

유사부품 검색을 위한 개선된 SARM

Enhanced SARM for Similar Components Retrieval

한정수*, 김귀정**

천안대학교*, 건양대학교**

Han Jung-Soo*, Kim Gui-Jung**

Cheonan University*, Konyang University**

요약

본 연구는 기존의 유사 부품 검색 알고리즘인 spreading activation 방법의 단점인 속도의 문제를 해결하여 보다 효율적인 부품 검색을 할 수 있도록 개선시켰다. 또한 개선된 SARM을 이용한 부품 정보와 질의어 관련 인터페이스를 구축하였고 이를 통한 부품 검색 시스템을 구축하였다.

Abstract

This paper enhanced similar components retrieval algorithm-spreading activation method. The solution is that retrieval speed is overcome and more efficiently components are retrieved. Also we implemented interface for component information and query, and constructed component retrieval system.

I. 서론

객체 지향 방법론(Object-Oriented Methodology)의 영향으로 객체의 효율적 관리와 사용 방법에 관한 관심이 높아지고 있으며 OMT(Object Modeling Technique), OOA/OOD 등의 객체지향 방법론이 등장하였다[1]. 또한 이들의 단점을 보완한 UML(Unified Modeling Language) 등의 방법론이 제시됨에 따라 객체지향 소프트웨어를 개발하는데 필요한 많은 객체들이 요구되고 있다. 이러한 객체들을 효율적으로 저장, 관리, 검색, 재사용 등의 활용을 위한 주요 역할을 하는 것이 정보저장소(Repository)이다. 객체지향 부품(Object-Oriented Components)은 기존의 소프트웨어를 분석하여 재사용 가능한 부품단위로 정보저장소에 저장한 후 사용된다[2].

본 연구는 기존의 부품들을 보다 효율적으로 검색하기 위하여 기존의 SARM 검색방법의 속도 문제를 개선시키고, 유사 부품 검색을 위한 개선된 SARM (enhanced spreading activation retrieval method)에 관하여 기술하였다.

II. 관련연구

RSL[3]의 검색 메커니즘은 키워드를 비교하여 일치되는 부품들을 검색한다. 검색된 부품들은 품질 평가항목의 중요도에 따라 재평가된 후 가장 적합한 부품을 제공한다. 그러나 부품 저장과 검색에 사용되는 키워드에 따라 시스템의 효율이 결정된다는 특징이 있다. 즉, 검색할 때 사용되는 키워드의 의미를 파악하기 어렵기 때문에 질의어 재구성이나 브라우징을 통한 부품관리가 어려운 단점을 가지고 있다.

Diaz 분류 메커니즘[4]은 모듈들이 갖는 열거 속성들을 각각 패킷 분류하여 원소들을 조합시켜서 원하는 기능을 정의할 수 있고 분류 대상에 따라 적절히 적용하기 쉬운 뿐 아니라 지속적으로 모듈의 집합이 확장되는 경우에도 계층적 방법보다 쉽게 대처할 수 있다. 그러나 한번 분류가 설계되면 고정적이 된다는 제한성을 갖는다.

CATALOG[5]는 메뉴 인터페이스와 전문가를 위한 명령어 중심의 탐색 인터페이스를 제공하고, 부울리언 질의와 부분적인 스트링 매칭기법을 허용한다. 그러나 라이브러리의 확장성, 부품의 이해, 부품의 선택과 수정을 통한 통합 등의 문제를 해결하지 못하였다.

SARM[6]은 부품과 질의어 사이에 질의어 기능을 포함하는 유사한 부품들을 검색하여 보다 더 넓은 범위의 부품들을 찾을 수 있는 방법이다. 그러나 SARM은 활성화율을 이용한 반복 계산으로 유사도를 측정하기 때문에 검색시간을 지연시키는 단점이 있다. 이는 부품들이 증가할수록 활성화율의 계산회수가 지수적으로 증가하기 때문에 부품이 많을수록 검색에는 어려움이 있다.

