

클래스 재사용을 위한 검색 모델 연구

허종오*, 박만곤

부경대학교 산업대학원 전산정보학과

A Study on the Searching Model for Class Reuse

Jong-Oh Hur and Man-Gon Park

Dept. of Computer and Information, Pukyong National Univ.

요 약

최근의 소프트웨어 개발은 객체지향 프로그램 개발 기법을 사용하여 소프트웨어를 개발하고 있다. 초보 개발자에게는 적절한 객체를 식별하고 정확한 속성과 행위를 부여하는 객체 지향 기법은 상당히 힘든 문제이다. 객체의 집합, 즉 클래스 모델링 기법에도 재사용 기법을 사용하면 이미 작성된 모델을 참조하여 객체 지향 개발 기법을 처음 접하는 초보 개발자에게는 학습의 기회를 제공하고, 숙련된 개발자에게는 검증된 모델을 통해 모델링 실패를 방지할 수 있는 안전성을 제공할 수 있다. 본 논문에서는 유사성(Similarity) 측정 기법을 적용하여 클래스간의 유사도를 판단하고, 관계 일치여부를 분석하여, 재사용 가능한 클래스를 검색하는 모델을 제안한다.

1. 서론

소프트웨어 재사용은 기존에 개발된 경험과 모델 등을 새로운 소프트웨어 개발 시에 재사용함으로써 개발 시간을 단축시키고, 동일한 모델이 다수의 다른 소프트웨어 개발의 생산성 향상과 품질 개선을 보장하고 새로운 소프트웨어 개발 기법을 배우는 초보자에게는 중요한 수단이 된다. [1,2,3,13]

1990년대초 Booch와 Rumbaugh 등의 객체 지향적 소프트웨어 설계방법론 발표 이후에 객체지향 기법을 이용한 소프트웨어 개발이 최근에 주류를 이루고 있다.[4,5,6,9,10,11] 그러나, 객체 모델링 경험이 없는 초보 개발자들에게는 자신에게 주어진 개발 대상에 대해 적절한 객체와 객체의 집합, 즉 클래스를 식별하고 정확한 속성과 행위를 부여하는 것이 쉽지 않다.[4,5,14,15] 유사한 분야에서 이미 작성된 모델을 참조할 수 있다면, 개발자는 개발 대상에 대한 개념을 쉽게 이해할 수 있고, 이를 이용하여 새로운 모델

을 생성할 수 있을 것이다. 즉, 객체 모델링 분야도 재사용의 필요성이 대두되고 있다.

본 논문에서 모델링은 객체들의 집합적인 관계로 이루어져 있으므로, 개개의 객체로 구성된 객체 모델링 보다는 객체의 집합인 클래스를 기준으로 모델링을 구성하여, 질의 클래스 모델과 가장 유사한 클래스 모델을 라이브러리에서 찾아내는 클래스 검색 모델을 제안한다. 이를 위해 클래스와 클래스간 관계에 대한 표현을 정의하고, 유사성(Similarity) 측정과 관계성(Relationship) 분석을 통해 최적의 재사용 모델을 제시하는 검색모델을 구축하며, 예를 들어 적용성을 확인한다.[7,9,13]

2장에서는 클래스와 릴레이션의 표현정의와 유사성 및 릴레이션 테이블 작성 모델을 정의한다. 3장에서는 유사도 테이블과 릴레이션 테이블을 통해 질의 클래스 모델과 가장 유사한 결과 클래스 모델을 찾

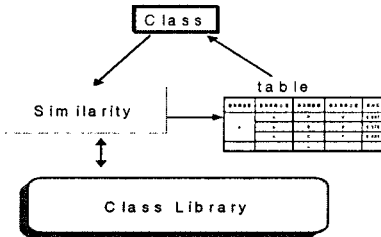
는 클래스 모델링 검색 모델을 제안한다. 4장에서는 클래스 모델링 검색 모델을 구현한다. 5장에서는 결론과 향후 연구 방향에 관하여 기술한다.

2. 테이블 작성 모델 정의

본 장에서는 클래스와 클래스 관계 표현을 정의하고, 유사성(Similarity) 측정을 통한 유사도 테이블 작성과 관계성(Relationship) 분석을 통한 릴레이션 테이블 작성 모델에 대해 정의한다.

2.1 유사도 테이블

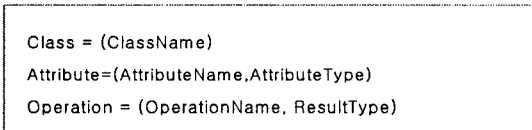
본 절에서는 질의 클래스, 클래스 라이브러리, 유사성(Similarity) 측정 프로세서, 유사도 테이블로 구성된 클래스 유사도 테이블 작성 모델을 (그림 1)과 같이 제안 한다.



