

Drools를 이용한 Nginx와 Ajax기반의 Medical Expert System 개발

장원용, 김태우, 류준영, 차은채, 최은미*

국민대학교 소프트웨어학부

*e-mail:emchoi@kookmin.ac.kr (교신저자)

Medical Expert System Development based on Drools, Ajax and Nginx

Wonyong Jang, TaeWoo Kim, JunYeong Ryu, EunChae Cha, Eunmi Choi*
College of Computer Science, Kookmin University

요 약

본 논문은 웹 기반의 Medical Expert System을 위하여, 오픈 소스 Drools 기반으로 Spring MVC 프레임워크와 Ajax를 이용하여 구현하였다. 클라이언트와 서버 간에 비동기적 통신으로 JSON 데이터를 주고받아 서버의 부담을 감소하였으며, 데이터베이스 접근은 기존의 복잡한 JDBC 단점을 보완하고자 Mybatis 프레임워크를 적용하여 RDBMS의 성능을 향상시켰다. 또한, 웹 기반의 장점을 최대한 활용하여 사용자 수가 늘어남에 따라 서버 트래픽 문제를 해결하기 위하여 Nginx를 이용한 로드밸런싱 구조를 구축하여 서버의 가용성을 확대하였다. 본 Medical Expert 시스템의 처리 프로세스는 일반 사용자가 웹으로 접근하여 설문을 작성한 후 기본적인 진단 및 평가를 제공하여 주는 서비스를 제공한다.

1. 서론

Expert System의 엔진을 사용하여 다양한 추천 시스템을 여러 분야에서 점점 확산되어 가고 있는 동향을 비추어 볼 때, Medical 분야에서 Expert System의 접근성과 가용성, 그리고 성능 면에서 좀 더 효율적이고 확장 가능 필요성이 요구되고 있다. Medical Expert System은 Java 기반의 라이브러리인 Drools를 이용하여 Medical 분야의 특징적인 rule들을 추가하고 변경사항이 있을 때는, 전체 프로그램을 수정할 필요 없이 rule 파일만을 수정하여 진단을 지원할 수 있도록 한다. 하지만, Expert system을 사용할 때 사용자의 접근성을 확대하기 위하여, 웹 기반의 사용자 인터페이스를 설계 및 제공할 필요성이 있다. 웹 기반 서비스의 경우 트래픽이 많을 때 기능에 장애를 줄 수 있기 때문에 이것을 예방하고 해결하기 위해 요청을 분산시켜줄 서버 단의 로드 밸런싱 구조도 본 논문에서는 추가하여 구현하였다. Drools와 로드 밸런싱을 최대한 활용하고, 성능 향상을 위해 Spring MVC 구조와 Ajax를 사용해 웹 기반의 Medical Expert System을 설계 및 구현하였다. Medical 분야 활용은 한방의 분야에 적용하여 료를 적용하였다.

2. 관련 연구

본 장에서는 Expert System을 구축하기 위한 관련 연구 동향을 알아보고, 효율적이고 확장 가능한 Expert System 구축을 위해 사용한 기술에 대한 소개를 한다. 인공지능

및 Expert System은 다양한 분야에서 활용이 되기 위해서는 분야 별 신뢰 할 수 Domain Rule을 정립이 되어야 한다[7]. 본 논문은 Expert System의 활용을 한방 분야에 확대하고, 웹 기반 서비스로 사용자의 접근성을 제공하여 availability, scalability, 그리고 modifiability 등을 초점을 맞추어 시스템을 구현하였다.

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(Forward-chaining) 기반 규칙 엔진을 모두 갖추고 있다.

2.2 Mybatis

Mybatis[6]는 개발자가 지정한 SQL, 저장프로시저 등 고급 매핑을 지원하는 퍼시스턴스 프레임워크이다. JDBC로 처리하는 많은 부분의 소스코드와 parameter 설정 및 결과 매핑을 대신해준다. 또한, DB 레코드에 원시 타입과 Map 인터페이스, Java POJO를 설정해서 매핑하기 위해 XML과 Annotation을 사용할 수 있다. 객체지향 언어인 자바의 관계형 데이터베이스 프로그래밍을 쉽게 할 수 있도록 도와주고 기존의 JDBC의 단점인 Connection과 Statement를 통해 쿼리를 전송하면 연결 받은 후 Close

하는 방식을 보완하기 위해 개발되었다. 따라서 JDBC 코드를 간편하게 사용할 수 있고 SQL 문장과 프로그래밍 소스코드가 분리되어 있어 사용하는데 편리하다.

