

## 자바브라우저를 위한 고성능 Rendering 모듈의 설계 및 구현<sup>1)</sup>

진민식\*, 최원호\*, 황옥철\*, 임동기\*, 권오형\*, 정민수\*

\*경남대학교 컴퓨터공학과

### Design and Implementation of high effectiveness Rendering Module For JavaBrowser

Min-Sik Jin\*, Won-Ho Choi\*, Uk-Chul Whang\*, Dong-Ki Lim\*, Oh-Hyung Kun\*, Min-Su Jung\*

\*Dept. of Computer Engineering, Kyungnam University

#### 요 약

자바언어의 특성이라고 할 수 있는 플랫폼 독립성과 객체지향성을 바탕으로 하여 일반 PC 나 IMT 2000 기기, PDA, 인터넷 TV, 인터넷 가전 등의 정보통신 기기와 윈도우, 유닉스, 리눅스와 같은 운영체제 모두 적용할 수 있는 자바브라우저의 개발은 필수적이라고 할 수 있다. 즉, 자바가상머신(JVM, KVM)이 탑재되어 있는 모든 플랫폼에서 동작한다는 특징 하나만으로도 현재 Explorer 와 Netscape 사의 상용 브라우저 사이의 양분화와 호환성문제를 해결해 준다고 볼 수 있다. 그래서, 본 논문에서는 현재 상용의 브라우저보단 기능면에서는 미약하지만 커다란 확장성과 가능성을 가진 자바언어를 가지고 모든 기기에 동작할 수 있는 자바브라우저를 설계하고 구현하였다.

#### 1. 서론

자바(JAVA)는 90년대 중대형 기기의 범용적인 컴퓨팅 환경을 위한 플랫폼의 개발과 기기간 호환성을 가지는 프로그램 언어의 구현을 위하여 Sun Microsystems 사에 의하여 개발되었다. 따라서 자바는 한번의 프로그램으로 어느 기기에서나 독립적으로 수행 될 수 있는 (Write Once, Run Anywhere) 강력한 호환성과 확장성이라는 강점을 갖게 되었다.

- 플랫폼 독립성 혹은 아키텍처 중립성
- 객체지향적 언어
- 컴파일 및 인터프리트 언어
- 단순성과 용이성
- 안정성과 신뢰성
- 안전성

브라우저의 Rendering 모듈은 웹 서버로부터 읽어온 문서를 파싱하고, 전달 받은 파싱 결과를 브라우저의 디스플레이 화면에 맞게 웹 문서를 표시하는 부분이다. 즉, 요청한 웹 페이지를 브라우저 화면에 알맞게 문자열과 이미지, 그리고 자바 클래스 파일의 실행 결과들을 배치한다. 그리고 화면상에 버튼들의 위치와 스크롤의 범위를 정하는 부분이다.

본 논문에서는 자바 언어로 씌어 지고, 객체 지향 언어로 아키텍처 독립적으로 디자인되었을 뿐만 아니라 새로운 포맷과 프로토콜을 취급할 수 있는 능력이 있어 동적으로 연결될 수 있는 자바브라우저의 구현 부분에서 핵심이라고 할 수 있는 고성능의 Rendering 모듈을

구현함으로써 기존의 PC 나 인터넷 TV, 셋탑 박스, PDA 등에 부착되어질 수 있는 브라우저의 구현에 목적이 있다.

#### 2. 관련 연구

본 장에서는 관련연구로서 Sun 사에서 자바로 작성되어진 Hotjava 브라우저에 대해 알아보고, Java 가 아닌 C 로 작성되어진 기존의 상용브라우저인 Mozilla 브라우저에 대하여 알아 봄으로써 Hotjava 에서 HTML 에 삽입되어 있는 Applet 태그가 어떻게 rendering 되는지를 알아보고 기존의 브라우저의 Rendering 모듈은 어떻게 동작하는지를 조사한다.

