재사용 부품 검색 시스템에서 객체기반 시소러스를 이용한 패싯 질의의 확장

(Facet Query Expansion with an Object-Based Thesaurus in Reusable Component Retrieval Systems)

최 재 훈 † 김 기 헌 † 양 재 동 †† 이 동 길 †††

(Jae-Hun Choi) (Ki-Heon Kim) (Jae-Dong Yang) (Dong-Gil Lee)

요 약 패싯기반 재사용 부품 검색 시스템에서 사용자가 검색하고자 하는 부품들의 특징은 일반적으로 패싯 질의에 의해 명시된다. 본 논문에서는 객체기반 시소러스를 이용하여 사용자의 검색 요구를 명확히 표현할 수 있는 확장된 패싯 질의를 정형화하며, 이 질의를 평가할 수 있는 부품 검색 시스템을 설계하고 구현한다. 시스템의 정확한 검색을 위해 사용자 질의는 검색하고자 하는 부품들의 특징을 구체적으로 명시할 수 있어야 한다. 그러나, 기존의 패싯 질의는 단지 사용자에 의해 직접 입력된 패싯 값들의 나열로만 표현되기 때문에 구체적인 사용자 의도를 자연스럽게 표현할 수 없다는 단점을 가지고 있다. 본 논문에서 정형화되는 확장된 패싯 질의는 이 단점을 보완하기 위해 퍼지 불리언 연산자와 객체기반 시소러스를 이용한다. 전자는 패싯 질의와 관련 부품에 대한 퍼지 연관 정도를 논리적으로 표현할 수 있게 하며, 후자는 사용자가 구체적인 의미의 패싯 값들을 질의에 쉽게 이용할 수 있도록 한다. 즉, 사용자는 시소러스 질의를 통해 자신의 의도와 의미적으로 일치하는 패싯 값들을 그 퍼지 관련 정도와 함께 시소러스로부터 효과적으로 탐색할 수 있으며, 사용자가 요구할 경우 검색 시스템은 이들을 이용하여 퍼지 패싯 질의를 자동으로 구성할 수도 있다.

Abstract In reusable component retrieval systems with facet-based schemes, facet queries are generally used for representing the characteristics of components relevant to users. This paper proposes an expanded facet query equipped with an object-based thesaurus to precisely formulate user's intents. To evaluate the query, a component retrieval system is also designed and implemented. For exactly retrieving the components, user's query should include relevant facet values capable of fully specifying their characteristics. However, simply listing a series of facet values directly inputted by users, conventional queries fails to precisely represent user's intents. Our query, called expanded facet query, employs fuzzy boolean operators and object-based thesaurus; the former logically expresses the fuzzy connectives between facet queries and required components, whereas the latter helps users appropriately select the specific facet values into the query. A thesaurus query is provided to recommend the relevant facet values with their fuzzy degrees from the thesaurus as well. Furthermore, our retrieval system can automatically formulate queries with the recommended facet values, if necessary.

* 본 논문은 전북대학교 영상정보신기술연구소의 지원으로 수행되었음.

† 비 회 원 : 전북대학교 컴퓨터과학과
jhchoi@cs.chonbuk.ac.kr
†† 정 회 원 : 전북대학교 컴퓨터과학과 교수
jdyang@jiri.chonbuk.ac.kr
††† 비 회 원 : 한국전자통신연구원 개발환경연구실 연구원
dglee@sde.etri.re.kr

논문접수 : 1999년 6월 25일
심사완료 : 1999년 12월 20일

1. 서 론

소프트웨어의 복잡성이 점점 증가함에 따라 그 개발 비용과 신뢰성 문제를 해결하기 위해 이미 개발되어 검증된 소프트웨어 부품들을 재사용하기 위한 많은 연구들이 진행되고 있다[1,2]. 이들은 크게 재사용 부품들을 효과적으로 분류하고 검색하기 위한 연구들과 검색된 부품들을 실제 소프트웨어 개발에 적용하기 위한 연구들로 구분된다[3].

일반적으로 소프트웨어 부품은 다음과 같은 과정에 따라 재사용된다. 먼저, 재사용 가능한 부품들은 그 특징에 따라 적절한 분류 값들이 부여되어 체계적으로 데이타베이스에 등록된다. 이때, 원시 코드뿐만 아니라 소프트웨어의 전체 개발 단계에서 생성되는 모든 문서들이 재사용 부품으로 포함될 수 있다. 다음으로 사용자가 자신이 검색하고자 하는 부품들과 의미적으로 연관된 분류 값들로 질의를 표현하면, 질의에 기술된 모든 분류 값들을 가진 부품들이 검색된다. 마지막으로 검색된 부품들은 여러 지원 도구들을 통해 사용자에게 이해되며 이들 중 적합한 부품들만이 선택되어 소프트웨어 개발에 활용된다[1,2,3]. 이 과정에서 특히 정확한 검색은 사용자가 검색된 부품들을 이해하고 필요한 부품들을 선정하기 위해 소요되는 재사용 비용을 현저히 감소시킬 수 있게 한다[4,5,6].

정확한 검색을 위해 사용자 질의는 검색하고자 하는 부품들의 특징을 구체적으로 명시할 수 있어야 한다[7,8,9]. 현재, 많은 재사용 시스템들은 소프트웨어 부품들을 패싯 구조로 분류하고, 패싯 질의를 통해 적절한 부품들을 검색할 수 있도록 설계되어 있다[7,10,11]. 그러나, 패싯 질의는 단지 사용자에 의해 직접 입력된 패싯 값들의 나열로만 표현되기 때문에 구체적인 사용자 의도를 정확히 표현할 수 없다는 단점을 가지고 있다[4,12,13].

본 논문에서는 연구 [14]에서 설계한 객체기반 시소러스를 이용하여 사용자가 자신의 검색 요구를 정확히 표현할 수 있도록 하는 확장된 패싯 질의를 정형화하며, 이 질의를 평가할 수 있는 부품 검색 시스템을 설계하고 구현한다. 확장된 패싯 질의는 기존 패싯 질의의 단점을 보완하기 위해 퍼지 불리언 연산자와 시소러스를 이용한다. 전자는 사용자가 요구하는 부품들과 패싯 질의 사이의 퍼지 연관 정도를 논리적으로 표현할 수 있게 하며, 후자는 사용자가 구체적인 의미의 패싯 값들을 질의에 효과적으로 이용할 수 있도록 한다. 이때, 각 패싯 값에 부여된 퍼지 관련 정도는 질의와 검색된 부품들 사이의 관련 정도를 평가할 수 있게 하여 시스템이 이들을 순위화된 형태로 제시할 수 있게 한다. 사용자는 시소러스 질의를 통해 자신의 의도와 의미적으로 정확히 연관된 패싯 값들을 그 퍼지 관련 정도와 함께 효과적으로 참조할 수 있으며, 사용자가 요구할 경우 검색 시스템은 이들을 이용하여 패싯 질의를 자동으로 구성할 수도 있다. 따라서, 시소러스 질의는 정확한 의도의 퍼지 패싯 질의를 표현하기 위한 주된 기능이다.

본 논문의 구성 방식은 다음과 같다. 먼저, 2절에서는 기존의 부품 검색 시스템들과 이들이 이용하는 질의 표현 방법을 살펴보고, 3절에서는 본 논문에서 제안한 확장된 패싯 질의를 처리할 수 있는 부품 검색 시스템을 설계한다. 특히, 패싯 방법으로 분류된 부품을 저장하기 위한 데이타베이스와 패싯 값에 대한 도메인 전문가의 지식을 사용자가 활용할 수 있는 객체기반 시소러스를 정의한다. 4절에서는 사용자가 구체적인 의미의 패싯 값들을 시소러스를 통해 참조할 수 있게 하는 시소러스 질의와 참조된 패싯 값들로 구성될 수 있는 확장된 패싯 질의를 정형화하고 그 평가 방법을 설명한다. 5절에서는 확장된 패싯 질의를 이용한 부품 검색 시스템의 구현 내용을 기술하며, 마지막으로 6장에서는 결론 및 향후 연구를 제시한다.

