

컴포넌트 Granularity 를 정의하는 수학적 인 방법에 대한 연구

이종국, 김수동

승실 대학교 컴퓨터

{jkle690@selab, sdkim@computing}.soongsil.ac.kr

A Formal Approach of Defining Measure of Component Granularity

Jong Kook Lee, Soo Dong Kim

Dept. of Computer Science, Soongsil University

요 약

소프트웨어 산업계는 컴포넌트 기반의 개발이 소프트웨어의 생산성과 품질을 향상 시킬 것으로 기대하고 있다. 최근에 많은 연구를 통해 한 시스템의 컴포넌트들은 각각 granularity 를 가지는 것이 밝혀졌다. 그러나 지금까지 제시된 많은 방법론들이 컴포넌트 granularity 를 정확하고 세부적으로 도출하지 못하고 있다. 본 연구에서는 컴포넌트의 granularity 를 도출하는 수학적 방법을 제시한다. 그리고 이론적인 연구를 통해 비즈니스 단위의 컴포넌트와 공통 컴포넌트가 분리된다는 것을 보인다. 사례 연구를 통해 우리는 우리가 제시한 granularity 의 도출 방법이 유용함을 증명하고 컴포넌트를 분리시키는 경계선이 존재한다는 것을 보인다.

1. 서론

컴포넌트 소프트웨어는 소프트웨어의 위기를 해결하는 방법으로 매우 각광 받고 있다. 현재 연구되고 있는 컴포넌트의 특성 중의 하나는 granularity 이다. granularity 는 컴포넌트가 가지는 크기를 의미한다. 컴포넌트의 granularity 가 매우 작으면 컴포넌트의 재사용성이 감소되고 granularity 가 너무 크면 컴포넌트를 사용하여 유연하게 시스템을 디자인할 수 없다. 따라서 적절한 컴포넌트의 granularity 를 결정하는 것은 매우 중요한 문제이다.

CATALYSIS 에서는 granularity 는 점진적인 방법으로 확정된다. 처음에는 가장 큰 컴포넌트를 도출하고 두번째는 가장 큰 컴포넌트는 작은 컴포넌트로 쪼갬다. Business Component Factory 에서는 컴포넌트를 정의하기 전에 미리 granularity 의 level 을 결정한다. 본 논문에서는 두개의 방법을 종합한다. 우리는 도메인별로 granularity 의 레벨을 분석하고 이 레벨에 컴포넌트를 배치한다. 우리는 granularity 레벨을 도출하는데 수학적 방법을 사용한다.

2. Minimal Function Point와 Granularity

Semantic primitive 는 컴포넌트의 의미상의 최소 단위이다. 예를 들어 업무 규칙, 데이터 저장 등은 semantic primitive 가 될 수 있다. 그러나 포트를 여는 것과 메모리를 할당하는 것은 semantic primitive 가 될 수 없다. semantic primitive 를 도출하는 최상의 방법은 use case 시나리오나 use case 설명서를 분석하는 것이다. 우리는 semantic primitive 를 minimal function point(M.F.P)라

고 부른다.

■ Definition 1

M.F.P는 업무 도메인안에서의 semantic primitive 이다.

다음 장에서 우리는 M.F.P를 도출하는 세부적인 방법을 제시한다. 다음에는 node set N 을 정의한다.

■ Definition 2

Node set $N = \{f_i \mid f_i \text{ is a M.F.P}\}$

다음에는 Node 들 간의 거리를 정의한다.

■ Definition 3

$D_{ij} = 1/\text{semantic relation of M.F.P pair}$

다음에는 edge set E 를 정의한다.

■ Definition 4

$E = \{D_{ij} \mid D_{ij} \text{ is a distance of M.F.Ps}\}$

다음에는 정의 2와 정의 4를 합쳐서 graph를 정의한다.

■ Definition 5

업무 도메인 Graph $G = G(N, E)$

G 는 업무 도메인의 구조를 정의한다. 특별히 G 는 업무 도메인 구성 요소 사이의 관계를 정의한다. 관계가 강한 업무 구성요소들은 G 에서 gluster를 이룰 것이다. G 의 gluster는 컴포넌트의 후보가 된다. 우리는 조직적으로 G 의 gluster를 도출하는 방법을 제시하려 한다.

■ Definition 6

크기 d 의 complete cluster는 G 의 subgraph이다. complete cluster의 노드는 다른 노드와는 관계가 없다. complete cluster안의 모든 노드는 서로 연결되어 있다.

■ Definition 7

semi-cluster는 G 의 subgraph이다. semi-cluster의 노드는 다른 노드들과 연결되었다. semi-cluster안의 node들 간의 거리는 d 보다

작고 semi-cluster안의 node와 다른 M.F.P 사이의 거리는 d보다 크다. semi-cluster안의 모든 node는 연결되어 있다. 그림 1은 semi-cluster의 한 예를 보여준다.

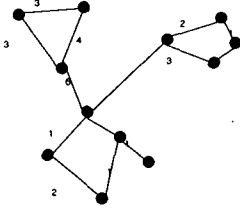


그림 1. semi-cluster의 한 예

■ Definition 8

M.F.D_i 와 M.F.D_j의 Clustering relation C_d는 M.F.D_i와 M.F.D_j가 거리 d의 한 semi-cluster나 complete-cluster에 있는 것을 의미한다.