#### 2.1 Hotjava 브라우저

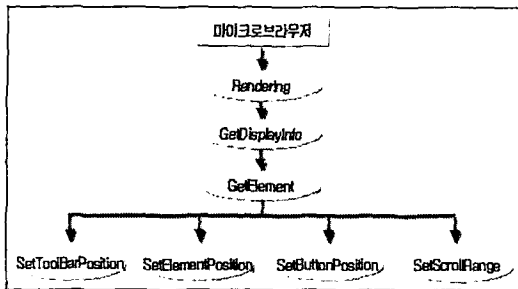
자바로 작성되어진 Hotjava 브라우저가 다른 브라우저와 근본적인 차이라고 하면 필요한 데이터 유형들(type) 프로토콜(protocols) 및 그 사용 방법들에 대한 내용을 브라우저가 일일이 알아야 할 필요가 없다는 것이다. 즉, 컨텐츠 개발 업체들은 각기 다른 자신의 프로토콜과 컨텐츠로 개발하더라도 자바브라우저는 프로토콜 핸들러와 컨텐츠 핸들러를 일단 자신의 파일시스템에서 찾는다. 만약, 없다면 참조할 사이트를 검색하여 해당 프로토콜의 핸들러와 컨텐츠 핸들러를 찾은 다음 자신의 로컬컴퓨터에 이 핸들러들을 다운로드하여 설치하는 방식으로 동작한다. 이러한 이유로 많은 융통성을 제공할 뿐만 아니라 새로운 능력들을 보다 쉽게 브라우저에 부가할 수 있도록 해준다는 것이다. 그리고, 브라우저의 Rendering 은 텍스트와 이미지들을

<sup>1)</sup> 이 과제는 정보통신부에서 지원하는 2000 대학 기초연구지원 사업으로 수행

Java의 AwtToolkitAPI 과 Component 들을 이용하여 사용자 화면에 보여주는 방식으로 처리된다.

### 2.2 Mozilla 브라우저

Mozilla 는 NGT(Next-Generation-Technology)라는 새로운 브라우저 엔진을 이용하여 보다 빠르고 효율적으로 데이터를 사용자의 모니터에 나타내게 만들 뿐 아니라 HTML 4.0, CSS(Cascading Style Sheets), CSS2, DOM(Document Object Model), XML(eXtensible Markup Language)등 웹 표준들을 지원하는 공개 소프트웨어이다.



[그림 1] Mozilla 브라우저의 렌들링 부분의 구성

생성된 트리의 Element 들에 대한 위치를 정한 후 배치하는 메소드와 Scroll 에 대하여 설정하는 메소드와 웹 브라우저에 기본적으로 설치가 되는 ToolBar 에 대한 배치를 정하는 메소드로 구성되어 있다.

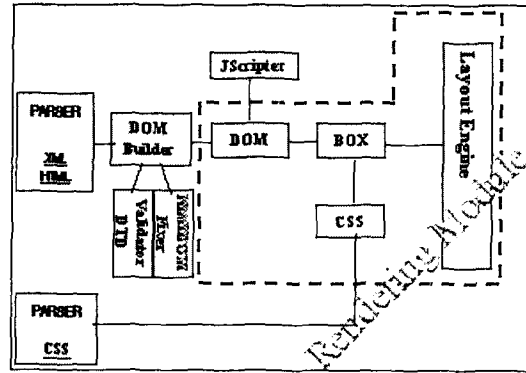
- ◆ Rendering  
파싱 부분에서 생성된 트리와 각 태그의 처리 결과를 전달 받는다.
- ◆ GetDisplayInfo  
HTML 문서를 화면에 표시하기 위해 화면에 대한 정보를 가져온다.
- ◆ GetElement  
화면에 표시할 항목들의 수, 이미지 그리고 자바 클래스 파일에 대한 처리결과와 위치들에 대한 정보를 가져온다.
- ◆ SetToolBarPosition  
마이크로 브라우저 화면에 툴바에 위치를 설정하고, 툴바를 화면에 배치한다.
- ◆ SetElementPosition  
각각의 항목에 대한 크기와 위치에 따라 화면에 배치한다.
- ◆ SetButtonPosition  
버튼에 대한 처리가 있는 경우 버튼의 위치를 설정하고 화면에 배치한다.
- ◆ SetScrollRange  
실제 마이크로 브라우저 화면과 표시될 HTML 문서의 전체 화면의 크기를 비교하여 스크롤바의 범위를 설정 한다.

