

객체지향 패러다임에서 저해요인에 기반한 응집도 척도

김 성 애[†]·최 완 규^{††}·이 성 주^{†††}

요 약

객체지향 패러다임에서 응집도가 높은 클래스들은 단 하나의 데이터 요소만이 정의되고 모든 메소드(또는 연산)들이 그 데이터 요소 상에서 수행되거나, 메소드와 데이터 요소간에 최대연결이 이루어진 클래스이다. 따라서 데이터 요소의 증가와 최대연결을 저해하는 연산들은 클래스의 응집을 저해하는 주 요인들이다. 본 연구에서는 이러한 저해요인들에 근거하여, 응집도에 관한 우리의 직관과 더욱 일치하는 측정을 할 수 있는 새로운 클래스 응집도 척도인 *CM*(Cohesion Metric)을 제안하였다. 본 연구에서는 응집도를 저해하는 요인의 식별을 위해 참조 그래프(reference graph)를 이용하고, 참조 그래프의 분석을 기초로 하여 클래스의 응집도를 측정하였다. 또한 제안된 *CM* 측정값이 응집도 척도가 만족해야하는 특성들을 만족함을 보였고, 기존 응집도 척도들과의 비교평가를 통해서 *CM*이 기존의 척도들보다 향상된 측정을 제시함을 보였다.

Cohesion Measure by the Decreasing Factors for an Object-Oriented Paradigm

Sung-Ae Kim[†] · Wan-Kyoo Choi^{††} · Sung-Joo Lee^{†††}

ABSTRACT

In object-oriented paradigm, highly cohesive classes define only a data factor(instance variable) whose all methods(or operations) refer to, or have the maximum connection among elements in a class. Therefore, an increase in the number of the data factors, and the operations to obstruct the maximum connection among elements in a class are the main factors to decrease the class cohesion. In this paper, we propose a new measure, called *CM*(Cohesion Measure), which can measure the class cohesion more consistently with our intuition on the basis of factors to decrease it. We use the reference graph in order to find the decreasing factors of the class cohesion, and measure the class cohesion on the basis of analysis of the domain reference graph. We also show that the proposed *CM* metrics satisfy the properties that the cohesion measures should satisfy, and that it can provide more upgraded measures than the existing measures through comparing it with the measures of existing cohesion.

1. 서 론

객체지향 패러다임에서 소프트웨어 품질, 프로그램 생산성, 프로그램 복잡도와 같은 다양한 속성들을

측정하기 위한 척도들이 제안되었다. 모듈 독립성은 소프트웨어 복잡도를 감소시켜 소프트웨어의 품질과 신뢰성을 향상시킬 수 있는 핵심으로 간주되었다. 또한 더욱 신뢰성 있고 유지보수 가능한 소프트웨어를 생산하기 위해 연구자들은 응집도라고 불리는 또 다른 속성을 측정해왔다[1].

응집도는 모듈 구성 요소들의 연관성이라고 하는 소

[†]준 회원 : 조선대학교 대학원 전자계산학과
^{††}준 회원 : 광주대학교 컴퓨터전자통신공학부 교수
^{†††}정 회원 : 조선대학교 컴퓨터공학부 교수
논문접수 : 2000년 8월 12일, 심사완료 : 2000년 11월 17일

소프트웨어의 속성으로, 소프트웨어의 응집도가 높을수록 소프트웨어는 이해하고 유지보수하기가 쉽다[2].

객체지향 패러다임에서 응집도의 개념은 절차적 패러다임에서 응집도의 개념을 확장하여 정의된다. 모듈이 함수나 프로시저로 정의되는 절차적 패러다임과 달리, 객체지향 패러다임에서 모듈은 클래스로 정의할 수 있다. 비공개(private) 인스턴스 변수들과 공개(public) 메소드들을 갖는 클래스는 추상데이터 타입(abstract data type : ADT)이라 할 수 있다. 하나의 클래스는 어떤 기능들을 수행하기 위한 여러 개의 메소드를 갖는다.

ADT에서의 응집도를 측정하기 위한 여러 연구들[1, 3, 4-11]이 제안되었다. 이런 연구들은 클래스의 응집도는 클래스 내의 인스턴스 변수들간의 상호작용의 함수 또는 클래스에 의해 인스턴스 변수들이 사용되어지는 방법이라고 가정하고, 인스턴스 변수들 간의 상호작용을 증대하는 메소드들의 연결성의 정도에 의해 응집도를 측정한다.

그러나, 기존의 척도들은 인스턴스 변수들간의 관계를 측정한다고 가정하고 있지만, 주로 메소드들간의 연결 관계나 메소드와 인스턴스 변수간의 연결의 개수에만 근거하여 응집도를 측정하였다. 이것은 응집도에 대한 우리의 이해와 측정값간의 모순을 발생시키는 주요 이유가 된다.

따라서 본 연구에서는 응집도의 정의에서 응집도를 저해하는 요인들을 분석하고, 저해요인들을 반영하여 응집도를 측정할 수 있는 새로운 응집도 척도를 제시한다.

본 연구에서의 분석에 의하면, 객체지향 패러다임에서 응집도를 저해하는 요인들은 클래스의 데이터 요소인 인스턴스 변수의 증가와 메소드와 인스턴스 변수간의 최대 연결을 저해하는 연산들이다.

응집도 측정이 의미 있는 측정을 제공하기 위해서는 척도가 엄밀하게 정의되어야하고, 측정을 위한 속성을 정확히 반영해야하고, 이런 속성들을 포착하는 모델에 근거해야한다[12, 13].

따라서 본 연구에서는 응집도를 저해하는 요인의 식별을 위해 참조 그래프를 도입하고, 저해요인 분석에 근거하여 참조 그래프 상에서 클래스의 응집도를 측정한다.

2장에서는 객체지향 패러다임에서 기존 응집도 척도들을 조사하고, 기존 척도들의 문제점을 분석한다. 3장에서는 클래스의 응집도를 저해하는 요인들을 분석하

고, 저해요인들을 고려하여 클래스의 응집도를 측정할 수 있는 척도를 제안한다. 4장에서는 응집도 척도가 만족해야할 필요한 특성들을 제시하고, 제안된 척도가 제시된 특성을 만족하는 가를 살펴본다. 5장에서는 기존 연구들과 제안된 척도를 비교 평가한다. 6장은 본 연구의 요약이며 향후 연구 방향을 제시한다.

2. 관련연구

2.1 Embley의 응집도 기준

Embley[14]는 클래스 내의 요소들이 얼마만큼 관련을 맺고 있는가를 구분하기 위해서 ADT에 대해 분리가능 응집도(separable Cohesion), 다면적 응집도(multifaceted Cohesion), 비대표적 응집도(non-delegation Cohesion), 은닉 응집도(concealed Cohesion), 이상적 응집도(model Cohesion)등 5가지의 응집도 범주를 정의하였다.

Embley에 의하면, 분리가능 응집도를 갖는 클래스보다 다면적 응집도를 갖는 클래스가 응집도가 높고, 이상적 응집도를 갖는 클래스가 가장 바람직한 클래스이다.

