

# Drools 기반의 메디컬 Expert 시스템 아키텍처와 프로세스 연구

장원용, 최유나, 양성수, 최은미\*

국민대학교 소프트웨어학부

\*e-mail:emchoi@kookmin.ac.kr (교신저자)

## Studying on a Medical Expert System Architecture and Process based on Drools

Wonyong Jang, Yuna Choi, Seongsoo Yang, Eunmi Choi\*

College of Computer Science, Kookmin University

### 요 약

본 논문은 병원에서 환자가 내원할 때 의사에게 환자 진단과 처방을 지원을 할 수 있는 메디컬 시스템에 대한 연구를 하였다. 메디컬 프로세스를 고려한 환자의 증상에 대한 파악과 진단, 처방에 이르기까지의 지원할 수 있는 병원의 증상 진단 지원 Expert System을 연구한다. 본 Expert System을 구축하기 위하여 오픈소스 프로젝트인 Drools를 사용하였다. 본 시스템은 시스템 운영 중, 병원에서 사용하는 증상 진단 지원 rule들을 용이하게 추가하며 사용할 수 있도록 하며, 사용한 Drools의 호환성 해결과 다양한 기능성을 사용하기 위해 Maven을 이용한 증상 진단 지원 Expert System을 설계하였다. 이 시스템의 전체 구성 시스템 아키텍처뿐만 아니라, 환자가 증상을 입력한 이후의 환자 증상, 진단, 처방에 이르기까지의 병원에서 일어나는 프로세스를 구현 및 연구를 하였다.

### 1. 서론

본 논문은 환자가 진료를 받기위하여 병원에 내원할 때 의사의 진단과 처방시간을 효율적이고 정확한 진단을 위한 병원의 증상 진단 지원 Expert System을 연구하였다. 병원에서는 특정 과목의 진료나 특정 시간 등에 환자수가 증가하여 환자가 진단을 받기위한 대기시간이 길어진다. 또한 병원은 때때로 환자의 증상에 대한 진단 실수를 한다. 이에 따라 본 논문은 Java기반의 라이브러리인 Drools의 rule과 컴파일러인 Maven을 활용하고자 한다.

병원에서 설문지 등으로 환자들의 증상정보를 사전에 얻고 이를 체계적으로 진단과 처방을 내리기 위해서는 Expert System에서 사용하는 Drools의 rule이 중요하게 사용된다. Drools가 rule의 삭제 또는 추가가 용이하게 적용되므로 본 논문에서는 오픈소스 프로젝트인 Drools를 사용하였다. 이를 사용함으로써 병원에서 환자의 증상을 판단하는 의사의 개인적인 견해는 개별적으로 작성하여 후에 환자의 증상 진단에 종합적으로 내용이 나타나도록 처리한다. 또한, 의사가 증상에 대한 진단을 처방하고자 하는 내용도 증상 진단 지원 Expert System이 Drools의 rule에 의해 의사에게 처방의 방향을 제시해 준다. 이로써 연구한 증상 진단 지원 Expert System으로 환자에게는 진단과 처방시간 단축을 의사에게는 진단과 처방 편의성을 제공하고자 한다.

다음 장은 관련 연구를 조사하며, 3장에서는 메디컬

Expert System 의 프로세스와 시스템 설계, 4장에서는 구현 부분을 다루며, 마지막 장에서 결론을 나타낸다.

### 2. 관련 연구

#### 2.1 Drools

Drools는 BRMS(Business Rule Management System)[1]로써 Java Rule Engine API(JSR-94)를 지원한다[2]. Drools는 비즈니스 규칙 관리 및 복잡한 이벤트 처리가 빠르고, 신뢰할 수 있는 평가를 가능하게 하는 전방 연쇄(Forward-chaining) 및 후방 연쇄(Reverse-chaining) 기반 규칙 엔진을 모두 갖추고 있다. 규칙은 DRL 언어로 되어 있고[4], 아래 그림 1과 같은 형식을 갖는다.

```
declare Bean
    delay    : String = "30s"
    period   : long = 60000
end

rule "Expression timer"
    timer( expr: $d, $p )
when
    Bean( $d : delay, $p : period )
then
end
```

