碩士學位論文

서비스지향 동적 협업시스템의 설계 및 구현

濟州大學校 大學院

컴퓨터工學科

李 惠 善

2007年 12月

서비스지향 동적 협업시스템의 설계 및 구현

指導敎授 郭 鎬 榮

李 惠 善

이 論文을 工學 碩士學位 論文으로 提出함

2007年 12月

李惠善의 工學 碩士學位 論文은 認准함

審査委員長 _____ 印

委 員 _____ 印

委 員 _____ 印

濟州大學校 大學院

2007年 12月

The design and implementation of Dynamic collaboration system

Hye-Sun Lee
(Supervised by professor Ho-Young Kawk)

A thesis submitted in partial fulfillment of the requirement for the degree of Master of Computer Engineering

2007. 12.

This thesis has been examined and approved.

Thesis director, _____

Thesis director, _____

Thesis director, _____

December 2007

Department of Computer Engineering
Graduate School
Cheju National University

목 차

그 림 목 차

표 목 차

국문초록

서비스지향 동적 협업시스템의 설계 및 구현

컴퓨터공학과 이혜선
지 도 교 수 곽호영

최근, 기업들은 빠르게 변화하는 비즈니스 요구사항에 신속하게 대응할 수 있는 전략과 데이터 공유 기반의 경쟁력 강화를 필요로 하고 있다.

하지만 기업들은 복잡하고 서로 연동되기 힘든 다양한 플랫폼 환경으로 구축되어 재사용성과 애플리케이션의 통합 및 비즈니스 환경 변화에 융통성과 민첩성을 적용하지 못하고 있다. 또한, 웹 환경의 반복적인 페이지 리로딩 현상은 갱신된 페이지에 대한 사용자의 확인과정이며, 이로 인해 서버의 응답성 및 서비스 구현에 문제를 제기하기에 이르고 있다.

따라서 기업의 융통성과 민첩성을 확보하고 또한 웹 서버의 효율적인 서비스 환경과 사용자의 리로딩에 따른 문제점들을 해결하기 위해 본 논문은 기업 간 메타데이터 정보를 통합 관리하고 사용자에게 동적 인터페이스를 제공할 수 있는 서비스 지향 동적 협업 시스템을 제안하고 구현한다. 제안된 시스템에 적용된 기술은 서비스지향 아키텍처와 Ajax이며, 이를 통해 동종 기업 간 데이터 공유 및 통합을 추진할 수 있으며, 또한 사용자에게 동적인 페이지를 제공할 수 있게 된다.

특히, 지역적 특성을 고려하여 제안된 시스템은 현재 독자적으로 운영 관리되고 있는 렌트카 업체를 대상으로 동적 사용자 인터페이스 계층, 마스터서버 계층, 그리고 질의 전송 및 응답계층으로 설계되고 구현된 부분을 제시하였으며, 구현 결과로서 동종 기업 간 업무 연계의 가능성, 그리고 중복된 데이터의 표준화를 구축하였다.

또한 구현된 제안 시스템에 대한 효용성 분석은 Firebug 분석도구를 이용하여 실시간 처리 응답성과 기존 시스템과의 트래픽 요소에서 측정하여 제안시스템의 효용성을 제시하였으며, 이로 인해 향후 제안시스템을 이용한 새로운 서비스 시스템 구축은 비용 절감 효과와 유지관리에 효율성을 높일 수 있을 것이다.

ABSTRACT

The design and implementation of Dynamic collaboration system

LEE, HYE-SUN
Department of Computer Engineering
Graduate School
Cheju National University

Recently, enterprises need the strategy which can confront rapidly changing business requirements and reinforcement of competitiveness based on data's share.

But, the enterprises can't apply flexibility and promptitude to a reusability characteristic, integration of application, and the environmental change of business because of the platform that is complicated and hard to interconnect each other. Also, the pages phenomenon of repeated reloading in a web environment are the process of user's confirmation at a renewed page and it results in bringing up the problem in responsiveness of server and implementation of service.

Consequently, To keeping on enterprises' flexibility and adaptability and solving the problem which be an efficient service environment of web server and page's reloading, this paper is design and implement the service oriented dynamic collaboration system that can integrate and manage meta-data information among enterprises and provide the dynamic page. This system's technique are service oriented architecture and Ajax, and it will lead the share and integration of similar enterprises' data and provide dynamic page

to user.

Specially, this system which considers regional properties is designed to the dynamic user's interface layer, master server layer, and query translation and response layer for rent-a-car enterprises which are currently operated, and implemented to a possibility of collaboration system among similar enterprises and a standardization of duplicated data.

Also, the efficiency analysis in the proposed system suggested that usefulness of the proposed system through measuring in real-time responsiveness and traffic with an existing system by using Firebug's analysis tool. Therefore, the proposed system is efficiency than an existing system in cost cutting and maintenance managing.

약 어 표

SOA	Service Oriented Architecture
IT	Information Technology
C/S	Client Server
TCO	Total Cost of Ownership
ROI	Return On Investment
XML	Extensible Markup Language
Ajax	Asynchronous Javascript and XML
HTML	Hypertext Markup Language
OASIS	The Organization for the Advancement of Structured Information Standards
CORBA	Common Object Request Broker Architecture
WSDL	Web Service Description Language
UDDI	Universal Description, Discovery, and Integration
SOAP	Simple Object Access Protocal
EAI	Enterprise Application Integration
ESB	Enterprise Service Bus
B2Bi	Business to Business integration
SLA	Service Level Agreement
RIA	Rich Internet Application
CSS	Cascading Style Sheets
DOM	Document Object Model
XSL	Extensible Style Language
XSLT	XSL Transformations
W3C	World Wide Web Consortium
XPath	XML Path Language
XHTML	Extensible Hypertext Markup Language

SGML	Standard Generalized Markup Language
WML	Wireless Markup Language
RaCSS	Rent-a-Car Service System

I. 서 론

1. 연구 배경 및 목적

최근 인터넷이 발달되고 사용자 요구사항이 증가되면서 산업의 가치 초점은 기업 중심에서 사용자 중심으로 변화되고 있으며, 이에 따라 사용자는 기업에서 제공하는 서비스에 대한 편리한 인터페이스와 또한 빠른 응답을 기대하고 있다.

그러나, 최근까지 기업들은 로컬, Client and Server(C/S) 그리고 웹 환경 기반으로 시스템들을 구축하였기 때문에 서로 연동되기 힘든 다양한 플랫폼, 애플리케이션의 통합에 따른 어려움 그리고 재사용성, 융통성 그리고 민첩성의 부재로 비즈니스 환경 변화에 용이하게 적응하지 못하는 문제점을 안고 있다. 또한 기업에서 제공하는 웹 정보 시스템은 네트워크 인프라가 발전하여 네트워크 속도가 향상되었지만 새로운 내용을 보여주기 위한 반복적인 웹 페이지 전체의 리로딩은 사용성과 응답성의 한계로 지적되고 있다. 이에 위와 같은 문제점들을 해결하면서 다양한 정보기술 자원을 통합하고, 기업 간 상호 연동할 수 있으며 사용자들에게 다양한 서비스를 제공할 수 있는 시스템을 필요로 하고 있다[1,2,3,4].

반면, 기업의 요구사항을 만족시킬 수 있는 시스템에 대한 필요성이 증가하면서 서비스지향 아키텍처(Service Oriented Architecture : SOA)에 대한 관심이 확대 되고 있다. 서비스지향 아키텍처는 기존 애플리케이션의 서비스를 조합함으로서 새로운 애플리케이션 서비스를 구현할 수 있고 기업의 정보시스템을 서로 공유할 수 있을 뿐만 아니라 재사용이 가능한 서비스와 컴포넌트 중심으로 구성된 정보기술 아키텍처이다. 그러나 현재 서비스지향 아키텍처의 관심은 확대되고 있지만 이에 비해 실제 서비스지향 아키텍처 개념을 적용하여 시스템을 개발한 기업들은 소수에 불과하다.

또한 가트너 보고서에 의하면, 현재 정보기술에 있어서 핵심 이슈인 통합은 기

업의 투자비용 절감 그리고 날로 늘어나는 솔루션 관리의 문제 해결에 아주 필요한 요소라고 강조하고 있다. 즉 효율적인 비즈니스 통합은 총소유비용(Total Cost of Ownership : TCO)을 줄이고, 또한 투자 수익(Return On Investment : ROI)을 극대화할 수 있는 방안으로 인식되고 있다[6,7].

따라서 본 연구는 고객 만족도를 향상시키고 기업 경쟁력을 강화하며 또한 시스템 운용비용 절감과 데이터 및 애플리케이션 그리고 프로세스들을 보다 체계적이고 효율적으로 운영할 수 있는 서비스지향 아키텍처 개념을 적용한 서비스지향 동적 협업 시스템을 제안하고 구현한다. 제안된 시스템은 협업을 통해 기업의 융통성과 빠른 응답성을 제공하고 또한 Ajax를 통한 비동기 방식의 편리한 사용자 인터페이스를 사용자에게 제공한다. 시스템 구현결과, 동종 기업 간 업무 연계가 가능하여, 개발 언어 및 플랫폼에 상관없이 기업 간 중복된 데이터 표준화 및 이들을 최소화한 상태에서 서비스를 제공할 수 있었다.

2. 연구 내용

본 연구는 서비스지향 아키텍처와 비동기 기술인 Ajax를 적용한 동종 기업 간 협업 시스템을 제안하고 구현하였다. 동종 기업 간 정보교환을 위한 메커니즘으로 메타데이터 관리 서버를 구축하였으며, 또한 기업 간 정보 교환을 위해 메타데이터를 상호 변환할 수 있는 변환기를 구현하였다. 그럼으로써 클라이언트들에게 동종 기업 간 협업 시스템 내에서 자유롭게 원하는 질의를 수행하고 또한 원하는 질의 결과를 획득할 수 있는 환경을 구성하였다. 또한 반복적인 페이지 리로드가 없는 편리한 사용자 인터페이스를 Ajax 비동기 통신을 통해 구현하였다.

그럼으로써 동종 기업 간 업무 연계를 가능하게 하였으며, 또한 중복된 데이터들에 대한 표준화 및 최소화를 이루었으며 그리고 개발 언어 및 플랫폼에 상관없이 서비스를 제공할 수 있는 환경을 구축하였다. 끝으로 협업 시스템의 처리 결과에 대한 검색 가능성과 Mozilla Firefox 브라우저의 Firebug 분석도구를 이

용하여 제안 시스템의 효율성을 증명하였다.

3. 논문 구성

본 논문의 전체 구성은 다음과 같다. 2장에서는 기존 동종 기업들 간 분산 시스템들의 문제점들과 서비스지향 아키텍처 및 협업시스템 종류, Ajax 그리고 애플리케이션 간 데이터 전송 형식인 XML 기술들을 알아보았다. 3장에서는 서비스지향 아키텍처와 Ajax 기술을 적용한 동적 협업시스템을 제안하고 각 구성 요소들의 기능과 역할들을 서술하였다. 4장에서는 제안된 시스템을 구현하였으며 5장에서는 타 개발 방법론과 비교 분석하고 협업 시스템의 처리 결과에 대한 검색 가능성과 분석 도구인 Firefox의 Firebug를 이용하여 웹 페이지 리로드에 소요되는 페이지 트래픽 양을 분석하였다. 그 결과로서 제안된 시스템의 효용성을 도출하였다. 끝으로 6장에서는 결론 및 향후 연구 방향을 제시하였다.

II. 관련 연구

본 장에서는 기존 동종 기업들 간 분산 시스템들의 문제점들을 살펴보고, 이를 해결할 수 있는 서비스지향 아키텍처와 협업시스템 종류, Ajax, 또한 애플리케이션 간 데이터 전송 형식인 XML 기술들을 알아본다.

1. 분산 시스템들의 문제점

기존 분산 시스템들은 다음과 같은 다양한 문제점들을 갖고 있다.

- 비즈니스와 IT사이의 단절로 비즈니스 분야의 변화를 수용하기 위해 필수적인 민첩성(agility)과 유연성(flexiblity), 그리고 효율성(efficiency)이 부족하다.
- 컴포넌트 기능과 형식들이 다양하고 또한 서로 밀접하게 연관되어 있어서 시스템 일부나 전체를 변경하기가 어렵다.
- 기업 간 데이터 통합 시 통합에 관련된 데이터 체계 분류가 부재하며, 또한 각각의 기업에 종속된 데이터베이스를 설계하고 구축해 왔기 때문에 기업 간 중복된 데이터가 다량으로 존재하고 또한 많은 구축비용이 요구된다.
- 분산 컴포넌트 간 사용하는 프로토콜이 서로 상이하여 분산 컴포넌트 간 데이터 교환 시 데이터 변환에 어려움이 있다.
- 기존 웹 어플리케이션에서는 웹 브라우저가 웹서버에 질의를 요청하면 웹서버는 사용자의 요청을 처리한 뒤 결과를 HTML로 재구성하여 이를 웹브라우저에 전송하게 되고, 웹 브라우저는 전송받은 HTML을 분석한 뒤 그 결과를 화면에 보여주게 된다. 이 때 요청과 응답의 사이클 동안 사용자는 선점하지 못하고 대기해야 한다.
- 이상적인 네트워크 환경에서도 페이지 리로딩은 사용성 및 응답성에 있어 문제로 대두되고 있다.

