

HTML5 환경에서 No ActiveX를 위한 C/C++ 코드 재활용 시스템 설계

최태범*, 박석천**, 오정복***

*가천대학교 일반대학원 모바일소프트웨어학과

**가천대학교 컴퓨터공학과 정교수(교신저자)

*** (주)비주얼인포시스 차장

e-mail:vwctb@daum.net

Design of C/C++ Code Recycle System for No ActiveX in HTML5 Environment

Tae-Bum Choi*, Seok-Cheon Park**, Jung-Bok Oh***

*Dept. of Mobile Software, Gachon University

**Dept. of Computer Engineering, Gachon University(Corresponding Author)

***Deputy General Manager, Visual Infosys co., ltd

요 약

ActiveX는 DLL(C/C++ 컴포넌트) 연동과 로컬 리소스를 사용하여 웹의 한계를 극복할 수 있는 웹 애플리케이션을 기술이다. 최근 까지 국내에서 널리 활용되어 왔지만 호환성과 보안문제로 인해 폐지될 예정이다. 현재 HTML5가 대안으로 고려되고 있지만 지원되는 API 안에서만 자유롭고 DLL(C/C++ 컴포넌트) 호출이 불가능하기 때문에 ActiveX를 완벽하게 대체하기는 사실상 불가능하다. 따라서 본 논문에서는 기존의 ActiveX의 DLL(C/C++ 컴포넌트) 코드를 재활용하기 위해 GNU 빌드 시스템과 통합 빌드 시스템을 통해 코드 의존성을 분석하고 LLVM을 통해 비트코드로 컴파일한 후 최종적으로 Emscripten을 통해 자바스크립트 코드로 재활용하는 시스템을 설계 하였다.

1. 서론

ActiveX 컨트롤은 DLL(Dynamic Linking Library) 연동과 로컬 리소스를 활용하여 웹 애플리케이션을 로컬 애플리케이션 수준으로 웹에서 구동하게 하는 기술이다. 국내 사이트를 이용하다 보면 페이지 상단 또는 팝업으로 표시되는 경고 문구를 쉽게 만날 수 있다. 국내에서는 ActiveX가 보안(27.4%), 인증(18.8%), 결제(15.1%) 등 주로 금융·결제 관련 기능에 사용되지만 해외에서는 주로 멀티미디어(46.9%), 전자문서(32.6%)에서 사용되며 기능에서도 큰 차이를 보이고 있다.

현재 HTML5가 나오면서 기존에 ActiveX로만 가능하던 멀티미디어관련 기능 등을 해결하여 ActiveX를 대체할 대안으로 여겨지고 있지만 DLL(C/C++ 컴포넌트) 연동이 불가능하고, HTML5가 지원하는 한정적인 API안에서만 자유롭기 때문에 기존 ActiveX의 영역인 보안, 인증, 결제, 금융 관련기능 등을 100% 대체할 수 없다[1].

또한 기존에 개발된 수많은 ActiveX 기반 소스코드를 버리고 다시 HTML5환경에 맞는 코드로 새롭게 개발해야 하는 문제가 발생하기 때문에 코드 재활용 시스템이 연구되고 있다. 그러나 수많은 소스코드로 이루어진 프로그램은 각 각의 소스코드 파일별 코드 의존성이 발생하여 대형 소프트웨어는 코드를 재활용하기 어렵다.

따라서 본 논문에서는 기존 ActiveX 소스코드인 DLL

(C/C++ 컴포넌트)를 GNU Build System을 통해 소스 코드 패키지를 포팅하여 통합하고 통합 빌드 시스템을 구축하여 코드 의존성 문제를 해결한 후 LLVM(Low Level Virtual Machine)으로 .bc(Bitcode)로 컴파일하고 이것을 다시 Emscripten을 통해 HTML5환경을 위한 자바스크립트 코드로 컴파일 한다. 따라서 기존 ActiveX 소스코드인 DLL(C/C++ 컴포넌트)를 자바스크립트로 재활용하여 HTML5 환경에서 사용할 수 있으며 이를 통해 새롭게 개발해야 하는 시간과 비용을 크게 줄일 수 있을 뿐만 아니라 HTML5가 가지는 한계를 넘어 완전한 No ActiveX를 실현하고자 한다.