그림 1. Drools 의 DRL 규칙 사례

규칙은 직접 호출되어 사용될 수 없고, 반드시 메소드를 통해 실행되어, 패턴을 따라 매칭 된다. 규칙 엔진은 기본

빌딩 블록을 생성하여 전문가의 의사 결정 능력을 에뮬레이트하는 인공지능을 구축할 수 있다. 규칙 엔진은 로직이 수시로 변경되는 경우에 사용이 용이하다. 규칙 엔진은 별도로 저장해둔 룰을 프로그램에서 쓸 수 있도록 해 주는 기능을 제공하고, 프로그램에 있던 로직이 규칙 엔진으로 옮겨가고, 프로그램에서는 규칙 엔진을 통해 규칙을 실행하기만 하면 된다.

또한 규칙은 생산 메모리에, 팩트는 작업 메모리에 저장된다. 분석 세션이 진행되는 동안 생산 메모리는 변경되지 않고, 작업 메모리는 내용을 변경할 수 있다[3]. 규칙 엔진을 통해 규칙을 추가, 삭제, 변경하여도 작업 메모리의 변경 후 규칙 엔진이 트리거 되어 어떤 규칙이 참값이 되는지 결정되기 때문에 실행 순서를 관리하여 충돌 해결 전략만 세워준다면 프로그램을 변경하지 않아도 된다. 이러한 특성으로 수시로 로직이 변경되는 곳에 규칙 엔진을 적용함으로써 개발 시간을 단축하고, 새로운 규칙을 도입하기 수월한 장점이 있다[1].

### 3. 메디컬 Expert System 프로세스 정립

본 장에서는 메디컬 Expert System 의 프로세스를 정립을 하여 의사의 진료를 지원할 수 있는 시스템을 설계하고자 한다. 일반 환자는 병원에 방문하여 전문의사의 진료를 받기 전까지 설문지를 작성함으로써, 의사의 진료 이전에 환자에 대한 정보를 병원에서 취득할 수 있다. 이 정보에 근거하여, 설문지에 대한 데이터를 가지고 병원에서 미리 정해 놓은 룰(rule)들 중 조건에 충족하는 진단과 처방이 처리되어 나오도록 한다. 의사들에게는 그 결과를 확인하도록 하고, 보완할 부분이 있다면 그 부분만 수정할 수 있도록 하여, 보다 더 정확한 처방과 진단에 대한 프로세스를 제공할 수 있다. 의사의 진단과 처방이 이러한 룰(rule)들을 기반으로 체계적으로 진단에 대해 지원을 할 수 있다.

#### 3.1 진단 지원 Expert System 의 구성

본 논문에서 제안된 처방전 관리 시스템은 그림 2과 같이 구성되었다. 시스템은 크게 3부분으로 구성된다. 환자가 작성한 설문지의 이미지 데이터를 텍스트 데이터로 변환해주는 부분, 의사가 환자가 작성한 설문지를 확인하고 판단하는 JavaFX 기반의 GUI 부분, 그리고 GUI를 이용하여 룰(rule)들 중 조건에 만족하는 결과 값을 도출해내는 Drools 부분으로 이루어진다.

