

# RDBMS 기반의 토픽맵 무결성 검사 기법

## (RDBMS based Topic Map Constraint Checking Mechanism)

이 한 준 <sup>†</sup>      민 경 섭 <sup>\*\*</sup>      김 형 주 <sup>\*\*\*</sup>  
 (Han-jun Lee)    (Kyung-Sub Min)    (Hyoung-Joo Kim)

**요약** 지식을 효과적으로 표현하고 검색하는 방법에 대한 관심이 증가함에 따라 토픽맵을 비롯한 지식 표현 기법들이 점점 중요해지고 있다. 토픽맵은 지식을 구조화하여 표현하게 되는데 지식은 매우 복잡한 의미적 관계를 포함하고 있기 때문에 이를 의미적으로 일관성 있게 유지 관리하는 것은 매우 중요한 문제가 된다. 현재 토픽맵 무결성 검사를 위하여 TMCL을 비롯한 여러 토픽맵 제약 조건 언어와 시스템들이 제안되었다. 그러나 이들은 단순한 의미적 제약들에 대해 지원하고 있지만 의존성과 같은 복잡한 제약들에 대한 것들은 다루고 있지 못하다. 또한, 단순히 응용 레벨에서 제약들을 검사함으로써 좋지 않은 성능을 보인다. 이에 본 논문에서는 기존의 정보 시스템 분야 및 관련 분야에서 제공하고 있는 제약 언어들의 특성을 기반으로 TMCL을 확장하였다. 또한 이를 효율적으로 지원하기 위한 관계형 데이터베이스 기반의 토픽맵 무결성 검사 기법을 제안하고 구현하였다. 구현한 시스템에서는 TMCL에서 제공하고 있는 제약 조건을 검사할 때 템플릿을 이용하여 질의를 생성시켜 효율적인 무결성 검사가 가능하게 함으로써 기존 시스템들의 검사 방식에 비해 높은 성능을 보임을 확인하였다.

**키워드** : 토픽맵, 무결성, 제약 조건

**Abstract** Due to a growing interest in searching and expressing knowledge effectively, knowledge management methods such as Topic Map are becoming more important. Topic Map organizes knowledge that is full of intricate relations, so maintaining and managing Topic Map consistently is very essential. TMCL and other constraint languages have limits as they can check simple constraints but can not support complex constraints like dependency constraints. Current constraint checking systems operating at the application level are also showing an inferiority in performance. In this paper, we extend TMCL based on the characteristics of other constraint languages in the information system field and related fields. We build and propose an RDBMS-based Topic Map constraint checking system to support the extended constraint language effectively. This new system handles complex types of constraints like dependency constraint as well as basic Topic Map constraints present in the TMCL. As the system examines each constraint it uses templates to generate queries for effective checking and overall shows a higher performance level than current systems.

**Key words** : Topic Map, Integrity, Constraint

· 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성 지원 사업(MITA-2007-C1090-0701-0031)의 연구결과로 수행되었음

논문접수 : 2004년 2월 4일

심사완료 : 2007년 10월 22일

<sup>†</sup> 비회원 : 한국국방연구원 연구원

leehj@idb.snu.ac.kr

<sup>\*\*</sup> 비회원 : 삼성전자 연구원

ksmin@idb.snu.ac.kr

<sup>\*\*\*</sup> 종신회원 : 서울대학교 전기컴퓨터공학부 교수

hjk@oopsia.snu.ac.kr

논문접수 : 2004년 2월 4일

심사완료 : 2007년 10월 22일

: 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 데이터베이스 제34권 제6호(2007.12)

Copyright©2007 한국정보과학회

## 1. 서 론

네트워크와 컴퓨터의 발달로 정보의 양이 폭발적으로 증가되고 있다. 이에 따라 정보의 효과적인 검색과 관리 방안이 중요한 문제로 부각되고 있으며 이러한 배경에서 ISO 표준인 토픽맵(Topic Map)을 이용한 지식 표현과 추출에 대한 관심이 증가하고 있다[1].

지식 및 정보들의 관계는 단순하지 않기 때문에 기존의 정보 시스템에서는 정보의 일관성 유지를 위해 다양한 형태의 제약을 필요로 한다. 토픽맵을 구축하고 관리할 때도 토픽맵을 의미적으로나 구조적으로 일관성 있게 유지하기 위한 제약이 필요하다.

현재, 토픽맵에 대한 의미적 무결성 보장을 위하여 몇 가지 언어가 제안되었는데 그 중 가장 대표적인 것은 TMCL이다. 기존의 정보 시스템에서 제공하는 무결성 보장을 위한 언어들은 보다 복잡하고 정교한 형태를 제공하며 다른 지식 표현 형태인 RDF나 DAML+OIL도 다양한 형태의 제약을 제공하고 있다. 반면 TMCL은 아직 그 구성이 미비한 편이며 보완 및 확장이 필요한 상태이다. 추가적으로 토픽맵을 기반으로 한 의미적 무결성을 제공하는 시스템들이 제안되었으나, 이들은 응용 레벨에서 검사를 지원하여 검사의 효율성이 크게 떨어지는 문제점이 있다.

그래서, 본 논문에서는 토픽맵을 기반으로 구성된 지식맵에 대한 의미적 무결성을 좀 더 체계적으로 보완할 수 있는 방법을 제안하려고 하였다. 이를 위해, TMCL에 기존의 언어들로부터의 특징들을 보완하고 확장하여 제안하였으며, 이를 효율적으로 지원하기 위한 관계형 데이터베이스 기반 무결성 검사 시스템을 구현하였다. 또한 본 시스템에서는 TMCL의 기본 제약 조건들 외에 토픽맵 항목들 간의 의존도에 대한 제약을 추가하여 확장하였다. 이는 토픽맵 임의의 항목에 대한 변경이 이루어질 때 의존 관계에 있는 항목들에 대한 처리를 함께 하도록 하기 위한 제약이다. 그리고 본 시스템에서는 제약 조건 검사에 효율적인 형태의 저장 모델을 제안하였고 이를 통해 검사에 필요한 질의가 각 제약 조건별로 템플릿을 이용하여 용이하게 생성할 수 있도록 하였다. 템플릿을 이용하여 질의 레벨에서 최적화된 검사를 시행함으로써 기존의 응용 레벨에서의 검사에 비해 효율적인 검사가 이루어지도록 하였다. 특히 의존도에 대한 제약 처리의 경우 검사가 복잡하여 많은 시간이 소요되기 때문에 이를 효과적으로 처리하기 위한 별도의 알고리즘을 제안하여 구현하였으며 이상의 기법들을 통하여 토픽맵 제약 조건 검사에 대한 성능 향상을 보였다.

