

Composition과 Partition을 이용한 클러스터링 방법

김중대, 최은만

{bskylark, emchoi}@cakra.dongguk.ac.kr

동국대학교 컴퓨터공학과

Clustering Method Using Composition and Partition

Jong-Dae Kim, Eun-Man Choi

Dept. of Computer Engineering, Dongguk University

요약

최근 시스템 환경의 급격한 변화에 따라 기존 시스템의 유지보수와 재사용에 관련된 여러 가지 문제점이 발견되었다. 객체지향개념은 우리에게 재사용과 유지보수를 용이하게 해줄 것으로 생각되었지만 기대만큼의 효과는 거두지 못하고 있다. 이러한 현실적인 문제를 해결하기 위해 많은 클래스들을 관련성을 이용해 패키지로 분류하는 클러스터링 기법들이 연구되었다. 대부분이 소프트웨어의 품질을 평가하기 위한 매트릭스를 기반으로 한 것이었지만, 실제 현장에서 개발자들의 경험에 의한 클러스터링 기법과는 많은 차이가 발생한다. 본 논문에서는 매트릭스를 이용한 Composition 방법과 개발자들의 경험을 이용한 Partition 방법을 혼합하여 정량화된 수치를 제공하며 개발자들의 경험을 충족시키는 방법을 제시하고자 한다.

1. 서론

네트워크 환경이 보편화되면서 객체지향 개념이 적용된 시스템 중에도 환경에 적응하기 위한 유지보수가 필요하였다. 또한 산업계에서도 하나의 제품을 재사용하여 여러 소비자에게 제공할 경우 높은 신뢰성과 생산성을 올릴 수 있다. 그러나 이러한 기대를 갖고 객체지향 개념과 방법론을 사용한 경우에도 많은 클래스들의 분류가 어려워져 재사용이 쉽지 않다. 특히 개발자의 경험에 의해 만들어진 패키지의 경우 내부의 구성과 연관성, 효율성에 대한 문제점으로 인해 이해가 쉽거나 효율적인 구성은 불가능하였다. 이를 해결하고자 소프트웨어 매트릭스를 이용하여 정량화된 결과를 얻고 이를 클러스터링에 이용하는 방법이 제시되었으나 개발자들의 실질적인 이용과는 거리가 멀었다. 개발자들의 경우 클래스들을 클러스터링 하는 과정은 클래스 다이어그램을 만드는 도중에 관련 클래스들을 묶어서 하나의 패키지에 포함시키는 과정과 패키지를 새로 생성해서 내부에 클래스들을 생성시키는 과정을 거치는 두 가지 방법을 혼용하고 있다. 이러한 개발자들의 패턴을 소프트웨어 매트릭스를 이용한 클러스터링 기법과 통합하여 제시하려는 것이 본 논문의 목표이다.

2. 관련연구

2.1 클러스터링 방법의 분류

UML과 모델링 도구를 설명하는 많은 책들에서 발견되는 것이 관련성 있는 클래스들을 패키지로 묶어내는 작업을 시행하는 것이다. 이러한 패키지들은 일반적으로 인터페이스와 관련된 부분과 데이터 베이스에 관련된 부분, 그리고 실제 시

스템 내부와 관련된 부분 등으로 나누어진다. [1]에서는 Package를 대상시스템의 환경에 관한 것, Actor의 정보와 관련된 것, Actor와의 인터페이스에 관련된 것으로 나누고 있다. 이렇게 나누어진 각각의 패키지 내부에 클래스들을 배치하여 패키지 다이어그램을 완성한다. 이러한 방법은 시스템을 3개 이상의 계층으로 나누어 유지보수가 간편하다는 장점을 갖는다.

소프트웨어 매트릭스를 이용하여 클래스들을 클러스터링 하는 방법은 크게 두 가지로 요약된다. 하나는 Composition 방법인데, [2]에서는 각각의 클러스터에 seed 클래스를 선택하고 이를 중심으로 주변의 나머지 부분에 대해 확장하는 방법을 사용하였다. 두 번째로 Partition 방법은 [7]에서 제시된 방법으로 하나의 패키지가 모든 클래스를 포함하게 하고 이를 단계별로 분할하여 가장 효율적인 형태를 생성하는 것이다. 두 가지 모두 장단점을 가지고 있는데 첫 번째 방법의 경우 전체적 복잡도가 고르게 분배되는 효과를 가지고 있으나 클러스터링을 중단시키는 임계값의 결정이 개발자의 주관적 의지에 의존해 있으며 두 번째 방법의 경우 Task의 집합이 한쪽에 편향되어있는 경우 복잡도가 끝고루 분산되지 못해서 하나의 프로세서에 복잡도가 편중될 가능성이 있다.