높은 클래스의 응집도는 객체의 캡슐화를 증대시키므로 바람직하다. 낮은 응집도는 복잡도를 증가시켜 개발 과정에서 오류가 발생할 가능성이 높으므로 클래스의 응집도가 결핍된 상태이면 클래스는 두 개 이상의 클래스로 분할하여야 한다.

클래스의 응집도 등급의 결정이 주관적이고 클래스의 의미론적 분석에 기초하므로, 정의들은 특정 상황에서 주어진 응집도 등급의 근사치를 측정하는 구분론적 기반 척도를 유도하는 지침서로 사용되어야 한다[15].

따라서 본 연구에서는 Embley의 기준을 클래스 응집도 측정을 위한 지침서로 이용한다.

2.2 객체지향 패러다임에서 응집도

2.2.1 Chidamber와 Kemerer의 LOCM

Chidamber와 Kemerer[3]는 LCOM(Lack of Cohesion in Methods)을 제안하였다.

n 개의 메소드 즉, M_1, \dots, M_n 을 갖는 클래스 C_1 에서, 메소드 M_j 가 사용하는 인스턴스 변수의 집합을 I_j 라고 하고, n 개의 I_j 가, 즉 I_1, \dots, I_n 이 있고, $P = \{(I_i, I_j) \mid I_i \cap I_j = \emptyset\}$ 이고, $Q = \{(I_i, I_j) \mid I_i \cap I_j \neq \emptyset\}$ 이고, n 개의 집합 I_1, \dots, I_n 이 모두 \emptyset 이면 $P = \emptyset$ 라 할 때, LCOM은 다음과

같이 측정된다.

$$LCOM = |P| - |Q|, \text{ if } |P| > |Q| \quad (1)$$

$$= 0, \text{ otherwise}$$

LCOM에서는 측정값이 작을수록 응집적인 클래스가 되므로, LCOM = 0인 클래스가 가장 응집적인 클래스이다.

LCOM의 문제점은 LCOM이 두 클래스의 구조적 응집성을 구분할 수 없으며[5], 메소드의 추가에 대해서 모순된 결과를 보여주는 점이다.

2.2.2 Hitz and Montazeri의 Connectivity Metric

Hitz and Montazeri[4, 5]는 Chidamber와 Kemerer가 제안한 응집도 척도를 그래프 이론에 기반하여 수정하였다.

클래스 X가 인스턴스 변수들의 집합 I_X 와 메소드의 집합 M_X 를 갖는다고 할 때, $G_X(V, E)$ 는 클래스의 메소드들을 나타내는 단순한 무방향성 그래프이다. 여기서 $V = M_X$ 이고 $E = \{(m, n) \in V \times V \mid (\exists i \in I_X : (m \text{은 } i \text{를 액세스}) \wedge (n \text{은 } i \text{를 액세스}))\}$ 이다. LCOM(X)는 G_X 에서 연결된 컴포넌트들의 개수로 정의된다. 여기서 $1 \leq LCOM(X) \leq |M_X|$ 이고, “m은 i를 액세스”는 m이 인스턴스 변수 i를 사용하는 것을 의미한다.

Hitz 등은 동일한 LCOM 값을 갖는 클래스 집합의 메소드들간의 구조적 차이를 측정할 수 있도록 LCOM을 수정하여 연결 메트릭(connectivity metric)을 정의하였다.

$$C = 2 \left| \frac{E - (n-1)}{(n-1)(n-2)} \right| \quad (2)$$

여기서 E는 그래프에서 간선의 개수이고, n은 정점의 개수이다. $0 < C < 1$ 이고, C값이 클수록 클래스의 응집도는 높다.

Hitz등은 연결 메트릭이 LCOM의 단점을 해결했다고 하지만[4, 5], 연결 메트릭이 LCOM과 결합하여 사용되므로, LCOM의 최대값(bound)을 결정하는 문제를 해결하지 못했다[1].

2.2.3 Chen과 Lu의 Metric

Chen과 Lu[6]는 중복 인수(overlapping argument)를 갖는 연산(즉, 메소드)들은 연관되어 클래스를 더 응집적으로 만든다는 개념에 기초하여 응집도 메트릭을 제안하였다.

클래스 C가 N개의 연산, 즉 F(1), ..., F(N)과 N개의 인수들, 즉 I(1), ..., I(N)을 갖고, 이때 M은 N개의 인수 집합의 교집합에 의해 형성되는 서로소인 집합이라고 하면, 응집도 메트릭은 다음과 같이 정의된다.

$$Cohesion\ Metric = \frac{M}{N} \times 100\% \quad (3)$$

LCOM이 메소드들에 의해 사용되는 인스턴스 변수들의 교집합을 이용하는데 비해, Chen과 Lu는 인수들의 교집합을 사용한다. 이 메트릭의 값이 적을수록 클래스의 응집도는 높다.

Chen & Lu의 메트릭의 문제점은 파라미터 집합들간의 관계성의 정도를 구분하지 못함에 있다[1].

2.2.4 Baxter, Chu와 Patel의 Composite cohesion measure

Baxter, Chu와 Patel[7]은 Ada 패키지에서 응집도를 측정할 합성 응집도 척도(composite cohesion measure)를 제안하였다.

하나의 프로시저 또는 함수(즉, p_i)의 집합을 의미하는 합성 모듈 $P = \{p_i, \dots, p_m\}$ 이 주어지면 P의 응집도는 다음과 같이 측정된다.

$$Cohesion(P) = \frac{\sum_S Sim_2(p_i, p_j)}{\sum_{i=1}^{m-1} i} \quad (4)$$

여기서 $S = \{(i, j) \mid i, j \in (1, m) \wedge i > j\}$ 이고, $Sim_2(p_i, p_j)$ 는 모듈에서 다른 프로시저 순서쌍들 간의 유사성의 합계이고 다음과 같이 정의된다.

$$Sim_2(p_i, p_j) = \frac{\sum_S p_i p_j}{\sqrt{\sum_1^n x_i^2} \times \sqrt{\sum_1^n y_i^2}} \quad (5)$$

여기서 n은 모듈에서의 멤버들의 개수이고, x_i 는 프로시저 p_i 에서 참조되는 특정 유형의 변수들의 개수이고, y_i 는 프로시저 p_j 에서 참조되는 특정 유형의 변수들의 개수이다. $\sum_{i=1}^{m-1} i$ 는 유사성 척도가 계산된 순서쌍의 집합이다. 이 척도의 값이 클수록 클래스의 응집도는 높다.

합성 응집도 척도가 인스턴스 변수 자체가 아니라 변수의 유형에 근거하여 응집도를 측정하므로, 데이터

유형 선택에 더욱 정확한 프로그래머가 소수의 데이터 유형을 갖는 프로그래머에 비해 항상 낮은 응집도를 갖는 클래스들을 작성하게 된다. 따라서 좋은 프로그래밍에 대한 보편적인 이해와 응집도와 이 척도간에는 모순이 존재한다[1]. 이는 하나의 메소드가 유형 참조(type reference)를 통해 다른 메소드에게 영향을 미칠 수 없기 때문에 발생하는 문제이다[2].

2.2.5 Bieman과 Kang의 Class cohesion measure

Bieman과 Kang[8]은 클래스 응집도와 비공개(private) 재사용간의 관계를 평가하는 2개의 클래스 응집도 척도인 TCC(tight class cohesion)와 LCC(loose class cohesion)를 제안하였다.

