

절차지향 소프트웨어로부터 클래스와 상속성 추출

최정란¹ 이철¹ 이연식² 이문근³
전북대학교 컴퓨터학과¹, 군산대학교 컴퓨터학과², 전북대학교 전자정보공학부³

Extraction of Classes and Inheritance from Procedural Software

Jeong-Ran Choi¹ Chol Lee¹ Yun-Sik Lee² Moon-Kun Lee³

Dept. of Computer Science, Chonbuk National University¹

Dept. of Computer Science, Kunsan National University²

Division of Electronics and Information Engineering Chonbuk National University³

요 약

본 논문은 절차지향 소프트웨어로부터 클래스와 상속성을 추출하기 위한 방법론을 제안한다. 본 논문에서 제안한 방법론은 모든 경우의 클래스 후보군과 그들의 상속성을 생성하여 클래스 후보군과 영역 모델 사이의 관계성과 유사 정도를 가지고 최고 또는 최적의 클래스 후보군을 선택하는데 초점을 둔다. 클래스와 상속성 추출 방법론은 다음과 같은 두드러진 특징을 가지고 있다: 정적(속성)과 동적(메소드)인 클러스터링 방법을 사용하고, 클래스 후보군의 경우는 추상화에 초점을 두며, m 개의 클래스 후보와 n 개의 클래스 후보 사이의 상속 관계의 유사도 측정 즉, 2차원적 유사도 측정은 m 개의 클래스 후보와 n 개의 클래스 후보 사이의 전체 그룹에 대한 유사도를 구하는 수평적 측정과 클래스 후보군들에서 상속성을 가진 클래스의 집합과 영역 모델에서 같은 클래스 상속성을 가진 클래스 집합사이의 유사도를 위한 수직적 측정방법이 있다. 이러한 방법론은 최고 또는 최적의 클래스 후보군을 선택하기 위해 재공학 전문가에게 광범위하고 통합적인 환경을 제시하고 있다.

1. 서론

소프트웨어의 생산성 증가, 유지보수 비용의 절감, 새로운 환경의 적응에 대한 유연성 등의 요구사항에 따라 기존 소프트웨어를 재공학하기 위한 필요성이 강하게 대두되고 있다. 최근 많은 연구자들은 절차지향 소프트웨어(Procedural Software, PSW)를 객체지향 소프트웨어(Object-Oriented Software, OOSW)로의 재공학 초점을 맞추고 있다. 그러나, 이러한 연구는 OOSW의 독특한 결정론적 형태를 처리하는 데에는 제약이 따른다. 결론적으로 OOSW의 수많은 다른 패턴을 생성하기 위한 유연성이 존재하지 않는다.

이러한 제약을 극복하기 위해, 본 논문에서는 OOSW의 수많은 클래스들의 경우의 수를 생성하기 위해 새로운 두드러진 방법론을 기술한다. 이러한 방법론은 클래스들간의 추상화와 각 단계별 유사도에 기반을 두고 있다. 첫 번째로, 모든 가능한 경우의 클래스 후보군들이 생성된다. 두 번째로, 그룹내의 클래스들 간의 유사도를 기반으로하여, 클래스 후보군의 계층을 갖는 상속성이 추출된다. 세 번째로, 클래스 후보군의 클래스들의 집합과 영역 모델에서 같은 클래스 계층을 갖는 클래스 집합 사이의 유사도 뿐만 아니라, 클래스 후보군과 영역 모델 사이의 그룹간의 유사도가 측정된다. 마지막으로, 최고 또는 최적 클래스 후보군이 그룹과 계층 간의 유사도 비교에 의해 선택된다. 만약 그러한 그룹이 존재하지 않는다면, 전체적으로 최적의 후보를 가진 완벽한 그룹을 생성하기 위해 각 그룹으로부터 부분적으로 최고 또는 최적의 클래스 후보들이 결합된다. 본 논문에서 제안한 방법론의 가장 중요한 이점중의 하나는 상속성을 가진 최고 또는 최적의 클래스 후보군을 선택하기 위해 재공학 전문가에게 포괄적이고 통합적인 환경을 제공하는 것이다.