논문의 구성은 다음과 같다. 2장에서는 토픽맵 모델의 기본적인 내용을 소개한다. 3장에서는 기존의 토픽맵 제

약 조건 관련 연구들을 살펴봄 4장에서는 본 논문에서 제안한 토픽맵 제약 조건 모델과 효율적인 제약 조건 검사를 위하여 제안한 방법을 구체적으로 설명한다. 5장에서는 토픽맵 기반의 전체 시스템의 구조와 이 시스템에서 이루어지는 토픽맵 무결성 검사 과정을 살펴본다. 6장에서는 기존 시스템과의 비교 실험 결과를 통해 성능 향상을 확인한다. 마지막으로 7장에서는 결론과 함께 향후 연구에 대해 알아본다.

## 2. 배경 지식

지식이나 정보를 효율적으로 관리하기 위한 연구는 이미 여러 분야에서 이루어져왔다. 문헌 정보학에서는 색인, 용어 해설, 시소러스 등을 통해 문서 등에 담긴 많은 정보들을 효과적으로 구조화하는데 사용해왔으며 인공 지능 분야에서는 인간과 기계 사이의 의사 전달을 위해 지식과 의미를 표현할 수 있는 방법을 연구해왔다.

지식 표현에 널리 사용된 방법 중 하나는 개념과 개념들 간의 관계로 구성된 개념도(conceptual graph)이다. 이 개념도는 '시맨틱 네트워크(semantic network)', '분할 네트워크(partitioned network)' 등의 다양한 이름으로 구현되어 왔다[2]. 이 시맨틱 네트워크의 기본 모델은 전통적인 분류 기법인 색인 용어 해설, 시소러스에서 나타나는 토픽(Topic), 어소시에이션(Association)과 매우 유사하기 때문에 이 두 기법을 접목하여 정보 관리와 지식 관리 분야에 많은 이점을 제공할 수 있으며 이것이 토픽맵이라는 새로운 표준을 만들게 된 이유라고 할 수 있다. 토픽맵은 토픽/어소시에이션 모델에 토픽/어커런스를 추가함으로써 지식 표현과 정보 관리 분야 사이의 연결 고리 역할을 담당할 수 있게 한다. 국제 표준화 기구(International Organization for Standardization)에서는 1999년 토픽맵을 지식맵 표준으로 제정하였다. 토픽맵은 정보나 개념을 나타내는 토픽과 토픽간의 연관 관계를 나타내는 어소시에이션 그리고 토픽에 해당하는 정보에 대한 위치를 나타내는 어커런스를 중심으로 구성된다. 또한 세부적으로 타입(type), 룰(role), 스코프(scope)등 다양한 부가 항목들을 통해 정보와 정보간의 관계를 구체적으로 표현할 수 있고 다시 이를 통해 새로운 지식을 얻을 수 있는 구조를 제공한다. 특히 정보의 위치를 명시하는 어커런스에서는 URL뿐만 아니라 Xpath나 데이터베이스의 특정 레코드에 이르기까지 다양한 표현이 가능하기 때문에 여러 종류의 정보에 대한 효과적인 조직화가 가능하다. 본 장에서는 토픽맵에 대한 기본적인 개념을 소개하고자 한다[3].

### 2.1 토픽

토픽은 사람, 사물, 개념, 의미 등 실제 존재하는 것, 또는 특정 속성이나 어떤 의미 등 어떠한 것도 될 수

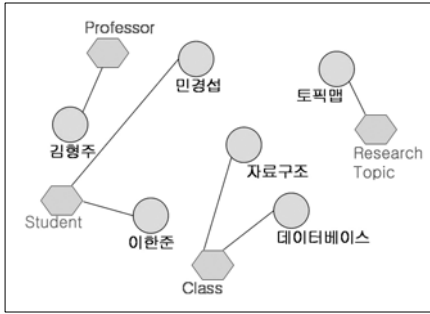


그림 1 토픽과 토픽타입

있다. 즉 특정 문서의 토픽은 정보를 기술하는 사람이 나타내고자 하는 대상이 되며 명사형으로 표현된다. 그림 1은 서울대학교 객체 지향 연구실 홈페이지에서 추출한 토픽의 예이다. 기술하고자 하는 대상으로서 '민경섭', '자료구조', '토픽맵' 등이 토픽으로 지정되었다. 하나의 토픽맵에는 수많은 토픽들이 존재하며 이들을 서로 연결짓기 위해서는 먼저 비슷한 유형에 속하는 것끼리 분류를 해야 한다. 이 때 토픽 타입을 사용한다. '자료구조' 토픽은 'class'라는 토픽 타입을 가지며 '이한준' 토픽은 'student'라는 토픽 타입을 가진다. 이처럼 토픽들은 토픽 타입을 이용하여 자신이 속한 스코프를 나타낼 수 있으며 이는 클래스와 인스턴스의 관계와 유사하다고 볼 수 있다. 토픽맵에 있는 토픽들은 토픽 타입이 없거나 있다면 하나 이상 여러 타입에 속할 수도 있다.

2.2 어커런스

각 토픽은 자신이 참조하는 하나 이상의 실제적인 지식 내용과 연결될 수 있으며 이를 어커런스를 통해 나타낸다. 그림 2는 연구실과 관련된 토픽들과 어커런스를 나타내고 있다. '데이터베이스'라는 토픽은 "http://oopsla.snu.ac.kr/class/DB"라는 강의 홈페이지 자원과 연결된다. 이러한 연결 정보를 어커런스라고 하며 어커런스는 문서 파일, 이미지 파일, 비디오 파일, 데이터베이스 내 특정 레코드 등 여러 형태로 나타날 수 있기

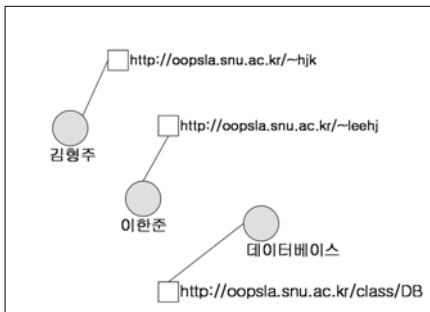


그림 2 어커런스

때문에 여러 형태의 정보를 토픽맵을 통해 통합할 수 있는 강력한 기능을 제공한다.

2.3 어소시에이션

토픽간의 연관 관계는 어소시에이션을 통해 표현된다. 이것은 토픽 타입과 같은 상하관계를 표현하는 것이 아니라 둘 이상의 토픽들 간의 관계를 정의하는 것이다. 토픽간의 관계가 어소시에이션으로 표현됨과 동시에 그 관계가 어떤 것인지는 어소시에이션 타입을 통해 나타내며, 이 관계에서 각 토픽이 어떤 역할을 하는지는 물을 통해 기술된다. 그림 3은 연구실 홈페이지에서 추출한 토픽들 간의 어소시에이션을 나타내고 있다. 먼저 '이한준' 토픽과 '토픽맵' 토픽은 '연구'이라는 관계를 가지고 있으며, 이 관계에서 '이한준' 토픽은 '연구원'의 롤 타입을 가지며 '토픽맵' 토픽은 '연구 분야'의 롤 타입을 가진다. 이것은 '이한준 연구원이 토픽맵이라는 연구 분야를 가지고 연구한다.'라는 사실을 표현하고 있다. 또한 어소시에이션에서는 일대일 관계 뿐 아니라 일대다 관계 혹은 다대다 관계에 대한 표현도 가능하다 예를 들어 '강의' 어소시에이션은 한명의 강사와 여러 과목에 대한 토픽 관계를 정의하게 된다. 이와 같은 방법으로 어소시에이션은 토픽간의 관계를 상세하게 기술할 수 있다.