2.3 Nginx

Nginx[4]는 트래픽이 많은 웹사이트를 위해 확장성을 위해 설계한 비동기 이벤트 기반구조의 웹서버 소프트웨어이다. 대표적인 웹서버 Apache는 thread, process 기반 구조로 요청 하나당 쓰레드 하나가 처리하는 구조이기 때문에 사용자가 많으면 많은 쓰레드를 생성하고 메모리 및 CPU 낭비가 심하다는 단점이 있다. 반면 Nginx는 비동기 Event-Driven 기반 구조로써 더 작은 쓰레드로 Client의 요청들을 처리할 수 있다. 또한, 보안에서는 앞 단의 Nginx로 리버스 프록시를 사용하고 뒷단에서는 WAS를 설치하여 외부에 노출되는 인터페이스에 대해 Nginx WAS 부분만 노출할 수 있다. Backend-service에 대해서는 max fails, fail timeout 시 백업 서버로 진입할 수 있도록 처리할 수 있다.

3. Medical Expert System의 시스템 구조 연구

3.1 Medical Expert System 의 구조도

본 논문에서 제안된 Medical Expert System은 웹의 장점을 살려 일반 사용자가 장소에 제한받지 않고 설문작성을 통해 간단한 진료를 받도록 서비스한다. 아래 그림 1과 같이 먼저 신뢰할 수 있는 Medical 분야의 전문가 Rule을 Knowledge base에 축적한다. 일반 사용자는 웹에 접근해서 간단한 설문을 작성하면 Rule 엔진 중 하나인 Drools 엔진을 이용해서 축적해 놓은 지식과 사용자의 설문 결과를 비교해서 가장 적합한 Rule을 도출해 진단을 내리게 된다.

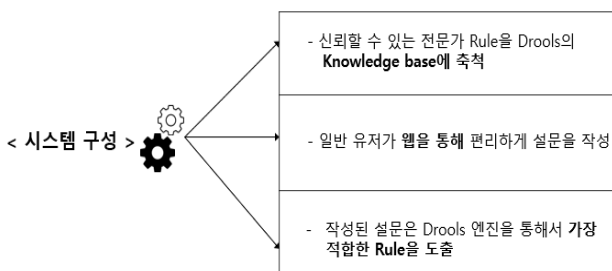


그림 1. Medical expert system 의 시스템 구성

3.2 확장 가능한 Expert System의 시스템 구조

본 논문에서 제안된 시스템은 웹 기반으로 설계 및 구현되었고 Client 요청이 많을 때 처리 속도가 느려진다는 웹의 최대 단점을 극복하기 위하여 서버를 분산하는 로드밸런싱 구조를 구성하고 테스트를 진행하였다. 로드밸런싱은 Client 요청이 많아 발생하는 서버 트래픽이 폭주할 때 여

러 서버로 분산시켜 전체 시스템의 부하를 처리할 수 있도록 한다. 이에 따라서 부하로 인한 속도 저하, 서버의 상태 등을 고려하여 적절히 서비스를 분산시켜 줄 수 있다. 또한, 1대의 서버 장애가 발생하여도 서비스 중단 없이 다른 서버로 적절히 자동으로 분배하여 서비스가 계속 운용할 수 있도록 할 수 있다. 현재 시스템은 WAS인 Tomcat 서버 2개로 분산시켰고, 앞 단에 로드밸런서인 Nginx를 위치시키고 연동하였다. 아래 그림 2는 여러 서버들에 Expert System을 구동할 수 있도록 Nginx를 적용한 시스템 구조도이다.