TCC와 LCC는 메소드 순서쌍간의 직접 또는 간접 연결에 근거하여 클래스의 응집도를 측정한다. 하나 이상의 공통 인스턴스 변수를 사용하는 두 메소드는 직접 연결된다고 하고, 다른 직접 연결된 메소드를 통해 연결되는 두 메소드는 간접 연결된다고 한다.

NDC(C)가 클래스에서 직접 연결된 메소드들의 개수이고, NIC(C)가 간접 연결된 메소드의 개수라 하고, NP(C) = N × (N - 1) / 2는 클래스에서 가능한 최대 연결 개수라고 하면, TCC와 LCC는 다음과 같다.

$$TCC(C) = NDC(C) / NP(C) \tag{6}$$

$$LCC(C) = NIC(C) / NP(C) \tag{7}$$

TCC와 LCC 값이 클수록 클래스의 응집도는 높다.

Bieman과 Kang의 응집도 척도의 문제점은 많은 메소드들을 갖는 클래스들은 항상 낮은 응집도를 갖는다는 것[1]과 메소드와 인스턴스 변수들간의 연결 패턴의 차이를 구분하지 못하는 점이다.

2.2.6 Mehra의 Cohesion measures

Mehra[1]는 Bieman과 Otto[12]의 개념들을 도입하여 객체지향 패러다임에서 메트릭 데이터 슬라이스에 기초한 데이터 응집도 척도들인 SDC(strong sata cohesion), WDC(weak data cohesion), A(adhesiveness)를 제안하였다.

$$SDC(C) = \frac{|SG(CSA(C))|}{|tokens(C)|} \tag{8}$$

$$WDC(C) = \frac{|G(CSA(C))|}{|tokens(C)|} \tag{9}$$

$$A(C) = \frac{\sum_{d \in G(CSA(C))} \# \text{ slices containings } d}{|tokens(C)| \times |CSA(C)|} \tag{10}$$

여기서, 클래스 C의 슬라이스 추상화 CSA(C)는 클래스를 데이터 슬라이스의 집합으로 모델링한다. SG(CSA(C))는 CSA(C)에서 super-gluе 토큰들의 집합을 나타내고, G(CSA(C))는 glue 토큰들의 집합을 나타낸다. tokens(C)는 클래스 C의 모든 데이터 토큰들을 나타낸다. SDC, WDC, A의 값이 클수록 클래스의 응집도는 높다.

Mehra의 응집도 척도들은 인스턴스 변수들에 대한 슬라이스에 근거한 개개의 메소드에 대한 기능 응집도(functional Cohesion)[12]가 0이 되면, 메소드의 개수에 상관없이 응집도 측정값이 0이 된다는 문제점이 있다.

2.2.7 Briand와 Morasca의 RCI

Briand와 Morasca[10, 11]은 응집도 척도 RCI(ratio of cohesive interaction)를 제안하였다. RCI는 클래스를 데이터 선언과 메소드들의 집합으로 간주한다.

데이터 선언 a의 변화 또는 사용이 데이터 선언 b의 변화 또는 사용을 유발하면, a는 데이터 선언 b와 연결(interact)된다. 이것을 DD-연결(DD-interacts)이라 한다. 데이터 선언 a가 메소드 m의 적어도 하나의 데이터 선언과 DD-연결되면 a와 m간에 DM-연결(DM-interaction)이 존재한다.

CI(c)가 클래스 c에서의 실질적인 DM-연결과 DD-연결의 집합으로 정의되고, Max(c)가 모든 가능한 DD-연결과 DM-연결의 집합으로 정의될 때, RCI는 다음과 같이 정의된다.

$$RCI = \frac{|CI(c)|}{|\max(c)|} \tag{11}$$

RCI는 0과 1사이의 값을 갖고, 0은 최소 응집도, 1은 최대 응집도를 나타낸다.

RCI가 연결 개수만을 고려하고 연결 패턴을 고려하지 않으므로, RCI는 연결 패턴의 차이에 따른 응집도에 관한 직관과 모순되는 결과를 초래하는 문제점이 있다.

2.2.8 Park 등의 Class Cohesion

Park등[9]은 메소드간의 연결강도에 근거한 클래스 응집도 척도를 제안하였다. I는 클래스 내에서 정의된 인스턴스 변수들의 집합이고 MP는 클래스 내의 모든

메소드들의 조합인 메소드-순서쌍의 집합이라고 하면, 연결강도(CI : connectivity intensity)와 클래스 응집도(CC : class cohesion)는 다음과 같다.

$$CI = \frac{|\{i \in I \mid m_1 \text{ and } m_2 \text{ use } i, (m_1, m_2) \in MP\}|}{|I|} \quad (12)$$

$$CC = \frac{\sum_{(m_i, m_j) \in MP} CI \text{ of } (m_i, m_j)}{|MP|} \times 100 \quad (13)$$

0 ≤ CC ≤ 100이고, CC의 값이 클수록 클래스의 응집도는 높다.

RCI와 마찬가지로, Park등의 클래스 응집도는 역시 연결 개수만을 고려하고 연결 패턴을 고려하지 않으므로, RCI와 동일한 문제점을 내포하고 있다.

3. 클래스 응집도 척도

객체지향 패러다임에서는 소프트웨어의 구성 요소들을 하나의 기본 단위 구조로 파악하고, 기본 단위 안에 있는 요소들간의 관계 정도를 측정하여 응집도를 나타낸다[16].

본 연구에서는 클래스의 응집도를 저해하는 요인들의 분석에 기초하여 클래스 응집도를 측정할 수 있는 척도를 정의하기 위하여 클래스 응집도 척도의 개발을 위한 모델을 도입하고, 모델에 근거하여 척도를 정의한다.

3.1 참조 메소드 집합(Reference method set)

응집도 저해 요인 분석과 응집도 척도를 개발하기 위해 참조 그래프(Reference graph)[14, 17]에서의 참조 메소드 집합(Reference method set)을 정의한다.

정의 1 : 참조 그래프

$D = \{d_1, d_2, \dots, d_n\}$ 가 클래스에서 정의된 데이터 요소들인 인스턴스 변수들의 집합이고, $M = \{m_1, m_2, \dots, m_k\}$ 이 클래스 내에서 정의된 메소드들의 집합이라 하자. $m(m \in M)$ 이 $d(d \in D)$ 를 m 내에서 사용하면 m 이 d 를 참조한다고 하고, $d \leftarrow m$ 으로 나타낸다. 참조 그래프 $G(V, E)$ 는 방향성을 가지는 그래프로서, $V = (DU \cup M)$ 이고, $E = \{(d, m) \mid d \leftarrow m\}$ 이다.

(참조) 그래프에서의 참조 메소드 집합(reference method set)을 다음과 같이 정의한다.

