김일성종합대학학보 주체109(2020)년
정보과학 제66권 제3호

객체지향쏘프트웨어의 계승계층구조의 재분해방법

김충혁, 신춘옥

경애하는 최고령도자 **김정은**동지께서는 다음과 같이 말씀하시였다.

《첨단돌파전을 힘있게 벌려야 나라의 과학기술전반을 빨리 발전시키고 지식경제의 토대를 구축해나갈수 있습니다.》

객체지향쏘프트웨어에서 하나의 계승계층구조에 너무 많은 기능이 들어있으면 그 계승계층구조를《높은 응집도와 낮은 결합도》의 원리에 따라 개선하여 더 많은 계승계층구조를 창조하여야 한다.

선행연구[1]에서는 계승계층구조에 대하여 비계승계층구조의 재분해방법으로 그 구조에 들어있는 모든 클라스들을 하나의 실체모임에로 통합하고 그것을 재그룹화하였다. 따라서 코드의 동작을 보존하기 매우 어렵다.

선행연구[2－4]에서는 클라스들을 원래클라스들의 부분클라스 또는 상위클라스로 추출하는 방법으로 재분해를 진행하였다. 따라서 계승나무의 평균깊이와 직접적인 부분클라스의 수는 증가하고 수정하기가 더 어렵다.

론문에서는 계승나무를 쪼개는 방법으로 재구축되는 계승나무의 깊이와 부분클라스들의 수를 조절하는 계승계층구조의 재분해방법을 제안하였다.

1. 계승계층구조의 재분해를 위한 전처리

객체지향쏘프트웨어를 클라스준위의 다중의존관계망[1](CMDN: Class－level Multi－relation Directed Network) $G = G_1 \cup G_2$로 모형화한다. 여기서 $G_1 = (V, E_1)$은 정점을 이루는 클라스들사이의 비계승관계들을 서술하는 그라프이고 $G_2 = (V, E_2)$는 클라스들사이의 계승계층구조를 서술하는 그라프이다.

계승계층구조의 재구축을 진행하기 전에 비계승관계그라프의 모든 정점들을 이루는 클라스들과 계승계층구조의 잎정점들에 놓이는 클라스들에 대한 재분해를 선행연구[1]에서와 같이 진행한다.

계승계층구조의 클라스들에 대한 재분해를 진행하기 위한 전처리를 다음과 같이 진행한다.

① 비계승계층구조에 대한 재분해결과에 따라 다중의존관계망 $G' = G'_1 \cup G'_2$를 다시 구축한다.

② 망 G'_2우에서 련결성분

$$Trees = \{tr_1, tr_2, \cdots, tr_{|Trees|}\}$$

를 탐색한다. 이때 탐색된 매 련결성분 $tr_k (1 \le k \le |Trees|)$는 하나의 계승나무를 이룬다.

③ 고립정점에 대응하는 비계승클라스들을 모두 망 G'_2에서 축출하여 계승계층구조들만을 남긴다.

2. 계승계층구조의 재분해알고리듬

계승계층구조의 재분해에서는 계승계층구조를 이루는 매 계승나무 tr_k 들을 뿌리에 놓이는 클라스로부터 아래로 내려가면서 분해한다. 만일 무리짓기분석후 새로운 클라스들을 원래의 상위클라스로부터 추출하면 분해된 상위클라스의 구조에 기초하여 해당한 부분클라스들을 분할하는 방법으로 응집도를 높인다.

재분해알고리듬은 잎정점들이 모두 처리되였을 때 끝난다.

V_{root} 는 계승나무 $tr_k(1 \le k \le |Trees|)$ 들의 뿌리정점들의 모임이고

$$clas_k^{root} \in V_{root}(|V_{root}| = |Trees|)$$

는 계승나무 tr_k 에서 뿌리정점을 표시한다.

계승계층구조에 대한 재분해알고리듬은 다음과 같다.

입력: 계승망 G_2'

출력: 재구축된 계승망 G_2^R

for $k = 1,\ k \le |V_{root}|,\ k++$

단계 1 뿌리정점 $clas_k^{root} \in V_{root}$ 에서 정의된 모든 실체들(속성들과 메쏘드들)과 그것들사이의 관계를 메쏘드준위의 무게붙은 망으로 모형화한데 기초하여 클라스 $clas_k^{root}$ 를 재그룹화하기 위한 선행연구[1]에서 제안한 알고리듬에 따라 무리짓기분석을 진행한다.