### 3. 브라우저의 구조와 렌들링 모듈의 설계

### 3.1 자바브라우저의 특징

자바브라우저는 Sun 의 자바기술로 작성된 WebBrowser Component 이다. 어떤 자바 application 에 통합(integrated)될 수 있으며 웹 패드, 인터넷 스크린폰, 디지털 셋탑 박스 등과 같은 인터넷기기에 탑재될 수 있도록 제작되었고 디바이스에 내장되어서 standalone 방식의 자바 응용프로그램으로도 동작할 수 있다.

자바브라우저는 인터넷 상에서 동적인 content handling 처리를 기반으로 하는 아키텍처로 설계되었다. 동적인 content handling 을 위해서 Pilot 이라는 개념을 사용하였으며 각각의 content 들에 대하여 동적으로 Pilot 객체를 생성함으로써 기존 웹 브라우저의 plug-ins 방식과는 다른 개념을 사용하여 content 들을 다룬다.



[그림 2] 자바 브라우저의 framework 구조

### 3.2 자바브라우저의 모듈별 구성

HTML4 Pilot - 이 파일럿이 Framework 의 핵심이다. 현재 많은 content-type 이 존재하지만 아직까지 HTML 과 XML 이 중심적인 역할을 담당하고 있다. text/html 과 text/xml 의 content-type 의 문서를 만나게 되면 동적으로 HTML4 Pilot 객체를 생성하여 content 를 다루게 된다. 이러한 HTML4 Pilot 이외에도 multimedia 를 다루기 위한 JMF Pilot, 3D 모델을 다루기 위한 MD2 Pilot 이 있으며 content-type 에 대한 handler 를 자바언어를 이용하여 작성한 후 객체를 생성시키면 어떤 content 에 대한 처리도 가능하다.

ECMAScript(JavaScript) Scripter - 이 모듈은 ECMAScript(JavaScript)를 처리하기 기능을 담당한다.이 모듈은 Rhino Javascript Engine 을 기반으로 하여 제작되었으며 <SCRIPT>...</SCRIPT>태그사이의 명령을 랩슐화 하여 실질적인 Scripting 언어로 바꾸는 역할을 담당하고 있다.

### 3.3 HTML4 Rendering Engine(HTML4 Pilot)

HTML4 Rendering Engine 은 내부에 여러 개의 Components 로 구성된 모듈이다. 각각의 Part 들은 객체지향개념을 기반으로 작성되었으며 이 모듈의 전체에 영향을 주지 않으면서 어느 한 부분을 수정하거나 대체할 수 있다.

록 구현되었다.

HTML/XML PARSER - 이 파서는 HTML 과 XML 의 입력을 처리하는 역할을 담당한다. 문서 안의 모든 태그들에 대하여 포함관계와 형제관계를 나타내는 data tree(문서 계층 구조)를 생성하고 DOMBuilder 객체와 상호작용 하는 논리적인 Components 라고 할 수 있다.

CSS PARSER - 이 파서는 일단 브라우저가 실행이 되면 브라우저 자체에 내장되어 있는 CSS 파일(기본적으로 적용이 되는 StyleSheet 파일)을 파싱하여 CSSDecl 객체와 CSSRule 객체를 생성하는 역할을 한다. 그리고, HTML/XML PARSER 에서 파싱 중 나타나는 <STYLE >...</STYLE>태그 사이에 있는 스타일 명령들에 대하여 다시 한번 CSSDecl 객체와 CSSRule 객체를 생성한다.

DOMBuilder - 이 모듈은 실질적인 DOM 객체를 만드는(build) 역할을 담당한다. WEB 상의 대부분의 HTML 문서들은 invalid 한 문서들이라고 할 수 있다. invalid 한 문서에서 valid 한 DOM 객체가 생성되지 않으므로 입력으로 들어온 invalid 한 문서들에 대해서는 DTD Validator 와 HtmlDOMFixer 객체를 생성하여 Valid 한 문서를 생성하는 역할까지도 수행한다.

DOM - HTML4 Pilot 의 핵심은 DOM presentation 이라고 할 수 있다. Java application 은 W3C 에서 발표한 DOM API 를 통하여 직접 현재의 문서에 접근할 수 있다. 인터페이스를 구현하여서 현재의 문서의 모든 노드들을 탐색할 수 있다.

