

비즈니스 룰의 안전성을 위한 증명기법*

신용녀*, 차리서*, 최진영*, 배민오**
고려대학교 컴퓨터학과 정형기법연구실*
동덕여자대학교 전자계산학과**

Formal Verification for Business Rule Safety

Yong-Nyuo Shin*, Reeseo Cha*, Jin-Young Choi*, Mino Bai**

Formal Methods Lab., Dept. of CSE, Korea University*
Dept. of Computer Science, Dongduk W University**

요 약

바람직한 비즈니스 룰의 기본은 어디에 어떠한 형태로 구현되더라도 유일하고, 일관적인 모델 이어야 한다는 것이다. 또한 비즈니스 룰 자체의 정립은 의사결정 시스템을 사용함으로써 경영 전략이 바뀌어도 시스템 담당자에게 의뢰할 필요 없이 기획자가 바로 코드 값만 변경시키면 기존의 시스템을 그대로 사용할 수 있을 정도의 유연성을 가지게 된 현실에 비추어 볼 때 매우 요원한 일이다. 즉, 비즈니스 룰은 DRM(Digital Rights Management) 시스템뿐만 아니라 e-CRM을 실현하는 모든 시스템에서 사용되며, 이 비즈니스 룰의 충돌로 인해서 시스템의 작동 여부가 결정될 수 있다. 룰 자체가 자연어로 기술되고 언젠가는 모순을 유발할 가능성이 높기 때문에 먼저 명확한 논리식으로 표현하고 LEGO라는 정형기법 도구를 통해 검증하도록 한다.

1. 서론

비즈니스 룰(business rule)이란 업무에 필요한 데이터 처리를 수행하는 응용프로그램의 일부를 말한다. 현재 전자상거래에서 비즈니스 룰은 특정 언어나 프로그래밍 메커니즘으로 자동 실행 가능하도록 인터넷 전자 상거래에서 폭넓게 사용되고 있다. 본 논문에서는 다양한 채널을 통해 유통되는 디지털 콘텐츠를 불법 사용으로부터 보호하고, 콘텐츠 소유자가 정한 사용 규칙이 지속적(persistent)으로 관리되도록 하는 기술 [3]인 DRM 시스템의 비즈니스 룰을 모델로 한다. DRM 시스템의 가동에 있어서 비즈니스 룰(business rule)간의 충돌은 DRM시스템의 작동 여부를 판가름 지을 만큼 중요한 문제임을 발견하였다. 최악의 경우

DRM 기술이 적용된 콘텐츠를 사용자가 돈을 내지 않아도 볼 수 있는 상황이나 돈을 아무리 많이 내도 못 보는 상황이 발생할 수 있다. 또한 비즈니스 룰 자체의 정립은 의사결정 시스템을 사용함으로써 경영 전략이 바뀌어도 시스템 담당자에게 의뢰할 필요 없이 기획자가 바로 코드 값만 변경시키면 기존의 시스템을 그대로 사용할 수 있을 정도의 유연성을 가지게된 지금의 현실에 비추어 볼 때 매우 요원한 일이다. 즉, 비즈니스 룰은 DRM 시스템뿐만 아니라 e-CRM을 실현하는 모든 시스템에서 사용되며, 이 비즈니스 룰의 충돌(conflict)로 인해서 시스템의 작동여부가 결정될 수 있다. 또한 전략 변경 자체에 업무를 집중시키는 데도 일조를 할 수 있게 된다. 애매한 자연어의 비즈니스

룰을 정형화시킴으로써 충돌 가능성을 최소화 시켜 EC에 이용될 수 있는 시스템의 안정성을 높이는데 이 논문의 목적이 있다. 본 논문은 2장에서는 비즈니스 룰과 XML에 대해 기술하고 3장에서는 비즈니스 룰을 로직화 시키는 것을 살펴보고, 4장에서는 논리식으로 표현된 비즈니스 로직이 모순이 발생하는 상황을 LEGO를 통해 보이고, 5장에서 결론을 맺는다.