거리 d의 Clustering relation은 다음과 같은 특징이 있다.

대칭율: M.F.D_i는 자기 자신과 clustering relation을 갖는다.

M.F.P_i, C_d, M.F.P_j

반사율: M.F.D_i가 M.F.D_j와 clustering relation을 가지고 있다면

M.F.D_j도 M.F.D_i와 clustering relation을 가지고 있다.

M.F.P_i, C_d, M.F.P_j → M.F.P_j, C_d, M.F.P_i

추이율: M.F.D_i가 M.F.D_j와 clustering relation을 가지고 있고

M.F.D_j가 M.F.D_k와 clustering relation을 가지고 있다면 M.F.D_i는

M.F.D_k와 clustering relation을 갖는다.

M.F.P_i, C_d, M.F.P_j and M.F.P_j, C_d, M.F.P_k → M.F.P_i, C_d, M.F.P_k

증명:

M.F.D_i와 M.F.D_j는 한 semi-cluster나 complete cluster에 속한다. 그 cluster에 속한 임의의 node를 M.F.D_m이라고 부르자. M.F.D_i and M.F.D_j는 한 semi-cluster나 complete cluster에 속한다. 그 cluster에 속한 임의의 node를 M.F.D_n라고 부르자. semi-cluster와 complete cluster의 정의에 의해서 M.F.D_m은 M.F.D_i와 연결되어 있고 M.F.D_n도 M.F.D_j와 연결되어 있다. 따라서 M.F.D_m은 M.F.D_n와 연결되어 있다. C_d의 정의에 의해서 M.F.D_m and M.F.D_n가 연결되어 있는 node 사이의 거리는 d보다 작아야 한다. 따라서 두 cluster의 모든 노드는 연결되어 있고 그 안의 임의의 두 node 사이의 거리는 d보다 작다. 그러므로 두 cluster는 거리 d의 한 cluster이다. 따라서 M.F.P_i, C_d, M.F.P_k이다.

반사율, 대칭율, 추이율은 G의 partition이 존재한다는 것을 암시한다. 한 partition은 semi-cluster이거나 complete cluster이다. 그림 2는 partition의 한 예를 보여준다.

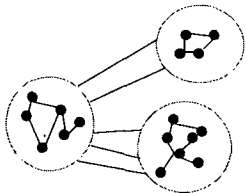


그림 2. partition의 한 예

우리가 D > d 인 거리를 사용하여 G의 partition을 구하면 D의 partition은 d의 partition을 포함한다. 이런 과정이 zoom-in,

zoom-out의 효과를 만들어 낸다. Zoom-in, zoom-out의 효과를 사용하여 업무 도메인을 분석할 수 있다. 가장 작은 partition으로 G를 나누는 것은 가장 낮은 granularity가 있다는 것을 암시하고 가장 큰 partition으로 G를 나누는 것은 가장 높은 granularity가 있다는 것을 암시한다. 한 partition의 granularity는 G의 최대 거리와 partition의 거리의 비이다.

■ Definition 9

Granularity of a partitions = distance of partitions / maximum distance of G.

4. Fine-grained component에서 coarse-grained component의 분리

이번 장에서는 fine-grained component와 coarse-grained component가 어떻게 분리되는지 생각해 본다.

■ Definition 10

G가 node 집합 N과 edge 집합 E로 이루어진 graph라고 하자.G

는 $\binom{N}{2}$ 개의 edge로 부터 랜덤하게 선택된 edge로 구성된다면 random graph라고 부른다.

■ Erdős-Renyi theorem:

만일 $E = \frac{N \log N}{2} + cN$ $N \rightarrow \infty$ 일때 G가 연결될 확률

은 $e^{-e^{-2c}}$ 에 수렴한다. 이 정리의 따름 정리는 $E < \frac{N \log N}{2}$

이면 G가 연결될 확률은 0이고 아니면 G가 연결될 확률은 1이다.

이 이론은 Erdős-Renyi[ER60]에 의해 증명되었다. 이 이론에 의해 random graph의 cluster들은 zoom-out 효과를 사용하여 cluster를 축소하면 어떤 값 이상에서 한 cluster가 될 것이다. 한 업무 도메인의 graph G에 node가 충분히 많다면 G를 random graph로 가정할 수 있다. 따라서 granularity가 높아지면 임계치가 존재하여 임계치 이상에서는 한 cluster만 나타날 것이다. 그 임계치 이상을 coarse-grained region이라고 부르고 임계치 이하를 fine-grained region이라고 부른다. 이 이론은 업무 도메인도 granularity를 기준으로 공통 영역과 순수 업무 영역으로 나누어진다는 것을 암시한다.

5. 사례 연구

사례 연구는 은행 시스템의 여신, 수신 업무이다. 표 1은 시나리오를 작성하여 나온 주요한 M.F.P이다.