2.2 결합도 측정 소프트웨어 매트릭스

[표 1]에서는 결합도를 측정하는 매트릭스들이 많은 측정상의 차이를 내포하고 있음을 보여준다. 결합도(Coupling)에 대한 연구는 Chidamber와 Kemerer[6]가 객체지향 매트릭스 집합 중 CBO, RFC를 제시한 이후로 이에 대한 실험과 연구

는 꾸준히 이루어져 왔다. CBO는 메소드 호출과 클래스간의 attribute 참조가 이루어진 경우, 관련된 클래스 수를 측정하는 방법을 제시하였고 RFC는 호출된 메소드의 개수를 측정하는 방법을 사용하였다. [9]에서 제시된 MPC의 경우 하나의 클래스 내에서 다른 클래스의 메소드를 호출하는 총 개수를 측정하는 방법을 사용하였고, Ce는 외부클래스들과 연관된 내부 클래스 개수를 측정하고 Ca는 내부 클래스들과 연관된 외부 클래스 개수를 측정하였다. DAC는 한 클래스 내부에 정의된 다른 클래스 타입의 어트리뷰트 개수를 측정하여 결합도를 측정하였다. [8]에서는 클래스와 객체를 구별하여 결합도를 측정하는 방법을 제시하고 있으며, 또한 [2]에서 제시한 Coupling이라는 측정법의 경우 클래스들간의 관계에서 가중치를 정의하고 상호간의 메시지 수에 곱하여 측정한다.

메트릭스	측정방법	간접결합	용도	참고문헌
CBO	#결합클래스	X	효율성, 재사용성	[6]
RFC	#호출된메소드	O	이해용이성, 유지보수	[6]
MPC	#메소드호출	X	이해용이성, 유지보수	[9]
Ce	#결합내부클래스	X	유지보수	[5]
Ca	#결합외부클래스	X	유지보수	[5]
DAC	#어트리뷰트	X	재사용성	[9]
Coupling	#메소드호출 X 가중치	X	효율성	[2]
Affinities	동기화호출값+ 비동기화호출값	X	효율성, 유지보수성	[7]

[표 1] 결합도와 관련된 메트릭스

2.3 덩이화(Chunking) 이론

[10]에서는 프로그램을 이해하는 과정을 작은 단위에서 공통된 성질을 추출하여 추상화하는 것을 반복함으로써 이루어진다고 설명하고 있다. 또한 단기기억 속에 관리될 수 있는 한계수로 G.Miller의 Hrair Limit 이론에서 제시된 7±2를 사용하여 실험을 실시하여 문제해결 활동 전반에 적용될 수 있음을 보였다. 본 논문에서는 이렇게 제시된 한계 수를 이용하여 패키지 내부의 클래스 개수를 제한할 것이다.

3. 클러스터링 알고리즘

본 논문에서는 다음과 같은 방법으로 클러스터링을 실시할 것이다.

단계 1> 유스케이스 다이어그램에서 액터를 추출.

단계 2> 액터 정보를 관리하는 클래스와 액터와 시스템이 상호작용하는 클래스를 추출하여 각각 하나의 패키지에 포함

단계 3> 나머지 클래스를 하나의 패키지로 포함

단계 4> 위에서 나눈 3개의 패키지 내부에서 상속관계의 클래스들이 발견되는 경우 패키지로 묶어줌

단계 5> 터미널 클래스가 존재하는 경우 결합도를 고려하여 클러스터링

단계 6> 집단화, 컴포지션, 일관참조관계인 경우 클래스

사이의 의존관계를 결합도 메트릭스를 사용하여 정량화하여 가장 큰 의존관계를 갖는 경우부터 차례로 클러스터링

단계 7> 같은 결합도 값을 갖는 경우 LCOC값이 적거나 내부 클래스 값이 적은 패키지로 클러스터링

단계 8> 단계 6, 7을 실행할 때 LCOC값을 구하여 클러스터링 가부를 결정

단계 9> 모든 클래스가 패키지로 묶일 때까지 단계 4부터 8까지를 반복

위와 같은 알고리즘을 통하여 클러스터링을 실시하는 경우 전체 시스템은 3개의 기본 패키지로 나뉘게되어 3계층으로 관리할 수 있고 패키지 내부를 메트릭스를 이용하여 클러스터링함으로써 정량적인 결과를 제시할 수 있다.

