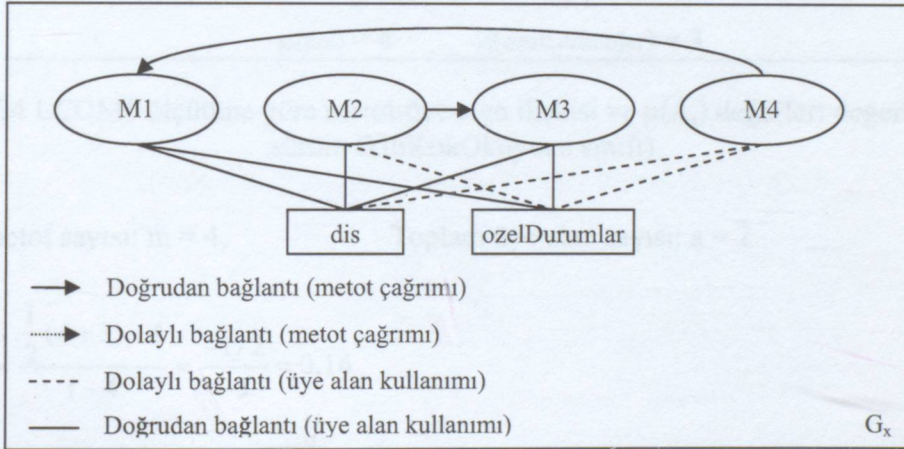
Üye alanların üye metotlar içerisinde kullanımı görebilmek için metot-alan ilişkisini gösteren bir grafik çizilir. Eğer bir metot bir alanı kullanıyorsa o zaman metot ve alan arasında bir çizgi oluşturulur. Şekil 6.22'de bu şekilde elde edilen G_{xB} grafiği görülebilir.



Şekil 6.22 Doğrudan üye alan kullanımı grafiği G_{xB} değerleri (0.6.4 sürüm İkiliKokOkuyucu sınıfı)

iii) Metot Çağırımı ve Üye Alan Kullanımı

Önceki adımlarda oluşturulan G_{xA} ve G_{xB} grafikleri bu adımda tek bir grafikte birleştirilerek G_x grafiği elde edilir. Dolaylı metot çağırımlarının incelenmesi ve dolaylı üye alan kullanımlarının tespit edilmesi ile G_x grafiğinin son hali elde edilir. Önceki bölümlerde verilen örnek grafikler kullanılarak elde edilen G_x grafiği Şekil 6.23'te görülebilir.



Şekil 6.23 Nihai ilişki grafiği G_x değerleri (0.6.4 sürüm İkiliKokOkuyucu sınıfı)

Çizelge 6.33 Metot çağırım ve üye alan paylaşım durumu değerleri (0.6.4 sürüm İkiliKokOkuyucu sınıfı)

	Metot çağırım durumu	Üye alan paylaşım durumu		Metot çağırım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	0	2	(M ₂ , M ₃)	1	2
(M ₁ , M ₃)	0	2	(M ₂ , M ₄)	0	2
(M ₁ , M ₄)	1	2	(M ₃ , M ₄)	0	2

$$COMIAS = \frac{[(0+2) + (0+2) + (1+2) + (1+2) + (0+2) + (0+2)]}{\left(\frac{4 \times (4-1)}{2} \times 1\right) + \left(\frac{4 \times (4-1)}{2} \times 2\right)} = \frac{14}{18} = 0.77$$

b) 2.2.1 sürüm İkiliKokOkuyucu sınıfı:

Üye alanlar: dis ve özelDurumlar

Metotlar: İkiliKokOkuyucu1, İkiliKokOkuyucu2, hepsiniOku, oku

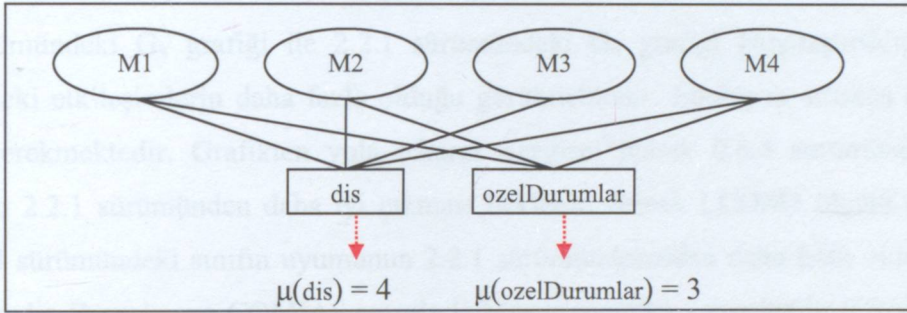
M₁: İkiliKokOkuyucu1(), I₁: {özelDurumlar, dis}, Çağrılan metotlar: {-}

M₂: İkiliKokOkuyucu2(), I₂: {özelDurumlar, dis}, Çağrılan metotlar: {-}

M₃: hepsiniOku(), I₃: {dis}, Çağrılan metotlar: {oku()}

M₄: oku(), I₄: {özelDurumlar, dis}, Çağrılan metotlar: {-}

LCOM5 ölçütüne göre uyum hesaplaması:



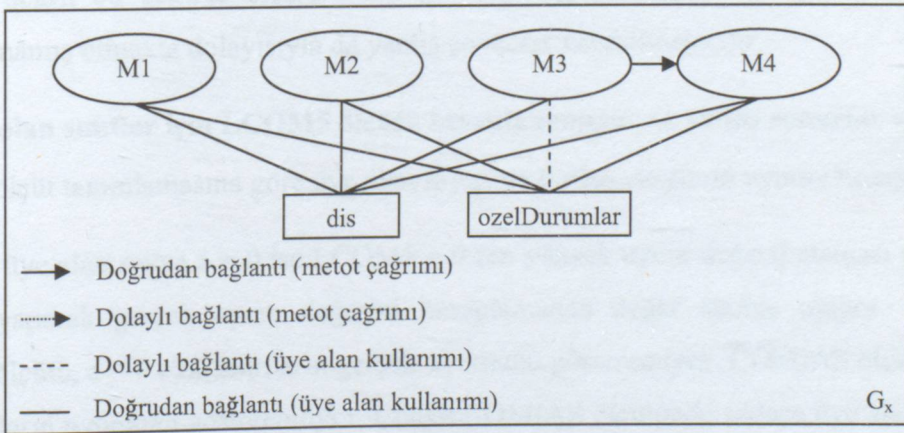
Şekil 6.24 LCOM5 ölçütüne göre metot-üye alan ilişkisi ve $\mu(A_j)$ değerleri (2.2.1 sürüm İkiliKokOkuyucu sınıfı)

Toplam metot sayısı: $m = 4$,

Toplam üye alan sayısı: $a = 2$

$$LCOM5 = \frac{\frac{1}{2}(4+3) - 4}{1-4} = \frac{-1/2}{-3} = 0.16$$

COMIAS ölçütüne göre uyum hesaplaması:



Şekil 6.25 Nihai ilişki grafiği G_x değerleri (2.2.1 sürüm İkiliKokOkuyucu sınıfı)

Çizelge 6.34 Metot çağrım ve üye alan paylaşım durumu değerleri (2.2.1 sürüm İkiliKokOkuyucu sınıfı)

	Metot çağrım durumu	Üye alan paylaşım durumu		Metot çağrım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	0	2	(M ₂ , M ₃)	0	2
(M ₁ , M ₃)	0	2	(M ₂ , M ₄)	0	2
(M ₁ , M ₄)	0	2	(M ₃ , M ₄)	1	2

$$COMIAS = \frac{[(0+2) + (0+2) + (0+2) + (0+2) + (0+2) + (0+2)]}{\left(\frac{4 \times (4-1)}{2} \times 1\right) + \left(\frac{4 \times (4-1)}{2} \times 2\right)} = \frac{13}{18} = 0.72$$

0.6.4 sürümündeki G_x grafiği ile 2.2.1 sürümündeki G_x grafiği karşılaştırıldığında 0.6.4 sürümündeki etkileşimlerin daha fazla olduğu görülmektedir. Etkileşim arttıkça da uyumun artması gerekmektedir. Grafikten yola çıkarak sezgisel olarak 0.6.4 sürümündeki sınıfın uyumunun 2.2.1 sürümünden daha iyi çıkması beklenir. Ancak LCOM5 ölçütü sonuçlarına göre 0.6.4 sürümündeki sınıfın uyumunun 2.2.1 sürümündekinden daha kötü olduğu sonucu görülmektedir. Buna karşın COMIAS sezgilerle benzer sonuçlar vermektedir (Çizelge 6.35).

