

객체지향 프로토타이핑 지원을 위한 컴퍼넌트 이해 시스템 개발에 관한 연구

김 행 곤[†] · 차 정 은^{††}

요 약

객체지향 소프트웨어의 재사용은 사용자 관점에서 사고하도록 함으로써 기술 분업화를 가능하게 하고, 확장성과 재사용성 측면에서 개발 환경과 사용의 편리성에 대한 요구를 가장 근접하게 해결할 수 있다. 재사용의 현실화는 편리한 재사용 시스템, 특히 사용자 지향적인 검색 시스템의 제공으로 가능하다. 따라서 본 논문에서는 재사용 시스템 사용자들이 편리하고 정확하게 원하는 재사용 컴퍼넌트를 검색하고 수정하며 사용자의 관점에서 새롭게 조립할 수 있는 다중 템플릿 뷰(Multiple-Template Views: MT-Views)를 개발하였다. MT-Views는 혼합형 검색 방법에 따른 유사성 평가를 사용하여 재검색의 정보를 제공함으로써 검색의 효율성을 기하며 미숙한 사용자에 대한 편의와 검색된 부품의 이해를 위한 충분한 정보를 제공한다.

A Study on Development of Component Understanding System for Object-Oriented Prototyping

Haeng-Kon Kim[†] · Jung-Eun cha^{††}

ABSTRACT

Object-oriented software reusability enables to specialize the software techniques by leading the user friendly thought and meets the user's need only concentrating on convenience of development environment. Reuse effect can be maximized through retrieval system. In this paper, we developed the MT-Views(Multiple-Template Views) to realize the reuse effect as integrated reuse system including retrieval, understanding and rebuilding tools based on class library. We used the hybrid retrieval model for easy and efficient retrieval of similar software components. With using these tools, users can retrieve the reusable components and reuse the components for constructing their own applications. Also, we can acquire many informations about retrieved components, so can rebuild components conforming our original intend.

1. 서 론

‘고품질 소프트웨어의 효율적인 생산’이라는 소프트웨어 공학의 궁극적인 목표 달성의 큰 전제 아래

소프트웨어 분야는 끊임없이 제기되는 ‘소프트웨어 위기’와 이를 극복하기 위한 다방면의 방법론과 도구의 제안이 공존하고 있다. 실제, 소프트웨어 기술 발전에도 불구하고 호환성과 상호 운용성에서의 혼란을 비롯하여 지속적인 변경과 기능 추가 요구는 기존 시스템의 유지보수에 인력, 비용, 시간이 묶여 개발 적체가 심화되는 90년대의 또 다른 소프트웨어 위기를 야기

[†] 정 회 원: 대구효성가톨릭대학교 컴퓨터공학과
^{††} 준 회 원: 대구효성가톨릭대학교 대학원 전산통계학과
논문접수: 1997년 1월 15일, 심사완료: 1997년 4월 23일

시켰다. 이에 대한 소프트웨어 공학적 측면의 해결책은 80년대 후반 이후 점차 객체지향 방법론으로 전환되었다. 객체지향 방법론은 재사용이 편리한 소프트웨어 부품화를 이루어 하나의 소프트웨어를 이미 만들어져 있는 표준화되고 신뢰성이 인정되는 소프트웨어 모듈들을 조합함으로써 개발 시간과 비용을 줄일뿐 아니라 유지보수에 능동적으로 대처할 수 있다. 또한, 시스템 구축과 운용에서 사용자 관점으로 사고할 수 있어 확장성과 재사용성 관점에서 개발 환경과 사용의 편리에만 관심을 집중하는 사용자 요구를 가장 근접하여 해결할 수 있다.

재사용 시스템은 재사용 클래스의 저장 및 검색을 위한 도구를 중심으로 사용자들로부터 원하는 소프트웨어 부품을 검색한 후 그 결과를 다른 검색에 필요한 형태로 라이브러리에 새로이 구성하여 다른 응용으로 조립할 수 있는 구축기도 포함한다.

따라서 본 논문에서는 가시적인 효과가 미흡한 소프트웨어 재사용의 실제화를 목적으로 편리하고 정확하게 원하는 재사용 부품을 검색하고 수정하며 사용자 관점에서 새롭게 조립할 수 있는 다중 템플릿뷰(Multiple-Template Views: MT-Views)를 구현하였다. 혼합형 검색 방법에 따른 유사성 평가를 통해 재검색 정보를 제공함으로써 검색의 효율성을 기하며 미숙한 사용자의 편의와 검색된 부품의 이해를 위한 정보를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 재사용의 일반적 개념과 부품 분류와 검색 방법에 대해 설명한다. 그리고 3장에서는 MT-Views를 설계하고 4장에서는 시스템의 구현과 적용 예를 제시한다. 마지막 5장에서 결론 및 향후 연구 계획으로 맺는다.

2. 관련 연구

2.1 재사용에 대한 기본 개념

소프트웨어 재사용은 소프트웨어 생산과 유지보수에 있어서 기존에 개발된 소프트웨어의 표준화된 공통 부분과 잘 알려진 프로세스 사용을 극대화함으로써 품질과 생산성을 향상시키며 유지보수에 능동적으로 대처할 수 있도록 하는 계획되고 체계화된 행위들의 집합이다[1]. 소프트웨어 개발 과정에서 재사용의 체계적인 행위는 두 가지 범주로 나눌 수 있다[3].

재사용을 위한 개발(Develop for reuse)은 고도의 재사용 가능성을 갖는 소프트웨어를 생산하는 것이다. 즉, 내부에 고도의 재사용 잠재성을 내재하고 표현할 수 있는 개발될 소프트웨어의 초기 식별이다. 그리고 재사용에 의한 개발(Develop by reuse)은 기존 소프트웨어의 사용을 극대화하는 것으로 다시 그대로 재사용(Reuse as-is)과 수정(modification)-일반화(generation)-조합(composition) 과정을 통하는 변경을 통한 재사용(Reuse with change)으로 구분된다.

2.2 재사용 기술의 개발 현황

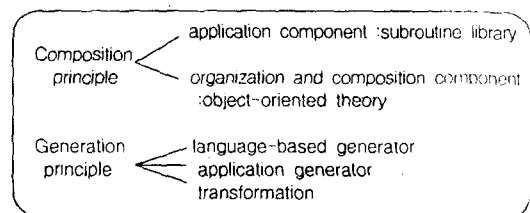
2.2.1 재사용으로의 접근 방법

(1) 재사용되어진 컴퍼넌트에 따른 접근

이 방법에 의해서는 블록 구축(Building Block) 재사용과 패턴(Pattern) 재사용으로 구분되는데 블록 구축은 재사용 가능한 모듈을 라이브러리에 모아두고 필요에 따라 알맞은 모듈을 사용하는 것이다. 그리고 패턴 재사용은 소프트웨어 모듈의 일반적인 형태를 작성한 후 적절한 매개변수를 가하여 알맞은 소프트웨어 모듈을 생성해 나가는 방법이다.