2. 비즈니스 룰(Business Rule)

2.1 비즈니스 룰

비즈니스 룰이란 업무에 필요한 데이터 처리를 수행하는 응용프로그램의 일부를 말한다. 이것은 데이터 입력, 수정, 조회 및 보고서 처리 등을 수행하는 루틴, 좀더 엄밀히 말하면 보이는 것의 그 뒤에서 일어나는 각종 처리를 의미한다. 비즈니스 룰은 비즈니스 정책들을 표현한다. 문자 그대로 비즈니스 행동을 제어하고 정책에 영향을 준다. 즉, 비즈니스를 수행하면서 무엇이 가능하고 바람직하지 혹은 어떤 것이 그렇지 않을지를 알려 준다. 비즈니스 룰의 기본적인 특성은 다음과 같이 정의 할 수 있다.

- 명백한 표현

비즈니스 룰은 다른 의미로 해석되지 않도록 명시적으로 표현하여야 한다. 따라서 그래픽이나 정형언어(로직기반)로 표현되어지는 것이 바람직하다.

- 선언적인 성질

비즈니스 룰은 선언적이어야지 절차적이어서는 안 된다. 비즈니스 룰은 어디에 삽입되던지 또는 어느 문맥에서 나타나던지 간에 동일한 의미로 해석될 수 있도록 표현하여야한다.

- 일관적인 표현

모든 비즈니스 룰은 단일하고 일관적인 모델로 표현되는 것이 바람직하다.

바람직한 비즈니스 룰의 기본은 어디에 그들이 구현되더라도 그들의 원천이 무엇이나는 것에 상관없이 유일하고, 일관적인 모델 이어야 한다는 것이다. 이러한 일관적인 모델을 선언성을 가진다고 하는데, 이러한 선언성은 이중의 어플리케이션 또는 기업간에 룰의 교

환, 룰의 수정, 인간이 룰을 이해할 때 도움이 된다.

요즘 EC에서 룰은 특정언어나 프로그래밍 메커니즘으로 자동 식해 가능하도록 이터네 저자 사거래에서 폭넓게 사용되고 있다. 상품이나 서비스 판매자의 제안, 구매자의 요구사항들, 계약 협정, 인증 정책, 보다 일반적으로 e-business 정책이나 프로세스 등을 표현하고자 할 때 사용될 수 있다. 또한, 룰간의 충돌은 비즈니스 룰을 갱신하거나 합칠 때 많이 발생한다. EC 어플리케이션에서 룰은 마케팅 매니저 같은 비즈니스 도메인 전문가들에 의해 구체화되고 다양한 조직의 다양한 사람들에 의해 만들어진 룰이 합쳐질 때 자주 수정되어 진다.

2.2 XML

성숙한 디지털 콘텐츠 시장에서 저작권을 보호하기 위한 신뢰할 수 있는 시스템을 만들어서 디지털 콘텐츠가 저작권을 인증 받은 상태에서 온라인 상에 판매되고 다양한 플랫폼에 독립적으로 실행될 수 있도록 하기 위한 표준언어의 요구가 있어왔다. 컴퓨터에 의해 번역되어 지는 규칙은 반드시 객관적이어야 하며 "디지털 계약"과 일치해야 한다[1]. 콘텐츠의 특정한 용어나 조건에 의해 허가되지 않은 기능을 수행하도록 요청 받는다면 신뢰할 수 있는 시스템은 수행하지 않아야 한다. 사용규칙은 디지털 내용물과 함께 패키징되어야 하며 수행에 있어 자동성을 가져야 한다. XML(eXtensible Markup Language)은 웹 상에서 구

```
<?xml version="1.0"?>
<!DOCTYPE brml SYSTEM "brml.dtd">
<clp>
  <erule rulelabel="emptyLabel">
    <head>
      <cliteral predicate="giveDiscount">
        <function name="percent5"/>
        <variable name="?Cust"/>
      </cliteral>
    </head>
    <body>
      <and>
        <fcliteral predicate="shopper">
          <variable name="?Cust"/>
        </fcliteral>
        <fcliteral predicate="loyalCustomer">
          <variable name="?Cust"/>
        </fcliteral>
      </and>
    </body>
  </erule>
</clp>
```

그림 1 shopper이고 loyalCustomer 일 경우 5% discount를 적용한다는 비즈니스 룰

조화된 문서를 전송 가능하도록 설계된 표준화된 마크업 언어이다[2]. XML은 안전한 디지털 내용이 정상적인 결재 없이 교환되는 접근을 막고 사용 통제뿐만 아니라 디지털 콘텐츠의 출판과 판매를 지원한다.