Çizelge 6.35 İkiliKokOkuyucu sınıfı ölçüm sonuçları

	sürüm 0.6.4	sürüm 2.2.1
LCOM5	0.5	0.16
COMIAS	0.77	0.72

COMIAS ölçütü metot çağrılarını ve dolaylı bağlantıları hesaba kattığı için etkileşim grafiğini doğru bir şekilde oluştururken LCOM5 ölçütü oluşturduğu grafikte uyumu net yakalayamamış olmakta dolayısıyla da yanlış sonuçlar verebilmektedir.

2) $a = 0$ olan sınıflar için LCOM5 ölçütü hesaplanamıyor ve yanlış sonuçlar verebiliyor

LCOM5 ölçüt tanımlamasına göre üye alan sayısı $a = 0$ olan sınıfların uyumu hesaplanamıyor.

Program, üye alan sayısı $a = 0$ ise $LCOM5 = 0$ (en yüksek uyum değeri) ataması yapıyor. Bu atamayı yaparak gerçek uyum değerini hesaplamadan değer atamış oluyor. Dolayısıyla LCOM5 ölçütü, $a = 0$ olan sınıfların gerçek uyumunu gösteremiyor. COMIAS ölçütü ise $a = 0$ olan sınıfların uyumunu gösterebiliyor. Çünkü COMIAS ölçütünde sadece üye alan paylaşımı değil metot çağrıları da göz önüne alınmaktadır.

LCOM5 ölçütünün $a = 0$ olan sınıflar için yanlış sonuçlar verdiği 2 farklı sınıf üzerinde gösterilmektedir:

- a) jarowinkler sınıfı
- b) kokokuyucu sınıfı

a) 0.6.4 sürümündeki jarowinkler sınıfı

Üye alanlar: -

Metotlar: init, benzerlikOrani, jaroBenzerlikOrani, halfLengthOfShorter, commonChars, transpositions, commonPrefixLength

M₁: init(), I₁: {-}, Çağrılan metotlar: {-}

M₂: benzerlikOrani(), I₂: {-}, Çağrılan metotlar: {jaroBenzerlikOrani (), commonPrefixLength}

M₃: jaroBenzerlikOrani(), I₃: {-}, Çağrılan metotlar: {halfLengthOfShorter, commonChars, transpositions}

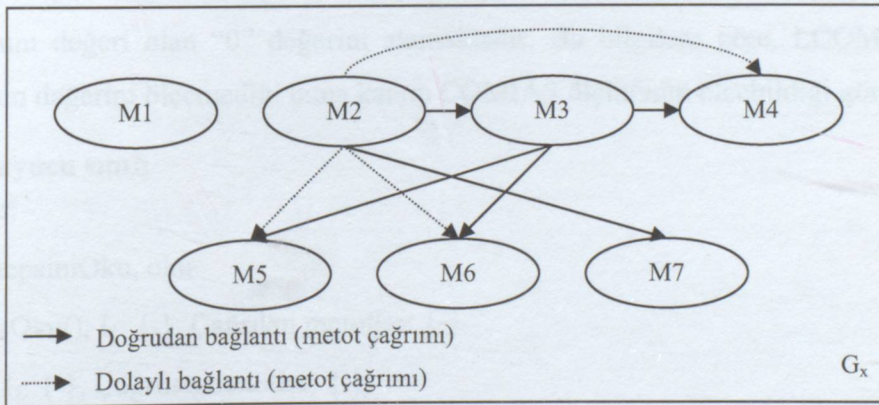
M₄: halfLengthOfShorter(), I₄: {-}, Çağrılan metotlar: {-}

M₅: commonChars(), I₅: {-}, Çağrılan metotlar: {-}

M₆: transpositions(), I₆: {-}, Çağrılan metotlar: {-}

M₇: commonPrefixLength(), I₇: {-}, Çağrılan metotlar: {-}

COMIAS ölçütüne göre uyum hesaplaması:



Şekil 6.26 Nihai ilişki grafiği G_x değerleri (0.6.4 sürüm jarowinkler sınıfı)

Çizelge 6.36 Metot çağrım ve üye alan paylaşım durumu değerleri (0.6.4 sürüm jarowinkler sınıfı)

	Metot çağrım durumu	Üye alan paylaşım durumu		Metot çağrım durumu	Üye alan paylaşım durumu
(M ₁ , M ₂)	0	0	(M ₃ , M ₄)	1	0
(M ₁ , M ₃)	0	0	(M ₃ , M ₅)	1	0
(M ₁ , M ₄)	0	0	(M ₃ , M ₆)	1	0
(M ₁ , M ₅)	0	0	(M ₃ , M ₇)	0	0
(M ₁ , M ₆)	0	0	(M ₄ , M ₅)	0	0
(M ₁ , M ₇)	0	0	(M ₄ , M ₆)	0	0
(M ₂ , M ₃)	1	0	(M ₄ , M ₇)	0	0
(M ₂ , M ₄)	1	0	(M ₅ , M ₆)	0	0
(M ₂ , M ₅)	1	0	(M ₅ , M ₇)	0	0
(M ₂ , M ₆)	1	0	(M ₆ , M ₇)	0	0
(M ₂ , M ₇)	1	0			

$$COMIAS = \frac{[8]}{\left(\frac{7 \times (7-1)}{2} \times 1\right) + \left(\frac{7 \times (7-1)}{2} \times 0\right)} = \frac{8}{21} = 0.38$$

Yukarıdaki hesaplama göre COMIAS ölçütü 0.38 sonucunu verirken, LCOM5 ölçütü bu sınıf için bir uyum değeri hesaplaması yapamamakta ve Program LCOM5 ölçütüne göre en yüksek uyum değeri olan “0” değerini atamaktadır. Bu bilgilere göre, LCOM5 ölçütünün gerçek uyum değerini ölçemediği buna karşın COMIAS ölçütünün ölçebildiği görülmektedir.

b) kokokuyucu sınıfı

Üye alanlar: -

Metotlar: hepsiniOku(), oku

M₁: hepsiniOku(), I₁: {-}, Çağrılan metotlar: {-}

M₂: oku(), I₂: {-}, Çağrılan metotlar: {-}

Program bu sınıf için a = 0 olduğundan hesaplama yapmadan LCOM5 ölçütüne göre en yüksek uyum değeri olan “0” değerini, COMIAS ölçütü ise ölçüm yaparak COMIAS ölçütüne göre en düşük uyum değeri olan “0” değerini atamaktadır. Gerçekte de beklenen uyum değeri en düşük olmalıdır. Çünkü sınıfın yapısı incelendiğinde a = 0’dır ve metotlara bakıldığında metot çağrımı da bulunmamaktadır. Bu bilgilere göre, LCOM5 ölçütünün gerçek uyum değerini ölçemediği üstelik tamamen yanlış bir uyum değeri verdiği buna karşılık COMIAS ölçütünün olması gereken doğru yönde bir uyum değeri verdiği görülmektedir (Çizelge 6.37).

Çizelge 6.37 a = 0 olan sınıflar için ölçüm sonuçları

	jarowinkler	kokokuyucu
LCOM5	0 (hesaplama yapmadan)	0 (hesaplama yapmadan)
COMIAS	0.38	0

6.9 Sınıflandırma

Literatürde bulunan yapısal uyum ölçütlerinin sınıflandırıldığı 5. bölümdeki sınıflandırma kriterlerine göre COMIAS ölçütünün sınıflandırması Çizelge 6.38’de verilmektedir.

Çizelge 6.38 COMIAS ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Alta yatan yaklaşım	Bunge’ın “benzerlik” ontolojisine dayanan Chidamber ve Kemerer’in ölçüt çalışmalarından esinlenerek önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Sezgisel olarak tahmin edilen sonuçlarla benzer sonuçlar veriyor.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt yansıtır.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik ve deneysel geçerlemesi yapılmıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan ve dolaylı üye alan paylaşımı, doğrudan ve dolaylı metot çağırımı (DAS, IAS, DMMR, IMMR (Çizelge 5.2))
10	Özel metotlar	Özel metotlar göz önüne alınmamıştır.
11	Kalıtım	Kalıtım özelliği göz önüne alınmamıştır.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Her hangi bir kalite faktörüne etkisi belirtilmemiştir.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan kullanımı göz önüne alınmıştır. Üye alan paylaşımı göz önüne alınmıştır. Metot çağırımları göz önüne alınmıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmıştır.