(2) 재사용 원칙에 의한 접근

Biggerstaff[11]는 소프트웨어 개발시 적용되는 재사용의 원칙에 따라 재사용 범주를 (그림 1)과 같이 분류하였다. 조합 원칙(Composition Principle)은 소프트웨어 부품을 블록으로 만든 후 끼워 맞추어 가며 소프트웨어를 완성시키는 방법이다. 그리고 생성 원칙(Generation Principle)은 규칙에 따라 형식 언어 형태의 명세가 자동으로 프로그램을 작성하는 방식으로 추상적 표현이 패턴 매칭을 통해 구체화된다.



(그림 1) Biggerstaff의 재사용 범주
(Fig. 1) Biggerstaff's reuse category

(3) 객체지향 관점에 따른 접근
 객체지향 환경은 재사용 컴퍼넌트의 내용 변경 유무에 따라 (그림 2)와 같이 가지로 구분된다.

화이트박스 재사용	· 상속 구조 및 관련성을 통해 클래스의 구현 사항을 재사용
블랙박스 재사용	· 작은 부품에 대한 동적인 재사용(정보 은닉)

(그림 2) 객체지향 환경에서 재사용 관점
 (Fig. 2) S/W reuse in object-oriented environment

2.2.2 소프트웨어 재사용의 연구 사례

재사용 시스템은 사용자 요구를 받아들여 재사용 시스템에서 추출된 소프트웨어를 제공하는 것이다. 기존 연구들은 강력한 브라우저를 기반으로 클러스터링에 의한 효율적 저장과 탐색 시간 단축에 주력하면서 저장된 소프트웨어 사이에 관련성을 만들고 이를 시각적으로 보여주는 방법을 찾고 있다[12].

본 논문에서 구축한 MT-Views와 유사한 기존 재사용 시스템에 대해 살펴보면 다음과 같다[5, 7, 13]:

① 중앙대의 이 경환 교수님 연구팀에 만들어진 객체지향 재사용 시스템인 CARS(Computer Aided Reuse System)는 객체지향 클래스 부품을 개발, 관리하고 재사용시 필요한 부품을 추출해주는 시스템이다. 라이브러리 관리 시스템과 검색 시스템, 품질 평가, 이해 시스템으로 구성되며 부품 등록시 정해진 인덱스와 가중치를 기준으로 검색한다.

② 전남대의 김 병기 교수님 연구팀의 시스템은 혼합형 분류 방식과 검색을 위해 벡터와 가중치 모델 그리고 브라우징 방법을 사용하였다.

③ 과기원의 권 용래 교수님 연구팀에 의해 제시된 시스템은 열거형과 패킷을 결합한 분류 체계를 바탕으로 확장된 부울리언 질의어와 부품의 계층 구조를 직접 탐색하는 브라우징을 사용하였다.

2.3 소프트웨어 부품의 표현 방법

2.3.1. 부품의 분류

재사용 시스템에서의 부품 분류는 (그림 3)과 같이 요약할 수 있다[5, 7, 9, 10]. 패킷(Facet) 분류 방식은 부품들의 공통된 특성을 수집하여 패킷을 구성한 후 컴퍼넌트에 알맞은 속성을 패킷 항목으로 표현한다.

그리고 열거형(Enumerative) 분류 방식은 넓은 영역의 컴퍼넌트 집합을 점차 좁은 클래스로 분할하면서 계층적 관계를 표현하는 방법으로 클래스들을 미리 계층적으로 정의한 다음 각 부품들을 알맞은 클래스에 할당하는 방법이다. 혼합형 분류 방식은 기본적으로 패킷 분류 방식을 따르면서 검색 시간 단축과 검색 실패라는 문제점을 해결하기 위해 부분적으로 열거형 분류 방식을 혼합한 방식이다.

2.3.2 검색 모델 분석

(1) 검색 시스템

검색 시스템은 사용자로 하여금 미확정된 정보를 통해서도 관심 있는 영역만을 탐색 하도록 하며 정확한 컴퍼넌트가 없더라도 유사한 부품의 제공이 가능해야 한다. 사용자 입장에서 이러한 도구들의 이용은 검색 시스템의 인터페이스를 통해 가능하다[12].

구분	Facet	Enumerative	Hybrid
특성	· 유연성 · 질의 작성 용이	· 계층적 · 구조의 사전 설계	· ENUFACE (ENUMerative+ FACETed)

(그림 3) 소프트웨어 부품의 분류
 (Fig. 3) Component classification

구분	Boolean model	vector model	Hybrid model
특성	· AND,OR,NOT	· Vector 표현 · 항목에 따라 가중치 부여	· Boolean + Vector

(그림 4) 소프트웨어 부품의 검색 모델
 (Fig. 4) Retrieval model of components

(2) 검색 모델

사용자 요구 표현 방식에 따라 부품 검색 방법은 (그림 4)와 같이 분류할 수 있다. 부울리언 모델은 항목들을 공통 특성의 집합으로 표현한다. 즉, 원하는 항목을 복수의 인덱스로 표시하고 그것들간의 관계를 파악하여 질의 항목들을 AND, OR, NOT의 부울리언 연산자로 조합함으로써 질의를 작성한다. 벡터 모델은 여러 항목을 벡터로 표현하여 질의를 작성하는 것으로 질의어 각 항목에 중요도에 따라 일정한 가중치를 부여할 수 있다. 그리고 혼합형 모델은 질의를 만족하는 정도에 따라 검색된 부품들에 순위를

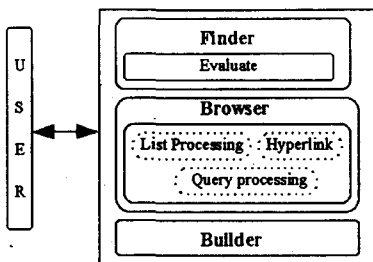
부여함으로 사용자들이 필요 정보 획득에 소모되는 시간을 최소화 한다[5, 6, 7, 10].

3. 프로토타이핑 재사용 환경: MT-Views

3.1 MT-Views의 구성

재사용 시스템은 사용자 측면에서 시스템 자체의 활용성을 강조한다면 사용자가 원하는 부품에 대해 질의를 작성하고 결과를 얻을 수 있는 인터페이스를 갖는 검색 시스템이 중심이 된다.