2. 관련 연구

현재, 재사용 부품 검색 시스템들은 일반적으로 분류에 의한 검색 방법과 색인에 의한 검색 방법을 사용하고 있다[7,10,15]. 분류에 의한 방법은 크게 열거형 분류(enumerative classification)에 의한 검색과 패싯 분류(faceted classification)에 의한 검색으로 구분된다. 전자에서는 미리 정의된 분류 계층에 따라 모든 부품들을 적절히 분할하여 분류하고, 사용자는 이 계층을 순차적으로 브라우징하여 원하는 부품을 검색한다[15,16,17]. 이 방법은 사용자가 직접 부품 계층을 브라우징하기 때문에 많은 시간을 요구하며, 탐색 방향성을 상실할 위험이 있다는 단점을 가진다[7,10,13]. 후자에서는 미리 정의된 분류 패싯에 부품의 특징을 잘 반영할 수 있는 패싯 값들을 할당하여 부품들을 분류하고, 시스템은 질의 패싯에 입력된 패싯 값들로 분류된 부품들을 검색한다[7,11,12]. 이 방법은 확장이 용이하며 빠른 검색 시간을 제공하기 때문에 많은 응용 도메인에서 사용되고 있다[10]. 그러나, 패싯 질의만으로는 사용자가 검색하고자 하는 부품들의 특징을 효과적으로 표현할 수 없으며, 부품을 분류하기 위해 전문가가 사용한 구체적인 의미의 패싯 값들을 질의에 직접 이용하기가 매우 어렵다는 단점을 가진다[6,12,13].

색인에 의한 검색에서는 부품의 특성이 잘 명세된 텍스트 문서들로부터 색인어를 추출하여 역파일을 구성하고, 사용자 질의에 표현된 탐색어들을 색인어로 가진 부품들이 역파일로부터 검색된다[12,17,18]. 이때, 사용자의 검색 의도를 표현하기 위해 불리언 질의와 벡터 질의가 일반적으로 사용된다[9]. 전자는 AND나 OR와 같은 불리언 연산자를 이용하여 탐색어들 사이의 논리적 관계를 자연스럽게 표현할 수 있다는 장점을 가지는 반

면, 탐색어들이 모두 같은 관련 정도로 표현되기 때문에 사용자 의도에 따라 검색된 부품들을 순위화할 수 없다는 단점을 가진다[8,17,19]. 후자는 질의 벡터의 각 요소에 해당 관련 정도를 부여할 수 있기 때문에 이 단점을 해결할 수 있지만, 탐색어들 사이의 논리적 관계를 자연스럽게 표현할 수는 없다는 단점이 있다[9,18]. 특히, 색인에 의한 이들 방법은 텍스트 문서 검색에 적합한 방법으로 정확한 검색을 요구하는 소프트웨어 부품 검색 시스템에 이용될 때, 적절한 색인어 선정에 관한 문제, 색인어와 탐색어 사이의 의미적 불일치 문제 그리고 너무 많은 검색 결과를 생성하여 재사용 비용을 증가시키는 문제 등을 가지고 있다[9,7,18].

패싯을 이용한 시스템들은 일반적으로 Prieto-Diaz가 [11]에서 제안한 패싯들을 사용하고 있으며, 대표적인 시스템으로 RSL(Reusable Software Library)[10]과 REBOOT(REuse Based on Object Oriented Techniques)[8]가 있다. 특히, [6,10]에서는 불리언 연산자를 이용하여 자연스러운 패싯 질의를 표현할 수 있게 하였으며, 사용자는 시소러스에 표현된 패싯 값들을 참조하여 질의를 구성할 수 있도록 하고 있다. 그러나, 이 시스템에서는 패싯 값에 대한 사용자 관련 정도를 질의에 표현할 수 없기 때문에 검색된 결과를 순위화할 수 없으며, 시소러스의 구조가 너무 복잡해서 사용자가 이를 직접 참조하기가 매우 어렵다. 색인을 이용한 대표적인 시스템으로 GURU(GUide to ReUse econonmics)[17]가 있으며, LaSSIE(Larg software System Information Environment)[4]와 ROSA(Reuse of Object-oriented Specifications through Analogy)[12]는 자연어를 제한적으로 인식할 수 있는 지식을 이용하여 보다 지능적인 색인어 선정 방법을 지원하고 있다. 그러나, 이들은 자연어를 정확히 인식할 수 있는 지식 개발의 문제와 색인에 의한 검색 방법의 문제들을 개선하지 못하고 있다.

3. 확장된 패싯 질의를 이용한 부품 검색 시스템

본 논문에서 설계 및 구현된 시스템은 크게 사용자가 객체기반 시소러스를 참조하여 확장된 패싯 질의를 구성하는 모듈 그리고 질의와 의미적으로 일치하는 부품들을 데이타베이스로부터 검색하는 모듈로 구성된다. 그림 1은 이 모듈들의 처리 과정을 나타내는 시스템 구성도이다. 여기서, 확장된 패싯 질의(Q)는 각 패싯에 대한 퍼지 불리언 질의(Q_i)로 구성되며, Q_i는 해당 시소러스(T)로부터 시소러스 질의(TQ)에 의해 참조된 퍼지 패싯 값들을 이용하여 표현된다. 이때, 질의의 각 패싯 값에 부여된 퍼지 정도는 하나의 패싯 값에 대한 사용자 만족 정도를 나타내며 시소러스 질의와 참조된 패싯 값들 사이의 의미적 관련 정도로 계산된다. 검색 시스템은 이 확장된 패싯 질의를 평가하여 사용자 의도와 의미적으로 일치하는 부품들을 데이타베이스(CDB)로부터 순위화된 형태로 검색한다. 따라서, 본 논문에서 설계된 부품 검색 시스템은 다음과 같이 정의할 수 있다.

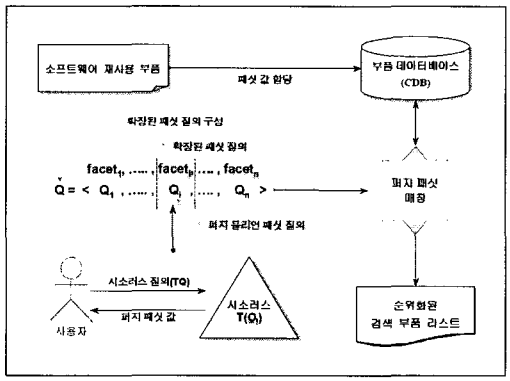

그림 1 확장된 패싯 질의를 이용한 부품 검색 시스템

[정의 1] 부품 검색 시스템 CIR은 다음과 같이 4 투플로 정의된다.

$$CIR = \langle CDB, T, TQ, Q \rangle$$

여기서, CDB는 패싯 방법으로 분류된 부품들을 저장하기 위한 데이타베이스이며, T는 패싯들에서 사용되는 객체기반 시소러스들의 집합이다. TQ는 T로부터 사용자 의도와 의미적으로 일치하는 퍼지 패싯 값들을 참조하기 위한 시소러스 질의들의 집합이며, Q는 CDB로부터 사용자가 의도한 부품들을 검색하기 위한 확장된 패싯 질의들의 집합이다.

3.1 부품 데이타베이스

패싯 방법으로 데이타베이스에 분류된 부품들은 식별자, 이름, 기술자 그리고 실제 재사용되는 부품 내용으로 구성된다. 부품 식별자는 데이타베이스로부터 특정 부품을 유일하게 구별할 수 있게 하며, 모든 부품은 그 이름으로 사용자들에게 대표된다. 부품 기술자는 각 특징에 따라 부품의 분류 패싯에 부여된 패싯 값들의 집합들로 구성되며, 이들은 질의 패싯 값들과 함께 검색에 이용된다. 부품 내용은 소프트웨어 개발 단계에서 생성되는 재사용 가능한 모든 문서들이다. 즉, 요구 분석문, 설계 문서, 원시 코드 그리고 테스트 쉬트 등이 부품 내

용으로 포함될 수 있다. 따라서, 부품 데이타베이스는 다음과 같이 정의될 수 있다.

[정의 2] 부품 데이타베이스 *CDB*는 다음과 같이 4 투플로 정의된다.

$$CDB = \langle C, NAME, CD, COMP \rangle$$

여기서, *C*는 시스템에 의해 생성되는 부품 식별자(surrogate)들의 집합, *NAME*은 부품 이름들의 집합, *CD*는 부품 기술자들의 집합 그리고 *COMP*는 부품 식별자로 참조되는 부품 내용들의 집합이다.

기존의 많은 패싯기반 부품 데이타베이스에서는 부품 기술자를 구성하는 패싯들이 단지 하나의 패싯 값만을 가질 수 있도록 설계되기 때문에 다양한 특징의 부품들을 여러 관점에 따라 명확히 분류하기 어렵다는 단점을 가지고 있다. 본 논문에서는 이 단점을 보완하기 위해 [20]와 같이 부품 기술자의 모든 패싯들이 여러 패싯 값들을 가질 수 있도록 설계하였다. 부품 기술자는 다음과 같이 정의된다.

