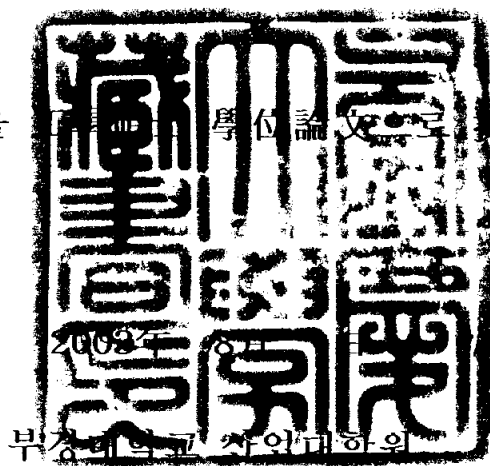


공학석사 학위논문

클래스 재사용에 사용되는 검색도구의 구현

지도교수 박 만 곤

이 論文을 學位論文으로 提出함



전산정보학과

허 중 오

이 논문을 허종오의 공학석사
학위논문으로 인준함

2003년 6월 21일

주 심 공학박사

김 영 봉



위 원 공학박사

여 정 모



위 원 이학박사

박 만 곤



목 차

표목차	ii
그림 목차	iii
Abstract	iv
1. 서론	1
2. 관련 연구	3
2.1 S/W 재사용	3
2.2 재사용 대상	5
2.2 재사용 대상 추출기법	6
3. 클래스 재사용에 사용되는 검색도구	8
3.1 클래스 검색도구	8
3.2 관련 테이블의 설계	10
3.3 클래스 재사용에 사용되는 검색도구 구현	18
4. 검색도구 적용 예	23
5. 결론	30

표 목차

표1. 클래스 검색도구 구성요소 -----	9
표2. 클래스 가중치 -----	13
표3. 클래스 스코어 -----	13
표4. 클래스별 유사도 측정 결과 -----	14
표5. 유사도 테이블 -----	14
표6. 관계성 분석 -----	16
표7. 유관 테이블 -----	17
표8. 검색도구 개발환경 -----	18
표9. 검색도구 세부구성 -----	18
표10. 유사도 테이블 -----	26
표11. 관계성 분석 -----	27
표12. 유관 테이블 -----	28

그림 목차

그림1. 클래스 검색도구 설계도 -----	8
그림2. 유사도 테이블 설계도 -----	10
그림3. 속성 질의 표현 -----	11
그림4. 유관 테이블 설계도 -----	15
그림5. 관계 질의 표현 -----	15
그림6. 속성 질의 입력 -----	19
그림7. 유사도 측정 -----	19
그림8. 관계 질의 입력 -----	20
그림9. 관계성 분석 -----	20
그림10. 결과 클래스 출력 -----	21
그림11. 속성 표현 입력 -----	21
그림12. 관계 표현 입력 -----	22
그림13. 클래스 설계 입력 -----	22
그림14. 질의 클래스(Motor 판매관리) -----	23
그림15. Motor 클래스 속성 질의 표현 -----	24
그림16. 대상 클래스 속성 표현 -----	24
그림17. Motor 클래스 관계 질의 표현 -----	26
그림18. Library 클래스 관계 질의 표현 -----	26
그림19. 결과 클래스 -----	29

The Implementation of A Searching Tool for Class Reuse

Jong - Oh Hur

*Dept. of Computer and Information Graduate School of
Pukyung National University*

Abstract

Recent softwares have been developed by OOP technique. To novice, OOP technique -which identify proper instance and give exact attributes, operations- is difficult. If developers use reusing technique for set of instance, class, they will not only provide novice developers with opportunities of study but also supply experienced developers with safeties which can prevent failure of development through ensured class.

In this paper, I measure similarity score between query class and class in the class library by method of similarity measurement. And I analyse whether they are on agreement through relation analysis.

My final aim is to implement a searching tool which find out reusing class that is close to query class.

1. 서론

소프트웨어 재사용은 기존에 개발된 경험과 기능 등을 새로운 소프트웨어 개발 및 기존 소프트웨어의 유지보수 시에 재사용 함으로써 개발 시간을 단축시키고, 동일 또는 상이한 응용업무의 소프트웨어 개발 시 생산성 향상과 품질 개선을 보장[1,2,3,4]한다.

이러한 소프트웨어 재사용 관련연구에는 CBSD(Componet-Based Software Development)와 소프트웨어 재공학으로 구분되어 활발한 연구가 진행되고 있다. 1990년대초 Booch와 Rambaugh 등의 객체 지향적 소프트웨어 설계방법론 발표 이후에 객체지향 기법론을 이용한 소프트웨어 개발이 최근에 주류[5,6,7,8,9,10]를 이룸에 따라 재사용 대상에는 객체 및 클래스 설계가 대부분이며, 재사용 추출기법에는 소프트웨어 역공학 및 설계패턴 추출 기법이 사용되고 있다. 그러나, 객체 개발 경험이 없는 초보 개발자들에게는 자신에게 주어진 개발 대상에 대해 적절한 객체와 객체의 집합, 즉 클래스를 식별하고 정확한 속성과 행위를 부여[5,6,11,12]하는 것이 쉽지 않다. 유사한 분야에서 이미 작성된 클래스를 참조할 수 있다면, 개발자는 개발 대상에 대한 개념을 쉽게 이해할 수 있고, 이를 이용하여 새로운 클래스를 생성할 수 있을 것이다. 적절한 재사용 클래스를 찾기 위해서는 클래스의 속성과 행동의 유사도와 클래스간 관계의 일치성 등이 종합적으로 고려되어야한다. 그러나, 현재 통용되는 검색방식은 클래스 설계 패턴의 유

사성만을 비교하여 재사용 클래스를 검색하는 경우가 대부분이다. 이러한 방식은 정확한 재사용 클래스 검색에 한계를 가지고 있다. 이러한 한계를 해결하기 위해 본 논문에서는 질의 클래스와 클래스 라이브러리에 있는 대상 클래스의 속성간 유사성[4,8,13]과 관계성을 비교하여 최적의 재사용 클래스를 찾는 검색도구를 구현하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 소프트웨어 재사용의 개요 및 재사용 대상, 추출기법에 대한 관련 연구를 정리한다. 3장은 클래스 재사용에 사용되는 검색도구와 관련테이블을 설계하고 구현한다. 4장에서는 구현된 검색기를 예를 들어 적용한다. 5장에서는 결론 및 향후 연구과제를 제시한다.

2. 관련 연구

소프트웨어 재사용 관련연구는 크게 CBSD와 소프트웨어 재공학 분야로 나뉘어 진다. 소프트웨어 재사용 대상으로는 클래스와 클래스 설계 연구가 있으며, 추출 기법으로는 클래스 설계 패턴과 소프트웨어 역공학으로 구분하여 활발한 연구가 진행되고 있다.