[표 2] 질의어와 부품의 link 참조 회수

Link	Link 참조회수 (회)	
A - W	✓	4
A - X		12
B - W	✓	4
C - X		9
C - Z		9
D - X		10
D - Y	✓	6
D - Z		8

III. 개선된 SARM

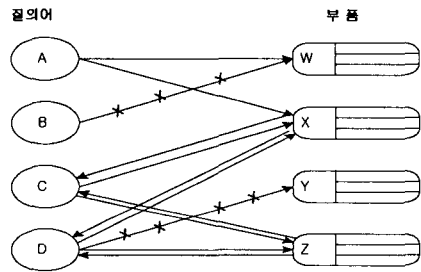
1. SARM 개선 방법

SARM은 질의어에는 연관되어 있지만 정확히 연결되지 않는 부품을 검색하기 위해 사용하지만 많은 검색시간을 요구한다. 표 1은 부품과 질의어 크기에 따른 활성화율 계산 회수를 나타낸 것이다. 부품과 질의어의 크기가 커짐에 따라 활성화율 계산 회수가 지수적으로 증가함을 알 수 있다. SARM은 검색이 끝날 때까지 노드간 연결을 따라 계속 활성화율이 전달된다. 노드간 연결이 계속 참조되는 순환이 반복됨에 따라 어떤 연결은 자주 참조되지만, 어떤 연결은 다른 연결들 보다 훨씬 적게 참조된다.

표 2는 그림 1에서 검색 과정이 끝날 때까지의 link 참조 회수를 측정된 결과이다. 이 실험에 의하면 직접 연결된 노드와 같이 질의어의 결과로 검색되는 노드들은 검색되지 않는 노드들 보다 많이 참조된다. 반대로 많이 참조되지 않는 노드들은 검색되지 않는 노드들의 연결을 제거시키면 노드의 활성화율 계산 회수를 줄여 검색 시간을 단축시킬 수 있다.

[표 1] 질의어에 대한 활성화율 계산 회수

질의어와 부품 크기	SARM 활성화율 계산 회수
200	22084.09
400	35121.7
600	82655.78
800	116527.01
1000	231957.428



▶▶ 그림 1. 질의어와 부품의 연결 제거

그림 1에서 질의어 A로 검색할 때, 검색 결과는 W, X, 그리고 Z이다. 여기서 Z 검색은 A→X→C→Z→D→X→A와 A→X→D→Y로의 두 가지가 된다. 두 흐름의 참조 회수를 비교해 보면 첫 번째 흐름보다 A→X→D→Y의 흐름이 더 적게 참조된다. 두 흐름 중 Z를 검색해 내는 흐름은 A→X→C→Z→D→X→A이다. 그러므로 결과적으로 볼 때, Z를 검색하는데 있어서 A→X→D→Y로의 흐름은 사실상 불필요한 흐름이 되고 만다. 이와 마찬가지로 질의어 A와 관련된 것을 검색하는데 별 영향을 미치지 못하는 질의어 B와 부품 W의 연결도 다른 연결보다 적게 참조된다. 그런데도 불구하고, 활성화율 계산은 A와 관련된 것을 검색하는데 불필요한 B나 Y까지도 모두 계산해야 한다. A→X→D→Y로의 흐름에서 D→Y의 연결을 제거하면 이 흐름은 없어지게 되고, 그만큼 계산 횟수는 줄어들게 되고, 검색 결과에는 영향을 미치지 않는다.

2. 개선된 SARM 검색 방법

E-SARM의 검색 방법은 표 3과 같다. E-SARM의 검색 순서 1~4까지는 SARM의 순서와 동일하다. 초기

활성값으로는 1.0의 값이 할당되고, 이 초기 활성값은 순환이 반복될 때마다 다시 시작 항목에 주어진다. SARM의 검색 결과는 최대한 유지하면서, 검색 시간은 최소한으로 줄이기 위한 제거 기준을 결정하기 위해 참조 회수를 이용하여 시뮬레이션 한다. 우선 평균 참조 회수를 가장 큰 기준으로 설정하고, 평균 참조 회수의 10%, 50%, 80%를 각각 기준으로 설정하여 검색결과와 활성값 계산 회수를 산출한다. 이것을 같은 샘플을 사용했을 때 SARM을 사용한 경우와 비교하여 가장 작은 검색 결과의 차를 갖으면서, 계산 회수는 적은 제거 기준을 찾는다. 순환이 반복될수록 활성값은 안정되며, 참조 회수가 기준에 미달되는 부분은 자동으로 제거되어 계산 과정이 종료된다. 여기서 참조회수의 기준은 E-SARM의 시뮬레이션 결과 평균 참조회수의 88%가 되는 참조 회수를 기준으로 정의하였다.