2. 서비스지향 아키텍처(Service Oriented Architecture)

1) 서비스지향 아키텍처

서비스지향 아키텍처는 그림 1.에서 보여 지는 것처럼 이미 CORBA나 DOCOM등의 분산 객체 기술에서 그 기본 개념으로 사용되었으나 기술적인 미성숙과 벤더마다 독자적인 데이터 형식과 전송 프로토콜, 인터페이스 정의 언어를 사용하는 공개 표준의 부재로 서로 다른 벤더 간의 상호 운영이 어려웠다. 그러나 XML기반 웹 서비스 기술이 효율적인 통합기술로 각광받으면서 동적이고 유연한 인터페이스 방식의 서비스지향 아키텍처 개념이 다시 대두되었다[9,10].

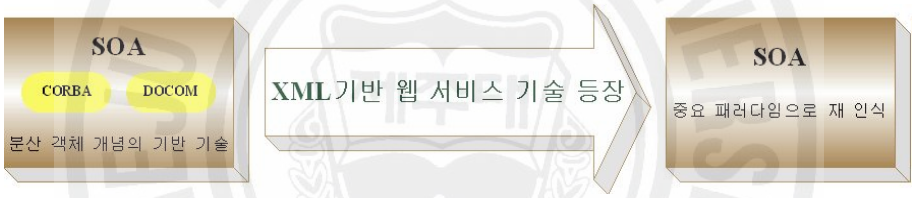


그림 1. 서비스 지향 아키텍처 출연 배경

서비스지향 아키텍처는 1996년 가트너 그룹에 의해 “서비스지향 아키텍처는 각기 다른 소유권을 가진 다양한 영역에 속해있는 분산된 기능을 조직화하고 활용하기 위한 패러다임으로서 측정 가능한 사전 조건 및 기대치와 일치하고 원하는 결과를 도출하기 위한 기능을 제공, 발견, 상호작용 및 사용하는 단일 수단을 제공한다.” 라고 소개되었다[8]. 그리고 서비스 지향 아키텍처 참조 모델은 OASIS(The Organization for the Advancement of Structured Information Standards)협회에서 그림 2.와 같이 정의되었다.

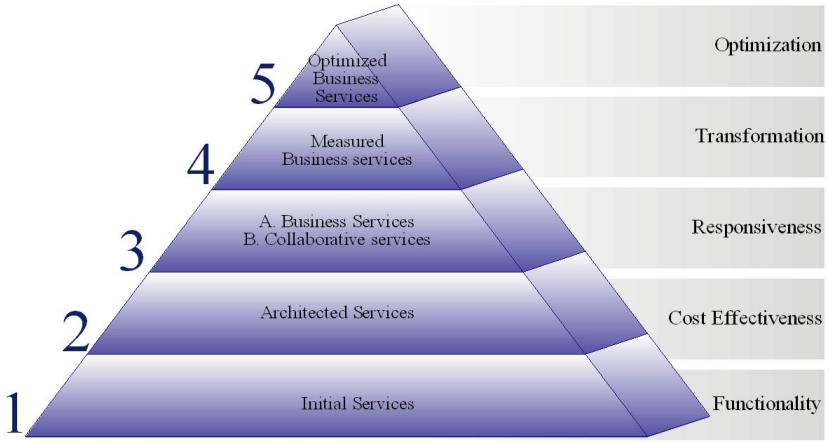


그림 2. 서비스지향 아키텍처 성숙 단계

서비스지향 아키텍처는 특정 구현에 얽매이지 않는 서비스들 간 약 결합(loose coupling)으로서 각 서비스의 내부 구조 및 구현의 변화에 대응할 수 있다. 반면 강 결합(tight-coupling)은 애플리케이션의 다양한 컴포넌트들이 기능과 형식면에서 밀접하게 연관되어 있어 애플리케이션 일부나 전체를 변경하기가 어렵다. 이러한 문제를 해결하기 위해 기존 기업들은 많은 비용과 시간을 투자하고 있다. 약 결합 시스템의 필요성은 비즈니스 애플리케이션이 변화하는 환경에 보다 민첩하게 대응할 수 있게 해준다는 것이다[5].

2) 서비스지향 아키텍처 특징과 이점

서비스지향 아키텍처를 적용한 서비스는 다음 특징들을 제공한다.

- 서비스가 필요한 사용자에게는 작업 수행 중에 필요로 하는 서비스를 찾고 그것을 사용할 수 있게 해준다.
- 각각의 서비스는 독립적으로 개발, 유지, 관리되며 서로의 실행 환경에 큰 영향을 미치지 않는다. 즉, 서비스는 컴포넌트와 같이 독립된 모듈이다.
- 서비스는 플랫폼에 관계없이 상호 운용이 가능하다.
- 서비스는 상호 작용하기 위해 필요한 최소한의 정보를 제외한 다른 어떤 것

도 공유할 필요가 없다.

- 서비스는 약 결합을 기초로 상호 작용하도록 설계된다. 그리고 이러한 약 결합은 추상화 및 서비스 자율성과 연관되기 때문에 그 상태가 유지된다.
- 서비스는 서비스 정의와 위치 정보를 UDDI와 같은 레지스트리(Registry)에 저장하여 위치 투명성을 제공한다. 따라서 사용자들은 실제 서비스의 정보가 변경되더라도 사용자는 서비스 저장소를 통해 서비스를 호출할 수 있게 된다.
- 기업 간의 통합을 효과적으로 구현할 수 있으며, 서비스 요청자가 서비스를 발견하고 이해하는데 용이하다.

또한, 서비스지향 아키텍처는 엔터프라이즈 규모에서 전통적인 IT간 경계를 허물어 중앙 IT, 비즈니스 유닛 및 조직 전체에 다음과 같은 이점을 제공한다.

- 서비스 인터페이스를 표준화하며 또한 공유 서버 자원의 유틸리티를 통해 전통적인 조직의 사일로(Silo)방식 모델에서 공유자원 모델로 발전할 수 있는 기회 및 이를 통한 통합의 효율성을 제공한다.
- 서비스를 재사용함으로써 새로운 애플리케이션의 기능 및 개발 시간을 단축시킨다.
- 해당 메타데이터가 파일 또는 실행 단계 서비스 수준 동의서(SLA : Service Level Agreement)로 인코딩된 비즈니스 프로세스인지 여부에 관계없이 코드를 변경하지 않고 메타데이터를 통해 배치된 IT 시스템 작동을 변환함으로써 비즈니스 이해 관계자에게 유용한 민첩성을 제공함으로, 서비스지향 아키텍처는 빠르게 변화하는 IT 시스템 요소를 메타데이터로 구성하여 IT 변화 속도를 증가시킬 수 있는 기회를 준다.
- 기업 간 데이터 교환 시 메시징 레이어 중계 기능은 네트워크 가시성을 향상시키며, 또한 IT 시스템 작동에 대한 의미를 분석케 하며, 그리고 기술 시스템과 비즈니스 성과 간의 강력한 연결 관계를 가능케 한다.
- 끝으로, 서비스지향 아키텍처의 표준 기반 통신 기능은 다양한 IT 시스템을 완벽하고 신속하게 통합한다[12,13].

3) Enterprise Service Bus (ESB)

ESB는 비즈니스 내에서 서비스, 애플리케이션, 자원들을 연결시켜주거나 통합시켜주는 일종의 미들웨어 패턴이다. 또한 여러 서비스들이 연계 되는 Back Bone의 역할을 수행하며, 새로운 상호 운영성을 제시한다. 그리고 정보가 필요한 시점에 이를 필요로 하는 사람에게 소통되게 함으로써 의사 결정에서의 대응력 및 정확성을 높인다. ESB의 구성은 그림 3.과 같다.

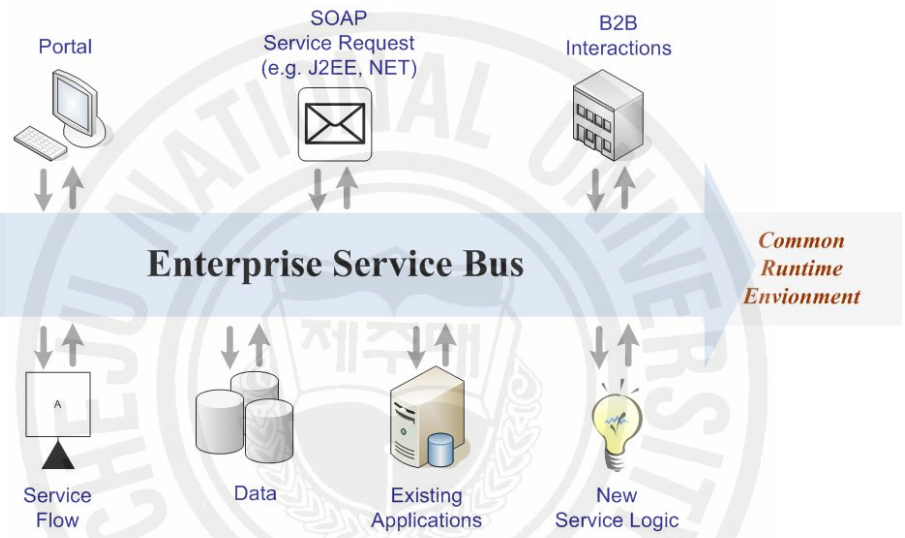


그림 3. ESB 구성도

ESB는 기업 IT 환경에서 중추적인 역할을 수행하는 것으로 각 레거시 시스템과의 연동을 위한 다양한 표준 프로토콜의 지원을 기본으로, 재사용 가능한 컴포넌트들을 조립함으로써 서비스 지향적인 기업 환경을 만들 수 있는 기반이 된다.

ESB는 다음과 같은 특성을 갖는다[14,15].

- XML 기반의 동기/비동기 SOAP messaging
- 안정적이고 신뢰성을 보장하는 messaging
- Intelligent routing and intermediaries
- 데이터 변환(XSLT, XQuery)

비즈니스 통합 솔루션은 다음과 같이 기업애플리케이션 통합(EAI), 기업간 통합(B2Bi) 그리고 비즈니스 프로세스 관리(BMP)로 구성된다.

- 기업애플리케이션 통합(EAI)은 기업 내부의 상이한 애플리케이션과 데이터를 단일 미들웨어를 사용하여 유기적으로 통합함으로써 인터페이스의 개발 및 유지 보수를 위한 신속성과 높은 생산성은 물론 관리의 정확성과 효율성을 제공 한다.
- 기업 간 통합(B2Bi)은 기업들 사이에 송수신되는 표준적인 데이터에 대한 가공 및 흐름제어를 통하여 인터넷상에서 기업 간에 발생하는 연계업무를 자동화(Business Process Automation) 한다.
- 비즈니스 프로세스 관리(BMP)은 기업 내, 외부의 사람과 컴퓨터 시스템을 업무 프로세스에 따라 자동화함으로써 업무 효율을 획기적으로 개선할 수 있는 새로운 애플리케이션으로 EAI와 B2Bi 솔루션을 근간으로 한다. 이러한 BPM의 목적은 비즈니스 프로세스를 실행하는 애플리케이션들로부터 프로세스 로직을 분리함으로써, 비즈니스 프로세스를 설계, 분석, 최적화하여 자동화하는 것이다.

위에서 제시한 주요 솔루션들의 공통적인 동향으로는 XML 기반의 웹 서비스를 지원 한다. 웹 서비스가 비즈니스 통합에 주목을 받기 시작한 이유는 하나의 표준 기술로 EAI와 B2Bi 모두를 구현할 수 있으며 이를 이용하여 구현된 통합 애플리케이션 간에는 상호 운영성이 보장 된다[16].

4) 서비스지향 아키텍처와 웹서비스 관계

현재 XML 표준을 바탕으로 약 결합된 애플리케이션 컴포넌트 형태로 개발되고 있는 웹 서비스(Web Service)는 서비스지향 아키텍처에 기반 한 기술이다. 따라서 서비스지향 아키텍처와 웹 서비스 아키텍처의 서비스 구조에서 제공되는 서비스의 정의와 이에 따른 위치정보들은 단일한 저장소에 저장되며, 이는 클라

이언트들에게 저장소에 있는 정보를 위치에 관계없이 등록, 호출할 수 있는 위치 투명성을 제공하고 있다. 그림 4.는 이와 같은 특징을 갖는 웹 서비스 구조를 나타낸다.