2.1 S/W 재사용

소프트웨어 재사용이란 개발된 객체, 문서화, 시험사례 프로그램 등을 동일한 응용업무, 서로 다른 응용업무, 혹은 서로 다른 회사간에 재사용 하거나 일부 수정 후 다시 사용할 수 있는 개념을 말한다.

소프트웨어 재사용은 여러 요소 중에서 프로그램만 재사용 되는 것이 아니며, 다양한 코드를 라이브러리 내에 구축하고 있다고 해서 재사용도가 높아지는 것은 아니다. 소프트웨어의 개발 정보 전체가 그 대상이 되어야 한다. 즉 개발자가 소프트웨어를 개발하는데 필요할 만한 모든 정보가 재사용 될 수 있다.[14] 이러한 소프트웨어 재사용의 노력의 목표는 효율적으로 기존의 소프트웨어에서 구성요소를 산출하여 활용하는 것으로써 이미 만들어진 컴포넌트를 관련된 분야에 재사용 하는 CBSD와 기존의 시스템을 평가 및 수정하여 재사용 하는 소프트웨어 재공학으로 나뉘어 진다.

(1) CBSD

CBSD는 새로운 시스템을 개발하는 경우 기존의 유사한 시스템에서 개발한 컴포넌트들을 이용하여 어플리케이션을 개발하는 것을 말한다. 이러한 컴포넌트 기반의 소프트웨어를 개발하기 위해서는 프로세스 모델들 가운데 컴포넌트 조합 모델(Component Assembly Model)과 같은 방법을 이용할 수 있다. 이 모델은 기존의 프로세스 모델들 가운데 하나인 나선형 모델과 같이 계획, 위험 분석, 개발 및 보급, 고객 평가의 과정을 거쳐, 새로운 소프트웨어 개발 과정에서 컴포넌트를 기본적으로 재사용 하여 개발[15,16]한다

(2) 소프트웨어 재공학

소프트웨어 재공학은 자동화 도구를 이용하여 기존의 시스템을 평가하거나 수정하는 활동이다. 재공학의 목적은 유지보수성, 생산성, 품질의 향상 등을 포함한다. 일반적으로 재공학은 데이터 이름의 변경이나 정의, 프로세스 로직의 재구성과 같은 형식의 변경을 포함하지만, 기본적으로 기능성은 변하지 않은 상태로 형식이나 구조 변경만을 포함한다. 사용자의 요구가 전문화 되고, 초 스피드화 되는 현실에서는 기본적인 구조에 새로운 기능성을 추가하는 작업이 요구되는데, 이때에는 재설계 과정까지 포함한 재공학이 이루어진다. 즉, 재공학은 소프트웨어 유지보수 과정을 향상시키고 새로운 기술과 도구를 유지보수에 적용함으로써 기존 시스템의 기능을 향상시키는 활동[15,16,17]

을 포함한다.

2.2 재사용 대상

소프트웨어 재사용 대상에는 기존 원시 코드와 핵심 함수 속성 등으로 이루어진 객체의 집합 즉, 클래스와 소프트웨어내 클래스간의 관계로 이루어진 클래스 설계가 있다.

(1) 클래스

객체지향 프로그래밍에서 클래스의 재사용은 새로운 소프트웨어를 개발할 때, 라이브러리에서 관리되고 있는 클래스의 상속 메커니즘을 통해 재사용하는 것을 의미한다. 여러 클래스들에 의해 공유되는 객체를 생성할 때 상위 클래스로부터 메소드와 속성을 상속받아 응용 프로그램에서 필요한 클래스를 만들어 낸다. 여기서 상위 클래스로부터 코드를 상속받는 클래스, 즉 하위 클래스는 상속받는 메소드나 속성을 재정의하고 새로운 메소드나 속성을 추가함으로써 사용자 또는 개발자가 원하는 최종의 클래스를 얻을 수 있다. 이러한 클래스의 상속은 사용자가 새로운 클래스를 정의할 때, 사용자가 원하는 클래스와 가장 유사한 클래스를 찾아 이를 상위 클래스로 선택하고 약간의 다른 점만을 하위 클래스에 추가 또는 재정의 함으로써 사용자가 최종 클래스를 얻을 수 있는 스타일을 제공한다.

(2) 클래스 설계

객체보다 큰 규모의 재사용 단위인 클래스 설계 패턴은 객체들간의 관련성 구조를 포함하고 있다[18]. 시스템을 구성하고 있는 각각의 객체는 “단독으로 고립된 객체는 존재하지 않는다”[19]는 문장에서도 알 수 있듯이 궁극적으로 모든 객체는 결합이라는 방법을 통해 재사용 되는 것이다. 이러한 결합은 객체 하나 하나가 공동의 일을 달성하기 위해 서로 상호작용하는 객체의 연결로 구성된다. 이러한 객체간의 관계는 각 객체 다이어그램으로 표현될 수 있으며 다이어그램을 통해 클래스 설계를 분석할 수 있다.

2.3 재사용 대상 추출기법

소프트웨어 재사용 대상을 추출하기 위한 방법에는 기존 소프트웨어의 산출물로부터 프로그램 구조, 코드 등을 추출하는 역공학 방법과 클래스간의 관계를 추출하는 클래스 단위의 설계 패턴 방법으로 나뉘어진다.

(1) 소프트웨어 역공학

소프트웨어 역공학은 기존 코드나 소프트웨어 생명주기의 마지막 단계에서 얻어지는 산출물로부터 초기단계의 설계 사양이나 요구분석서, 데이터 흐름, 프로그램 구조, 데이터 베이스 설계, 파일구조 등의 정보나 문서를 생성하는 것으로써, 원시코드로부터 재사용 가능한 부

품을 추출하는 연구이다.[20]

(2) 설계 패턴 추출

역공학의 한계를 극복하기 위해 설계문서의 추상화와 특정영역의 일반적인 해결책에 대한 정보표현 및 그 관계를 효과적으로 나타내기 위해서는 패턴형식이 적절하다. 설계 패턴은 객체지향설계의 기본문제 해결을 위해 특정문제 해결에서 한정적일 수밖에 없는 개인적 경험을 추상화를 통해 다른 사람들과 공유하게 하며 개발자들간의 의사 교환 도구로 사용될 공통어휘를 제공한다. 즉, 패턴은 객체지향 패러다임에서 객체의 정형화를 위한 방법으로 적용되고 있으며, 프로젝트를 설계하는데 사용되는 기본적인 청사진을 제공해 준다. 패턴의 근본 개념은 프로그램 설계에 대한 개념과 일치하며 구체적인 사례를 일반적인 형태로 추상화시키는 과정을 부가하는 것이다. 객체지향 소스코드를 기반으로 클래스와 멤버함수, 멤버변수들을 추출한 후에 추출된 객체들간의 추상화된 디자인 패턴(Compositor, Strategy)을 추출하는 알고리즘과 그 패턴 정보들을 저장하기 위한 데이터베이스 스키마를 설계한다.

