

SaaS 기반의 전자도서관 통합관리 서비스 시스템의 개선 Improvement of Intergrated Management Service System for Digital Library Based on SaaS

민 병 원, 오 형 용, 오 용 선
목원대학교 정보통신공학과

Min Byoung-Won, Oh Hyoung-Yong, Oh Yong-Sun
Mokwon Univ.

요약

현재, 도서관시스템의 소프트웨어 사용 방식은 클라이언트/서버 및 ASP 방식으로 서비스를 제공함으로써 하드웨어 및 소프트웨어 구매비, 설치 및 배포, Customization, Upgrade, 문제점 관리, 라이선스의 고비용 등 소프트웨어 전반에 걸쳐 관리가 힘들고 고비용의 문제점이 있다. 이러한 문제점을 해결하기 위해 SaaS 기반의 전자도서관 통합관리 서비스 시스템에서는 SaaS 성숙도 레벨 3 이상에서 구현이 가능한 핵심요소들을 개발하였다. 그러므로 초기 투자비용이 거의 없고, 쉽고, 간편하며, 저비용 IT 서비스가 가능한 SaaS 기반의 소프트웨어 온-디맨드 방식의 서비스 모델로 성능을 개선하였다.

I. 서론

SaaS는 사용자가 필요한 소프트웨어를 인터넷을 통해 온라인 서비스로 이용할 수 있도록 하는 최신의 소프트웨어 배포 모델로 정의할 수가 있으며[1]-[3] 또한, 응용 소프트웨어를 인터넷을 통하여 다수의 사용자에게 온라인 서비스로 제공하는 기술로도 정의할 수 있다[4]-[5]. 즉, 사용자는 인터넷을 통해 소프트웨어를 사용하고 그에 대한 비용만 지불하는 방식으로 복잡한 소프트웨어 및 하드웨어의 관리라는 부담에서 벗어날 수 있다[6].

SaaS를 기존의 ASP 방식과 비교해 보면, 사용자의 입장에서 소프트웨어를 인터넷을 통해 사용하는 점에서는 큰 차이는 없으나 사용자를 위한 커스터마이징을 ASP처럼 소프트웨어 공급자가 하지 않고 사용자가 직접 할 수 있다는 점에서 차별성을 갖는다.

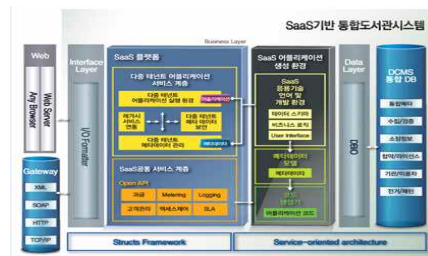
본 논문에서 제안한 SaaS 플랫폼 기반 통합도서관시스템은 사용자의 커스터마이징을 메타데이터를 활용해서 지원하며 사용자들 또는 사용자 그룹으로 표현되는 테넌트들을 하나의 소프트웨어 인스턴스로 지원(single instance multi-tenant) 한다는 점에서 차별성을 갖는다. 이는 커스터마이징에 많은 비용이 들고 인스턴스를 개별적으로 띄우기 때문에 규모의 경제를 실현하지 못했던 ASP의 단점을 해결 할 수 있도록 성능을 개선하였다.

II. 본론

1. 시스템 개념도

SaaS 플랫폼 기반 통합도서관시스템은 (그림 1)과 같이 다중 테넌트 어플리케이션 서비스 계층, SaaS 공통

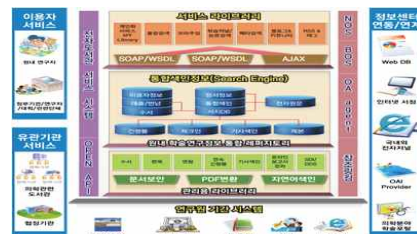
서비스 계층, SaaS 어플리케이션 생성 환경 계층으로 구성되어 있다.



▶▶ 그림 1. 시스템 개념도

2. 서비스 개념도

현재 (SaaS 플랫폼 기반 통합도서관시스템의 서비스 개념도는 (그림 2)와 같이 이용자 서비스, 유관기관서비스, 정보센터 연동/연계서비스, 관리자 서비스 등으로 구성된다.