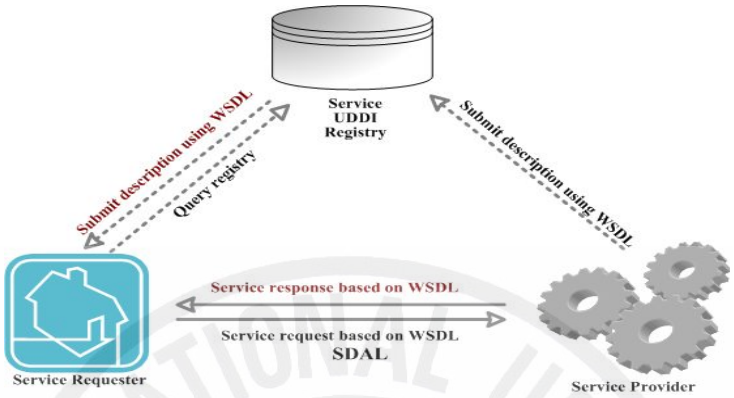


그림 4. 서비스지향 아키텍처 기반 웹 서비스 구조

그림 4.에 제시된 바와 같이 서비스지향 아키텍처 기반의 웹서비스는 서비스 사용자(Service Requester), 서비스 제공자(Service Provider), 서비스 중계자(Service Registry)로 구성된다. 서비스 등록 과정은 서비스 인터페이스, 데이터 타입 그리고 위치 정보 등 서비스 호출 시 필요한 상세 정보를 서비스 제공자가 WSDL(Web Service Description Language)로 표현하여 UDDI(Universal Description, Discovery and Integration)저장소에 등록하게 된다. 그리고 서비스 사용자는 UDDI 저장소에서 원하는 서비스 상세 정보를 검색하고 확인된 서비스 위치에 접속함으로서 서비스를 호출한다. 또한 서비스를 등록하고 호출할 때 사용되는 프로토콜은 SOAP(Simple Object Access Protocal)를 이용한다.

위와 같은 구성은 서로 약 결합되어 있기 때문에 독립적으로 운영되며 이를 통해 특정 애플리케이션이 다른 애플리케이션에 미치는 영향을 최소화할 수 있고 서로 유연하고 보편적인 통합을 실현하게 된다[1,3,9,11].

3. Asynchronous Javascript and XML(Ajax)

1) 개념

Ajax는 XML에 기반 한 종합기술로 'Asynchronous JavaScript And XML'을 줄인 말이며, JavaScript와 XML을 이용한 비동기 통신 방식을 뜻한다. 그리고 Ajax에 대한 Adaptive Path의 Jesse James Garrett은 "Ajax는 새로운 기술이 아니라 실제로는 몇 개의 기술로 이루어 졌으며, 각각 그 자체로 충분하지만 합치면 새롭고 강력한 방식으로 작용한다." 라고 정의했다.

Ajax는 다음과 같은 기술들을 포함한다. XHTML과 CSS와 같은 표준화된 기술을 이용하여 웹 브라우저에 독립적인 사용자 인터페이스를 제공하고, 문서 객체 모델(Document Object Model : DOM)을 이용하여 동적으로 웹 페이지 변경 및 페이지 요소 간 상호 작용을 제공하며, 또한 XML과 XSLT를 사용하여 데이터 교환 및 처리를 수행하고, XMLHttpRequest를 사용하여 비동기 데이터 통신을 지원한다. 그리고 이러한 모든 것을 바인딩 하는 JavaScript를 통해 제어된다 [17]. 그림 5.는 Ajax 동작과정을 나타낸다.

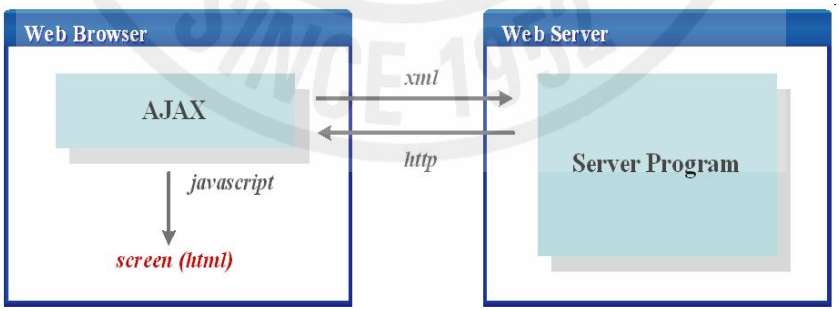


그림 5. Ajax 동작과정

Ajax와 전통적인 웹 서비스와의 차이점은 일반적으로 웹 페이지는 웹브라우저가 HTTP요청을 서버로 보내고, 웹 서버는 그 요청에 해당하는 HTML문서를 전송하고, 웹 브라우저가 HTML 문서를 받아서 표시하는 방식으로 동작하는 반면, Ajax는 웹 브

라우저 위에서 실행되는 JavaScript가 HTTP 요청을 보내서 XML문서로 응답을 받으면 자동으로 XML개체를 생성하고, JavaScript로 XML개체에 접근하여 작업을 하는 방식으로 동작한다[18]. 따라서 Ajax는 HTML 기반보다는 XML 기반으로 메시지를 처리하고 HTTP 프로토콜을 사용하기 때문에 기존 웹 서비스와는 상이한 메커니즘으로 동작되고, 웹 서비스를 사용하는 서비스지향 아키텍처에서는 매우 쉽게 적용된다.

2) 통신방식

그림 6.는 전통적인 웹 애플리케이션과 Ajax 웹 애플리케이션의 통신 방식을 나타낸 것으로 전통적인 웹 애플리케이션은 동기식 통신 방식을 사용하지만, Ajax 웹 애플리케이션은 비동기식 통신 방식을 사용함으로, 빠른 응답성을 제공한다[17]. 따라서 기존 웹 페이지에는 새로운 데이터를 불러올 때마다 전체 페이지를 다시 리로딩 해야 하지만 Ajax는 서버의 처리를 기다리지 않는 비동기 요청 방식이기 때문에 기존 방식과 달리 브라우저의 사용자 화면 구성에 필요한 서비스만을 서버에 호출하고 그 결과를 수신하는 방식으로 데이터양을 줄이는 장점을 갖는다. 그러므로 기업의 업무가 복잡하고 빠른 사용자 인터페이스 처리가 필요시 되는 애플리케이션에서는 Ajax을 이용하여 보다 동적으로 화면을 구성할 수 있게 된다[19,20].

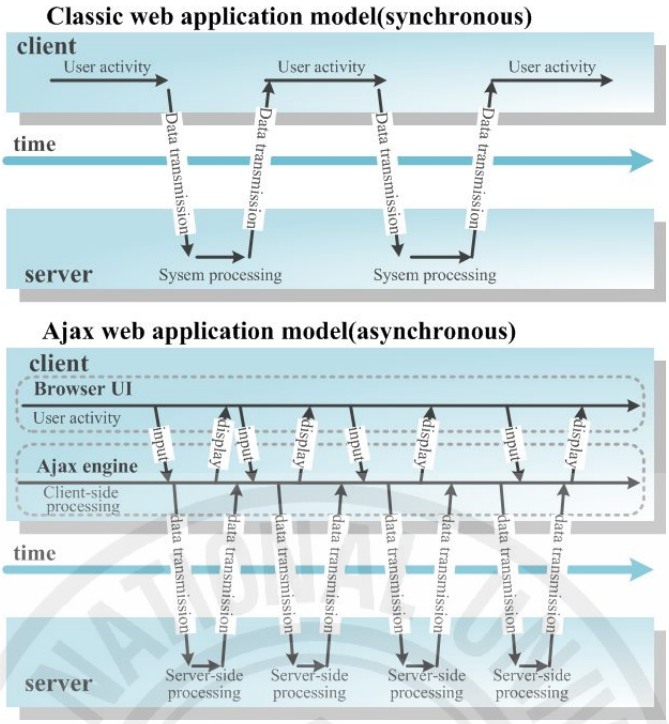


그림 6. Ajax의 비동기 통신 방식

또한 웹에서 해당 서비스를 사용하는데 별도의 프로그램(ActiveX, Flash)을 설치하거나 해당 기능을 갖춘 새 창을 띄울 필요 없이 일반 브라우저 화면에서 그대로 이용할 수 있다. 이처럼 Ajax는 기존의 정적인 HTML문서에서 불가능했던 획기적인 기능들을 가능하게 한다.

따라서 Ajax를 사용하면 XML 데이터 형식으로 응답을 받으므로 전통적인 웹 애플리케이션에 비해 요구된 데이터에 대한 빠른 응답 속도를 서버는 제공한다.

4. Extensible Markup Language (XML)

XML은 1996년 W3C(World Wide Web Consortium)에서 제안한 것으로서, 웹상에서 구조화된 문서를 전송하도록 설계된 표준화된 텍스트 형식이며, SGML의 단순화된 부분집합이지만, 수많은 종류의 데이터를 기술하는데 적용되고 있다.

[21]. 또한 XML은 주로 다른 시스템, 특히 인터넷에 연결된 시스템 간 데이터를 용이하게 교환할 수 있게 하여 HTML의 한계를 극복할 목적으로 만들어졌다. 그리고 가독성과 유연성이 높은 특징을 갖고 있기 때문에 플랫폼에 종속적이지 않고 서비스지향 아키텍처의 대부분이 XML을 사용하여 서비스의 메서드를 정의하고 있다[22].

반면, XML 기술들에는 XML 문서의 형식을 정의하는 Schema, XML 문서 내용을 검색하는 XPath, XML문서를 변환하기 위한 XSLT 그리고 XML 문서의 내용을 데이터베이스의 SQL처럼 질의하기 위한 Xquery 등이 있다[23]. 그림 7.은 XML 기술 간의 관계를 보인다.

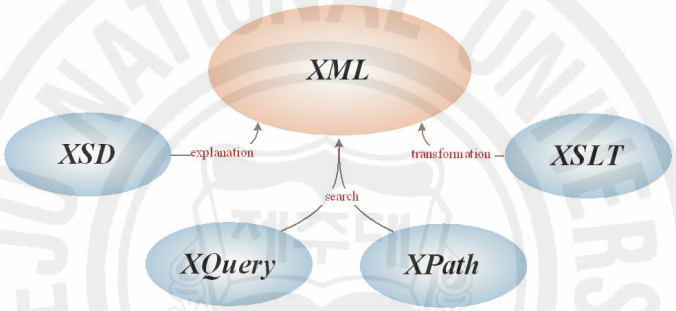


그림 7. XML 관계

1) Schema Definition Language (XSD)

XML 스키마는 XML 객체의 속성과 요소 간 상호 관계의 추상적 표현으로 문서의 스키마는 그 문서의 구조 분석과 각 구조 요소를 정의한다. 예를 들어, 웹 시스템의 경우 웹 시스템 요소와 웹 페이지 요소 및 기타 콘텐츠 요소를 XML 또는 HTML 태그를 사용하여 작성할 수 있다. 또한 XML 스키마는 SGML 문법을 사용하는 DTD와는 달리 XML이 사용되며, 네임스페이스 권고안을 완벽히 지원하고 있다. 그리고 내장 데이터 타입 및 사용자 정의 데이터 타입 기반의 텍스트 요소 콘텐츠의 유효성 검증을 지원한다. 또한 복잡한 콘텐츠 모델과 재사용 가능한 콘텐츠 모델을 용이하게 만들 수 있으며 객체 상속 및 타입 대체와 같은 개념들을 모델링할 수 있게 해준다는 장점을 가지고 있다. 그림 8.과 같이 XML

스키마는 구조, 검증 규칙, 형식 제한 그리고 요소 간 상호 연관성을 제공함으로써 XML 문서 데이터의 무결성을 제공한다.

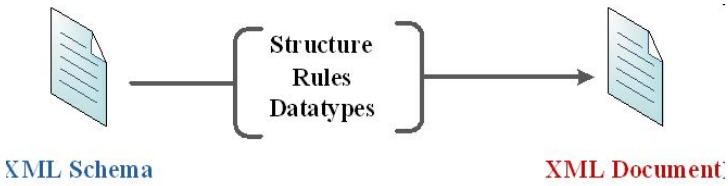


그림 8. XML 스키마가 XML 문서와 연관되는 방법

2) Extensible Stylesheet Language (XSL)

XSL은 변환언어인 XSLT(XSL Transformation)와 출력언어인 XSL-FO(XSL Formating Objects)로 구성되며, XML문서의 스타일을 정의하고 View를 담당하며, 또한 XML문서를 포장하고 HTML의 Cascading Style Sheet(CSS)까지 포함하는 언어이다. 여기서 XSLT는 XML로 작성되는 선언형 프로그래밍 언어이며 XML을 다른 형태로 변환시킨다. 그림 9.는 XML 문서가 XSLT를 통해 HTML, WML등의 문서로 변환되어 웹 브라우저에 디스플레이 되는 과정을 나타낸 것이다[22,23].

