

비즈니스 규칙 시스템과 객체 모델

강상승, 손주찬, 함호상
한국전자통신연구원 인터넷컴퓨팅연구부
e-mail : kss@etri.re.kr

Business Rule System and Object Model

Sang-Seung Kang, Joo-Chan Sohn, Ho-Sang Ham
Internet Computing Department, ETRI

요 약

비즈니스 규칙은 비즈니스의 어떤 모습을 정의하거나 제한사항을 나타내는 문장으로 비즈니스 정책과 로직을 기술한 것이다. 본 논문에서는 비즈니스 규칙과 비즈니스 규칙 시스템을 분석하고, 분석한 내용을 기반으로 비즈니스 규칙 표현과 시스템에서의 요구조건과 고려사항을 제시하며, 새로운 비즈니스 규칙 시스템을 설계한다. 또한 비즈니스 규칙 시스템을 위한 기본 모형인 비즈니스 규칙 객체 모델을 제안한다.

1. 서론

일반적으로 비즈니스 규칙은 기업의 비즈니스를 하는 방법에 관한 문장이다. 그것은 고객을 만족시키기 위한 방편으로 정책, 절차나 다른 제한사항을 반영하고, 자원을 잘 활용하고, 법률이나 비즈니스 관례에 맞게 하는 것이다.

비즈니스 규칙은 비즈니스를 성공하기 위한 전략적인 중요성이라 추정할 수 있다. 이것은 정보 기술자나 소프트웨어 엔지니어에 의해서가 아니라 비즈니스 관련 사람들에 의해 정형화되고 분석되는 것이라는 것을 의미한다. 즉 비용, 품질, 책임 및 좋은 서비스에 관한 결정은 시스템 디자이너나 프로그래머에게 위임되는 것이 아니라 비즈니스 관련 담당자가 할 수 있어야 한다.

기존 시스템의 경우, 비즈니스 규칙이 시스템 내에 비즈니스 로직으로서 응용 프로그램 코드 내에 존재되어 있고, 프로그램 코드의 알고리즘으로서 표현되어 있는 비즈니스 규칙의 제어 로직 코딩량이 방대하며, 프로그래밍 언어에 의존하여 등록 및 변경을 수행해야 하며, 전담 시스템 담당자가 필요하다.

반면, 비즈니스 규칙 시스템의 경우, 시스템 내에 비즈니스 로직으로서 산재해 있는 비즈니스 규칙을 응용 프로그램 코드에서 분리하여 한 곳에서 집중 관리할 수 있고, 프로그램 코드의 알고리즘으로서 표현되어 있는 비즈니스 규칙의 제어를 독립 엔진에 의한 자동화가 가능하며, 프로그래밍 언어에 의존하지 않고

업무 용어 기반의 비프로그래밍 방식으로 등록 및 변경을 수행할 수 있는 장점을 가진다. 따라서 업무 절차와 제어 코드가 혼재되어 있어 유지보수가 어려운 기존 시스템에 비하여 비즈니스 규칙 시스템은 비즈니스 규칙의 독립 관리 및 자동 제어가 가능하여 업무의 효율성을 높이고 개발 및 유지보수가 유용하다.

본 논문에서는 비즈니스 규칙과 비즈니스 규칙 시스템을 분석하고, 분석한 내용을 기반으로 비즈니스 규칙 표현과 시스템에서의 요구조건과 고려사항을 제시하며, 새로운 시스템 구조를 설계한다. 또한 비즈니스 규칙 시스템을 위한 기본 모형인 비즈니스 규칙 객체 모델을 제안한다.

본 논문의 구성은 다음과 같다. 먼저 2 장에서는 비즈니스 규칙과 비즈니스 규칙 시스템에 대한 기반이 되는 내용을 분석하여 기술하며, 3 장에서는 비즈니스 규칙 표현과 비즈니스 규칙 시스템에서의 요구조건과 고려사항을 분석한다. 4 장에서는 새로운 비즈니스 규칙 시스템을 설계하고, 설계된 구조에서의 비즈니스 규칙 객체 모델을 제안하고 세부 기능을 명세한다. 마지막으로 5 장에서는 결론을 맺는다.