7. SONUÇ

Günümüz insanının çevresinde gelişen ve gittikçe artan yazılım sistemlerinde ortaya çıkabilecek bir yazılım hatası, insan hayatının yok oluşu, mali kayıp veya zaman gecikmesi gibi kötü sonuçlar doğurabilecek güce sahiptir. Bu nedenle, bugünün yazılım sistemleri hatasız ve tutarlı çalışabilen sistemler olarak çalışabilmek zorundadırlar. Artan yazılım kalitesi talebi, “kalite” karakteristiğini ürünler arasında önemli bir ayırıcı faktör olarak karşımıza çıkarmaktadır.

Yazılım uyumu, yazılımın içsel kalite özellikleri arasında bağlaşım ile birlikte önemli bir yere sahiptir ve bir modülün elemanlarının birbirine aitliğinin derecesi olarak tanımlanabilir. İyi tasarlanmış bir sistemde uyumun yüksek ve bağlaşımın düşük olması beklenir. Uyum eksikliği, sınıfların iki veya daha fazla alt sınıfa ayrılmalrı gerekliliğini gösterir. Düşük uyum karmaşıklığı arttırarak geliştirme sürecindeki hataların olma olasılığını arttırır. Literatürde önerilen uyum ölçütlerinden hiçbirisi bir standart olarak yaygın kabul görmemiştir (Poshyvanyk ve Marcus, 2005). Bu duruma yol açan etkenlerden biri de önerilen ölçütler ile ilgili daha fazla deneysel gerçekleştirme çalışmasına gereksinim duyulmasıdır (Briand vd., 1999).

Bu tez çalışmasında yapılan çalışmalar şunlardır:

- Yazılım Uyum Ölçütlerinin kullanım amaçları anlatılmıştır
- NY sistemlerde uyumu ölçmek için kullanılan yaklaşımlar anlatılmıştır
- Literatürde önerilen yapısal uyum ölçütleri anlatılmış ve örnekler üzerinde her bir ölçütün uygulaması yapılarak değerlendirilmeler verilmiştir
- Yapılan literatür araştırması ile elde edilen yapısal uyum ölçütlerinin incelenmesi sonucu ölçütlerin sınıflandırılmasının yararlı olacağı düşünülmüş ve bu kapsamda var olan bazı sınıflandırma kriterlerine ek olarak yeni kriterler tanımlanmıştır
- Mevcut yapısal uyum ölçütler belirtilen sınıflandırma kriterlerine göre değerlendirilmiştir
- Yapısal uyum ölçütlerinden yola çıkılarak temelde LCOM ölçütlerinin eksikliklerini adresleyen yeni bir ölçüt COMIAS önerilmiştir. COMIAS ölçütü ortaya konan görüşler ve hipotezlerle açıklanmaya çalışılmıştır. Yapılan örneklerle COMIAS, LCOM ölçütleri ile karşılaştırılmıştır. LCOM ölçütlerinin sezgilerle uyuşmayan sonuçlar verdiği durumlarda COMIAS ölçütünün uyuşan sonuçlar verdiği görülmüştür.
- Ölçütün Weyuker özellik kümesi kullanılarak teorik geçerlemesi yapılmıştır.
- Teorik geçerlemenin yanı sıra Zemberek adı verilen açık kaynak kodlu projenin çeşitli sürümleri kullanılarak COMIAS ölçütü deneysel olarak da geçerlenmiştir.
- Deneysel geçerleme amacıyla CKmetrics.jar dosyası oluşturulmuştur. Bu dosya içerisindeki CKMetricsGUI.java, ClassVisitor.java ve ClassMetrics.java sınıflarına kod eklentileri yapılmıştır.
- Zemberek programından örnek sınıflar alınarak kalıtım konusu değerlendirilmiş ve genel olarak kalıtım alınmış metot ve üye alanları hesaplamaya almanın uyumun düşük çıkmasına neden olduğu sonucuna varılmıştır
- Zemberek programının deneysel geçerlemesi esnasında LCOM5 ölçütünün yanlış sonuçlar verdiği örnekler vurgulanmıştır

- Zemberek programının deneysel geçerlemesi esnasında $a = 0$ olan sınıflar için LCOM5 ölçütünün hesaplanmadığı ve hatta yanlış yönde sonuçlar verdiği örnekler vurgulanmıştır
- COMIAS ölçütü belirtilen sınıflandırma kriterlerine göre değerlendirilmiştir

Bu tez kapsamında önerilen COMIAS ölçütü, normalize oluşu, dolaylı ilişkileri göz önünde bulundurması ve sezgilerle paralel sonuçlar vermesi özellikleri ile ümit vericidir. Ancak COMIAS kalıtım ile aktarılan metotları ve özel metotları dikkate almamakta, metot sayısı 0/1 olan sınıflara en yüksek uyum değeri atadığı için yanıltıcı sonuç verebilmektedir. Ölçüt bu yönlerden geliştirilmeye açıktır.

KAYNAKLAR

- Abubakar, A., "Implementation and Validation of Object-Oriented Design-Level Cohesion Metrics", King Fahd University of Petroleum & Minerals Dhahran 31261, Saudi Arabia, January 2005
- Abubakar, A., AlGhamdi, J. ve Ahmed M., "Can Cohesion Predict Fault Density", IEEE International Conference on Computer Systems and Applications, Vols 1-3., 2006.
- Allen, E. B., Khoshgoftaar, T. M., ve Chen, Y., "Measuring Coupling and Cohesion of Software Modules: an Information-theory Approach", in Proceedings of 7th International Software Metrics Symposium (METRICS'01), April 4-6 2001, pp. 124-134.
- Aman H., Yamasaki K., Yamada H. ve Noda M., "A Proposal of Class Cohesion Metrics Using Sizes of Cohesive Parts", Knowledge-based Software Engineering, T. welzer et al.(Eds.), pp102-107, IOS Press, Sept. 2002.
- Badri, L. ve Badri, M. "A Proposal of a New Class Cohesion Criterion: An Empirical Study". Journal of Object Technology, 3(4), Apr. 2004. Special issue: TOOLS USA 2003.
- Baldassari, B., Robach, C. ve du Bosquet, L., "Early Metrics for Object Oriented Designs", IEEE Testability Assessment, 2004. IWOTA 2004. Proceedings. First International Workshop p.62-69, 2004.
- Bansiya, J. ve Davis, C. G., "A Hierarchical Model for Object-Oriented Design Quality Assessment", IEEE Transactions on Software Engineering, vol. 28, no. 1, January 2002, pp. 4-17.
- Bieman, J. ve Kang, B-K., "Cohesion and Reuse in An Object-Oriented System", in Proceedings of ACM Symposium on Software Reusability (SSR'95), April 1995, pp. 259-262.
- Bonja, C. ve Kidanmariam, E., "Metrics for Class Cohesion and Similarity between Methods", ACM SE'06, March 10-12, 2006, Melbourne, Florida, USA.
- Briand, L.C., Daly, J.C. ve Wüst J. "A unified Framework for Cohesion Measurement in Object-Oriented Systems", Technical Report ISERN-97-05. Kaiserlautern:Germany: Fraunhofer Institute for Experimental Software Engineering, 1997.
- Briand, L.C., Daly, J.C. ve Wüst J. "A unified Framework for Cohesion Measurement in Object-Oriented Systems", Empirical Software Eng.; An Int'l J., vol.3, no. 1, pp. 65-117, 1998a.
- Briand, L. C., Daly, J. W., Porter, V., ve Wüst, J., "A Comprehensive Empirical Validation of Design Measures for Object-Oriented Systems", in Proc. of International Software Metrics Symposium, Bethesda, MD, Nov. 21 1998b, pp. 43-53.
- Briand, L. C., Daly, J. W., ve Wüst, J. K., "A Unified Framework for Coupling Measurement in Object-Oriented Systems." IEEE Transactions on Software Engineering, vol. 25, no. 1, January/February 1999.
- Briand L., Morrasca S., and Basili V., "Defining and Validating Measures for Object-Based High-Level Design" IEEE Transactions On Software Engineering, vol. 25, no. 5, September/October 1999.
- Briand, L. C., Wüst, J., Daly, J. W., ve Porter, V. D., "Exploring the Relationship between Design Measures and Software Quality in Object-Oriented Systems", Journal of Systems and

Software, vol. 51, no. 3, May 2000, pp. 245-273.