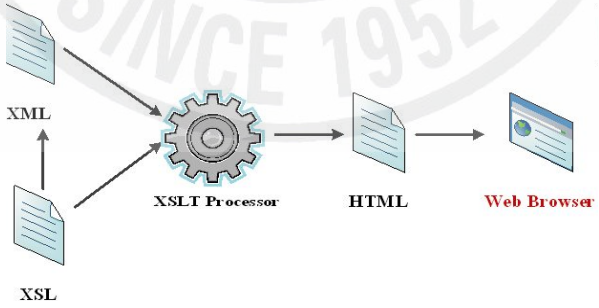


그림 9. XSL Transformation 처리 과정

3) XPath (XML Path Language)

XPath는 W3C의 표준으로 XML문서를 새로운 구조의 XML문서로 변환할 때 기본 문서에서 가져올 특정 노드나 노드의 집합을 지정하기 위한 용도로 사용되는 언어이고, 또한 XPath는 XML 표현보다 쉽고 약어로 되어 있는 특징 때문에 XSL 변환(XSLT)과 XML 지시자 언어(XPointer)에도 용이하게 사용된다. 그리고 XPath는 XML 문서의 노드를 정의하기 위한 경로식을 사용하며, 수학 함수와 기타 확장 가능한 표현들을 갖는다.

따라서 XML 문서의 요소 값은 다양한 XPath 기능을 사용하게 되면 검색 및 필터링 될 수 있으며, 특히 XPath 표현 구문은 다른 유형의 XML 문서 안에 내장될 수 있는 특징을 갖는다[23,24,25]. 그림 10.은 XSLT 스타일 시트에서 사용된 XPath 표현식을 보여주고 있다.

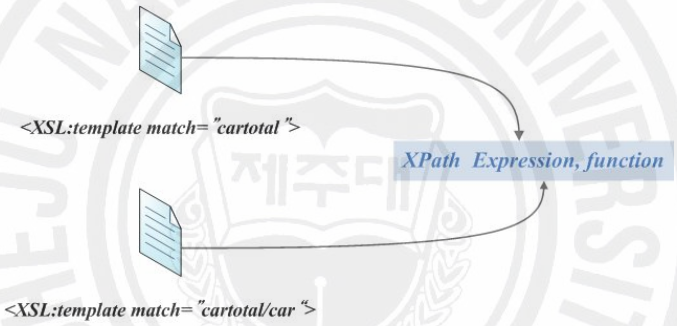


그림 10. XSLT 스타일 시트에 사용하는 XPath 표현

지금까지 분산시스템의 문제점, 서비스지향 아키텍처, Ajax 그리고 XML 관련 기술들을 살펴보았다. 분산 시스템들의 문제점들은 기업에서 제공되는 서비스 간 통합될 수 없는 환경, 중복된 데이터, 상이한 프로토콜 그리고 사용자의 대기와 리로딩의 문제점들을 가지고 있었다. 그러나 이러한 문제점들을 해결할 수 있는 방법은 바로 시스템의 유연성 확보와 재사용이 가능한 서비스지향 아키텍처이다. 그리고 추가적으로 사용자 인터페이스의 편리성은 웹 2.0기술 중 가장 대표적인 Ajax, 그리고 시스템 간 데이터를 용이하게 교환할 수 있는 XML 기술이다.

따라서 본 연구는 위 기술들을 융합하여 분산 시스템들의 문제점들을 해결하고 기업 간 통합을 유도할 수 있으며, 사용자 인터페이스의 편리성을 추구하는 서비스지향 동적 협업 시스템을 제안하고자 한다.

III. 제안 시스템

본 장에서는 동종 기업 간 메타데이터 정보를 통합 관리하며 또한 전체 페이지 리로딩 없이 사용자 질의 및 응답성을 제공하는 서비스 지향 동적 협업 시스템을 기술한다.

1. 시스템 구성도

제안 시스템은 그림 11.에서 보는 바와 같이 Ajax 기술과 서비스지향 아키텍처 개념을 적용하여 3개의 컴포넌트 레이어(Danamic User Interface, Master Server, Service)로 구성되었다.

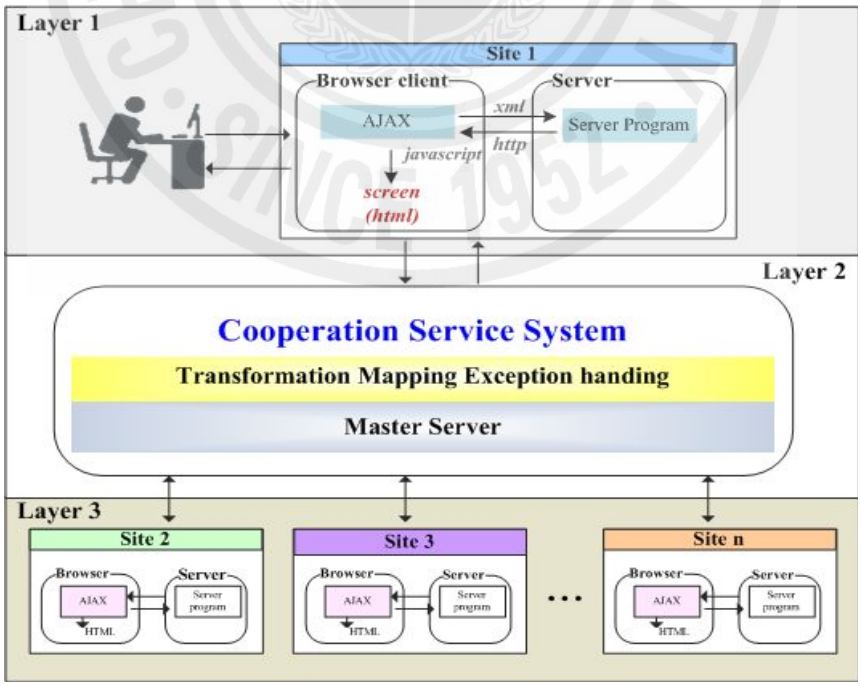


그림 11. 시스템 구성도

1) 컴포넌트 레이어 1 : Dynamic User Interface

클라이언트와 서버 간 비동기 통신 방식인 Ajax 기술이 사용되었으며, 기능은 사용자 편의성과 질의 처리를 담당하며 재전송 요청에 대한 처리는 다음과 같다.

- 사용자가 질의를 요청해 오면 서버는 사용자의 질의에 대한 결과를 로컬데이터베이스에서 검색하고, 그 결과를 사용자에게 반환한다. 그러나 사용자가 요청한 질의에 대한 결과가 로컬 데이터베이스에서 검색되지 못한 경우에는 컴포넌트 레이어 2인 마스터 서버로 랜덤 방식을 통하여 메타데이터를 가지고 오며 또한 마스터 서버의 처리 결과를 받아서 사용자에게 전송하는 역할을 담당한다.

특히 컴포넌트 레이어 1의 전체 동작 과정은 Ajax 기술로 인해 전체 페이지 리로딩 없이 사용자는 처리 결과를 얻을 수 있도록 지원해 준다.

2) 컴포넌트 레이어 2 : Master Server

이는 협업 서비스 시스템이며 마스터 서버이다. 주요 기능은 메타 데이터 등록과 메타데이터 간 변환 그리고 컴포넌트 레이어 1에 의해 위임된 질의 처리를 수행한다. 여기서 메타데이터들은 n개의 협업 시스템들로부터 수집된 필드들의 데이터 집합이다.

- 메타데이터 등록은 특정 기업이 협업 서비스 시스템에 새롭게 가입하고자 할 때 그 기업의 로컬 데이터베이스 및 추가 정보들을 마스터 서버에 등록할 수 있도록 지원한다. 그럼으로써 기업 간 협업에 사용되는 메타데이터들을 관리한다.

- 메타데이터 간 변환은 XML변환기에 의해 컴포넌트 레이어 1에서 위임된 질의를 처리하기 위한 새로운 질의어를 생성할 때 수행된다. 즉, 특정 클라이언트

가 로컬 서버에 질의를 요청하고 로컬 서버에 해당되는 클라이언트의 결과가 존재하지 않을 경우, 로컬 서버는 클라이언트의 질의를 마스터 서버로 재요청하게 된다. 이 때 마스터 서버는 클라이언트의 질의를 협업 기업으로 요청하게 되는데 이를 위해서는 협업 기업에 맞도록 질의를 재구성해야 한다. 따라서 마스터 서버는 메타데이터를 검색하고 협업 기업에 해당되는 질의로 클라이언트의 질의를 수정 변경해야 하고, 변경된 질의를 협업 기업으로 전송해야 한다. 그러므로 컴포넌트 레이어 2인 마스터 서버는 메타 데이터 변환을 통해 질의를 재구성 및 협업 기업으로 요청 그리고 질의에 대한 결과를 요청한 클라이언트로 전송한다. 즉 메타데이터 변환은 XML 변환기의 XML 파서에 의해 질의가 파싱되고 이를 XPath 질의로 재작성 되며, 이는 곧 XML이라는 특징으로 인해 기업 간 정보 공유가 가능하며, 위치에 상관없이 서비스를 사용할 수 있게 된다. 이에 대한 메타데이터 구성은 그림 12.과 같다.

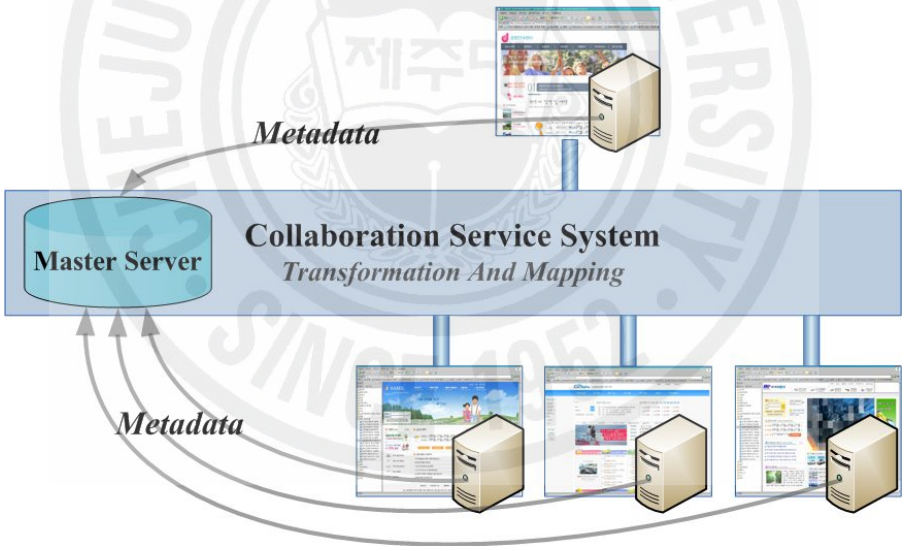


그림 12. 메타데이터 구성

그림 13.는 메타데이터 정보를 DOM 객체의 트리 구조로 표현한 것이며, 이 구조에 따라 DOM 파서는 각각의 XML 데이터에 접근하게 된다. 그럼으로써 제안 시스템의 마스터 서버는 메타데이터를 등록, 검색, 삭제할 수 있게 된다.

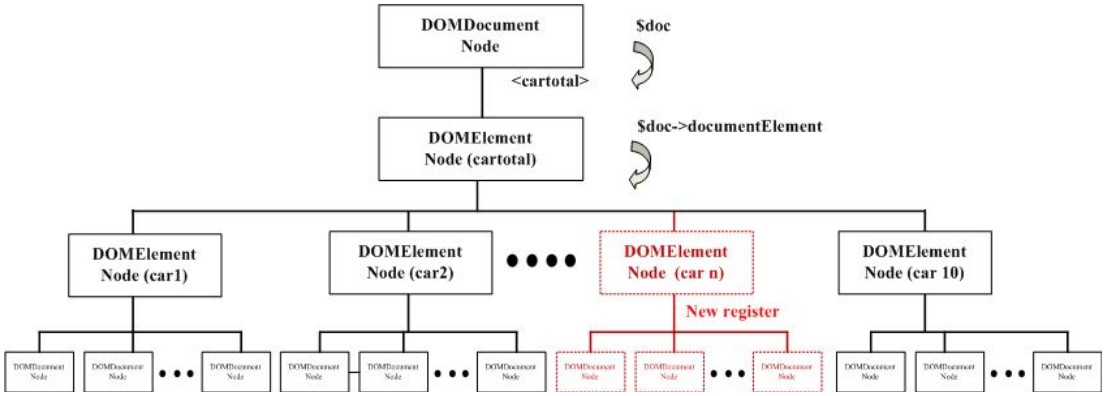


그림 13. 메타데이터 DOM 표현

- 마지막으로, 위임된 질의 처리는 위의 메타데이터 변환 과정에서 기술한 바와 같이 컴포넌트 레이어 1에서 위임된 질의를 협업 기업으로 재 변환하여 요청하고 또한 그 결과를 클라이언트로 전송해 주는 역할을 담당한다.

3) 컴포넌트 레이어 3 : Service

이는 컴포넌트 레이어 1과 동일한 서로 다른 위치의 협업 기업 서비스이다.

2. 메타데이터 형식

제안 시스템에서 사용된 메타데이터는 n개의 협업 시스템들로부터 수집된 필드들의 데이터 집합이라고 앞서 기술하였다. 제안 시스템을 이용하여 동종 기업간 협업하기 위해서는 각 기업의 정보들을 제안 시스템의 마스터 서버에 등록해야 한다. 따라서 메타 데이터에 대한 정의와 그 형식이 요구되며, 본 연구에서는 표 1.처럼 필요한 메타데이터를 도출하였고, 그리고 표 2.처럼 이에 형식을 XML 스키마로 정의하였다. 또한 표 3.는 제안 시스템에 등록될 협업 시스템들의 등록 정보를 XML 문서로 모델링한 내용이다.

