

절차지향 SW로부터의 클래스와 상속성 추출에 관한 연구

최정란, 박성욱, 이문근
전북대학교 컴퓨터 과학과

e-mail : {jlchai, sopark, mklee}@cs.chonbuk.ac.kr

A Study on Class and Hierarchy Extraction from Procedural SW

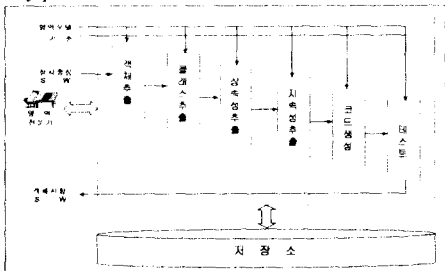
Joeng-ran Choi, Sung-og Park, Moon-kun Lee
Dept. of Computer Science, Chonbuk National University

요 약

절차지향 SW를 객체지향 SW로 변환하면 SW 유지·보수 비용을 수 절감할 수 있으며, 시스템에 새로운 요구를 수용할 수 있게 되는 등 많은 장점을 가지게 된다. 본 논문은 객체지향 SW로 변환과정 중 필요한 클래스와 상속성 추출 방법을 제안하였다. 클래스와 상속성 추출 때 관계를 고려한 의미적인 방법을 제시하였으며 영역 모델과의 비교시 의미적이며 통계적인 방법을 사용하였다. 논문에서 제안한 방법을 사용함으로써 영역 전문가에게 다양한 선택 기회를 제공할 수 있어 의미 있는 코드 생성에 도움을 줄 수 있다.

1. 서론

소프트웨어는 제작된 후 지속적인 유지·보수가 필요하다. 이러한 흐름에 맞추어 시스템을 재개발할 때 유지·보수 및 재사용 측면에서 기존의 방식들보다 유리한 객체지향 파라다임을 적용하여 기존의 시스템으로 재개발한다면, 소프트웨어 생산성을 향상시킬 수 있고, 소프트웨어의 유지·보수 비용을 절감할 수 있으며, 시스템에 새로운 요구를 수용할 수 있게 되는 등 많은 장점을 가지게 된다[3].



<그림 1> 절차중심 SW의 객체중심 SW로의 재공학 절차 흐름도

일반적으로 재공학 방법은 <그림 1>과 같은 절차로 진행되는 것이 바람직하다. 객체 추출 단계에서는 전역 변수, 사용자 자료형(UDT, User-defined Type), 함수와 파라미터를 기반으로 관련성 정도를 기준으로 클러스터링한다. 클래스 추출 단계에서는 클러스터링된 객체 후보들의 공통적인 특성을 추출하여 클래스를 추출한다. 클래스 추출 단계의 결과는 클래스들간에 대등한 평면 관계를 이루고 있다. 상속관계 추출 단계는 전 단계에서 추출된 평면화된 클래스들의 공통적 특성을 추출하여 part-of 관계나 is-a 관계를 추출하여 계층 구조를 만든다. 지속성 추출 단계에서는 절차지향 프로그램에는 존재하지 않는 객체 지향 특성들과 동적인 부분을 추가한다. 이러한 것들은 클래스의 생성자, 소멸자, 동적 메모리 할당/해제 등이 있다. 코드 생성 단계에서는 전단계에서 생성된 뼈대 코드(skeleton code)를 기반으로 실행 가능한 객체 지향 프로그램을 생성한다. 테스트 단계에서는 생성된 객체지향 프로그램이 정상적으로 작동하는지와 원래의 절차지향 프로그램과

를 한다.

본 논문에서는 객체지향 프로그램으로 변환과정 중 두 번째 단계와 세 번째 단계인 클래스 추출과 상속성을 추출한다. 첫 번째 단계인 객체 추출은 [4]의 방법을 기준으로 사용한다. 클래스 추출 단계에서는 클래스와 이들간의 part-of 관계를 추출하며 상속성 추출 단계에서는 상속성(is-a 관계)을 추출한다.

