

정보검색시스템의 모델링 및 정형 검증⁰

⁰김종환* 박혜숙* 방기석** 장성호** 백두권*
고려대학교 컴퓨터학과
*{ange196, phs, baik}@software.korea.ac.kr

Formal Modeling and Verification of an Information Retrieval System

⁰Jong-Hwan Kim, Hea-Sook Park, Ki-Seok Bang, Sung-Ho Jang, Doo-Kwon Baik
Dept. of Computer Science & Engineering, Korea University

요 약

정보검색시스템은 지역적으로 분산된 이기종의 컴퓨터들 간의 데이터 교환과 다양한 종류의 DB에 대한 접근을 가능하게 함으로써, 기업에게 공급망 관리에 대한 통합된 정보를 제공한다. 정보검색시스템은 X3.285에 기반한 데이터 레지스트리 모델을 이용하여 모델링되었고, 모델링이 주어진 가정 하에서 요구사항을 만족하는지를 검증하기 위해 핵심 부분인 메타데이터 레지스트리 관리자(MDR Manager) 부분만을 SMV를 이용하여 검증하였다.

1. 서 론

기업집합환경 하에서 대부분의 제조 시스템은 속성상 지역적으로 분산되어 있으며, 각 지역별로 분산된 컴퓨터들의 응용 프로그램들의 상호 작용에 의하여 운용된다. 따라서 분산된 응용 프로그램들이 서로 간의 협조 하에 자신들의 역할을 수행하도록 하고, 이들을 체계적이고 효율적으로 조정 및 통합하는 소프트웨어는 매우 중요한 역할을 한다.

이러한 제조 시스템을 대상으로 개발되는 소프트웨어는 개발시 많은 시간과 비용이 든다. 또한 넓은 범위의 환경에 적용할 수 있기 위해서는 분산된 컴퓨터들 간의 데이터 교환과 다양한 종류의 DB에 대한 접근이 가능해야 한다. 이러한 기능을 지닌 소프트웨어는 기업의 공급망 관리에 대한 의사결정을 지원하기 위하여 기업에게 통합된 정보를 제공할 수 있다.

본 논문에서는 X3.285[1]를 이용한 데이터 레지스트리 모델(Data Registry Model)[2] 기반의 정보검색시스템을 제안하여 기업에게 통합된 정보를 제공하고자 한다. 정보검색시스템은 메타데이터 레지스트리와 스키마 레지스트리를 이용하여 모델링되었다.

또한 정보검색시스템의 모델링이 주어진 가정 하에서 요구사항을 만족하는지를 검증하기 위해 정형기법[3]을 사용한다. 정형기법은 수학과 논리학에 기반을 둔 방법으로 하드웨어나 소프트웨어 시스템을 명세하거나 검증하는 것이며, 이러한 정형기법에는 크게 정형명세와 정형검증이 있다. 정형명세는 정형논리 또는 수리논리에서 사용되는 기호 등을 이용하여 시스템이 동작할 환경에

대한 가정, 시스템이 만족해야 할 요구사항, 그리고 요구사항을 수행할 시스템 설계 등을 기술하는 것이다. 또한 정형검증은 역시 정형논리 또는 수리논리에서 제공하는 증명방법 등을 이용하여 정형명세를 분석하여 무모순성 및 완전성을 검증하거나, 설계가 주어진 가정에서 요구사항을 만족하는지를 검증하는 기법이다. 여기서는 정보검색시스템 모델의 검증을 위해 정형검증도구인 SMV를 이용하였다.

본 논문은 다음과 같이 구성되어 있다. 2 장에서는 정보검색시스템의 아키텍처에 대해서 설명한다. 3 장에서는 정보검색시스템의 모델링과 SMV를 통한 검증에 대해 설명한다. 마지막으로 4 장에서는 결론을 맺는다.

2. 정보검색시스템(IRS : Information Retrieval System)

기업의 의사결정을 지원하기 위한 정보검색시스템은 크게 네 부분으로 구성되며, 그림 1 과 같은 아키텍처를 가진다.

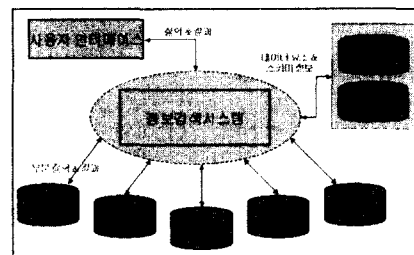


그림 1 정보검색시스템 아키텍처

⁰ 본 연구는 한국과학재단 2000년 특정기초연구로 이루어졌습니다.