MT-Views는 클래스 라이브러리를 바탕으로 시스템의 실질적인 인터페이스가 되는 검색 시스템인 동시에 검색된 부품의 이해 시스템으로 재사용 클래스의 검색과 편집, 재구축이 가능한 다중 템플릿(Multiple Template)이다. MT-Views는 부품 검색을 위한 검색기(Finder)와 부품에 대한 이해기(Browser), 새로운 부품 생성을 위한 구축기(Builder)로 구성된다. 검색기는 원하는 부품의 쉬운 검색을 유도하기 위해 질의어 처리와 하이퍼링크 처리를 포함하며 이해기는 유사 부품의 순위 평가를 위한 처리들을 가진다. 구축기는 검색된 정보 중 특정 정보를 사용자 작업 영역으로 옮겨와서 다른 응용에서 재사용할 수 있는 새로운 부품을 작성한다.



(그림 5) MT-Views의 시스템 개략도
(Fig. 5) MT-Views's Architecture

3.2 컴퍼넌트 라이브러리

3.2.1 부품의 분류

기존의 재사용 시스템은 획일적인 패킷 분류나 상속 구조에 기반을 둔 부문적 정보에 의해 분류된 것이므로 재사용 라이브러리의 활용도를 크게 제한하고 있다[8]. 따라서 MT-Views에서는 사용자의 개념을

그대로 시스템 부품 표현 방식으로 매핑시키고 다양한 방법에서의 검색을 위해 열거형과 패킷형을 단계적으로 결합한 혼합형 방식을 사용했다. 그리고 브라우징 방식의 검색을 위해 계층적 정보 구조를 제공할 뿐만 아니라 패킷으로 부품을 나타내어 질의와 부품의 정확한 표현과 유사한 부품의 구분이 가능하다. 또한 비슷한 부품끼리의 브라우징이 가능하도록 유사 속성의 부품을 하나의 클러스터로 모으고 계층적인 클래스로 상세화 하였다. MT-Views는 부품의 패킷을 type, object, setting method, OS 그리고 function으로 분류하였다. 여기서 타입은 객체가 저장된 장소나 실제 작업이 수행되는 장소에 관한 정보의 표현으로 자료 구조나 컴퓨터 시스템의 자원을 의미한다. 또한 객체는 작업이 적용되는 대상이며 함수는 컴퍼넌트가 수행하는 기본 작업이다. setting method는 멀티미디어 시스템 구축을 위한 프로토타입 환경 지원을 목적으로 하는 응용에서의 필수적인 것으로 앞으로 구축될 소프트웨어가 멀티미디어 요소를 포함하고 있을 것이라는 타당성 하에서 재사용 가능성을 위해 설정되었다.

3.2.2 부품의 검색 모델

MT-Views의 검색 모델은 사용자 임의에 의해 질의를 최상으로 작성하고 시스템으로의 적용이 잘 이루어지도록 기존의 검색 모델을 독립적인 성격으로 통합하였다. 또한 인터페이스 수준에서 구현 수준의 상세한 정보까지 얻을 수 있는 사용자 편의성을 제공하므로 사용자는 획득된 컴퍼넌트 정보 수준과 선호 방식에 따라 검색을 다양하게 시도할 수 있다.

3.2.3 검색 알고리즘

정확한 부품의 검색이 이루어지지 않을 경우 가장 유사한 부품을 사용자에게 반환하므로써 재검색을 유도하고 질의를 정제 할 수 있다. 따라서 MT-Views에서는 사용자가 원하는 검색 방법의 선택을 위해 복합된 컴퍼넌트 분류 상에서 다양한 검색의 시도가 가능하도록 융통성 있는 검색 인터페이스를 제공한다. (그림 6)은 MT-Views의 검색 알고리즘이다.

3.2.4 유사성 측정

관련된 부품의 검색을 위한 유사성 측정 방법으로

1단계	복합된 분류 방식 상에서 원하는 검색 방법 결정
2단계	지정 검색을 원하면 제시된 인덱스 선택을 조합하여 라이브러리와 매칭한 결과를 반환하고, 아니면 패시별로 제공되는 기본 항목들을 선택하여 질의 작성
3단계	동의어 처리를 통한 검색방법이 선택되면 동의어 사전과 매핑을 통해 질의를 정제.
4단계	완성된 질의가 라이브러리의 부품 표현과 일치하면 검색한 부품을 출력하고 아니면 단계 5로 진행.
5단계	인덱스 키워드 매칭에서 매핑되어진 인덱스의 값이 임계치 이상인 것만 검색.
6단계	유사도 계산으로 재검색할 후보 컴퍼넌트를 출력

(그림 6) 검색 알고리즘
(Fig. 6) Retrieval algorithm

MT-Views에서는 데이터베이스의 데이터와 매칭된 질의어의 인덱스의 개수 및 검색 처리된 가중치의 합을 계산함으로써 구현하였다.

(그림 7)은 검색을 위해 SA(Structured Abstract) 방식을 이용할 경우의 유사성 측정의 예로 항목에 가중치의 합으로써 유사 부품의 검색이 유도된다. 즉, 검색시 데이터베이스에 저장된 항목과 질의에 사용된 용어가 일치하면 질의 작성시 부가된 가중치를 합한다. 그러나 여기서 검색의 정확성을 보완하기 위해 매칭되지 않은 항목의 가중치가 높은 순서로 (0.5, 0.4, 0.3, 0.2, 0.1)을 빼주어 질의 처리의 값을 계산한다.

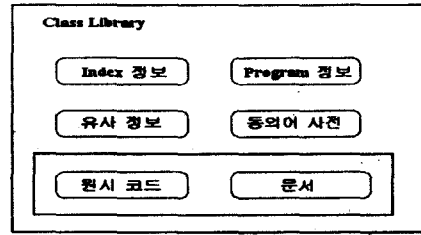
Items (Language, Domain, Function, Object, OS)
Weight (0.2, 0.6, 0.8, 0.5, 0.2)
Threshold 1.4
DB의 내용 (C++, Advesting, add, task, Win95)
Query 내용 (C++, Sales, add, task, Win95)
→ 질의 처리 : 합 1.3
∴ 재검색을 위한 정보 제공

(그림 7) 유사성 측정의 예
(Fig. 7) Example of similarity estimation

3.2.5 라이브러리 구조

MT-Views의 라이브러리 구조는 (그림 8)과 같이 도식화될 수 있으며 라이브러리 각 구성 요소의 설명은 아래와 같다:

- 원시 코드: MT-Views 내에서 실제적인 재사용의



(그림 8) MT-Views의 라이브러리 구조
(Fig. 8) Library structure in MT-Views

대상이 되는 클래스를 나타낸다.