Brito e Abreu, F. ve Goulao, M., "Coupling and Cohesion as Modularization Drivers: Are we being overpersuaded?" in Proceedings of 5th European Conference on Software Maintenance and Reengineering, 2001, pp. 47-57.

Bunge, M., Treatise on Basic Philosophy: Ontology I : The Furniture of the World. Boston: Riedel, 1977.

Chae, H. S. ve Kwon, Y. R., "A Cohesion Measure for Classes in Object-Oriented Systems", : Software Metrics Symposium, 1998. Metrics 1998. Proceedings. Fifth International.

Chae, H. S., Kwon, Y. R. ve Bae, D. H., "A Cohesion Measure for Object-Oriented Classes" Softw. Pract. Exper., Vol. 30, No. 12. (2000), pp. 1405-1431.

Chidamber, S. R. ve Kemerer, C. F., "Towards a Metrics Suite for Object Oriented Design", in Proceedings of OOPSLA'91, 1991, pp. 197-211.

Chidamber, S. R. ve Kemerer, C. F., "A Metrics Suite for Object Oriented Design", IEEE Transactions on Software Engineering, vol. 20, no. 6, 1994, pp. 476-493.

Cho, E. S., Kim, C. J., Kim, D. D., ve Rhew, S. Y., "Static and Dynamic Metrics for Effective Object Clustering", in Proceedings of Asia Pacific International Conference on Software Engineering, 1998, pp. 78 - 85.

Counsell, S., ve Swift, S., "The Interpretation and Utility of Three Cohesion Metrics for Object-Oriented Design", ACM Transactions on Software Engineering and Methodology, Vol. 15, No. 2, April 2006, Pages 123-149.

Ebert, C. ve Dumke, R., "Software Measurement: Establish - Extract - Evaluate - Execute", Springer, 2007.

El-Emam, K., "Object-Oriented Metrics: A Review of Theory and Practice", in Advances in software engineering, Springer-Verlag, New York, 2002, pp. 23-50.

Ertemel O., Selçuk Y.E., ve Kalıpsız O., "Nesneye Yönelik Sistemler İçin Bir Uyum Ölçütü Önerisi: COMIAS", 4. Ulusal Yazılım Mühendisliği Sempozyumu, 2009a.

Ertemel O., Selçuk Y.E., ve Kalıpsız O., "A Cohesion Metric Proposal For Object-Oriented Systems: COMIAS", 13. "World Scientific and Engineering Academy and Society International Conference on Computers", 2009b

Etzkorn, L. H. ve Davis, C. G., "Automatically Identifying Reusable OO Legacy Code", IEEE Computer, vol. 30, no. 10, October 1997, pp. 66-72.

Etzkorn L. ve Delugach H., "Towards a Semantic Metrics Suite for Object-Oriented Design", IEEE TOOLS (34) 2000: 71-80.

Fowler, M., Beck, K., Brant, J. and Opdyke, W.: Refactoring: Improving the Design of Existing Code, Addison-Wesley, 1999

Gui, G. ve Scott, P.D., "Coupling and Cohesion Measures for Evaluation of Component Reusability", MSR'06, May 22-23, 2006, Shanghai, China.

Hitz, M. ve Montazeri, B., "Measuring Coupling and Cohesion in Object-Oriented Systems", Proceedings of International Symposium on Applied Corporate Computing. (Monterrey, Mexico, 1995).

Jarallah, A., Wasiq, M. ve Ahmed, M. A., "Principle and Metrics for Cohesion-Based Object-Oriented Component Assessment". Confidential Draft Copy, 2001.

Kramer, S. ve Kaindl, H., "Coupling and Cohesion Metrics for Knowledge-Based Systems Using Frames and Rules", *ACM Trans. on Soft. Engineering and Methodology (TOSEM)*, vol. 13, no. 3, July 2004, pp. 332-358.

Kriz, J., *Facts and Artifacts in Social Science: An Epistemological and Methodological Analysis of Empirical Social Science Techniques*. McGraw Hill, 1988.

Lee, J. K., Jung, S. J., Kim, S. D., Jang, W. H., ve Ham, D. H., "Component Identification Method with Coupling and Cohesion", in *Proc. of 8th Asia-Pacific Software Engineering Conference (APSEC'01)*, Dec. 2001, pp. 79-86.

Lieberherr, K., Holland, I., ve Riel, A., "Object Oriented Programming: An Objective Sense of Style," in *Third Annu. ACM Conf: Object Oriented Prog. Syst., Lung. and Applicat. (OOPSLA)*, 1988, pp. 323-334.

Maletic, J. I. ve Marcus, A., "Supporting Program Comprehension Using Semantic and Structural Information", in *Proceedings of 23rd International Conference on Software Engineering*, Toronto, Canada, May 12-19 2001, pp. 103-112.

Meyers, T. M. ve Binkley, D., "Slice-based Cohesion Metrics and Software Intervention", in *Proceedings of 11th Working Conference on Reverse Engineering (WCRE'04)*, Nov. 8-12 2004, pp. 256-265.

Montes de Oca, C. ve Carver, D. L., "Identification of Data Cohesive Subsystems Using Data Mining Techniques", in *Proceedings of International Conference on Software Maintenance (ICSM'98)*, November 1998, pp. p. 16-23.

Nil Seri Bilgisayar Projesi "Nesneye Yönelik Programlamada Kalite Ölçütlerinin Hesaplanması", YTÜ, İstanbul 2008.

Quah, T.-S. ve Thwin, M. M. T., "Application of Neural Networks for Software Quality Prediction Using Object-Oriented Metrics", in *Proceedings of International Conference on Software Maintenance*, September 22-26 2003, pp. 116-125.

Poshyvanyk, D. ve Marcus, A., "The Conceptual COhesion of Classes". 21st IEEE International Conference on Software Maintenance (ICSM'05)1063-6773/05 \$20.00 © 2005 IEEE.

Stevens, W., Myers, G., ve Constantine, L., "Structured Design," *IBM Systems J.*, vol. 13, no. 2, pp. 115-139, 1974.

Weyuker, E. J., "Evaluating Software Complexity Measures", *IEEE Transactions On Software Engineering*, vol.14. no. 9. September 1988.

Zhou, Y., Xu, B., Zhao, J., and Yang, H., "ICBMC: An Improved Cohesion Measure for Classes", in *Proceedings of International Conference on Software Maintenance*, October 3-6 2002, pp. 44-53.

EKLER

Ek 1 Ölçütlerin sınıflandırılması

No	Ölçütlerin sınıflandırılması	Ölçütler
1	Genel değerlendirme	Genel olarak "benzerlik" kavramına dayanarak, ölçütlerin bir arada değerlendirilmesi ve "benzerlik" kavramına dayanarak değerlendirilmesi.
2	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
3	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
4	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
5	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
6	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
7	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
8	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
9	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
10	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
11	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
12	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
13	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
14	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
15	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.
16	Yükseklik	Yükseklik kavramına dayanarak değerlendirilmesi.