본 논문은 기존 연구와 비교하여 다음과 같은 두드러진 특징을 가지고 있다: 1) 클러스터링 순서가 고정된 결정론적 방법을 사용, 2) 추상화와 일반화를 기반으로 1개의 클래스 후보군이 아닌 가능한 경우의 수에 따른 다중 클래스 후보군 생성, 3) 속성과 메소드의 상호 연결도를 기반으로 주관적 결정 규칙을 가진 클러스터링 방법 사용, 4) 다수의 클래스 후보군과 영역 모델의 클래스들 사이에서 그룹과 계층간의 유사도를 측정하기 위한 알고리즘 사용, 5) 최적 또는 최고의 클래스 후보군과 상속성 결정에 대한 다양한 선택 기회를 제공하기 위해 영역 모델의 다수의 클래스와 클래스 후보군에서 클래스 후보들 사이의 유사도에 따른 통계적 자료를 제공한다.

본 연구는 한국과학재단 특정기초연구(1999-2-203-003-3) 지원로 수행되었다.

2. 관련 연구

COREM에서는 PSW를 역공학한 내용과 영역 전문가의 순공학에 의한 영역 모델링의 정보를 이용하여 상위 추상화 수준의 목적 모델을 생성한 후 최종적으로 OOSW로 변환한다[5].

[7]에서는 PSW에서 VTF(Global Variable, Type, Function)들 관계성을 이용하여 객체 추출, 클래스 추출, 클래스간의 관계성을 추출하였다. VTF를 이용하여 관계성 그래프를 만들 수 있으며 이들간의 가중치값은 사용자에게 의하여 입력된 기준에 의한 관계성의 정도에 의하여 결정된다. VTFG(VTF Graph)를 만든 후 서 그래프에 1개의 노드를 추가했을 경우 IC(Internal Connectivity)이 증가하면 서브 그래프에 추가하여 클러스터링 하였다. 클러스터링 결과의 공통된 부분을 추출하여 클래스·관계성을 추출하였다.

위 방법론만 아니라 그 외 다른 방법[2, 3, 4, 8, 10]들은 공통적으로 함수의 파라미터, 전역변수와 자료형을 이용하여 클래스의 속성을 추출하였고, 이들에 대한 메소드는 함수를 분리후 분할하여 배정하였다. 또한 이들 과정은 1개의 객체 후보군에 대하여 반복적으로 수행되며 각각의 단계에서 전문가가 개입을 한다. 이러한 접근 방법의 문제점은 개개의 다른 클러스터링 순서가 비결정적으로 다른 결과를 생성하는 것이다. 게다가 유일한 객체 후보군이 영역 모델과 비교된다.

3. FTV 그래프와 TVCC 그래프

FTV(Function-Type-Variable) 그래프 즉, G^{FTV} 는 노드가 함 자료형, 전역변수이고, 에지가 함수 호출, 자료형 참조, 전역 변수 참조형으로 이루어진 부분적인 G^{VPR} 그래프[9]이다. G^{FTV} 는 PSW로부터 클래스와 상속성을 추출하기 위해 필요한 정보를 가시화하고 분석하기 위한 것이다.

FTV 그래프는 매우 복잡해질 수 있다. 이러한 복잡성을 줄이기 위해, 강하게 결합된 노드들은 하나의 결합된 노드로서 클러스터링 될 수 있다. 첫 번째로, 동일한 자료형의 전역변수들은 1개로 클러스터링 될 수 있다. 이러한 클러스터링 형태는 TVCC(type-based variable connected component)로서 정의되며, G^{FTV} TVCC의 그래프와 노드는 각각 G^{TVCC} 와 N_{TVCC} 라 한다. 두 번째로, 그래프의 동적인 부분, 즉 함수의 경우도 클러스터링 할 수 있다.

4. 클래스 추출

이 절은 클래스 추출에 대하여 기술한다. 클래스 추출은 다음과 같이 4단계로 이루어진다: 결합, 속성 클러스터링, 메소드 클러스터링, 그리고 추상화.

4.1. 결합 단계

이 단계의 목적은 PSW에서 가능한 최대한의 클래스 후보군을 추출하기 위한 것이다. 전 절에서의 G^{TVCC} 가 입력으로 사용되어, N_{TVCC} 의 원소 n 개가 존재할 경우, n 개를 이용하여 만들 수 있는 모든 경우의 수를 생성한다. 클러스터링 된 개개의 그래프는 $G_{ab}(c_1, \dots, c_i)$ 라 한다. 여기에서, a 와 b 는 입력 그래프의 식별자(ID)이고, 각각 그래프에서 클러스터의 수이다. c_1, \dots, c_i 는 클러스터에서 TVCC들을 순서쌍으로 표현한 것이다. c_1, \dots, c_i 의 합이 그래프에서 TVCC의 수이다.