토픽맵 표준안에서는 지금까지 살펴본 토픽, 어커런스, 어소시에이션과 함께 다양한 요소들을 제공하여 정보를 효과적으로 구조화할 수 있도록 한다.

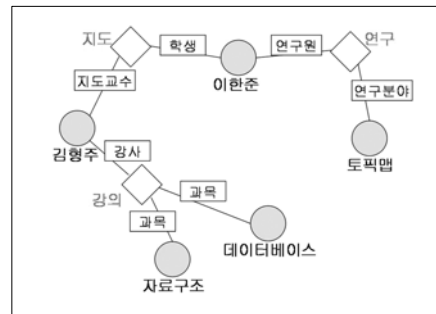


그림 3 어소시에이션

3. 관련 연구

토픽맵 제약 조건 관련 연구로써는 TMCL(Topic MapConstraint Language), AsTMA!(Asymptotic Topic Map Notation, Constraining), OSL(Ontopia Schema Language) 등 여러 가지로 진행 중이나 아직 표준은 정해지지 않은 상태이다. 이들 중 가장 많은 연구가 이루어진 TMCL이 대표적인 토픽맵 제약 조건 언어라 할 수 있다. TMCL과 AsTMA!의 경우 제약 조건 표준을

위한 연구로써 진행되어 온 반면 OSL은 온토피아(ONTOPIA)사(社)에서 만든 토픽맵 시스템에서 무결성 검사를 위해 만들어진 언어이다. XTCh는 문맥(contextual) 제약 조건과 스키마 제약 조건으로 나누고 이를 지원하는 별도의 프로세서를 제안하여 토픽맵의 무결성을 보장한다[4-7]. 이러한 언어들은 토픽맵 무결성 유지를 위한 제약 조건을 특정 기준에 따라 분류하고 나열한다는 점에서 유사하나 분류 기준과 제약조건을 나타내는 표현 방식 면에서 다소 차이가 있다.

토픽맵 분야에서 뿐만 아니라 다른 여러 분야에서 이미 오래 전부터 제약 조건과 관련된 연구들이 진행되어 왔다. 기존 RDB 분야에서도 도메인 제약조건 참조 무결성, 트리거, 함수적 종속 등의 다양한 종류의 제약을 이용하여 저장된 정보의 일관성과 정확성을 보장해왔다.[8][9] 토픽맵과 함께 지식 관리를 위한 대표적인 연구 분야에 속하는 RDF(Resource Description Framework), DAML+OIL(DARPA Agent Markup Language+ Ontology Inference Layer) 등의 시맨틱 웹 분야에서도 제약 조건에 관련된 연구가 진행되어 왔다. RDF는 문서나 임의의 데이터에 대하여 메타 정보를 표시하는 방법을 정의한 기술이다. RDF에서는 RDF 스키마를 이용하여 클래스 속성에 대한 제약 조건을 부과한다[10,11]. DAML은 여러 대학, 회사, W3C 멤버들과 팀 버너스리가 참여하는 DARPA 프로젝트의 중요한 부분으로써 XML 기술을 기반으로 고안된 의미론적 언어라고 할 수 있다. DAML은 RDF-SHOE-OIL과 같은 시맨틱 웹 언어들의 장점을 결합시키는 시도를 하고 있는데 현재는 OIL의 개발과 합쳐져서 DAML+OIL로 명명되었다. DAML+OIL에서는 여러 종류의 엘리먼트들을 통해 클래스의 속성 값에 대한 여러 종류의 제약 조건을 기술할 수 있다. 이상의 관련 연구들은 토픽맵에서보다 일찍 연구가 시작되었으며 현재 다양하고 많은 제약들을 통해 무결성을 보장한다[12-14].

지금까지 이루어져 온 관련 연구들을 잘 활용하여 토픽맵에 적합한 제약 조건을 적용하고 확장하는 것이 필요하다. 본 논문에서는 TMCL을 토픽맵 제약 조건 언어의 기본 모델로 하여 무결성 관련 연구에서 응용할 수 있는 제약 조건을 확장하였고 기존 토픽맵 기반 시스템에서의 검사 방식을 논의의 대상으로 하였다.

#### 4. 제약 조건 검사 모델

이번 장에서는 본 논문에서 다루고 있는 제약 조건들 및 확장한 내용, 제안한 제약 조건 검사 모델 및 검사 방식에 대하여 소개한다. 4.1절에서는 그동안의 연구를 바탕으로 토픽맵 제약 조건을 통합하여 분류하고 확장한 내용에 대해 다룬다. 4.2절에서는 제약 조건 문서에

대한 기술 형식을 소개하며 4.3절에서는 시스템에서 사용한 관계형 데이터베이스 기반의 저장 모델에 대해 설명한다. 4.4절에서는 질의 템플릿 예제를 통해 구체적인 제약 검사 방식에 대하여 알아보고 의존성 제약 조건 검사를 위해 제안한 알고리즘을 소개하고자 한다.

#### 4.1 제약 조건 분류

##### 4.1.1 기존 연구에서 분류된 제약 조건

TMCL에서는 크게 타입, 카디널리티(cardinality), 데이터, 구문(syntax) 제약조건으로 분류하며 이들은 다시 적용되는 토픽맵 항목별로 나누어진다. 구문에 대한 제약 조건은 토픽맵 문서가 XTM 형식에 맞는지 검사하는 것으로써 본 논문에서 제안한 시스템은 이를 토픽맵 파서가 담당하도록 하였기 때문에 별도의 제약 조건으로 분류하지는 않았다. AsTMa!나 OSL에서의 제약 조건 항목 중에 TMCL과 중복되지 않는 것은 추가하였다. 각 제약 조건을 분류하면 다음과 같다.

첫째로 타입 제약 조건은 토픽이나 어소시에이션, 어커런스 등의 타입과 관련된 제약 조건을 말한다. 토픽이 가질 수 있는 어커런스의 타입에 대한 제약이나 어소시에이션이 가질 수 있는 멤버(Member)의 타입에 대한 제약, 그리고 토픽이 갖는 타입의 상위 타입에 대한 제약을 말한다. 이 중 하나에 대해 예제를 통해 살펴보면 그림 4와 같다.