본 논문의 목적은 영역 전문가에게 영역 모델과 가장 유사도가 높은 클래스 및 상속성을 제시하는 것이 아니라, 비록 최적은 아니라도 다양하게 존재하는 클래스를 제시하여 영역 전문가가 다양한 선택 기회를 제시할 수 있도록 하는 것이다.

2. 관련연구

CORET에서는 절차 지향 프로그램으로부터 객체 관계 다이어그램(ERD, Entity-Relationship Diagram)을 생성하여 객체간의 상속성을 추출하였다. 생성된 ERD의 객체의 관계를 구별하기 위해 두 가지 관계(is-a, part-of)를 정의하였다[1, 2].

영역 모델(CORET에서는 FooAM)은 클래스 및 클래스간의 관계로 이루어지며 영역 전문가에 의하여 생성된다. 절차지향 프로그램으로부터 생성된 RooAM과 FooAM은 사상 관계를 통하여 유사도를 측정한다. 두 모델의 비교는 구조적 유사성 추출, 속성의 추출 및 메소드 추출하는 순서로 진행된다. 이러한 세 가지 과정을 거치며 소속이 모호한 부분은 영역 전문가에 의하여 결정된다. 클래스 추출 및 상속성 추출의 기준은 FooAM을 사용하였다[1, 2].

3. 클래스 추출 및 상속성 추출

클래스 추출 및 상속성 추출에서는 객체 추출단계에서 생성된 영역 비순환 그래프(G^{RA})[4]를 계속적으로 사용한다.

3.1 클래스 추출

클래스를 추출하는 경우는 크게 두 분류로 구분할 수 있다. 첫 번째 경우는 영역 모델의 클래스를 이용하여 다음 단계로 계속 진행하는 것이다. 두 번째 경우는 원시 프로그램으로부터 추출된 객체 후보로부터 클래스를 추출하여 다음 단계로 계속 진행하는 것이다. 어떠한 모델을 이용하는가에 따라 각기 장단점이 있다. 원시 프로그램으로부터 추출된 객체 후보를 이용하여 클래스를 추출할 경우 영역 전문가에게 프로그램의 프로토타입만을 제공하는 것만으로도 객체 지향 프로그램 작성에 비용을 줄일 수 있을 것이다. 그뿐만 아니라 원시 프로그램이 객체 지향 프로그램과 1:1로 매핑되며 영역 전문가가 미세한 부분만을 수정한다면 프로토타입만을 제공하는 것보다 비용을 줄일 수 있을 것이다.

본 연구는 한국 과학재단 특정기초연구(과제번호 1999-2-2-303-003-3) 지원으로 수행되었음

변환된 객체지향 프로그램이 의미상으로 동등한지에 대한 검사

논문에서는 영역 전문가에게 1개의 선택이 아닌 다양한 선택 기회를 주어 정확한 프로그램 변환을 유도할 수 있기 때문에 원시 프로그램으로부터 클래스를 추출하였다.

클래스 추출 절차는 객체 추출 단계의 결과 생성되는 G^{RA} 로부터 클래스 후보를 추출, 클래스의 속성을 추출하고 클래스의 메소드를 추출하는 순서로 진행된다.

3.1.1 역 비순환 그래프와 클래스 후보

클래스 후보를 추출할 때, 원칙적으로 1개의 객체후보(OC, Object Candidate)가 1개의 클래스 후보로 된다. 이것은 G^{RA} 를 생성할 때, 다양한 경우로 클러스터링을 하였고 클러스터링 된 1개의 단위(즉, OC)가 1개의 클래스로 될 것이라 가정했기 때문이다. 어떠한 계층에서 1개의 OC가 1개의 클래스 후보(CC, Class Candidate)가 될지라도 하위 계층에서는 다수의 클래스 후보가 될 수 있기 때문에 1개의 OC는 1개의 클래스 후보가 된다.

UDT가 내부에 존재할 경우 각각의 UDT도 클래스 후보로 추출한다. 이것은 절차 지향 프로그램의 UDT는 객체지향 프로그램에 존재하는 메소드가 없이 속성만을 가진 클래스와 유사하기 때문이다. 1개의 OC로부터 클래스를 추출하는 기준은 다음과 같다.