3. 클래스 재사용에 사용되는 검색도구

본 장에서는 재사용 추출기법을 통해 구축된 클래스 라이브러리로 부터 질의 클래스와 속성의 유사성 및 관계의 일치성이 가장 높은 최적의 재사용 클래스를 검색하는 클래스 검색도구와 검색 결과를 저장하는 관련 테이블을 설계한다.

3.1 클래스 검색도구

클래스 재사용을 위한 도구는 속성 질의와 관계 질의로 구성된 질의 클래스를 입력받아, 클래스 속성의 유사도 측정과 관계의 일치성 분석을 통해 유사도 테이블과 유관 테이블을 작성하여, 테이블 내에서 속성의 유사도와 관계의 일치성이 가장 높은 클래스인 결과 클래스를 출력한다. 클래스 검색도구 설계도는 그림 1과 같다.

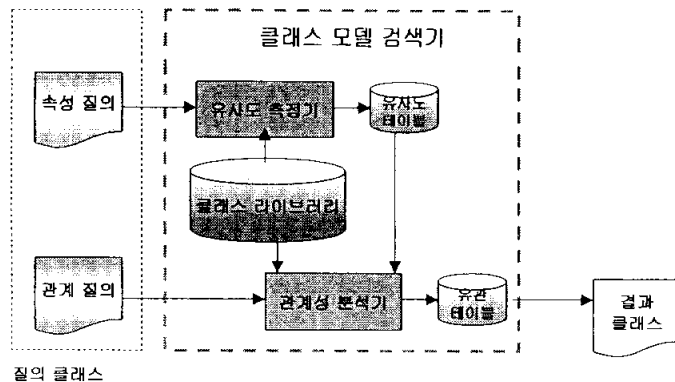


그림 1. 클래스 검색도구 설계도

표 1은 클래스 검색도구의 구성요소별 세부설명이다.

표 1. 클래스 검색도구 구성요소

구성요소	설 명
질의 클래스	클래스 라이브러리에서 최적의 재사용 클래스를 찾기 위해 주어지는 질의 클래스로써, 속성 질의와 관계 질의로 구성
결과 클래스	유사도 측정과 관계성 분석을 통해, 검색되어진 질의 클래스와 가장 유사한 클래스 라이브러리 내의 재사용 가능 클래스
유사도 측정기	질의 클래스와 클래스 라이브러리내의 대상 클래스간의 유사도를 측정하기 위한 프로세서
관계성 분석기	질의 클래스에 대해 유사도가 가장 높은 클래스를 유사도 테이블로부터 가져와 질의 클래스와 대상 클래스의 관계를 비교 분석하여, 일치, 비 일치, 확장여부를 확인하는 프로세서
유사도 테이블	유사도 측정을 통해 나온 결과 값을 저장하는 테이블
유관 테이블	관계성 분석을 통해 나온 결과 값을 저장하는 테이블
클래스 라이브러리	기존 소프트웨어로부터 추출한 클래스와 설계를 일정한 형식으로 표현하여 저장해 놓은 라이브러리

3.2 관련 테이블의 설계

최적의 재사용 클래스 검색을 위해서 질의 클래스와 검색 대상 클래스간 속성의 유사도를 측정하여 그 결과 값을 저장하는 유사도 테이블과 관계의 일치여부를 확인하여 그 결과 값을 저장하는 유관 테이블을 설계한다.

(1) 유사도 테이블

유사도 테이블 작성을 위해서는 속성 질의를 유사도 측정기에 입력하여 클래스 라이브러리에 있는 대상클래스와 속성의 유사도를 비교하여 그 결과를 유사도 테이블에 입력한다. 유사도 테이블 설계도는 그림 2와 같다.

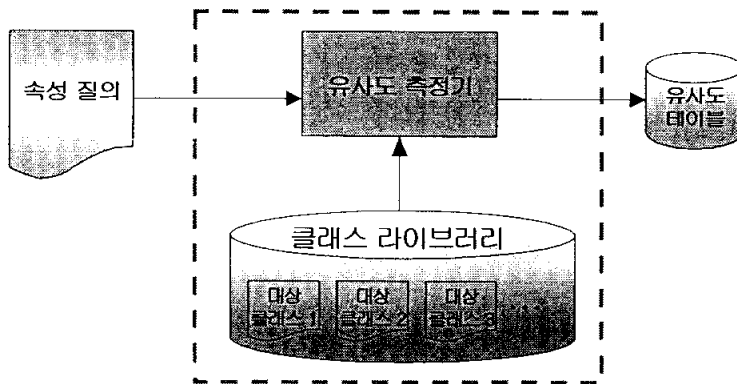


그림 2. 유사도 테이블 설계도

속성 질의를 유사도 측정기에 입력하기 전에 그림 3과 같이 표현한다.

Class = (ClassName)
Attribute = (AttributeName, AttributeType, Identifier)
Operation = (OperationName, ResultType, ArgumentNum, ArgumentType)

그림 3. 속성 질의 표현

유사도 측정은 식 1을 사용한다.

$$S_{ij} = \frac{\sum_{k=1}^n f_k(t_{ik}, t_{jk}) \cdot r_k(t_{ik}, t_{jk})}{\sum_{k=1}^n f_k(t_{ik}, t_{jk}) \cdot x_k} \quad - \text{(식 1)}$$

여기서 k 는 1, 2, ..., n 이고, x_k 는 스코어 함수의 값($0 \leq x_k \leq 1$), 클래스 속성의 중요성이나 신뢰성을 반영하기 위하여 속성 가중치 f_k 사용하였으므로 가중 함수 $f_k(t_{ik}, t_{jk})$ 를 k 번째 성분에 대응시키면, 질의 클래스 i 와 라이브러리 클래스 j 사이의 유사도는 $S_{ij}[4,21]$ 로 나타낸다.

유사도를 측정하여 유사도 테이블에 입력하는 순서를 설명하면 다음과 같다.

1) 유사도 측정기에 질의 클래스 A를 그림 3과 같은 표현으로 입력한다. 이 때 라이브러리에 있는 클래스가 B, C, D 일때, 질의 클래스 A와 대상 클래스 B의 유사도 측정은 식 2를, 질의 클래스 A와 대상 클래스 C의 유사도 측정은 식 3을, 질의 클래스 A와 대상 클래스 D의 유사도 측정은 식 4를 사용한다.

$$S_{AB} = \frac{\sum_{k=1}^n f_k(t_{Ak}, t_{Bk}) \cdot r_k(t_{Ak}, t_{Bk})}{\sum_{k=1}^n f_k(t_{Ak}, t_{Bk}) \cdot x_k} \quad - \text{ (식 2)}$$

$$S_{AC} = \frac{\sum_{k=1}^n f_k(t_{Ak}, t_{Ck}) \cdot r_k(t_{Ak}, t_{Ck})}{\sum_{k=1}^n f_k(t_{Ak}, t_{Ck}) \cdot x_k} \quad - \text{ (식 3)}$$

$$S_{AD} = \frac{\sum_{k=1}^n f_k(t_{Ak}, t_{Dk}) \cdot r_k(t_{Ak}, t_{Dk})}{\sum_{k=1}^n f_k(t_{Ak}, t_{Dk}) \cdot x_k} \quad - \text{ (식 4)}$$

이때, 유사도 측정 시 클래스 속성의 중요성이나 신뢰성을 반영하기 위하여, 속성에 따라 가중치(f_k)를 표 2와 같이 부여하였으며, 질의 클래스와 대상 클래스간 속성명의 의미일치 또는 의미유사 여부에 따라 스코어(x_k)를 표 3과 같이 부여한다.