즉, ① 사용자의 질의를 받기 위한 사용자 인터페이스 부분 ② 사용자의 질의 함수를 분산되어 있는 기업 DB에 맞추어 분석, 변환, 처리, 통합하여 사용자가 원하는 정보를 제공하는 정보검색시스템 부분 ③ 데이터 레지스트리와 스키마 레지스트리 정보를 저장하는 메타데이터 레지스트리와 스키마 레지스트리 부분 ④ 실제 데이터를 저장하고 있는 Low 데이터베이스인 분산된 데이터베이스 부분으로 구성된다.

기업의 데이터베이스는 각각 담당 부서나 기업에 따라 다른 형태의 테이블과 데이터 요소를 갖고 있다. 사용자는 이러한 각각의 데이터베이스의 스키마 형태를 고려하지 않고 사용자가 원하는 정보를 얻기 위한 질의를 한다. 정보검색시스템은 다른 테이블 구조를 갖는 실제 데이터베이스를 통합하기 위해 메타데이터를 이용한 메타데이터 레지스트리 관리기(MDR Manager)에 의해 가공된 정보를 MDR에 저장한다. MDR Manager는 같은 의미를 지닌 스키마의 의미를 통합하여 MDR에 조회, 변경, 삭제, 추가할 수 있는 기능을 제공한다.

3. 정보검색시스템의 모델링과 검증

정보검색시스템의 모델링과 검증 과정은 그림 2와 같다. 정보검색시스템 아키텍처를 기반으로 정보검색시스템을 모델링하였다. 정보검색시스템의 핵심부분은 메타데이터 레지스트리 관리기(MDR Manager)와 스키마 레지스트리 관리기(Schema Manager)로서, 여기서는 정보검색시스템의 MDR Manager 부분만을 STATEMATE를 사용하여 Statechart로 명세하였고, MOCES를 이용하여 SMV Model로 자동 변환한 뒤 Model Checker인 SMV를 통하여 정형 검증하였다.

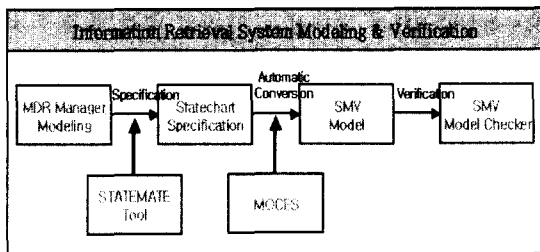


그림 2 정보검색시스템 모델링 & 검증

3.1 정보검색시스템 모델링

정보검색시스템 아키텍처에 기반하여, 메타데이터 레지스트리와 스키마 레지스트리를 이용한 정보검색시스템을 그림 3과 같이 모델링하였다. 정보검색시스템에서의 정보 흐름은 사용자 인터페이스를 통한 질의를 정보검색시스템 내의 질의함수처리기, MDR Manager, Schema Manager, 질의분석기, 질의변환기, 질의수행기, 결과통합기라는 프로시저를 거쳐서 원하는 정보를 사용자에게 제공하도록 하는 시스템이다.

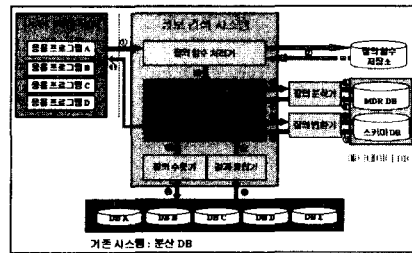


그림 3 정보검색시스템 모델링

정보검색시스템의 수행 절차는 다음과 같다.

- ① 사용자 인터페이스를 통하여 사용자 질의를 받은 후 질의함수처리기로 넘겨준다.
- ② 질의함수처리기는 질의함수저장소를 이용하여 질의함수를 호출한 뒤 호출된 함수를 MDR Manager로 전달한다.
- ③ MDR Manager는 같은 의미를 지닌 스키마의 의미를 통합하여 질의분석기를 거쳐 MDR에 조회, 변경, 삭제, 추가할 수 있는 기능을 제공하며, Schema Manager에 정보를 전달해 준다.
- ④ Schema Manager는 스키마 레지스트리를 참조하여 실제 분산되어 있는 데이터베이스에 대한 스키마 정보로 질의를 변환한 뒤(질의변환기), 각각의 데이터베이스에 부분 질의를 수행하여 결과를 얻어오며(질의수행기), 이러한 결과를 평가하고 통합하여(결과통합기) 그 결과를 사용자에게 원하는 형태로 가공하여 보여 주게 된다.