그림 4는 어소시에이션의 멤버가 롤 타입에 어긋나는 경우를 예제로 다루고 있다. 'T'는 토픽, 'A'는 어소시에이션을 가리킨다. '강의' 타입의 어소시에이션은 '강사' 롤 타입의 토픽과 '과목' 롤 타입의 토픽 간에 '강의' 관계를 기술해주는 어소시에이션이다. 예제에서 '교수' 타입의 토픽은 '강사' 롤 타입에 포함되나 '과목' 롤을 하는 멤버로써 적합지 않은 '학생' 타입의 토픽이 입력되어 의미적으로 잘못된 관계를 보여주고 있다.

두 번째 경우는 카디널리티 제약 조건이다. 예를 들어 '저자' 토픽과 '논문' 토픽이 서로 'written by'라는 어소시에이션으로 관계를 맺고 있을 때 '저자' 토픽의 수를 두 명 이상 열 명 미만으로 제한할 수 있다. 이처럼 어소시에이션이 가질 수 있는 멤버 수의 최소, 최대값에 대한 제약을 말한다. 토픽 이 가지는 어커런스 수의 최

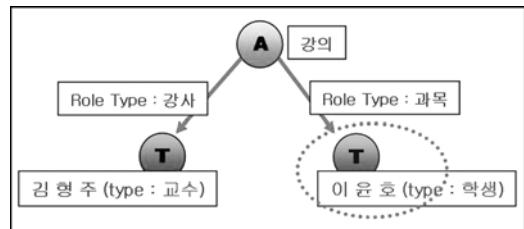


그림 4 타입 제약 조건 예제

소, 최대값에 대한 제약도 동일한 카디널리티 제약 조건에 해당된다.

세 번째는 존재성(Existence) 제약 조건이다. 토픽이나 어소시에이션, 어커런스들은 타입을 갖게 되며 이 타입은 토픽으로 지정이 된다. 존재성 제약은 참조하고 있는 타입에 해당하는 토픽이 반드시 존재하는 것을 보장하는 제약 조건이다.

네 번째는 데이터 제약 조건인데 이는 토픽이나 어커런스가 갖는 값에 대한 제약이다. 토픽의 경우 가진 값이 문자열인지 혹은 정수형이나 실수형인지 등에 대한 제약을 지정할 수 있으며 문자열이라면 길이라던가 숫자라면 값의 범위 등도 지정할 수 있다. 어커런스 역시 속성 값에 대한 동일한 제약이 적용될 수 있다. 이상에서 기존 제약 조건들을 분류해 보았다

4.1.2 제약조건의 확장

토픽맵은 토픽 항목들이 상호 관련성에 따라 어소시에이션을 통해 상호 연관 관계를 맺고 있는 형태로 구성이 되며 이러한 연관 관계는 토픽맵에 있어 중요한 지식 표현의 요소가 된다. 그러나 토픽맵에는 토픽 항목들간의 연관 관계의 종류는 설명할 수 있으나 상호 의존 관계를 기술하는 방법은 없는 상태이다 따라서 본 논문에서는 토픽맵 항목 간의 연관 관계를 의미적인 의존도에 따라 제약을 부여하여 이를 해당 항목의 변경이 있을 때 반영할 수 있도록 하였다. 관계형 데이터베이스에서도 외래키와 참조 무결성, 연쇄 삭제 등을 통해 제약을 부과했으며 ODBMS에서도 유사한 제약을 부여할 수 있다. 이들을 토픽맵에 적합한 형태로 응용한 것이라 볼 수 있다[15,16].

그림 5는 객체 지향 연구실에 대한 토픽맵의 일부를 나타낸다. 연구실 구성원들은 '객체지향 연구실'의 일원인 동시에 특정 팀(예를 들어 'KM팀')의 일원이다. 연구실에는 여러 개의 팀들이 존재하며 각 팀마다 해당되는 관련 분야가 있다.

이러한 상황에서 하나의 토픽을 삭제하는 경우를 생

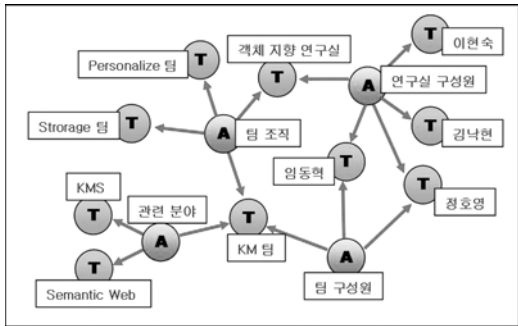


그림 5 객체 지향 연구실 토픽맵

각해 보자. 다른 항목과의 의존도에 따라 아래 세 가지 중 하나의 형태로 처리가 되어야 한다.

- 삭제되어서는 안되는 경우
- 해당 항목만 제거되면 되는 경우
- 해당 항목에 관련되어 있는 종속적인 항목들과 함께 제거되어야 하는 경우

그림 5에서 '객체 지향 연구실' 토픽과 같이 '연구실' 타입의 토픽은 해당 토픽맵 전체 구성에 있어서 삭제되면 곤란한 항목이므로 제거될 수 없도록 제약을 준다. 'KM팀' 항목의 관련 분야인 'KMS' 토픽과 같이 '분야' 타입 토픽인 경우는 다른 항목들과의 의존 관계가 없으므로 삭제하고자 할 때 이 항목만 삭제가 되도록 한다. '팀' 타입의 토픽인 'KM팀' 토픽은 의미적으로 그림 6과 같은 의존 관계를 가진다. 즉 '팀' 타입의 토픽은 '관련 분야'나 '팀 구성원' 타입의 어소시에이션들과 의존 관계가 있으며 '관련 분야' 타입의 어소시에이션은 '분야' 타입의 토픽에 대한 의존 관계가 있다. 그 외의 관련 항목들과는 독립적이다. 예를 들어 'KM' 토픽을 제거하면 종속 토픽들까지 같이 제거되어야 하며 결과적으로 그림 7과 같은 형태가 되어야 한다.

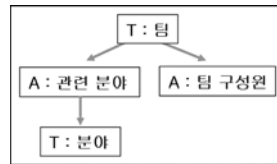


그림 6 의존성 제약 조건 관련 예제

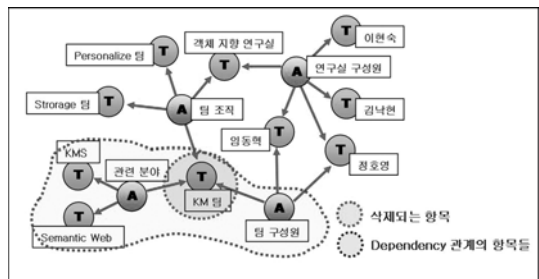


그림 7 의존성 제약 조건 관련 예제

이처럼 토픽맵 항목들 사이의 의존 정도에 따라 삭제 시 함께 고려되어야 할 필요가 있으며 본 논문에서는 이런 제약을 의존성(Dependency) 제약 조건으로 정의하여 다루었다.