이 알고리즘은 참조할 때마다 각각의 질의어와 부품에 대한 활성값이 계산한 후 평균 방문회수와 비교하여 기준이 미달되는 노드를 제거시킨다. 이와 같은 방법으로 참조회수에 대한 노드를 제거시켜 활성값 계산회수를 줄임과 동시에 속도를 개선시키게 된다. 검색 결과는 기존의 SARM의 결과를 최대한 유지시킨다. 제거 기준은 시뮬레이션을 통하여 설정하였고, 기존의 SARM과 1.23개의 검색 부품의 오차는 존재하지만 최대한 유사 부품을 보다 빠르게 검색할 수 있도록 개선시켰다.

[표 3] E-SARM 검색 방법

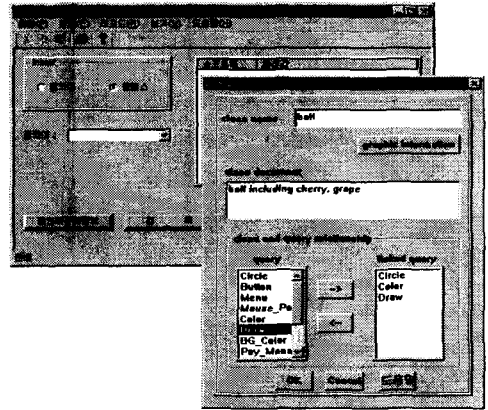
1. 검색하고자하는 시작 질의어를 입력받는다.
2. 시작 질의어에 초기 활성값 1.0을 할당한다.
3. 초기 활성값은 질의어에 연결된 부품으로 이동
4. 활성값은 시작 질의어에 연결된 부품으로 이동
5. 순환이 1회 반복되었는지 확인한다.
6. 2회 이상 반복되면 노드간 연결의 참조 회수를 이용하여 제거 기준을 설정한다.
7. 참조 회수가 제거 기준에 못 미치면 연결 삭제
8. 5, 6의 과정을 최대 순환까지 반복한다.
단, 직접 연결된 것은 삭제 않는다.
9. 순환이 끝나면 시작 질의어와 직접 연결된 부품의 활성값과 유사한 값을 갖는 부품들을 검색

IV. 객체지향 부품 정보와 검색

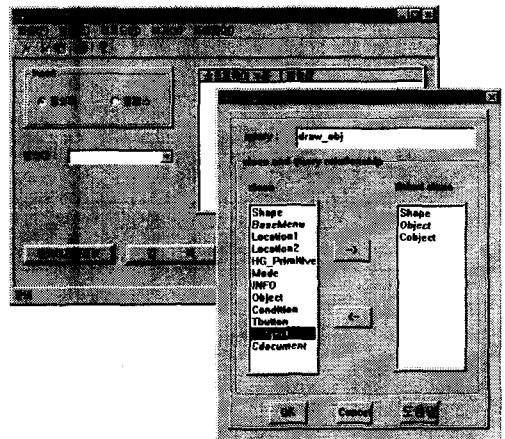
1. 부품 정보와 질의어

소프트웨어 부품 정보는 클래스명, 멤버함수, 속성, 슈퍼클래스와 서브클래스의 관계, 소스코드, 부품과 질의어 관계로 구성된다. 각 부품들은 클래스들로 구성되어 있기 때문에 부품명을 최상위 클래스명으로 정의하였다.

그림 2는 클래스와 질의어를 데이터베이스에 추가, 삭제할 때 관계정보를 입력하는 인터페이스이다. 또한 질의어 삽입 역시 그림 3과 같이 질의어를 입력하고 부품 목록에서 관계되는 부품들을 선택하여 연결시킬 수 있도록 하였다. 이와 같이 부품과 질의어 생성 인터페이스를 제공하여 새로운 질의어와 부품 사이의 연결 정보를 설정, 삭제가 가능하도록 하였다.