위의 알고리즘에서 사용되는 결합도와 응집도에 관한 메트릭스는 다음과 같이 정의하도록 한다.

결합도 = 메시지 수 * 가중치
 응집도 결여도(LCOC) = $|P| - |Q|$, 단 $|P| > |Q|$
 = 0, $|P| \leq |Q|$

결합도를 구하는 메트릭스에서 메시지 수는 해당 클래스에 포함된 모든 메소드가 한번씩 실행될 경우 접근되는 다른 클래스의 메소드의 개수를 측정하여 정하며 가중치는 클러스터링의 우선 순위를 나타내는 것으로 컴포지션, 집단화, 일반 참조의 순서로 가중치를 부여한다.

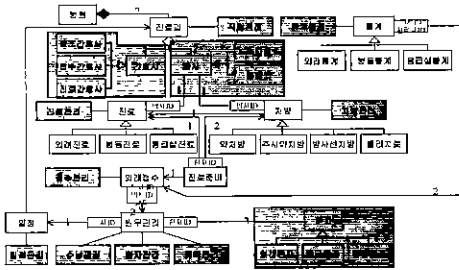
응집도를 구하는 메트릭스는 [6]에서 제시한 LCOM을 변형한 것으로써 클래스를 패키지로, 메소드를 클래스로, 인스턴스 변수를 메소드로 확장함으로써 클래스 내부의 LCOM을 구하는 것을 패키지 단위로 확장한 모습이다. 이렇게 클래스에 대한 LCOC를 사용한 이유는 Hrair Limit 이론에서 제시된 7±2란 수를 임계값으로 사용하기 위해서이다. 즉 패키지 내부를 구성하는 단위가 5개 이상인 경우 상관관계의 강약에 따라 개수를 제한하기 위한 방법이다. 즉, 패키지를 이루는 단위의 개수를 노드로 표현하고 개수를 n개라고 하고 각 노드간의 상관관계가 있는 경우를 링크로 표현하면 최소 개수는 n-1개가 된다. 그런데 모든 노드들의 조합의 개수는 n(n-1)/2이므로 여기서 상관관계가 있는 링크의 개수를 빼면 상관관계가 없는 링크의 개수를 구할 수 있다. 그러므로 상관관계가 없는 링크의 수에서 상관관계가 있는 링크의 수를 빼면 그 결과에 따라 응집도를 구할 수 있다. 예를 들어 응집도 결여도가 0이 되기 위해서는 노드가 5인 경우 최소 5개 이상의 상관관계 조합이 존재해야한다. 그러나 5개의 노드가 하나의 패키지가 되는 경우 최소 상관관계의 개수는 4개이므로 최소 조합 이외의 다른 상관관계가 있지 않는 경우 5개의 노드가 하나의 패키지에 들어가지 못한다. 이러한 원리를 이용하여 패키지 내부의 단위 개수를 제한할 것이다.

위에서 제시한 알고리즘과 메트릭스를 이용하는 경우 앞에서 기본적으로 나눈 3개의 패키지 각각을 하나의 도메인으로 생각하고 각 도메인 상에서 결합도가 큰 순서대로 클래스들을 클러스터링 한다. 이러한 과정을 반복할 때, 제시된 응집

도 결여도 매트릭스를 이용하여 단기기억에서 관리할 수 있는 개수를 넘어서지 않게 제어함으로써 최악의 경우 하나의 패키지로 묶이는 상황을 미연에 방지하고 부하를 분산시킬 수 있으며 유지보수 시 시스템을 이해하는 것을 효율적으로 할 수 있다.

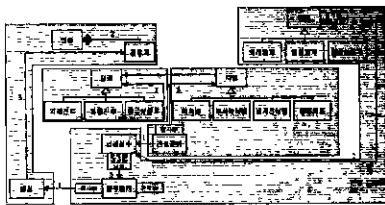
4. 실험사례

실험에 사용될 병원관리 시스템은 외래업무를 기준으로 작성되었지만 병동업무나 응급실 업무를 추가할 경우를 대비하여 설계하였다.