4.2. 속성 클러스터링 단계

속성 클러스터링 단계에서, G^{TVCC} 의 N_{TVCC} 가 개개의 경우의 수에 따라 클러스터링 된다. 이 단계는 함수를 제외한 변수와 자료형만을 가지고 수행되기 때문에 클래스 추출 과정 중 정적인 부분이라 할 수 있다. N_{TVCC} 를 클러스터링 할 때, 사용자 또는 영역 전문가에게 다음과 같은 클러스터링 기준이 요구된다:

$$C^1 = \langle \text{LIMIT}, \text{WET} \rangle$$

where,

$$\text{WET} = \{ \text{WET}_{(N_{TVCC}, N_{TVCC})}, \text{WET}_{(N_{TVCC}, N_{spec})}, \text{WET}_{(N_{spec}, N_{spec})} \}$$

$\text{WET}_{(N_{TVCC}, N_{TVCC})} : e \in (N_{TVCC}, N_{TVCC})$ 의 중요도
 $\text{WET}_{(N_{TVCC}, N_{spec})} : e \in (N_{TVCC}, N_{spec})$ 의 중요도
 $\text{WET}_{(N_{spec}, N_{spec})} : e \in (N_{spec}, N_{spec})$ 의 중요도
 (단, $\text{WET}_{(N_{spec}, N_{spec})} \leq \text{WET}_{(N_{TVCC}, N_{spec})} \leq \text{WET}_{(N_{TVCC}, N_{TVCC})}$)

C^1 의 LIMIT값, 즉 C^1_{LIMIT} 은 두 개의 N_{TVCC} 가 클러스터링되기 위한 최소 값을 의미하고, C^1 의 WET(Weight of Edge Type) 즉 C^1_{WET} 은 노드 사이에 존재하는 에지의 중요도를 의미한다. C^1_{WET} 과 C^1_{LIMIT} 의 값은 0과 1사이의 실수 값을 갖는 것을 원칙으로 한다.

두 개의 N_{TVCC} 를 클러스터링하기 위한 관계값(RV : Relative Value)은 다음과 같다.

$$RV_{(n_i, n_j)} = \frac{\prod_{i=1}^j w(e_{(n_i, n_j)}) \times \text{WET}}{\sum_{i=1}^j (w(e_{(n_i, -)}) \times \text{WET})}$$

단, $\text{path}(n_i, n_j) = \langle n_i, n_{i-1}, n_{i-2}, \dots, n_{j-1}, n_j \rangle$
 $n_i, n_j \in N_{TVCC}$
 $n_{i-1}, n_{i-2}, \dots, n_{j-1} \in N_{TVCC}$

최대값을 갖는 RV를 MRV라 한다.

속성 클러스터링을 위한 알고리즘은 다음과 같다. 알고리즘의 입력은 TVCC 그래프, 결합단계에서 생성된 경우의 수, 그리고 클러스터링 기준값이다.

```

입력 :  $G_i^{TVCC}$ ,  $(a_{y1}^x, a_{y2}^x, \dots, a_{y2}^x)$ ,  $C^1$ 
출력 :  $G_{i,k}^{TVCC}$ 
while (ConditionOfCombNo( $G_i^{TVCC}$ ,  $a_{y1}^x$ )) {
  for each  $N_{TVCC}$  { calculate  $\text{MRV}(N_{TVCC}, N_{TVCC})$  }
  if (max(MRV) >  $C^1_{LIMIT}$ ) {
    merge( $N_{from \text{MAX}(\text{MRV})}$  to  $N_{to \text{MAX}(\text{MRV})}$ );
    restructure  $G_i^{TVCC}$ 
  }
}
    
```

<알고리즘 1> 속성 클러스터링 알고리즘

4.3 메소드 클러스터링 단계

이 단계는 2개의 부분적 단계로 구성된다: 1) 클러스터링 알고리즘으로 함수를 클러스터링 하고, 2) 알고리즘 적용 후 클러스터

링 되지 않은 함수가 존재할 경우 영역 전문가에 의하여 함수의 소속이 결정된다.

메소드 클러스터링 단계에서는 동적인 정보인 함수를 N_{TVCC} 에 클러스터링 된다. N_{TVCC} 와 N_{spec} 가 클러스터링된 것을 TEF(TVC Extended to Function) 노드라 하고, N_{TEF} 가 존재하는 그래프 G^{TEF} 라 한다.