[정의 3] 임의의 부품 $c \in C$의 부품 기술자 CD_c는 다음과 같이 정의된다.

$$CD_c = \{\langle facet, value_set \rangle \mid facet \in FACET,\ value_set \in 2^V\}$$

여기서, *FACET*은 모든 패싯들의 집합이며, *V*는 *facet*에서 사용될 수 있는 모든 패싯 값들의 집합이다. *value_set*은 부품 *c*의 패싯 *facet*에 부여된 패싯 값들의 집합이며, *c.facet*으로 참조될 수 있다.

C	NAME	CD			COMP	
		FUNCTION	SYSTEM TYPE	...	REQ	...
...	...	...	...	...	.....	...
100	AXE10 Signal Controller	Signal Control, Signal Send, Signal Receive	Circuit Switching, Public Exchange, AXE10	...	AXE10_spec.req	...
...	...	...	.....	...	.....	...

그림 2 부품 데이타베이스 예

[예제 1] 그림 2는 패싯기반 부품 데이타베이스의 한 예이다. 여기서, 두 패싯 'FUNCTION', 'SYSTEM TYPE'$\in$*FACET*이며, 부품 이름이 *c.NAME*='AXE10 Signal Controller'인 $c \in C$에 대해 *facet*='SYSTEM TYPE'일 때, *c.facet*={Circuit Switching, Public Exchange, AXE10}이다. 또한, *c*의 부품 내용들 중 요구 분석문은 "AXE10_spec.req"$\in$*c.REQ*와 같이 참조될 수 있다.

3.2 객체기반 시소러스

시소러스는 패싯 값에 대한 도메인 전문가의 지식을 사용자가 활용할 수 있도록 함으로써, 분류를 위해 사용된 패싯 값들과 질의에서 사용될 패싯 값들 사이의 의미적 일관성을 유지시킬 수 있게 한다. 즉, 전문가가 부품들을 분류하기 위해 사용한 구체적인 의미의 패싯 값들을 사용자가 질의 표현에 적절히 이용할 수 있게 한다. 또한, 유사한 의미를 가진 패싯 값들 사이의 퍼지 관련 정도를 추론할 수 있게 하여 질의와 부품들 사이의 의미적 관련 정도를 평가할 수 있도록 지원한다.

본 논문에서는 하나의 패싯에서 사용될 수 있는 패싯 값들의 의미적 관계를 객체지향 방법으로 표현한 객체기반 시소러스[14]를 이용한다. 객체기반 시소러스는 패싯 값들 사이의 의미 관계를 사용자의 직관과 일치할 수 있도록 표현할 수 있으며, 사용자가 의도한 특정한 의미의 패싯 값들을 정확히 추출할 수 있도록 지원하기 때문에 정확한 검색을 요구하는 재사용 시스템에 적합하게 이용될 수 있다. 객체기반 시소러스에서 패싯 값들은 모두 객체로 간주되며, 구체적인 의미의 객체는 일반적인 의미의 객체에 대한 인스턴스 객체가 되거나 하위 개념 객체가 된다. 따라서, 하나의 상위 개념 객체는 수직적으로 하위 객체들과 일반화 또는 클래스화 관계를 가지게 되어 객체 계층을 형성하며, 모든 객체들은 수평적으로 다른 객체 계층의 객체들과 집성화 또는 연관화 관계를 가질 수 있다(그림 3 참조).

[정의 4] 부품 *c*의 *i* 번째 패싯 $facet_i \in FACET$에 대응되는 객체 기반 시소러스 $T(O_i)$는 다음과 같이 투플들의 집합으로 정의된다.

$$T(O_i) = \{t = \langle r, o_1, o_2, w_{12} \rangle \mid t \in R \times O_i \times O_i \times [0,1]\}$$

여기서, R={*sub-concept-of, instance-of, part-of, association-of*}이고 $O_i \subseteq V$이다. O_i가 명시될 필요가 없을 때, $T(O_i)$는 T로 간단히 표시한다.

[예제 2] 그림 3은 그림 2의 2번째 패싯 '*SYSTEM TYPE*'$\in$*FACET*에 대응되는 객체기반 시소러스 $T(O_2)$의 예이며, [정의 4]에서 정의된 투플들의 집합으로 구성된다. 예를 들어, <sub-concept-of, Switching System, Circuit Switching, 0.92>, <part-of, Switching System,

Switch Network, 0.85> 그리고 <instance-of, Circuit Switching, AXE10, 0.93>$\in T(O_2)$이다.

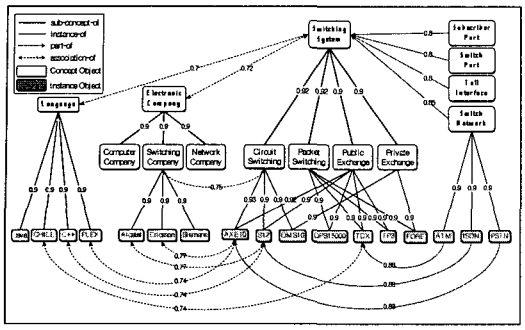

그림 3 객체기반 시소러스의 예

[정의 5] R_1={*sub-concept-of, instance-of*}이고 R_2={*part-of, association-of*}일 때, 모든 t=<r, o_1, o_2, w_{12}>$\in T$에 대해 $r\in R_1$이면 $t\in T_1$이고, $r\in R_2$이면 $t\in T_2$이다. 따라서, $R=R_1\cup R_2$이고 $T=T_1\cup T_2$이다.

[정의 6] $T_1\subseteq T$이고 $r\in R_1$에 대해 t_1=<*sub-concept-of*, o_i, o_j, w_{ij}>, t_2=<r, o_j, o_k, w_{jk}>, t_3=<r, o_i, o_k, w_{ik}>라 하자. T_1의 전이 폐쇄(transitive closure) T_1^+은 다음과 같이 정의된다.

1) $t\in T_1$이면, $t\in T_1^+$이다.
2) $t_1\in T_1^+$이고 $t_2\in T_1$이면, $t_3\in T_1^+$이다.

여기서, $w_{ik}=\max_j[\min(w_{ij}, w_{jk})]$ 이다.

[예제 3] 그림 3에서 <sub-concept-of, Switching System, Circuit Switching, 0.92>이고 <instance-of, Circuit Switching, AXE10, 0.93>이며, <sub-concept-of, Switching System, Public Exchange, 0.9>이고 <instance-of, Public Exchange, AXE10, 0.92>임으로 <instance-of, Switching System, AXE10, w>$\in T_1^+$이다. 여기서, w=*max*(*min*(0.92,0.93), *min*(0.9,0.92)).

다음은 객체기반 시소러스를 구성하는 객체 계층을 정의하고 객체들 사이의 상계 집합과 하계 집합을 정의한다. 상계 집합과 하계 집합은 4.1절에서 시소러스 질의를 평가하기 위해 사용된다.

[정의 7] $T_1(O)$에서 $o\in O$를 최상위 객체로 하는 객체 계층 $T_1(O_o)\subseteq T_1(O)$은 다음과 같이 정의된다.

$$T_1(O_o)=\{t=\langle r, o_1, o_2, w_{12}\rangle \mid t\in R_1\times O_o\times O_o\times[0,1]\}$$

여기서, $r\in R_1$에 대해 $O_o=\{o'\in O \mid o'=o\vee\langle r, o, o', w\rangle\in T_1^+\}$이다. 따라서, T_1의 모든 객체 O={o_1, o_2, ..., o_n}에 대해 $T_1(O)=T_1(O_{o_1})\cup T_1(O_{o_2})\cup\ldots\cup T_1(O_{o_n})$이다.

[정의 8] 객체 계층 $T_1(O_o)\subseteq T_1$에 대해 임의의 객체 집합 $O'\in 2^{O_o}$의 상계 집합(UBS: Upper Bound Set)과 하계 집합(LBS: Lower Bound Set)은 각각 다음과 같이 정의된다.

$$UBS(O')=\bigcup_{\forall o'\in O'} sup(o'),\quad UBS(O')=\bigcup_{\forall o'\in O'} sub(o')\ [21].$$

여기서, $r\in R_1$과 $o'\in O'$에 대해 각각 $sup(o')=\{o\in O_o \mid o=o'\vee\langle r, o, o', w\rangle\in T_1^+\}$이고, $sub(o')=\{o\in O_o \mid o=o'\vee\langle r, o', o, w\rangle\in T_1^+\}$이다.

하나의 최상위 객체로부터 파생되는 객체 계층은 유일한 최소 상계(LUB: Least Upper Bound)와 최대 하계(GLB: Greatest Lower Bound)를 항상 가질 수는 없기 때문에 격자(lattice) 구조로는 표현될 수 없다. 따라서, 다음과 같이 최소 상계 집합(LUBS: Least Upper Bound Set)과 최대 하계 집합(GLBS: Greatest Lower Bound Set)을 정의할 필요가 있다.