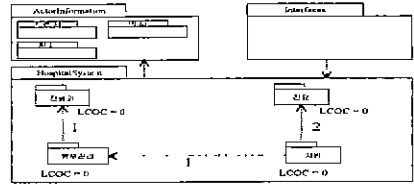
[그림 1] 실험에 적용되는 클래스 다이어그램

[그림 1]에서 액터의 정보와 관련된 클래스들인 환자, 의사, 간호사와 이들에게서 상속받은 클래스 모두를 ActorInformation 이라는 패키지에 넣는다. UI와 관련된 클래스들로는 직원명부등록, 통계자료확인, 진료내용작성, 진료접수작성, 처방전작성, 일정입력, 수납자료처리, 예약진료관리, 환자자료작성 등인데, 이들을 Interface라는 패키지에 넣는다. 실제 시스템과 관련된 부분은 HospitalSystem이라는 패키지로서 진료, 처방, 외래접수, 진료준비, 원무관리, 일정관리, 통계, 진료과, 병원에 관련된 클래스들이 포함된다.



[그림 2] HospitalSystem Package

[그림 2]가 HospitalSystem 패키지 내부의 클러스터링 결과인데 각 클래스간의 메시지 개수를 측정하여 가중치를 곱하여 결합도 값을 구하였고 LCOC 값을 기준으로 패키지 내부의 클래스 개수를 제한하였다. [그림 2]는 LCOC값이 0인 경우에 한해 클러스터링 한 결과이다 따라서 LCOC값을 정해주는 것에 따라 클러스터링 결과는 달라질 수 있으나 본 논문에서는 LCOC 값이 0인 경우 한계값을 잘 정의할 수 있으므로 표준값으로 설정하였다. 패키지 다이어그램은 [그림 3]과 같다



[그림 3] 패키지 다이어그램

5. 결론 및 향후과제

재사용과 유지보수는 생산성 향상과 급변하게 변화하는 환경에 기존의 시스템을 맞추어 나가는데 필수적인 요소이다. 본 논문에서는 분할과 결합의 원리를 혼합하여 실무 개발자들이 이용할 수 있는 방법을 제시하였다. 또한 패키지 내부의 복잡도를 조정하여 시스템의 이해에도 도움을 주었다.

향후과제는 본 클러스터링 연구를 바탕으로 자동화된 도구를 만들어 내는 것과 사용하고자 하는 용도에 따라 기존의 클래스 다이어그램을 패키지로 클러스터링을 실시하는 도구를 개발하려고 한다. 이러한 연구는 컴포넌트 개발을 자동화하는데 도움이 될 것이다.

참고문헌

- [1] Terry Quatrani, Visual Modeling with Rational Rose and UML, Addison-Wesley, 1998
- [2] 김동관 외 1인, "클라이언트 서버 및 분산 S/W 개발을 위한 객체 클러스터링 기법", 정보과학회논문지(B), 1998
- [3] Martin Fowler 외 1인, UML Distilled, Addison-Wesley, 1997
- [4] 김종대. 최은만, "UML을 이용한 병원관리 시스템의 설계", 동국대학교 대학원 연구논집, 동국대학교 대학원, 1998
- [5] Lionel C. Briand 외 2인, "A Unified Framework for Coupling Measurement in Object-Oriented Systems", IEEE TSE Vol. 25. No 1, January/February 1999
- [6] Chudamber, Kemercer, "A Metrics Suite for Object-Oriented Design", IEEE TSE, Vol 20, No 6, 1994
- [7] Umesh Bellur 외 2인, "Clustering : Composition for Active Object Systems", In Proc. 27th Hawaii Int'l Conf System Science, January 1994
- [8] Martin Hitz and Behzad Montazeri, "Measuring Product Attributes of Object-Oriented Systems", Object Currents Vol.1 No 4, April 1996
- [9] W. Li and S. Henry, "Object-Oriented Metrics that Predict Maintainability," J Systems and Software, Vol. 23, no 2, pp. 111-122, 1993
- [10] 문양선, 유철중, 장옥배, "인지심리 이론을 반영한 객체 지향 설계 및 프로그래밍 스타일 지침", 정보과학회논문지(B) Vol. 25 No. 3, 1998