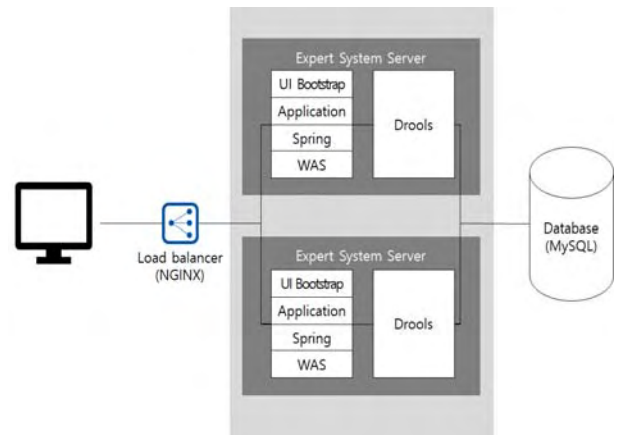


그림 2. Medical Expert System 의 분산 서버 구조

Nginx를 이용하여 로드밸런싱 구성하기 위해서는 upstream 설정이 필요한데 cluster에 참여하는 서버 정보와 port를 upstream 지시자로 설정하며 첫 번째 설정한 서버가 먼저 응답을 처리하게 된다. 또한, 라운드 로빈 방식으로 순회하며 서비스하는 것이 기본 설정이지만, 라운드 로빈 방식은 Client 요청 시 일정한 시간으로 분배해서 작업을 할당해 주지만 작업량이 많은 프로세스가 있을 경우는 Worst case이기 때문에 본 프로젝트 설정은 Client 연결 개수가 가장 적은 서버로 전달하는 방식인 least_connection 방식으로 두 WAS의 부하의 균형을 맞추기 위한 scheduling 으로 설정하였다.

3.3 세션 클러스터링

WAS 시스템을 2개로 분산하면서 두 서버의 세션 유지하여 관리하기 위해서 세션 클러스터링을 해주었다. 세션을 저장하기 위해서 세션에 추가되는 모든 attribute는 Serializable 인터페이스를 이용하여 객체 직렬화를 해주었다. 아래 그림 3은 세션 정보를 저장하는 클래스 일부 소스코드 그림이다.

```
import java.io.Serializable;
public class DoctorVO implements Serializable{
    private String userId;
    private String userPw;

    private boolean useCookie;
    public String getUserId() {
        return userId;
    }
}
```

그림 3. 세션 저장을 위한 객체 직렬화 코드

또한, 세션 정보를 저장하고 유지되는지 확인하기 위해서 쿠키와 세션을 이용한 로그인 상태를 유지할 수 있는 자동 로그인 기능을 구현하였다. 먼저 사용자가 로그인에 성공한 경우 쿠키를 이용하여 사용자 PC에 세션 아이디를 저장해 놓고 세션에는 사용자 객체정보를 저장한다. 동시에 DB에 해당 세션 ID와 유효기간을 기록하고 로그인 상태를 유효기간만큼 자동 로그인 서비스를 제공한다. 세션 정보를 바탕으로 URL을 통해 직접 접근하려고 시도할 때 로그인을 통한 세션 정보가 남아 있게 되면 접근을 허용하였다. 아래 그림 4은 로그인 상태 체크박스를 클릭한 후 로그인이 성공했을 때 현재 날짜에 더해진 유효기간을 보여주는 그림이며 세션 ID와 같이 저장된다.

```
mysql> select * from login;
```

userId	userPw	sessionKey	sessionLimit
admin	123	A99AA19B748CA84C406255008A84D6E0	2018-04-12 00:00:00

그림 4. 쿠키와 세션을 이용한 자동 로그인

3.4 Expert System 의 효율적인 시스템 구조

본 논문에서는 Expert System을 웹 기반으로 확장했을 때 장점을 활용하기 위해 Spring MVC 프레임워크를 적용하였다. Spring MVC 내부 구성은 아래 그림 5과 같다. Front-end는 Html, CSS, Bootstrap으로 구현되어있고 Back-end는 DI, AOP 지원으로 객체 간의 의존성을 낮추고 유연성을 제공하는 Spring MVC 프레임워크, 관계형 데이터베이스를 쉽게 사용할 수 있도록 하는 Mybatis 프레임워크로 구성하였다.