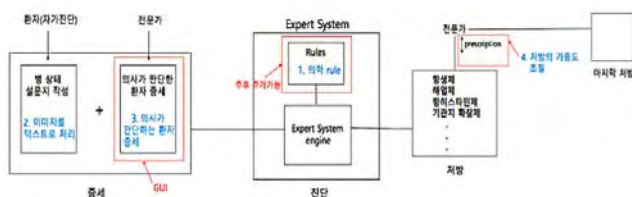


그림 2. 메디컬 Expert System 구성 개념도

아래의 그림 3은 본 시스템의 전체 구성도이다. 먼저 Patient 클래스는 환자 한 명에 대한 환자번호, 이름, 성별, 나이, 몸무게 등의 정보를 가지고 있으며 그 Patient 클래스의 배열 전체를 Patient Factory 클래스가 가지고 있게 된다. Expert Controller는 Patient Factory를 가지고 GUI를 이용하여 환자에 대한 증상을 확인하고 Drools를 이용하여 조건에 맞는 룰(rule)을 확인하게 된다. 그러면 Drools는 Patient Diagnosis 클래스에 검사한 환자에 대한 정보와 진단 내용, 처방 등을 기록하며 동시에 GUI 화면에 보여주고 의사가 확인 할 수 있게 한다. 또한 환자 한명은 하나의 병이 아닌 여러 종류의 질병을 가지고 있을 수 있으므로 1:n 관계로 Patient Sickness 클래스를 가지고 각각을 진단을 받게 된다. 마지막으로 의사는 확인한 진단과 처방 내용을 바탕으로 하여서 좀 더 환자에게 맞는 처방을 내려 줄 수 있게 된다.

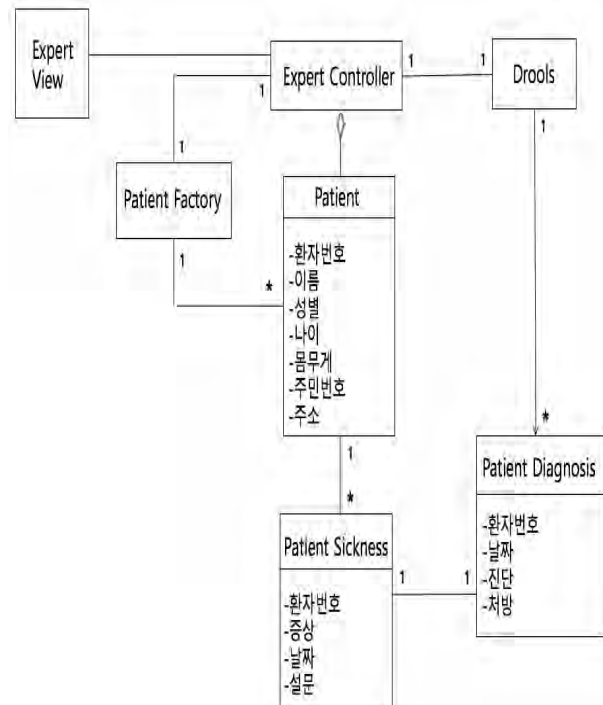


그림 3. 시스템 아키텍처 구성도

#### 3.2 Maven pom.xml 설계 와 구현

본 논문에서 제안된 시스템은 Maven을 이용하여 좀 더 모듈간의 의존성을 효율적으로 관리하였다. 아래의 그림 4는 프로젝트에서 사용된 Maven dependency 중 ecj는 drl 컴파일러가 JavaFX의 Map 이나 HashMap 등을 컴파일하지 못하는 문제를 발견하였다.

java.lang.NoClassDefFoundError 이러한 에러를 해결하기 위해 사용되었고 또 다른 dependency인 controlsfx는 조금 더 유용하고 높은 품질의 UI control들을 구현하기 위해서 사용되었다. 예를 들면, controlsfx 사용 후에는 TextFields.bindAutoCompletion 와 같이 쓸 수 있어서 본 논문에서는 여러 증상 리스트들에서 검색기능을 구현할 수 있었다.

```

<dependencies>
<dependency>
<groupId>org.eclipse.jdt.core.compiler</groupId>
<artifactId>ecj</artifactId>
<version>4.5.1</version>
</dependency>
<dependency>
<groupId>org.controlsfx</groupId>
<artifactId>controlsfx</artifactId>
<version>8.40.13</version>
</dependency>
</dependencies>

```

그림 4. Maven pom.xml 의 dependency

### 3.3 Drools의 진단 룰(rule) 정의

본 논문에서 제안된 시스템은 Drools를 이용하여 룰을 정의한다. 아래 그림 5는 Drools의 drl파일에서 여러 룰(rule)들 중 하나의 소스코드를 이해하기 쉽게 그림으로 표현했다. Drools 내부에서는 환자에 대한 객체가 들어오게 되면 미리 정해 놓은 rule들을 검사하게 된다. 이때 rule의 이름은 'Number21 : 감기증상'이고 when에는 환자에 대한 증상들이 있다. 환자가 이 조건들을 만족 한다면 then 이후에 있는 진단과 처방을 찾아주게 되는데, 이는 결과 값을 Medical Expert System의 GUI 화면에 보여지게 된다.