4.2 제약 조건 표현 방법

본 시스템에서 제약 조건 검사를 위한 문서는 토픽맵 문서와는 별개의 텍스트 문서로 입력을 받게 된다. XTM 형식에 따라 기술되었기 때문에 입력된 제약 조건 문서

[1] TMConstraint	::= '<TMConstraint>' description* '</TMConstraint>'
[2] description	::= ('<CType=""type"">' type-param '</ConstraintType>')* ( '<CType=""cardinality"">' cardinality-param '</ConstraintType>')* ( '<CType=""data"">' data-param '</ConstraintType>')* ( '<CType=""dependency"">' dependency-param '</ConstraintType>')*
[3] type-param	::= '<type>' type '</type>' ('<member-type>' type '</member-type>')+
[4] cardinality-param	::= '<type>' type '</type>' '<min>' vlaue '</min>' '<max>' value '</max>'
[5] data-param	::= '<type>' type '</type>' '<data-type>' ('int'   'real'   'string') '</data-type>'
[6] dependency-param	::= '<type>' type '</type>' '<dependent-type>' type '</dependent-type>'
[7] type	::= ([a-zA-Z0-9_])+
[8] value	::= ('   ε ) [0-9]+ ( ε   '.' [0-9]+ )

그림 8 제약 조건 문서 기술 형식

는 XTM 파서를 이용하여 데이터베이스에 저장된다.

그림 8에서는 제약 조건 문서에 대한 기술 형식을 보여주고 있다. 토픽맵 제약 조건은 토픽맵 항목의 타입별로 부여되며 제약 조건의 종류별로 검사에 필요한 각각의 매개 변수와 함께 기술된다.

#### 4.3 저장 모델

토픽맵 기반의 기존 시스템에서는 토픽맵 저장을 위하여 관계형 데이터베이스나 파일을 이용한다. 파일 형태로 저장할 경우 데이터베이스를 사용하는 경우에 비해 관리 및 자료 이용시 약점을 갖게 되기 때문에 본 논문에서 구현한 시스템에서는 관계형 데이터베이스를 사용하였다. 토픽맵의 토픽은 RDBMS의 엔티티(Entity)와 유사하며 어소시에이션은 릴레이션(Relation)과 유사하기 때문에 직관적으로는 토픽맵 모델을 E-R 모델로 변환하여 관계형 데이터베이스에 저장하는 것이 유리하다. 그러나 이 경우 타입별로 테이블이 생성되기 때문에 스키마 구조가 토픽맵 내의 타입 형태에 종속적으로 나타나게 된다. 그 결과 토픽맵 항목들의 변경에 따라 스키마도 변경되어야 하는 문제점이 발생한다. 또한 모델 간 변환 과정 비용이 크다는 것과 효과적인 제약 조건 검사가 어렵다는 것 역시 단점이 된다. 이 때문에 본 논문에서는 토픽맵 항목들의 타입별로 저장 구조를 만들지 않고 토픽, 어소시에이션, 어커런스 등 항목 유형별로 저장하는 구조로 설계하였다.

토픽맵을 관계형 데이터베이스에 저장하기 위해 본 논문에서 설계한 스키마는 그림 9와 같다. 토픽맵, 토픽, 어소시에이션, 멤버, 어커런스 테이블들이 중심이 되어 토픽맵이 구성된다. 각 테이블의 필드들은 토픽맵의 의미 정보에 따라 상호 연관되어 있다. 토픽맵에 적용될 제약 조건들은 앞서 설명한 제약 조건의 분류에 따라 테이블별로 저장이 되는데 타입, 카디널리티, 데이터,

의존성 테이블들이 제약 조건 검사에 이용된다. 이를 통해 모델 간 변환 비용을 최소화하고 타입과 무관한 형태의 저장 구조를 갖도록 하여 스키마 구조가 타입들과 독립적으로 일관된 형태를 지니도록 하였다. 그 결과 제약 조건 검사에 필요한 질의들이 일정한 형태를 갖게 되어 효율적인 검사가 가능하도록 하였다. 4.4절에서 더 다루도록 한다.

#### 4.4 제약 조건 검사 방식

본 논문의 시스템에서는 제약 조건별 템플릿에 기반하여 필요한 질의를 생성시킨다. 검사 방식의 과정은 다음과 같다. 임의의 토픽맵을 가정할 때, 특정 항목에 대한 변경 사항이 발생하면 이 항목이 템플릿에 입력된다. 우선 입력된 항목에 대한 제약 조건이 있는지 여부를 조사하게 된다. 해당 항목에 대한 제약 조건이 없는 경우라면 별도의 검사 없이 반영되나 관계된 제약 조건이 하나 이상 있는 경우에는 제약 조건들을 각각 검사하게 된다. 제약 조건의 종류에 따라 질의 템플릿이 검사에 필요한 질의들을 생성해낸다. 질의 결과를 통해 변경하고자 하는 내용이 유효한 것인지를 밝힌 다음 유효할 경우 이를 토픽맵에 반영시키고 그렇지 않은 경우 변경을 무효화시킨다.

그림 10에서는 템플릿에서 질의를 생성시키는 예제를 나타내고 있다. 검사에 필요한 정보들을 해당 제약 조건 테이블들에서 추출하여 제약 조건별로 검사에 필요한 질의를 생성해낸다. 존재성 제약의 경우 'student' 타입을 갖고 있는 '이한준'이라는 토픽이 입력된다면 템플릿은 타입 테이블에서 student 타입의 토픽이 테이블에 있는지를 검색하는 질의를 생성시킨다. 카디널리티 제약의 경우 'written by' 타입을 갖는 어소시에이션에서 '저자'를 타입의 토픽 수에 대한 제약이 있다면 우선 멤버 테이블과 'Player' 테이블에서 해당 어소시에이션의 '저

