

클래스 부품 검색을 위한 Viewer의 설계 및 구현

정미정*, 송영재**
경희대학교 전자계산공학과

Design and Implementation of a Viewer for Class Components Retrieval

Mi-Jung Jung* Young-Jae Song**
Dept. of Computer Engineering, KyungHee University

Abstract

Many similar class components are stored in object-storage but the object-storage has needed the retrieval function of correct component for reuse. Accordingly this paper designed the class component retrieval viewer of the object-storage by using the improved spreading activation strategy. Object-storage has made up of information of inheritance relation, superclass, subclass, and we defined the queries about each class function. Also we specified connectionist relaxation of the each class and query, finally we gained retrieval result which showed highest activation value order of class component information including the query function.

I. 서론

소프트웨어 재사용은 소프트웨어 위기 현상을 극복하기 위한 방안으로 재사용(reuse) 방법이 널리 연구되고 있다. 현재의 재사용 기술은 객체저장소 구축에 필요한 높은 비용과 재사용할 수 있는 정보를 정확히 검색하는 효과적인 검색도구의 부족으로 어려움을 겪고 있다. 따라서 정보를 효율적으로 검색하기 위해서는 보다 정확한 정보검색 알고리즘이 선행되어야 한다[1].

정보검색 알고리즘 중 Spreading Activation[2] 방법은 질의어와 연관되어 있지만 정확히 일치되지 않는 유사 클래스를 검색하기 위해 사용된다. 그러나 검색과정에 많은 시간이 소요되는 단점이 있다. 이는 검색시 활성값을 이용하여 유사도를 측정하기 때문에 활성값은 정확한 결과를 생성하는데 가장 큰 역할을 하지만 검색시간을 지연시키는 역할을 하는 것도 사실이다.

본 연구는 클래스 부품의 재사용을 목적으로 사용자가 검

색하고자 하는 기능을 담고 있는 클래스 부품을 정확히 알고 있지 않을 때, 폭 넓은 확장검색을 함으로서 객체의 효율적 검색이 이루어지도록 하기 위해 Spreading Activation 알고리즘을 이용하였다.

Spreading Activation 알고리즘을 가지고 검색 결과를 최대한 유지하면서 검색시간을 최대한 줄이는 방법으로 개선시켰다. 그리고 객체저장소는 슈퍼 클래스, 서브클래스, 상속관계 정보로 구성되는 클래스 부품들로 구성하였다. 그리고 새로운 클래스를 추가하여서 검색할 수 있도록 구성하였다. 검색을 위한 질의어는 여러 클래스들을 포함하는 클래스 기능으로 선정하였다. 질의어는 리스트화 시켜 사용하기 용이하도록 하였고, 새로운 클래스 부품의 추가 시 질의어와의 관계를 정의할 수 있도록 클래스, 질의어 관계 인터페이스를 설계하였다. 따라서 본 연구는 Spreading Activation 방법을 개선시키고, 객체저장소에 있는 클래스 부품과 질의어 관계를 이용하여 효율적인 클래스 부품 검색 과정을 Viewer를 통해서 구현하였다.

II. 관련연구

소프트웨어 부품을 재사용하기 위한 분류 방식과 검색방법은 소프트웨어 재사용을 실용화시키기 위해 매우 중요하다. 부품을 분류하는 방식에 따라 분류된 부품을 포함하는 소프트웨어 라이브러리의 구조와 사용자의 요구사항에 일치하는 부품을 검색하는 검색 방법의 선택이 결정되기 때문에 분류 방식과 검색 방법은 서로 밀접한 관계를 유지한다. 다음은 이러한 검색 방법과 분류방법을 담고 있는 정보검색시스템이다.

1. 정보검색시스템

① RSL

RSL[3]은 재사용 부품의 속성으로서의 재사용 주석을 몇

가지 정의하고 각각의 속성 이름과 기능 설명을 가진 레이블을 할당한다. 정보 추출기는 적절하게 주석이 달린 부품의 PDL과 소스코드로부터 재사용에 필요한 정보를 추출하여 RSL 데이터베이스에 저장한다. 메뉴 방식으로 처리되는 사용자 질의어는 레이블에 있는 재사용에 필요한 정보를 바탕으로 작성되는데, RSL은 정보검색방법을 기반으로 하고 있다. 검색 메커니즘은 질의어의 키워드 집합을 받아들여 저장되어 있는 키워드와 비교한 후 일치되는 부품을 찾아낸다.