- 문서: 부품 기능성에 대한 정보를 가진 서술적 구조를 의미한다. 문서는 부품의 정리된 메타 정보를 이루고 있으므로 코드로부터 얻을 수 없는 의미적 정보를 획득하기 위해 사용된다.
- Index 정보: 문맥에 대한 인덱스를 구축하기 위한 구조로 문서로부터 추출되어진 인덱스 항목의 집합은 그 문서의 프로파일을 형성한다.
- Program 정보: 재사용 클래스의 원시 코드에 대한 생성 및 관계 정보를 상속이나 연관 모델에 따라 정의한 정보를 가진다.
- 동의어 사전: 사용자 관점 질의와 시스템이 이해하는 질의 사이에서 의미적 차이를 최소화하기 위해 대표 항목과 관련 용어를 제공한다.
- 유사 정보: 검색이 완전히 일치하지 않을 경우 대체 가능한 다른 유사 부품의 제공으로 재검색으로의 자연스러운 유도 및 부품의 관련성 정보를 제시한다.

3.2.6 동의어 사전

검색 시스템을 위한 동의어 사전은 주어진 영역에 관련된 지식을 나타내는 항목들과 그 항목들 간의 개념적 관계를 기술하는 정보 구조로서 동의어 사전의 항목들은 부품을 기술하고 사용자 질의어에 사용될 항목들에 대한 표준 어휘 역할을 한다. 이들 항목들 간의 개념적 관계는 질의어에 의한 검색 과정과 질의어 확장 과정에서 편리하게 이용되어진다.

MT-Views에서의 동의어 사전은 사용자의 요구 질의를 시스템 내의 기존 컴퍼넌트로 매핑시키기 위한 의미적 일치성을 평가하는 기능을 담당한다. 또한 동의어의 추가, 삭제, 찾기 등의 동의어 사전 관리를 재사용자에게 허용함으로써 사용자 자신의 응용 도메인에 적합한 특화된 동의어를 가질 수 있다.

3.3 MT-Views의 설계

3.3.1 시스템 기능

재사용을 실체화하는 통합된 재사용 시스템인 MT-Views에 대한 기능적 요구는 크게 검색(Browsing), 수정(Editing), 합성(Composition)의 세가지로 요약되어진다.

(1) 재사용 부품 및 정보 검색

- 질의 처리와 클래스 네비게이션
- 우선 순위 평가를 통한 유사 부품 검색
- 재사용 정보의 검색
- 동의어 사전 인터페이스 제공 및 처리

(2) 재사용 부품의 수정

- 저장된 클래스 부품의 코드를 수정하여 재등록

(3) 재사용 부품 합성 지원

- 클래스 부품의 특성을 상속받아 서브 클래스 형태의 새로운 C++ 코드 생성
- 클래스 부품을 합성하여 새로운 C++ 코드 생성

MT-Views는 위에서 기술한 요구 조건들이 충분히 수용되도록 설계되고 구현되었다.

첫째, 검색 과정에서 인덱스 항목들 간의 관계를 이용하기 위하여 동의어 사전을 구축했다.

둘째, 인덱스 항목들을 패킷별로 관리하고 항목들 간의 관계를 미리 정의하여 질의어 작성이나 검색 과정에서 이것을 이용할 수 있도록 허용했다.

셋째, 부품을 추상적 기능에 따른 열거형 방식의 검색 및 브라우징 등 다양한 검색 방법을 지원했다.

네째, 검색이 완료된 부품을 자신의 응용을 위해 적당하게 수정하고 또한 다른 응용에서 재사용 가능한 완전한 컴퍼넌트로 활용할 수 있도록 편집기를 생성하고 컴파일을 위한 통합 환경을 제공하였다.

3.3.2 개발 환경

MT-Views 개발을 하드웨어 환경은 PC Pentium을 기본으로 하여 한글 Windows 95를 사용한다. 그리고 소프트웨어 환경으로는 Delphi로 기본적인 인터페이스와 로컬 데이터베이스를 구축하였다.

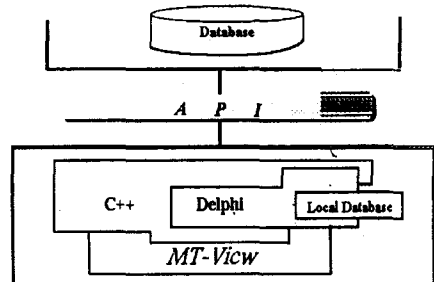
(그림 9)는 MT-Views의 소프트웨어 환경이다.

3.3.3 MT-Views의 서브 시스템

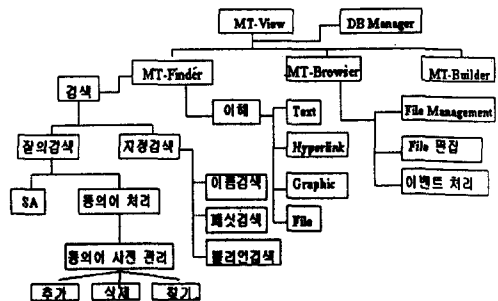
MT-Finder는 컴퍼넌트 검색과 이해를 위한 부분으로 검색은 SA(Structured Abstract) 방식과 동의어 처

리 및 사전 관리를 포함하는 질의 검색과 이름, 패킷, 불리언 방식에 의한 지정 검색으로 구성된다. 그리고 이해 부분은 컴퍼넌트에 대한 충분한 정보를 제공함으로써 원래의 재사용자 의도대로 그리고 구축될 시스템의 기능적 착오가 발생하지 않도록 텍스트, 하이퍼 링크, 그래픽, 일반 파일 형태로 제공된다. 하이퍼 링크 검색은 다른 질의 검색 방식에 의해 반환되어진 유사 부품의 계층적 관련성 상에서 이루어진다. MT-Editor는 검색 완료된 부품의 히스토리와 편집 환경을 제공함으로써 검색과 편집의 연계성을 유지할 수 있다. 그리고 편집이 완료된 부품의 재구축을 위해서 데이터 변환 기법을 이용해 MSVC의 통합 개발 환경으로 데이터를 전달해 컴파일 한다. 이것은 개발 작업의 단순화 뿐 아니라 완벽한 컴파일 기능의 제공으로 라이브러리 유지 및 향후의 버그 가능성을 최소화할 수 있다.

MT-Views 개발을 위한 개략적인 기능적 블록도는 (그림 10)과 같다.