이 중 정보검색시스템의 MDR Manager는 데이터 레지스트리를 바탕으로 한 데이터 레지스트리 모델에 기반한다. 데이터 레지스트리(Data Registry)란 데이터를 묘사, 검색, 분석, 분류하기 위해서 필요한 데이터의 특징들에 대한 사실을 보유하고 있는 저장소(Repository)로서 이를 바탕으로 하여 메타데이터 요소들의 상호운용성, 재사용성, 표준화 등의 핵심 기능을 갖는다[4]. 데이터 레지스트리 모델은 Object Class Entity와 Property Entity 개념을 통합하여 Data Element Concept Entity를 구성하고, Value Domain Entity와 Data Element Concept Entity를 통합한 DR의 핵심인 Data Element Entity를 구현하는 모델이다.

3.2 Statechart로의 명세

조회 버튼(OBJECT, PROPERTY, DATA_ELEMENT_CONCEPT, VALUE_DOMAIN, DATA_ELEMENT)을 클릭하였을 때 MDR과의 접속이 되지 않는다면 바로 "Connect Failure or SQL Error!!!"라는 에러 메시지를 띄우기 때문에, MDR와의 접속이 원활히 된다고 가정하고, STATEMATE를 이용하여 설계한 MDR Manager의 Statechart 명세는 그림 4와 같다. Statechart[5]는 시스템의 동작을 상태 기계라는 형식의 그림으로 명세하기 때문에 정형기법에 익숙하지 않은 사람도 쉽게 이해할 수 있고, 시스템의 동작을 보다 명확하고 가시적으로 시뮬레이션 할 수 있는 장점도 있지만, 시스템의 특성을 증명하는 정형검증의 기능은 제공하지 못한다. 따라서 Statechart로 명세된 것을 SMV를 통하여 정형검증한다.

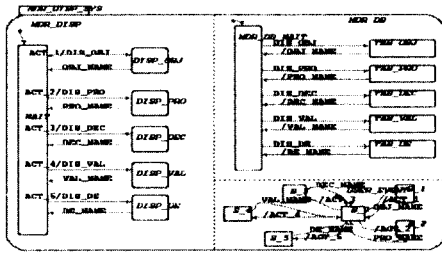


그림 4 MDR Manager의 Statechart 명세

3.3 SMV를 이용한 검증

정형검증 툴로는 VERSA[6], SPIN[7], SMV[8] 등 다양하다. 여기서는 SMV를 이용하여 검증한다. SMV 시스템은 유한상태시스템이 CTL[9]로 표현된 요구 명세를 만족하는지를 검증하는 정형검증도구이다. SMV의 입력 언어는 유한상태시스템을 명세하기 위해 만들어졌으며, 시스템은 이 입력 언어를 통해서 동기적인 밀리 머신이나, 비동기적인 네트워크로 손쉽게 명세될 수 있다. SMV는 입력 언어로 나타내어진 모델이 CTL로 표현된 요구 명세를 만족하는지의 여부를 효율적으로 검사하기 위해서 OBDD[10]를 기반으로 하는 심볼릭 모델 체크 알고리즘을 사용한다.

이와 같은 SMV를 이용하여 MDR Manager가 조화 버튼을 클릭하였을 때 MDR에 접속하여 각각의 데이터를 정확하게 가져올 수 있는가를 SMV를 이용하여 검증한다. 이를 위해 Statechart 명세를 MOCES를 이용하여 SMV 모델로 변환한 후, 모델의 동작을 검증하기 위해서 아래와 같은 CTL 명세를 SMV 모델에 추가한다. CTL 명세는 각각 다음과 같은 의미를 지닌다.

- ① AG(gen_ACT_1 -> AF gen_OBJ_NAME) : 버튼을 1이 눌렸을 경우 즉, OBJ_NAME을 검색하는 요청이 들어왔을 경우에는 언제나 OBJ_NAME을 출력해야 한다. SMV 모델에서는 요청과 응답이 이벤트로 표현되어 있기 때문에 해당하는 이벤트가 출력되는지를 증명하게 된다.
- ② AG(gen_ACT_2 -> AF gen_PRO_NAME) : ①과 마찬가지로 PRO_NAME을 검색하는 요청이 들어왔을 경우 PRO_NAME을 출력해야 한다는 것을 나타낸다.
- ③ AG(!(gen_ACT_1 & gen_ACT_2)) : 이 식은 두 가지 이상의 요청이 동시에 입력되면 안 된다는 것 즉, 상호 배제적인 동작을 명세한 것이다.