표 2. 클래스 가중치

구 분	속 성	가 중 치
Class	Name	0.8
Attribute	Name	0.8
	Type	0.5
	Identifier	0.8
Operation	Name	0.8
	Result Type	0.7
	Argument Number	0.5
	Argument Type	0.5

표 3. 클래스 스코어

구 분	속 성	가 중 치
속성명	일 치	1.0
	유 사	0.5
	비일치	0.0

2) 유사도 측정식을 통해 질의 클래스 A와 대상 클래스들 B, C, D의 유사도를 측정한 결과는 표 4와 같다.

표 4. 클래스별 유사도 측정 결과

질의 클래스 명	대상 클래스 명	유사도
A	B	0.897
A	C	0.578
A	D	0.685

3) 질의 클래스 A에 대해 가장 유사도가 높은 클래스인 대상 클래스 B의 질의 클래스 명과 대상클래스 명, 유사도를 표 5와 같이 유사도 테이블에 입력한다.

표 5. 유사도 테이블

질의 클래스 명	대상 클래스 명	유사도
A	B	0.897
E	D	0.879
F	G	0.786

(2) 유관 테이블

유관 테이블 작성을 위해서는 관계 질의를 관계성 분석기에 입력하여 클래스 라이브러리에 있는 대상 클래스와 관계 일치성을 비교하여 그 결과를 유관 테이블에 입력한다. 이때 대상 클래스는 유사성 테이블에서 가장 유사성이 높은 것으로 나타난 클래스를 사용한다. 유관 테이블 설계도는 그림 4와 같다.

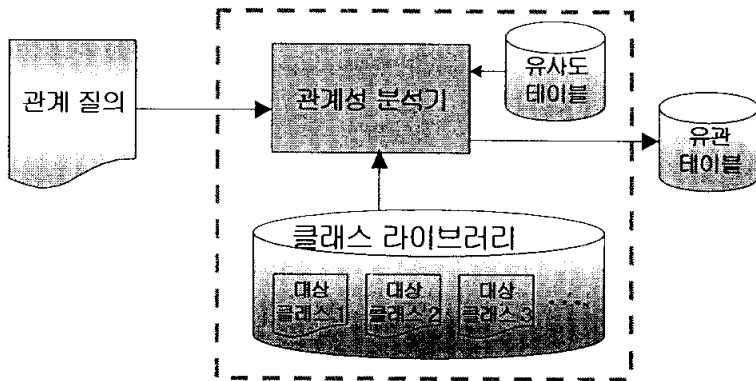


그림 4. 유관 테이블 설계도

유관테이블을 작성하기 위해서는 관계 질의를 관계성 분석기에 입력하기 전에 그림 5와 같이 표현한다.

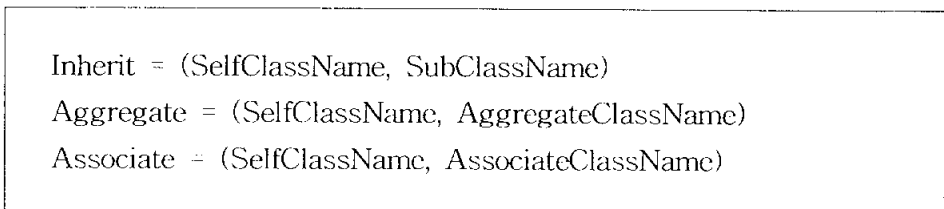


그림 5. 관계 질의 표현

관계 일치성을 유관테이블에 입력하는 순서를 설명하면 다음과 같다.

1) 관계 질의가 관계성 분석기에 입력되면 질의 클래스와 대상 클래스간의 관계를 표 6과 같이 분석한다. 이 때 관계 종류는 inherit, associate, aggregate 중에 하나가 된다

표 6. 관계성 분석

클래스 종류	클래스 명	관계 클래스 명	관계 종류
질의 클래스	A	A	Inherit
	A	A	Aggregate
대상 클래스	B	B	Inherit
	B	B	Aggregate
	B	B	Associate
	B	B	Assoociate
	...	...	...

일치
 일치
 확장

2) 질의 클래스와 대상 클래스간의 관계 분석결과인 클래스 종류, 클래스 명, 관계 클래스 명, 관계종류, 결과를 표 7과 같이 유관 테이블에 저장한다.

표 7. 유관 테이블

클래스 종류	클래스 명	관계 클래스 명	관계 종류	결과
질의 클래스	A	A	Inherit	일치
대상 클래스	A	A	Aggregate	일치
질의 클래스	B	B	Inherit	일치
대상 클래스	B	B	Aggregate	일치
대상 클래스	B	B	Associate	확장
	B	B	Assoociate	확장
	...	...	...	...

3.3 클래스 재사용에 사용되는 검색도구 구현

클래스 재사용 검색도구의 구성은 유사성 측정, 관계성 분석, 라이브러리로 구성되어 있으며, 개발 환경은 표 8과 같다.

표 8. 검색도구 개발환경

개발 환경
○ Processor : Intel P-IV 2GHz ○ 메모리 : 512MB ○ HDD : 20GB ○ O/S : MS Windows 98Se ○ 개발언어 : Visual Basic 6.0 ○ DBMS : MS Access 2000

검색도구 세부구성은 표 9와 같다.

표 9. 검색도구 세부구성

유사도 측정	관계성 분석	라이브러리
1. 속성 질의 입력 2. 유사도 측정 3. 유사도 테이블 입력	1. 관계 질의 입력 2. 관계성 분석 3. 유관 테이블 입력 4. 결과 클래스 출력	1. 속성 표현 입력 2. 관계 표현 입력 3. 클래스 설계 입력

(1) 유사도 측정

그림 6은 속성 질의 입력을 나타내는 화면이다. 그림 7은 유사도 측정 결과를 나타내는 화면이다. 테이블 입력 버튼을 클릭하면 유사도 측정 결과를 유사도 테이블에 입력한다.