- UDT가 존재할 경우 1개의 UDT는 1개의 클래스 후보로 존재
- UDT와 관련된 함수를 제외한 나머지 부분이 1개의 클래스 후보로 존재

따라서 UDT가 n 개이면 클래스 후보는 최대 $n+1$ 개가 나오며 최소 n 개가 나온다. 1개의 OC로부터 생성되는 클래스의 모임을 CCs(Class Candidates)라 부르기로 한다.

OC 전체와 관련된 클래스 후보를 추출한 후 클래스 이름을 명명하는 문제가 남아있다. 본 논문에서는 영역 전문가가 필요할 경우 클래스 이름을 명명한다고 가정한다.

3.1.2 속성의 추출

3.1.1절에서 추출된 CC는 UDT로 추출된 CC 후보와 OC로부터 추출된 CC의 2종류가 존재한다는 것을 알 수 있다. UDT로부터 추출된 클래스의 속성은 UDT가 가지고 있는 내부 변수를 사용할 수 있기 때문에 해결해야할 문제가 없다. 그러나 OC로부터 추출된 클래스 후보는 속성을 추출하기 위해서 별도의 작업이 필요하다.

OC에서 UDT와 함수를 제외하면 OC에는 전역변수만이 존재하거나 존재하지 않을 수 있다. 전역 변수가 존재할 경우 OC에 존재하는 모든 전역 변수가 OC로부터 추출된 클래스의 속성이 된다.

클래스의 속성을 추출한 결과를 보면 클래스 사이에 존재하는 part-of 관계가 존재한다는 것을 발견할 수 있다.

3.1.3 메소드의 추출

속성을 추출한 후 OC에 남아 있는 부분은 함수이다. 남아있는 함수는 OC 전체 클래스의 메소드로 존재하든지 UDT의 클래스의 메소드로 존재해야 한다. 메소드의 클래스에 대한 소속의 결정은 OC 내부에 존재하는 관계성(에지)의 종류와 방향성을 이용하면 된다.

노드에 따른 에지의 종류와 방향성을 생각하면 OC에는 총 7가지의 에지가 있고 함수와 관련된 에지는 6가지가 존재한다. 이들 6가지 에지의 의미를 이용하여 메소드의 소속을 결정한다. 다음은 6가지 에지와 의미, 어떠한 클래스에 포함될 것인가에 대한 것이다.

- 1) $\langle N_{func}, N_{var}, FORWARD \rangle_{memory}$: 변수의 값을 사용하는 것으로 N_{var} 이 소속된 곳에 N_{spec} 이 메소드로 포함된다.
- 2) $\langle N_{func}, N_{var}, BACKWARD \rangle_{memory}$: 변수의 값을 기록하는 것으로 N_{var} 이 소속된 곳에 N_{spec} 이 메소드로 포함된다.
- 3) $\langle N_{type}, N_{func}, BACKWARD \rangle_{type}$: 함수가 UDT(N_{type})를 반환하

는 것이기 때문에 N_{func} 가 N_{type} 의 메소드로 포함된다.

- 4) $\langle N_{func1}, N_{func2}, BACKWARD \rangle_{call}$: N_{func1} 이 다른 함수(N_{func2})로부터 호출되는 것으로, N_{func2} 가 포함된 클래스에 N_{func1} 도 클래스의 메소드로 포함된다.
 - 5) $\langle N_{func1}, N_{func2}, FORWARD \rangle_{call}$: N_{func1} 이 N_{func2} 를 호출하는 것으로 N_{func2} 가 소속된 클래스에 N_{func1} 이 메소드로 소속된다.
 - 6) $\langle N_{type}, N_{func}, FORWARD \rangle_{type}$: 함수에서 N_{type} 의 변수를 지역 변수로 사용하는 것이므로 N_{type} 으로 생성된 클래스가 아닌 다른 클래스의 메소드로 포함되어야 한다.
- 6가지 예지는 순차적으로 적용해야 한다. 이것은 각각의 예지는 함수가 객체 지향 프로그램의 메소드로 변경될 때 특별한 의미를 표현하기 때문이다.