[정의 9] 객체 계층 $T_1(O_o)$과 임의의 집합 $O'\subseteq O_o$의 모든 객체 $o'_i\in O'$, i=1,...,n에 대한 LUBS와 GLBS는 다음과 같이 정의되며, 각각 $\vee_{i=1}^{n} o'_i$와 $\wedge_{i=1}^{n} o'_i$로 표시한다.

$$LUBS(O')=\vee_{i=1}^{n} o'_i$$
$$=UBS(O')-\{o \mid o, o'\in UBS(O'), \langle r, o, o', w\rangle\in T_1(O_o)\}$$
$$GLBS(O')=\wedge_{i=1}^{n} o'_i$$
$$=LBS(O')-\{o \mid o, o'\in LBS(O'), \langle r, o', o, w\rangle\in T_1(O_o)\}$$

여기서, $r\in R_1$이며, $LUBS(O')\subseteq UBS(O')$이고 $GLBS(O')\subseteq LBS(O')$이다.

[예제 4] 그림 3에서 O'={Circuit Switching, Public Exchange}에 대해 *sup*(Circuit Switching)={Circuit Switching, Switching System}이고 *sup*(Public Exchange)={Public Exchange, Switching System}이다. 따라서, $UBS(O')$=*sup*(Circuit Switching) ∩

sup(Public Exchange)로부터 $LUBS(O')$= {Switching System}를 구할 수 있다. 또한, *sub* (Circuit Switching)={Circuit Switching, AXE10, S12, DMS10}이고 *sub*(Public Exchange)={Public Exchange, AXE10, S12, DPS1500, TDX, TP3}이다. 따라서, $LBS(O')$=*sub*(Circuit Switching) ∩ *sub*(Public Exchange)로부터 $GLBS(O')$={AXE10, S12}를 구할 수 있다.

4. 확장된 패싯 질의의 구성 및 평가

소프트웨어 재사용을 위한 부품 검색 시스템은 매우 정확한 검색을 수행해야 하기 때문에 사용자의 의도를 구체적으로 표현할 수 있는 사용자 질의가 필수적으로 요구된다. 이 절에서는 이를 위한 확장된 패싯 질의에 대해 기술한다. 이 질의는 시소러스로부터 참조되는 구체적인 의미의 퍼지 패싯 값들과 불리언 연산자로 표현된다. 이때, 시소러스 질의는 시소러스로부터 특정한 의미의 퍼지 패싯 값들을 직접 참조하기 위해 사용되며, 사용자는 이들을 이용하여 퍼지 패싯 질의를 자동으로 또는 선택적으로 구성할 수 있다. 4.1 절에서는 시소러스로부터 사용자가 의도한 패싯 값과 그 퍼지 관련 정도를 참조할 수 있는 시소러스 질의를 정의하고 평가하며, 4.2절에서는 참조된 패싯 값들로 구성된 확장된 패싯 질의를 정의하고 평가한다.

4.1 확장된 패싯 질의 구성을 위한 시소러스 참조

시소러스 질의는 시소러스를 브라우징할 때, 사용자의 참조 범위를 지정해줌으로써 사용자가 원하는 퍼지 패싯 값들을 쉽게 참조할 수 있게 하며, TQ=[*MAIN_QUERY*, *RESTRICTION_QUERY*]와 같이 두 개의 부분 질의로 표현된다. 여기서, *MAIN_ QUERY*는 객체기반 시소러스의 여러 객체 계층들 중 하나를 지정함으로써 사용자의 참조 범위를 한정하며, *RESTRICTION_QUERY*는 *MAIN_QUERY*의 참조 범위를 동시(AND) 또는 선택적(OR)으로 제한하기 위해 사용된다. 예를 들어, 그림 3에서 시소러스 질의 *TQ*=[Switching System, PSTN AND Ericsson]은 사용자가 교환기 시스템(Switching System)을 최상위 객체로 하는 객체 계층 내의 모든 객체들 중에서 교환 네트워크 "PSTN" 그리고 교환기 회사 "Ericsson"과 동시에 $r \in R_2$인 관계에 있는 객체들로부터 의미적으로 일정 정도 관련을 가진 모든 객체들의 집합을 참조하고자 할 때 사용된다. 또한, 참조된 객체에 대한 사용자의 퍼지 만족 정도는 시소러스를 이용하여 시스템이 자동으로 계산한다. 사용자는 이들 중 자신이 의도한 객체만을 선택하여 퍼지 불리언 패싯 질의를 구성할 수 있으며, 시스템에게 자동으로 질의를 구성하도록 할 수도 있다.

[정의 10] 시소러스 질의 *TQ*=[*MAIN_QUERY*, *RESTRICTION_QUERY*]의 *MAIN_QUERY*와 *RESTRICTION_QUERY*는 다음과 같이 정의된다.

1) $MAIN_QUERY = \vee_{i=1}^{n} o_i$ 또는 $\wedge_{i=1}^{n} o_i$
2) $RESTRICTION_QUERY = OR_{i=1}^{m}(AND_{j=1}^{n_i} o_{ij})$

여기서, $T_1(O_o) \subseteq T(O)$에 대해 *MAIN_QUERY*에 표현된 모든 객체들은 $o_i \in O_o$이고 *RESTRICTION_ QUERY*에 표현된 모든 객체들은 $o_{ij} \in O - O_o$이다.

[정의 11] $T_1(O_o)$이고 임의의 집합 $O' \subseteq O_o$의 모든 객체 $o_i \in O'$, i=1,...,n로 *MAIN_QUERY*가 표현될 때, $O(MAIN_QUERY)$는 다음과 같이 정의된다.

1) $MAIN_QUERY = \wedge_{i=1}^{n} o_i$일 때,
O(MAIN_QUERY)=GLBS(O).
2) $MAIN_QUERY = \vee_{i=1}^{n} o_i$일 때,
O(MAIN_QUERY)=LUBS(O).

[정의 12] $T_2(O)$이고 임의의 집합 $O' \subseteq O$의 모든 객체 $o'_{ij} \in O'$, i=1,...,m, j=1,...,n_i에 대해 $RESTRICTION_QUERY = OR_{i=1}^{m}(AND_{j=1}^{n_i} o'_{ij})$일 때, $O(RESTRICTION_QUERY)$는 다음과 같이 정의된다.

$$O(RESTRICTION_QUERY) = \bigcup_{i=1}^{m} \left(\bigcap_{j=1}^{n_i} REST(o'_{ij}) \right)$$

여기서, $o' \in O'$에 대해 $REST(o') = \{o \mid \langle r, o, o', w \rangle \in T_2(O)\}$이다.

$O(TQ)$는 시소러스로부터 TQ에 의해 추출되는 객체들의 집합이며, 이 집합의 모든 객체는 시소러스 질의 TQ와 동일한 의미를 가진다. $O^+(TQ)$는 $O(TQ)$의 객체들과 일정한 의미 관계를 가지는 시소러스 내의 참조 객체 집합이며, 이 집합은 참조 그래프 $G(O^+(TQ))$로 시각화된다. 다음은 이들에 대한 정의이다.

[정의 13] $T_1(O_o) \subseteq T_1^+$에 대해 TQ가 추출한 객체들의 집합이 $O(TQ) \subseteq O_o$일 때, 참조 객체 집합 $O^+(TQ)$는 다음

과 같이 정의된다.

$$O^+(TQ)=\{\ o \mid o'\in O(TQ)\text{에 대해},\ o=o' \vee \langle r,o,o',w\rangle \in T_1^+ \vee \langle r,o',o,w\rangle \in T_1^+\}$$

여기서, $r\in R_1$이고 $G(O^+(TQ))=\{t=\langle r,o_1,o_2,w_{12}\rangle \mid o_1, o_2\in O^+(TQ), t\in T(O_o)\}$은 $O^+(TQ)$에 대한 참조 그래프를 형성한다.

[정의 14] TQ=[*MAIN_QUERY*, *RESTRICTION_QUERY*]일 때, TQ가 추출하는 객체 집합 $O(TQ)$는 다음과 같이 정의된다.

$$O(TQ)=O^+([MAIN_QUERY,*])\cap O(RESTRICTION_QUERY)$$

여기서, $O([MAIN_QUERY,*])=O(MAIN_QUERY)$이다.

[보조 정리 1] $T_1(O_o)$과 $T_2(O)$에 대해 $o\in O-O_o$일 때, $TQ=[MAIN_QUERY, o]$이고 $TQ'=[MAIN_QUERY,*]$이면, $O(TQ)=O^+(TQ')\cap REST(o)$이다.