② CATALOG

CATALOG[4]는 비 구조화된 자료를 처리할 수 있으며 구축된 데이터베이스 내의 각 모듈이 헤더 기술서로부터 인덱싱에 필요한 정보를 얻어낸다. 검색은 초보자를 위한 메뉴 인터페이스와 전문가를 위한 명령어 위주의 탐색 인터페이스를 제공하고, 질의어와 부분적인 스트링 매칭 기법을 허용한다. 또한 의미 없는 단어들을 정지 리스트(stoplist)에 의해 처리된다.

③ Diaz

Diaz[5]의 분류 메커니즘은 모듈들이 갖는 열거 속성들 각각 패시 별로 나누어 이해하므로 모든 모듈들은 각 패시에 해당하는 원소들을 조합시켜 원하는 기능을 조합시켜 원하는 기능을 정의할 수 있고, 분류 대상에 따라 적절히 적용하기 쉬울 뿐만 아니라 지속적으로 모듈의 확장되는 경우에도 계층적 방법보다 쉽게 대처할 수 있다. 검색에 사용되는 질의어는 수정과 확장이 가능하도록 구성되어 있다. Diaz의 검색 방법은 영역 모델이 설정되어 질의어 구성과 재구성에 사용되게 된다.

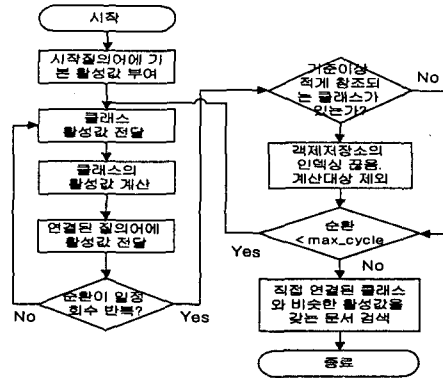
2 개선된 Spreading Activation 알고리즘

Spreading Activation[2] 검색은 클래스와 질의어 사이에서 질의어 기능을 포함하고 있는 유사한 클래스를 찾아냄으로써 정확한 결과를 알지 못하고 있는 사용자에게 확실한 결과를 찾을 수 있게 도와주는 데에 중점을 맞춘 알고리즘이다. 검색과정은 정보저장소에서 클래스의 기능을 포함하는 질의어로 시작해서 질의어와 클래스의 관계성을 계산하기 위해서 활성값을 사용하게 된다. 각 질의어와 클래스 사이의 연결은 질의어가 클래스의 색인임을 나타낸다. 이 연결은 각각의 강도를 가지고 있으며, 연결강도는 클래스와 질의어의 관계정도를 나타낸다.

Spreading Activation 검색방법에서 활성값을 계산하는 순환을 반복하는 것은 정확한 활성값을 얻기 위함이다. 이 방법으로는 각 질의어와 클래스의 활성값은 다른 질의어와 클래스들의 활성값을 계산하는데 영향을 주게된다. 따라서 인덱싱이 적은 클래스들을 끊어버림으로서 활성값 계산이 줄어들게 된다. Spreading Activation 방법을 개선한 구조는 <그림 1>과 같다. 기본 활성값을 이용하여 계산하는 과정에 연결된 클래스의 활성값 역시 계산된다. 순환과정이 일정 수준 반복된 후 기준에 미치지 못하는 질의어들은 인덱싱을 끊어 계산 대상에서 제외시킨다[6].

그 결과는 검색하고자 하는 클래스와 질의어들까지 활성

값에 따라 검색할수 있고, 검색속도를 37.8%를 향상시켰다.