표 1. 메타데이터 요소

요소명	Data Type	의미
address	varchar	DB서버 IP 주소 및 도메인 이름
user	varchar	DB서버 접근 ID
pwd	varchar	DB서버 접근 암호
dname	varchar	DB 이름
tname	varchar	테이블 이름
dbtype	varchar	DB 서버 타입
type	varchar	차종
gear	varchar	변속기 타입(자동/수동)
chk	varchar	렌트여부(필드 이름)
author	varchar	등록자 이름
publisher	char	등록 날짜

표 2. XML 스키마

```
<?xml version='1.0' encoding='euc-kr'?>
<xsd:schema xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
    <xsd:element name='cartotal'>
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name='car' minOccurs='1' maxOccurs='unbounded'>
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name='address' type='xsd:string' />
                <xsd:element name='user' type='xsd:string' />
                <xsd:element name='pwd' type='xsd:string' />
                <xsd:element name='dname' type='xsd:string' />
                <xsd:element name='tname' type='xsd:string' />
                <xsd:element name='dbtype' type='xsd:string' />
                <xsd:element name='type' type='xsd:string' />
                <xsd:element name='gear' type='xsd:string' />
                <xsd:element name='chk' type='xsd:string' />
                <xsd:element name='cname' type='xsd:string' />
                <xsd:element name='author' type='xsd:string' />
                <xsd:element name='publisher' type='xsd:string' />
              </xsd:sequence>
              <xsd:attribute name='site' type='xsd:string' use='required'/>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

표 3. XML 인스턴스

```
<?xml version="1.0" encoding="euc-kr"?>
<cartotal>
  <car site="cheju">
    <address>203.253.213.111</address>
    <user>hyesun</user>
    <pwd>hyesun11052</pwd>
    <dname>car</dname>
    <tname>car</tname>
    <dbtype>MySql</dbtype>
    <type>type</type>
    <gear>gear</gear>
    <chk>rent</chk>
    <author>aaaa</author>
    <publisher>2007-10-11</publisher>
  </car>
  <car> ..이하생략...</car>
  <car> ..이하생략...</car>
...이하생략...
  </cartotal>
```

3. 시스템 질의 처리

제안 시스템의 질의 처리 과정은 그림 14.과 같다.

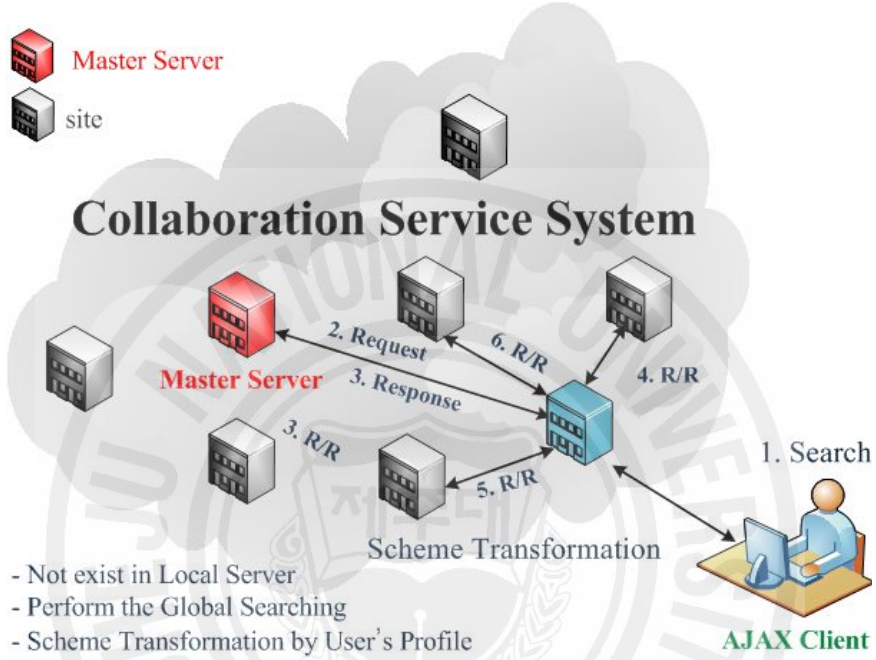


그림 14. 질의 처리 과정

1. 사용자가 Ajax 기반인 웹 서비스 시스템에게 검색 질의를 요청한다.
2. 검색 질의를 요청받은 웹 서비스 시스템은 로컬데이터베이스에서 질의에 해당되는 내용을 검색하고 검색된 결과를 되돌려 준다. 만약 요청 받은 질의가 없는 경우 마스터 서버에 있는 메타데이터 검색한다.
3. 마스터 서버에서 획득된 협업기업의 메타 정보를 이용하여 XML변환기를 통해 SQL 질의를 보내고 시스템에 있는 사용자 질의를 검색한다.
4. 시스템에 사용자가 요청한 질의가 있을 경우 클라이언트에게 질의 결과를 사용자가 이해하기 쉽도록 HTML문서로 변환하여 준다.
5. 5, 6 시스템에 사용자가 요청한 질의가 없을 경우 다시 마스터서버로부터 응답받은 시스템의 메타데이터를 통해 시스템에 검색 질의를 재요청한다.

IV. 시스템 구현 및 결과

본 연구에서 제시된 시스템 구현은 지역적 특성과 그리고 이에 따른 동종 기업들의 운영 현실 등을 고려하여 현재 독자적으로 운용되고 관리되고 있는 렌트카 예약 시스템을 대상으로 구현되었다. 따라서 3장에서 기술된 메타데이터의 범위는 렌트카 예약 시스템의 데이터베이스 필드들로 한정되며, 또한 메타 데이터 변환 과정 역시 동종 기업 간 서로 다른 데이터베이스간의 범위로 한정된다. 구현된 시스템의 명칭은 RaCSS(Rent-a-Car Service System)이며, 이는 기업 간 메타데이터 정보를 통합 관리하며 또한 메타데이터간 변환을 수행하여 기업 간 데이터 정보 공유를 지원하고, 비동기 통신 방식인 Ajax 기술의 사용으로 인해 동적 사용자 인터페이스 및 사용자의 편리성을 제공한다.

1. 동적인 사용자 인터페이스

비동기 통신 방식 기술인 Ajax와 XML기술을 사용하여 사용자의 요청에 빠르게 대응할 수 있도록 동적화면과 또한 검색 결과를 용이하게 확인할 수 있는 검색결과 화면으로 동적인 사용자 인터페이스를 구현하였다.

1)동적화면

RaCSS의 동적화면은 그림 15.과 같다. 그림 15.은 사용자(클라이언트)가 렌트카를 예약하고자 할 때 사용자(클라이언트)의 주민등록번호를 그림 15.의 ① 폼에 입력 후 자동차 type을 선택하는 라디오 버튼을 클릭하면 브라우저는 클릭이벤트를 통해 회원 정보란 또는 적어도 한번이라도 렌트카 예약을 통해 예약한 경우에 렌트카 예약 정보에서 사용자 정보를 추출하여 자동으로 입력된다. 즉,

Ajax 기술에 의해 자동으로 고객정보 입력란이 채워지며, 서버가 입력된 주민번호를 이용하여 사용자 정보를 검색하는 동안 사용자는 대기하지 않고 차량의 유형을 선택할 수 있다. 그리고 그림 15.의 ② 라디오 버튼은 차량의 type 과 gear 선택 영역으로서 차량 type의 라디오 버튼을 클릭하면 원하는 차량 type들에 대한 이미지와 라디오 버튼이 Ajax 기술을 통해 동적으로 표현된다. 이러한 과정은 본 연구에서 제시한 바와 같이 전체 페이지 리로딩 없이 동일 페이지에서 요구하는 정보에 대한 부분만 적용되기 때문에 RaCSS 서버의 부하를 절감시키며 또한 RaCSS 서버의 응답을 기다리지 않고 또 다른 작업을 수행 가능하도록 하는 빠른 응답성 제공임을 알 수 있다.

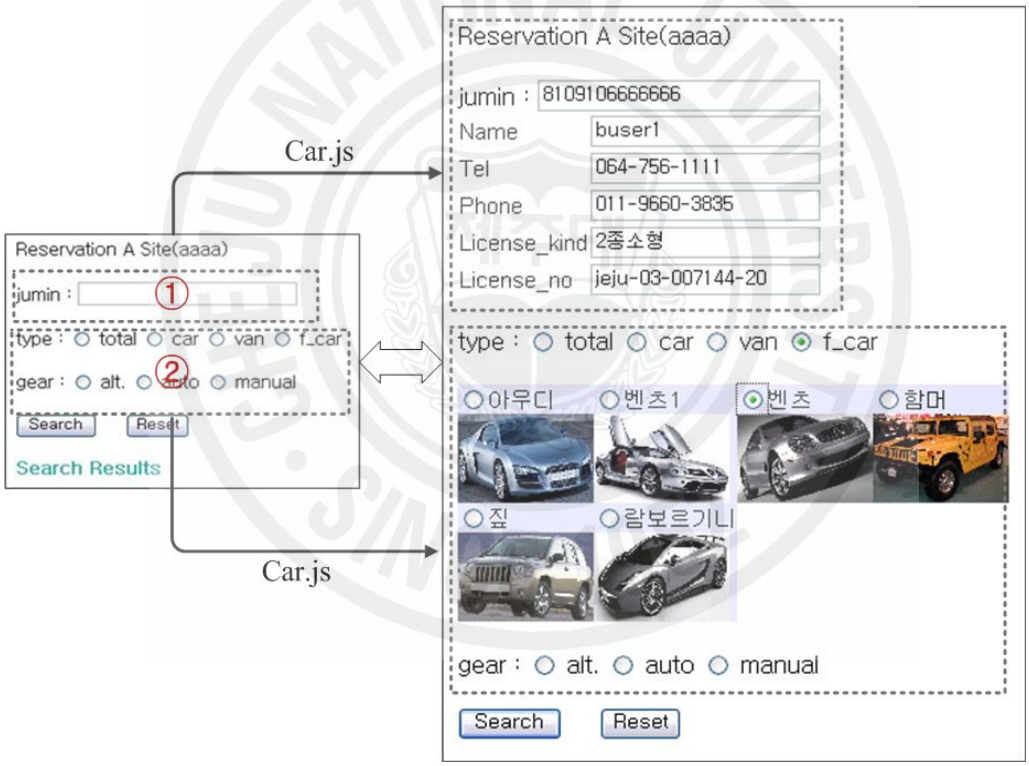


그림 15. 동적 화면

2) 결과화면

또한 RaCss의 결과화면은 그림 16.과 같이 사용자가 원하는 검색결과가 로컬

데이터베이스에 없는 경우 사용자가 요청한 질의를 이용하여 RaCSS에 있는 컴포넌트 레이어 2인 마스터 서버로 질의가 위임된다. 그러면 마스터 서버는 메타데이터 정보를 검색하고 질의를 로컬 데이터베이스에 보내서 재작성한 후 해당되는 사이트에 질의를 전송하고 그 결과를 받아서 클라이언트로 리턴 시켜 준다. 즉, 로컬 데이터베이스에서 사용자가 원하는 차량이 없을 경우 메타데이터 변환을 통해 사용자가 원하는 차량을 검색하고 그 결과를 보여주게 된다.

그림 16. 검색 결과

2. Master server

컴포넌트 레이어 2인 마스터 서버는 크게 메타데이터 관리, 변환 그리고 질의

재작성 및 질의 처리과정을 수행토록 구현되었다. 먼저, 제안 시스템에 신규 서비스 시스템이 등록할 때 신규 서비스 시스템의 메타데이터 정보를 제안 시스템의 규칙에 따라 등록, 검색 그리고 삭제할 수 있는 메타 데이터 관리 기능부터 살펴본다.

1) 메타데이터 관리

메타데이터 관리 기능은 그림 17.에서처럼 RaCSS에 신규 시스템이 등록하는 경우 신규 시스템의 메타데이터를 입력할 수 있도록 입력 폼과 저장된 XML문서를 확인할 수 있는 기능을 제공한다. 제안 시스템에 새롭게 등록하는 경우 자신의 메타데이터를 등록 폼에 입력하고 "***register***" 버튼을 클릭하면 *createElement()*, *createAttribute()* 메소드를 통해 각각의 메타 데이터에 대한 앨리먼트와 속성이 생성되며, 또한 *appendChild()* 메소드에 의해 각각의 메타 데이터들은 앨리먼트와 속성 그리고 Textnode로 구분되어 XML 문서에 추가된다. 또한 이러한 과정은 Ajax 콜백함수 *registeronclick()*를 통해 비동기 방식으로 처리되도록 하였다.

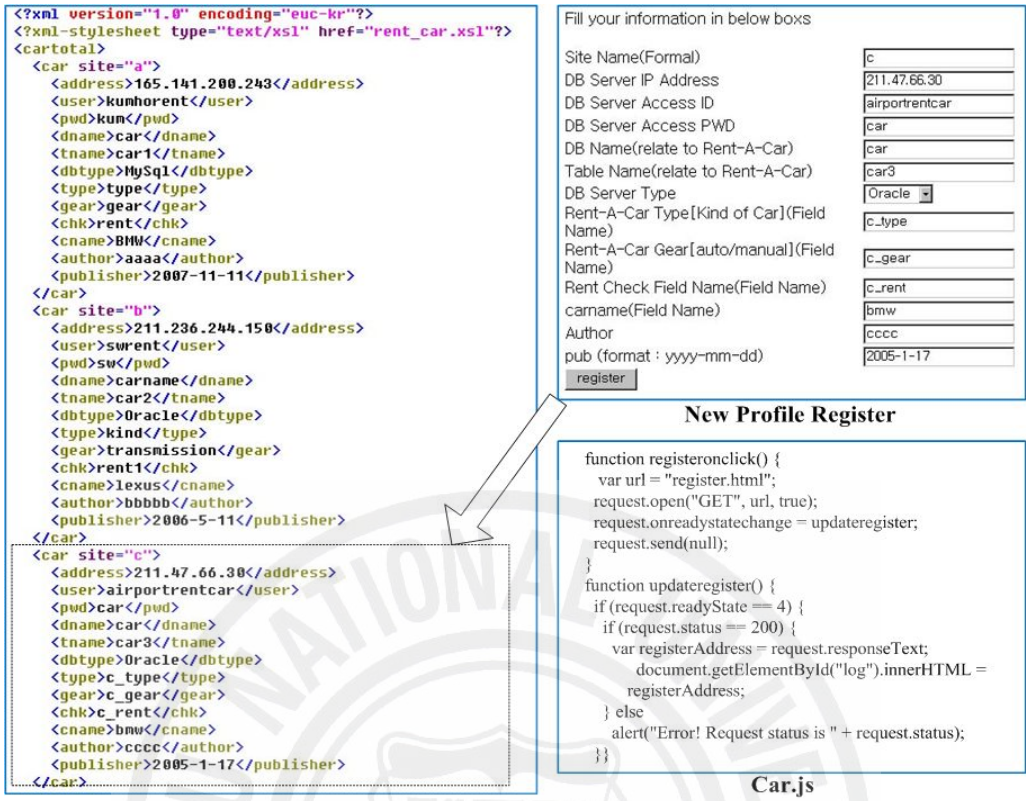


그림 17. 메타데이터 정보 생성

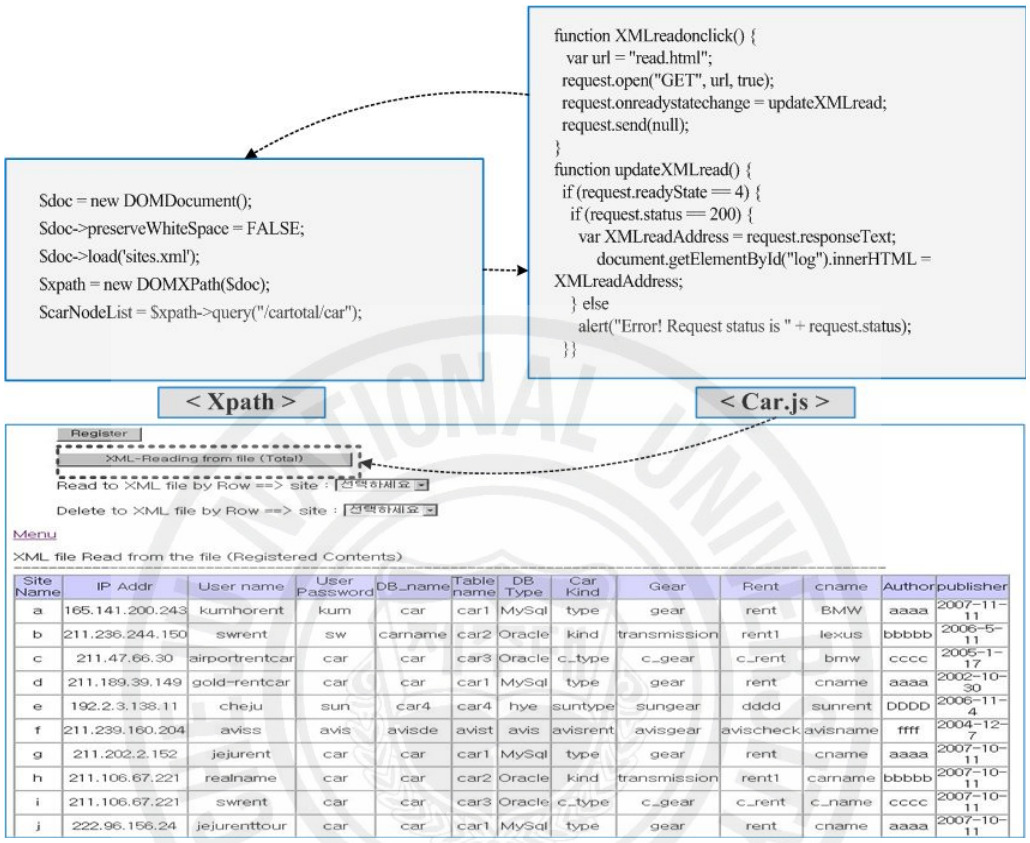
표 4.에는 메타데이터 각각의 요소와 속성 그리고 Textnode들을 생성시키고, 추가 시키는 PHP 프로그램 소스 일부를 나타내었다.

표 4. Element와 Attributes 정보 생성