[증명] TQ'의 참조 객체 집합은 $O^+(TQ')$이며 $RESTRICTION_QUERY=o$에 대해 $O(RESTRICTION_QUERY)=REST(o)$이다. 따라서, [정의 14]에 의해 TQ가 추출하는 객체 집합은 $O(TQ)=O^+(TQ')\cap REST(o)$이다.

[보조 정리 2] $TQ=[MAIN_QUERY, AND_{j=1}^{n} o_j]$이고 $TQ_j=[MAIN_QUERY, o_j]$, $j=1,..,n$에 대해 $O(TQ)=O(TQ_1)\cap O(TQ_2)\cap ...\cap O(TQ_n)$이다.

[증명] $TQ'=[MAIN_QUERY,*]$라고 할 때, TQ'의 참조 객체 집합은 $O^+(TQ')$이며 [정의 12]에 의해 $RESTRICTION_QUERY=AND_{j=1}^{n} o_j$의 $O(RESTRICTION_QUERY)$는 $REST(o_1)\cap REST(o_2)\cap ...\cap REST(o_n)$이다. 따라서, [정의 14]에 의해 TQ가 추출하는 객체 집합은 $O(TQ)=O^+(TQ')\cap(REST(o_1)\cap ...\cap REST(o_n))$이다. 그런데, [보조 정리 1]에 의해 $O(TQ_j)=O^+(TQ')\cap REST(o_j)$, $j=1,...,n$이므로 $O(TQ)=O(TQ_1)\cap O(TQ_2)\cap ...\cap O(TQ_n)$이다.

[보조 정리 3] $TQ=[MAIN_QUERY, OR_{i=1}^{m} o_i]$이고, $TQ_i=[MAIN_QUERY, o_i]$, $i=1,..,m$에 대해 $O(TQ)=O(TQ_1)\cup O(TQ_2)\cup ...\cup O(TQ_m)$이다.

[증명] [보조 정리 2]와 유사함으로 이 증명은 생략한다.

[정리 1] $TQ=[MAIN_QUERY, OR_{i=1}^{m}(AND_{j=1}^{n_i} o_{ij})]$이고, $TQ_{ij}=[MAIN_QUERY, o_{ij}]$, $i=1,..,m$, $j=1,..,n_i$에 대해 $O(TQ_{ij})$일 때, $O(TQ)=(O(TQ_{11})\cap ...\cap O(TQ_{1n_1}))\cup ...\cup(O(TQ_{m1})\cap ...\cap O(TQ_{mn_m}))$이다.

[증명] 이 정리는 [보조 정리 2]와 [보조 정리 3]을 이용하여 증명될 수 있다.

[예제 4] TQ=[Circuit Switching ∧ Public Exchange, (C++ AND Ericsson AND PSTN) OR (PLEX AND Ericsson AND PSTN)]은 TQ'=[Circuit Switching ∧ Public Exchange, *]에 대해 $O(TQ')$={AXE10, S12}임으로 $O^+(TQ')$={Switching System, Circuit Switching, Public Exchange, AXE10, S12}이다. 여기서, 'AXE10'$\in O^+(TQ')\cap(REST$('PLEX')$\cap REST$('Ericsson')$\cap REST$('PSTN'))임으로 $O(TQ)$={AXE10}이고, $O^+(TQ)$={AXE10, Circuit Switching, Public Exchange, Switching System}이다. 다음 그림 4는 참조 그래프 $G(O^+(TQ))$를 보이고 있다.

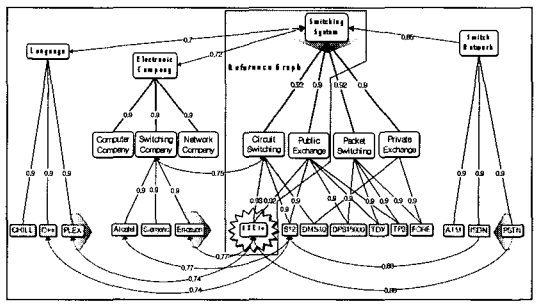

그림 4 시소러스 질의 처리 예

$O(TQ)$가 TQ와 의미적으로 정확히 일치하는 객체들의 집합인 반면, $O^+(TQ)$는 TQ와 의미적으로 유사한 객체들의 집합이기 때문에 이 집합은 모든 객체들이 사용자 의도와 일정한 정도의 의미적 거리를 가지는 퍼지 집합으로 나타낼 수 있다. 이때, 의미적 거리는 객체 $o\in O^+(TQ)$에 대한 사용자의 퍼지 만족 정도를 나타내는 퍼지 소속성 정도로 표현되며, 이 소속성 정도는 시소러스 내의 객체 관계 정보를 이용하여 집합 $O(TQ)$와 o의 관련 정도로 평가될 수 있다. 다음은 이를 위한 퍼지 소속성 함수 $\mu_{O^+(TQ)}(o)$에 대한 정의이다.

[정의 15] TQ에 의해 참조되는 객체 $o\in O^+(TQ)$에 대한 퍼지 소속성 함수 $\mu_{O^+(TQ)}(o)=\alpha$, $0<\alpha\leq 1$은 $o\in O(TQ)$이면 α=1.0이고, $o'\in O(TQ)$에 대해 $\langle r, o, o', w\rangle$ 또는

$<r, o', o, w'> \in T_1^+$이면 $\alpha = w'$이다. 여기서, $<r, o, o'', w''>$ 또는 $<r, o'', o, w''> \in T_1^+$인 모든 $o'' \in O(TQ)$에 대해 $w' \geq w''$이다.

$O^+(TQ)$는 시소러스 질의 TQ와 의미적으로 유사한 객체들의 집합이며, $\mu_{O^+(TQ)}(o)$는 $o \in O^+(TQ)$에 대한 사용자의 만족 정도를 자동 평가하는 퍼지 소속성 함수이다. 따라서, 참조 객체들 중 적절한 객체가 패싯 질의의 값으로 선택되면, 이 함수는 선택된 객체에 대한 퍼지 만족 정도를 시스템이 자동으로 계산할 수 있게 한다. 따라서, 사용자가 퍼지 패싯 질의를 표현하기 위해 모든 패싯 값에 대한 만족 정도를 직접 부여할 필요가 없으며, 단지 보다 섬세한 질의 표현을 원하는 경우에만 자동으로 계산된 관련 정도를 적절히 수정하여 명시할 수 있다.

4.2 확장된 패싯 질의의 평가

본 논문에서 제안한 확장된 패싯 질의는 모든 패싯 질의를 퍼지 패싯 값들과 불리언 연산자로 표현한다. 예를 들어, 그림 2에서 2번째 패싯 'SYSTEM TYPE'에 대해 퍼지 불리언 질의는 "Circuit Switching/0.87 ⊔ Packet Switching/0.85"와 같이 표현될 수 있다. 이 질의는 시소러스 $T(O_2)$에 시소러스 질의 TQ를 수행하여 얻은 패싯 값 $v \in O^+(TQ)$과 각 패싯 값에 대해 자동으로 평가되는 사용자 퍼지 만족 정도 $0 < \alpha = \mu_{O^+(TQ)}(v) \leq 1$를 할당함으로써 구성된다. 이 절에서는 확장된 패싯 질의를 정의하고 그 평가 방법에 대해 기술한다. 이 평가 방법은 [22]를 재사용 시스템에 적합한 형태로 수정하여 사용하였다.

[정의 16] TQ에 의해 참조되는 패싯 값이 $v \in O^+(TQ)$이고 그 패싯 값에 대한 사용자의 퍼지 만족 정도가 $\alpha = \mu_{O^+(TQ)}(v)$, $0 < \alpha \leq 1$일 때, 단일 패싯 질의(mono facet qeury)는 $Qm = v/\alpha$으로 정의된다.

[정의 17] Qm_j, $j=1,..,s$를 단일 패싯 질의들이라고 할 때, 이접 패싯 질의(conjunctive facet query)는 단일 패싯 질의의 퍼지 논리합 $Q = \sqcup_{j=1}^{s} Qm_j$으로 정의된다.

[정의 18] Q_i, $i=1,..,n$를 이접 패싯 질의들이라 할 때, 연접 패싯 질의(disjunctive facet query)는 이접 패싯 질의의 퍼지 논리곱 $Q = \sqcap_{i=1}^{n} Q_i$으로 정의된다.

확장된 패싯 질의를 구성하는 모든 퍼지 불리언 패싯 질의의 일반적인 형태는 [정의 18]과 같이 이접 질의 요소들에 대한 연접 질의로 표현된다. 따라서, 확장된 패싯 질의는 다음과 같이 각 패싯 질의에 대한 연접 질의의 의미로 표현될 수 있다.