3.2 상속성 추출

절차 중심 SW를 객체 중심 SW로의 재공학 절차의 세 번째 단계는 상속성 추출이다. 클래스를 추출과 마찬가지로 상속성 추출도 크게 두 분류로 구분할 수 있다. 본 논문에서는 원시 프로그램으로부터 추출된 클래스를 기준으로 삼는다.

상속성 추출 과정은 클래스 추출 과정과 마찬가지로 상속성을 구할 대상 클래스의 결정, 상속될 속성의 결정과 상속될 메소드의 결정 순서로 이루어진다.

상속성 추출에는 영역 전문가의 개입이 필요하다. 첫째, 클래스들간의 유사도를 비교하기 위한 기준을 영역 전문가가 선정하여야 한다. 둘째, 상속 구조를 가진 영역 모델링과 상속성을 가진 클래스 구조를 비교하여 유사도가 측정될 경우 높은 유사도를 가진다고 잘 변환된 프로그램이라 할 수만은 없다. 이것은 객체 추출 단계의 결정 과정에서 마찬가지로 상속성을 가진 영역 모델링이 원시 프로그램보다 세밀한 부분에 대한 정보를 가지고 있지 않기 때문이다. 영역 모델링은 단지 개략적인 정보만을 가지고 있다. 따라서 상속성을 가진 영역 모델링이 결정하지 못하는 부분은 영역 전문가가 결정하여야 한다.

3.2.1 상속성 추출 대상의 선정

객체 지향 프로그램에서 상속성을 가진 클래스들은 모듈화가 잘 된 절차지향 프로그램이라면 응집도는 높지만 결합도가 낮은 것이다. 연관 관계의 많고 적음에 따라 상속성을 추출한다는 것은 논란의 여지가 많다. 본 논문에서는 클래스 후보들간의 연관 관계를 고려하지 않고 모든 클래스 후보들간의 상속성을 추출하도록 한다.

상속성을 추출할 때 첫 번째 과정은 상속성의 추출을 하기 위한 비교 대상에 관한 것이다. CCs와 CCs 사이의 상속성을 추출하는 것은 G^{RA} 의 전체적인 측면에서 본다면 CCs 내부에서 상속성을 추출하는 것과 동일한 결과를 가져온다. 따라서 본 논문에서는 CCs와 CCs 사이에서 추출하는 방법을 사용하도록 한다.

상속성 추출은 두 개의 클래스를 비교하여 공통된 부분을 찾아 공통된 부분이 상위 클래스로 나머지부분이 각각의 하위 클래스로 된다.

클래스들간의 유사도의 수치를 이용하여 상속성을 추출할 클래스들이 정해지면 속성과 메소드가 채워지지 않은 임시 클래스를 생성한다. 이것은 영역 전문가가 임시 클래스의 이름을 명명함으로써 클래스에 의미를 부여할 수 있다.

3.2.2 상속될 속성의 추출

상속성을 추출할 때 두 번째 과정은 두 클래스 사이의 유사도의 정도이다. 이들 클래스 사이의 유사도는 객체 추출의 결정 단계에서 적용한 DSS(Degree of Signature Similarity) 공식[4]을 사용한다. 두 클래스 사이의 유사도가 높은 것에서 상속성을 추출할 수도 있고 반대로 유사도가 낮은 것에서 추출할 수도 있다. 단지 차이가 있다면 높은 유사도를 가진다는 것은 상위 클래스로 존재

할 부분이 많아진다는 것이고, 반대로 낮은 유사도를 가진다면 상위 클래스로 존재할 부분이 적어진다는 것이다. 낮은 유사도가 존재하여도 상속성이 존재할 수 있다. 이러한 경우는 아마도 일반적으로 여러 단계의 상속 과정을 거친 경우일 것이다. 따라서 직접적인 상속 관계를 가질 경우는 유사도가 높을 것이다. 따라서 두 클래스 사이의 유사도가 영역 전문가가 결정하는 임계값을 초과하는 것을 대상으로 한다.