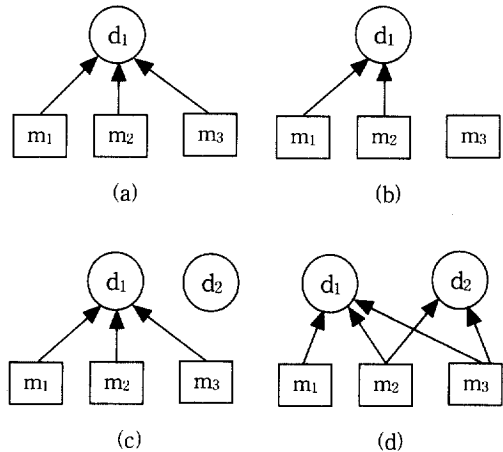
정의 2 : 참조 메소드 집합

$d(d \in D)$ 를 참조하는 $m(m \in M)$ 들의 집합으로 정의되는 참조 메소드 집합은 다음과 같다.

$$Ref_d = \{m_j \mid d_i \leftarrow m_j, d_i \in D, m_j \in M, j=1, \dots, k\}$$

k : 메소드 집합(M)내의 메소드의 총수

(그림 1)은 참조 그래프의 예를 보여준다. Embley의 응집도 범주의 정의[14]에 의하면, (그림 1-a)는 하나의 인스턴스 변수만이 정의되어 있고 모든 메소드들이 그 인스턴스 변수에 관련되어 있으므로 이상적(model) 응집도를 갖고, (그림 1-b)와 (그림 1-c)는 각각 참조가 발생하지 않는 메소드와 참조되지 않는 인스턴스 변수가 존재하므로 분리가능 응집도(seperable)를 갖는다. (그림 1-d)는 두 개의 인스턴스 변수를 갖고 있으며, 복수 참조 노드 m_2 와 m_3 을 갖고 있으므로 이상적인 응집도 보다 낮은 다면적(mutifaceted) 응집도를 갖는다.



(그림 1) 참조 그래프

3.2 응집도 저해 요인

이상적 응집도를 갖는 클래스는 단 하나의 인스턴스 변수만을 정의하고 클래스의 모든 메소드가 그 인스턴스 변수를 참조하도록 정의된 클래스이다[14]. 이러한 클래스는 단일 인스턴스 변수에 대한 최대 연결이 이루어지므로 가장 응집적인 클래스이다. 클래스의 멤버들이 그들간에 최대 연결을 가질 때, 클래스는 가장 응집적이다. 즉, 클래스 내의 모든 메소드가 클래스 내의 모든 인스턴스 변수와 연결될 때 클래스는 가장 응

집적이며[17], 응집적인 클래스의 모든 메소드들은 대부분의 인스턴스 변수들을 사용한다[1].

따라서 본 연구에서는 클래스의 응집도를 저해하는 요소들을 다음과 같이 분석하였다.

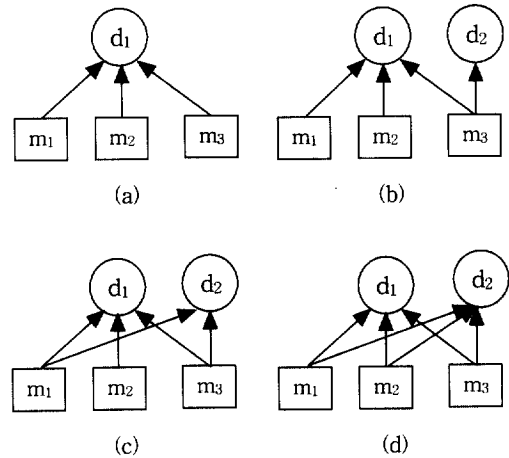
저해요인 1 : (데이터 요소의 증가) 참조 그래프에서 $|D|=1$ 이고 이상적 응집도를 갖는 클래스에서 데이터 요소가 추가되면 응집도는 감소한다. 이상적 응집도를 갖는 클래스에 데이터 요소가 추가되어 $|D| \geq 2$ 되고 참조그래프가 연결되면 클래스는 다면적 응집도를 갖고, 연결되지 않으면 분리가능 응집도를 갖는다. 즉, 데이터 요소의 증가는 클래스 내의 메소드와 인스턴스 변수들간의 최대 연결을 위해 필요한 간선의 개수(또는 비율)를 증가시키므로 응집도를 감소시킨다.

저해요인 2 : (모든 데이터 요소를 참조하지 않는 메소드의 증가) 모든 데이터 요소 상에서 수행하는 메소드의 추가는 일부 데이터 요소들에서 수행하는 메소드보다 훨씬 더 유사한 것으로 간주할 수 있으므로 모든 인스턴스 변수 상에서 수행하는 메소드의 추가는 응집도를 증가시키지만 일부 인스턴스 변수들에서 수행하는 메소드의 추가는 응집도를 감소시킨다. 즉, 모든 데이터 요소를 참조하지 않는 메소드의 증가는 클래스 내의 메소드와 인스턴스 변수들간의 최대 연결을 위해 필요한 간선의 개수(또는 비율)를 증가시키므로 응집도를 감소시킨다.

저해요인 3 : (참조 그래프에서 서로소인 부분 그래프) 인스턴스 변수 참조 그래프에서 서로소인 부분 그래프를 갖는 클래스는 분리가능 응집도를 갖는다. 이러한 클래스는 더 높은 응집도를 갖도록 분리가능한 부분들로 분할할 수 있다. 그러므로, M과 D가 유사한 상황에서 서로소인 부분 그래프를 갖는 클래스는 하나의 연결 그래프로 구성된 클래스보다 낮은 응집도를 가져야하고, 따라서 서로 소인 부분 그래프의 존재는 응집도를 저해하는 요인이 된다.

(그림 2)에서, (그림 2-b)는 (그림 2-a)에 인스턴스 변수 d_2 와 참조관계 $m_3 \rightarrow d_2$ 의 추가에 의하여 다면적 응집도를 갖는다. 따라서 (그림 2-b)의 응집도는 (그림 2-a)보다 낮게된다. (그림 2-c)는 (그림 2-b)에 참조관계 $m_1 \rightarrow d_2$ 의 추가에 의하여 여러 인스턴스 변수와 관련된 메소드들이 증가한다. 따라서, (그림 2-c)는 메

소드와 인스턴스 변수들간의 최대 연결을 위해 필요한 간선의 개수(1개 : $m_2 \rightarrow d_2$)가 (그림 2-b)의 개수(2개 : $m_1 \rightarrow d_2, m_2 \rightarrow d_2$)보다 적게되므로 (그림 2-c)의 응집도는 (그림 2-b)의 응집도보다 높게된다. (그림 2-d)는 메소드와 인스턴스 변수간에 최대 연결이 이루어지므로 (그림 2-c)보다 더 응집적인 클래스가 된다.



(그림 2) 참조 그래프

3.3 척도의 정의

클래스의 응집도를 저해하는 요인들을 고려하여 클래스의 응집도를 측정하기 위한 응집도 척도를 다음과 같이 정의한다.