2. 비즈니스 규칙 및 시스템

비즈니스 규칙이란 용어뿐만 아니라 규칙이란 용어에 대한 산업 표준 정의는 없으나, Business Rules Group 에 따르면 비즈니스의 어떤 모습을 정의하거나

제한점을 기술한 문장이라고 정의하고 있다. 즉 비즈니스 정책과 로직을 기술한 것이라고 표현할 수 있다. 이러한 비즈니스 규칙은 비즈니스 용어의 정의, 데이터 무결성 제한조건, 수학적 및 기능적 도출, 논리적 추론, 처리 순서, 비즈니스에 관한 사실들간의 관계 등의 여러 가지 형태가 될 수 있다.

또한 비즈니스 규칙은 항, 사실, 규칙의 세가지 주요 카테고리로 분류할 수 있다. 항은 정의에 일치하는 명사 혹은 명사구이며, 사실은 비즈니스 관련 내용으로 전치사, 동사구를 포함한 항들로 연결된 문장이다. 항과 사실은 규칙 후면의 시맨틱 이라 할 수 있으며, 논리적 데이터 모델, 물리적 데이터베이스, 비즈니스 객체 모델의 기초가 된다. 규칙은 로직을 정보 값에 적용시킨 서술문으로, 새로운 정보의 발견 혹은 행위에 관한 결정 중의 하나의 결과로서 일어난다. 따라서 규칙은 입력으로 정보를, 출력으로 정보 혹은 행위를 사용하는 실행가능한 로직이 될 것이다.

비즈니스 규칙을 발견, 문서화, 모델링, 구현하도록 특정 작업을 반복 제어하는 것을 비즈니스 규칙 라이프 사이클이라 하며, 전략, 운영, 조직 요구조건 등의 다양한 측면에서의 접근을 포함한다. 다음은 전형적인 비즈니스 규칙 라이프 사이클이다.

- 목적 간결문에 기업 직무 기술
- 가설을 세우고 외부 요소에 관한 데이터를 모음
- 기업의 강점과 약점 평가
- 기업 직무와 연결된 목적, 목표, 방책 수립
- 목표와 목적에 부합하도록 전략 및 운영 계획 수립
- 직무 절차를 설계하고 통합
- 기업 프로세스와 결정 처리 지원 정보 시스템 구현
- 목표와 목적에 부합하도록 보장을 위한 성능 평가
- 목표, 목적, 프로세스 및 방책 등을 재평가 및 변경

이러한 단계에 따라 비즈니스 규칙은 발견, 문서화, 모델링되며, 비즈니스 요구조건과 비즈니스 규칙은 목표, 전략, 목적, 작업, 중대한 성공요소, 성능 등의 실제 용어로 정의된다. 비즈니스 요구조건이 정의됨에 따라 논리적 모델을 사용하여 개발된다.

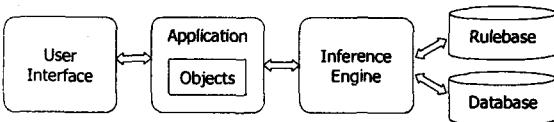


그림 1. 비즈니스 규칙 시스템

전형적인 비즈니스 규칙 시스템은 그림 1 과 같이, 추론엔진, 규칙베이스, 사용자 인터페이스로 구성되며, 데이터베이스 시스템과 데이터 교환이 필연적으로 발생하기 때문에 데이터베이스 시스템을 포함할 수 있다.

여기서 추론 엔진은 내부적으로 규칙을 이용하여 문제를 해결하는 추론 알고리즘에 따른 프로그램과

처리 과정에서 필요한 정보를 메모리에 저장하는 작업 영역으로 구성된다. 이러한 비즈니스 규칙을 위한 추론 엔진은 비즈니스 규칙베이스를 바탕으로 규칙의 검색과 실행을 자동화하며, 정방향 추론 및 역방향 추론과 같은 다양한 추론 기법을 제공하고, 강력한 추론 알고리즘인 Rete 알고리즘을 지원해야 한다. 또한 제어 로직 및 제어 객체의 기능을 자동화하고 업무 로직의 구축을 대폭 간략화하도록 규칙 적용을 자동화할 수 있어야 한다.

3. 요구 분석 및 고려사항

비즈니스 규칙은 기업의 비즈니스 목적으로부터 발생하는 요구조건들이다. 먼저 비즈니스 규칙 표현과 관련한 요구조건 및 고려사항을 분석하면 다음과 같다.