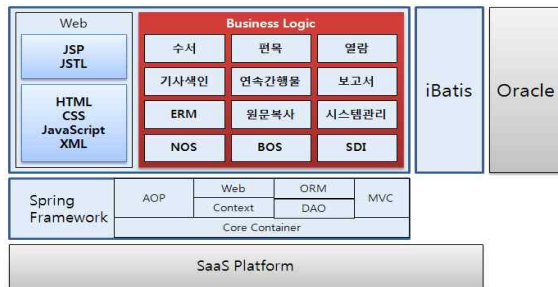


▶▶ 그림 2. 서비스 개념도

3. 개발환경

SaaS 플랫폼 기반 통합도서관시스템은 (그림 3)과 같

이 3개의 Layer로 구성되어 있다. 첫 번째 Layer는 SaaS Platform인 Nereus 기반으로 API를 제공한다. 두 번째 Layer는 Spring Framework, AOP, ORM, DAO, MVC, Web, Context 등으로 구성된다. Application 구현을 위한 세 번째 Layer는 Web(JSP, JSTL, HTML, CSS, JavaScript, XML), Business Logic(수서, 편목, 열람, 기사색인, 연속간행물 등), iBatis, Oracle로 구성 되어 있다.



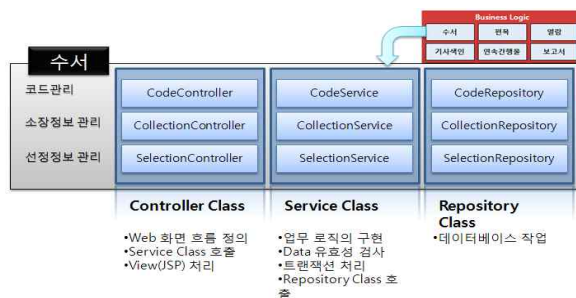
▶▶ 그림 3. 개발환경

Spring Framework은 오픈 소스 프레임워크로서 기업 애플리케이션 개발의 복잡성 문제를 다루기 위해 만들어 졌다. Spring Framework의 가장 큰 장점 중 하나는 레이어 아키텍처라는 점이다. J2EE 애플리케이션 개발에 일관된 프레임워크를 제공함과 동시에 원하는 컴포넌트만 선택적으로 사용할 수 있다는 이점이 있다.

iBatis는 비즈니스 로직과 데이터베이스 접근을 분리하여 객체지향을 가능하게 한다. SQL문과 코드를 독립적으로 작성할 수 있어 데이터 처리에 대한 유연성을 제공한다. 또한 트랜잭션 및 캐시 기능을 제공한다.

4. 기능별 클래스 구조

SaaS 플랫폼 기반 통합도서관시스템의 Application 구현을 위한 클래스는 (그림 4)에서 같이 크게 3개의 구조로 구성이 된다. Controller Class에서는 Web 화면 흐름 정리, Service Class 호출, View(JSP) 처리 등을 수행한다. Service Class에서는 업무로직의 구현, Data 유효성 검사, 트랜잭션 처리, Repository Class 호출 등의 처리한다. Repository Class에서는 데이터베이스를 위한 SQL, Business Logic 등을 수행한다.



▶▶ 그림 4. 기능별 클래스 구조

5. 기능 개선 구현

메인 화면은 (그림 5)에서와 같이 핵심 디자인 컨셉은 사용자화(Customise)이다. 각각의 카테고리(메뉴)별로 사용자가 원하는 항목을 편집할 수 있고 화면의 상단으로 편집할 수 있게 디자인하였다. 전체적인 컬러컨셉은 은은하고 화면을 보았을 때 눈의 피로를 덜어줄 수 있는 Green색상을 사용하였다.



▶▶ 그림 5. 메인 화면

III. 결론

현재, 도서관시스템의 소프트웨어 사용 방식은 클라이언트/서버 및 ASP 방식으로 서비스를 제공함으로써 하드웨어 및 소프트웨어 구매비, 설치 및 배포, Customization, Upgrade, 문제점 관리, 라이선스의 고비용 등 소프트웨어 전반에 걸쳐 관리가 힘들고 고비용의 문제점이 있다. 이러한 문제점을 해결하기 위해 SaaS 기반의 전자도서관 통합관리 서비스 시스템에서는 SaaS 성숙도 레벨 3이상에서 구현이 가능한 핵심요소들을 개발하였다. 그러므로 초기 투자비용이 거의 없고, 쉽고, 간편하며, 저비용 IT 서비스가 가능한 SaaS 기반의 소프트웨어 온-디맨드 방식의 서비스 모델로 기능을 개선하였다.

■ 참고 문헌 ■

- [1] 신현석, "MS 클라우드 컴퓨팅과 애저 서비스 플랫폼 이해," ZDNet Korea, 2008, 12, 12.
- [2] 김영만, "웹기반 SaaS 플랫폼," 2008, 12, 17., <http://www.software.or.kr/ICSfile/afieldfile/2008/12/18/5.pdf>
- [3] 디지털타임스, "SaaS와 소프트웨어의 미래," 2006.5. 4.
- [4] http://en.wikipedia.org/wiki/Software_as_a_service
- [5] What is SaaS?, <http://www.salesforce.com/naas/>
- [6] 김형완 외 12, "SaaS 기술 개발 동향", 한국전자통신연구원, 전자통신동향분석 제24권 제4호, 2009, 8.