SMV를 이용해서 검증한 결과는 그림 5 와 같다.

```

-- specification EF (TRIG_DISP_PRO_WAIT1 & EX EFF_DISP_PR... is true
-- specification EF (TRIG_TAB_DEC_MDR_DB_WAIT1 & EX EFF_I... is true

-- specification EF (TRIG_TAB_DE_MDR_DB_WAIT1 & EX EFF_TA... is true
-- specification AG (gen_ACT_1 -> AF gen_OBJ_NAME) is true
-- specification AG (gen_ACT_2 -> AF gen_PRO_NAME) is true
-- specification AG (!(gen_ACT_1 & gen_ACT_2)) is true

resources used:
user time: 640.66 s, system time: 0.24 s
BDD nodes allocated: 534234
Bytes allocated: 9306112
BDD nodes representing transition relation: 147117 * 1
    
```

그림 5 MDR Manager의 SMV Model 검증 결과

위에서 보는 바와 같이, 모든 요구사항을 만족하며, 이는 정보검색시스템의 MDR Manager가 정확하게 MDR로부터 데이터를 가져옴을 의미한다.

4. 결론

정보검색시스템은 지역적으로 분산된 기기종의 컴퓨터들 간의 데이터 교환과 다양한 종류의 DB에 대한 접근을 가능하게 함으로써, 기업에게 공급망 관리에 대한 통합된 정보를 제공한다. 정보검색시스템은 X3.285에 기반한 데이터 레지스트리 모델을 이용하여 모델링되었고, 모델링이 주어진 가정 하에서 요구사항을 만족하는지를 검증하기 위해 핵심 부분인 메타데이터 레지스트리 관리자(MDR Manager) 부분만을 Statechart로 명세한 뒤 SMV를 이용하여 검증하였다. 검증결과, 정보검색시스템의 MDR Manager가 모든 요구사항을 만족하며, 이는 MDR Manager가 MDR로부터 정확하게 데이터를 가져옴을 의미한다.

이러한 데이터 레지스트리와 스키마 레지스트리를 기반으로 하는 정보검색시스템의 모델링과 검증을 통한 기대 효과는 다음과 같다. 첫째, 데이터 레지스트리에 등록된 데이터 요소는 사용자나 응용 프로그램 개발자에게 전체 데이터베이스에 대한 통합적인 뷰를 제공할 수 있다. 둘째, 분산 환경 하의 데이터베이스들의 스키마 구조가 변할 때, 스키마 레지스트리에 등록된 정보만을 변경함으로써 시스템 전체의 독립성 및 유연성을 확보할 수 있다. 셋째, 분산 및 독립적으로 존재하는 기업의 정보에 대하여 통합 및 관리가 가능하다.

이번 연구에서는 정보검색시스템의 모든 구성요소를 명세하고 검증하지는 않았다. 실제로 정보검색시스템을 정형검증하기 위해서는 스키마 레지스트리 관리자, 질의 분석기, 질의변환기, 질의수행기, 결과통합기 등의 모든 컴포넌트가 명세되고 검증되어야만 할 것이다.

참고문헌

- [1] "Metamodel for the management of sharable data", ANSI X3.285, WD, 1998.
- [2] 박해숙 외, "데이터 레지스트리에 기반한 전자상거래 메타데이터 공유환경", 정보과학회 학술대회 논문집(B), pp.60-62, 2000. 4.
- [3] Edmund M. Clarke and Jeannette M. Wing, "Formal Method : State of the Art and Future Directions", ACM Computing Surveys, pp.626-643.
- [4] "Metamodel for the Management of Shareable Data", ANSI X3.285, Feb. 1999.
- [5] David Harel, "Statecharts : A Visual Formalism for Complex Systems", Science of Computer Programming 8, 1987.
- [6] Duncan Clarke, "VERSA : Verification, Execution and Rewrite System for ACSR", University of Pennsylvania, 1994.
- [7] Gerald J. Holzmann, "The Model Checker SPIN", IEEE Transactions on Software Engineering, VOL. 23, NO 5, MAY 1997, pp. 279-295., 1997
- [8] Kenneth L. McMillan, "SYMBOLIC MODEL CHECKING", Kluwer Academic Publisher, 1993.
- [9] Zohar Manna and Amir Pnueli, "The Temporal Logic of Reactive and Concurrent systems - Specification", Springer-Verlag, 1996.
- [10] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation", IEEE Transaction Computers, vol. 35, no. 6, pp. 677-691. 1986.8.