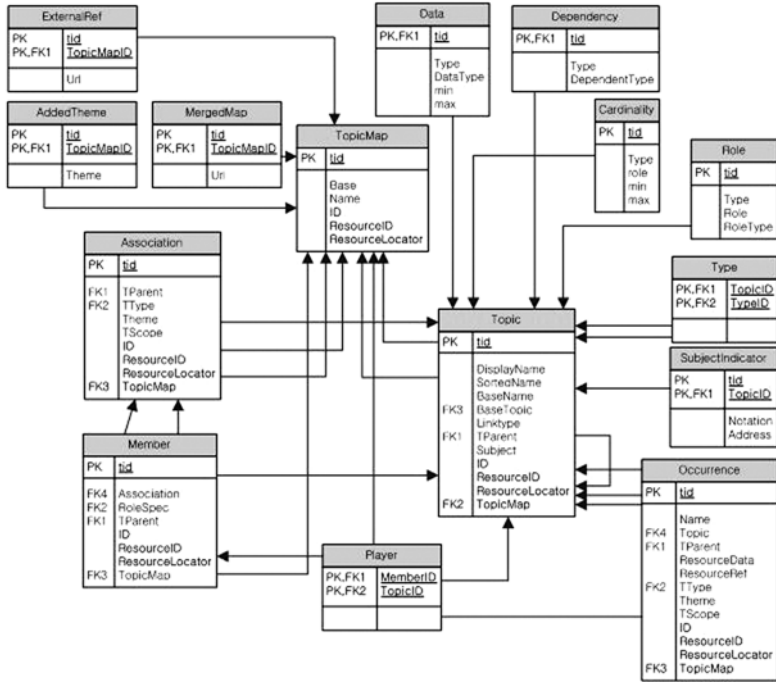


그림 9 스키마 구성

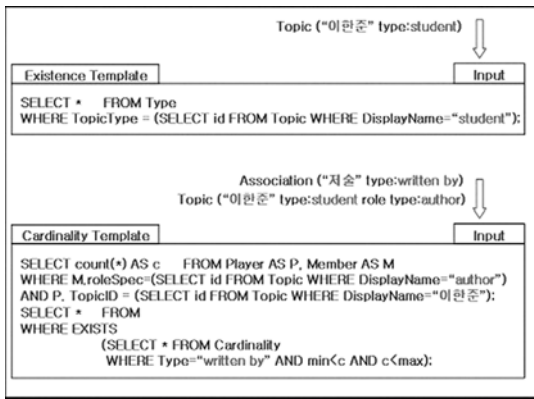


그림 10 질의 템플릿 예제

자' 를 타입에 해당하는 토픽의 아이디의 개수를 확인한다. 카디널리티 테이블에서는 'written by' 타입에 대한 카디널리티 범위 값을 담고 있는 'min', 'max' 필드 값과 비교하여 제약 조건 검사가 진행된다. 이와 같은 방식으로 템플릿에 기반하여 검사에 필요한 질의들을 생성시키게 된다.

의존성 제약 조건의 경우에도 템플릿을 이용하여 해당 항목들에 대한 처리를 할 수 있다는 점은 유사하나 해당 항목과 의존 관계에 있는 항목들의 범위가 일정하지 않기 때문에 다른 제약 조건과는 달리 해당 범위의

항목들을 찾아내는 작업이 선행되어야 한다는 차이점을 갖는다. 앞서 살펴본 그림 6의 예제에서와 같이 임의의 토픽을 삭제하기 위해서는 의존성 제약 관계를 참조하여 관련된 항목들을 모두 추출해야 한다. 이를 위해서, 토픽맵은 토픽과 어소시에이션들의 그래프로 생각할 수 있기 때문에 기존의 방식으로는 의존 관계에 있는 항목들을 찾기 위해서 해당 항목에 대한 의존 관계에 있는 때까지 순회해야 한다. 제약은 타입별로 부과되므로 하나의 토픽에 대한 제약 범위를 알기 위해서는 해당 토픽의 타입을 찾아 이 타입에 종속적인 항목의 타입을 찾고 다시 이 타입의 항목을 해당 토픽에서 찾아가는 방식을 반복하며 이 순회 과정 중에 제약 조건에 해당되는 항목들을 각각 삭제해 간다. 예제에서 'KM팀' 토픽의 경우 '관련 분야' 타입의 어소시에이션과 의존 관계가 있음을 확인하고 이에 대한 인스턴스에 해당되는 항목을 토픽맵에서 찾은 뒤 다시 '관련 분야'와 의존 관계가 있는 타입들을 찾는 과정을 반복해야 한다. 이 때문에 제약 조건의 범위의 크기가 크면 클수록 많은 시간이 소요되는 문제점이 있다.

본 논문에서는 이 제약 조건에 대한 처리를 질의 템플릿과 함께 별도의 자료 구조를 이용하여 보다 효율적으로 처리하고자 하였다. 의존성 제약 조건은 그림 6과 같이 트리 형태로 나뉘어질 수 있다는 점에 착안하여 토픽맵 항목들을 차례차례 순회하지 않고 해당 항목들

을 직접적으로 접근하도록 하고자 했다. 이를 위해 structural join 알고리즘을 이용하였다. 각 노드의 항목에 structural join 알고리즘에 의해 시작 번호와 끝 번호의 쌍으로 이루어진 인덱싱 키 값을 부여할 수 있으며 이 키 값을 이용하면 특정 노드로부터 해당 노드의 하위 노드들을 트리 순회 없이 추출해낼 수 있다 [17] 임의의 타입으로부터 의존 관계에 있는 항목들의 타입들을 알 수 있기 때문에 위에서 살펴본 예제에서 '팀' 타입의 항목인 경우 의존 관계에 있는 타입은 '팀구성원', '관련 분야', '분야' 임을 키 값을 통해 추출해낼 수 있다. 의존 관계의 타입들은 시작 번호와 끝 번호의 쌍으로 알 수 있지만 의존 관계의 인스턴스들에 대한 정보는 추가적인 정보가 필요하다. '팀' 타입의 인스턴스인 'KM 팀' 항목의 경우 의존 관계가 있는 'KMS', 'Semantic Web' 항목 등을 직접적으로 추출하기 위해서 인스턴스별 번호를 추가로 부여한다. 결과적으로 토픽맵의 모든 항목들은 두 개의 키 값인 시작 번호와 끝 번호, 그리고 인스턴스 번호의 세 개의 필드로 이루어진 아이디를 갖게 된다. 키값과 인스턴스 번호를 통해 토픽맵의 모든 항목들은 각각 유일한 아이디를 가지게 되며 임의의 항목에 대한 아이디만 알게 되면 별도의 작업 없이 하위노드에 해당되는 인스턴스들을 질의를 통해 일괄적으로 추출해낼 수 있게 된다. 이 때문에 제약 조건에 해당되는 항목들의 범위와 무관하게 일정한 시간 내에 처리가 가능하게 된다.

## 5. 시스템의 구조

전체 시스템의 개략적인 모습은 그림 11과 같다. 서버측에서는 토픽맵에 대한 서비스를 제공하게 되며 클라이언트 측에서는 브라우저를 통해 접속하여 토픽맵을 검색하거나 갱신 등의 연산을 하게 된다. K Box는 서울대 객체 지향 시스템 연구실에서 만든 어플리케이션으로써 토픽맵 생성 및 탐색을 지원하는 위한 API들을 제공한다. 래퍼(Wrapper)는 RDB TM4J를 기반으로 만들어졌다. RDB TM4J는 온토피아사에서 공개 소스(open source)로 만든 TM4J 토픽맵 Engine을 RDB 상에서 사용할 수 있도록 수정된 것으로 토픽맵에 대한 래퍼 역할을 하게 된다. 제약 조건을 처리하는 제약 조건 검사 엔진(Constraint Engine)은 RDB TM4J와 관계형 데이터베이스 기반의 토픽맵 저장소(RDBMS TM storage) 사이에 위치하게 된다.