(그림 1) 클래스 유사도 테이블 작성 모델

2.1.1 클래스 표현

유사성(Similarity) 측정을 수행하기 위해서는 (그림 2)와 같은 표현으로 클래스를 정의한다



(그림 2) 클래스 표현 정의

2.1.2 유사성 측정법

클래스 라이브러리에서 질의 클래스와 유사한 클래스를 찾는 방법에는 클래스 모델 다이어그램을 이용하여 모델 속성과 관계를 직접 비교해 보는 유추적 기법(Analogy)과 속성에 가중치와 스코어를 부여하여 수치적 데이터 계산으로 유사성(Similarity)을 측정하는 기법이 있다

전자는 자동화 시스템에는 어울리지 않는 수동

적인 방법이므로, 본 논문에서는 후자인 유사성(Similarity) 측정 기법을 적용한다.[8] 유사도 측정은 (정의 1)을 사용한다.

$$S_{ij} = \frac{\sum_{k=1}^n f_k(t_{ik}, t_{jk}) \cdot r_k(t_{ik}, t_{jk})}{\sum_{k=1}^n f_k(t_{ik}, t_{jk}) \cdot X_k}$$

(정의 1) 유사도 측정

여기서 k는 1, 2, ..., n이고, X_k 는 스코어 함수의 값($0 < X_k < 1$), 클래스 속성의 중요성이나 신뢰성을 반영하기 위하여 속성 가중치 f_k 를 사용하였으므로 가중 함수 $f_k(t_{ik}, t_{jk})$ 를 k번째 성분에 대응시키면, 질의 클래스 i와 라이브러리 클래스 j사이의 유사도는 S_{ij} 로 나타낸다. [8,12,13]

2.1.3 유사도 결과 입력

본 절에서는 유사성(similarity) 측정을 통해 질의 클래스와 클래스 라이브러리에 있는 대상 클래스 간의 유사도를 측정하고, 가장 유사도가 높은 결과 클래스를 추출하여, 유사도 테이블에 저장하는 방법을 단계적으로 설명한다.

단계 1. 유사성(Similarity) 측정을 통해 질의 클래스(A)에 대한 라이브러리 클래스(A', B..M)들의 유사도 산출 법을 (정의 2)와 같이 정의한다.

$$S_{AA'} = \frac{\sum_{k=1}^n r_k(t_{Ak}, t_{Ak'})}{\sum_{k=1}^n X_k}, \quad S_{AB} = \frac{\sum_{k=1}^n r_k(t_{Ak}, t_{Bk})}{\sum_{k=1}^n X_k}, \quad \dots, \quad S_{AM} = \frac{\sum_{k=1}^n r_k(t_{Ak}, t_{Mk})}{\sum_{k=1}^n X_k}$$

(정의 2) 유사도 산출 정의

이때, 유사성(Similarity) 측정 시 클래스 속성의 중요성이나 신뢰성을 반영하기 위하여, 속성에 따라 가중치(f_k)를 (표 1)과 같이 부여하였으며, 질의 클래스와 대상 클래스간 속성명의 의미일치 또는 의미유사 여부 따라 스코어(X_k)를 (표 2)와 같이 부여한다.

구분	속성	가중치
Class	Name	0.8
	Type	0.5
Attribute	Name	0.8
	Type	0.5
Operation	Name	0.8
	Result type	0.7

(표 1) 클래스 가중치

구분	기준	스코어
속성 명	의미일치	1
	의미유사	0.5
	의미다름	0

(표 2) 클래스 스코어

단계 2. (정의2)의 유사도 산출 정의를 통해 질의 클래스 모델(A)과 대상 클래스 모델(A')의 유사도를 측정 한 후, 질의 클래스(α)에 대해 가장 유사도가 높은 클래스인 결과 클래스(α')를 (그림 3)과 같이 추출한다.

질의모델명	질의클래스명	결과모델명	결과클래스명	유사도
A	α	A'	α'	0.897
	α	B	β	0.578
	α	C	ϵ	0.685
...