DTDValidator/HtmlDOMFixer - 거의 모든 HTML 문서는 invalid 하다. 예로써, HTML 문서에서 파싱 도중<TABLE>태그가 없는 <TR>태그를 만나게 되면 <TABLE>태그를 적당한 곳에 삽입 한 후 <TR>태그를 자식으로 등록하도록 하는 역할을 담당한다.

CSS - Cascading Style Sheets(CSS)는 웹 문서에 style (fonts, colors, spacing)을 적용하는 모듈이다. 문서가 Rendering 되어 질 때 media 에 맞는 style 정보를 제공함하여 문서를 스크린이나 프린터에 보여주거나 보내진다.(CSSAttribs 객체)

Layout Engine - data tree(문서 계층)로부터 생성된 DOM Element 를 CSS 에 적용하여 스크린상에 보여주는 역할을 담당한다.

#### 4. 렌더링(Rendering) 모듈의 구현

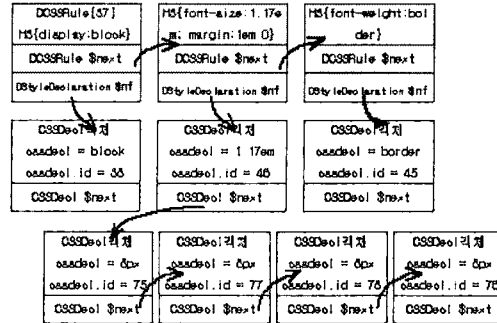
##### 4.1 CSS(CSSAttribs)클래스

이 클래스(객체)는 문서의 Element 들에 대하여 각각 하나씩 존재한다. 하지만, pseudo-element 인 A 태그나 Button 태그에 대해서는 한 개 이상 존재한다. 그리고, Element 마다 생성되는 CSSAttribs 객체는 CSS 스펙에서 나온 속성의 전달과 비전달을 정확하게 구현하기 위해 부모의 속성을 매개변수로 전달하여 CSSAttribs 객체를 생성한다.

이렇게 부모의 스타일 속성이 적용된 CSSAttribs 객체는 스타일 명령들을 저장하는 CSSDecl 객체들 중에서 현재 작업중인 태그에 적용이 되는 CSSDecl 객체를 만들 선

택한 후 CSSAttribs 의 멤버 변수들을 변경한다.

CSSAttribs 클래스는 CSS2 스펙에 있는 모든 문법을 지원하기 위해 각 Element 에 적용이 되어질 수 있는 속성들에 대한 모든 정보들을 클래스 내부에 포함할 수 있도록 클래스 자체에 포함하고 있어야 한다.



[그림 3] DCSSRule 과 CSSDecl 클래스의 자료구조

int display	int position	int top
int bottom	int left	int right
byte align	byte align	byte align
CSSReset oip	int border-top-width	int border-bottom-width
int border-left-width	int border-left-width	int border-left-width
int border-right-width	Color border-top-color	Color border-bottom-color
Color border-left-color	Color border-left-color	Color border-left-color
Color border-right-color	int border-top-style	int border-bottom-style
int border-left-style	int border-left-style	int border-left-style
int border-right-style	int margin-top	int margin-bottom
int margin-left	int margin-left	int margin-left
int margin-right	int margin-right	int margin-right
int padding-top	int padding-bottom	int padding-left
int padding-right	int padding-right	int padding-right
Color background-color	byte background-attach	int background-repeat
int width	int height	int CellSpacing
int cursor	Color Font-Color	short indentation
int x-index	byte decoration	int outline-width
Color outline-color	short outline-style	byte visibility
int page	int font-family	int font-size
int font-weight	int font-weight	int font-weight
Font font	FontMetric.fm	int getwidth() #이

[표 1] CSSAttribs 클래스에 있는 멤버변수

#### 4.2 박스 모델(Box Model)

CSSAttribs 클래스의 멤버변수 중 position 과 display 의 값에 따라서 Element 에 맞는 Box 객체를 생성한다. Box 는 문서 계층(tree)에서 Element 에 생성된 사각형 박스를 기술하고, 보이는 양식화 모델에 따라 배치된다. 생성되는 Box 의 종류에 따라 배치되고 보여지는 방식에 차이가 있다.