<그림 1> 개선된 Spreading Activation 구조

<fig. 1> construction of improved spreading Activation

III. 클래스 부품 검색 Viewer

1. 클래스 부품 정보와 질의어

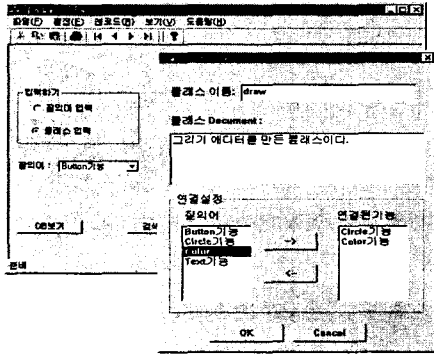
클래스 정보는 클래스 명, 멤버함수, 속성, 슈퍼 클래스, 서브클래스의 관계, 클래스와 질의어의 관계들로 구성되고 이 정보를 이용하여 각 클래스와의 상속관계를 중심으로 정보를 추출하였다. 이를 위하여 클래스 부품을 구문 분석하고 클래스내의 모든 정보를 상호 관련 정보로 구축하였다. <그림 2>는 Viewer에서 사용하는 데이터들을 정의해놓은 것이다. 또한 질의어는 각 클래스의 기능을 포함할 수 있도록 정의하여 각 클래스와의 관계를 지정해주면 활성값이 자동으로 계산된다. 질의어는 인터페이스 상에서 리스트 형식으로 보여줌으로서 질의어 사용에 편리함을 제공하였다. 질의어를 이용하여 검색을 시작하면 질의어와 직접 연결된 클래스 부품과 간접 연결된 클래스 부품들이 검색된다.

<그림 2> 데이터베이스 구성

<Fig. 2> construction of database

<그림 3>는 클래스와 질의어 관계 정보를 입력하는 대화

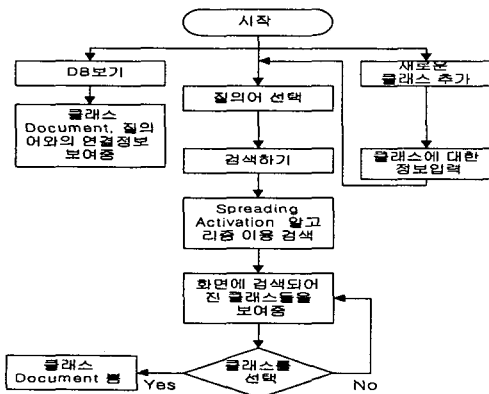
상자이다. 클래스 이름과 클래스 도큐먼트를 넣고 클래스와 질의어 사이에 연결 정보를 넣는다. 이때 연결설정에 주어지는 리스트들은 데이터 베이스 내에 있는 질의어를 자동으로 읽어와서 사용자가 잘못된 질의어와 연결되지 못하도록 하였다. 또한 버튼을 통해서 추가와 삭제가 가능하며 연결된 질의어를 한눈에 보여주므로 잘못된 연결을 사전에 방지해 준다.



<그림 3> 클래스와 질의어 관계 정보 입력
<Fig 3> information of class and query relationship

2. 검색 Viewer 설계

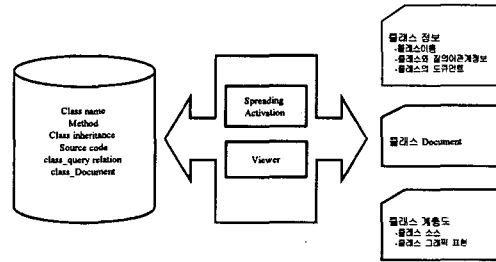
본 논문의 Viewer는 객체저장소에 저장되어 있는 객체 정보와 연결정보를 이용하여 Spreading Activation 검색 과정을 사용하였다. <그림 4>는 Viewer의 알고리즘을 보여주고 있다. Viewer는 클래스 입력, 데이터베이스 보기, 검색하기, 도큐먼트 보기로 설계하였다. 클래스 입력창은 검색할 클래스의 확장성을 고려하여 현재 데이터 베이스 내에 있는 클래스에 추가, 검색이 가능하도록 하였다. 입력창에서는 새 클래스와 질의어의 관계를 규정하였고, 질의어와 클래스의 관계를 나타낼 때는 리스트에 허용 가능한 질의어를 보여주므로써 사용자에게 잘못 선택할 수 있는 경우를 배제하게 해주었다.