Ek 1 Ölçütlerin sınıflandırılması

Çizelge Ek1.1 LCOM1 ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'in "benzerlik" ontolojisine dayanır. Uyum, bu ontolojiye göre "metotların benzerlik derecesi" şeklinde tanımlanıyor.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Doğru bir şekilde uyum özelliğini yansıtmıyor sadece bir fikir veriyor. Yanlış veya belirsiz sonuçlar verebiliyor. Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor. Sadece 1 metoda sahip sınıf için LCOM1 uygulanamamaktadır.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt bazen yansıtmıyor.
6	Normalizasyon	Normalize edilmemiştir.
7	Geçerleme	Teorik geçerlemesi yapılmıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı (DAS (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Yazılımın bakımı esnasında yardımcı olması amacıyla ölçütlere ihtiyaç olduğu belirtilmiştir. Bu ölçüt hata eğilimliliğine etkisi yönünden incelenmiştir (Briand vd., 2000). Bir sınıfın uyumu düştükçe hata eğilimliliğinin artabilme ihtimali konusunda bir gösterge olabileceği görülmüştür. LCOM1 ölçütü ile yazılım sistemindeki hata sayısı arasında bir ilişki var mı diye yapılan başka bir çalışmada hata sayısı ile ölçüt arasında anlamlı bir ilişki bulunamamıştır (Abubakar vd., 2006).
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağrılarını göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.2 LCOM2 ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanır. Uyum, bu ontolojiye göre "metotların benzerlik derecesi" şeklinde tanımlanıyor.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamiyle bulunur.
4	Sağlamlık/Geçerlilik	Doğru bir şekilde uyum özelliğini yansıtmıyor sadece bir fikir veriyor. Yanlış veya belirsiz sonuçlar verebiliyor. Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor. Sadece 1 metoda sahip sınıf için LCOM1 uygulanamamaktadır.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt bazen yansıtıyor.
6	Normalizasyon	Normalize edilmemiştir.
7	Geçerleme	Teorik ve deneysel geçerlemesi yapılmıştır.
8	Yorumlama	Yüksek LCOM değerinin sınıf tarafından sağlanan fonksiyonellikte farklılığı gösterdiği belirtilmiştir. Ölçüt, farklı amaçlar gerçekleştiren sınıfları belirlemede kullanılabilir diye belirtilmiştir. Yüksek LCOM değerine sahip sınıfların daha fazla hata eğilimliliği ve daha zor test edilebilir olduğunu ve iki veya daha fazla sınıfa ayrılmaları gerektiği vurgulanmıştır.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı (DAS (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Artan tekrar kullanım durumu ve test kolaylığı gibi kalite faktörleri hakkında bilgi vereceği belirtilmiştir. Bu ölçüt hata eğilimliliğine etkisi yönünden de incelenmiştir (Briand vd., 2000). İyi tahmin edici olmadığı görüşüne varılmıştır. LCOM2 ölçütü ile yazılım sistemindeki hata sayısı arasında bir ilişki var mı diye yapılan başka bir çalışmada hata sayısı ile ölçüt arasında anlamlı bir ilişki bulunamamıştır (Abubakar vd., 2006).
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.3 LCOM3 ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan LCOM2 ölçütüne dayanır. LCOM2 ölçütünün anomaliliğini çözmek amacıyla önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Doğru bir şekilde uyum özelliğini yansıtmıyor sadece bir fikir veriyor. Yanlış veya belirsiz sonuçlar verebiliyor. Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmemiştir.
7	Geçerleme	Teorik ve deneysel geçerlemesi yapılmamıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı (DAS (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Bu ölçüt hata eğilimliliğine etkisi yönünden incelenmiştir (Briand vd., 2000). Bir sınıfın uyumu düştükçe hata eğilimliliğinin artabilme ihtimali konusunda bir gösterge olabileceği görülmüştür. LCOM3 ölçütü ile yazılım sistemindeki hata sayısı arasında bir ilişki var mı diye yapılan başka bir çalışmada hata sayısı ile ölçüt arasında anlamlı bir ilişki bulunamamıştır (Abubakar vd., 2006).
14	Hesaplama Kolaylığı	Veri toplama zor, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.4 LCOM4 ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan LCOM3 ölçütüne dayanır. LCOM3 ölçütünün anomaliliğini çözmek amacıyla önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamiyle bulunur.
4	Sağlamlık/Geçerlilik	Doğru bir şekilde uyum özelliğini yansıtmıyor sadece bir fikir veriyor. Yanlış veya belirsiz sonuçlar verebiliyor. Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor. Yönsüz bir grafikte 1 tane bağlı bileşen bulunduğu durumda her zaman LCOM4 =1 değeri elde ediliyor. LCOM4 değeri 1 çıkan sonuçlarda farklı uyum özelliği taşımak mümkün.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmemiştir.
7	Geçerleme	Teorik ve deneysel geçerlemesi yapılmamıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı ve doğrudan metot-metot çağırımı (DAS ve DMMR (Çizelge 5.2))
10	Özel metotlar	Erişim metotlarının göz önüne alınması gerektiği belirtilmiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Bu ölçüt hata eğilimliliğine etkisi yönünden incelenmiştir (Briand vd., 2000). Bir sınıfın uyumu düştükçe hata eğilimliliğinin artabilme ihtimali konusunda bir gösterge olabileceği görülmüştür. LCOM4 ölçütü ile yazılım sistemindeki hata sayısı arasında bir ilişki var mı diye yapılan başka bir çalışmada hata sayısı ile ölçüt arasında anlamlı bir ilişki bulunamamıştır (Abubakar vd., 2006).
14	Hesaplama Kolaylığı	Veri toplama zor, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağırımları göz önüne alınmıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.5 Co ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan LCOM4 ölçütünün bir varyasyonudur. LCOM4 =1 değerine sahip sınıfların uyum özelliğini ayırt etmek üzere bağlı bileşenin kenar sayısı hesaba katılarak Co ölçütü önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Doğru bir şekilde uyum özelliğini yansıtmıyor sadece bir fikir veriyor. Yanlış veya belirsiz sonuçlar verebiliyor. Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor. Co ölçütü bağlı bileşen 1 tane olduğu durumlarda uygulanabilen bir ölçüttür.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik ve deneysel geçerlemesi yapılmamıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı ve doğrudan metot-metot çağırımı (DAS ve DMMR (Çizelge 5.2))
10	Özel metotlar	Erişim metotlarının göz önüne alınması gerektiği belirtilmiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Bu ölçüt hata eğilimliliğine etkisi yönünden incelenmiştir (Briand vd., 2000). Bir sınıfın uyumu düştükçe hata eğilimliliğinin artabilme ihtimali konusunda bir gösterge olabileceği görülmüştür.
14	Hesaplama Kolaylığı	Veri toplama zor, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağırımları göz önüne alınmıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.6 TCC ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan Chidamber ve Kemerer'in ölçüt çalışmalarından esinlenerek önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Doğru bir şekilde uyum özelliğini yansıtmıyor sadece bir fikir veriyor. Yanlış veya belirsiz sonuçlar verebiliyor. Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor. Eğer ölçülen sınıfta hiç metot yoksa veya sadece 1 tane metot varsa o zaman ölçüt sonsuz bir değere çıkıyor. İyi bir ölçüt her zaman mantıklı sonuçlar veren ve böyle sonuçlar alma ihtimali olmayan ölçüttür.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik ve deneysel geçerlemesi yapılmamıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı (DAS (Çizelge 5.2))
10	Özel metotlar	Yapılandırıcı ve sonlandırıcı metotları hesaplamaadan çıkarmayı önermişlerdir.
11	Kalıtım	Bieman ve Kang bir sınıfın uyumunu incelemek için 3 seçenek önermişlerdir (Bieman ve Kang, 1995): Kalıtım alınmış metot ve üye alanları analizden çıkarmak Kalıtım alınmış metot ve üye alanları analize almak Kalıtım alınmış metotları analizden çıkarmak üye alanları analize almak
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Ölçüt sınıf uyumu ve tekrar kullanım için deneysel olarak değerlendirilmiştir. Çalışmalar sonucu görülmüş ki, bir sınıf ne kadar sık Kalıtım yoluyla tekrar kullanılıyorsa o kadar az uyuma sahiptir. Bu ölçüt hata eğilimliliğine etkisi yönünden de incelenmiştir (Briand vd., 2000). Bir sınıfın uyumu düştükçe hata eğilimliliğinin artabilme ihtimali konusunda bir gösterge olabileceği görülmüştür. TCC ölçütü ile yazılım sistemindeki hata sayısı arasında bir ilişki var mı diye yapılan başka bir çalışmada hata sayısı ile ölçüt arasında anlamlı bir ilişki bulunamamıştır (Abubakar vd., 2006).
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.7 LCC ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan Chidamber ve Kemerer'in ölçüt çalışmalarından esinlenerek önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Doğru bir şekilde uyum özelliğini yansıtmıyor sadece bir fikir veriyor. Yanlış veya belirsiz sonuçlar verebiliyor. Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor. Eğer ölçülen sınıfta hiç metot yoksa veya sadece 1 tane metot varsa o zaman ölçüt sonsuz bir değere çıkıyor. İyi bir ölçüt her zaman mantıklı sonuçlar veren ve böyle sonuçlar alma ihtimali olmayan ölçüttür.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik ve deneysel geçerlemesi yapılmamıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan ve Dolaylı üye alan paylaşımı (DAS ve IAS (Çizelge 5.2))
10	Özel metotlar	Yapılandırıcı ve sonlandırıcı metotları hesaplamadan çıkarmayı önermişlerdir.
11	Kalıtım	Bieman ve Kang bir sınıfın uyumunu incelemek için 3 seçenek önermişlerdir (Bieman ve Kang, 1995): Kalıtım alınmış metot ve üye alanları analizden çıkarmak Kalıtım alınmış metot ve üye alanları analize almak Kalıtım alınmış metotları analizden çıkarmak üye alanları analize almak
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Ölçüt sınıf uyumu ve tekrar kullanım için deneysel olarak değerlendirilmiştir. Çalışmalar sonucu görülmüş ki, bir sınıf ne kadar sık kalıtım yoluyla tekrar kullanılıyorsa o kadar az uyuma sahiptir. Bu ölçüt hata eğilimliliğine etkisi yönünden de incelenmiştir (Briand vd., 2000). Bir sınıfın uyumu düştükçe hata eğilimliliğinin artabilme ihtimali konusunda bir gösterge olabileceği görülmüştür. LCC ölçütü ile yazılım sistemindeki hata sayısı arasında bir ilişki var mı diye yapılan başka bir çalışmada hata sayısı ile ölçüt arasında yakın bir ilişki bulunmuştur (Abubakar vd., 2006).
14	Hesaplama Kolaylığı	Veri toplama zor, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağrımları, dolaylı yoldan üye alan kullanımına bakmak amacıyla kullanılmış, metotların birbirini çağırma anlamında metot ilişkisi göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Metot çağrımları yoluyla olan dolaylı üye alan paylaşımı göz önüne alınmıştır.