BlockBox	ButtonBox	ObjectBox
BreakBox	CssBox	InlineBox
BulletBox	FloatBox	PositionedBox
TableBox	TableCellBox	TableColBox
TableRowBox	TableRowGroupBox	TextBox

[표 2] CSSAttribs 값에 따라 생성되는 Box 객체들

### 4.3 박스의 배치

박스들을 패널에 배치하기 위해 Layout Engine 이 FloatManger 객체를 이용하여 일단 가상의 공간에 박스들을 배치한다. 그리고, 부모관계를 가지고 있더라도 아직까지 부모 Box 의 실질적인 용기블럭의 높이와 너비를 알지 못하는 것들이 있으므로 자식 Box 를 가상공간에 배치를 해가면서 부모 Box 의 너비와 높이를 계산한다.

박스의 특성에 따라서 화면에 보여줄 것인지 아니면 flow 라는 속성을 이용하여 이미지 옆으로 흐르는 텍스트를 보여줄 것인지 등을 여기서 결정한다.

### 4.4 스크린(screen)에 배치

가상공간에 박스들을 한번 배치를 함으로써 부모 용기블럭의 정확한 속성값(너비,높이,배경색)들을 모두 알고 있다. 그런 다음 컨테이너인 패널(Panel)에 다가 부모 Box 부터 차례대로 배치를 한다.

Box 마다 paintBackground(), paintBorder()메소드를 호출하여 배경색, 보더를 패널에 그린 후 Paint()메소드를 호출한다.

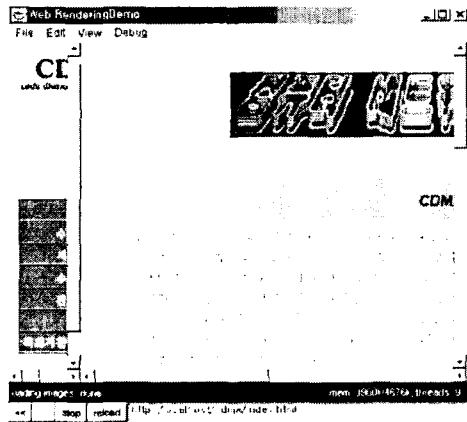
그리고, 패널객체는 스레드(runnable)를 사용하여 구현함으로써 여러 개의 Frame 으로 구성되어 있는 문서에 대하여도 각각의 패널객체를 생성함으로써 동시에 여러 Frame 을 처리하도록 구현하였다.

또, 자바의 GUI 특성중의 하나인 후면 버퍼를 화면에 빠르게 복사하는 방식으로 화면을 채우는 이중버퍼를 사용하여 화면의 떨림 현상을 방지하였으며 사용자가 보기에는 화면이 훨씬 더 부드럽게 갱신되는 것으로 보이게 된다.

할 수 있는 특성들을 모두 지원해 준다고 할 수 있으므로 기존에 다른 언어로 작성되어진 상용의 다른 브라우저보다 많은 확장성과 가능성을 가지고 있다고 할 수 있다. 이러한 자바기반의 브라우저가 갖는 특성으로는 IMT 2000, PDA, 인터넷 TV, 인터넷 가전 등의 정보통신 기기와 윈도우, 유닉스, 리눅스와 같은 운영체제 모두 적용할 수 있다는 것이다.

### 참고 문헌

- [1] <http://java.sun.com>
- [2] <http://www.mozilla.com>
- [3] <http://mobile.daesangit.com/mobile/index.jsp>
- [4] <http://javanuri.co.kr>
- [5] <http://trio.co.kr/webrefer/html/index.html>
- [6] SKTelecom, '꿈의 이동통신 IMT-2000', '중앙연구원'
- [7] 이현우, 김형국, 홍성민 공저, 'Java Programming Bible', '영진출판사'
- [8] <http://www.w3.org/>
- [9] 정보과학회지 2000. 5 이동인터넷



[그림 3] 자바브라우저의 실행 화면

### 4. 결론

이 논문에서 설계하고 구현한 브라우저는 자바를 기반으로 하고 있으며 아직까지 HTML/XML, JavaScript 만을 지원하고 있다. 하지만, 자바가 가지는 여러 이점이라고