http://dx.doi.org/10.7236/JIIBC.2013.13.4.55
JIIBC 2013-4-8

Smart Client 기반 BIT 시각화 설계

Data Visualization Design of Bus Information Terminal using Smart Client Platform

김주환*, 남두희**

Joohwan Kim, Doohee Nam

요 약 인터넷이 발달하기 이전에는 클라이언트에서 실행되는 응용 소프트웨어들이 주로 사용되었다. 그러나 클라이언트 기반의 응용 프로그램은 운영체재 또는 다른 소프트웨어와 충돌 문제를 일으켰고, 클라이언트에 배포를 하는데 있어 버전 문제, 운영체재의 상이한 구조 차이에서 오는 실행 오류 등이 존재했었다. 이런 단점들이 인터넷이 발달되며, 클라이언트-서버 기반의 웹 응용 프로그램으로 대체되기 시작했다. 그러나 웹 브라우저가 가지는 기본적인 제약으로 인해 응용 프로그램은 서로 다른 수 많은 언어로 제작되고, 또 그로 인한 실행 성능에 심각한 문제를 야기하기도 했다. 이런 문제점들을 해결하는 대안으로 최근 스마트 클라이언트라는 기술이 대두되고 있다. 스마트 클라이언트는 과거의 리치 클라이언트와 씬 클라이언트가 가지는 장점을 모두 포함하고 있다. 스마트 클라이언트는 기본적으로 웹 서비스 기반인데, 사용자에게 응용 프로그램의 실행과 관련된 기능 일부를 전담하게 하고 서버는 데이터만 받아서 가공하는 개념이다. 본 연구에서는 버스정류장에 설치된 BIT(Bus Information Terminal)를 중심으로 현재 실무에서 이루어지는 개발방식을 비교하고 웹브라우저를 활용한 보다 풍부한 기능 사용이 가능한 스마트 클라이언트 기반 BIT를 설계 적용하였다.

Abstract Smart client is a term describing an application environment which delivers applications over a web HTTP connection and does not require installation and/or updates. The term "Smart Client" is meant to refer to simultaneously capturing the benefits of a "thin client" (zero-install, auto-update) and a "fat client" (high performance, high productivity). A "Smart Client" application can be created in several very different technologies. Over the past few years, ITS has started to move towards smart clients, also called rich clients. The trend is a move from traditional client/server architecture to a Web-based model. More similar to a fat client vs. a thin client, smart clients are Internet-connected devices that allows a user's local applications to interact with server-based applications through the use of Web services. Smart Client applications in BIT bridge the gap between web applications and desktop applications. They provide the benefits of a web applicationwhile still providing the snappy look and feel inherent to desktop applications.

Key Words : web2.0, Bus Information Terminal, BIT, Ajax, Mesh-up

*정회원, 한성대학교 정보시스템공학과
**정회원, 한성대학교 정보시스템공학과 (교신저자)
접수일자 : 2013년 6월 20일, 수정완료 : 2013년 7월 22일
게재확정일자 : 2013년 8월 16일

Received: 20 June, 2013 / Revised: 22 July, 2013 /
Accepted: 16 August, 2013
*Corresponding Author: doohee@hansung.ac.kr
Dept. of Information & System Engineering, Hansung University, Korea

Ⅰ. 서 론

초기 인터넷 웹 페이지에서 사용자 인터페이스 구현을 위한 프로그래밍은 매우 비생산적인 노동 집약형 이었으나, 점차 추상화의 수준을 강화하고 생산성이 향상되는 방향으로 발전을 거듭하게 된다[1].