Çizelge Ek1.8 LCOM5 ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'in "benzerlik" ontolojisine dayanan Chidamber ve Kemerer'in ölçüt çalışmalarından esinlenerek önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Sezgisel olarak tahmin edilen sonuçlarla benzer sonuçlar veriyor. Kısmen normalize edilmiştir (Varsayımlar yapıldığı durumlarda). Formülü incelediğimizde bazı noktalarda LCOM5 ölçütünün istenmeyen sonuçlar verebileceğini görüyoruz. Şöyle ki, eğer ölçülen sınıfta hiç üye alan yoksa veya sadece 1 tane metot varsa o zaman ölçüt sonsuz bir değere çıkıyor. İyi bir ölçüt her zaman mantıklı sonuçlar veren ve böyle sonuçlar alma ihtimali olmayan ölçüttür.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt bazen yansıtıyor.
6	Normalizasyon	Kısmen normalize edilmiştir.
7	Geçerleme	Teorik ve deneysel geçerlemesi yapılmamıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Metot üye alan referanslaması (MAR (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Bu ölçüt hata eğilimliliğine etkisi yönünden incelenmiştir (Briand vd., 2000). Bir sınıfın uyumu düştükçe hata eğilimliliğinin artabilme ihtimali konusunda bir gösterge olabileceği görülmüştür. LCOM5 ölçütü ile yazılım sistemindeki hata sayısı arasında bir ilişki var mı diye yapılan başka bir çalışmada hata sayısı ile ölçüt arasında yakın bir ilişki bulunamamıştır (Abubakar vd., 2006).
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan kullanımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.9 ICH ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Alta yatan yaklaşım	Bir sınıfta metot çağrımları boyunca oluşan bilgi akışına dayalı bir uyum ölçüt kümesi olarak önerilmiştir.
2	Tanesellik	Metot, sınıf ve sınıf kümesi seviyesinde uyum ölçümü yapan ölçütler tanımlanmıştır.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Eğer 2 ilişkisiz fakat yüksek uyuma özelliğe sahip sınıf tek bir sınıfta bir araya getirilirse, sınıfın uyumu her iki sınıfın ayrı ayrı uyum ölçümlerinin toplamı olacaktır. Bu da birleştirilmiş sınıfın her iki sınıftan daha yüksek uyuma sahip olmasına neden olur. Bu sonuç beklenmeyen bir sonuçtur. Birleştirilmiş sınıfın bir objesi 2 ayrı anlam kavramlarını temsil edeceği için daha az uyumlu olmalıdır. Bu sonuç bize ICH ölçütünün pek de uyum ölçütü olmadığını gösteriyor. Yanlış veya belirsiz sonuçlar verebiliyor. Sezgilerle paralellik gösteremeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmemiştir.
7	Geçerleme	Teorik geçerlemesi yapılmıştır. Deneysel geçerlemesi yapılmış ve uyum için uygun bir ölçüt olmadığı sonucuna varılmıştır. Karmaşıklık ölçütü özelliği taşıyor diye düşünülmüştür.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Metot-metot çağırımı ve metot çağırımındaki parametre sayısı (DMMR, NPI (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım özelliğinin hesaba katılacağı belirtilmiştir.
12	Tanesellik esnekliği	Paket seviyesine çıkartılabilir.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Bu ölçüt hata eğilimliliğine etkisi yönünden incelenmiştir. Uyum arttıkça hata eğilimliliği artıyor diye bir sonuç çıkmış ve bu da iyi bir uyum ölçütü olmadığını göstermiştir (Briand vd., 2000).
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan kullanımı/paylaşımı göz önüne alınmamıştır. Metot çağrımları göz önüne alınmıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.10 RCI ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Briand ve arkadaşlarının nesne tabanlı sistemler için önerdikleri uyum ölçüt kümesine dayanarak nesneye yönelik sistemler için önerilmiştir
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Sezgisel olarak tahmin edilen sonuçlarla benzer sonuçlar veriyor.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt yansıtır.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik ve deneysel geçerlemesi yapılmıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Veri-Veri etkileşimi ve Veri-Metot etkileşimi (DDI ve (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Briand ve arkadaşları analiz yaparken "kalıtım alınmış üye alanları analizden çıkarmak, kalıtım alınmış metotları analize almak" seçeneğini önermişlerdir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Yapılan bir çalışmada (Briand vd., 1999) RCI ölçütünün nesne tabanlı programlamada hata-eğilimliğinin iyi bir tahmin edicisi olduğu raporlanmıştır.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan kullanımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.11 Coh ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	LCOM5 ölçütünün bir varyasyonudur.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Sezgisel olarak tahmin edilen sonuçlarla benzer sonuçlar veriyor. Formülü incelediğimizde bazı noktalarda Coh ölçütünün istenmeyen sonuçlar verebileceğini görüyoruz. Şöyle ki, eğer ölçülen sınıfta hiç üye alan yoksa o zaman ölçüt sonsuz bir değere çıkıyor. İyi bir ölçüt her zaman mantıklı sonuçlar veren ve böyle sonuçlar alma ihtimali olmayan ölçüttür.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt bazen yansıtıyor.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik ve deneysel geçerlemesi yapılmamıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Metot üye alan referanslaması (MAR (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Bu ölçüt hata eğilimliliğine etkisi yönünden incelenmiştir (Briand vd., 2000). Bir sınıfın uyumu düştükçe hata eğilimliliğinin artabilme ihtimali konusunda bir gösterge olabileceği görülmüştür.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan kullanımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.12 CBMC ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan Chidamber ve Kemerer'in ölçüt çalışmalarından esinlenerek önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Bazı problemleri var bu ölçütün şöyle ki; Bu ölçüt ile karmaşık etkileşimleri olan bir sınıf basit etkileşimleri olan bir sınıf ile aynı uyum ölçümüne sahip çıkabiliyor. (g grafiğine bir etkileşim kenarı eklendiğinde eğer bileşenleri sayısı azalır ve yapı ağaçları yüksekliği değişmezse) Bu ölçüt ile karmaşık etkileşimleri olan bir sınıf basit etkileşimleri olan bir sınıftan daha düşük bir uyum ölçümüne sahip çıkabiliyor.(g grafiğine bir etkileşim kenar eklendiğinde eğer bileşenleri sayısı azalır ve yapı ağaçları yüksekliği artarsa)
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt bazen yansıtır (Zhou vd., 2002).
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Deneysel geçerlemesi yapılmıştır.
8	Yorumlama	Ölçüt sonucunu yorumlamak için herhangi bir belirtim yapılmamıştır fakat ölçüt formülünde bulunan Fc ve Fs sonuçları için yorumlama yöntemi sunmuşlardır. Şöyle ki; <u>Yüksek Fc, Yüksek Fs:</u> Sınıfın üyeleri sıkı bağlıdır. Sınıf bileşenlerine ayrılırsa, her bir bileşen yüksek uyum gösterecektir. "Bu durumda sınıf iyi tasarlanmış bir sınıf olarak düşünülebilir, kalite açısından herhangi bir şey yapmaya gerek yoktur" görüşünü sunmuşlardır. <u>Düşük Fc, Yüksek Fs:</u> Sınıf birkaç yapışkan metodun çıkarılmasıyla yüksek uyumlu bileşenlere ayrıştırılabilir. Yüksek uyumlu bileşen tasarım safhasında kaçırılmış olan bir sınıfı yakalayabilir. Orijinal sınıf her bir bileşen için yeniden oluşturulabilir ve her bir bileşene ait üye alan da sınıfın üye alanı olarak yeniden oluşturulabilir. <u>Yüksek Fc, Düşük Fs:</u> Sınıfın üyeleri sıkı bağlıyken birkaç yapışkan metodun çıkarılmasıyla bileşenlere kolayca ayrıştırılabilir. "Bu durumda sınıf zayıf tasarlanmış bir sınıf olarak düşünülebilir" görüşünü sunmuşlardır. <u>Düşük Fc, Düşük Fs:</u> Sınıfın üyeleri birbirleriyle çok az ilişkilidir. "Bu durumda sınıf gereksizdir, üyeler ilgili başka sınıflara dağıtılabilir" görüşünü sunmuşlardır.
9	Bağlantı tipi	Metot üye alan referanslaması (MAR (Çizelge 5.2))
10	Özel metotlar	Özel metotlardan bağımsız olarak tanımlanmıştır.
11	Kalıtım	Kalıtım alınmış üye alanların ve kalıtım alınmış public metotların hesaba katılmasını belirtiyor.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Düşük CBMC değerine sahip sınıfların bakım yapılabilirlik ve tekrar kullanılabilirlik gibi kalite faktörlerine dolaylı etkileri olabileceği görülmüştür. Yüksek CBMC ölçüm sonucunun daha kolay bakım yapılabilirliğin ve tekrar kullanılabilirliğin bir göstergesi olabileceği düşünülmüştür.