(그림 3) 클래스별 유사도 측정 결과

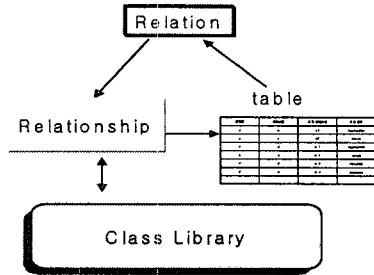
단계 3. 질의 클래스의 모델명(A) 과 클래스명(α), 결과 클래스의 모델명(A')과 클래스명(α')의 유사도(0.897)를 유사도 테이블(그림 4)와 같이 입력한다.

질의모델명	질의클래스명	결과모델명	결과클래스명	유사도
A	α	A'	α	0.897
	β	B	β	0.879
...

(그림 4) 유사도 테이블

2.2 릴레이션 테이블

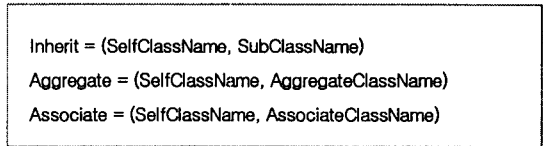
본 절에서는 질의 클래스와 클래스 라이브러리, 관계성(Relationship) 분석 프로세서, 릴레이션 테이블로 구성된 릴레이션 테이블 작성 모델을 (그림 5)와 같이 제안한다.



(그림 5) 릴레이션 테이블 작성 모델

2.2.1 릴레이션 표현

관계성(Relationship) 분석을 위해 (그림 6)과 같은 표현으로 정의한다



(그림 6) 릴레이션 표현 정의

2.2.2 관계성(Relationship) 분석

질의 클래스 모델과 클래스 라이브러리의 클래스간의 관계를 (그림 7)과 같이 분석한다.

모델명	클래스명	관계 클래스명	관계 종류
A	α	$\alpha 1$	Aggregate
A	α	$\alpha 2$	Inherit
A'	α'	$\alpha' 1$	Aggregate
A'	α'	$\alpha' 2$	Inherit
A'	α'	$\alpha' 3$	Associate
A'	α'	$\alpha' 4$	Associate
...

(그림 7) 관계도 분석

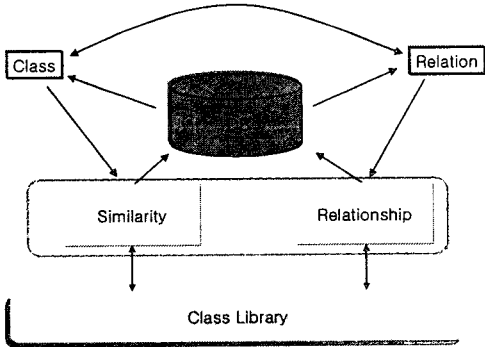
이 때 관계 종류는 inherit, associate, aggregate 중에 하나가 된다

2.2.3 릴레이션 결과 입력

질의 클래스 모델과 대상 클래스간의 관계를 릴레이션 테이블에 저장한다. 저장내용은 모델명, 클래스명, 관계 클래스명, 관계종류로 구성된다.

3. 클래스 모델링 검색 모델

본 장에서는 최적의 재사용 모델을 찾는 클래스 모델링 검색모델을 (그림 8)과 같이 제안하고, 단계순으로 자세히 설명한다.



(그림 8) 클래스 모델링 검색 모델

단계 1. 유사도 테이블(그림 4)에서 유사관계에 있는 질의 클래스와 결과 클래스를 확인한다. 즉, 질의 클래스 α 는 결과 클래스 α' 와 유사도 0.897의 유사관계를 가지고 있다.

단계 2. 유사관계에 있는 클래스 A, A'가 모델내에서 관계하는 클래스들과의 관계종류가 일치하는지를 릴레이션 테이블(그림 7)에서 확인한다.

질의 클래스 모델(A) 내에서 질의 클래스(α)는 $\alpha1$ 이라는 클래스와 aggregate 관계에 있고, $\alpha2$ 라는 클래스와는 inherit 관계에 있다. 결과 클래스 모델(A')의 결과 클래스 (α')는 $\alpha'1$ 과는 aggregate 관계, $\alpha'2$ 와는 inherit 관계, $\alpha'3, \alpha'4$ 와는 associate 관계에 있다. 이때 $\alpha \alpha1 = \alpha' \alpha'1, \alpha \alpha2 = \alpha' \alpha'2$ 의 관계 종류가 일치한다는 것을 알 수 있다.