강력해진 Browser를 기반으로 해서 클라이언트(Client)와 소통을 원하는 기업(Enterprise)은 클라이언트의 다양한 요구를 발전된 웹 응용 프로그램을 이용하여 제품 명세서, 고객 주문서등과 같은 사용자 정보나 기타 기업에 필요한 정보를 입력 받는데 사용하였다. 대부분의 기업들이 클라이언트 정보를 입력받는 Entry Point로 웹 브라우저를 사용하게 되는데, 그것은 클라이언트의 요구 사항이 반영되어 있고, 그것의 효과적인 제어를 위해 많은 양의 클라이언트 스크립트를 사용한다. 이것은 디자인 유지를 위해 매우 중요한 요소 중의 하나였고, 이를 통해 사용자 친화적인 인터페이스를 꽤 만족할 만한 수준으로 구현해 낼 수 있었다

이와 같은 인터페이스 구성 요소들의 유지를 위해 사용된 클라이언트 스크립트는 기능이 복잡해짐에 따라 리소스 증가나 서비스 시간을 증가시키는 주요 요인이 되었다. 따라서, 브라우저가 가지는 이러한 단점을 개선하기 위해, 브라우저 기반이 아닌 응용 어플리케이션에서 웹과 같은 역할을 수행하기 위한 기술들에 주목을 하기 시작했다. 이런 응용 어플리케이션의 대표적인 예로 스마트 클라이언트(Smart Client)를 들 수 있다.

본 연구의 목적은 스마트 클라이언트 기술을 이용하여 BIT를 위한 사용자친화적인 시각화 기법을 위한 인터페이스를 구축하는데 목적이 있다.

Ⅱ. BIT현황 및 문제점

1. 정류소 안내기 구성 및 내용

가. 버스정보시스템

버스정보시스템은 버스차량에 센터와 송수신이 가능한 차내단말기 OBE(On Board Equipment)를 설치하고, 차량의 위치와 정류소 출·도착시간을 송수신하여 정류장에 대기하는 승객들에게 정류소 안내기(Bus Information Terminal)를 이용하여 버스의 도착예정시간을 알려주는 시스템이다.

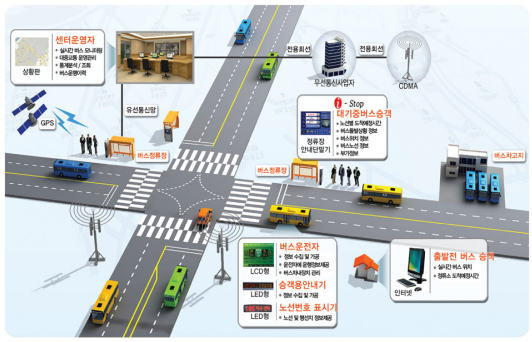

그림 1. 버스정보시스템 구성도
Fig. 1. Bus Information System Structure

정류소안내기는 버스정보시스템의 기본적인 기기로써, 버스정류장에서 대기중인 시민에게 버스 도착정보와 버스위치 및 기타 정보를 제공하는 기능을 수행한다. 하드웨어로는 크게 제어모듈, 표출모듈, 감시모듈, 전원모듈, 기타기능으로 구성되어 있으며 효과적인 버스도착정보 및 버스위치 정보를 제공하기 위해 대형 LCD화면을 채택하고 있다.

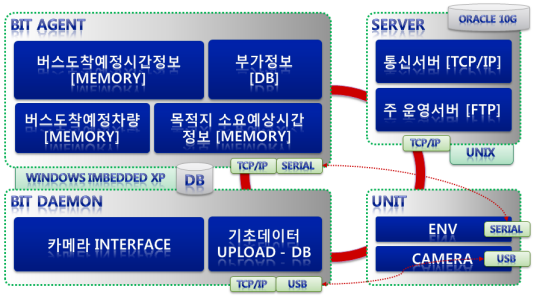

그림 2. BIT S/W 아키텍쳐
Fig. 2. BIT S/W Atchitecture

Ⅲ. 기존 어플리케이션 분석

60-70년대의 시스템구조는 메인 프레임을 중심으로 한 호스트기반의 중앙집중 구조 였다. 이 구조에서는 모든 시스템의 부하가 서버에 작용함으로 시스템의 성능향상을 위해서는 서버를 계속 증설 할 수밖에 없는 구조적인 단점이 있었다. 80년대 후반부터 이런 단점을 해결하기 위해 등장한 시스템구조가 클라이언트/서버형태로 클라이언트 프로그램에 시스템기능의 일부를 수행하게 하

는 구조이다. 90년대부터 클라이언트/서버구조의 단점을 보완하고자 등장한 시스템인 3-티어 구조는 표현기능만 담당하는 클라이언트 영역과 비즈니스 로직만을 수행하는 애플리케이션 서버영역, 데이터베이스기능을 담당하는 데이터베이스 서버영역으로 이루어져 있다. 이러한 구조를 분산시스템구조라고 부른다.