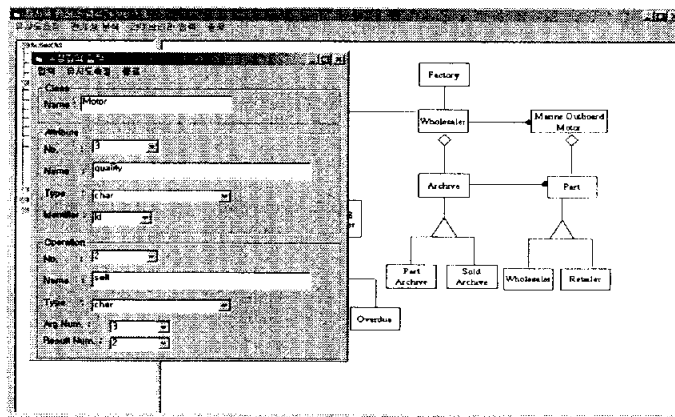


그림 6. 속성 질의 입력

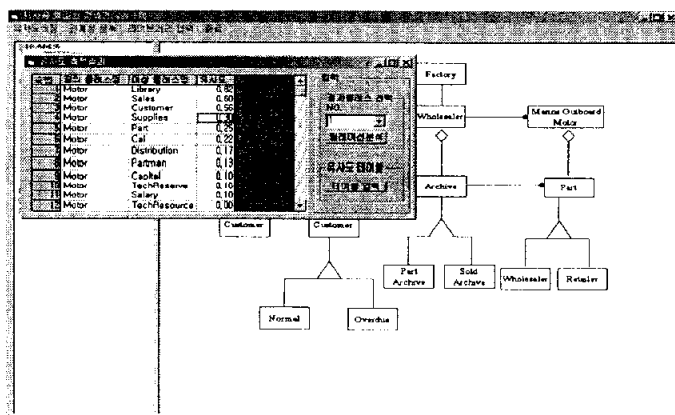


그림 7. 유사도 측정

(2) 관계성 분석

그림 8은 관계 질의 입력을 나타내는 화면이다. 그림 9는 관계성 분석 결과를 나타내는 화면이다. 테이블 입력 버튼을 클릭하면 관계성 분석 결과를 유관 테이블에 입력한다. 그림 10은 최적의 재사용 가능 클래스인 결과 클래스를 출력하는 화면이다.

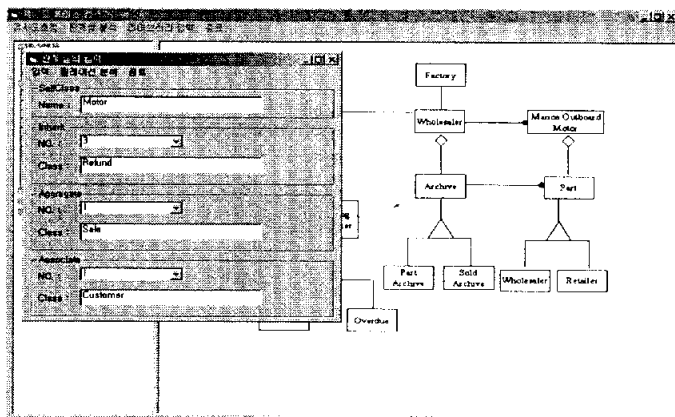


그림 8. 관계 질의 입력

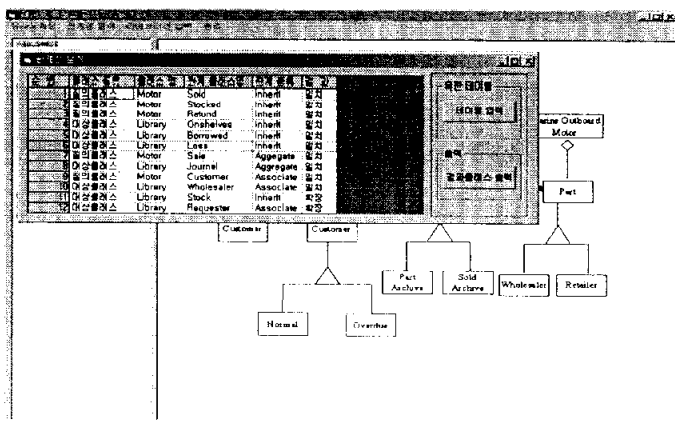


그림 9. 관계성 분석

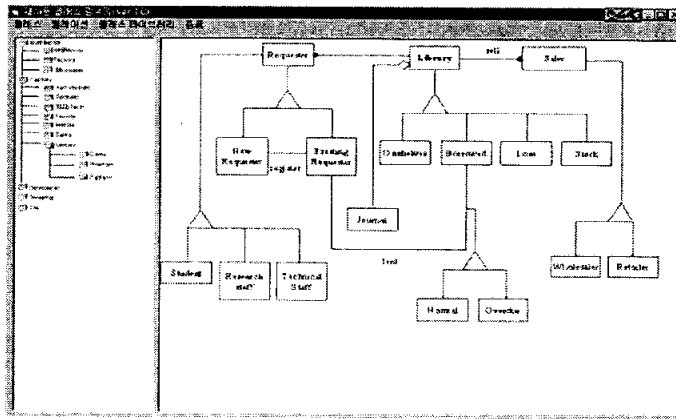


그림 10. 결과 클래스 출력

(3) 라이브러리 입력

그림 11은 속성 표현 입력을 나타내는 화면이다. 그림 12는 관계 표현 입력을 나타내는 화면이다. 그림 13은 클래스 설계 입력을 나타내는 화면이다.

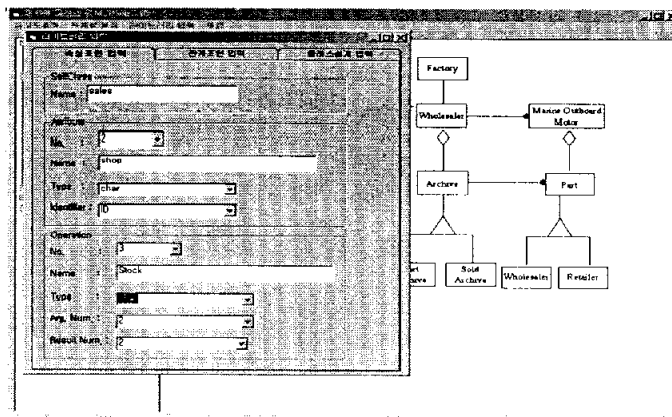


그림 11. 속성 표현 입력

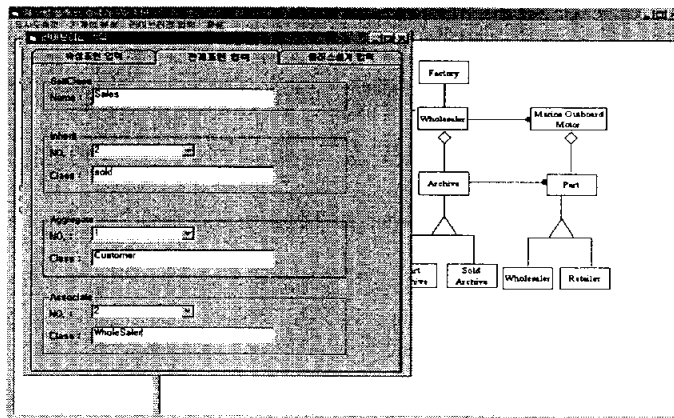


그림 12. 관계 표현 입력

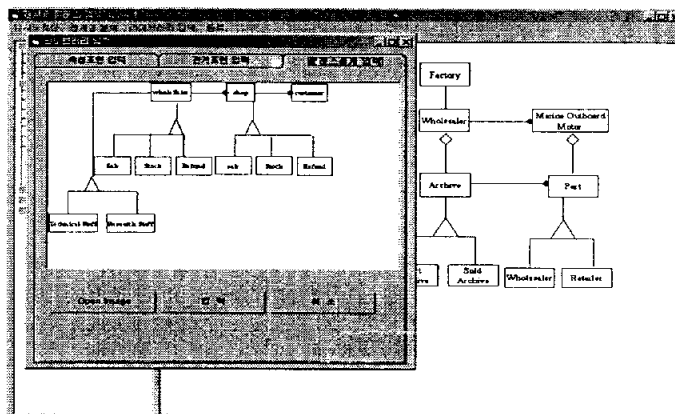


그림 13. 클래스 설계 입력

4. 검색도구 적용 예

본 장에서는 클래스 재사용에 사용되는 검색도구를 사례를 들어 적용한다.