정의 3 : Cohesion Measure(CM)

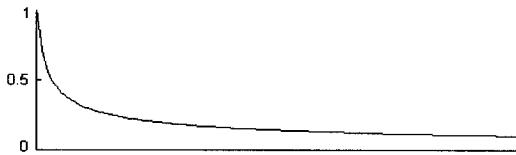
D와 M이 하나의 클래스 C 내에서 정의된 데이터 요소인 인스턴스 변수들과 메소드들의 집합이라 하고, 참조 그래프 $G(V, E)$ 는 방향성을 가지는 그래프로서, $V = (D \cup M)$ 이고, $E = \{(d, m) \mid d \leftarrow m, d \in D, m \in M\}$ 이고, $d(d \in D)$ 에 대한 참조 메소드 집합을 Ref_d 라 하고, DS를 그래프 G에서 서로소인 부분그래프의 개수라 하고, $d_i \in D$ 라 하면, 응집도 척도 CM은 다음과 같다.

$$CM(C) = 0, \text{ if } M = \emptyset \text{ or } D = \emptyset \tag{14}$$

$$= \frac{\sum_{i=1}^{|D|} |Ref_{d_i}|}{|M| \times |D| \times \sqrt{DS}}, \text{ Otherwise}$$

식 (14)에서 Ref_d 는 데이터 요소와 메소드간의 연결관계들, $|D|$ 은 데이터 요소의 개수를, $|M|$ 은 메소드

의 개수를 반영한다. 따라서, 참조 그래프에서 데이터 요소 또는 메소드의 증가가 동일한 비율로 데이터 요소와 메소드간의 참조관계를 증가시키지 않으면 식 (14)는 응집도를 감소시킨다. 또한 DS는 서로소인 부분그래프의 개수로서 데이터 요소와 메소드간의 연결 패턴을 반영한다. 서로소인 부분 그래프를 갖는 클래스는 분리가능 응집도를 갖게되므로, DS의 증가는 (그림 3)과 같은 비율로 응집도를 감소시킨다.



(그림 3) DS의 증가에 따른 응집도의 감소

<표 1>은 (그림 1)과 (그림 2)에 대한 CM 측정값들의 예를 보이고 있다.

<표 1> 그림 1과 2에 대한 응집도

	$\sum Ref_{id} $	$ M $	$ D $	DS	CM
(그림 1-a)	3	3	1	1	1
(그림 1-b)	2	3	1	2	0.4714
(그림 1-c)	3	3	2	2	0.3536
(그림 1-d)	5	3	2	1	0.8333
(그림 2-a)	3	3	1	1	1
(그림 2-b)	4	3	2	1	0.6667
(그림 2-c)	5	3	2	1	0.8333
(그림 2-d)	6	3	2	1	1

(그림 1-a), (그림 2-a)와 (그림 2-d)는 메소드와 인스턴스 변수간에 최대 연결이 이루어지고 1개의 서로소인 부분그래프를 가지므로 CM = 1이다. (그림 1-b)와 (그림 1-c)는 서로소인 부분그래프의 개수가 2개이므로 분리가능 응집도를 갖게되고, 다른 참조그래프들보다 현저하게 낮은 응집도를 갖는다. 유사한 상황에서 (그림 1-b)가 (그림 1-c)보다 메소드와 인스턴스 변수간의 최대 연결에 가까우므로, (그림 1-b)가 (그림 1-c)보다 더 응집적이다.

4. 응집도 척도(CM)의 검증

Fenton에 의하면 소프트웨어 척도의 검증(validat-

ion)은 “척도가 주장된 속성의 적당한 숫자적 특징임을 보증하는 과정”이라 하였으며[18], 많은 연구자들이 소프트웨어 척도의 원하는 다양한 특성(property)들을 제안했다[19]. 그러한 프레임워크는 새로운 척도의 연구를 위한 지침이 되며, 또한 기존의 척도들은 이 특성들에 근거하여 정당화된다.

Briand, Morasca와 Basili에 의해 제시된 특성은 광범위한 측정 개념들, 개념 정의의 정형화, 측정 이론과의 일치성 등에서 이전의 연구들에서 나타나지 않은 분명한 장점을 갖는다[20, 21].

따라서 본 연구에서는 CM을 검증하기 위해서 Briand 등의 응집도 특성을 토대로 한다.

$G = (V, E)$ 가 임의의 클래스에 대한 참조그래프라고 할 때, Briand 등에 의해 주어진 CM이 만족해야할 특성들은 다음과 같다.

특성 1 : (Normalization)

$$Cohesion(G) \in [0, Max]$$

특성 2 : (monotonicity)

$$G_{C1} \subseteq G_{C2} \Rightarrow Cohesion(G_{C1}) \leq Cohesion(G_{C2})$$

특성 3 : (Cohesive module)

$$G_{C3} = G_{C1} \cup G_{C2} \text{ and } G_{C1} \cap G_{C2} = \emptyset \\ \Rightarrow \max\{Cohesion(G_{C1}), Cohesion(G_{C2})\} \geq \\ Cohesion(G_{C3})$$

특성 4 : (Non-negativity and Null value)

$$Cohesion(G) \geq 0 \text{ and } \\ E = \emptyset \Rightarrow Cohesion(G) = 0$$

특성 5 : (symmetry)

$$(G = (V, E) \text{ and } G^{-1} = (V, E^{-1})) \\ \Rightarrow Cohesion(G) = Cohesion(G^{-1})$$

Mehra[1]는 Weyuker[19]의 복잡도 특성들 중 응집도 특성과 관계된 다음의 특성들을 추가하였다.

특성 6 : (renaming) C_1 를 C_2 라고 이름을 바꾸어도,

$$Cohesion(G_{C1}) = Cohesion(G_{C2})$$

특성 7 : (transitive)

$$Cohesion(G_{C1}) < Cohesion(G_{C2}) \text{ 이고 } \\ Cohesion(G_{C2}) < Cohesion(G_{C3}) \text{인 } \\ \text{클래스 } C_1, C_2, C_3 \text{가 있을 때, } \\ Cohesion(G_{C1}) < Cohesion(G_{C3}) \text{이다.}$$

특성 8 : (coarse)

$$Cohesion(G_{C1}) \neq Cohesion(G_{C2}) \text{인 클래스 } C_1 \text{과}$$

C_2 가 존재한다.

Briand등이[20] 제시한 특성들과 Mehra[1]가 추가한 특성들은 응집도 척도가 만족해야할 충분조건들은 아니지만 필요 조건들이다.

특성 1)의 의미는 클래스가 응집도를 갖지 않거나 최대 응집도를 가질 수 있는 것이다. 즉 하나의 클래스의 응집도는 특히 $[0, \max]$ 구간에 있어야한다. 클래스 구성 요소들간에 관련이 없으면 응집도는 0이고, 모든 구성요소들이 최대로 관련되어 있으면 응집도는 max이다. 정규화는 응집도 값을 동일한 구간에 존재하게 함으로 서로 다른 클래스들간의 응집도 비교가 가능하도록 해준다. 메소드와 인스턴스 변수들간에 연결이 없으면 CM 은 0이고, 서로소인 부분그래프의 개수가 1이고 최대 연결이 이루어지면 $CM = 1$ 이므로, $0 \leq CM \leq 1$ 이다. 따라서 CM 은 특성 1을 만족한다.

특성 2)의 의미는 클래스 내부 관계(intra-class relationship)를 더하는 것이 클래스 응집도를 감소시킬 수 없다는 것을 의미한다. 임의의 클래스에 대한 참조 그래프에서 메소드와 인스턴스 변수간의 참조관계의 추가는 CM 의 값을 증가시키므로 CM 은 특성 2를 만족한다.