14	Hesaplama Kolaylığı	Ölçüt hesaplaması karmaşık diye tanımlanmış çünkü yapışkan metotların karmaşıklığı söz konusu ve ölçüt özyineli olarak tanımlanmıştır. Veri toplama zor, verileri formülde kullanma zor (özyineli durum söz konusu) diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan kullanımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.13 CAMC ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Ölçüt, metotların parametre listeleri ile ilgileniyor.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Yapılan bir çalışmada (Bonja ve Kidanmariam, 2006) sezgisel olarak tahmin edilen sonuçlarla benzer sonuçlar vermediği görülmüştür. Bazı problemleri var bu ölçütün şöyle ki; Parametre oluşum matrisleri 'nde aynı sayıda 1 olan iki ayrı sınıfı ayırt edemiyor. CAMC ölçütü küçük sınıfları yüksek uyuma sahip olarak hesaplıyor. Sadece parametre tiplerinin benzerliği ile ilgileniyor onların sınıf içinde kullanımı ile ilgilenmiyor. İki parametre aynı tipte olabilir fakat ikisi farklı fonksiyonellik için kullanılıyor olabilir, ikisi benzer olarak göz önüne alınmamış olabilir. Bütün metotları da parametresiz olan bir sınıf için CAMC ölçütü uygulanamamaktadır.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor (Counsell ve Swift, 2006).
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Deneyisel geçerlemesi yapılmıştır.
8	Yorumlama	Bansiya ve arkadaşları 0.35 ve üzeri CAMC değerine sahip sınıfları uyumlu sınıf kabul ediyorlar.
9	Bağlantı tipi	m metodunun p parametresini içermesi (MPI (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	CAMC ölçütü ile yazılım sistemindeki hata sayısı arasında bir ilişki var mı diye yapılan bir çalışmada hata sayısı ile ölçüt arasında anlamlı bir ilişki bulunamamıştır (Abubakar vd., 2006).
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan kullanımı/paylaşımı göz önüne alınmamıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.14 NHD ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Ölçüt, metotların parametre listeleri ile ilgileniyor. İkili matriste satırlar arası uyumayı ölçer. NHD ölçütü, bu anlamda CAMC ölçütünün bir alternatifi olarak kullanılabilir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	NHD ölçütü, Parametre oluşum matrisleri'nde aynı sayıda 1 olan iki ayrı sınıfı ayırt edebilmektedir. Bu durum teorik olarak ispatlanmıştır (Counsell ve Swift, 2006). Bazı problemleri var bu ölçütün şöyle ki; NHD ölçütü büyük matrisli sınıfları yüksek uyuma sahip olarak hesaplıyor. Sadece parametre tiplerinin benzerliği ile ilgileniyor onların sınıf içinde kullanımı ile ilgilenmiyor. İki parametre aynı tipte olabilir fakat ikisi farklı fonksiyonellik için kullanılıyor olabilir, ikisi benzer olarak göz önüne alınmamış olabilir.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt yansıtıyor (Counsell ve Swift, 2006).
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Deneysel geçerlemesi yapılmıştır.
8	Yorumlama	Counsell ve arkadaşları 0.5 ve üzeri NHD değerine sahip sınıfları uyumlu sınıf kabul ediyorlar. Fakat bu öneri güçlü bir şekilde savunulamamıştır.
9	Bağlantı tipi	Doğrudan metot – metot parametre paylaşımı (DMMPP (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Her hangi bir kalite faktörüne etkisi belirtilmemiştir.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan kullanımı/paylaşımı göz önüne alınmamıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.15 CC(X) ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'nın "benzerlik" ontolojisine dayanan LCOM ölçütlerinin eksikliği adreslenmiş.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	CC(X) ölçütü, bir sınıfın uyum özelliğini sezgilerle benzer olarak yansıtmaktadır ve bunun yanı sıra LCOM ve CAMC ölçütlerinin eşit uyumlu veya eşit uyumsuz olarak tespit ettiği sınıfları da ayırt edebilmektedir.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt yansıtır.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik geçerlemesi yapılmıştır. Küçük bir uygulama kullanılarak deneysel geçerlemesi de yapılmıştır. Deneysel geçerlemede LCOM ve CAMC ölçütlerine karşı yeni önerilen CC(X) ölçütünün daha güçlü olduğu sonucuna varılmıştır (Bonja ve Kidanmariam, 2006).
8	Yorumlama	Sınıf 1 tane metoda sahipse veya hiç metodu yoksa ya da hiç üye alanı yoksa yüksek uyumlu düşünülebilir ve CC değeri olarak 1 atanır diye görüş önerilmiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı (DAS (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Her hangi bir kalite faktörüne etkisi belirtilmemiştir.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.16 ICBMC ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan Chidamber ve Kemerer'in ölçüt çalışmalarından esinlenerek önerilmiştir. CBMC'nin problemlerini adresleyen bir ölçüttür.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	CBMC ölçütüne göre referans grafiğinin etkileşim kalıbını daha iyi karakterize ettiği görülmektedir. Bazı durumlarda sezgilerle uyuşmayan ölçüm sonuçlarına neden olabildiği ölçüt tanımında belirtilmiş fakat Örnek 2 için sezgilerle benzer sonuçlar verdiğini görüyoruz.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt yansıtıyor (Zhou vd., 2002).
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Deneysel ve teorik geçerlemesi yapılmamıştır.
8	Yorumlama	Bir sınıfın ICBMC ile ölçülen yüksek uyum değerinin, elemanlarının bağlı olduğunu gösterdiği, ölçülen düşük uyum değerinin ise bu bağın zayıf olduğunu gösterdiği şeklinde yorumlanması yönünde bir belirtim yapılmıştır.
9	Bağlantı tipi	Metot üye alan referanslaması (MAR (Çizelge 5.2))
10	Özel metotlar	Özel metotlardan bağımsız olarak tanımlanmıştır.
11	Kalıtım	Herhangi bir belirtim yapılmamıştır fakat CBMC ölçütünden türediği için orada ifade edilen düşünce "Kalıtım alınmış üye alanların ve kalıtım alınmış public metotların hesaba katılması" geçerli olabilir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	ICBMC ölçütünün, kalite değerlendirmesi ve zayıf tasarımlı sınıfların yeniden inşa edilmesi yönünde rehber olabileceği görüşü düşünülmüştür.
14	Hesaplama Kolaylığı	Ölçüt hesaplaması karmaşık diye tanımlanmış çünkü yapışkan metotların karmaşıklığı söz konusu ve ölçüt özyineli olarak tanımlanmıştır. Veri toplama zor, verileri formülde kullanma zor (özyineli durum söz konusu) diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan kullanımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.17 CCM ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Kalıtım hiyerarşisinin izlediği bazı 4 tasarım kuralına dayalı olarak önerilmiş (Jarallah vd., 2001).
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamiyle bulunur.
4	Sağlamlık/Geçerlilik	Doğru bir şekilde uyum özelliği yansıtmıyor sadece bir fikir veriyor. Yanlış veya belirsiz sonuçlar verebiliyor. Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik geçerlemesi yapılmıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı ve doğrudan metot-metot çağırımı (DAS ve DMMR (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanarak ve ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Uyumun yazılımın güvenilirliği üzerinde rolü var mı diye deneysel bir çalışma yapılmıştır. CCM ölçütü ile yazılım sistemindeki hata sayısı arasında bir ilişki var mı diye yapılan bir çalışmada hata sayısı ile ölçüt arasında anlamlı bir ilişki bulunamamıştır (Abubakar vd., 2006).
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağırımları göz önüne alınmıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.18 ECCM ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Kalıtım hiyerarşisinin izlediği bazı 4 tasarım kuralına dayalı olarak önerilmiş (Jarallah vd., 2001).
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Gerçekleme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Doğru bir şekilde uyum özelliği yansıtmıyor sadece bir fikir veriyor. Yanlış veya belirsiz sonuçlar verebiliyor. Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik geçerlemesi yapılmıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı ve doğrudan metot-metot çağırımı (DAS ve DMMR (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım alınmış metotların ve yeniden tanımlanmış metotların kullanımından sakınılması gerektiği önerilmiş.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanarak ve ortalaması alınarak paket seviyesine yükseltilebilir ancak metot çiftlerine bakıldığı için metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Her hangi bir kalite faktörüne etkisi belirtilmemiştir.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağırımları göz önüne alınmıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.19 OCC ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan Chidamber ve Kemerer'in ölçüt çalışmalarından esinlenerek önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik geçerlemesi yapılmıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı, doğrudan metot-metot çağırımı (DAS ve DMMR (Çizelge 5.2))
10	Özel metotlar	Erişim metotları normal metotlarmış gibi işlem görmüştür.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Her hangi bir kalite faktörüne etkisi belirtilmemiştir.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağırımları, dolaylı yoldan üye alan kullanımına bakmak amacıyla kullanılmış, metotların birbirini çağırma anlamında metot ilişkisi göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmıştır.