(1) 질의 클래스

클래스 모델 검색을 위한 처음단계로 질의 클래스를 개략적인 모양으로 구성한다. 본 논문에서는 모터 판매관리 소프트웨어 개발을 위해 그림 14와 같이 모터의 판매, 재고, 교환 클래스와 연관된 모터 판매관리 클래스를 질의 클래스로 한다.

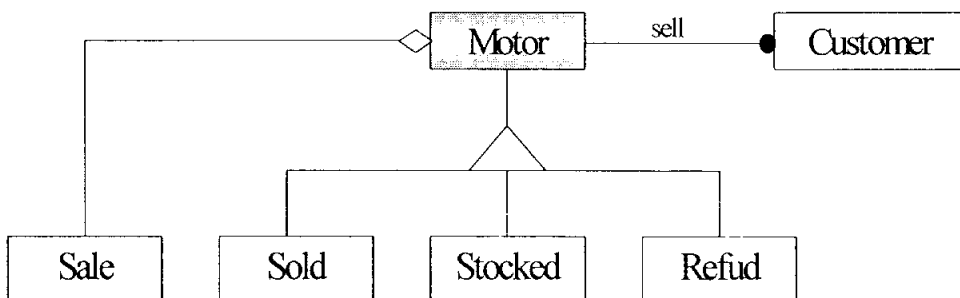


그림 14. 질의 클래스(Motor 판매관리)

질의 클래스의 속성질의를 그림 15와 같이 표현한다.

Class = (Motor)
Attribute 1 = (item, char, Id), Attribute 2 = (price, int, Nid),
Attribute 3 = (quality, char, Nid)
Operation 1 = (Stock, char, 5, 2), Operation 2 = (sell, char, 3, 2)

그림 15. Motor 클래스 속성 질의 표현

클래스 라이브러리 내에 저장된 대상 클래스의 속성 표현은 그림 16과 같다.

Class = (Library)
Attribute 1 = (item, char, Id), Attribute 2 = (price, int, Nid),
Attribute 3 = (quality, char, Nid), Attribute 4 = (volume, char, Nid)
Operation 1 = (onshelves, char, 5, 2), Operation 2 = (sell, char, 4, 2)
Operation 3 = (borrowed, char, 4, 3)

Class = (Sales)
Attribute 1 = (item, char, Id), Attribute 2 = (price, int, Nid),
Attribute 3 = (amount, char, Nid)
Operation 1 = (sale, char, 3, 5), Operation 2 = (refund, char, 3, 1)

Class = (Customer)
Attribute 1 = (name, char, Id), Attribute 2 = (age, int, Nid),
Operation 1 = (manpower, char, 5, 2), Operation 2 = (salary, int, 4, 1)
Operation 3 = (sell, char, 3, 2)

그림 16. 대상 클래스 속성 표현

(2) 유사도 테이블 작성

유사성 측정기는 Motor 질의 클래스와 클래스 라이브러리에 있는 대상클래스의 유사도를 측정한다. 질의 클래스 Motor와 대상 클래스 Library의 유사도는 식 5를, 질의 클래스 Motor와 대상 클래스 Sales의 유사도는 식 6을, 질의 클래스 Motor와 대상 클래스 Customer의 유사도는 식 7을 사용한다. 이 때 표 2의 가중치와 표 3의 스코어를 사용한다. 유사도 결과는 표 10과 같이 유사도 테이블에 입력한다.

$$\begin{aligned} S_{\text{Motor,Library}} = & [(0.8*0.0)+(0.8*1.0+0.5*1.0+0.8*1.0)+(0.8*1.0+0.5*1.0+0.8*1.0)+(0.8*1.0 \\ & +0.5*1.0+0.8*1.0)+(0.8*0.0+0.5*1.0+0.5*1.0+0.5*1.0)+(0.8*1.0+0.5*1.0 \\ & +0.5*0.0+0.5*1.0)] / [(0.8*1.0)+(0.8*1.0+0.5*1.0+0.8*1.0)+(0.8*1.0+0.5*1.0 \\ & +0.8*1.0)+(0.8*1.0+0.5*1.0+0.8*1.0)+(0.8*1.0+0.5*1.0+0.5*1.0+0.5*1.0) \\ & +(0.8*1.0+0.5*1.0+0.5*1.0+0.5*1.0)] = 0.82 \quad \text{-(식 5)} \end{aligned}$$

$$\begin{aligned} S_{\text{Motor,Sales}} = & [(0.8*0.0)+(0.8*1.0+0.5*1.0+0.8*1.0)+(0.8*1.0+0.5*1.0+0.8*1.0)+(0.8*0.0 \\ & +0.5*1.0+0.8*1.0)+(0.8*0.0+0.5*1.0+0.5*0.0+0.5*0.0)+(0.8*0.0+0.5*1.0 \\ & +0.5*0.0+0.5*1.0)] / [(0.8*1.0)+(0.8*1.0+0.5*1.0+0.8*1.0)+(0.8*1.0+0.5*1.0 \\ & +0.8*1.0)+(0.8*1.0+0.5*1.0+0.8*1.0)+(0.8*1.0+0.5*1.0+0.5*1.0+0.5*1.0) \\ & +(0.8*1.0+0.5*1.0+0.5*1.0+0.5*1.0)] = 0.60 \quad \text{-(식 6)} \end{aligned}$$

$$\begin{aligned} S_{\text{Motor,Customer}} = & [(0.8*0.0)+(0.8*0.0+0.5*1.0+0.8*1.0)+(0.8*0.0+0.5*1.0+0.8*1.0)+(0.8*0.0 \\ & +0.5*0.0+0.8*0.0)+(0.8*0.0+0.5*1.0+0.5*1.0+0.5*1.0)+(0.8*1.0+0.5*1.0 \\ & +0.5*1.0+0.5*1.0)] / [(0.8*1.0)+(0.8*1.0+0.5*1.0+0.8*1.0)+(0.8*1.0+0.5*1.0 \\ & +0.8*1.0)+(0.8*1.0+0.5*1.0+0.8*1.0)+(0.8*1.0+0.5*1.0+0.5*1.0+0.5*1.0) \\ & +(0.8*1.0+0.5*1.0+0.5*1.0+0.5*1.0)] = 0.55 \quad \text{-(식 7)} \end{aligned}$$

표 10. 유사도 테이블

질의 클래스 명	대상 클래스 명	유사도
Motor	Library	0.82
	Sales	0.60
	Customer	0.55

유사도 테이블 표 10을 통해 질의 클래스 Motor와 가장 유사도가 높은 대상 클래스는 Library 클래스임을 알 수 있다.