특성 3)의 의미는 두 개의 관계없는 클래스들을 묶어서 생성된 클래스의 응집도는 원래 클래스의 응집도보다 크지 않아야 한다. 관련이 없는 2개 이상의 클래스들이 합병되면, 관련 없는 요소들이 함께 캡슐화되므로 응집도는 증가할 수 없다. 서로 관련 없는 클래스의 결합을 통해 생성된 클래스는 참조 그래프에서 서로소인 부분 그래프의 개수를 증가시키므로 원래 클래스의 응집도를 크게 감소시킨다. 따라서 CM 은 특성 3을 만족한다.

특성 4)의 의미는 클래스의 응집도는 음수일 수 없다는 것이다. 최악의 경우에, 클래스의 구성요소들이 전혀 관련되지 않을 때 클래스 응집도는 0이어야 한다. 메소드와 인스턴스 변수들간에 참조관계가 없으면 $CM=0$ 이므로, CM 은 특성 4를 만족한다.

특성 5)의 의미는 응집도가 클래스의 원소들간의 관계를 표현하기 위해 선택된 표현방식에 독립적이어야 한다는 것이다. 응집도는 구성요소들의 관계를 표현하는 화살표의 방향에 관련된 표현 방식에 무관해야 한다. CM 은 메소드와 인스턴스 변수간의 관계를 표현하는

방식에 무관하므로 특성 5를 만족한다.

특성 6)의 의미는 클래스의 이름을 바꾸어도 응집도는 같아야 한다는 것이다. CM 이 클래스의 이름에 의존하는 것이 아니므로, CM 은 특성 6을 만족한다.

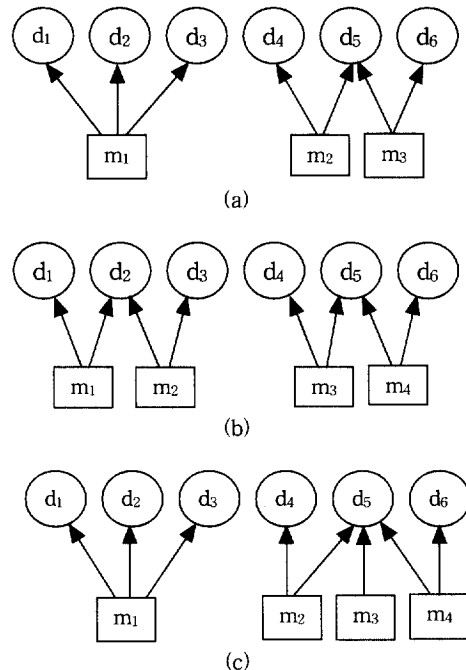
특성 7)의 의미는 클래스 응집도가 추이적 특성을 만족해야한다는 것이다. CM 은 응집도에 관한 추이적특성을 만족한다.

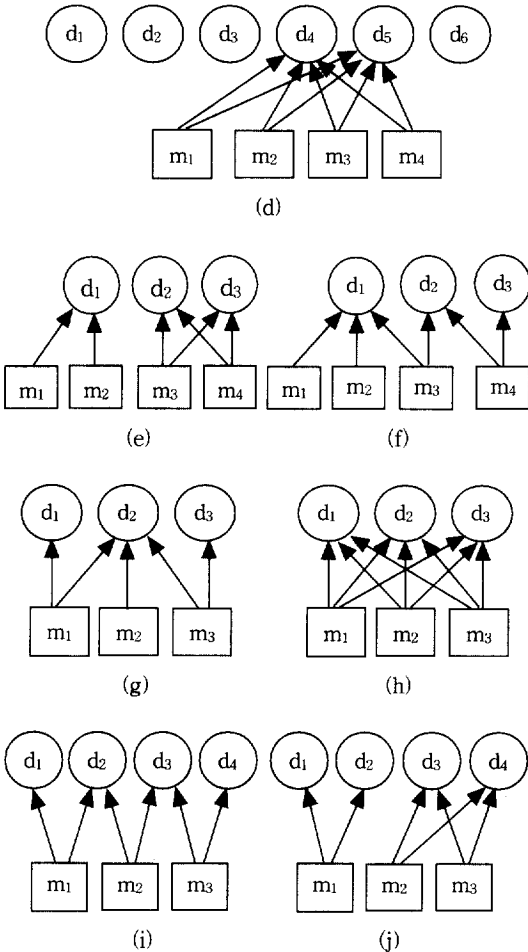
특성 8)의 의미는 응집도가 다른 클래스들이 존재해야 한다는 것이다. 즉, 두 개 이상의 클래스들을 구분할 수 있어야 한다. CM 의 값이 0에서 1까지 변할 수 있으므로, CM 은 특성 8을 만족한다.

위의 결과를 분석하면, 본 연구에서 제안한 CM 은 응집도 척도가 갖추어야할 필요 조건들은 만족한다고 할 수 있다.

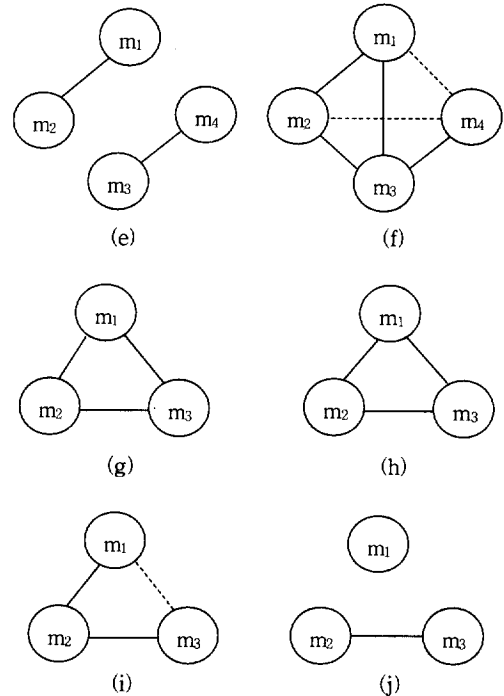
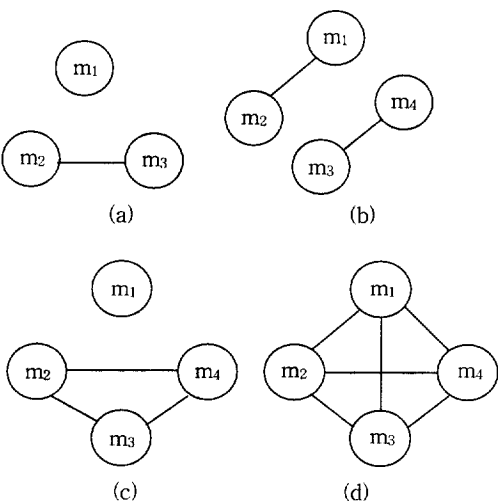
5. 기존 응집도 척도와 비교

기존 척도들과 CM 과의 관계를 (그림 4)와 (그림 5)가 보여준다. (그림 5)는 (그림 4)에 대한 메소드 연결 그래프(Method connectivity graph)[8]이며, 실선은 직접 연결, 점선은 간접 연결을 나타낸다.





(그림 4) 참조그래프



(그림 5) 메소드 연결 그래프

<표 2>는 (그림 4)의 참조그래프들에 대한 측정값들을 보여준다.