인터넷의 등장은 3-티어구조에서 해결하지 못한 클라이언트 프로그램의 배포 및 관리에 대한 문제까지 해결해주는 기반환경이 되었다. 인터넷환경에서는 웹서비스를 담당하는 웹서버가 별도로 필요하게 되었고 이 구조가 N-티어구조이다. 이는 클라이언트영역의 웹브라우저와 표현영역에 해당하는 웹서버, 비즈니스 로직영역의 애플리케이션서버, 데이터베이스서버로구성이 되고 클라이언트 웹 브라우저만 있으면 3-티어구조에서 프로그램의 배포, 관리, 클라이언트 PC성능에 따라 시스템의 성능에 영향을 미치던 문제까지 해결이 되었다[2].

1. 클라이언트/ 서버 방식의 어플리케이션

클라이언트/서버 방식의 구조는 1계층, 2계층, 3계층으로 나누어진다. 1계층 클라이언트/서버 구조는 사용자와 인터페이스 하기 위한 화면운영, 업무 규칙의 구현, 그리고 자료의 접근 규칙의 운용을 한 개의 서버 시스템에서 수행하는 방식이다. 모든 구성 요소가 한 시스템에서 표현되므로 서버가 비대해지는 현상이 발생하며 서버 부하가 기하 급수적으로 커지게 되고, 새로운 시스템의 도입과 시스템의 확장이 어려운 단점이 있다.

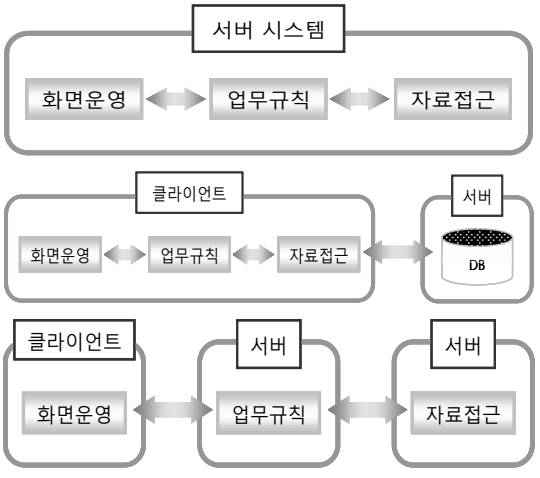

그림 3. 클라이언트,서버 방식 계층 구조(1,2,3)
Fig. 3. Client-Server Structure(1,2,3 tier)

2계층 클라이언트/서버 구조는 기존에 서버가 수행하던 작업 중, 사용자와 인터페이스 하기 위한 화면 운용, 업무 규칙의 구현, 그리고 자료의 접근 규칙의 운용에 대한 작업을 처리하고 서버는 주로 데이터를 관리하는 작업을 하면서 클라이언트와 서버가 물리적으로 서로 독립된 시스템에 존재하는 형태로 구현되었다. 3계층 클라이언트/서버 구조는 사용자와 인터페이스 하기 위한 화면 운용을 클라이언트 시스템이 담당하고, 업무 규칙의 구현과 자료의 접근 규칙의 운용을 서버 시스템에서 담당하는 방법으로 서버와 클라이언트가 작업을 분담하는 형태로 시스템을 운용하는 방식이다. 2계층 또는 3계층 형태로 구현되는 클라이언트/서버 시스템은 사용자에게 GUI(Graphical User Interface)환경을 제공하여 시스템 이용 효율성을 높여주며, 분야별 분산처리, 표준화 및 개방형 시스템 지향 등을 통해 업무처리 환경과 어플리케이션의 발전을 도모할 수 있는 장점을 지니고 있다.