표 1. 주요한 M.F.P 들

사용자 인증, bookkeeping 정보 생성, 승인 요청 정보 알람 메시지, 로그인, 로그 오프, 고객 신용 정보 조회, 수신 상담, 여신 상담, 수신 정보 출력, 고객 번호 출력, 주민등록 번호 출력, 고객 패스워드 입력, 대출요청, 신용정보에 대한 의견입력, 승인, 네트워크 트랜잭션 체크, 여신 트랜잭션 저장, 텔러 작업 마감
--

다음에 시나리오를 M.F.P의 배열로 표현한다.

표 2. 변형된 시나리오

시나리오 이름	M.F.P 배열
수신	사용자 인증, 패스워드 체크, 통장 번호 저장, 고객 정보 저장,
Loan	사용자 인증, 패스워드 체크, 여신 상담, 대출 요청, 승인, 담보 설정

다음에는 M.F.P 사이의 거리를 계산한다. 수신에서 사용자 인증과 계좌 번호 저장 사이에는 한 개의 M.F.P가 있으므로 거리는 3이다. 그런 다음 모든 시나리오에서 같은 M.F.P 쌍의 거리의 합을 계산한다. 그 합을 쌍이 나타난 횟수의 제곱으로 나눈다.

$$D_{ij} = \frac{\sum_{all\ scenario} d(M.F.P_i, M.F.P_j)}{(\# pair\ of\ M.F.P_i, M.F.P_j)^2}$$

쌍이 나타나는 횟수의 제곱으로 나눈 이유는 쌍이 많이 나타날수록 재사용성이 증가하기 때문이다. 위의 공식을 사용하여 거리를 계산한 예를 다음 표에서 제시한다.

테이블 1. 거리 계산에 대한 몇가지 예

M.F.P 쌍	거리
승인 메시지 도착했음을 알림, 승인 요청	0.25
고객 정보 프린트, 주민등록 번호 프린트	0.25
전표 프린트, 할인을 프린트	0.3333
사용 금지된 터미널 체크, 마감 여부 체크	0.5
의견 입력,	11

거리 최소값은 0.25이고 최대값은 75이다. 이 거리와 앞 장의 이론을 사용하여 partition을 구할 수 있다.

granularity 0.333 이상에서는 어떤 컴포넌트도 호출되지 않는다. 예상했던 대로 fine-grained region과 coarse-grained region이 나누어진다. coarse-grained region을 더 세분화하기 위해 M.F.P에서 fine-grained region에 속한 M.F.P를 제외하고 clustering을 다시 시작했다. 그 결과는 다음 그림과 같다.

application		0.2
deposit/loan	related to approval	0.065
related to approval	deposit counsel loan counsel/loan application	0.052
request approval print	input customer info deposit/new account create financial data common check 0.013	
request approval print	input customer info deposit/withdrawal create new account create financial data 0.026	
request approval	input customer info deposit/withdrawal, create new account 0.051	
request approval	print customer info print fee	0.033

그림 4. 수정된 granularity level

다음은 위에서 만들어진 granularity level을 고려하여 작성된 컴포넌트 diagram이다.

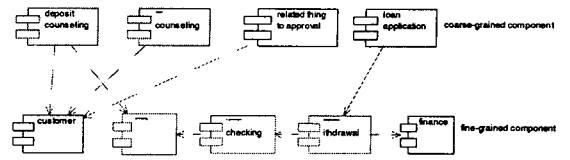


그림 5. granularity 를 고려한 컴포넌트 diagram

위의 컴포넌트 다이어그램은 비즈니스 도메인을 순수 업무 영역과 공통 영역으로 분리했다.

6. 결론

이상에서 특정 업무 도메인의 granularity level을 구하는 방법을 제시하고 사례 연구로 은행 업무를 이 granularity level을 사용하여 컴포넌트로 분리했다. 다음 연구에서는 이상의 결과를 컴포넌트 아키텍처와 통합하는 방법을 제시하겠다.

참고문헌

[Advisor] Sterling Software Component-Based Development Method, <http://www.sterling.com/cool>

[Amm99] Yacoub S, Ammar H, Mili A: Characterizing a Software Component, 1999 Workshop on Component-Based Software Engineering

[Digre98] Digre T: Business Object Component Architecture, *IEEE Software* Vol.15, No.5, September/October 1998

[ER60] P.Erdős and A.Reny. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:569-573,1960.

[Her2000] Herzum P and Sims O, *Business Component Factory*, OMG, 2000

[Kang99] Kang K: Issues in Component-Based Software Engineering, 1999 International Workshop on Component-Based Software Engineering

[Lee99] Lee, Sang Duck, Yang, Young Jong, Cho, Eun Sook, Kim, Soo Dong, "COMO : A UML-Based Component Development Methodology", *Asia-Pacific Software Engineering Conference(APSEC99)*, Takamachu, Japan, PP. 54 - 61, Dec. 7-10, 1999.

[Szy98] Szyperski C., *Component Software: Beyond Object-Oriented Programming*, Addison Wesley Longman, Reading, Mass., 1998

[Sou98] D'souza D. F. and Wills A. C., *Objects, Components, Components with UML*, Addison-Wesley, 1998.