- 다양한 규칙 표현을 지원하고 효율적으로 처리할 수 있어야 하며, 응용 프로그램에서 폭넓게 사용될 수 있는 실용적인 측면도 고려해야 한다.
- 관련 규칙 표준들로부터 규칙을 표현 가능하고, 웹 표준들과 호환될 수 있어야 한다.
- 규칙은 지식 기반 시스템에서 중심 요소이므로, 서로 다른 규칙 기술로 구현된 시스템 간의 지식 교환을 허용해야 한다.
- 응용 도메인에 따라서 규칙은 다양한 종류의 정보를 운영 가능해야 하며, 필요에 따라 유효기간 등의 방법에 의해 제한될 수도 있다.
- 이질적으로 표현된 지식에 대해 그 표현 양식에 상관없이 일관성을 유지하도록 식별자를 필요로 한다.

또한 표현된 비즈니스 규칙을 운영 및 관리하기 위한 시스템에서의 요구조건 및 고려사항을 분석하면 다음과 같다.

- 비즈니스 규칙 시스템에서 추론 엔진이 사용할 수 있는 전형적인 형태의 규칙으로 변환할 수 있어야 하며, 변환된 규칙은 규칙베이스에 저장되며, 저장된 규칙은 다양한 응용 시스템에서 활용될 수 있어야 한다.
- 분산된 규칙 언어의 연결성을 통해 하나의 지식을 수정하면 다양한 형태의 지식이 수정되어 항상 동일한 지식을 유지하도록 한다. 이는 지식의 관리를 매우 용이하게 할 수 있다.
- 지식의 공유 및 교환을 위해서 각종 시스템 간의 상호운용성을 지원해야 한다. 마크업 언어가 갖는 장점을 기반으로 운영 시스템이나 하드웨어의 특성과 관계없이 규칙을 저장 및 교환할 수 있도록 한다.

규칙 표현 모델을 만들기 위해서는 두가지 프로세스가 필요한데, 하나는 관련된 속성들과 함께 규칙 클래스를 정의하는 것이고, 다른 하나는 이들 클래스 간의 관계를 설정하는 것이다. 생성된 코드는 자동 검사 프로세서에서 사용되기 전에 분류되며, 분류 조항들은 규칙 모델인 사실로서 지식베이스에 저장된다. 따라서

비즈니스 규칙 객체는 사실과 규칙 전제조건으로부터 생성되고 지식베이스에 저장되며, 이들은 규칙 언어로 기술된다.

4. 설계 및 고찰

비즈니스 규칙 시스템의 기본 구성은 그림 2 와 같다. 사용자 인터페이스를 제공하는 규칙 편집기, 추론 기법을 이용하여 규칙 적용을 하는 추론 엔진, 규칙 관련 정보를 저장하는 규칙 저장소, 규칙 관련 클래스 정보를 표현하는 ROM 을 제어하는 ROM 핸들러, 규칙 편집기와 ROM 핸들러 사이의 인터페이스 역할을 수행하는 규칙 어댑터, 규칙 저장소를 제어하는 규칙 저장소 핸들러 등으로 구성된다.

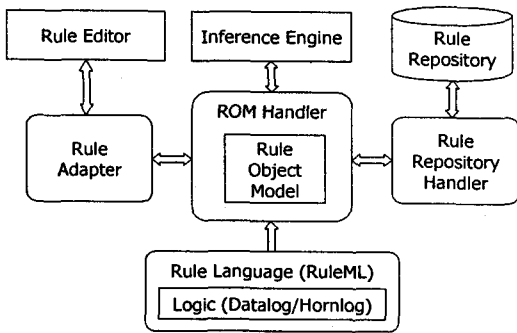


그림 2. 시스템 구성도

- 규칙 편집기(Rule Editor): 비즈니스 규칙 시스템을 구성하는 응용 프로그램의 일종으로 사용자가 규칙을 쉽고 편리하게 저작할 수 있는 GUI 환경을 제공한다. 비즈니스 규칙 관련 다른 응용 프로그램들이 위치할 수도 있다.

- 규칙 어댑터(Rule Adapter): 규칙 편집기로부터 정해진 구조로 기술된 규칙 프로젝트 파일을 입력받아 파싱 기능을 수행하며, 규칙 객체 모델과 외부 응용 프로그램 간의 인터페이스를 정의한다.

- 규칙 객체 모델(Rule Object Model): 규칙 관련 클래스 정보(규칙 프로젝트 내용)를 표현하는 객체 모델이다.