또한, Rule 엔진을 통해 사용자의 설문 결과를 분석해줄 Drools 라이브러리를 이용하여 Java 소스코드와 Rule을 분리하여 유지보수를 용이하게 하였다. 시스템의 기본적인 동작 원리는 UI를 통해 Client의 모든 요청은 톰캣 서버가 Dispatcher Servlet에 전달한다. Dispatcher Servlet은 해당 요청을 받아 Mapping 되어 있는 Controller에게 전달 후 요청에 대한 Service를 해주고 결과값과 결과를 보여줄 jsp 페이지를 Dispatcher Servlet에 다시 전달한다. Client에게 결과 페이지를 보여주기 위해 Spring에서 제공하는 View Resolver가 최종적으로 리턴 되는 jsp파일이름을 연결하여 보여주게 된다. 본 논문에서 제안된 Medical Expert System은 주로 사용자가 작성한 설문지를 Drools 엔진을 통해서 진단 결과를 보여주는 서비스를 제공한다.

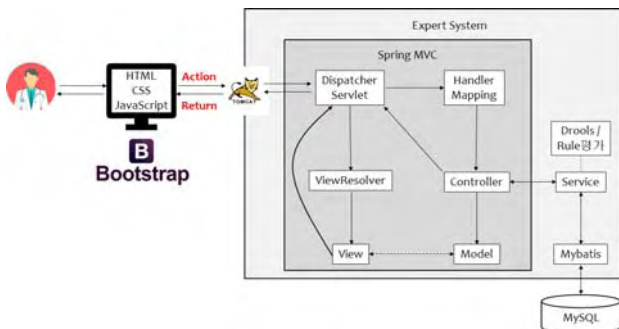


그림 5. Spring MVC 내부 동작 원리

4. Medical Expert System의 처리 프로세스

4.1 Medical Expert System 의 진단 프로세스

본 논문은 일반 사용자에게 설문을 통한 진단을 받을 수 있는 서비스를 제공하기 위해 검증된 사상체질 설문을 적용했다. 아래 그림6과 같이 사상체질 진단을 위해서는 통상적으로 사용되던 단일 단계 설문의 단점을 보완하여 2단계로 구성하였다. 1단계는 22문항으로 이루어져 있으며 음/양 판단을 위한 단계이고, 2단계는 1단계로부터 판단된 정보에 따라 음이면 태음/소음 판단, 양이면 태양/소양 판단으로 각각 18문항/10문항으로 진행한다. 자가 진단으로 음/양 판단에 있어서 모호한 부분을 2단계의 과정으로 의사결정 나무 방식을 통해 해결하였다. 설문을 작성한 뒤 사용자의 사상체질을 진단해주고 그에 따라 몸에 맞는 음식과 건강관리법을 제시해 준다.

이 이미지는 '사상체질 Survey' 웹 페이지의 캡처입니다. 페이지 상단에는 '1단계(양인/음인판단)설문 선택시 자신에 해당하는 사항을 선택하고, 되도록 설문에 오랜시간을 기울이지 않는 것을 추천드립니다.'라는 안내문이 있습니다. 설문 내용은 '1. 당신의 눈은 다음 중 어느 것에 가깝습니까?'로 시작하며, 선택지로는 '① 눈빛이 강하다 눈자위가 많다 눈초리가 올라갔다 눈꺼풀이 얇다. 눈매가 날카롭다.'와 '② 눈빛이 순하다 눈자위가 흐리다 눈초리가 내려갔다 눈꺼풀이 두껍다. 눈매가 부드럽다.'가 있습니다. 이 선택지들은 '음/양 판단'으로 이어집니다. 화면 하단에는 '태음/소음 판단'과 '태양/소양 판단'의 선택지가 표시되어 있습니다.

그림 6. 사상체질 진단 설문 화면

4.2 Medical Expert System의 Rule 구조

아래 그림 7 은 Drools 엔진이 검사하는 Rule을 소스 코드와 그림으로 보여준다. Rule 마다 고유 이름이 존재 하고 attribute를 이용하여 Rule간의 충돌과 무한루프 등을 예방해 줄 수 있다. 먼저 첫 번째 소스코드에서 When 절의 조건을 만족 한다면 Then 의 문장을 수행하는 구조이다. 음/양 판단 Rule을 검사 했을 때 음으로 판정한 경우 Rule 엔진은 그에 맞는 Rule 5번을 찾아가 검사를 하게 된다.

```
rule "Number 15 : 음 / 양 판단"
lock-on-active true
when
    surveyObject: DecisionYinYang ( count >= 12 )
then
    modify(surveyObject){
        surveyObject.setYinYang("음"),
        surveyObject.getQuestionB()
    };
end
```