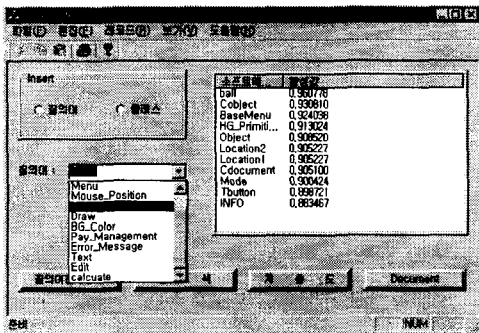
▶▶ 그림 2. 부품 추가와 질의어 관계 설정



▶▶ 그림 3. 질의어 추가와 부품 관계 설정

2. 객체지향 부품 재사용

정보저장소에 저장되어 있는 부품 정보와 연결정보를 이용하여 질의어와 직접 연결된 부품과 간접 연결된 유사 부품들이 개선된 SARM 알고리즘을 통하여 검색될 수 있도록 하였다. 모든 데이터는 정보저장소에 저장되고, 질의어에 연결된 활성값이 계산되어 안정적이 될 때까지 반복된다. 입력 인터페이스에서는 새로운 부품과 질의어와의 관계를 규정하였고, 질의어와 부품과의 관계를 나타낼 때는 리스트에 허용 가능한 질의어를 보여줌으로써 오류를 방지할 수 있도록 하였다. 그림 4는 질의어 Color를 입력하여 이와 연결된 부품들 중 활성값이 큰 순서로 부품들이 검색된 결과를 보여준다. 검색 결과는 참조회수가 가장 많은 부품 ball을 중심으로 11개의 부품이 검색되었다. Color와 직접 연결된 부품들은 상위 5개인 부품 Object까지 직접 연결된 부품이고 나머지는 간접 연결된 유사한 부품들이다. 검색된 부품들은 질의어 Color의 기능을 갖는 클래스들을 포함하고 있다.



▶▶ 그림 4. 부품 검색 시스템

만 유사부품 검색이 가능하고, 검색 결과가 질의어에 연결강도가 큰 활성값 순으로 검색되기 때문에 부품의 선택이 보다 쉽다는 장점을 갖고 있다. 또한 활성값 순으로 검색되기 때문에 패턴매칭에 의한 검색 보다 부품 선택 비용이 적다는 사실을 시뮬레이션을 통하여 증명하였다. 이 결과는 부품의 크기가 클수록 본 연구에서 제시한 방법이 패턴매칭 방법 보다 효율적인 부품 검색과 재사용 부품 선택 비용이 적다는 결과를 얻었다.

■ 참고문헌 ■

- [1] Rumbaugh, J. et al., "Object-Oriented Modeling and Design," Prentice-Hall, 1991.
- [2] James Petro and Michael E. Fotta, "Model-Based Reused Repositories-Concepts and Experience," IEEE Computer Society Press-Technical Council on Software Eng., pp.60-69, 1995.
- [3] 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.
- [4] R. Prieto-Diaz and P. Freeman, "Classifying Software for Reusability," IEEE Software, Vol.4, No.1, pp.6-16. Jan. 1987.
- [5] W. B. Frakes and B.A. Nejme, "An Information System for Software Reuse," Proceedings of the Tenth Minnowbrook Workshop on Software Reuse, 1987.
- [6] Scott Henninger, "Information Access Tools for Software Reuse," System Software, pp.231-247, 1995.

V. 결론

SARM 검색과정에서 부품의 참조회수가 적은 부분을 제거시키기 위한 제거 기준 설정을 위해 SARM과 E-SARM의 시뮬레이션을 실험하였고, 많은 시간이 걸리는 단점을 개선시켜 SARM에 비해 37.8% 단축된 시뮬레이션 결과를 얻었다. 그러나 본 연구의 검색 방법은 패턴매칭에 의한 방법보다는 속도가 느린 단점이 있지