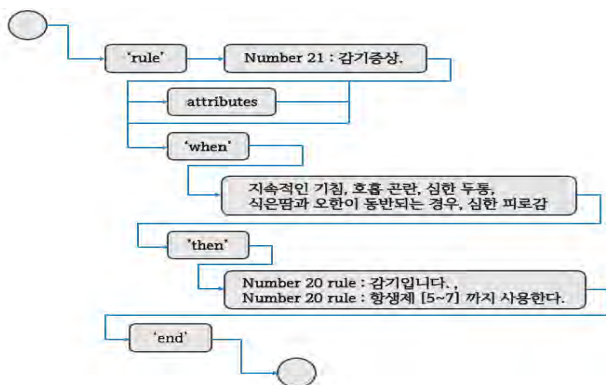


그림 5. Drools 의 감기증상 rule

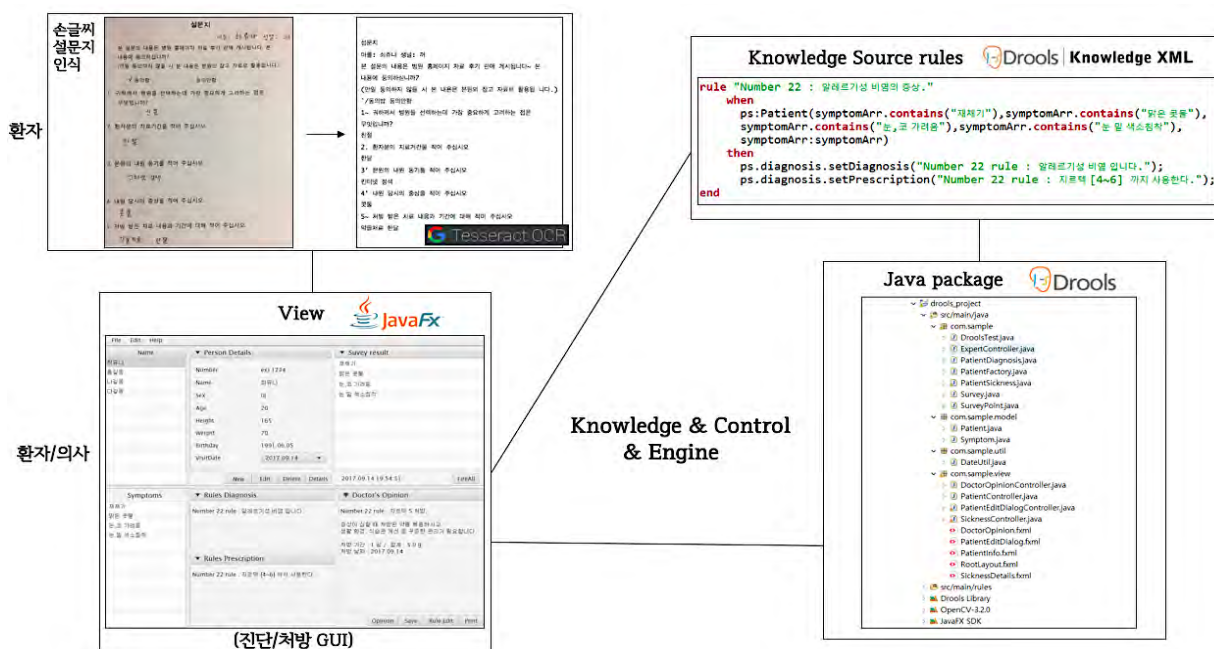


그림 6. 메디컬 Expert System의 전체 구성도

## 4. 메디컬 Expert System 구현과 결과

본 논문에서 연구한 시스템 아키텍처를 기반으로 그림6과 같이 Medical Expert System을 개발 및 구현하였다. 먼저 의사 진료 이전에 환자의 증상을 확인하기 위한 프로세스로써, Tesseract-ocr 엔진을 이용하여 환자가 작성한 설문지를 디지털 형식의 텍스트 데이터로 변환 하여 가져온다. 환자가 손으로 작성한 글씨와 해당 증상에 체크한 설문지를 자바에서 사용할 수 있는 OCR 오픈소스 라이브러리인 Tess4J를 사용하여 설문지 이미지를 텍스트화한 인풋 파일을 생성한다.