걸음 1 만일 $clas_k^{root}$ 가 분해되지 않았다면 정점 $clas_k^{root}$ 와 그것과 련결된 모든 릉들을 망 G_2' 로부터 삭제한다. 다음 클라스 $clas_k^{root}$ 의 직접적인 부분클라스모임 V_k^{sub} 안의 직접적인 부분클라스들을 뿌리정점들로 놓는다.

$V_{root} = V_{root} \setminus clas_k^{root} \cup clas_k^{sub}$ 로 놓고 걸음 1을 반복한다.

걸음 2 클라스 $clas_k^{root}$ 가 무리짓기분석에서 분할되였으면 그것을 클라스추출대상으로 한다. $clas_k^{root}$ 로부터 추출된 새로운 클라스들의 모임을 V_k^{new} 로, 재구축된 뿌리정점을 $clas_k^R$ 로 표기한다. V_k^{new} 의 매 클라스들은 추출되는 계승나무의 뿌리로 된다.

단계 2 클라스 $clas_k^R$ 에서 모임 V_k^{new} 에 속하는 클라스들의 실례변수들을 창조한다. 이때 추출된 계승나무들은 클라스 $clas_k^R$ 에 뿌리를 둔 주계승나무에 의하여 호출될수 있다.

단계 3 뿌리정점 $clas_k^{root}$ 가 클라스모임 V_k^{new} 로 분해되면 새로 추출된 클라스들의 구조에 따라 $clas_k^{root}$ 의 직접적인 부분클라스 $clas_{km}^{sub} \in V_k^{sub}$ 들을 다음과 같이 분해한다.

걸음 1 뿌리정점 $clas_k^{root}$ 로부터 추출된 매개 새 클라스에 대응하는 새로운 부분클라스들의 모임 V_k^{newsub} 를 창조한다. 그러면 $|V_k^{newsub}| = |V_k^{new}|$ 로 된다.

모든 $clas_{ki}^{newsub} \in V_k^{newsub}$, $i \in [1, 2, \cdots, |V_k^{new}|]$, $M_i^{over}, M_i^{inv} \subseteq clas_{km}^{sub}$ 에 대하여 초기에 $clas_{ki}^{newsub} = M_i^{over} \cup M_i^{inv}$ 로 놓는다. 여기서 $clas_{ki}^{newsub}$ 는 클라스 $clas_k^{root}$ 로부터 추출한 새로운 클라스 $clas_{ki}^{new} \in V_k^{new}$ 의 부분클라스를 표시한다. 그리고 M_i^{over} 와 M_i^{inv} 는 각각 클라스

$clas_{km}^{sub}$에 들어있으면서 클라스 $clas_{ki}^{new}$에서 정의된 상위메쏘드들을 재정의하거나 호출요청하는 메쏘드들의 모임을 표시한다. 이것들은 $\forall i, j \in [1, 2, \cdots, |V_k^{new}|], i \neq j$에 대하여 $M_i^{inv} \cap M_j^{inv} = \phi$, $M_i^{over} \cap M_j^{over} = \phi$을 만족시킨다.

걸음 2 $clas_k^{SR'}$를 재구축된 뿌리정점 $clas_k^R$의 부분클라스라고 하자. 그러면

$$clas_k^{SR'} = \left(M_{root}^{inv} \setminus M_{union}^{inv}\right) \cup \left(M_{root}^{over} \setminus M_{union}^{over}\right)$$

이다. 여기서 $M_{union}^{inv} = M_1^{inv} \cup M_2^{inv} \cup \cdots \cup M_{|V_k^{new}|}^{inv}$, $M_{union}^{over} = M_1^{over} \cup M_2^{over} \cup \cdots \cup M_{|V_k^{new}|}^{over}$이며 M_{root}^{inv}와 M_{root}^{over}는 각각 계승되는 메쏘드들을 호출요청하거나 재정의하는 메쏘드들을 모두 포함하는 모임으로서 $M_{root}^{inv}, M_{root}^{over} \subseteq clas_{km}^{sub}.M$을 만족시킨다.

걸음 3 $\overline{clas_{km}^{sub}} = clas_{km}^{sub} \setminus \left(clas_{k1}^{sub'} \cup clas_{k2}^{sub'} \cup \cdots \cup clas_{k|V_k^{new}|}^{sub'}\right)$라고 하자. 무리짓기분석을 시작할 때 $\overline{clas_{km}^{sub}}$와 $clas_{ki}^{sub'}$의 매 원소들과 $clas_k^{SR'}$를 무리로 놓는다. 즉 $\left|\overline{clas_{km}^{sub}}\right| + \left|V_k^{new}\right| + 1$개의 무리가 있다.