(3) 관계 표현 정의

질의 클래스인 Motor 클래스의 관계는 그림 17과 같이 표현한다. 클래스 라이브러리에 있는 대상 클래스인 Library 클래스의 관계 표현은 그림 18과 같다.

Inherit = (Motor, Sod, Stocked, Refund)
Aggregate = (Motor, Sale)
Associate = (Motor, Customer)

그림 17. Motor 클래스 관계 질의 표현

Inherit = (Library, Onshelves, Borrowed, Loss, Stock)
Aggregate = (Library, Journal)
Associate = (Library, WholeSaler, Requester)

그림 18. Library 클래스 관계 질의 표현

(4) 유관 테이블 작성

관계성 분석기는 Motor 질의 클래스와 클래스 라이브러리에 있는 대상클래스와 관계성을 표 11과 같이 분석한다. 관계 분석결과는 표 12와 같이 유관테이블에 입력한다.

표 11. 관계성 분석

클래스 종류	클래스 명	관계 클래스 명	관계 종류
질의 클래스	Motor	Sold	Inherit
		Stocked	Inherit
		Refund	Inherit
대상 클래스	Library	Onshelves	Inherit
		Borrowed	Inherit
		Loss	Inherit
질의 클래스	Motor	Sale	Aggregate
대상 클래스	Library	Journal	Aggregate
질의 클래스	Motor	Customer	Associate
대상 클래스	Library	WholeSaler	Associate
대상 클래스	Library	Stock	Inherit
		Requester	Associate

- Motor.Sold와 Library.Onshelves 클래스는 Inherit 관계로 일치
- Motor.Stocked와 Library.Borrowed는 Inherit 관계로 일치
- Motor.Refund와 Library.Loss는 Inherit 관계로 일치
- Motor.Sale과 Library.Journal은 Aggregate 관계로 일치
- Motor.Customer와 Library.WholeSaler는 Associate 관계로 일치
- ※ Library에는 Stock(Inherit), Requester(Associate) 확장 관계 존재

표 12. 유관테이블

클래스 종류	클래스 명	관계 클래스 명	관계 종류	결과
질의 클래스	Motor	Sold	Inherit	일치
		Stock	Inherit	일치
		Refund	Inherit	일치
대상 클래스	Library	Onshelves	Inherit	일치
		Borrowed	Inherit	일치
		Loss	Inherit	일치
질의 클래스	Motor	Sale	Aggregate	일치
대상 클래스	Library	Journal	Aggregate	일치
질의 클래스	Motor	Customer	Associate	일치
대상 클래스	Library	WholeSaler	Associate	일치
대상 클래스	Library	Stock	Inherit	확장
		Requester	Associate	확장

(5) 검색 결과

지금까지 질의 클래스와 대상클래스간의 유사성 측정과 관계성분석을 통해 다음과 같은 결과를 얻었다.

1) 결과1. Motor 클래스와 Library 클래스는 유사도가 0.82이며, Inherit, Aggregate, Associate 관계가 일치한다.

2) 결과2. Library 클래스는 Motor 클래스에는 없는 Library. Stock(Inherit), Library.Requester(Associate) 관계가 존재한다.

따라서, Motor 클래스와 Library 클래스간에는 높은 유사성과 관계의 일치성이 존재하는 것을 알 수 있다.

결과 1을 통해 클래스 라이브러리 내에 있는 Library 클래스가 Motor 클래스와 가장 유사성(Similarity)이 높으며, 결과 2를 통해 Library 클래스는 Motor 클래스보다 확장된 관계를 가진 클래스임을 알 수 있다. 따라서, Library는 Motor 클래스를 대체할 수 있는 결과 클래스 즉, 재사용 가능 클래스임을 알 수 있다.

클래스 검색기로 검색된 결과 클래스는 그림 19와 같다.

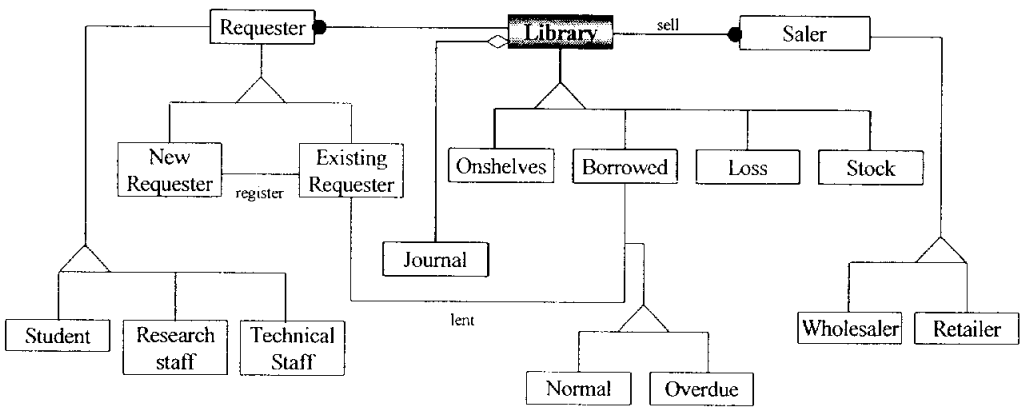


그림 19. 결과 클래스

5. 결론

소프트웨어 재사용은 이전부터 많은 분야에 사용되고 있었으나, 그 효과에 대한 확신이 없어, 많은 각광을 받지 못해왔다. 그러나 객체 지향소프트웨어 기법이 개발의 주류를 이루면서, 별다른 수정 없이 필요한 컴포넌트만을 재사용 할 수 있는 컴포넌트와 사용자가 원하는 기능성을 추가 및 수정하는 소프트웨어 재공학 기법 등으로 발전하게 되었다.

그에 따라 재사용 대상은 함수와 속성을 가진 객체에서 전체적인 소프트웨어 재개발을 위한 클래스 설계까지 재사용의 대상이 확대되었다. 이러한 재사용 대상을 추출하기 위해서 소프트웨어 재공학과 설계패턴 추출기법이 새로운 방식으로 나타났다.