3. 비즈니스 룰의 적용

각각의 비즈니스 룰 자체는 명확하더라도 룰간의 모순 관계가 있다면 충돌이 발생할 수 있다. 전통적인 프로그래밍 언어를 이용하여 운영 프로그램에 직접 룰을 코딩을 하든지, 삽입된 프로시저를 이용하여 데이터베이스에 트리거 시키는 지와는 상관없이 비즈니스 룰간의 모순은 실현하고자하는 시스템의 작동 여부를 결정짓게 할 수 있을 정도로 중요한(critical) 문제로 볼 수 있다. 룰 자체가 자연어로 기술되고 언젠가는 모순을 유발할 가능성이 높기 때문에 먼저 명확한 논리식으로 표현하고 LEGO라는 정형기법 도구를 통해 검증하도록 한다. 이러한 비즈니스 충돌을 사전에 예방하거나 이미 발생된 비즈니스 룰간의 모순을 발견하기 위해 비즈니스 룰 자체의 검증은 요원하며, 예시를 통하여

1. 이용자가 선불로 지불해 두고 아직 요금 결제에 사용하지 않은 액수를 선납료라고 한다. 모든 eContent는 열람 1회당 1000원의 요금이 부과되며, 이는 선납료에서 공제한다. 따라서 선납료가 1000원 미만인 이용자는 다시 요금을 지불하여 선납료를 1000원 이상으로 만들지 않는 한 열람 권한을 얻을 수 없게 된다.
2. eContent가 18세 이하의 고객에게는 열람이 금지된 것이면, 이 eContent를 보기 위해서는 이용자의 연령이 18세 이상이어야만 한다. 고객이용자의 연령은 이용자에 관한 정보를 저장하는 DB에서 읽어들이는 이용자의 생년월일과 시스템이 제공하는 현재 시각을 이용하여 정보를 산출한다.
3. 선납료가 100만원이 되면 그 회원은 그 이후 영구적으로 VIP회원이다.
4. VIP 회원은 모든 eContent를 수시로 임의 열람할 수 있다. 열람 건당 따로 지불해야 하는 요금은 전혀 없다. 즉, 선납료에서 공제하지 않는다.

표 1. DRM 디스플레이 비즈니스 룰

이러한 경우를 사전에 예방하거나 문제 발견 시 그 문제를 해결할 방법을 제안하고자 한다. DRM 시스템의 디스플레이 관련 빈에서 가정할 수 있는 비즈니스 룰은 [표 1]과 같다.

기존 비즈니스 룰은 1, 2, 3으로 이루어져 있었으나 이후에 4의 비즈니스 룰을 추가하는 경우에 룰간 모순이 일어난다. 즉, 4 번 룰에 따르면 VIP 회원일 경우 모든 eContent를 열람할 수 있는데, 18세 이하도 VIP가 될 수 있는 경우를 고려하지 않았기 때문에 비즈니스 룰 자체가 모순이 된다.

$N = \{x \mid x \text{ is natural number}\}$ $f \in N, a \in N$ Usage : $N \rightarrow N \rightarrow \text{Proposition}$ VIP : $N \rightarrow N \rightarrow \text{Proposition}$ Rule 1 : $\forall f, a \in N \cdot f < 1000 \Rightarrow \neg \text{Usable}(f, a)$ Rule 2 : $\forall f, a \in N \cdot a < 18 \Rightarrow \neg \text{Usable}(f, a)$ Rule 3 : $\forall f, a \in N \cdot f > 1000000 \Rightarrow \text{VIP}(f, a)$ Rule 4 : $\forall f, a \in N \cdot \text{VIP}(f, a) \Rightarrow \text{Usable}(f, a)$
--

표 2. Predicate Formulas

3.1 Predicate Logic으로의 변환

범용성을 확보하기 위해서 대부분의 비즈니스 룰은 first-order logic으로 변환하는 것이 이상적이다[4].

[표 3]과 같이 18세 미만의 VIP 회원이 18세 미만에게는 금지되는 콘텐츠를 사용할 수 있게 되는 상황이 발생하게 된다면 비즈니스 룰간에 충돌(conflict)이 발생한 것이다.