3.2.3 상속될 메소드의 추출

속성들의 공통된 부분에 대한 상속성을 추출한 후 메소드에 대한 상속성을 추출한다. 메소드의 상속성은 상위 클래스에 소속될 속성들만을 사용하는 메소드로 한정한다. 만일 상위 클래스로 소속될 속성들만이 아닌 하위 클래스의 속성을 사용한다면 하위 클래스의 메소드가 되어야 한다.

3.3 영역 모델링과의 비교

이 절의 목적은 전과정에서 생성된 상속관계를 가진 클래스들과 상속성이 존재하는 영역모델링(G^{HM})과의 유사도를 측정하여 영역 전문가에게 적합한 클래스를 선택할 수 있도록 제시하는 것이다. 이러한 유사도 측정은 클래스들 사이의 유사도와 클래스들 사이에 관계(is-a와 part-of)에 대한 유사도를 함께 생각해야 한다. 왜냐하면 객체 지향 프로그램에서 관계성에 대한 정보는 중요하여 이들 관계에 비교는 유사도 측정 결과에 상당한 영향을 주기 때문이다. 이러한 유사도 측정 공식은 다음과 같다 :

$$S^{CCG} = \sum_{m=1}^M \sum_{n=1}^N (S_{C_{i,m,e}}^{CCG} + S_{I_{i,m,e}}^{CCG})$$

이 곳에서 $S_{C_{i,m,e}}^{CCG}$ 은 클래스와 영역 모델 클래스의 유사도, $S_{I_{i,m,e}}^{CCG}$ 은 클래스와 영역 모델 클래스의 관계를 고려한 유사도, k 는 CCG의 클래스 수, l 은 영역 모델의 클래스 수를 말한다.

S_C^{CCG} 은 객체 추출 단계의 결정단계에서 사용했던 DSS공식 [4]을 적용하였다.

S_I^{CCG} 은 상속성이 존재하는 클래스들의 관계를 고려한 유사도이다. 관계를 고려한 유사도의 산출을 위해 클래스와 영역모델의 상속관계가 있는 클래스들 간의 공통된 부분을 수치화하여 이용한다. 관계를 고려한 클래스의 유사도 공식은 다음과 같다 :

$$S_I^{CCG} = S_C^{CCG} \times \left(\frac{\text{Min}(S_{CI}^{CCG}, S_{CI}^{HM})}{\text{Max}(S_{CI}^{CCG}, S_{CI}^{HM})} \right)$$

이 곳에서 S_C^{CCG} 은 클래스와 영역 모델 클래스의 유사도, S_{CI}^{CCG} 은 클래스의 상속 비율, S_{CI}^{HM} 은 영역 모델 클래스의 상속 비율을 말한다.

S_{CI}^{CCG} 와 S_{CI}^{HM} 의 상속비율이란 클래스들의 공통된 부분이 어느 정도의 비율을 차지하고 있는지를 측정하는 것이다. 이러한 클래스들간의 상속비율을 구하는 공식은 다음과 같다 :

$$S_{CI}^{CCG} = \frac{CCG_{CI}}{CCG_C + CCG_{CI}} \quad S_{CI}^{HM} = \frac{DM_{CI}}{DM_C + DM_{CI}}$$

이 곳에서 CCG_{CI} 은 상속하는 상위 클래스 후보 상속 비율, DM_{CI} 은 상속하는 상위 영역 모델 클래스 상속 비율, CCG_C 은 상속받는 하위 클래스 후보 상속 비율, DM_C 은 상속받는 하위 영역 모델 클래스 상속 비율을 말한다.

3.4 결정

결정 단계의 목적은 상속성 추출 단계에서 추출한 상속성을 가진 클래스와 SW의 요구사항으로부터 전문가에 의하여 구축된 상속성을 가진 영역모델과 유사도를 비교하여 적절한 상속 구조를 가진 클래스를 선택하도록 하는 것이다.