무리짓기알고리듬[1]에 따라 망에서 무리쌍들을 통합하기 전에 무게붙은 모듈도증분을 계산하는데 모임 $V_k^{sub'}$에 있는 부분클라스들을 집약시키지 않도록 하기 위하여 임의의 무리 Com_p와 Com_q에 대하여 $clas_{ki}^{sub'} \subseteq Com_p$, $clas_{kj}^{sub'} \subseteq Com_q$이면 모듈도증분 ΔQ_{pq}를 $\Delta Q_{pq} = 0$으로 놓는다. 여기서 $p, q \in [1, 2, \cdots, |C|]$, $i, j \in [1, 2, \cdots, |V_k^{new}|]$, $p \neq q, i \neq j$이다.

걸음 4 $clas_{km}^{sub}$의 실체들을 재그룹화하여 얻어진 무리모임 $C = \{Com_1, Com_2, \cdots, Com_{|C|}\}$에서 매 무리를 $clas_{km}^{sub}$로부터 추출된 새로운 클라스로 본다.

모든 $p, q \in [1, 2, \cdots, |C|]$, $i \in [1, 2, \cdots, |V_k^{new}|]$에 대하여 Com_p와 Com_q, $clas_{ki}^{new}$ 사이의 계승관계를 다음과 같이 결정한다.

· $clas_{ki}^{sub'} \subseteq Com_p$이면 Com_p는 클라스 $clas_{ki}^{new}$를 계승한다.

· $clas_k^{SR'} \subseteq Com_q$이면 Com_q는 클라스 $clas_k^R$를 계승한다.

걸음 5 클라스 $clas_k^{SR'}$에서 부분클라스 $clas_{km}^{sub}$로부터 추출된 새로운 클라스들에 대하여 $|C|-1$개의 실례변수들을 창조한다.

단계 4 만일 $clas_{km}^{sub} \in V_k^{sub}$, $m \in [1, 2, \cdots, |V_k^{sub}|]$가 직접적인 부분클라스들을 가지면 단계 3을 반복한다. 잎정점준위까지 모두 분석하였을 때 재분해를 끝낸다.

endfor

G_2'를 갱신하여 재구조화된 계승계층구조 G_2^R를 얻는다.

재구조화된 계승계층구조 G_2^R는 원천코드의 동작상의미를 보존한다. 계승계층구조에서 하위클라스는 상위클라스의 메쏘드들을 호출하거나 재정의한다. 알고리듬의 단계 3에서 보여준것처럼 재분해과정에 상위클라스의 메쏘드들을 호출하거나 재정의하는 메쏘드들을 포함하는 하위클라스들의 계승관계는 재구축되는 뿌리클라스 $clas_k^R$에 대해서나 클라스추출재분해로 새로 생겨나는 모임 V_k^{new}의 클라스들에 대하여 정확히 유지되므로 프로그람의 동작에 영향을 주지 않는다. 또한 계승계층구조의 재분해는 선행연구[1]에서 제

안한 메쏘드준위의 다중의존관계망우에서의 무리짓기분석에 기초하므로 계승계층구조의 모듈도를 증가시킨다.

맺 는 말

객체지향쏘프트웨어의 계승계층구조의 클라스추출재분해를 위하여 무리짓기알고리듬을 리용하는 한가지 방법을 제안하였다. 이 방법은 프로그람의 의미를 변경시킴이 없이 모듈도를 증가시킨다.

참 고 문 헌

[1] **김일성**종합대학학보 정보과학, 66, 1, 41, 주체109(2020).

[2] I. Moore; Proc. 11th Annu. ACM Sigplan Conf. Object-Oriented Program. Syst. Languages Appl., 31, 10, 235, 1996.

[3] M. Strekenbach; Proc. 18th Annu. ACM Sigplan Conf. Object-Oriented Program. Syst. Languages Appl., 39, 10, 315, 2004.

[4] Hamid Masoud; The Journal of Systems and Software, 93, 110, 2014.

주체109(2020)년 5월 5일 원고접수

A Method on Refactoring of Inheritance Hierarchies in Object-Oriented Software

Kim Chung Hyok, *Sin Chun Ok*

In this paper, we proposed a method on refactoring of inheritance hierachies using hierachical clustering in order to improve maintenance of object-oriented software.

Keywords: refactoring, cohesion, coupling, clustering