(그림 9) MT-Views의 소프트웨어 환경 (Fig. 9) MT-View's S/W environment



(그림 10) MT-Views의 기능적 블록도 (Fig. 10) MT-View's functional block diagram

3.3.4 시스템 구성

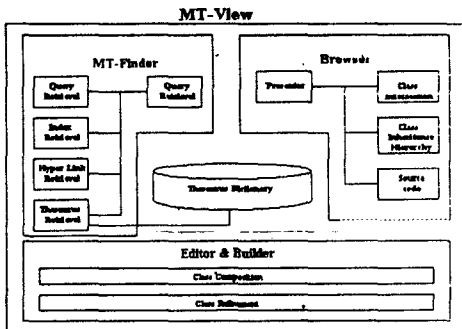
MT-Views는 간단히 Finder와 Browser 및 Builder로 구분된다. MT-Finder와 MT-Browser는 컴퍼넌트 검색과 이해를 위한 부분이며 MT-Builder는 검색을 통해 획득되어진 재사용 컴퍼넌트들은 사용자의 본질적인 요구를 충분히 만족 하도록 편집하고 재구성하는 작업을 담당한다.

(그림 11)는 MT-Views의 시스템 구성도이다.

4. MT-Views의 구현 및 적용 예

4.1 시스템 구현

4.1.1 컴퍼넌트 검색을 위한 질의 작성 서브 시스템
 사용자는 원하는 컴퍼넌트 검색을 위해서는 시스템이 제공하는 일정한 템플리트의 유도에 따라서 혹은 제시되어진 패시 항목의 선택에 의해서, 또는 부품에 대해 사용자가 고려한 임의의 대표 속성의 나열을 직접 작성함으로써 가능하다.



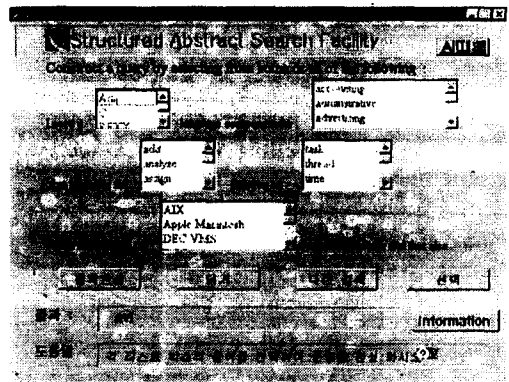
(그림 11) MT-Views의 시스템 구성도
 (Fig. 11) System configuration

(1) 질의 검색

(그림 12)는 SA 메소드로 구축된 질의 검색을 위한 사용자 인터페이스이다. 이 메소드는 일반적인 부품 표현 문장을 구조적인 추상화를 통해 도메인(Domain), 함수(Function), 객체(Object), 운영 체제(Operation system), 언어(Language)의 기능적 특성을 인덱스로 하여 구축된 템플리트이다. 논리적이고 해석적인 형식으로 소프트웨어에 관한 정보를 제시하는데, 같은 순서로 같은 종류의 정보를 나타냄으로써 컴퍼넌트 분류와 프로그램 이해에 관한 문제를 해결한다[4].

(2) 동의어 처리를 통한 검색

동의어 처리는 시스템과 재사용자 간의 어휘적 의미 차이를 최소화하고 명확하지 않는 요구를 구체화함으로써 사용자가 원하는 질의를 쉽게 작성할 수 있다. 이 방법은 구조적이고 관념적인 코드 관점의 정보를 지식 기반 관점에서 기능적 특성으로 통합한 것이다. MT-Views는 동의어 사전 관리 즉, 동의어의 추가, 삭제, 찾기를 사용자에게 허용하여 응용 도메인에 적합한 특화된 동의어를 갖도록 한다.



(그림 12) SA 메소드의 질의 검색 UI
 (Fig. 12) Query retrieval with SA method

(3) 지정 검색

사용자가 막연하며 불투명한 컴퍼넌트 개념으로 검색을 시도할 경우도 도메인에 적합한 부품을 선별 제시함으로써 점차적으로 응용에 필요한 부품의 검색으로 접근 가능 하도록 유도해야 한다. MT-View는 잘 분류되어진 항목들의 리스트를 제공하여 사용자가 추상적 의미의 개념을 검색의 구체화된 항목으로 대치시키도록 세 가지의 검색 방법을 제공한다.

① 이름 검색

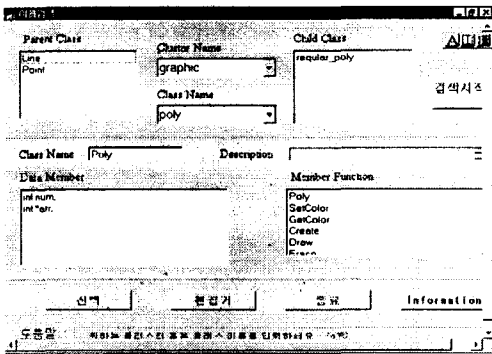
사용자가 응용 도메인을 의미하는 클러스터 항목을 선택하면 해당 클러스터에 속하는 클래스의 이름이 나열된다. 여기서는 제시된 항목의 선택 뿐 아니라 확인하는 부품의 이름을 직접 타이핑함으로써 부품을 획득할 수 있다(그림 13)[2].

② 패시 검색

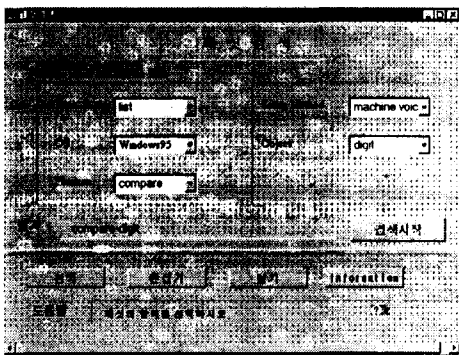
(그림 14)는 MT-Views의 패시 분류에 의한 인터페이스이다. 인터페이스에 나타난 패시는 부품의 효과적인 표현이 가능하도록 항목의 대표성을 지닌 패시로 결정한다.

③ 부울리언 검색

사용자는 제시된 키워드와 각 키워드 간의 관계를 부울리언 연산자를 선택함으로써 원하는 질의를 완성한다. 작성된 질의는 다음 검색에서 오류를 피하며 신속성과 재사용성을 고려하여 질의어 히스토리에 나타낸다.



(그림 13) 이름 검색
(Fig. 13) Name retrieval



(그림 14) 패시 검색
(Fig. 14) Facet retrieval

(4) 선택된 컴퍼넌트에 대한 DB

검색된 부품은 검색 윈도우의 선택 버튼으로 선택 데이터베이스에 등록된다. 이것은 잠정적인 검색 결

과를 수정할 필요가 있을 때 MT-Editor 상에서 검색된 부품의 히스토리를 통해 부품의 소스 코드를 로드할 수 있다. 즉, 통합된 재사용 시스템에서 검색, 편집, 재구축 단계가 일관성 있게 연계될 수 있다.