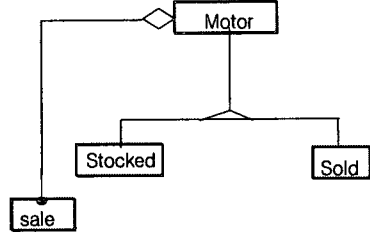
단계 3. 유사도 테이블과 릴레이션 테이블을 통해 클래스간의 유사성(Similarity)과 관계성(Relationship)을 확인할 수 있었으며, 그에 따라 A 클래스와 A' 클래스가 아주 유사한 관계에 있는 것을 확인할 수 있었다. 또한 $\alpha \alpha'3$ 와 $\alpha' \alpha'4$ 는 A 모델에는 없으나, 유사도나 관계측면에서 유사한 관계를 가지는 A'에 있는 관계 이므로, A모델에 적용할 수 있다. 즉, A' 클래스 모델은 A 클래스 모델을 대치 또는 확장 할 수 있는 재사용 모델(A*)이다.

4. 클래스 모델링 검색 모델 구현

본 장에서는 클래스 모델링 검색 모델을 예를 들

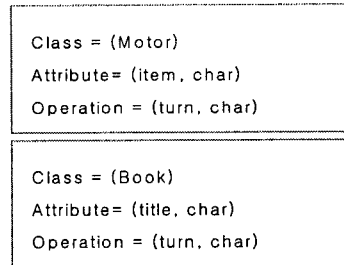
어 단계적으로 구현하였다.

단계 1. 클래스 검색을 위해 모터 판매관리 모델(그림 9)를 질의 클래스 모델로 한다.



(그림 9) 질의 클래스 모델(모터 판매관리)

단계 2. 질의 클래스 모델과 라이브러리에 있는 클래스 모델들을 (그림 10)과 같이 클래스 표현으로 정의한다.



(그림 10) 클래스 표현

단계 3. 클래스 표현을 이용하여 질의 클래스와 대상 클래스의 유사도를 (그림 11)과 같이 측정 한다. 이때 가중치(표 1)와 스코어(표 2)를 사용한다.

$$S_{\text{motor,book}} = \frac{0.8 \cdot 1.0 + 0.8 \cdot 0.5 + 0.5 \cdot 1.0 + 0.8 \cdot 1.0 + 0.7 \cdot 1.0}{0.8 \cdot 1.0 + 0.8 \cdot 1.0 + 0.5 \cdot 1.0 + 0.8 \cdot 1.0 + 0.7 \cdot 1.0} = 0.88$$

$$S_{\text{motor,sale}} = \frac{0.8 \cdot 0.5 + 0.8 \cdot 0.0 + 0.5 \cdot 1.0 + 0.8 \cdot 0.5 + 0.7 \cdot 1.0}{0.8 \cdot 1.0 + 0.8 \cdot 1.0 + 0.5 \cdot 1.0 + 0.8 \cdot 1.0 + 0.7 \cdot 1.0} = 0.55$$

...

(그림 11) 유사도 계산

위의 결과에 의해 질의 클래스(모터 판매관리)와 가장 유사도가 높은 결과 클래스로 Book 클래스가 산출된다.

단계 4. 유사도 테이블에 질의 클래스 모델명, 결과 클래스 모델명, 질의 클래스명, 결과 클래스명과 유사도를 (그림 12)와 같이 입력한다.

질의모델명	질의클래스명	결과모델명	결과클래스명	유사도
Motor sales	Motor	Library	Book	0.88
...

(그림 12) 유사도 테이블

단계 5. 질의 클래스 모델과 클래스 라이브러리에 있는 클래스 모델들의 관계를 (그림 13)과 같이 릴레이션 표현으로 정의한다.

Inherit = (Motor, Stocked, Sold) Aggregate = (Motor, Sale)
Inherit = (Book, Onshelves, Borrowed, Loss, Sales) Aggregate = (Book, Material) Associate = (Book, Journal)

(그림 13) 릴레이션 표현 정의

단계 6. 릴레이션 표현을 이용하여 클래스간의 관계를 분석하여, 모델명, 클래스명, 관계 클래스명, 관계 종류를 (그림 14)와 같이 릴레이션 테이블에 입력한다.

모델명	클래스명	관계 클래스명	관계 종류
Motor Sales	Motor	Sale	Aggregate
		Stocked	Inherit
		Sold	Inherit
Library	Book	Material	Aggregate
		Onshelves	Inherit
		Borrowed	Inherit
		Loss	Inherit
		Sale	Inherit
		Journal	Associate

(그림 14) 릴레이션 테이블

단계 7. 유사도 테이블에 저장된 유사관계가 있는 Motor 클래스와 Book 클래스의 관계 일치성을 아래와 같이 확인한다.