[정의 19] $facet_k \in FACET$에 대한 퍼지 불리언 패싯 질의가 Q_k일 때, 확장된 패싯 질의는 $Q = \sqcap_{k=1}^{u} Q_k$와 같이 정의된다. 여기서, $1 \leq u \leq |FACET|$이다.

다음은 확장된 패싯 질의와 의미적으로 연관된 부품들을 검색하기 위한 질의 평가 방법을 설명한다. 이를 위해 먼저 다음과 같은 두 정의가 필요하다.

[정의 20] $CDB \vdash_{\alpha} Q(c)$는 $c \in C$가 $\alpha \in [0,1]$ 정도로 Q를 만족시킴을 나타내며, $CDB \nvdash_{\alpha} Q(c)$는 c가 α 미만으로 Q를 만족시킴을 나타낸다.

[정의 21] $\|Q\|$는 Q를 만족하는 부품 $c \in C$에 대한 퍼지 집합으로 다음과 같이 정의된다.

$$\|Q\| = \{(c, \alpha) \mid CDB \vdash_{\alpha} Q(c),\ \text{모든 } CDB \vdash_{\alpha'} Q(c) \text{에 대해 } \alpha \geq \alpha'\}$$

하나의 퍼지 불리언 패싯 질의에 대한 평가 방법을 설명하기 위해 먼저, 단일 패싯 질의에 의해 평가되는 부품들의 퍼지 집합을 다음과 같이 정의한다.

[정의 22] $facet \in FACET$에 대해 $Qm = v/\alpha$일 때, $c \in C$가 $v \in c.facet$이면, $CDB \vdash_{\alpha} Qm(c)$로 정의한다. 즉, $\|Qm\| = \{(c, \alpha) \mid CDB \vdash_{\alpha} Qm(c)\}$이다.

다음으로 [보조 정리 4]에서는 단일 패싯 질의의 퍼지 논리합으로 표현된 이접 패싯 질의를 평가하고, 이를 이용하여 [보조 정리 5]에서는 하나의 퍼지 불리언 패싯 질의를 평가한다. 이 보조 정리을 이용하여 검색 시스템은 [정리 2]와 같이 확장된 패싯 질의를 평가한다.

[보조 정리 4] $Q = \sqcup_{j=1}^{s} Qm_j$일 때, $c \in C$가 $(c, \alpha_1) \in \|Qm_1\| \vee (c, \alpha_2) \in \|Qm_2\| \vee ... \vee (c, \alpha_s) \in \|Qm_s\|$이면 $(c, \alpha) \in \|Q\|$이다. 여기서, $\alpha = max(\alpha_1, \alpha_2, ..., \alpha_s)$.

[증명] $c \in C$가 $(c, \alpha_1) \in \|Qm_1\| \vee (c, \alpha_2) \in \|Qm_2\| \vee ... \vee (c, \alpha_s) \in \|Qm_s\|$라면, $CDB \vdash_{\alpha_1} Qm_1(c) \vee CDB \vdash_{\alpha_2} Qm_2(c) \vee ... \vee CDB \vdash_{\alpha_s} Qm_s(c)$이다. 또한, [정의 21]에 의해 α_i, $i=1,..,s$는 다음을 만족한다. 즉, $CDB \vdash_{\alpha_i'} Qm_i(c)$이고 $\alpha_i \geq \alpha_i'$, $i=1,..,s$. 다음으로, 모든 $CDB \vdash_{\alpha_i} Qm_i(c)$, $i=1,..,s$들 중 단지 한 개

만이 $\alpha_k>0$, $1\le k\le s$를 만족한다고 가정하자. 그러면, 모든 $1\le i\neq k\le s$에 대해 $\alpha_i=0$임으로 $\alpha_k=max(\alpha_1,\alpha_2,...,\alpha_k,...,\alpha_s)$이고 $(c,\alpha_k)\in\|Q\|$이다.

다른 가능한 경우로, $CDB\vdash_{a_i}Qm_i(c)$, $i=1,2,..,s$들 중 k개가 $\alpha_{i_k}>0$, $1\le i_1,i_2,...,i_k\le s$를 만족한다고 가정하자. 그러면, $1\le i\neq i_1,i_2,...,i_k\le s$에 대해 $\alpha_i=0$이므로, 이 경우는 다음과 같이 표현할 수 있다.

$$CDB\vdash_{a_{i_1}}Qm_{i_1}(c)\wedge CDB\vdash_{a_{i_2}}Qm_{i_2}(c)\wedge...\wedge CDB\vdash_{a_{i_k}}Qm_{i_k}(c) \quad (1)$$

여기서, 만약 $1\le i\le s$에 대해 모두 $\alpha_i>0$이면, (1)은 $CDB\vdash_{a_1}Qm_1(c)\wedge CDB\vdash_{a_2}Qm_2(c)\wedge...\wedge CDB\vdash_{a_s}Qm_s(c)$이 됨을 알 수 있다.

그런데, $Q=\sqcup_{j=1}^{s}Qm_j$임으로 (1)은 $CDB\vdash_{a_{i_1}}Q(c)\wedge CDB\vdash_{a_{i_2}}Q(c)\wedge...\wedge CDB\vdash_{a_{i_k}}Q(c)$이며, [정의 21]에 의해 $(c,\alpha)\in\|Q\|$, $\alpha=max(a_{i_1},a_{i_2},...,a_{i_k})$이다. 따라서, 모든 경우가 이 두 경우에 속하고 각 경우 모두 성립함으로 이 정리는 증명된다.

[보조 정리 5] $Q=\sqcap_{i=1}^{n}Q_i$일 때, $c\in C$가 $(c,\alpha_1)\in\|Q_1\|\wedge(c,\alpha_2)\in\|Q_2\|\wedge...\wedge(c,\alpha_n)\in\|Q_n\|$이면 $(c,\alpha)\in\|Q\|$이다. 여기서, $\alpha=min(\alpha_1,\alpha_2,...,\alpha_n)$.

[증명] $Q=\sqcap_{i=1}^{n}Q_i$에 대해, $(c,\alpha_1)\in\|Q_1\|\wedge...\wedge(c,\alpha_n)\in\|Q_n\|$임으로 $CDB\vdash_{a_1}Q_1(c)\wedge...\wedge CDB\vdash_{a_n}Q_n(c)$이다. 여기서, $CDB\vdash_{a_i'}Q_i(c)$이고 $a_i\ge a_i'$, $i=1,...,n$이 성립함으로 $\alpha=min(\alpha_1,\alpha_2,...,\alpha_n)$로 두면, $CDB\vdash_{\alpha}(Q_1(c)\wedge...\wedge Q_n(c))$이다. 이제, $CDB\vdash_{\alpha}Q(c)$이고 $(c,\alpha)\in\|Q\|$임을 증명하기 위해 $\alpha'>\alpha$이면, $CDB\vdash_{\alpha'}Q(c)$이 성립하지 않음을 보이자. 만약 $\alpha'>\alpha$이면, $\alpha=min(\alpha_1,\alpha_2,...,\alpha_n)$이기 때문에 $\alpha_k<\alpha'$, $1\le k\le n$인 $CDB\vdash_{a_k}Q_k(c)$가 반드시 하나 이상 존재해야한다. 그러나, [정의 21]에 의해 이것 역시 $CDB\nvdash_{\alpha'}Q_k(c)$임으로 불가능하다. 즉, $\alpha'>\alpha$이면, $CDB\nvdash_{\alpha'}Q(c)$이고 $(c,\alpha')\notin\|Q\|$이다. 따라서, 모든 α'에 대해 $CDB\vdash_{\alpha}Q(c)$, $\alpha\ge\alpha'$이므로 $(c,\alpha)\in\|Q\|$이다.

[정리 2] $facet_k\in FACET$이고 퍼지 불리언 패싯 질의가 Q_k일 때, 확장된 패싯 질의 $Q=\sqcap_{k=1}^{u}Q_k$에 대해 $c\in C$가 $(c,\alpha_1)\in\|Q_1\|\wedge(c,\alpha_2)\in\|Q_2\|\wedge...\wedge(c,\alpha_u)\in\|Q_u\|$이면 $(c,\alpha)\in\|Q\|$이다. 여기서, $\alpha=min(\alpha_1,\alpha_2,...,\alpha_u)$.

[증명] 이 정리는 [보조 정리 5]와 같은 방법으로 증명될 수 있다.