2. 웹 방식의 어플리케이션

개방 네트워크이고 플랫폼에 의존하지 않는 독립된 환경의 모든 브라우저에서 실행되며, HTML이나 XML 등과 같은 문서로 이루어지는 표준기반의 하부구조를 이루고 있다. 웹 어플리케이션 방식은 정적 웹 페이지, Client Scripting in Web Pages, Server Side Script, ActiveX 등으로 구성되어 있다. Script 언어로는 ASP(Active Server Pages), PHP, JSP(Java Server Pages) 등이 있으며, CGI(Common Gateway Interface)를 이용하여 양방향성 어플리케이션을 보다 쉽고 빠르게 개발하는 것이 가능하며, 동적으로 생성되는 어플리케이션의 로직 부분과 실제로 제공되는 페이지 표현묘사 부분의 분리가 가능하다

웹 방식의 문제점을 살펴보면 HTML로는 기능을 표현하기 어렵기 때문에 자바스크립트, VB 스크립트 , 자바애플릿, ActiveX 등 다른 언어와 같이 구현하여 구현하는 과정에서 많은 복잡성이 가지고 있고 사용자가 많은 경우 응답 속도가 느리다. 이는 서버부하로 이어진다[3].

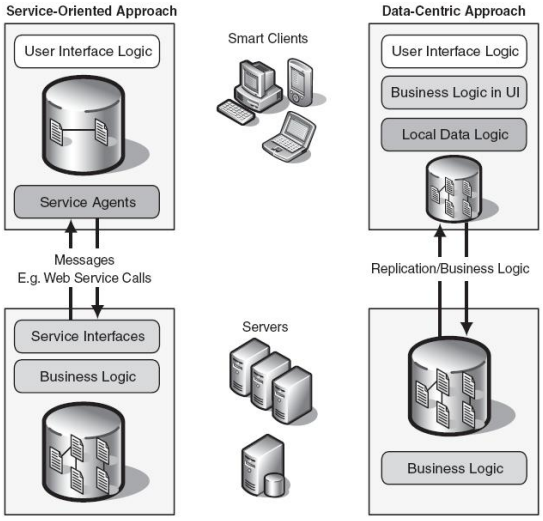

그림 4. Service-oriented approach와 data-centric approach
Fig. 4. Service-oriented approach & data-centric approach

3. 스마트클라이언트

스마트 클라이언트는 리치클라이언트가 가지는 강력한 사용자 인터페이스와 빠른 응답성, 그리고 개발 생산성의 장점과 씬 클라이언트의 쉬운 배포, 적은 메모리 요구사항, 프로그램관리의 용이성 등 양쪽의 장점을 모두 취해서 가지고 있다. 이러한 스마트 클라이언트는 Database Provider, Remoting, XML, 웹서비스 등과 같은 기능을 기본적으로 지원하는데 특히 웹서비스가 제공하는 UDDI와 WSDL을 통해 인터넷상의 서비스 목적과 메시지 포맷과 순서를 정의한 웹서비스를 제공 받고 있다. 스마트 클라이언트의 가장 큰 특징은 로컬 리소스를 활용할 수 있다는 것과 온라인 및 오프라인 모두에서 사용할 수 있다는 것이다.

.NET Framework의 NTD(No-Touch Deployment) 기술로[4] 인해 최종 사용자가 웹 서버에 필요한 구성 요소를 복사하는 방법으로 Windows 기반 스마트 클라이언트 응용 프로그램을 대상 PC에서 배포하거나 업데이트할 수 있다[5].

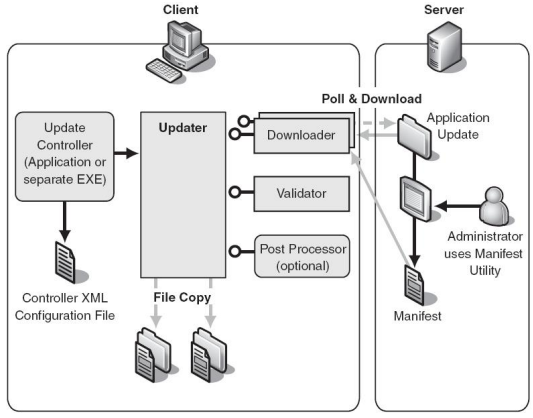

그림 5. Updater Application Block 구조
Fig. 5. Updater Application Block Structure

Ⅳ. 스마트클라이언트 구현

1. 스마트클라이언트 BIT 구성

BIT 내 응용프로그램을 스마트 클라이언트로 구현하기 위해 설치 및 업데이트 관리를 위한 배포 및 게시 서버가 필요하며, 게시된 응용프로그램 내의 Business Logic 부분과 Data Access 부분은 센터가 제공하는 Service를 참조하여 구현한다.