<그림 4> Viewer 알고리즘
<Fig. 4> Algorithms of Viewer

데이터베이스 보기는 클래스에 대한 정보를 사용자가 한눈에 확인할 수 있도록 클래스와 질의어와의 관계를 보여주고, 도큐먼트를 보여준다. 현재 입력된 클래스를 한눈에 파악할 수 있는 최소한의 데이터만을 보여준다. 또한 검색된 결과에서 각각의 클래스마다 도큐먼트를 볼 수 있도록 설계하였다.

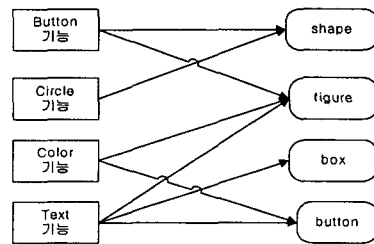
<그림 5>은 검색시스템의 구조도를 보여주고 있다. 모든 데이터는 데이터베이스에 저장되고, 질의어에 연결된 활성값이 계산되어 안정적이 될 때까지 반복된다. 검색 결과는 기준이 되는 활성값 범위에 해당하는 모든 클래스 부품의 최상위 클래스인 BaseClass가 검색된다. 그리고 Viewer의 기능을 통하여 검색된 클래스와 연결된 클래스 부품의 정보와 활성값을 좀더 효율적으로 보여준다. 활성값은 참조가 많이 될수록 값이 커지며 유사한 클래스 부품 역시 참조 기능으로서 검색된다. 이와 같이 검색된 부품들은 사용자에게 여러 개의 부품을 제공하기 때문에 재사용을 위한 클래스 부품을 선택할 수 있는 폭이 넓어지며, 사용자에게 보다 더 클래스 부품에 대한 이해도를 증진시킬 수 있다.



<그림 5> 검색 Viewer 구조
<Fig. 5> construction of retrieval Viewer

3. 클래스 부품 검색

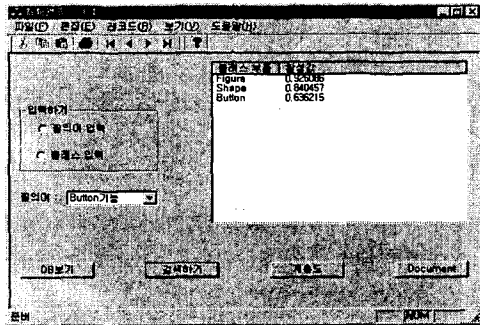
<그림 6>는 질의어와 클래스의 연결관계를 보여주고 있다. Button이라는 질의어를 통하여 3개의 클래스가 검색되어진다. 이때 질의어 Button은 shape와 figure 클래스에 직접 연결되어 있지만 button 클래스와 box 클래스는 연결되어 있지 않다. Button(질의어)→Figure(클래스)→Text(질의어)→button(클래스)을 통하여 선택되고, Button (질의어)→Figure(클래스)→Text(질의어)→box(클래스)를 통하여 2개의 클래스(box, button)가 연결됨을 알 수 있다. 하지만 box는 결과에서 적게 참조되므로 연결을 끊으므로써 결과화면에서 찾아볼 수 없다.



<그림 6> 클래스와 질의어의 관계

<Fig. 6> relationship of class and query

이처럼 각 질의어와 클래스 부품은 서로 연결되어 있는 노드를 참조해 가면서 활성값을 계산하게 된다. 또한 <그림 7>에서 보여주고 있는 검색 결과의 활성값은 shape 클래스가 figure 클래스 보다 높게 나타났다. 그 이유는 shape가 figure 보다 참조 회수가 더 많기 때문이다.



<그림 7> 검색 결과

<Fig. 7> Retrieval result

성되는 클래스 부품들로 구성하였고 검색을 위한 질의어는 각 클래스들을 포함할 수 있는 클래스의 기능으로 정의하였다. 질의어는 리스트화 시켜 사용하기 용이하도록 하였고 새로운 클래스를 추가하여서 검색되어질 수 있는 Viewer를 구현함으로써 부품의 이해도를 높이도록 하였다.