<표 2> (그림 4)에 대한 측정값

	a	b	c	d	e	f	g	h	i	j
LCOM	1	2	0	0	0	2	0	0	0	1
Hitz	LCOM	2	2	1	2	1	1	1	1	2
	C	1	0.33	0	1	0.33	0.33	1	1	0
Chen	66.6	50	50	25	50	25	33.3	33.3	33.3	66.6
TCC	0.33	0.33	0.5	1	0.33	0.66	1	1	0.66	0.33
LCC	0.33	0.33	0.5	1	0.33	1	1	1	1	0.33
CC	5.56	5.56	8.83	33.3	16.6	22.2	33.3	100	16.6	16.6
RCI	0.38	0.33	0.33	0.33	0.5	0.5	0.55	1	0.5	0.5
CM	0.27	0.23	0.23	0.15	0.35	0.5	0.55	1	0.5	0.35

Chidamber등의 LCOM[3], Hitz등의 매트릭[4, 5], Chen등의 매트릭[6], Bieman등의 척도[8]는 실질적으로 메소드 연결 그래프의 분석에, Baxter등의[7] 합성 응집도는 인스턴스 변수의 유형에, Mehra[1]의 척도들은 매트릭 데이터 슬라이스에, Park등의[9] CC, Brand의[10, 11] RCI와 본 연구에서 제안한 CM은 참조그래프의 분석에 근거하여 응집도를 측정한다.

메소드 연결 그래프 분석에 근거한 척도들은 메소드

와 인스턴스 변수들 간의 연결 개수의 차이에 따른 응집도의 차이와 인스턴스 변수와 메소드간의 연결 패턴을 구분할 수 없다.

(그림 4-a)와 (그림 4-j), (그림 4-b)와 (그림 4-e), (그림 4-g)와 (그림 4-h)는 메소드와 인스턴스 변수간의 연결 개수나 연결 패턴이 상이하지만 동일한 메소드간 연결을 가지므로, <표 2>에서 연결 그래프 기반 척도들의 측정값은 모두 동일하다.

또한 LCOM은 메소드와 참조 관계의 추가가 모순된 결과를 초래한다. 즉, (그림 4-a)에 하나의 메소드와 참조를 추가하면 (그림 4-b)와 (그림 4-c)가 되는데, (그림 4-b)의 경우는 응집도가 감소하지만 (그림 4-d)의 경우는 증가한다.

참조 그래프에 근거한 기존의 척도들은 참조그래프가 메소드간 연결을 내포하고 있으므로 메소드간의 연결, 메소드와 인스턴스 변수들간의 연결 개수의 차이에 따른 응집도의 차이를 구분할 수 있고, 메소드와 참조의 추가에 일정한 반응을 보이는데 비하여, 인스턴스 변수와 메소드간의 연결 개수가 같을 때, 메소드와 인스턴스 변수들간의 연결 패턴을 구분할 수 없다. Park등의 CC의 경우 (그림 4-b)와 (그림 4-c)와 (그림 4-d), (그림 4-e)와 (그림 4-f)를 구분할 수 있지만 (그림 4-i)와 (그림 4-j)를 구분할 수 없다. RCI의 경우는 (그림 4-b)와 (그림 4-c)와 (그림 4-d), (그림 4-e)와 (그림 4-f), (그림 4-i)와 (그림 4-j)를 구분할 수 없다. CC와 RCI가 단순히 연결 개수만을 고려하여 측정하므로 이러한 현상이 발생한다.

Park등의 CC는 또 다른 문제점을 내포하고 있다. CC가 (그림 4-b)와 (그림 4-c)와 (그림 4-d), (그림 4-e)와 (그림 4-f)를 구분할 수 있는 것은 메소드와 인스턴스 변수들간 연결의 개수가 특정 노드에 집중하면 메소드간의 연결강도(CI)가 증가하게 되어 전체 응집도가 증가하기 때문이다. 이러한 현상은 연결 강도를 측정할 때 인스턴스 변수들이 중복 계산되기 때문에 발생한다.

(그림 4-c)에서는 d_2 와 d_5 로 각각 두 개의 메소드가 연결되고, (그림 4-d)에서는 d_5 로 세 개의 메소드가 연결되므로 (그림 4-d)가 (그림 4-c)보다 CC 측정값이 더 높게 나타난다. (그림 4-d)의 경우, (그림 4-c)와 연결의 개수는 동일하지만 d_4 와 d_5 로 각각 4개의 메소드가 연결되므로 (그림 4-d)가 (그림 4-c)보다 CC 측정값이 높게 나타난다. 그러나 (그림 4-d)가 (그림 4-c)

보다 더 응집적이라고 하는 것은 응집도에 관한 직관과 일치하지 않는다.

Baxter등[7]의 합성 응집도 매트릭은 인스턴스 변수 자체가 아니라 인스턴스 변수의 유형에 근거하므로 메소드간의 연결 관계, 메소드와 인스턴스 변수들간의 연결 개수의 차이에 따른 응집도의 차이 뿐만 아니라 연결 패턴을 구분할 수 없다.

(그림 4-e)에서는 인스턴스 변수들이 동일한 유형으로 정의되었고, (그림 4-f)에서는 인스턴스 변수들이 각각 다른 3가지 유형으로 정의되었다고 하면, (그림 4-a)는 모든 인스턴스 변수들이 동일한 유형이므로 모든 메소드들이 높은 유사성을 갖게되고, 합성응집도=1이 된다. (그림 4-f)는 (그림 4-e)에 비해 메소드들이 낮은 유사성을 가지므로 합성응집도=0.721이다. 그러나 (그림 4-e)는 분리가능 응집도를 갖고, (그림 4-f)는 다면적 응집도를 갖게되므로, 실질적으로 (그림 4-f)의 응집도가 높아야 한다.

Mehra의[1] 응집도 척도들은 클래스 내에 정의된 개의 메소드에 대한 기능 응집도가 0이 되면, 메소드의 개수에 상관없이 응집도 측정값이 0이 되므로 메소드간의 연결 관계, 메소드와 인스턴스 변수간의 연결 개수의 차이에 따른 응집도의 차이 뿐만 아니라 연결 패턴을 구분할 수 없다.

본 연구에서 제안한 CM은 RCI와 Park등의 CC와 마찬가지로 참조그래프의 분석에 근거하여 응집도를 측정하므로, 메소드간의 연결, 메소드와 인스턴스 변수들간의 연결 개수의 차이에 따른 응집도의 차이를 구분할 수 있고, 메소드와 참조의 추가에 일정한 반응을 보인다.

그러나 CM은 RCI와 Park등의 CC와는 달리 메소드와 인스턴스 변수들간의 연결 패턴을 구분할 수 있다. <표 2>에서 CM은 (그림 4-i)와 (그림 4-j), (그림 4-e)와 (그림 4-f)를 구분할 수 있고, 특히 (그림 4-b)와 (그림 4-d)의 연결 패턴을 구분하여 응집도에 관한 직관과 일치하는 측정을 할 수 있다.

위의 결과에서 볼 때, 제안된 응집도 척도인 CM은 메소드와 참조의 추가에 일정한 반응을 보이며, 메소드간의 연결, 메소드와 인스턴스 변수들간의 연결 개수의 차이에 따른 응집도의 차이, 메소드와 인스턴스 변수들간의 연결 패턴을 구분하여 응집도에 관한 직관과 일치하는 매우 향상된 측정을 제공할 수 있다.