```
$doc = new DOMDocument();
$doc->preserveWhiteSpace = FALSE;
$doc->load("sites.xml");
$nNode = $doc->createElement("car");
$cNode1 = $doc->createElement("address", $address);
$cNode2 = $doc->createElement("user", $user);
$cNode3 = $doc->createElement("pwd", $pwd);
$cNode4 = $doc->createElement("dname", $dname);
$cNode5 = $doc->createElement("tname", $tname);
$cNode6 = $doc->createElement("dbtype", $dbtype);
$cNode7 = $doc->createElement("type", $type);
$cNode8 = $doc->createElement("gear", $gear);
$cNode9 = $doc->createElement("chk", $chk);
$cNode10 = $doc->createElement("author", $author);
$cNode11 = $doc->createElement("publisher", $publisher);
$nNode->appendChild($cNode1);
$nNode->appendChild($cNode2);
$nNode->appendChild($cNode3);
$nNode->appendChild($cNode4);
$nNode->appendChild($cNode5);
$nNode->appendChild($cNode6);
$nNode->appendChild($cNode7);
$nNode->appendChild($cNode8);
$nNode->appendChild($cNode9);
$nNode->appendChild($cNode10);
$nNode->appendChild($cNode11);
$isbnAttr = $doc->createAttribute("site");
$isbnAttr->value = $site;
$nNode->setAttributeNode($isbnAttr);
$doc->documentElement->appendChild($nNode);
$doc->encoding = "euc-kr";
$doc->formatOutput = true;
$doc->save("sites.xml");
```

또한 메타데이터 검색 기능은 전체 검색과 부분 검색으로 나누어지는데 먼저 전체 검색은 RaCSS의 전체 메타데이터를 검색하는 부분으로 관리자가 제안 시스템에 등록된 모든 협력업체의 전체 메타데이터 정보를 검색할 경우에 사용된다. "***XML-Reading***" 버튼을 선택하면 전체 메타데이터 정보를 검색할 수 있다. 또한 그림 18.는 마스터 서버에 등록 되어있는 메타 정보를 검색하는 과정을 보여주는 것으로 "***XML-Reading***" 버튼을 클릭하면 "*Car.js*" 파일에 정의된 *XMLreadonclick()* 함수를 호출한다. *XMLreadonclick()* 함수는 "*read.html*" 문서를 불러온다. 불러온 *"read.html"* 문서에서는 RaCSS를 통해 저장된 *"sites.xml"* 문서를 메모리에 적재시킨다. 이는 DOM객체로 적재되며, XPath의 Query 함수를 통해 전체 메타데이터 정보를 검색하게 된다. 검색된 결과는

*"Car.js"*에 정의된 *updateXMLread()* 함수를 통해 동일 페이지에서 전체 페이지 리로딩 없이 적용된다.



Site Name	IP Addr	User name	User Password	DB_name	Table name	DB Type	Car Kind	Gear	Rent	cname	Author	publisher
a	165.141.200.243	kumhorent	kum	car	car1	MySql	type	gear	rent	BMW	aaaa	2007-11-11
b	211.236.244.150	swrent	sw	carname	car2	Oracle	kind	transmission	rent1	lexus	bbbbb	2006-5-11
c	211.47.66.30	airportrentcar	car	car	car3	Oracle	c_type	c_gear	c_rent	bmw	cccc	2005-1-17
d	211.189.39.149	gold-rentcar	car	car	car1	MySql	type	gear	rent	cname	aaaa	2002-10-30
e	192.2.3.138.11	cheju	sun	car4	car4	hye	suntype	sungear	dddd	sunrent	DDDD	2006-11-4
f	211.239.160.204	aviss	avis	avisde	avist	avis	avisrent	avisgear	avischeck	avisname	ffff	2004-12-7
g	211.202.2.152	jejurent	car	car	car1	MySql	type	gear	rent	cname	aaaa	2007-10-11
h	211.106.67.221	realname	car	car	car2	Oracle	kind	transmission	rent1	carname	bbbbb	2007-10-11
i	211.106.67.221	swrent	car	car	car3	Oracle	c_type	c_gear	c_rent	c_name	cccc	2007-10-11
j	222.96.156.24	jejurenttour	car	car	car1	MySql	type	gear	rent	cname	aaaa	2007-10-11

그림 18. 메타데이터 정보 전체 검색

메타데이터 부분 검색은 RaCSS에 등록된 특정 시스템의 메타데이터를 검색하는 부분으로 관리자가 원하는 시스템을 ***"Read to XML"*** 폼에서 특정 시스템을 선택하면 Ajax를 통해 동일 화면에서 전체 페이지 리로드 없이 사용자가 원하는 메타데이터 정보만을 검색할 수 있다. 그림 18.은 마스터 서버에 등록 되어있는 메타데이터 정보에서 사용자가 원하는 메타데이터 정보를 보여주는 것으로 site 폼에서 특정 시스템을 선택하면 그림 19.와 같이 Ajax로 정의된 *searchonclick()* 함수를 호출한다. *searchonclick()* 함수는 전체 검색과 같이 *"rentcarsearch1.html"* 문서를 불러온다. 불러온 *"rentcar_search1.html"* 문서에 정의된 것처럼 전체 메타데이터 XML문서인 *"sites.xml"* 문서를 DOM객체를 만

들어서 메모리에 적재시킨다. XPath의 Qurey 함수에 정의된 바와 같이 질의 처리하고 그 결과를 *update search()* 함수에 의해서 동일페이지에 적용된다.

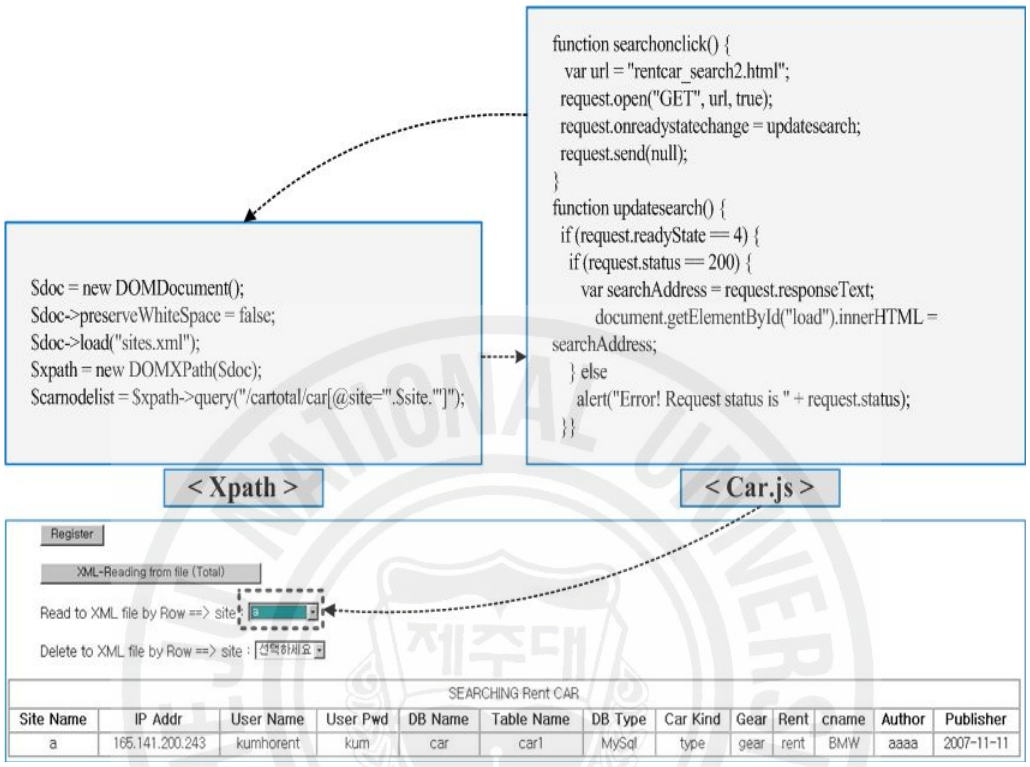


그림 19. 메타데이터 정보 검색

마지막으로 메타데이터 삭제기능은 RaCSS의 마스터 서버에 등록된 특정 시스템의 메타데이터 정보를 선택하여 삭제하는 것이다. 그림 20.은 특정 시스템의 메타데이터를 삭제하는 것으로 ***"Delete to XML file by Row"*** 의 site select Bar 에서 특정 시스템을 선택하면 *onchang* 이벤트가 발생 하면서 *"car.js"* 파일에 정의된 *deletecarnode()* 함수를 호출한다. 호출된 *delete carnode()* 함수는 변수 site에 저장된 파라메타 값을 갖고 *"delete_ok.html"* 문서를 불러온다. 불러온 *"delete_ok.html"* 문서에서 site 변수의 파라메타 값을 이용하여 특정 시스템의 childnode들을 *removechild()* 함수를 이용하여 삭제한다. 그림 21.은 위와 같은 일련의 과정을 보여주고 있다.

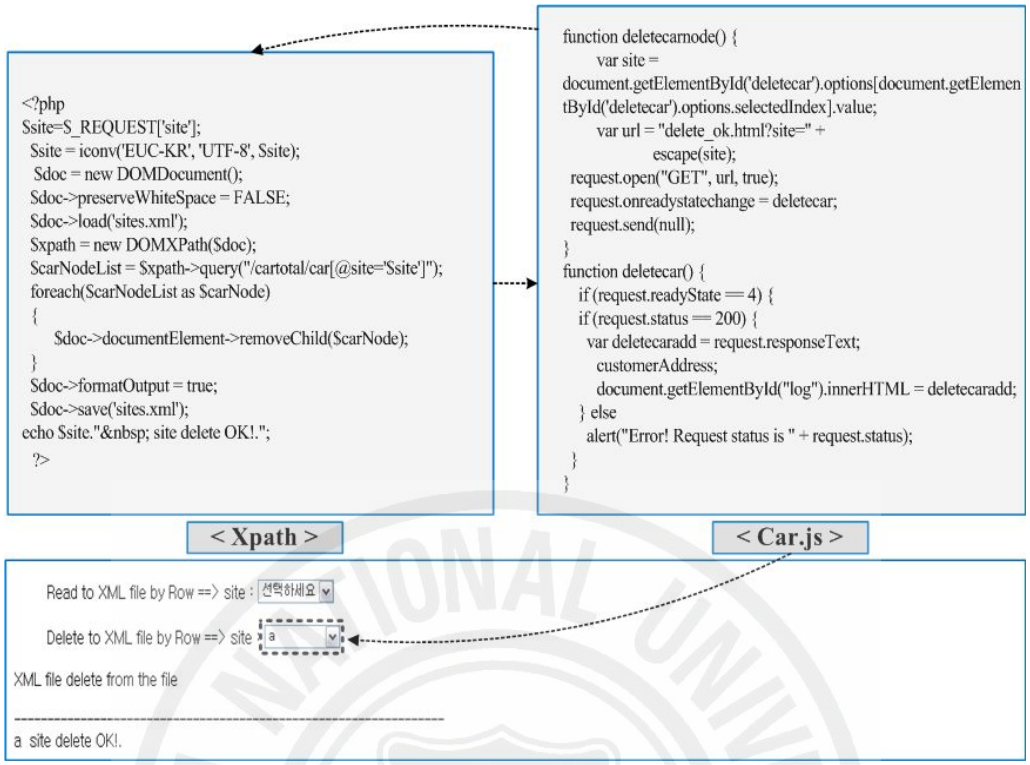


그림 20. 메타데이터 정보 삭제

2) 메타데이터 변환

사용자가 요청한 질의는 *"load.php"* 에서 정의한 모듈들에 의해서 처리된다. *"load.php"* 는 사용자 요청 질의에 대한 결과가 로컬데이터베이스에 없을 경우 *"tran.inc"* 에 있는 *trans()* 모듈를 호출하여 검색한다. *trans()* 모듈은 메타데이터를 검색하기 위해서 *"sites.xml"* 을 메모리에 적재시키고, *rand()* 함수를 이용하여 협업기업들의 메타데이터 정보를 검색하여 XPath의 *Qurey* 메소드 통해 검색된 메타데이터 정보를 추출한다. 메타데이터 변환의 과정은 다음과 같으며, 변환 및 매핑 그리고 질의 재작성과 질의 요청에 대한 처리 흐름도는 그림 21.와 같고 구현된 모듈은 표 5.과 같다.

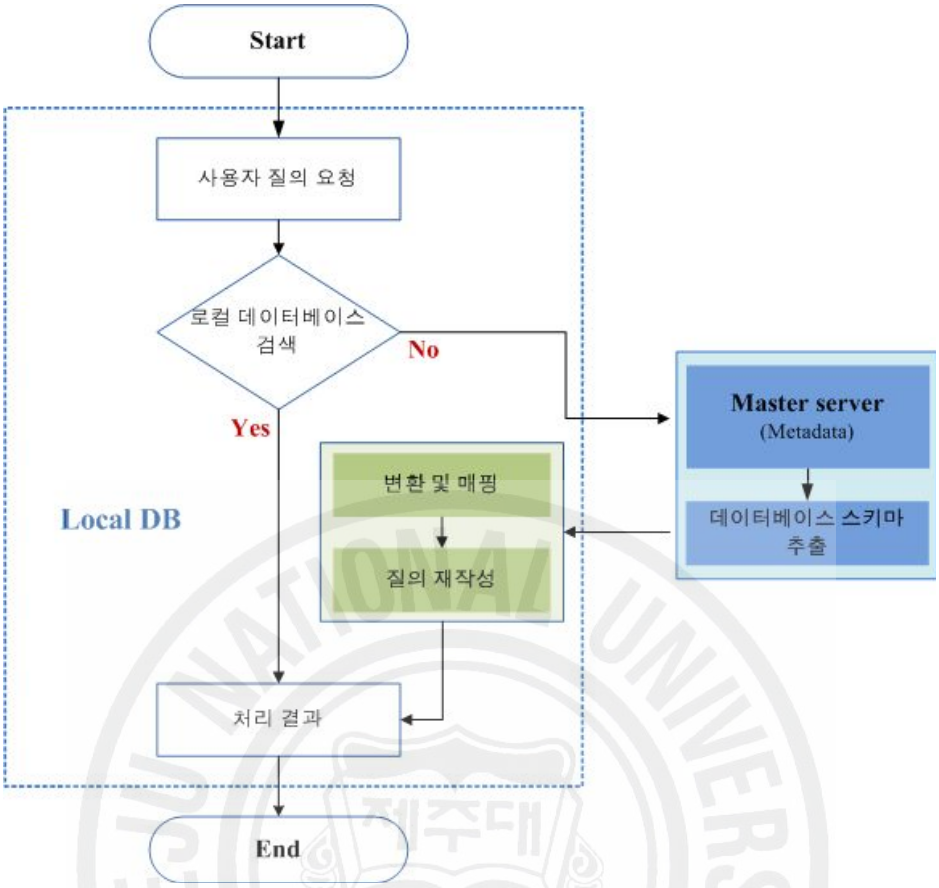


그림 21. 처리 흐름도

- **변환 과정**

1. Start
2. Ajax 기반의 협업시스템에 사용자가 질의 요청
3. 로컬 데이터베이스에서 사용자가 요청한 질의 검색
4. 마스터 서버에서 협업시스템 등록된 협업기업 시스템의 데이터베이스 스키마 추출
5. 마스터 서버에서 추출된 협업시스템 정보를 변환 및 매핑, 질의 재작성
6. 사용자에게 처리 결과를 웹브라우저에 출력
7. End

표 9. 변환 및 처리 모듈