함수 클러스터링을 위해, 다음 클러스터링의 기준은 사용자로부터 입력된다:

$$C^2 = \langle \text{LIMIT}, \text{EWET} \rangle$$

where,

$$\text{EWET} = \{ \text{EWET}_{\langle N_{TEF}, N_{spec}, \text{FORWARD} \rangle}, \text{EWET}_{\langle N_{TEF}, N_{spec}, \text{BACKWARD} \rangle}, \text{EWET}_{\langle N_{spec}, N_{spec}, \text{FORWARD} \rangle}, \text{EWET}_{\langle N_{spec}, N_{spec}, \text{BACKWARD} \rangle} \}$$

C^2 의 LIMIT(C^2_{LIMIT})는 N_{spec} 가 클러스터링되기 위한 최소값을, C^2 의 EWET(Extended WET)는 노드 사이에 존재하는 에지 종류와 방향성의 가중치를 말한다. C^2_{LIMIT} 와 C^2_{EWET} 의 값은 0과 1사이의 실수 값을 갖는 것을 원칙으로 한다

C^2 를 이용하여 N_{TEF} 와 N_{spec} 의 관계값을 구할 수 있다. 이 값은 에지의 방향성이 존재하기 때문에 RV의 확장(ERV : Extended RV)이라 볼 수 있다. 이를 수식으로 표현하면 다음과 같다.

$$ERV_{(n_i, n_j)} = \frac{\prod_{i=1}^{j-1} w(e_{(n_i, n_{i+1})}) \times \text{EWET}}{\sum_{i=1}^{j-1} (w(e_{(n_i, -)}) \times \text{EWET})}$$

단, $\text{path}(n_i, n_j) = \langle n_i, n_{i+1}, n_{i+2}, \dots, n_{j-1}, n_j \rangle$
 $n_i \in N_{spec}$, $n_j \in N_{TEF}$

노드 n_i 로부터 n_j 로의 여러 개의 경로가 존재할 경우 여러 개의 ERV값이 존재하는데, 이 중 최대값(maximum EVR, MEVR)을 취해야 두 노드 사이의 관계값을 대표할 수 있다.

N_{spec} 가 어느 N_{TEF} 에 속할 것인가를 확률적으로 계산한 값을 DR(Degree of Relativity)이라 할 때 이를 공식화하면 다음과 같다

$$DR(n_i, n_x) = \frac{\text{MERV}(n_i, n_x)}{\sum \text{MERV}(-, n_x)}$$

$0 \leq DR(n_i, n_x) \leq 1$,
 $n_x \in N_{spec}$: source node
 $n_i \in N_{TVCC} \cup N_{TEF}$: target node

이 곳에서 $\text{MERV}(-, n_x)$ 은 노드 n_x 와 연관된 N_{TEF} 나 N_{TVCC} 를 의미한다. 메소드 클러스터링 알고리즘은 속성 클러스터링 알고리즘과 유사하다.

본 논문에서 Procedural Remainder는 영역 전문가에 의한 두 지 처리 방법이 있다: 1) 어떤 클래스 후보(즉, N_{TEF})에 클러스터링 될 것인지를 영역 전문가가 결정, 2) 어떠한 클래스 후보에도 소속이 되지 않는 것을 영역 전문가가 선택할 수 있다. 클래스 후보에 Procedure Remainder가 소속이 결정될 경우 일반적인 클래스 후보와 동일한 방법으로 이후의 과정에서 처리된다. 어떠한 클래스 후보에도 소속이 되지 않는 Procedure Remainder은 재처리될 경우 개별적인 함수로 존재, 또는 속성을 가지지 않는 클래스의 메소드가 된다.

4.4 추상화 단계

$G_{ab}(c_1, \dots, c_i)$ 의 클래스 후보 사이에서 추상화 관계를 이용하면 역 비순환 그래프(Reverse Acycle Graph, G^{RA}) 형태가 된다. 그 중 G^{RA} 의 노드들은 일반화/추상화 관계를 가진다. 상위 노드(G_i^{TEF})는 하위 노드($G_{j,m}^{TEF}$)의 일반화 관계이다. 역으로 하위 노드는 상위 노드의 구체화이다. 크기에 따라 메소드는 하위 계층에서 만 보인다. 이러한 메소드는 에지의 방향에 따라 클래스 다른 계

층에서도 포함된다.