그 후 이와 같은 데이터를 입력받아서, Drools Engine이 미리 정립시켜 둔 룰(rule)들에 적용시켜서 검사를 진행하게 된다. 그러면 조건에 맞는 진단과 그에 맞는 처방이 GUI 화면에 보여 지게 된다. 이 룰을 가지고 진단을 함으로서 의사에게 환자에 맞는 진단과 처방에 대한 기준을 세우는데 참고사항이 된다. 이 진단과 처방 결과는 전문 의사들의 도메인 지식을 기반으로 정립이 된다. 그림 6의 우측 상단에 있는 Knowledge Source Rule에 정형화된 룰들을 볼 수가 있다.

우측 하단에는 Drools에 있는 자바 Package Library를 나타내고 있으며, 본 연구에서 개발되고 구현된 Maven, JavaFX, Drools의 Expert System의 모든 package 구조를 보여주고 있다.

마지막으로 의사는 메디컬 Expert System에 나타난 진단과 처방을 볼 수 있으며, 이 내용을 기준으로 환자에 더 적절한 처방을 신속히 처리하거나 수정에 대한 작업을 할 수 있다. 또한 의사는 환자의 상태에 따라서 처방의 가중여부를 결정할 수 있는 융통성을 적용할 수 있게 된다.

## 5. 결론

본 논문은 병원에서 환자가 진료를 받을 때 시간을 효율적으로 처리하기 위해 구글에서 제공하는 텍스트 인식하는 Tesseract-ocr 엔진과 오픈소스 Expert System 인 Drools, 용이한 GUI 구성이 가능한 JavaFx를 사용하여 메디컬 Expert System을 연구하였다.

그리고 환자의 상태를 좀 더 정확하고 세분화하여 관리하기 위해 Expert System에 각 병원의 룰들을 추가 가능하도록 하였다. 이로써 증상 진단 지원 Expert System의 사용이 의사의 진단에 대한 시간 단축과 정확성을 높이는 것이 가능하게 되었다. 본 논문의 증상 진단 지원 Expert System은 Drools의 장점을 최대한 활용하여 Expert System의 추가적인 변경 없이 룰을 추가 할 수 있기 때문에 추후 확립되는 다양한 룰에 의해 진단의 신뢰도를 높이고 System의 확장성을 지원할 수 있다. 그리고 Drools와 Java의 컴파일 문제를 해결하기 위해 Maven을 사용함으로써 Java의 다양한 버전에서 Drools를 사용가능하게 되었고, 환자가 본인의 증상을 추가하기 위한 검색기능이 가능하게 됨으로서 다양한 증상들에 대한 선택을 효율적으로 가능하게 하였다. 사용자의 편의를 도모한 GUI는 JavaFX를 이용하여 인터페이스를 구성함으로써 기존에 Java로 인터페이스를 구성하는데 쓰이던 Swing의 다양한 단점을 극복함으로써, 본 논문에서는 Medical Expert System을 이용하여 환자의 편의를 도모하고 의사의 진단과 처방의 신뢰를 높일 수 있었다.

## 참고문헌

- [1] <http://www.jboss.org/jboss>.
- [2] 박종문, 안형배, 이명준, “JCAF과 DROOLS를 이용한 상황 인식 어플리케이션 생성 도구”(A Toolkit for Generating Context-Aware Applications with JCAF and Drools), 한국정보과학회 학술발표논문집, Vol. 39, No. 6, pp 254-256, 2012.
- [3] Ei Thu, et al., “Model driven development of mobile applications using drools knowledge-based rule”, 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA) June 2017, pp 179-185 2017.
- [4] Jan Ruzicka, “Integrating DROOLS and R software for intelligent map system”, Geoinformatics FCE CTU, Vol 7, pp 85-92, 2012.