토픽맵은 XTM 1.0 명세에 따라 텍스트 형식의 XML 문서 형태로 입력을 받게 되며 제약 조건 관련 파라미터들이 담긴 텍스트 파일도 함께 입력을 받게 된다. 토픽맵 문서가 입력되면 이를 토픽맵 파서가 파싱을 하게 되는데 이 때 XTM 형식에 맞는지를 함께 검사하게 된

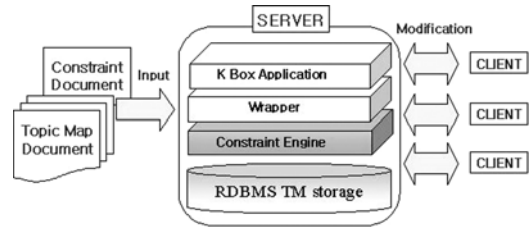


그림 11 시스템 구조도

다[18]. 문법 오류가 없는 경우 제약 조건 검사 엔진을 통해 4절에서 설명한 방식의 검사가 이루어지며 이를 데이터베이스에 반영하게 된다. 이후 사용자가 의한 변경이 있을 때마다 동일한 방식으로 이에 대한 유효성을 검사한 뒤 데이터베이스에 반영한다.

## 6. 실험 및 분석

이 장에서는 지금까지 본 논문에서 제안한 방식대로 SQL 질의 템플릿을 이용하여 질의 레벨에서 제약 조건 검사를 하게 될 경우 기존 시스템과 같이 응용 레벨에서 제약 조건에 대한 검사를 하게 될 경우에 비해 성능이 향상되었음을 실험 결과를 통해 보이고자 한다.

### 6.1 실험 환경 및 방법

실험을 위해 서울대학교 객체 지향 연구실 홈페이지의 내용을 기반으로 토픽맵을 구축하고 제약 조건을 정의하였다. 실험에 사용된 시스템은 Pentium 4 1.4GHz의 CPU와 512MB의 메인 메모리를 가지며 OS는 Windows XP를 사용하였고 DB는 Oracle 8.1.7을 사용하였다.

현재까지 개발된 기존의 토픽맵 제약조건 검사 시스템들은 응용 레벨에서 동작한다. 토픽맵에 대한 수정 사항이 있을 때 응용 레벨에서 질의를 생성시켜 검사가 이루어지는 방식이다. 본 논문에서는 질의 템플릿 검사 시스템 외에 기존의 응용 레벨 검사 방식의 시스템과의 비교를 위해 그림 11의 K-Box Application에 무결성 검사를 위한 질의 생성 부분을 추가한 별도의 시스템을 구현하여 제안한 시스템과 성능 비교 실험을 하였다. 실험의 목적은 본 논문에서 제안한 질의 템플릿 검사 방식이 기존의 응용 레벨 검사 방식에 비해 효율적임을 밝히는 것이다. 이 두 방식의 검사 방식을 두 가지 종류의 실험에 각각 적용하였다. 첫 번째 실험은 XTM 문서로부터 토픽맵을 파싱하여 저장하는 과정에서 제약 조건 검사가 일괄적으로 이루어질 때 소요되는 전체 검사 시간을 측정하였다. 이 때, 실험에 사용되는 토픽맵 문서의 크기를 달리하며 그 결과를 비교하였다. 두 번째 실험은 각 제약 조건 검사를 위해 걸리는 시간을 측정하여 검사 효율을 비교하였다. 제약 조건들 중 의존성 제약 조건의 경우에는 보다 자세한 비교를 위해 의존성



계약 조건에 연관된 항목들의 깊이를 달리해가면서 소요 시간을 측정하였다.

6.2 실험 결과 및 분석

첫 번째 실험에서는 각각 100KB, 200KB, 300KB 크기의 토픽맵 문서를 이용하였고 검사해야 할 제약 조건의 수는 문서의 크기에 비례하도록 하였다. 이 토픽맵 문서를 파싱하여 데이터베이스에 저장할 때 제약 조건을 검사하게 되는데 이 때 걸리는 시간을 측정하였다. 결과는 그림 12와 같았다.

실험 결과를 통하여 검사 시간이 제약 조건의 수에 비례적으로 증가되는 것을 확인할 수 있었으며 응용 레벨의 검사보다 질의 템플릿을 이용한 검사가 효율적임을 볼 수 있었다.

두 번째 실험으로 제약 조건 각각의 처리 시간에 대한 비교 실험을 하였다. 존재성 제약 조건은 임의의 토픽이 갖는 타입의 존재 여부에 대한 제약을 예제로 실험하였고 타입 제약 조건은 두 개의 멤버를 갖는 어소시에이션에서 롤 타입에 대한 제약을 이용했다. 카디널리티 제약 조건은 임의의 토픽이 속한 어소시에이션이 갖는 멤버들의 카디널리티 수에 대한 제약을 실험에 이용하였다. 실험 결과 그림 13과 같이 질의 처리가 보다 복잡한 형태일수록 시간이 많이 걸렸다. 존재성 제약 조건보다는 타입 제약 조건이, 타입 제약 조건보다는 카디널리티 제약 조건 검사가 더 복잡하거나 많은 질의가 생성되기 때문에 보다 더 많은 시간이 걸렸으며 응용 레벨 검사보다 템플릿 검사의 효율이 더 크게 드러났다.

의존성 제약 조건에 대한 검사는 의존성 제약이 얼마나 많은 항목에 해당되는가를 기준으로 실험하였다. 다른 항목들과 의존성 제약 조건이 없는 항목을 삭제할 때부터 깊이 3까지, 해당 항목과 보다 더 먼 깊이까지 제약이 걸린 경우로 깊이를 늘려가면서 비교하였다.

의존성 제약 조건을 처리할 때는 기본적으로 토픽맵을 해당 항목으로부터 순회하는데 필요한 질의들로 인해 시간이 많이 든다. 그림 14에서 확인할 수 있듯이 본 논문에서 제안한 방식을 통해 처리할 경우에 응용 레벨에서의 처리와 같이 제약 범위에 따라 증가하지 않고 일정 시간 내에 처리가 가능함을 확인할 수 있었다. 제약에 해당되는 항목들을 효과적으로 추출함으로써 검사에 필요한 시간을 크게 단축시킬 수 있었으며 특히 탐색의 깊이가 깊을수록 기존 방식에 비해 성능 향상이 더 크게 드러났다.