```
// 협업 시스템 검색
while($scount <3) {
      $j = rand(1, $carNodes->length);
      $i = $j - 1;
      $carAttr = $carNodes->item($i)->attributes;
      $carAttrs = $carAttr->getNamedItem('site');
      $carAtr_value = $carAttrs->value;
 if($site != $carAtr_value) {
    $scot = trans($data, $site, $carAtr_value, $name, $ctype, $cgear, $count); }
    $scount = $scount + $scot;
  }
}
// 협업 기업의 메타데이터 추출
function trans($data, $csite, $site, $name, $ctype, $cgear, $count)
       $doc = new DOMDocument();
       $doc->preserveWhiteSpace = false;
       $doc->load("sites.xml");

       $xpath = new DOMXPath($doc);
       if($site == 협업기업명1') { // 메타에이터 변환 및 매핑
          if($ctype == "1") $ctype = "승용";   if($ctype == "2") $ctype = "승합";
          if($ctype == "3") $ctype = "외제";   if($cgear == "1") $cgear = "자동";
          if($cgear == "2") $cgear = "수동";   }

      if($site == '협업기업명2') {
          if($ctype == "1") $ctype = "car";    if($ctype == "2") $ctype = "van";
          if($ctype == "3") $ctype = "f";      if($cgear == "1") $cgear = "auto";
          if($cgear == "2") $cgear = "man";  }