- 규칙 객체 모델 핸들러(ROM Handler): 규칙 마크업 언어로 작성된 문서의 파싱된 결과로부터 규칙 객체 모델을 생성하고 제어하며, 규칙 어댑터와 규칙 객체 모델 간의 인터페이스를 정의한다.

- 규칙 언어(Rule Language): 규칙의 표현을 위한 언어로, XML 기반의 규칙 마크업 언어인 RuleML 을 기반으로 하여 수정 및 확장하여 사용한다.

- 추론 엔진(Inference Engine): 추론 기법을 이용하여 규칙 적용을 자동화 수행하여 결론에 도달하도록 하는 메커니즘 프로그램이다.

- 규칙 저장소 핸들러(Rule Repository Handler): 규칙 저장소에 연결하여 각종 데이터를 저장 및 추출하는 컴포넌트로서, 규칙 저장소에 저장된 데이터에 용이하게 접근할 수 있는 인터페이스를 제공한다.

- 규칙 저장소(Rule Repository): 비즈니스 규칙 표현의 유연성을 지원하고 규칙 처리에 필요한 각종 정보를 저장하고 있는 저장소이다. 시스템 내에 비즈니스 로직으로서 산재해 있는 비즈니스 규칙을 응용 프로그램 코드에서 분리하여 한 곳에서 집중 관리하게 된다.

비즈니스 규칙을 저장하는 규칙 저장소는 일반적으로 이질 시스템으로 비즈니스 규칙을 널리 배포할 수 있는 구조로 되는데, 그것은 다수의 비즈니스 규칙 정의와 다중 언어에 대한 연관된 규칙 로직에 대한 지원을 포함해야 한다. 이러한 비즈니스 규칙 저장소는 비즈니스 규칙의 실행 순서를 평가하기 위해 비즈니스 규칙 응용 프로그램 인터페이스에 필요한 모든 속성들을 저장하게 되는데, 관련된 비즈니스 규칙 속성들은 다음과 같다.

- UID: 비즈니스 규칙을 식별하는 아이디
- NAME: 비즈니스 규칙의 명칭
- VERSION: 버전 정보를 나타내며 동일한 규칙의 다른 버전을 식별해줌. NAME 과 VERSION 속성을 이용해 규칙의 유일성 보장을 고려할 수 있음
- ACTION: 규칙이 저장소 내에서 연결되어 있는 비즈니스 액션의 명칭
- PROCEDURE: 규칙 실행을 호출하기 위한 물리적인 절차 혹은 프로그램의 명칭
- START_DATE: 비즈니스 규칙이 실행될 기간의 첫째 날 식별 필드
- END_DATE: 비즈니스 규칙이 실행될 기간의 마지막 날 식별 필드
- PRIORITY: 규칙이 다른 비즈니스 규칙과 관련되어 실행될 때 사용되는 우선순위를 식별하기 위한 숫자 필드로서, 낮은 숫자가 먼저 실행됨
- ACCESS_LEVEL: 사용자의 다른 클래스들에 대한 접근 권한을 식별하는 필드
- SCOPE: 비즈니스 규칙의 배포 단위를 식별하는 필드
- SUCCESS_ACTION: 성공적인 실행에 따라 다음 이벤트를 식별하는 필드
- FAILURE_ACTION: 실행 실패에 따라 다음 이벤트를 식별하는 필드

규칙 객체 모델 핸들러를 포함하는 비즈니스 규칙 객체 모델 기능과 규칙 어댑터를 포함하는 비즈니스 규칙 접근 컴포넌트 기능에 대한 세부 기능 설계 명세는 다음과 같다.

먼저 비즈니스 규칙 객체 모델 기능은, 규칙 마크업 언어로 작성된 문서의 파싱된 결과로부터 규칙 관련 클래스 정보를 표현하는 규칙 객체 모델을 생성하고 제어하기 위한 인터페이스를 제공하는 규칙 객체 모델 기능과 사실, 규칙, 목표 등의 논리 구문을 생성하거나 클래스, 객체 등의 프레임 지식을 표현하는 데 필요한 모델을 생성하기 위한 인터페이스 정의를 제공하는 논리 구문 표현 모델 기능으로 구성되며, 다음은 세부 기능 명세이다.