5. 클래스 상속성 추출

클래스 상속성 추출 단계의 목적은 4절에서 추출한 클래스 후보들 사이에서 클래스 계층성을 추출하기 위한 것이다. 이 단계는 클래스 후보군 내에 존재하는 클래스들 사이의 유사도를 기반으로 한다. 두 클래스 사이의 유사도가 높다는 것은 이로부터 수퍼 클래스가 생성될 가능성이 높다고 할 수 있다.

4절에서 추출한 클래스 후보군 내의 클래스 후보에서 상속 관계를 찾기 위해, DSS가 사용된다. 결론적으로 클래스 i 와 j 사이의 DSS의 값을 가진 $n \times n$ 행렬 S 가 생성된다. 여기에서 n 은 클래스 후보의 수이고, i 와 j 는 클래스 후보의 id 이다. 이 행렬 S 는 대칭적이고 대각선의 값은 항상 1이다.

속성 추출 단계에서 RAG와 유사하게 상속성을 가진 클래스 후보군의 비역순환 그래프(RAGI)가 생성될 수 있다.

6. 결정 단계

결정 단계의 목적은 영역 전문가가 작성한 영역 모델에서의 클래스와 클래스 후보군 사이의 유사도를 측정하여 최고 또는 최적의 클래스 후보군을 결정하도록 제공화자를 도와주기 위한 것이다.

6.1 2차원 유사도

2차원 유사도는 다음의 수평적 그리고 수직적 유사도로 구성된다.

수평적 유사도는 RAG-CCGI의 i 번째 클래스 후보군과 영역 모델의 클래스를 비교한 것으로 다음과 같이 정의된다:

$$S_i^{CCG} = \sum_{l=1}^m \sum_{k=1}^n (S_{C_{k,l}}^{CCG} + S_{I_{k,l}}^{CCG})$$

여기에서 k 는 클래스 후보군의 수, l 은 영역 모델의 클래스 수를 나타낸다. $S_{C_{k,l}}^{CCG}$ 는 클래스 후보군 내의 m 개의 클래스 후보와 영역 모델의 n 개의 클래스 사이의 속성과 메소드에 대한 클래스 유사도를 나타내는 행렬이다. 이는 m 개의 후보군과 n 개의 클래스 사이의 DSS값을 측정하여 구한다. $S_{I_{k,l}}^{CCG}$ 는 S_i^{CCG} 내의 k 번째 후보군과 l 번째 클래스 사이의 클래스 유사도를 나타낸다. S_i^{CCG} 는 m 개의 클래스 후보군의 클래스와 n 개의 영역 모델의 클래스 사이의 상속성 유사도를 나타낸다.

수직적 유사도는 전단계에서의 i 번째 클래스 후보군의 부분적인 클래스 후보 집합(C_p)과 영역 모델의 같은 클래스 계층의 클래스 집합(C_m)사이의 유사도이다. 상속성을 가진 C_m 과 비교하여 가장 유사한 C_p 가 RAG-CCGI의 그룹에서 최고 또는 최적의 C_p 이 될 것이다. 이러한 유사도는 다음과 같이 정의된다:

$$S_{C_p}^{CCG} = \sum_{k=1}^m \sum_{l=1}^n (S''_{C_{k,l}}^{CCG} + S''_{I_{k,l}}^{CCG})$$

여기에서, k 와 l 은 각각 클래스 후보와 영역 모델의 클래스의 수이다. $S''_{C_{k,l}}^{CCG}$ 와 $S''_{I_{k,l}}^{CCG}$ 는 각각 C_p 와 C_m 간의 클래스 유사도와 상속도를 나타낸다. 이러한 값들은 C_p 와 C_m 에 해당하는 $S_{C_{k,l}}^{CCG}$ 와 $S_{I_{k,l}}^{CCG}$ 로 구한다. $S''_{C_{k,l}}^{CCG}$ 와 $S''_{I_{k,l}}^{CCG}$ 는 각각 S_i^{CCG} 와 S_i^{CCG} 에서 k 번째 클래스 후보와 l 번째 클래스간의 클래스 유사도와 상속성 유사도를 나타낸다.

6.2 결정

최고 또는 최적의 클래스 후보군에 대한 결정은 수평적 수직적 유사도를 비교하여 결정된다. 수평적 유사도는 클래스 후보군과 영역 모델의 클래스 사이의 전체 유사도를 나타낸다. 수직적 유사도는 후보군과 영역 모델간의 개개의 클래스 계층에 대한 계층 유사도를 나타낸다. 영역 모델에서 많은 클래스 계층이 있을 때, 영역 모델에 대해 완벽한 최고 또는 최적의 클래스 후보군을 구축하기 위해 개개의 클래스 후보군으로부터 부분적으로 최고 또는 최적의 클래스 후보의 집합을 선택한 가능성이 있다.