6. 결 론

본 연구에서는 클래스 응집도의 정의에서 클래스 응집도를 저해하는 요인을 분석하고, 분석된 저해요인들의 증가가 응집도의 감소를 초래한다는 응집도에 대한 보편적인 관념과 일치하는 새로운 응집도 척도를 제안하였다.

본 연구에서는 응집도를 저해하는 요인의 식별을 위해 참조 그래프를 도입하였다. 참조 그래프에 기초하여 클래스의 응집도를 저해하는 요인을 "데이터 요소들인 인스턴스 변수의 증가", "일부 데이터 요소만을 참조하는 메소드의 증가", "참조 그래프에서 서로소인 부분 그래프의 개수"로 분석하였다. 다음으로 저해 요인들을 고려한 응집도 척도, 즉 CM (Cohesion Measure)을 정의하였다.

본 연구에서 제안한 CM 이 Briand등[20]이 정의하고 Mehra[1]가 확장한 응집도 척도가 만족해야 할 필요한 특성들을 만족함을 보여 주었고, 기존 척도와 비교를 통해서 CM 이 메소드와 참조의 추가에 일정한 반응을 보이며, 메소드간의 연결, 메소드와 인스턴스 변수들간의 연결 개수의 차이에 따른 응집도의 차이, 메소드와 인스턴스 변수들간의 연결 패턴을 구분하여 응집도에 관한 직관과 일치하는 매우 향상된 측정능을 제공함을 보여주었다.

그러나 본 연구에서 제안한 CM 은 클래스 응집도를 측정할 때, 클래스의 기본 특징인 상속관계를 고려하지 않는 문제가 있다. 따라서 상속관계를 고려한 클래스 응집도 측정에 관한 연구가 요구된다.

참 고 문 헌

- [1] Bindu Mehra, "A Critique of Cohesion Measures in the Object-Oriented Paradigm," *Masters Thesis*, Department of Computer Science, Michigan Technological university, 1997.
- [2] L. M. Otto et al., "Developing Measures of Class Cohesion for Object-Oriented Software.," *7th Annual Oregon Workshop on Software Metrics*, 1995.
- [3] Shyam R. Chidamber, Chris F. Kemerer, "Towards a Metrics Suite for Object-Oriented Design," *In Proc. OOPSLA '91*, ACM, pp.197-211, 1991.
- [4] Martin Hitz, Behzad Montazeri, "Measuring coupling and cohesion in object-oriented systems," *Proc. of the International Symposium of Applied Corporate Computing(ISACC '95)*, Oct. 1995.
- [5] Martin Hitz, Behzad Montazeri, "Chidamber and Kemerer's Metrics Suite : A Measurement Theory Perspective," *IEEE Transaction On Software Engineering*, Vol. 20, No. 6, pp.267-271, 1996.
- [6] J. Y. Chen, J. F. Lu, "A new metric for object-oriented design," *Information and Software Technology*, pp.232-240, 1993.
- [7] Rich Baxter, William Chu, Sukesh Patel, "A measure for composite model cohesion," *14th ICSE*, pp.38-48, 1992.
- [8] James M. Bieman, Byung-Kyoo Kang, "Cohesion and reuse in an object-oriented paradigm," *Proc. ACM Symposium on Software Reusability (SSR-95)*, pp.259-262, 1995.
- [9] Sunghee Park et al., "Metrics Measuring Cohesion and Coupling in Object-Oriented Programs," *Journal of Korean Information Science Society*, Vol.25, No.12, pp.1779-1787, 1998.
- [10] Lionel C. Briand, S. Morasca, V. R. Basili, "Defining and Validating High-Level Design Metrics," *Technical Report CS-TR-3301-1*, University of Maryland, Dept. of Computer Science, College Park, Md., 1994.
- [11] Lionel C. Briand, S. Morasca, V. R. Basili, "Defining and Validating High-Level Design Metrics," *IEEE Transactions on Software Engineering*, Vol. 25, No.5, pp.722-743, 1999.
- [12] J. M. Bieman, L. M. Otto, "Measuring functional cohesion," *IEEE Transaction Software Engineering*, Vol.20, No.2, pp.111-124, 1994.
- [13] Rachel Harrison, Steve J. Counsell, "An Evaluation of MOOD set of object-oriented software metrics," *IEEE Transaction Software Engineering*, Vol.24, No.6, pp.491-496, June 1998.
- [14] Embley, D. W. and Woodfield, S. N., "Assessing the quality of abstract data types written in Ada," *Proc. of Phoenix Conf. on Computers & Comm.*, pp. 205-213, 1987.
- [15] Lionel C. Briand, John W. Daly, and Juergen Wuest, "A Unified Framework for Cohesion Measurement," *Proceedings of the 4th International Software Metrics Symposium (METRICS '97)*, 1997.
- [16] Samadzadeh, M. H. and Khan, S. J., "Stability,

Coupling and Cohesion of Object-Oriented Software Systems," *PROC. 22nd Ann. ACM Computer Science Conf.*, pp.312-319, 1994.

- [17] HeungSeok Chae, YongRae Kwon, "A Cohesion Measure for classes in Object-Oriented Systems," *Proceedings of the 5th International Symposium on Software Metrics*, 1998.
- [18] N. Fenton, *Software Metrics-A Rigorous Approach*, Chapman and Hall, London, 1991.
- [19] E. Weyuker, "Evaluating software complexity measure," *IEEE Transaction Software Engineering*, Vol.14, No.9, pp.1357-1356, 1988.
- [20] L. C. Briand, S. Morasca, V. R. Basili, "Property-Based Software Engineering Measurement," *IEEE Transaction Software Engineering*, Vol.22, No.1, pp.68-86, 1996.
- [21] G. Poels, B. Dedene, "Comments on Property-Based Software Engineering Measurement," *IEEE Transaction Software Engineering*, Vol.23, No.3, pp.190-195, 1997.

김 성 애

e-mail : sakim@cafe.chosun.ac.kr
 1987년 조선대학교 전자계산학과 (학사)
 1990년 조선대학교 전자계산학과 (이학석사)
 1998년~현재 조선대학교 전자계산학과 박사과정

관심분야 : 소프트웨어공학, 품질평가, 객체지향시스템

최 완 규

e-mail : wkchoi@kwangju.ac.kr
 1988년 서울대학교 종교학과 졸업(학사)
 1992년~1993년 (주)공성통신 전산실
 1993년~1995년 한양시스템 전산실

1997년 조선대학교 전자계산학과(이학석사)
 2000년 조선대학교 전자계산학과(이학박사)
 2000년~현재 광주대학교 컴퓨터전자통신공학부 전임 강사
 관심분야 : 소프트웨어 공학, 프로그래밍 언어, 객체지향 시스템, 러프집합

이 성 주

e-mail : sjlee@mail.chosun.ac.kr
 1970년 한남대학교 물리학과 (학사)
 1992년 광운대학교 전자계산학과 (이학석사)
 1998년 대구 가톨릭대학교 (이학박사)

1988년~1990년 조선대학교 전자계산소 소장
 1995년~1997년 조선대학교 정보과학대학장
 1981년~현재 조선대학교 컴퓨터공학부 교수
 관심분야 : 소프트웨어 공학, 프로그래밍 언어, 객체지향 시스템, 러프집합