- 규칙 객체 모델 생성 기능: 규칙 어댑터로부터 파싱된 결과를 받아 규칙 객체 모델을 생성한다.
- 규칙 객체 모델 제어 기능: 규칙 관련 정보 파일의 기능 제어 및 규칙 접근 컴포넌트와의 중간 매개체 기능을 수행한다.
- Datalog 및 Hornlog 구문 표현 기능: Datalog, Hornlog를 포함하여 확장한 규칙 객체 모델 인터페이스 정의를 지원하며 프레임에 대한 지식 표현도 지원한다.
- 비즈니스 객체 모델 생성 기능: 외부 비즈니스 객체 데이터 소스에 정의된 규칙 관련 정보로부터 비즈니스 객체 모델을 생성한다.
- 비즈니스 객체 모델 제어 기능: 비즈니스 객체 모델의 기능을 제어하고 중간 매개체 기능을 수행한다.

다음으로 비즈니스 규칙 접근 컴포넌트 기능은, 다른 비즈니스 객체 데이터 소스에 정의된 규칙 관련 정보로부터 비즈니스 객체 모델을 생성할 수 있도록 импорт 기능을 수행하는 비즈니스 객체 데이터 소스 импорт 기능과, 규칙 관련 정보를 표현하기 위해 규칙 마크업 언어로 작성된 문서의 오류 검출 및 파싱 기능을 수행하는 규칙 언어 구문 검사 기능과, 규칙 객체 모델과 외부 컴포넌트 간의 인터페이스를 정의하는 규칙 접근 어댑터 기능으로 구성되며, 다음은 세부 기능 명세이다.

- 데이터 소스 импорт 어댑터 기능: 외부 비즈니스 객체 데이터 소스에 정의된 규칙 관련 정보로부터 비즈니스 객체 모델을 생성할 수 있도록 하기 위한 импорт 기능을 수행한다.
- 클래스 매핑 기능: 추출한 규칙 관련 정보들과 객체 간의 매핑 관계를 설정한다.
- 구문 예러 검사 기능: 규칙 관련 정보를 표현하기 위해 규칙 마크업 언어로 작성된 문서의 구문상 오류를 검출하고 디스플레이 한다.
- 규칙 언어 파서 기능: 정해진 구조로 기술된 입력 파일로부터 파싱 기능을 수행한다. 파싱 결과는 규칙 객체 모델 생성 및 제어 기능에서 활용된다.
- 규칙 접근 어댑터 기능: 규칙 객체 모델과 외부 컴포넌트 간의 인터페이스를 정의한다.

5. 결론

비즈니스 규칙 시스템을 통해 비즈니스 규칙을 응용 프로그램 코드에서 분리하여 한 곳에서 집중 관리

함으로써, 비즈니스 규칙의 독립 관리 및 자동 제어가 가능하여 업무의 효율성을 높이고 개발 및 유지보수가 유용하다.

본 논문에서는 비즈니스 규칙 표현과 시스템에서의 요구조건과 고려사항을 제시하고, 새로운 시스템 구조를 설계하였으며, 비즈니스 규칙 시스템을 위한 기본 모형인 비즈니스 규칙 객체 모델을 제안하였다. 또한 본 논문에서는 논리 구문의 표현을 위해 양의 리터럴이 많아 한 개만 있는 Horn 로직을 기반으로 하였으며, 비즈니스 규칙 시스템의 활용성을 높이기 위해 표현력을 확장하고, 프레임에 대한 지식 표현도 고려하였다. 향후 자바 객체 위주의 데이터 소스 импорт 기능을 데이터베이스 및 XML 스키마와 은톨로지 등으로 확대하고, 새로운 규칙 언어에 대한 연구도 진행할 예정이다.

참고문헌

- [1] Barbara von Halle, *Business Rules Applied*, John Wiley & Sons, 2002.
- [2] Ronald G. Ross, *The Business Rule Book*, Second Edition, Business Rule Solutions, LLC, 1997.
- [3] Rosca, D., et al., "A Decision Making Methodology in Support of the Business Rules Lifecycle," *Proceedings of the Third IEEE International Symposium on Requirements Engineering*, pp.236-246, Jan. 1997.
- [4] Perkins, A., "Business Rules = Meta-Data," *Proceedings of 34th International Conference on Technology of Object-Oriented Languages and Systems*, pp. 285-294, 2000.
- [5] ILOG, "ILOG JRules: Technical White Paper," October 2002.
- [6] FairIsaac, "Fair, Isaac Blaze AdvisorTM: How it Works," 2003.
- [7] RuleML Homepage, <http://www.dfki.uni-kl.de/ruleml/>
- [8] Business Rules Group Homepage, <http://www.businessrulesgroup.org/>