앞으로의 연구방향은 질의어와 클래스 부품정보를 확장하여 객체의 재사용 도구를 개발하는데 있다.

비교항목	RSL	Diaz	CATALOG	본 논문제시 방법
부품의 성격	함수	함수	함수	클래스
라이브리리 구성요소	함수형부품, 정해진 레이블에 따라 기술된문서	함수형 부품, 동의어 사전	비구조화된 자료에 헤더 기술서	클래스 정보, 클래스 vs 질의어의 관계정보, 설계문서
색인방법	정보검색 방법	정보검색방법, Fact의 term	정보검색 방법	계층구조, 클래스 및 질의어 관계정보
질의어 성격	자연어	제안된 Fact 자연어	자연어	자연어(클래스의 기능)
검색방법	keyword matching	keyword matching	스트링 matching	활성값 이용

<표 1> 기존시스템과의 비교

<Table 1> Comparison of existion systems

IV. 비교 평가

본 연구는 Spreading Activation 방법을 개선시켜 클래스 부품을 검색할 수 있도록 구현하였고, <표 1>처럼 기존의 시스템과 비교 평가하였다. RSL[3]은 저장시와 검색시에 사용되는 키워드의 의미를 파악하기 어렵고, 질의어 재구성과 브라우저 지원을 위한 부품 관리가 어렵다. CATALOG[4]는 라이브러리의 확장성, 부품의 이해, 부품의 선택과 수정을 통한 통합의 문제점이 있다. Diaz[5]는 한번 분류가 설계되고 나면 고정적이 된다는 제한성을 지닌다. 따라서 본 연구는 이러한 단점을 보완하기 위해서 질의어를 추가하여 재구성할 수 있도록 하였고, 부품의 이해를 위한 클래스 정보를 Viewer를 통하여 보여주도록 하였다. 데이터베이스에 클래스 부품들을 추가 삭제할 수 있도록 하여 검색범위의 확장성을 갖도록 하였고 Spreading Activation 방법을 보완하여 활성값 계산이 적은 부분을 끊음으로서 검색시간이 37.8%의 향상과 함께 검색 결과를 쉽게 볼 수 있도록 하였다.

V. 결론

본 연구는 소프트웨어 재사용을 위한 클래스 부품정보를 효율적으로 검색하기 위해 Spreading Activation 검색 결과를 최대한 유지하면서 검색시간을 최대한 줄이는 방법으로 클래스 부품 검색시스템을 구현하였다.

RSL[3]과 CATALOG[4]의 단점을 보완하고 질의어를 통한 Spreading Activation 방법으로 클래스 부품을 검색하도록 하였다. 또한 검색 결과는 질의어와 연관되어 있지만 정확히 일치되지 않는 유사 클래스까지 검색할 수 있도록 하였다. 객체저장소는 슈퍼 클래스, 서브클래스, 상속관계 정보로 구

참고문헌

- [1] Scott Henninger, "Information Access Tools for Software Reuse", System Software, pp. 231-247 1995.
- [2] Paul R, Cohen and Rick Kjeldsen, "Information Retrieval by Constrained Spreading Activation in Semantic Networks", Information Processing & Management, Vol. 23, No. 4, pp.255-268, 1987.
- [3] W.B.Frakes and B.A. Nejme, "An Information System for Software Reuse", Proceedings of the Tenth Minnowbrook Workshop on Software Reuse, 1987
- [4] B.A. Burton, R.W. Aragon, S.A. Bailey, K.D. koehler and L.A. Mayers, "The Reusable Software Library", IEEE software, pp.25~33, July 1987.
- [5] R.Prieto-Diaz and P.Freeman, "Classifying Software for Reusability", IEEE Software Vol.4, pp.6~16, Jan.1987.
- [6] 허선영, 송영재, "정보저장소 검색을 위한 Spreading Activation 방법의 개선 방안에 관한 연구", 한국정보처리학회 '98 추계 학술발표논문집, 제5권 제2호 pp. 494-497, 1998.