[예제 5] 그림 2의 데이타베이스에서 확장된 패싯 질의 Q가 1번째와 2번째 패싯 질의에 대해 Q_1="Signal Send/0.87 ⊓ Signal Control/0.85"과 Q_2="(Circuit Switching/0.93 ⊔ Public Exchange/0.92) ⊓ AXE10/1.0"이며 다른 패싯에 대해서는 모두 무조건 질의라면, 부품 이름이 $c.NAME$='AXE10 Signal Controller'인 $c\in C$는 $(c,\alpha_1)\in\|Q_1\|$, $\alpha_1=min(0.87,0.85)=0.85$이고, $(c,\alpha_2)\in\|Q_2\|$, $\alpha_2=min(1.0,max(0.93,0.92))=0.93$이다. 따라서, $(c,\alpha)\in\|Q\|$, $\alpha=min(0.85,0.93)$이다.

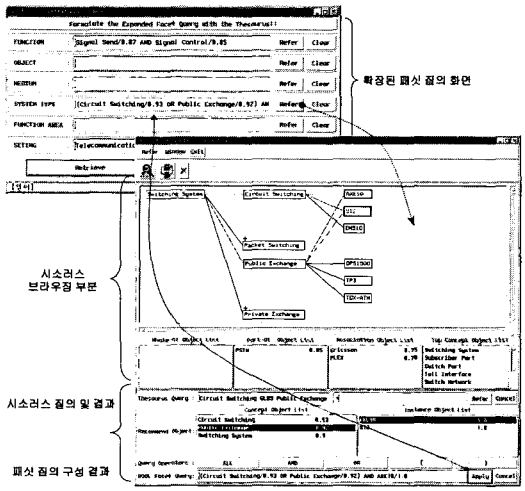

그림 5 시소러스 질의를 통한 선택적 패싯 질의의 구성

5. 구 현

이 절은 본 논문에서 구현된 시스템에서 시소러스 질의에 의한 확장된 패싯 질의의 구성 과정과 확장된 패싯 질의를 이용한 부품 검색 과정을 설명한다. 이 시스템은 Solaris 2.5에서 C++와 X/Motif를 통해 구현되었다. 테스트 재사용 부품들과 객체기반 시소러스는 객체지향 데이타베이스 UniSQL에 저장되며, 이를 접근하기 위해 표준 ESQL를 사용하였다. 또한, 객체기반 시소러스에 대한 시소러스 질의 처리기는 [14]을 재사용 시스템에 적합한 형태로 변형하여 구현하였다.

그림 5는 시스템이 시소러스 질의 [Circuit Switching ∧ Public Exchange, *]을 처리한 결과와 사용자가 선택적으로 퍼지 불리언 패싯 질의를 구성한 화면이다. 이때, 참조 그래프는 시소러스 부라우징 부분에

빨간 색으로 시각화되어 나타난다. 이 화면에서 'MAIN_QUERY'의 연산자 '∧'이 키보드에서 직접 입력될 수 없음으로 'GLBS'을 대신 사용하였다. 시소러스 질의에 의해 참조된 객체들은 사용자 만족 정도에 따라 순위화된 형태로 'Recommend Object' 영역에 제시되며, 이들을 이용하여 선택적으로 구성된 패싯 질의 "(Circuit Switching/0.93 OR Public Exchange/0.92) AND AXE10/1.0"은 'BOOL Facet Query' 영역에 나타난다. 이때, 사용된 연산자 'AND'와 'OR'는 4.2절에서 정의한 퍼지 논리곱 연산자(⊓)과 논리합 연산자(⊔)를 의미한다. 이 패싯 질의는 "Apply" 버튼에 의해 확장된 패싯 질의 화면에 반영된다. 또한, 시소러스 브라우징 부분에 표시된 모든 객체들은 하이퍼링크를 가지고 있기 때문에 선택된 객체에 대해 관련 정보는 화면을 통해 직접 참조될 수 있다. '+'기호로 표시된 'Packet Switching'과 'Private Exchange'은 현재 화면에 표시되지 않은 하나 이상의 하위 객체를 가지고 있음을 나타내며, 이 객체를 이중 선택함으로써 하위 객체들을 시각화할 수 있다. 이들 기능은 사용자가 관심 객체들을 쉽게 참조할 수 있도록 지원한다.

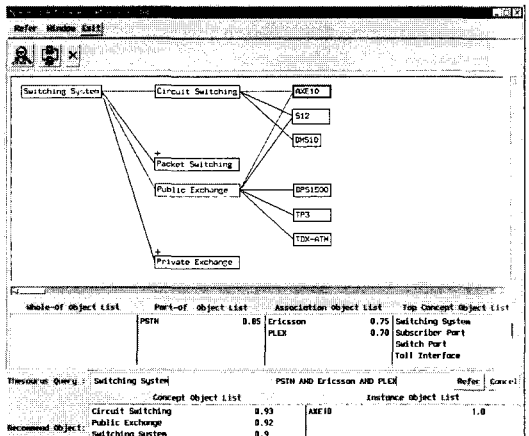

그림 6 시소러스 질의를 통한 자동 패싯 질의 구성

그림 6은 시소러스 질의 [Switching System, PSTN AND Ericsson AND PLEX]을 처리한 결과이다. 여기서, "AXE10/1.0 OR Circuit Switching/0.93 OR Public Exchange/0.92 OR Switching System/0.9"는 시스템에 의해 자동으로 구성된 퍼지 불리언 패싯 질의이다. 특히, 이 질의는 일정 수준의 정확률과 높은 재현률을 보장해야 하는 검색에 적절히 이용될 수 있을 것이다.

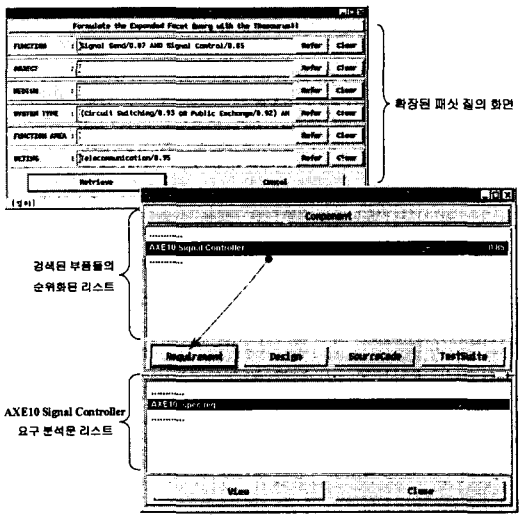

그림 7 확장된 패싯 질의를 이용한 부품 검색 결과

그림 7은 확장된 패싯 질의에 의해 검색된 결과이다. 확장된 패싯 질의 화면에서 비어 있는 질의 패싯은 무조건 항목을 나타내며, "Retrieve" 버튼은 시스템이 확장된 패싯 질의와 의미적으로 일치하는 부품을 검색하게 한다. 이때, 검색된 부품의 내용은 각 개발 단계에서 생성된 모든 문서들로 구성되기 때문에 사용자가 특정한 개발 단계를 선택하면 해당하는 재사용 가능한 문서 리스트가 제시된다.

그림 8은 관련 연구에서 비교된 REBOOT 시스템[8]이 패싯 질의를 표현하는 과정을 나타내고 있다. 이 시스템의 시소러스는 객체기반 시소러스에 비해 패싯 값들 사이의 다양한 관계와 관련 정도를 표현하고 있지 않으며, 패싯 질의에 시소러스 개념 "ActiveObject"을 추가하기 위해 사용자는 복잡한 구조의 시소러스를 순차적으로 브라우징하여 탐색해야 한다. 또한, 퍼지 불리언 패싯 질의를 이용하는 본 시스템에 비해 단순히 불리언 패싯 질의만을 이용하고 있기 때문에 사용자 의도를 정확히 패싯 질의로 표현하기 어렵다는 문제점을 가지고 있다. 따라서, 본 시스템은 높은 정확률을 요구하는 소프트웨어 재사용 환경에 보다 효과적으로 이용될 수 있다.