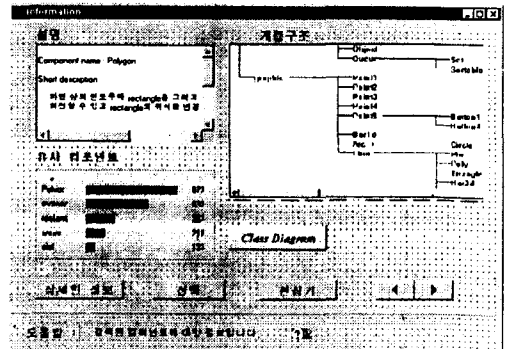
4.1.2 컴퍼넌트 이해를 위한 정보제공 서브시스템

원하는 부품을 검색한 후 자신의 프로그램으로 통합하기 전 잠정적으로 결정되어진 컴퍼넌트에 대한 이해가 필요하다.

MT-Views에서 제공하는 정보는 다음과 같다.

- Class information: 검색된 클래스 자체에 대한 메타 정보로 기능적 서술과 외부와의 인터페이스 함수, 컴퍼넌트 등록 정보 등이 제시된다.
- Class inheritance hierarchy: 클래스 상속 구조를 하이퍼링크로 제공한다.
- Source code: 실제 재사용 대상인 소스 코드를 제공하여 편집, 재구축 할 수 있도록 한다.

(그림 15)는 사용자 이해를 위한 기본 정보로 검색된 컴퍼넌트 이름과 기능적 서술 그리고 외부 인터페이스를 갖는 서술적 정보를 제공한다. 또한 유사 컴퍼넌트에 대해 수치적이며 가시적인 표현 형식으로 그래프와 퍼센트(%)를 나타내며 기능적 유사성에 의한 계층 구조와 클래스 다이어그램을 나타낸다. MT-Views는 상세정보로 원시코드, 속성상속 다이어그램, 컴퍼넌트 등록정보, 예제화일을 제공한다.



(그림 15) MT-View가 제공하는 컴퍼넌트에 대한 기본 정보
(Fig. 15) Basic information of component

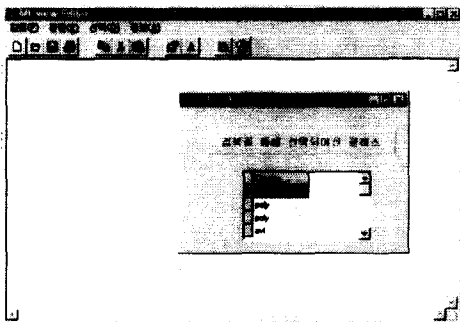
4.1.3 부품의 편집과 구축을 위한 인터페이스

획득되어진 재사용 컴퍼넌트들은 사용자의 원래 요구를 충분히 만족 하도록 자신의 응용 영역에 맞게 새로이 편집 및 재구성하는 작업이 필요하다. MT-Views 에서는 간단한 범용의 에디터인 MT-Editor를 제공하는데 소스 코드의 편집과 재구축 작업은 하나의 연계된 일관된 작업이라 할 수 있으므로 MT-Editor 상에서 통합된 작업으로 처리 가능하다. 다음은 이 서브 시스템의 주요 기능이다.

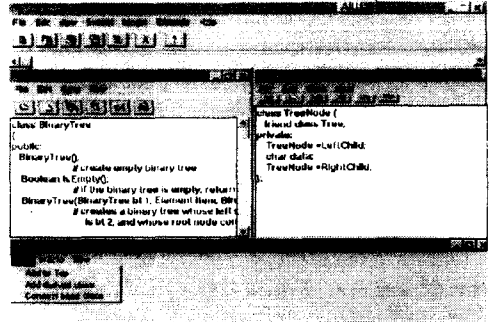
- Class Redefinement: 검색된 컴퍼넌트에 대한 정보를 바탕으로 새로운 클래스를 관련된 클래스들의 다중 상속을 통해 생성한다.
- Class Composition: 검색된 컴퍼넌트에 대한 소스 코드 윈도우를 참조로 일부 코드를 수정하여 새로운 클래스로 재구성 한다.
- C++ 컴파일러 호출: 생성된 컴퍼넌트가 완전한 하나의 클래스로 등록하기 위해 MSVC의 통합 환경(IDE)을 불러 컴파일 한다.

부품의 편집 및 사용자에게 의해 실질적으로 재사용 가능한 단일의 완전한 컴퍼넌트로 재구축하기 위해서 MT-Editor에서는 MSVC의 통합 개발 환경을 호출하여 데이터를 전달 한후 컴파일을 수행한다.

MT-Editor의 메뉴 구성은 기본적인 에디터 형식을 따르고 있으나 2가지의 특별한 기능을 가진다. 하나는 MT-Browser를 통해 검색된 컴퍼넌트 중 사용자가 잠정적으로 선택한 부품을 포함하는 선택 컴퍼넌트 데이터베이스를 제시하는 것이고 다른 하나는 수정



(그림 16) MT-Editor 상의 컴퍼넌트 선택 데이터베이스
(Fig. 16) Selection database in MT-Editor



(그림 17) 부품 상속을 위한 인터페이스
(Fig. 17) Interface for Component inheritance

작업이 완료된 재사용 단위인 C++ 소스 코드를 완전히 실행가능한 코드로 생성하기 위해 MSVC의 통합 환경을 호출한다.

검색 결과를 기록하는 선택 컴퍼넌트 데이터베이스는 (그림 16)에 나타나 있다. 이 데이터베이스로써 검색과 편집, 재구축 일관성 있게 수행할 수 있다.

(그림 17)은 상속을 통한 클래스 재생성을 지원하는 인터페이스로 검색된 부품의 원시 코드가 두개의 윈도우에 나타나 있다. 이것은 MT-Builder에서 제공하는 상속 성격 구분에 따라 새롭게 구성되는 상속 구조에 알맞는 부품을 생성할 수 있도록 지원한다.

4.2. 검색 시스템의 평가

4.2.1 검색 효율



(a) 재현율 = $\frac{\text{검색된 적절한 부품의 수}}{\text{전체 부품 중 적절한 부품의 수}}$

(b) 정확도 = $\frac{\text{검색된 적절한 부품의 수}}{\text{검색된 전체 부품 수}}$

(그림 18) 검색 효율 평가를 위한 부품의 분류
(Fig. 18) Partition of component

재사용 검색 시스템의 성능을 평가하기 위한 두 가지 관점이 있다. 먼저 검색 효율(retrieval effectiveness)은 사용자가 필요로 하는 정보를 제공하는 능력에 관계된 것이며 검색 효과(retrieval efficiency)은 검색을 수행하기 위해 필요한 비용과 노력 측정에 관한 것이다. effectiveness)은 사용자가 필요로 하는 정보를 제공하는 능력에 관계된 것이며 검색 효과(retrieval efficiency)은 검색을 수행하기 위해 필요한 비용과 노력 측정에 관한 것이다.