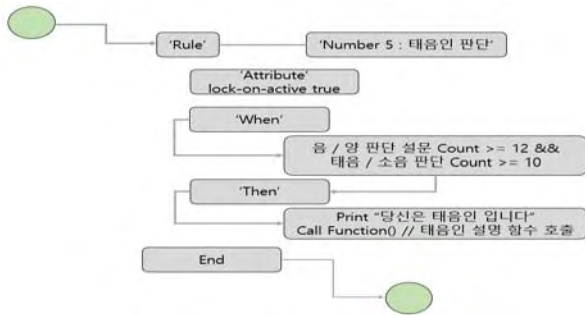


그림 7. Medical Expert System의 Rule

5. Medical Expert System 구현 결과

본 논문의 제안된 Medical Expert System에서 아래 그림 8은 사용자가 설문을 작성 완료한 후 볼 수 있는 진단 페이지 UI이며, 설문결과를 통해서 진단을 받을 수 있다.



그림 8. Medical Expert System 진단 UI

Drools 엔진에 의해서 진단을 받기 위해서는 이름을 통해서 작성한 설문의 결과를 Ajax를 이용해 요청 페이지에 나타내고 진단하기 버튼을 클릭했을 때 Drools 지식베이스에 축적되어있는 Rule들과 비교해서 결과 값을 사용자에게 보여준다. 그림 9은 사용자가 작성한 설문 결과를 보 있다.



그림 9. Medical Expert System 결과 화면

Rule 검사를 위해 증상데이터를 추출한다. 여기서 설문에서 놓친 증상이 있다면 증상 선택리스트에서 추가 또는 삭제를 통해서 업데이트할 수 있다. 증상데이터가 완료되면 진단하기 버튼을 눌렀을 때 결과를 확인할 수 있다.

6. 결론

본 논문은 Medical Expert System을 웹 기반에서 효율적으로 설계 및 구현하기 위하여 Spring MVC 프레임워크의 내부 동작 과정을 정확히 이해하고 Drools와 연동시킴으로써 Drools 기반으로 Medical Expert System을 설계 및 구현하였다. 또한, 서버의 부담을 줄이고 사용자의 편의성을 증진하고자 비동기적 통신을 하는 Ajax를 사용하여 페이지 전환을 최소화하였다. 데이터베이스 접근은 기존의 JDBC의 복잡한 소스코드의 단점을 극복하고자 Mybatis 프레임워크를 사용하여 데이터베이스 명령어와 java 소스를 분리해 유지보수를 용이하게 하였다. 웹을 기반으로 구현되었기 때문에 서버 트래픽 문제를 해결하기 위해 Nginx를 이용한 로드밸런싱 구성하고 테스트하여 추후 원활한 서비스 제공을 할 수 있도록 하였다.

본 연구는 중소기업부의 기술혁신개발사업의 일환으로 수행하였음. [A2017-0293]

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] Jan Ruzicka, “Integrating DROOLS and R software for intelligent map system”, Geoinformatics FCE CTU, Vol 7, pp 85-92, 2012.
- [4] <https://nginx.org>
- [5] <https://projects.spring.io/spring-framework/>
- [6] <http://www.mybatis.org>
- [7] 허종원(1993), “Expert System 구축의 문제점과 개선 방향”, 한국지능정보시스템학회 학술대회논문집, 81-101
- [8] 최승훈, “인터넷 기반 한의진단전문가 시스템 (KHU-PIPE) 개발”, Korean J. Oriental Physiology & Pathology 16(3):528-531, 2002
- [9] 장원용, 최유나, 양성수, 최은미 “메디컬 Expert 시스템을 위한 Drools와 JavaFX 기반의 사용자 인터페이스 설계 및 구현”, 한국정보처리학회 학술발표논문집, Vol. 24 No. 02 pp 0827-0830, 2017.
- [10] 김영원, 신동윤, 김정호, 최대성, 임미경, 이경로, 송정모, “사상체질 진단을 위한 2단계 설문지(TS-QSCD)개발 연구”, 사상체질의학회지, Vol. 18, No. 1, 2006