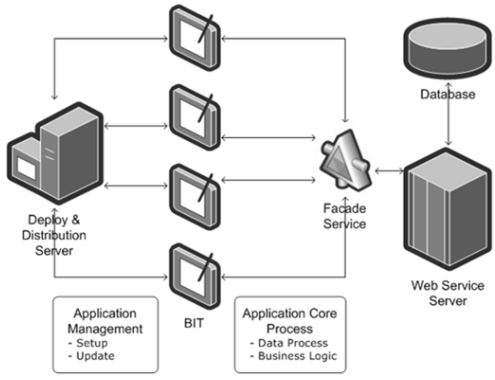

그림 6. 스마트클라이언트 BIT시스템 구성도
Fig. 6. BIT using Smart Clients

서비스는 논리적으로 여러 서비스로 나눌 수 있으며, Facade Service를 DMZ공간에 두고, 인증된 참조만 내부 서비스를 요청할 수 있도록 구성한다.

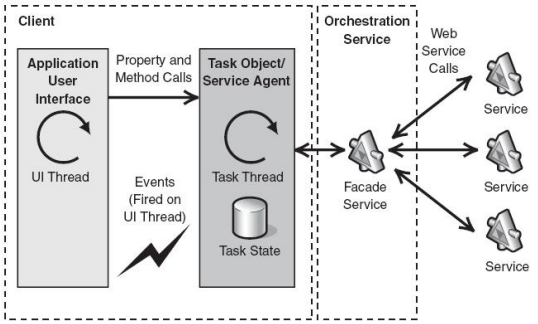

그림 7. Facade Service 에서의 클라이언트 참조 모델
Fig. 7. Client Service Reference in Facade Service

소프트웨어 구성은 다음과 같다. NET 스마트 클라이언트로 개발된 BIT는 서비스 참조를 통하여 Business Logic 및 Data Access 부분이 서버측에 있으므로, Business Logic이 변경되거나 DBMS 등 Data 형식이 변화되어도 BIT내의 응용프로그램을 업데이트 할 필요가 없다. BIT의 UI변경이 독립적으로 이루어질 수 있어, 동일한 시스템에 여러 UI의 응용프로그램을 개발 할 수 있게 된다[6].

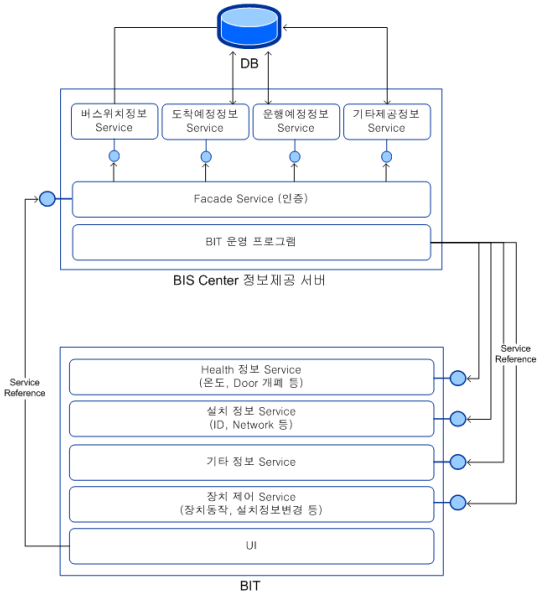

그림 8. 스마트클라이언트 BIT 소프트웨어 구성도
Fig. 8. BIT S/W Structure

2. 효율성 분석 및 기대효과

기대효과로 개발의 독립성, 변경관리 용이성, 유연한 서비스 관리, BIT Data 경량화, Packet 미사용으로 나눌 수 있다. Business Logic 및 Data Access Layer와 독립적으로 BIT 응용프로그램의 UI 변경이 가능하므로, 사용자의 다양한 UI 요구조건에 대응하기 수월하다.