영역 전문가에게 제시된 구조가 유사도가 높다고 항상 좋은 객체지향 구조일 수는 없다. 이것은 영역 모델링을 어느 정도로 상세히 하는가에 따라 달라 질 수 있다. 따라서 영역 전문가에게

다양한 선택기회와 이에 대한 유사도를 제시하여 영역 전문가가 적절한 클래스를 선택하도록 해야한다.

상속성을 가진 클래스를 선택할 수 있는 방법은 두 가지가 있다. 첫 번째 방법은 위의 3.3의 영역 모델과의 비교에서 사용된 바 있는 G^{RA} 에 존재하는 1개의 CCG(계층)를 선택하는 것이다. 이러한 방법은 간단히 프로그램 전체의 뼈대 코드가 나온다는 장점이 있다. 그러나 부분적으로 클래스의 형태와 상속성에 문제가 발생할 수 있다.

두 번째 방법은 각각의 개별적인 CCG에서 부분적인 클래스를 선택하는 것이다. 첫 번째 방법이 1차원적인 결정방법이라고 한다면 두 번째 방법은 2차원적인 결정방법이라고 할 수 있다. 이러한 개별적인 CCG에서 부분적 클래스를 선택하는 것은 적절한 클래스의 형태와 상속성을 구할 수 있다는 장점이 있다. 그러나 부분적으로 선택된 클래스들이 원시 프로그램 전체를 나타내지 못할 수 있다는 단점이 있다. 즉, 원시 프로그램에서는 부분적으로 중복된 부분이 존재, 부분적으로 빠진 코드가 존재할 수도 있다. 이러한 중복문제는 상위 클래스 1개가 여러 하위 클래스를 가지고 있고 하위 여러 클래스와의 관계가 있는 클래스와 중복이 된다면 낮은 계층에 있는 정제된 클래스를 선택하고, 하위 클래스가 존재하지 않는다면 관계성이 높은 클래스를 선택하는 방법을 두어 해결할 수 있다. 또한 부분적인 클래스의 제외 문제는 개별적으로 클래스가 존재할 경우는 해당 클래스를 선택하고, 어떠한 클래스와의 관계가 존재한다면 관계성이 존재하는 곳에 해당 클래스를 첨부하는 형태로 이루어지던 된다.

4. 결론 및 향후연구과제

절차지향 프로그램을 객체지향 프로그램으로 재공학하는 세 번째 단계인 클래스 추출 단계에서는 클러스터링된 객체 후보들의 공통적인 특성을 추출하여 클래스를 추출한다. 클래스 추출 단계의 결과는 클래스들간에 대등한 평면 관계를 이루고 있다. 네 번째 단계인 상속관계 추출 단계는 전 단계에서 추출된 평면화된 클래스들의 공통적 특성을 추출하여 part-of 관계나 is-a 관계를 추출하여 계층 구조를 만든다. 클래스와 상속성 추출 단계를 거침으로 뼈대 코드를 형성할 수 있다.

본 논문은 역 비순환 그래프(G^{RA})를 이용하여 클래스와 상속성을 추출한 결과 영역 전문가에게 비록 최적은 아니라도 영역 전문가에게 다양한 선택 기회를 제시한다.

향후 연구로는 상속성 이후의 과정인 지속성 추출, 객체 지향 코드 생성 및 최종적인 검증이다.

[참고문헌]

- [1] Harald C. Gall, Rene R. Klosch and Roland T. Mittermeir, "Architecture Transformation of Legacy System," Technica Report Number CS95-418, Seattle, April, 1995.
- [2] Harald Gall, Rene Klosch and Roland Mittermeir, "Object Oriented Re-Architecting," Proc. European Software Engineering Conference, Sep., 1995.
- [3] Robert S. Arnold, "Software Reengineering," IEEE Computer Society Press, 1994.
- [4] 박성욱, 노경주, 이문근, "최적화 객체 선정을 위한 다중 객체 추출," 한국 정보 과학회 논문집(B), 제26권, 제12호, p 1468-1481, 1999.
- [5] 이문근, 정창신, 정명선, "소프트웨어 아키텍처를 이용한 역학 모델," 한국 정보 과학회 논문집(B), 제25권, 제11호, p 1630-1647, 1998.