그에 따라 재사용 대상들은 라이브러리로 구성되어 새로운 소프트웨어 개발 시에 자연스럽게 사용되고 있다. 그러나, 소프트웨어 개발자가 찾고자하는 클래스에 가장 적합한 재사용 대상을 찾는 방법은 여전히 효율성 측면에서 문제를 가지고 있다. 객체 지향기법에서 원하는 클래스를 찾는다는 것은 일반 검색과 달라, 객체의 속성과 행동, 객체로 이루어진 클래스의 관계 등이 종합적으로 고려되어야 개발자가 원하는 클래스에 가장 적합한 재사용 클래스를 찾을 수 있다. 그러므로 개발자는 시행착오와 오류 없이 기존에 개발된 우수한 소프트웨어를 재사용 하여 시간과 비용을 절약함으로써 가장 생산성 높은 소프트웨어를 개발할 수 있을 것이다.

본 논문에서는 개발자가 요구하는 최소한의 질의 조건, 즉 질의 클래스를 통해, 클래스 라이브러리로부터 가장 적합한 재사용 클래스를 검색할 수 있도록, 유사도 측정을 통해 질의 클래스와 대상클래스간의 최고 유사도를 테이블에 저장하는 유사도 테이블 설계와 관계성 분석을 통해 질의 클래스와 대상 클래스간의 관계를 저장하는 유관 테이블 설계를 제시하고 클래스 재사용에 사용되는 클래스 검색도구를 예를 들어 적용하였다.

향후 연구과제는 현재의 객체 지향기법에서 재사용하고 있는 다양한 재사용 클래스들의 수집, 분류를 통해 광범위하고 세분화된 클래스 라이브러리 구축과 재사용 클래스 검색도구의 정확한 검색능력 향상을 위해 객체 속성별 유사도에 따른 가중치와 스코어 배점의 세부적인 기준정립이 필요하다.

참고 문헌

- [1] Gendrel 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.Merkel, A.M.Tjoa, and G.Kappel, "Learning the Semantic
Similarity of Reusable Software Components," 3rd
International Conference on Software Reuse, 33 - 41, 1994
- [4] 강문설, "소프트웨어 부품의 재사용을 위한 개선된 패킷 분류
방법과 의미 유사도 측정", 정보처리 논문지, 855 - 865, 1996
- [5] Peter Deutsch, "Design reuse and framework in the
Smalltalk-80 system," In Ted J. Biggerstaff and Alan
J. Perlis, Editors, Software Reusability(II), 57 - 72 ACM
Press, 1989
- [6] Martin Fowler, Analysis Patterns, Reusable Object Models,
Addison-Wesley, 1997
- [7] Erich Gamma, et. Al., Design Patterns: Elements of

- Reusable Object-Oriented Software, Addison-Wesley, 1995
- [8] 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 Methodology 1(3), 205~228, July 1992
- [9] J. Allen, Natural Language Understanding, 2nd Ed., The
Benjamin/Cummings Publishing Company, Inc., 1995
- [10] 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
- [10] 박만곤, "동적 제어 정보를 이용한 효율적인 프로그램 슬라이
싱 알고리즘", 한국멀티미디어학회 논문지, 92 - 100, 2002
- [11] 박순형, "동적 시스템 종속 그래프를 사용한 동적 슬라이싱"
한국멀티미디어학회 논문지, 331 -341, 2002
- [12] C. McClure, The Three Rs of Software Automation :
Reengineering, Repository, Reusability, Prentice-Hall, 1993
- [13] Serge Demeyer, Stephane Ducasse, "Object-Oriented
Reengineering", OOPSLA, 1999.
- [14] Scott Tilley, "A Reverse-Engineering Environment
Framework", Carnegie Mellon Software Engineering
Institute, April 1998
- [15] Nicolas Anquetil and Timothy Lethbridge, "Extraction

- Concepts From File Names; a New File Clustering Criterion", IEEE, 1998
- [16] Ivar Jacobson, "Re-engineering of old systems to an object-oriented architecture", OOPSLA, 1991.
- [17] R. Helm. I. Holland. and D. Gangopadhyay, "Contracts : Contracts : Specifying Behavioural Compositions in Object-Oriented Systems. "Proceeding of OOPSLA/ ECOOP 1990
- [18] K. Beck and W. Cunningham. " A Laboratory for Teaching Object-Oriented Thinking." Proceedings of OOPSLA 1989.
- [19] 김행곤 외 "기존 시스템에서 설계 패턴으로의 재공학 틀에 관한 연구", 한국정보처리학회 논문지 제5권 제9호, pp2334-2343, 1998.
- [20] T. Standish, "An Essay on Software Reuse," IEEE Transactions on Software Engineering 10(5), 494 - 497, September 1984
- [21] H. Mili, F. Mili, A. Mili, "Reusing Software; Issues and Research Directions" IEEE Trans. on Software Engineering 21(6) 528 - 561, June 1995

감사의 글

국토를 방위하고, 국민의 생명과 재산을 보호해야하는 국방의 의무를 짊어진 대한민국 해군으로써의 막중한 책임을 짊어지고, 학업에 대한 욕심으로 시작하였던 지난 2년여간의 대학원 생활도 어느덧 졸업을 앞둔 시간 앞에 와 있습니다. 진해와 부산을 오가며 몸과 마음도 많이 지쳐갔지만, 더 많은 것을 배우고 또 좀더 발전된 모습으로 성장하게 해준 값진 시간이었습니다. 논문을 마치면서 대학원 생활동안 주변에 저를 도와 주시고, 아껴주신 많은 분들께 부족한 지면을 빌어 감사의 말씀을 전합니다.

오늘이 있게끔 부족한 저를 애정과 열정으로 지도해 주시고 때로는 꾸짖음으로 바른 길을 갈 수 있도록 지도해주신 박만곤 지도교수님께 무한한 존경과 감사를 드립니다. 바쁘신 중에도 부족한 저의 논문에 격려와 조언을 아끼지 않으셨던 김영봉 교수님, 여정모 교수님께 머리 숙여 감사 드립니다. 그리고, 멀리 뉴질랜드에 계시지만 저의 논문이 나올 수 있도록 많은 도움을 주신 박순형 선생님께도 깊은 감사를 드립니다.

힘든 시간들을 함께 하며, 어머니 같은 사랑과 도움을 주신 이삼순 선생님, 바쁜 가운데에서도 많은 조언을 아끼지 않으신 최원석 선생님, 자신의 일처럼 도와주신 김만중, 정선정 선생님 늘 좋은 애기와 많은 도움을 주신 최기원, 서갑순, 조철범, 진미향 선생님과 소프트웨어공학 및 멀티미디어 정보처리 연구실 모든 선배님과 후배님들께 감사와 기쁨을 전합니다.

군인 신분으로써, 대학원 공부를 할 수 있도록 처음 길을 인도해 주신 방복동 소령님과 지금은 군인이 아닌 다른 곳에서 국가를 위해 헌신하고 있는 박민우 선배, 늘 함께 해준 후배 김종훈 대위에게 감사의 마음을 전합니다.

마지막으로 저를 낳아주시고, 사람이 살아야 될 정도를 가르쳐 주신 사랑하는 부모님과 큰 누님, 형님, 작은 누님 내외분, 막내누나, 조카들에게 이 논문을 바칩니다.

2003년 7월 허종오