표 1. Server – BIT(Client) 간 프로그래밍 의존성
Table 1. Serer BIT S/W Relationship

구분	Smart Client with Service				C/S with TCP/IP Socket			
	BLL	DAL	UI	Data	BLL	DAL	UI	Data
Server	V	V		V	V	V		V
BIT Client			V		V		V	V

BLL : Business Logic Layer　　DAL : Data Access Layer
UI : User Interface　　Data : Data Definition

기존 TCP/IP Socket 방식의 BIT는 Business Logic 및 Data 형식이 변경될 경우, Server측과 BIT 응용프로그램을 모두 변경해야하므로 독립적인 개발이 어려워지나, 서비스를 이용한 스마트 클라이언트의 경우, 프로그램밍 자체가 상호가 의존적이지 않아 개발의 효율성이 높고 Layer별 요구조건에 유연하다. Business Logic 및 Data Access Layer가 서비스로 되어 있어, 서버 측 서비스만 변경하는 것으로 원하는 변경관리를 할 수 있다. 서비스 참조 모델로 개발한 Smart Client는 서버 측 Data Object를 변경함으로써 클라이언트의 업데이트를 최소화 할 수 있다. 내부에 존재하는 여러 서비스의 End Point 및 서비스 노드밸런싱 등을 라우팅하고 인증하는 Facade 서비스를 구성하므로 내부 서비스 정보가 변경되어도 BIT 응용프로그램을 업데이트 할 필요를 최소화 할 수 있다. 또한, BIT내부에 데이터를 관리하고 저장할 필요가 없으며, 모든 데이터는 서비스를 통하여 서버에서 관리되므로 BIT의 자원을 보다 가볍게 구성할 수 있다.

기존의 BIT의 경우, 통신서버와 BIT간의 데이터 정의에 의해 설계된 Packet을 전달하고 로깅(Logging)하며, 분석하고, 오류를 찾으며, 명령 및 데이터 전달을 위해 정의에 의해 Packet을 조직화 하는 단계가 필요하였으나, 본 연구에서는 스마트 클라이언트가 서비스를 통하여 원격 센터의 객체를 직접 참조 가능하므로, 서버 측 객체를 그대로 사용할 수 있으며, Packet 정의가 변경될 때마다 발생하는 작업의 대부분이 불필요하게 된다. 마찬가지로 센터 측 서버에서 BIT의 서비스를 참조함으로써 BIT를 제어하기가 월등히 수월하다.

V. 결 론

스마트 클라이언트와 웹서비스를 활용하여, 현재의 BIT보다 개발, 관리, 유지보수에서 월등히 유연한 BIT를 개발 할 수 있으며, 이러한 구조의 개발은 개발기간의 단축, BIT 시스템의 하드웨어 원가 및 관리비용 절감까지 기대할 수 있다.

향후, BIT는 안정적인 네트워크 환경을 기반으로, 현재보다 화려한 UI에서 3D 모션까지 구현 가능한 강력한 비디오 자원이 요구될 것이며, 이러한 관점에서, 본 연구를 통해 프레젠테이션 부분, 즉 다양한 View를 제공할 수 있는 기술적 기반과 비디오자원을 보다 강화할 수 있는 BIT 하드웨어의 경제성을 한층 재고할 수 있게 될 것으로 본다.

References

[1] ChrisSells et al.,“Windows Form Programming using C#”

[2] Youngkyung Kim, “Methodology for Web Site Development”, Konkuk Univ. Master’s Thesis, 20005

[3] Jonghyun Park, “Document system using Smart Client”, Korea Univ. Master’s Thesis, 2006

[4] http://www.microsoft.com/korea/msdn/smartclient/understanding/definition

[5] Jungmin Son,“Rapid Deployment using Smart Clients”,2006

[6] http://msdn.microsoft.com/smartclient/

저자 소개

김 주 환(정회원)

• 한성대학교 정보시스템공학과 박사 과정

• 경력

<주관심분야 : TS, BIS, TAGO, 교통공학, 교통시스템>

남 두 희(정회원)

• Univ. of Washington 공학박사

• 경력

미국 워싱턴주 교통계획 감독관

한국교통연구원 책임연구원

<주관심분야 : ITS기술, Uc-City, 통방융합기술>