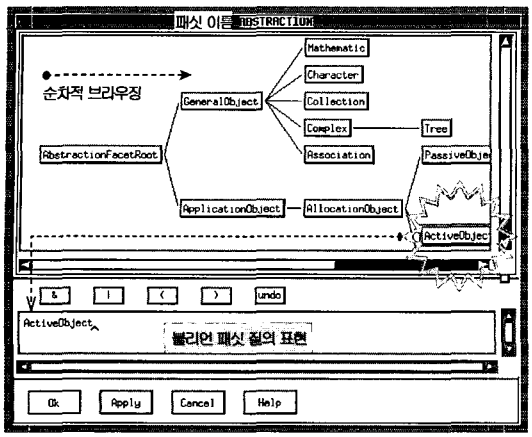

그림 8 REBOOT 시스템에서 "ABSTRACTION" 패싯 질의의 표현

6. 결론 및 향후 연구 과제

패싯기반 부품 검색 시스템에서 일반적으로 사용하는 패싯 질의는 의미 표현이 매우 제한적이기 때문에 사용자 요구를 충분히 검색 과정에 반영할 수 없다는 단점을 가지고 있다. 이 단점은 시스템이 실제 관련이 있는 부품들을 검색할 수 없게 하거나 관련이 없는 부품들까지 검색할 수 있게 하기 때문에 사용자가 검색된 부품들로부터 적절한 부품을 선정하기 위해 많은 시간과 노력을 소비해야하는 문제점을 가진다. 본 논문에서는 이 문제점을 해결하기 위해 사용자가 자신의 의도를 정확히 표현함으로써 시스템의 정확한 검색을 지원하는 확장된 패싯 질의를 정형화였으며, 이 질의를 처리할 수 있는 부품 검색 시스템을 설계하고 구현하였다. 이 시스템은 다음 두 관점에서 기존의 패싯기반 검색 시스템과 구별된다. 첫째, 확장된 패싯 질의에서 하나의 퍼지 불리언 패싯 질의는 패싯 값들 사이의 논리적인 의미 관계를 자연스럽게 표현할 수 있게 하며, 모든 패싯 값에 부여된 퍼지 정도는 검색된 부품들을 사용자 의도에 따라 순위화할 수 있게 한다. 둘째, 퍼지 불리언 패싯 질의를 구성하기 위해 이용되는 시소러스 질의는 사용자가 자신이 의도한 퍼지 패싯 값들을 직접 시소러스로부터 참조할 수 있도록 지원하기 때문에 도메인 전문가가 부품 분류를 위해 사용한 구체적인 의미의 패싯 값들을 질의에 효과적으로 반영할 수 있도록 한다. 특히, 시소러스 질의는 정확한 검색을 지원하기 위해 사용자가 요구하는 구체적인 의미의 패싯 값들을 쉽게 참조할 수 있게 하는 장점을 가진다.

향후 연구 과제로서 본 논문에서 구현된 확장된 패싯 질의를 이용한 부품 검색 시스템의 정량적인 평가를 들 수 있다. 특히, 이 평가를 위해 재사용 부품들에 대한 테스트 데이타베이스와 구조화된 객체기반 시소러스의 내용 구축이 선행적으로 요구된다.

참 고 문 헌

[1] B. Boehm, "Managing Software Productivity and Reuse," Computer, Vol. 32, No. 9, pp. 111-114, 1999.

[2] M. A. Lemley and D. W. O'Brien, "Encouraging Software Reuse," Stanford Law Review, Vol. 49, No. 2, pp. 255-305, 1997.

[3] R. J. Leach, *Software Reuse : Methods, Models, and Costs*, McGraw Hill, 1997.

[4] P. T. Devanbu, R. J. Brachman and P. G. Selfridge, "LaSSIE: A Knowledge-Based Software Information System," Communications of the ACM, Vol. 34, No. 5, pp. 34-49, 1991.

[5] S. Y. Liao, L. S. Cheung and W. Y. Liu, "An Object-Oriented System for the Reuse of Software Design Items," Journal of Object-Oriented Programming, Vol. 11, No. 8, pp. 22-29, 1999.

[6] H. C. Liao, M. F. Chen and F. J. Wang, "A Domain-Independent Software Reuse Framework Based on a Hierarchical Thesaurus," Software-Practice and Experience (SPE), Vol. 28, No. 8, pp. 799-818, 1998.

[7] E. Damiani, M. G. Fugini and C. Bellettini, "A Hierarchy-Aware Approach to Faceted Classification of Object-Oriented Components," ACM Transactions on Software Engineering and Methodology, Vol. 8, No. 3, pp. 215-262, 1999.

[8] G. Sindre, R. Conradi and E. Karlsson, "The REBOOT Approach to Software Reuse," Journal of Systems and Software, Vol. 30, No. 3, pp. 201-212, 1995.

[9] 이준호, 김명호, 이윤준, "긍정적 보상 연산자를 이용한 퍼지 집합 모델의 개선", 정보과학회 논문지(B), Vol. 20, No. 10, pp. 1417-1425, 1993.

[10] J. S. Poulin and K. P. Yglesias, "Experiences with a Faceted Classification Scheme in a Large Reusable Software Library (RSL)," 16th Annual International Computer Software and Applications Conference, Phoenix, AZ, 3-5 November, pp. 90-99, 1993.

[11] R. Prieto-Diaz, "Implementing Faceted Classification for Software Reuse," Communications of the ACM, Vol. 34, No. 5, pp. 89-97, 1991.

[12] M. R. Girardi and B. Ibrahim, "Using English to Retrieve Software," The Journal of Systems and Software, Special Issue on Software Reusability,

Vol. 30, No. 3, pp. 249-270, 1995.
[13] Y. Maarek, D. Berry and G. Kaiser, "An Information Retrieval Approach for Automatically Constructing Software Libraries," IEEE Transactions on Software Engineering, Vol. 17, No. 8, pp. 800-813, 1991.
[14] 최재훈, 한종진, 박종진, 양재동, "구조적인 시소러스 구축을 지원하는 객체 기반 정보 검색 모델", 정보과학회 논문지(B), Vol. 24, No. 11, pp. 1244-1256, 1997.
[15] C. Liu, C. Chang, C. Yuan, W. Chu and H. Yang, "A Formal Approach to Software Components Classification and Retrieval," IEEE International Computer Software and Application Conference(CompSac'97), Washington D.C., USA, August, 1997.
[16] S. C. Chou, J. Y. Chen and C. G. Chung, "A Behavior-based Classification and Retrieval Technique for Object-Oriented Specification Reuse," Software Practice & Experience, Vol. 26, No. 7, pp. 815-833, 1996.
[17] R. Helm and Y. S. Maarek, "Integrating Information Retrieval Approaches for Browsing and Retrieval in Object-oriented Class Libraries," In OOPSLA'91 Conference Proceedings, pp. 47-61, Oct., 1991.
[18] B. Y. Ricardo and R. N. Berthier, *Modern Information Retrieval*, Addison-Wesley, 1999.
[19] T. Isakowiz and R. J. Kauffman, "Supporting Search for Reusable Software Objects," IEEE Transactions on Software Engineering, Vol. 22, No. 6, pp. 407-423, 1996.
[20] 한종진, 배명남, 양재동, 이준경, "객체지향 소프트웨어 재사용 시스템을 지원하는 부품 항해 도구의 개발", 정보과학회 논문지(C), Vol. 4 No. 4, pp. 554-565, 1998.
[21] V. K. Balakrishnan, *Introductory Discrete Mathematics*, Dover Publications, 1996.
[22] J. D. Yang and H. J. Yang, "A Formal Framework for Image Indexing with Triples: Toward a Concept-Based Image Retrieval," International Journal of Intelligent Systems, Vol. 14, No. 16, pp. 603-622, 1999.

최 재 훈

1994년 2월 전북대학교 전자계산학 학사 졸업. 1996년 2월 전북대학교 전산통계학과 석사 졸업. 1996년 3월 ~ 현재 전북대학교 전산통계학과 박사 과정. 관심분야는 정보검색, 객체지향 데이타베이스, 소프트웨어 공학, 비디오 검색, 전자상거래 에이전트, 퍼지 이론, 지식기반 시스템.

김 기 헌

1907년 2월 전북대학교 전자계산학 학사 졸업. 1999년 2월 전북대학교 전산통계학과 석사 졸업. 1999년 3월 ~ 현재 전북대학교 전산통계학과 박사 과정. 관심분야는 객체지향 데이타베이스, 정보검색, 운영체제, 전자상거래 에이전트, 분산 시스템.

양 재 동

1983년 2월 서울대학교 컴퓨터공학과(학사). 1985년 2월 한국과학기술원 전산학과(석사). 1991년 2월 한국과학기술원 전산학과(박사). 1995년 1월 ~ 1996년 1월 Univ. of Florida, Visiting Scholar. 현재는 전북대학교 컴퓨터과학과 부교수, 영상·정보 신기술 연구소 연구원. 관심분야는 OODBs, Expert System, CASE

이 동 길

1983년 2월 경북대학교 전자공학 학사 졸업. 1985년 2월 KAIST 전산학 석사 졸업. 1996년 2월 KAIST 전산학 박사 졸업. 1985년 3월 ~ 현재 한국전자통신연구원 개발환경연구실 책임연구원 관심분야는 컴파일러, 프로그래밍 언어, 소프트웨어 공학