Çizelge Ek1.20 PCC ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan Chidamber ve Kemerer'in ölçüt çalışmalarından esinlenerek önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik geçerlemesi yapılmıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı, doğrudan metot-metot çağırımı ve bir metodun yazdığı üye alanı diğer metodun okuması (DAS, DMMR ve MIBAT (Çizelge 5.2))
10	Özel metotlar	Erişim metotları normal metotlarmış gibi işlem görmüştür.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Her hangi bir kalite faktörüne etkisi belirtilmemiştir.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağırımları, dolaylı yoldan üye alan kullanımına bakmak amacıyla kullanılmış, metotların birbirini çağırma anlamında metot ilişkisi göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmıştır.

Çizelge Ek1.21 OL_n ölçütü sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan Chidamber ve Kemerer'in ölçüt çalışmalarından esinlenerek önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamiyle bulunur.
4	Sağlamlık/Geçerlilik	Sezgilerle paralellik göstermeyebiliyor. Her ne kadar Örnek 2 için sezgilerle paralel sonuçlar verdiği görülse de bazı araştırmacıların yaptıkları çalışmalar (Zhou vd., 2002) sonucu şu görüş vardır; OL _n ölçütü pratikçilere uygun bir n iterasyon faktörü seçmeleri için rehberlik etmez ve böylece uyum ölçümü pratikte zordur. İkinci olarak kendi kendine çelişen sonuçlar verebilir.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt bazen yansıtır (Zhou vd., 2002).
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Teorik geçerlemesi yapılmıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan üye alan paylaşımı, doğrudan üye alan referanslaması (DAS, MAR (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Her hangi bir kalite faktörüne etkisi belirtilmemiştir.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma formül iterasyon içerdiği için zor diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan kullanımı ve paylaşımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmamıştır.

Çizelge Ek1.22 DC_D, DC_I, LCC_D ve LCC_I ölçütleri sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'ın "benzerlik" ontolojisine dayanan Chidamber ve Kemerer'in ölçüt çalışmalarından esinlenerek önerilmiştir.
2	Tanesellik	Sınıf seviyesinde uyum ölçümü yapar.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamıyla bulunur.
4	Sağlamlık/Geçerlilik	Sezgilerle paralellik göstermeyebiliyor. 2 tane farklı uyum özelliğine sahip sınıf aynı veya yakın değer alabiliyor.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt net yansıtmıyor.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Deneysel geçerlemesi yapılmış ve ilişkili metotları daha iyi yakaladığı görülmüştür (Badri ve Badri, 2003).
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan ve dolaylı üye alan paylaşımı, doğrudan ve dolaylı metot çağırımı (DAS, IAS, DMMR, IMMR (Çizelge 5.2))
10	Özel metotlar	Çalışmalarından çıkarmışlardır.
11	Kalıtım	İleriki bir çalışmada inceleneceği belirtilmiştir.
12	Tanesellik esnekliği	Bir sistemdeki sınıfların uyum değerleri toplanıp ortalaması alınarak paket seviyesine yükseltilebilir ancak metot seviyesine indirgenemez.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Her hangi bir kalite faktörüne etkisi belirtilmemiştir.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağırımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmıştır.

Çizelge Ek1.23 WTCoh ve ClassCoh ölçütleri sınıflandırması

No	Sınıflandırma kriteri	Ölçüt durumu
1	Altta yatan yaklaşım	Bunge'in "benzerlik" ontolojisine dayanan Chidamber ve Kemerer'in ölçüt çalışmalarından esinlenerek önerilmiştir.
2	Tanesellik	WTCoh sistem seviyesinde ClassCoh sınıf seviyesinde uyum ölçümü yapar.
3	Mevcudiyet	Tasarım safhasında kısmen vardır, gerçekleştirme safhasında tamamiyle bulunur.
4	Sağlamlık/Geçerlilik	Sezgisel olarak tahmin edilen sonuçlarla benzer sonuçlar veriyor. Bir sistemin uyumu hesaplanırken sınıflar birbirinden bağımsız düşünülüyor ve WTCoh ölçütü doğal olarak sistemin uyumunu düşürüyor. Bu da beklenen sonuçtur.
5	Duyarlılık	Sınıfın uyumu değiştiği zaman bu değişimi ölçüt yansıtıyor.
6	Normalizasyon	Normalize edilmiştir.
7	Geçerleme	Deneyisel geçerlilikleri yapılmış fakat teorik geçerlilikleri yapılmamıştır.
8	Yorumlama	Ölçüm sonucunda çıkan değere göre nasıl yorumlanacağı belirtilmemiştir.
9	Bağlantı tipi	Doğrudan ve dolaylı üye alan paylaşımı (DAS, IAS (Çizelge 5.2))
10	Özel metotlar	Özel metotların göz önüne alınıp alınmadığı belirtilmemiştir.
11	Kalıtım	Kalıtım konusunun göz önüne alınıp alınmadığı belirtilmemiştir.
12	Tanesellik esnekliği	ClassCoh ölçütü sistem seviyesine yükseltgenerek WTCoh ölçütü oluşturulmuştur.
13	Kalite Faktörüne Etkisi (ispatlanan ya da olduğu düşünülen)	Java bileşenlerinin tekrar kullanılabilirliğinin (reusability) değerlendirilmesi amacıyla önerilmişler ve deneyisel geçerlemede tekrar kullanılabilirliği daha iyi tahmin edici oldukları belirlenmiştir.
14	Hesaplama Kolaylığı	Veri toplama basit, verileri formülde kullanma basit diye nitelendirilebilir.
15	Üye alan/metot varlığı	Üye alan paylaşımı göz önüne alınmıştır. Metot çağrımları göz önüne alınmamıştır.
16	Dolaylı bağlantı varlığı	Dolaylı bağlantılar göz önüne alınmıştır.

ÖZGEÇMİŞ

Doğum tarihi 20.11.1979

Doğum yeri Antakya

Lise 1990-1997

Hüseyin Avni Sözen Anadolu Lisesi

Lisans 1997-2002

Yeditepe Üniversitesi Mühendislik Fak.
Bilgisayar Mühendisliği Bölümü

Yüksek Lisans 2006-2009

Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Müh. Anabilim Dalı,
Bilgisayar Müh. Programı

Çalıştığı kurum(lar)

2003-Devam ediyor TÜBİTAK Marmara Araştırma Merkezi Bilişim
Teknolojileri Enstitüsü - Uzman Araştırmacı