대부분의 재사용 검색 시스템들은 재현율과 정확도의 두 가지를 파라미터로 사용하여 성능을 측정한다. 정확도는 검색되어진 부품 중 적절한 부품이 얼마나 되는가에 대한 비율이며 재현율은 라이브러리에 존재하는 부품 중 사용자 질의에 부합하는 부품의 검색된 비율을 나타낸다. (그림 18)은 전체 라이브러리의 부품과 정확도와 재현율을 의미하는 부품들 사이의 관계를 보여준다.

MT-Views에서는 부품 평가 중 재현율과 정확도의 평가를 위해 유사 부품의 이름과 각 항목을 나타내는 인터페이스를 생성하여 실험하였다. 전체 250개의 부품을 저장하고 SA 방식으로 각 질의 입력시 가중치를 주어 검색하였다. 응용 영역 중 그래픽과 스트림 입출력에 대해 각 10개의 질의로 실험하였다.

(그림 19)는 검색 효율을 나타내기 위해 SA 사용자 인터페이스에 대한 부품 검색 결과이다. 여기서 사용자 질의는 "탑 다운 메뉴를 가지는 윈도우 생성"을 뜻하는 것으로 동의어 처리 과정을 거쳐 변환된 것이

분야	검색된 적절한 부품	검색된 부적절한 부품	검색되지 않은 적절한 부품	검색되지 않은 부적절한 부품
Graphic	13	2	2	23
I/O stream	9	3	2	26

(a) 검색된 부품의 수

(a) Number of components retrieved

	평균 측정율
재현율	84.5 %
정확도	81 %

(b) 검색 효율

(b) retrieval effectiveness

(그림 20) 검색의 효율성 평가

(Fig. 20) Evaluation of retrieval effectiveness

다. 이 질의로 검색되어진 결과 컴퍼넌트의 이름과 데이터베이스 저장 항목이 나타나 있다.

(그림 20)은 위에서 실험한 SA 방식에 의한 검색 효율을 측정된 것이다.

이름 검색일 경우 라이브러리에 존재한다면 완전하게 질의와 부합하는 컴퍼넌트를 검색할 수 있으나 유사한 다른 부품을 얻을 수 없으므로 한정적이었다. 그러나 다른 방식에 의한 검색에서는 평균 검색 효율이 기존의 검색 시스템의 효율 보다는 다소 높게 나타났다며 특히 이해 시스템으로서의 충분한 정보 제공으로 관련 부품의 획득이 유리하였다.

4.2.2 MT-Views의 특성

(1) 질의어 형식

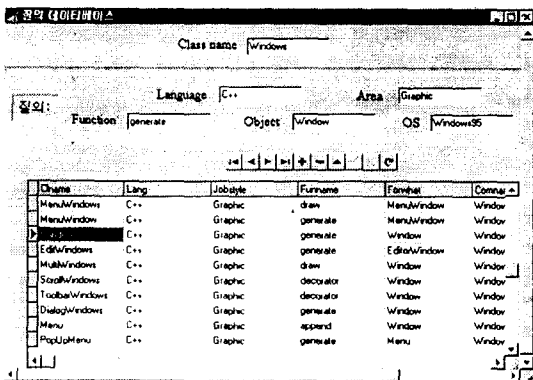
부품 검색을 위한 사용자의 선행 지식을 기반으로 개인적 선호방식에 따라 점진적이며 선택적으로 검색할 수 있는 다양한 질의 유형을 가진다.

(2) 부품 분류

사용자의 개념을 그대로 시스템의 부품 표현 방식으로 매핑시키고 다양한 방법론의 검색을 위해 열거형과 패시형 분류를 단계적으로 혼합한 혼합형(Hybrid) 방식을 사용했다.

(3) 이해 정보

대부분의 기존 시스템들은 검색 기능에만 치중한 나머지 부품의 충분한 이해를 위한 정보 제공이 미흡



(그림 19) SA 방식에 의한 검색 결과
(Fig. 19) Retrieval result in SA method

한데 비해 텍스트, 하이퍼링크, 그래픽 그리고 일반 텍스트 형태로 컴퍼넌트에 대한 기능적 서술, 유사 컴퍼넌트, 상속 구조 및 원시 코드와 예제 화일 등을 충분한 정보를 제공한다.

(4) 라이브러리

문서와 원시코드를 하부구조로 부품의 관련성 정보 및 메타 정보 구성을 위해 인덱스 정보, 프로그램 정보, 유사 정보, 동의어 사전 부분으로 이루어졌다.

(5) 부품 편집과 재구축

단순 검색 시스템이 아니라 통합된 재사용 시스템으로서 MT-Views는 사용자의 의도에 충분히 부합되도록 검색한 부품의 수정을 통해 새로운 컴퍼넌트로 생성할 수 있는 편집기를 제공한다.

4.2.3 다른 시스템과의 비교

관련 연구 논문지 및 기타 참고 도서들을 통해 수집한 기존 재사용 시스템들이 채택하고 있는 방식들을 MT-Views와 비교하면 (그림 21)과 같이 정리할 수 있다. 여기서 시스템 A, B, C는 각각 과기원의 권용래 박사[7], 중앙대의 이경환 박사[14], 전남대의 김병기 박사[5] 연구팀에 의해 제시된 시스템이다. MT-Views는 사용자 선호에 따른 다양한 검색 방법을 제공하며 브라우징에 의한 연결된 부품의 검색이 가능하다. 또한 사용자는 간단한 그래픽 인터페이스로써 쉽게 시스템을 운영할 수 있으며 특히 부품에 대한 상세한 정보 제공이 그 특징이라 할 수 있다.

	MT-Views	A	B	C
분류 모델	열거 + 패킷	열거 + 혼합	패킷	패킷
검색 모델	질의+인덱스+브라우징	질의+브라우징	질의	질의+브라우징
질의	혼합 질의	부울리언	부울리언	벡터 질의
컴퍼넌트 이해	제공		제공	
사용자 인터페이스	풍부한 GUI	GUI	GUI	명령어 중심

(그림 21) 다른 시스템과의 비교
(Fig. 21) Comparison with other systems

본 연구에서는 원하는 재사용 부품을 검색하고 수정하며 사용자의 관점에서 새로이 조립할 수 있는 다중 템플리트 뷰(Multiple-Template Views: MT-Views)를 구현함으로써 프로토타이핑을 위한 환경을 개발하였다. 즉, MT-Views는 요구하는 컴퍼넌트에 대한 개념을 기존 부품으로 빠르고 효과적으로 매핑시도록 사용자 지향의 편리성을 최대 목표로 하는 검색(Browsing), 수정(Editing), 합성(Composition)의 세가지 서브 시스템을 포함하는 통합되어진 재사용 시스템이다.