$\forall f, a \in N \cdot a < 18 \rightarrow f > 1000000 \rightarrow \text{Usable}(f, a)$

표 3. 증명할 Goal Formula

4. 비즈니스 룰의 증명

4.1 LEGO

LEGO는 Randy Pollack에 의해서 디자인되고 뉴저지 ML을 사용하여 구현된 대화식 증명 개발 시스템이다 [5]. natural deduction 스타일로 대화식 증명 개발에 매우 강한 툴이며 재정의의 증명(refinement proof)을 기본적 오퍼레이션으로 지원한다. LEGO는 상이한 논리

적 시스템을 정형화시키는데 사용되거나 정의된 로직에 기반한 정리(theorem)를 증명하는데 사용되어질 수 있다.

4.2 LEGO를 이용한 증명

비즈니스 룰간에 충돌이 일어나는 것을 사전에 막기 위해서는 룰들이 모순됨을 증명할 수 있어야 한다. LEGO에서 이를 증명하기 위해서 [표 3]의 Goal formula를 정의하였다. DRM 시스템 작동 시에 있어서는 발생되지 말아야하는 상황을 표현한 것이다. 이 formula를 LEGO 툴을 사용하여 참으로 증명함으로써 [표 2]의 비즈니스 룰간의 충돌이 발생했음을 증명할 수 있었다.

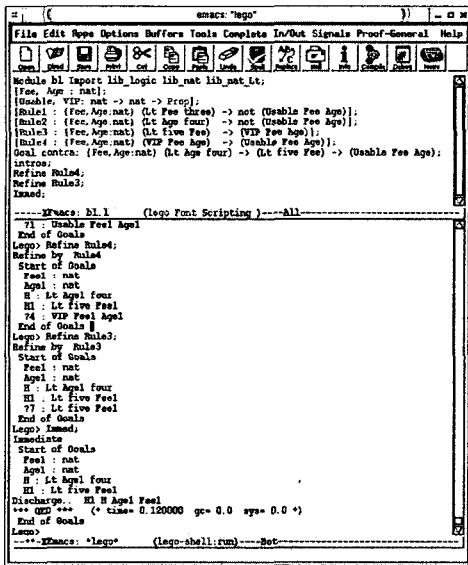


그림 2. LEGO 에서 Goal formula 증명

5. 결론

본 논문에서는 DRM 시스템을 EJB 아키텍처 상에서 구현한 후에, 비즈니스 룰의 안전성 획득의 필요성을 인식하고 정형적으로 기술하고 검증하는 방법을 기술하였다. 룰 자체가 자연어로 기술되고 언젠가는 모순을 유발할 가능성이 높기 때문에 먼저 명확한 논리식으로 표현하고 LEGO라는 정형기법 도구를 통해 검증하였다.

비즈니스 룰은 문자 그대로 비즈니스 행동을 제어하거

나, 영향을 준다. 또한 비즈니스를 수행하면서 무엇이 가능하고 바람직한지 혹은 어떤 것이 그렇지 않을지를 알려준다.

앞으로 보다 많은 콘텐츠에 대해 DRM 기술이 적용될 것이고 각각의 콘텐츠가 정당한 가치를 보장받아야 할 것이다. 비즈니스 룰 자체가 이러한 기능을 담당함은 물론이고 비즈니스 룰이 다양한 사용자와 다양한 콘텐츠에 대하여 유연성(flexibility)을 획득하게 됨으로써 DRM 기술을 보편화시키고 확산시킬 수 있을 것이다. LEGO 툴을 이용하여 충돌이 발생하는 상황을 찾아내어 증명할 수 있었다. 향후 연구과제로는 이 모순되는 경우를 사전에 막기 위해 어떠한 방식으로 비즈니스 룰을 적용하고 정형기법 적으로 검증할 수 있는지를 연구해 보고자 한다.

6. 참고문헌

- [1] Intertrust Inc, XML Object Builder's Guide, Commerce 1.2.1
- [2] IBM Research Report,
<http://www.research.ibm.com/people/g/grosfof>
- [3] Intertrust Inc, metatruster utility,
<http://www.Intertrust.com/main/metatruster/whatsdrm.html>
- [4] Dirk van Dalen, Logic and Structure, 1991
- [5] The LEGO Proof Assistant,
<http://www.dcs.ed.ac.uk/home/lego/>
- [6] Contentguard Inc, XrML :
Extensible rights Markup Language,
<http://www.xrml.org>