이상의 실험들을 통하여 본 논문에서 제안한 검사 방식이 기존의 검사 방식에 비해 효율적인 검사가 가능함을 확인할 수 있었다. 특히 처리에 많은 질의가 필요한 복잡한 제약일수록 처리에 걸리는 시간이 크기 때문에 이러한 제약을 효과적으로 처리할 수 있는 질의 템플릿

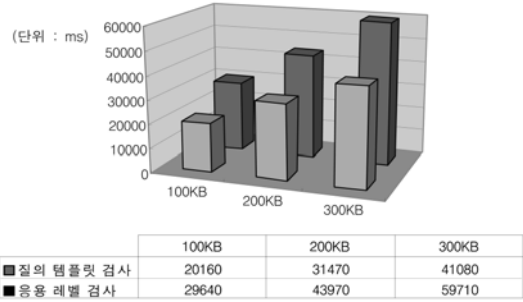


그림 12 토픽맵 크기에 따른 검사 시간 비교

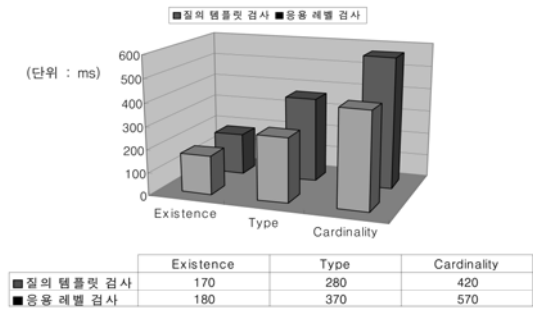


그림 13 제약 조건별 검사 시간 비교

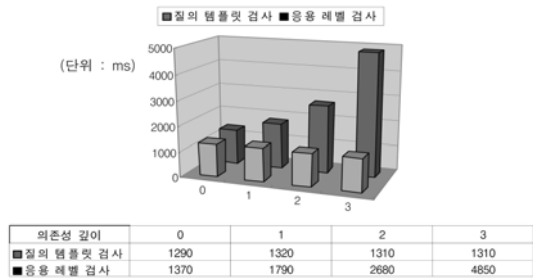


그림 14 거리에 따른 의존성 제약 조건 검사 시간과 평균값 비교

검사 방식이 큰 성능 향상을 가져올 수 있었다. 또한 의존성 제약 조건에 대한 처리에 대해 적용한 방식이 토픽맵 기반 시스템에서 제약 조건 처리의 효율성을 크게 향상시킬 수 있음을 확인하였다.

7. 결론

본 논문에서는 TMCL을 비롯한 제약 조건 관련 언어들 및 기존의 정보 시스템 분야에서 다루는 제약 조건들을 통합하여 적용하였다. 뿐만 아니라 의존성 제약 조건을 제안하여 다루는 제약 조건의 범위를 확장시켜 토픽맵의 의미적 무결성 보장을 강화시켰다.

또한 템플릿에 기반한 검사기법을 제안하여 각 제약

조건 항목들에 대한 질의들의 템플릿에 의해 생성되도록 하였다. 제약 조건별로 일정한 질의 형식을 갖게 되므로 템플릿을 이용할 때 효과적인 질의 생성이 가능하였다. 그리고 템플릿 기반의 검사 기법은 질의 레벨에서 검사를 처리하기 때문에 기존 응용 레벨 검사에서는 불가능했던 질의 최적화를 통해 기존 시스템보다 검사 효율을 높일 수 있었다.

향후 별개의 토픽맵을 서로 병합하거나 제약 조건이 변경되는 등 다양한 상황에서도 효율적으로 토픽맵 무결성을 보장하기 위한 여러 기법에 대한 연구가 필요할 것으로 보인다.

### 참 고 문 헌

- [1] Michel Biezunski, Martin Bryan, Steve Newcomb, "ISO/IEC 13250 Topic Maps," 2002.
- [2] Rafal Ksiezzyk, "Answer is just a question [of matching Topic Maps]", XML Conference and Exposition 2000, 2000.
- [3] Steve Pepper, "The TAO of Topic Maps," XML 2000 Conference and Exposition 2000, 2000.
- [4] Graham Moore, "Topic Map Constraint Language," ISO/IEC 1/SC34, 2005.
- [5] Graham Moore, "Topic Map Constraint Language (TMCL) Requirements and Use Cases," ISO/IEC 1/SC34, 2004.
- [6] R. Barta, "AsTMA= 2.0 Language Definition," 2004. "http://astma.it.bond.edu.au/"
- [7] Product White Paper, "The Ontopia Knowledge Suite," Ontopia, Inc.
- [8] Giovanni Librelotto, "XTche - A Language for Topic Maps Schema and Constraints," XML Conference and Exposition 2004, 2004.
- [9] Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts," McGraw-Hill.
- [10] Elmasri, Navathe, "FUNDAMENTALS OF DATABASE SYSTEMS," Addison-Wesley.
- [11] Ora Lassila, Ralph R. Swick, "Resource Description Framework(RDF) Model and Syntax Specification," W3C Recommendation 22 February 1999, URL:http://www.w3.org/TR/REV-rdf-syntax
- [12] Jack Park, Sam Hunting, "XML Topic Maps - Creating and using Topic Maps for the web," Addison-Wesley, 2003.
- [13] Horrocks, I., et. al. "The DAML+OIL Language Specification," URL:http://www.daml.org/2001/03/daml+oil.daml, 2001.
- [14] Ian Horrocks, Peter F. Patel Schneider, and Frank van Harmelen. "Reviewing the design of DAML+OIL: An ontology language for the semantic web," In Proc. of AAAI-2002, 2002.
- [15] Eric Freese, "Using DAML+OIL as a Constraint Language for Topic Maps," XML Conference and Exposition 2002, 2002.
- [16] C. J. Date, "Referential Integrity," VLDB, pp. 2-12. 1981.
- [17] H. V. Jagadish, Xiaolei Qian, "Integrity Maintenance in Object-Oriented Databases," VLDB, pp. 469-480, 1992.
- [18] Chun Zhang, Jeffrey Naughton, David DeWitt, Qiong Luo, Guy Lohman, "On Supporting Containment Queries in Relational Database Management Systems," SIGMOD Conference 2001.
- [19] Steve Pepper, Graham Moore, "XML Topic Maps (XTM) 1.0," TopicMaps.Org.



이 한 준

2001년 서울대학교 컴퓨터공학과 학사  
2004년 서울대학교 컴퓨터공학과 석사  
2004년 7월~2007년 6월 육군사관학교  
전자정보학과 전임강사. 2007년~현재 한  
국국방연구원 연구원. 관심분야는 시맨틱  
웹, 데이터베이스



민 경 섭

1995년 항공대학교 컴퓨터공학과 학사  
1997년 서울대학교 컴퓨터공학과 석사  
2005년 서울대학교 인지과학과 박사  
2005년~현재 삼성전자. 관심분야는 데이  
타베이스, XML, 지식 검색

김 형 주

정보과학회논문지 : 데이터베이스  
제 34 권 제 5 호 참조