MT-Views에서 부품의 분류는 사용자의 개념을 그대로 시스템의 부품 표현 방식으로 매핑시키며 다양한 방법에서의 검색을 위해 열거형과 패킷형 분류를 단계적으로 혼합한 혼합형(Hybrid) 방식을 사용하였으며 동의어 사전 및 관리를 제공하여 보다 완벽한 질의 작성을 지원토록 했다. 또한 검색을 위한 질의어 형식은 사용자가 개인적 선호 방식에 따라서 점진적이며 선택적으로 검색할 수 있도록 시스템이 제공하는 일정한 템플리트의 유도에 따라서 혹은 제시되어진 패킷 항목의 선택에 의해서, 또는 부품에 대해 사용자가 고려한 임의의 대표 속성의 나열을 직접 작성으로 가능하다. 그리고 텍스트, 하이퍼링크, 그래픽, 일반 텍스트 형태로 컴퍼넌트에 대한 기능적 서술, 유사 컴퍼넌트, 상속 구조 및 원시 코드와 예제 화일 등을 제공함으로써 부품에 대한 충분한 이해를 바탕으로 응용에 필요한 부품인지 결정할 때 도움을 준다. 다양하고 복잡한 방법에 의한 분류 및 검색 방법으로 많은 데이터 분류, 저장에 필요하며 데이터 관리의 많은 노력이 필요할 수 있다. 그러나 MT-Views가 재사용을 위한 검색 시스템이며 시스템 특성으로 컴퍼넌트의 이해와 사용자 용이성을 제안하고 있다. 따라서 보다 쉽고 정확하게 원하는 컴퍼넌트를 찾고, 찾은 컴퍼넌트에 대한 유사 컴퍼넌트 및 이해 정보를 얻기 위해 사용자 관점에서 보다 많은 시스템 접근 방법을 제공하고자 여러 가지 방법을 사용하였다. 또한 단순 검색 시스템이 아닌 사용자의 의도에 충분히 부합되도록 검색한 부품의 수정을 통해 새로운 부품으로 생성할 수 있는 편집기를 제공한다. 즉, 통합 재사용 시스템으로 새로운 재사용 시스템 구축의 모델로서 뿐 아니라 어플리케이션 생산에 적용으로 생산성과 품질을 향상시킬 수 있다.

5. 결 론

향후 연구 방향으로는 재사용 대상을 소스 코드와 제한적인 문서 인텍싱에 의한 질의 수준에서 설계 정보 및 그 이상의 추상화된 수준의 정보까지 확장하고 유사 부품 검색을 위한 보다 효율적인 검색 알고리즘을 적용하며 라이브러리 구축시 지식 기반의 정보를 활용함으로써 재사용 부품이 응용 개발의 모든 과정을 위해 사용되어질 수 있는 재사용 프레임워크 개발이 필요하다. 아울러 생산된 부품에 대한 품질 평가 도구를 가지는 통합 CASE로의 발전이 요구된다.

참 고 문 헌

[1] Carma McClure, *The Three Rs Of Software Automation: Re-Engineering, Repository, Reusability*, Prentice Hall, 1992.

[2] David E. Brumbaugh, *Object-Oriented Development: Building Case Tools With C++*, Wiley, 1994.

[3] Eliseo Mambella, Roberto Ferrari et al, "An Integrated Approach to Software Reuse Practice", SSR '95, Seattle WA, USA, pp. 63-71, 1995.

[4] Jeffrey S. Poulin and Keith J. Werkman, "Melding Structured Abstracts and the World Wide Web for Retrieval of Reusable Components", SSR '95, Seattle WA, USA, pp. 160-168, 1995.

[5] Moon-Sel Kang, Byung-Ki Kim, "An Extended Facet Classification Scheme and Hybrid Retrieval Model to Support Software Reuse", KIPS, Vol. 1, NO. 1, pp. 23-38, 1994.

[6] Moon-Sel Kang, Byung-Ki Kim, "A retrieval Model of Software Components using Similarity and Weights", KISS, Vol. 22, NO. 1, pp. 69-71, 1995.

[7] Myung-Sup Chang, In-Sang Chung, Yung Rae Kwon, "A Classification Scheme and Retrieval Method of Software Component for Reuse", KISS, Vol. 19, NO. 6, pp. 602-613, 1992.

[8] Richard Helm and Yoëlle S. Maarek, "Integration Information Retrieval and Domain Specific Approaches for Browsing and Retrieval in Object-Oriented Libraries", OOPSA '91, pp.

47-61, 1991.

[9] Rub n Prieto-Diaz, Peter Freeman, "Classifying Software for Reusability", IEEE Software, pp. 6-16, January, 1987.

[10] Salton, G. & McGill, M. J, *Introduction to Modern Information Retrieval*, McGraw Hill Book Company, 1983.

[11] Ted Biggerstaff and Charls Richter, "Reusability Framework, Assessment, and Directions", IEEE Software, pp. 41-49, Match, 1987.

[12] 김 행곤, 프로토타이핑 지원을 위한 재사용 시스템 개발에 관한 연구, 최종 보고서, 한국전자 통신연구소, 1996, 10.

[13] 이 경환, 소프트웨어 자동 생산 기술에 관한 연구, 과학 기술처, 1993. 10

[14] 최 은만, 김 진석, "객체지향 재사용과 CASE", KISS, Vol. 14, NO. 10. pp. 47-54, 1996.



김 행 곤

1985년 중앙대학교 전자계산학과 졸업(학사)
 1987년 중앙대학교 대학원 전자계산학과 졸업(이학석사)
 1991년 중앙대학교 대학원 전자계산학과 졸업(공학박사)
 1978년~1979년 미 항공우주국
 객원 연구원

1987년~1990년 한국전기통신공사 전임연구원
 1988년~1989년 AT&T 객원 연구원
 1990년~현재 대구효성가톨릭대학교 컴퓨터공학과
 부교수

관심분야: 객체지향시스템 설계, 사용자 인터페이스,
 소프트웨어 제공학, 유지보수 자동화툴,
 CASE



차 정 은

1995년 효성여자대학교 전자계산학과 졸업(학사)
 1997년 대구효성가톨릭대학교 대학원 전산통계학과 전자계산학 전공(이학석사)
 현재 대구효성가톨릭대학교 대학원 전산통계학과 전자계

산학 전공 박사과정 재학
 관심분야: 객체지향, 소프트웨어 재사용, CASE