    if($site == '협업기업명3') {  .....  }

// 질의 재작성 및 협업 시스템으로 질의 요청
    $conn = mysqli_connect($address, $user, $pwd, $dbname);
    $query = "select * from $tname
            where $type = '$ctype' and $gear = '$cgear' and $chk ='0'";
```

3) 질의 재작성 및 처리

2)절에서 설명된 바와 같이, 추출된 협업 시스템의 메타데이터 정보를 이용하여 사용자 질의가 재구성 된다. 표 6.은 추출된 협업 시스템에 접속하여 질의를 전송하는 경우로서 각각의 변수들은 추출된 협업 시스템의 메타데이터를 갖고 있다. 즉, *$address*는 추출된 협업 시스템의 데이터베이스 위치 정보, *$user*는 데이터베이스의 사용자 ID, *$pwd*는 데이터베이스 패스워드, 그리고 *$dbname*은 데이터베이스 이름으로 각각 저장되어 데이터베이스에 접속한다. 이와 같이 추출된 협업 시스템의 메타데이터는 협업 시스템으로 질의를 할 때 사용된다. 그리고 협업 시스템은 재구성된 질의문을 사용하여 그 결과를 반환하게 된다.

V. 구현 결과 분석

본 연구에서 제안한 서비스지향 동적 협업 시스템의 성능 분석은 타 개발 방법론간의 비교와 협업 시스템이 효율적으로 운영될 수 있는 가능성을 보이기 위하여 자체의 실행 환경에서 테스트 하였고, 그 내용은 협업 시스템의 처리 결과에 대한 검색 가능성과 Ajax에 대한 효용성이다.

첫째, 제안한 RaCSS 모델과 타 개발 방법론간의 비교 결과를 표 6.에 제시하였다.

둘째, 처리결과에 대한 검색 가능성은 기존 시스템의 경우 사용자가 원하는 결과가 로컬데이터베이스에 없을 경우에는 사용자가 원하는 서비스를 제공 받을 수 없지만, 제안시스템은 로컬 데이터베이스에 원하는 결과가 없는 경우, 협업된 기업들에서 원하는 결과를 검색해 낼 수 있는 가능성을 말한다. 때문에 기존 시스템과는 다르게 검색 가능성을 높여, 사용자 측면에서는 사용자가 원하는 시기에 필요한 정보 활용이 그 만큼 높아진다. 따라서 제안 시스템 환경에서는 보다 정확하고 필요한 데이터를 제공해 줄 수 있게 된다. 또한 로컬 사이트 운영자 측면에서는 원격 접속 사용자에게 필요한 정보를 제공해 줄 수 있기 때문에 정보기술 조직의 경쟁력을 강화 시킬 수 있다.

셋째, Ajax를 이용한 제안 시스템의 효용성은 전체 페이지 리로딩 없이 사용자의 요청을 출력하는지 여부를 분석하는 것이며, 이는 제안 시스템에서 전체 페이지 리로딩 없이 사용자가 요청한 서비스에 대해 기다림 없이 즉각적인 응답을 Ajax를 통해 제공되는지를 보이는 것이다. 효용성 분석을 위하여 Mozilla Firefox 브라우저와 Firebug 디버그 도구를 사용하였다. Firebug는 웹 분석 및 개발 도구이면서, 추가적으로 JavaScript, CSS, HTML 등을 관리하고 수정 편집할 수 있다. 표 7.은 제안 시스템에서 효용성 분석을 위해 사용된 파일로서 페이

지 리로드와 네트워크 트래픽을 분석용으로 사용되었다.

표 6. 시스템 통합 비교 분석

	EAI	B2BI	제안 모델 : RaCSS
통합 대상	기업 내 이기종 애플리케이션의 통합	기업 간의 시스템 통합	동종 기업 간의 서비스 통합
통합 범위	기업 내 애플리케이션에 국한됨	B2B 거래시 발생하는 비즈니스 프로세스를 중심으로 기업과 기업 간, 기업과 B2B e-마켓플레이스간, e-마켓플레이스간의 시스템 통합을 의미함	고객과 기업 사이에 발생하는 서비스를 동종 기업 간 서비스 통합을 의미함
통합 목적	기존 각각의 목적에 따라 각기 다른 시점에서 개발 구축해온 상이한 애플리케이션간의 상관관계를 비즈니스 프로세스 측면에서 제구성하고, 이를 통해 시장 변화에 유연하게 대응할 수 있는 IT 자원 환경 구축	각기 다른 기종의 시스템을 사용하는 기업 간 업무 프로세스가 자연스럽게 연계되도록 지원함으로써 여러 기업이 참여하는 B2B e-마켓플레이스와 각사의 내부 시스템 간 거래에 관여하는 물류, 통관, 결제 등 제 3자 서비스 군을 모두 통합해 협업이 이루어지도록 함	모든 메시지가 마스터서버로 들어오면 XML의 XPath와 Query로 메시지를 추출하여 적절한 행동을 취하게 함 기업이 처할 수 있는 다양한 예외 상황에 대응 가능하고 변화 예측이 가능한 유연한 시스템 통합
발전 방향	비정형 이진 데이터를 통합하며, 비교적 소규모 통합에 사용됨	B2B 시스템은 주로 경매, 역경매, 구매처리 등 프론트엔드 기능을 중심으로 발전해 왔으나 완전한 B2B 서비스를 위해서는 백엔드 시스템 차원의 통합이 필요	동종 기업 간 서비스 상호 공유를 통해 기업에 서비스 이익을 얻을 수 있음

표 7. 분석 파일

파일명	파일 용량
registerindex.html	5KB
Ajax.js	361byte
car.js	7KB
sites.xml	2KB
rent_car.xsl	2KB

그림 22.는 표 7.에 있는 파일들을 Firefox 웹 브라우저에서 읽어 들인 화면이다. 이 페이지는 *"registerindex.html"* 파일로서 새로운 협업 업체가 협업 시스템에 등록하고자 할 때 사용되는 ***"Register"*** 버튼과 협업 시스템에 등록된 모든 업체의 메타데이터 정보를 불러오는 ***"XML-Reading"*** 버튼 그리고 각각의 협업 업체들의 메타데이터 정보를 검색할 수 있는 ***"Read to XML"*** 버튼, 또한 각각의 협업 업체들의 메타데이터 정보를 삭제할 수 있는 ***"Delete to XML"*** 버튼 메뉴로 구성되어 있다.

분석 방법은 각각의 협업 업체의 메타정보를 불러오는***"Read to XML"*** 버튼 선택 시 각각 불러오는 문서의 양에 따른 네트워크 트래픽을 분석하는 것이다. 그림 22.는 협업 업체를 선택하기 전 페이지 트래픽 양(78ms)을 보였고, 그림 23.은 사용자가 협업 업체의 메타 정보를 클릭 했을 때의 트래픽 양(78ms)을 보였다. 그 결과 각각의 경우에 해당되는 트래픽 양은 거의 동일하였다. 이는 현재 페이지가 서버의 갱신된 정보를 가져오지만 전체 페이지 리로딩은 수행하지 않고 사용자가 요구하는 정보만을 Ajax 기술을 통해 가져옴을 알 수 있다. 즉, Ajax 기술이 적용됨으로써, 네트워크 트래픽에 큰 영향을 미치지 않으면서 서비스가 제공됨을 알 수 있다. 따라서 비동기 방식인 Ajax를 이용하게 되면, 객체나 컴포넌트들을 다시 읽어 오면서 생기는 트래픽양이 감소할 뿐만 아니라 서버의 부담까지 감소시켜주게 된다. 그리고 전체 페이지 리로드 없이 사용자가 원하는 정보만 텍스트 형태로 제공해주기 때문에 빠른 응답성을 제공하는 것을 확인할 수 있었다.

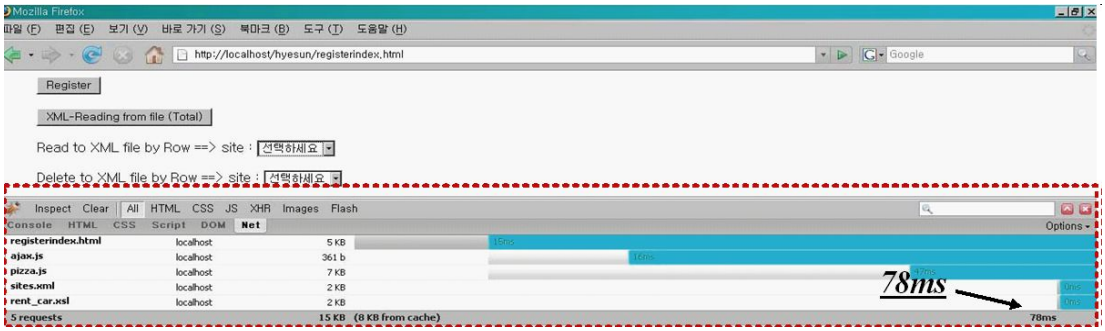


그림 22. 트래픽 분석(1)

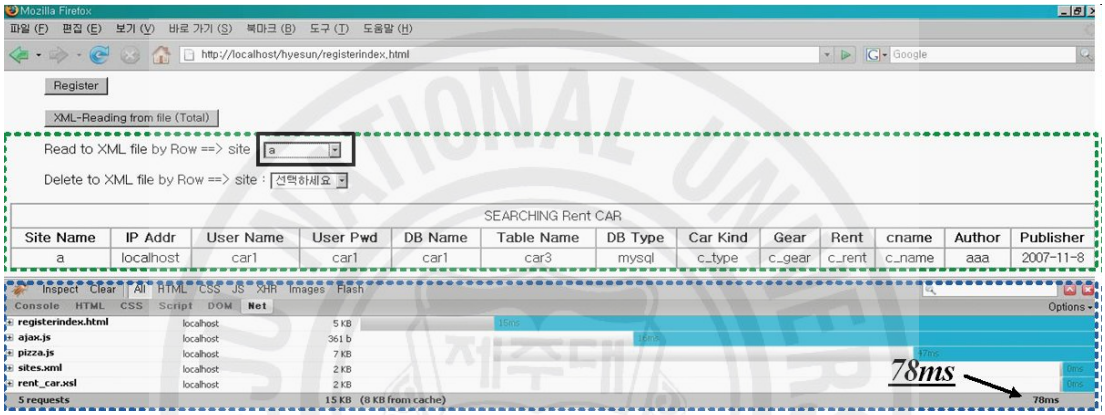


그림 23. 트래픽 분석(2)

VI. 결 론

기업들은 빠르게 변화하는 비즈니스 요구사항에 신속하게 대응할 수 있는 전략과 데이터 공유를 통한 경쟁력 강화가 필요하다. 하지만 기업에서 제공되는 서비스들은 복잡하고 서로 연동되기 힘든 다양한 플랫폼 환경으로 구축되어 재사용성과 타 애플리케이션 간 통합 및 비즈니스 환경 변화에 민첩성, 유연성, 그리고 효율성이 어려웠으며, 변경된 내용에 대한 반복적인 웹페이지 리로딩은 웹 서비스 사용성과 응답성의 문제로 대두되었다.

따라서 본 논문은 이러한 문제점을 해결하기 위해 동종 기업 간 메타데이터 정보를 통합 관리하고 사용자에게 동적 인터페이스를 제공할 수 있는 서비스지향 동적 협업 시스템을 제안하고 구현하였다. 제안된 시스템은 기업 간 메타데이터 정보를 통합 관리하기 위하여 서비스지향 아키텍처 개념을 적용하여 기업 간 업무를 통합하였으며, Ajax 기술을 통해 동적 사용자 인터페이스를 제공하여 사용자의 편리성을 추구하였다. 시스템 구현 결과, 동종 기업 간 업무 연계가 가능하여, 개발 언어 및 플랫폼에 상관없이 기업 간 중복된 데이터 표준화 및 이들을 최소화한 상태에서 서비스를 제공할 수 있었다. 또한 제안 시스템의 효용성 분석을 위해 분석 도구인 Firefox의 Firebug를 사용하여 실시간 응답성과 트래픽 량을 기존 시스템과 비교하여 분석하였으며, 그 결과로 제안 시스템 내의 구성 요소를 통한 새로운 서비스 시스템 구축은 비용 절감 효과와 유지 관리 효율성을 가져다 줄 것으로 기대된다.

향후 연구과제로는 다양한 동종 기업 간 메타데이터 표준화 시 표준 데이터 형식에 대한 정의와 이를 관리할 시스템, 또한 병렬 질의가 가능한 에이전트 기반의 서비스 질의 및 이에 대한 스케줄링 방식, 그리고 Ajax 처리 모듈에 의한 서버의 병목 현상 해결과 Ajax 구현 조건 사항들을 체크할 수 있는 별도의 체크 모듈들에 대한 추가 연구가 요구된다.

참고문헌

[1] Thomas Erl, 'Service-Oreented Architecture : A Field Guide to Integrating XML and Web Services', Prentice Hall, 2004.

[2] by Mathew Quinn, Ajax : Bringing SOA to the Front Lines, SOA Magazine issue Ⅲ, January, 2007.

[3] 박재년, 김유경, 윤홍란, 이혜선, 송영자, 이은정, '웹서비스 기반의 서비스지향 아키텍처(SOA) 구현방법 연구', 한국전산원, NCA Ⅳ-RER-05042, 9, 2005.

[4] 유성수, 노봉남, 'Ajax 기술과 보안', 한국정보과학회 가을학술발표논문집, Vol. 33, No.2(C), pp. 621-625, 2006.

[5] Mark Colan, 'Service-Oriented Architecture expands the vision of Web services, Part 1', available via at http://www.ibm.com/developerworks/library/ws-soaintro.htm, IBM, April, 2004.

[6] Yefim V. natis, 'Service-Oriented Architecture Scenario', Gartner, April, 2003.

[7] Ali Arsanjani, Liang-Jie Zhang, Michael Ellis, Abdul Allam, and Kishore Channabasavaiah, S3: A Service Oriented Reference Architecture, IT Pro, 2007.

[8] OASIS, Reference Model for service Oriented Architecture, committee Draft 1.0, Feurbary, 2006.

[9] 권수갑, 'SOA 개념과 동향', 전자부품연구원, 2005.

[10] By Bhavin Raichura and Shaurabh Bharli, 'Improve Integration Efficiency with Semantic SOA', pp. 67-80, 2007.

[11] 이정민, 'Real Time Enterprise 구현을 위한 Web Services 적용 방안에 관한 연구', 연세대학교, 2006.

[12] Progress.Sonic Movin' SOA On Up:Introducing a New Service-Oriented Architecture Maturity Model, September 19, 2005.

[13] Miko Mastsumura, 'SOA 거버넌스 및 수명 주기 관리를 위한 완벽 가이드

', pp. 11-13, 2007.

[14] 지은희, 'SOA가 바꿔놓을 세상', 한국 소프트웨어진흥원, 2006.

[15] 임철홍, 홍도석, 최정준, 'ESB기반 SOA Application에 대한 S/W Architecture 관점의 평가와 개선 방안에 대한 연구', 한국IT서비스학회지 제5권 제2호 pp. 169-178, 2006.

[16] 김광영, '비즈니스 통합 솔루션의 이해', 현대정보기술, 2006.

[17] Jess James Garrett, Ajax:A New Approach to Web Applications, available via at http://www.adaptivepath.com/ideas/essays/archives/000385.php, Essay, 2005.

[18] 이혁, 'Semantic web 구현을 위한 Ajax', FREELEC, 2006.

[19] 홍철기, 장상현, 황종선, 'Ajax 기반의 SCORM2004 시퀀싱 엔진 및 데이터 모델의 설계 및 구현', 제25회 한국정보처리학회 춘계학술발표대회 논문집 제13권 제1호, pp. 477-485, 2006.

[20] Nicholas C. Zakas, Jeremy McPeak, Joe Fawcett, 'Professional Ajax', wrox, pp. 1-21, 2006.

[21] Extensible Markup Language(XML) 1.0 (Fourth Edition) available via at http://www.w3.org/TR/xml, W3C Recommendation, August, 2006.

[22] Hunter, watt, Rafter, Duckett, Ayers, Chase, Fawcett, Gaven, Paterson, 'Beginning XML 3rd Edition', Adobe E-Book, 2004.

[23] XML Path Language(XPath) Version 1.0, available via at http://www.w3.org/TR/Xpath, W3C Recommendation, November, 1999.

[24] 이나영, '웹 2.0에서 XPath Injection의 脆弱性 分析 硏究', 고려대학교, 2007.

[25] 이승혁, PHP5 Web Programming Guide, 비비컴, 2006.