Motor Sale = Book Material : Aggregate

Motor Stocked = Book Onshelves : Inherit

Motor Sold = Book Borrowed : Inherit

Motor 클래스와 Book 클래스는 Motor Sale 관계와 Book Material 관계가 Aggregate 관계로 일치하고, Motor Stocked 관계와 Book Onshelves 관계가 Inherit 관계로 일치하고, Motor Sold 관계와 Book Borrowed 관계가 Inherit 관계로 일치하는 것을 알 수

있다. 그 외에 Book Loss, Book Sale 관계는 Inherit 관계, Book Journal 관계는 Associate관계인 것을 알 수 있다.

단계 1~7을 통해 아래와 같은 결과를 얻을 수 있다.

결과1. Motor 클래스와 Book 클래스는 유사도가 0.88이며, Aggregate과 Inherit 관계가 일치한다.

결과2. Book 클래스는 Motor 클래스에는 없는 Book Loss : Inherit 관계, Motor Sale : Inherit 관계, Book Journal : Associate 관계가 존재한다.

결과 1을 통해 Book 클래스가 Motor 클래스가 유사성(Similarity)이 높으며, 결과 2를 통해 Book 클래스는 Motor 클래스보다 확장된 관계를 가진 클래스임을 알 수 있다. 따라서, Library 모델은 Motor Sales 모델을 대치할 수 있는 재사용 모델임을 알 수 있다.

4. 결론

본 논문에서는 유사성(Similarity) 측정을 통해 질의 클래스와 대상클래스 간의 최고 유사도를 테이블에 입력하는 유사도 테이블 작성 모델 정의와 관계성(Relationship) 분석을 통해 질의 클래스 모델과 대상 클래스 모델의 클래스간의 관계를 입력하는 릴레이션 테이블 작성 모델 정의를 통해 클래스 재사용을 위한 클래스 모델링 검색 모델을 제안하였다.

향후 연구방향으로는 클래스 검색 모델을 구현한 자동화 프로그램 개발과 유사도 측정시의 가중치와 스코어의 세부적인 배정기준 정립이 필요하다.

[참고문헌]

- [1] Gendre Gentner, "The mechanisms of analogical learning," In Bruce G. Buchanan and David C. Wilkins, editors, Readings in Knowledge Acquisition and learning, Morgan Kaufmann Publishers, 673 - 694, 1993
- [2] R. Alan Whitehurst, Systemic Software Reuse Through Analogical Reasoning, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1995
- [3] D.Merkl, A.M.Tjoa, and G.Kappel, "Learning the

- Semantic Similarity of Reusable Software Components,” 3rd International Conferrenc on Software Reuse,33 - 41, 1994
- [4] Peter Deutsch, “Design reuse and framework in th Smalltalk-80 system,” In Ted J. Biggerstaff and Alan J. Perlis, Editors, Software Reusability(II), 57 - 72 ACM Press, 1989
- [5] Martin Fowler, Analysis Patterns, Reusable Object Models, Addison-Wesley, 1997
- [6] Erich Gamma, et. Al., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995
- [7] C. McClure, The Three Rs of Software Automation : Reengineering, Repository, Reusability, Prentice-Hall, 1993
- [8] T. Standish, “An Essay on Software Reuse,” IEEE Transactions on Software Engineering 10(5), 494 - 497, September 1984
- [9] E. Ostertag, J. Hendler, R. Prieto-Diaz, and C. Braun, “Computing Similarity in a Reuse Library System: An AI-Based Approach,” ACM Transactions on Software Engineering and Methodology1(3), 205~228, July 1992
- [10] J. Allen, Natural Language Understanding, 2nd Ed., The Benjamin/Cummings Publishing Company, Inc., 1995
- [11]R.D, Banker, R.J. Kauffman, and D. Zweig, “Repository Evaluation of Software Reuse,” IEEE Trans. On Software Engineering 20(80), 163 - 172, July 1994
- [12] H. Mili, F. Mili, A. Mili, “Reusing Software; Issues and Research Directions” IEEE Trans. on Software Engineering 21(6) 528 - 561, June 1995
- [13] 강문설, “소프트웨어 부품의 재사용을 위한 개선된 패킷 분류 방법과 의미 유사도 측정”, 정보처리 논문지, 855 - 865, 1996
- [14] 박만근, “동적제어 정보를 이용한 효율적인 프로그램 슬라이싱 알고리즘”, 한국멀티미디어학회 논문지, 92 - 100, 2002
- [15] 박순형, “동적 시스템 종속 그래프를 사용한 동적 슬라이싱” 한국멀티미디어학회 논문지, 331 - 341,2002