7. 실험 및 분석

본 논문의 접근 방법에 대한 가능성과 유연성을 증명하기 위해 몇 가지 실험이 수행되었다. <표 1>은 무작위로 선택된 입력 PSW를 기술한 것이다. 표에서 자료는 다음의 네가지 결과로 나타난다: 1) PSW의 크기에 따라 전처리와 클래스 추출 단계에서 많은 클러스터링의 변화가 존재하고, 2) 표에서 메소드 클러스터링 부분의 값을 살펴보면, 상대적으로 PSW의 크기는 클래스 후보군의 경우의 수를 결정하기 위한 주요 요인이 아니며, 3) 주요한 결정요인은 TVCC의 클러스터링의 수이며, 4) 전역 자료형의 수와 새로운 수퍼 클래스의 수간에 강한 의존성이 존재한다. 자료는 클러스터링과 클래스의 수가 상대적으로 전체적인 PSW의 노드 수에 비해 작고 다루기 쉽다는 것을 나타내고 있다.

PSW		Dfs.c	PM.c	Chory.c	Os.c
전처리	크기(Line of Code)	195	300	1065	1374
	전체 노드 수	147	254	817	1279
	자료형의 수	2	4	4	1
	전역 변수의 수	5	4	13	17
	함수의 수	9	15	40	14
	클러스터의 수	14	20	51	17
	TVCC 클러스터의 수	5	5	11	3
클래스추출	수퍼클러스터	9	15	40	14
	클러스터의 수	10*14	16*20	41*51	15*17
상속성추출	메소드클러스터	1*5	1*5	1*11	1*3
	새로운 수퍼 클래스의 수	0	1	1	0

<표 1> 실험 자료

8. 결론 및 향후 연구

본 논문은 PSW로부터 클래스와 상속성 추출을 위한 방법론을 제안하였다. 본 논문에서 사용한 방법론은 클래스의 그룹과 계층의 유사도를 가지고 최고 또는 최적의 클래스 후보군을 선택하기 위해 모든 가능한 클래스 후보군을 생성하는데 초점을 두었다. 향후 연구로는 지속성 추출과 PSW와 OOSW간의 동일성을 검증하는 것이다.

[참고 문헌]

- [1] Robert S. Arnold *Software Reengineering*. IEEE Computer Soc Press, 1994.
- [2] G. Canfora, A. Cimitile and M. Munro. An Improved Algorithm for Identifying Object in Code. *Software-Practice and Experie* Vol. 26(1), pp. 25-48, January, 1996.
- [3] Doris L. Carver. Reverse Engineering Procedural Code for Object Recovery. *Proceedings of Conf. of Software Engineering & Kno Engineering*, pp. 442-449. 1996.
- [4] P. Chen. The Entity-Relationship Model : Toward A unified View of Data. *ACM Transactions on Database System*, pp. 9-36. M 1976.
- [5] Harold C. Gall, Rene R. Klosch and Roland T. Mittermier. *Architecture Transformation of Legacy System*. Technical R Number CS95-418, Seattle, April, 1995.
- [6] E. Horowitz. A Expensive view of reusable software. *IEE Transaction on Software Engineering*, Vol. SE-10, No. 5, pp. 477-September, 1984.
- [7] Yunsokk Jin, Pyeong S. Mah and Gysang Shin. Deriving an Object Model from Procedural Programs. *Proceedings of TOOLS 1997*.
- [8] C. L. Ong and W. T. Tsai. Class and Object Extraction from Imperative. *Journal of Object-Oriented Programming*, pp. 5 March-April, 1993.
- [9] 이문근, 정장신, 정명선, 소프트웨어 아키텍처를 이용한 역공학 모델, 한국정보과학회논문집(B), 제25권, 제11호, pp.1630-1647, 1998.11
- [10] Panos E. Livadas and Theodore Johnson, A New Approach to Finding Objects in Programs. *Journal of Software Maintenance Research and Practice*. Vol. 6, pp. 249-260. 1994.
- [11] 박성욱, 노경주, 이문근, 최적합 객체 선정을 위한 다중 객체군 추출, 정보과학회논문집(B), 제26권, 제12호, pp.1468-1481, 1999.12.