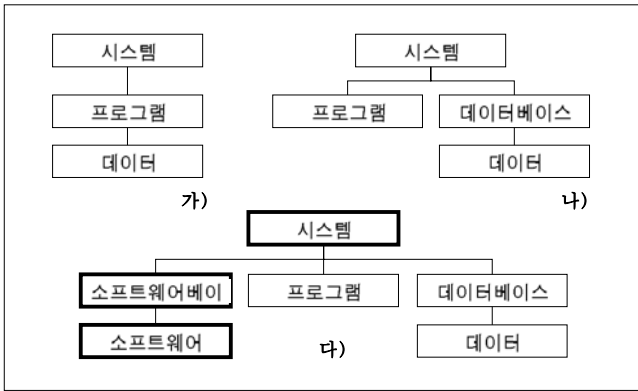
2. 관련연구

2.1 코드 재사용 시스템

코드 재사용 시스템이란 개발에 필요한 소프트웨어(문서포함)와 지식과 관련된 절차를 정형화, 표준화함으로써 이를 반복적으로 사용하는 활동이다.

(그림 1)의 (다)는 재사용 개념을 도입한 시스템의 모습을 보여주고 있다. (가)와 같은 모습의 초기 프로그램에서는 각기 데이터를 가지고 있어 같은 데이터를 중복해서 작성하는 단점과 데이터가 불일치되는 현상들이 발생하였다. 이로 인해 (나)와 같은 데이터베이스 개념이 나타나게

되었다. (다)의 경우 시스템은 소프트웨어베이스에 저장된 소프트웨어부품을 이용하여 데이터베이스와 프로그램에 필요한 기능들을 쉽게 가져다 쓸 수 있다. 이때 소프트웨어 부품이 바로 이전에 만들어진 재사용 부품이다[2].



(그림 1) GNU 빌드 시스템 구성도

2.2 ActiveX 컨트롤러

웹 페이지만으로 사용자에게 제공할 수 있는 서비스는 매우 제한적이다. HTML과 스크립트 언어를 이용하여 PC의 자원에 접근하는 것은 보안상의 이유로 극히 제한되어 있기 때문이다. 그래서 스크립트 언어들의 한계를 극복하고, 다양한 서비스를 제공하기 위해 ActiveX를 사용한다. ActiveX는 스크립트 언어에서 사용할 수 있으며, 로컬 응용프로그램과 동일한 권한으로 실행된다.

<표 1> ActiveX 컨트롤이 발전하게 된 배경

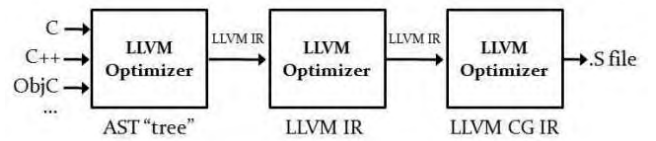
원인	결과
일찍 수립된 공인 인증 체계	코드사인 인증서 배포 1위
인터넷 이용자의 90% 이상이 인터넷 익스플로러 사용	IE 브라우저 독점화에 기술 중속
빠른 브로드 밴드 진입	플로그인 다운로드가 쉬움
비용과 효율만 중요시 하는 형태	고정화된 열악한 국내 웹 생산 시스템 구조
어플리케이션 웹의 발달	정보 제공 수단이 아닌 기능적 웹으로 진화
오락적인 인터넷 사용형태	온라인 게임, 채팅, 사용자 제어

ActiveX 컨트롤은 <표 1>과 같이 다양한 용도와 성능으로 국내에서 다양한 분야에서 사용되고 있다. 그러나 다운로드 된 ActiveX 컨트롤은 해지하기 힘들어 심각한 보안 허점이 있는 경우 제거하는 방법이 없고 ActiveX 컨트롤이 실행되면 그 실행 권한은 쉽게 제거가 되지 않는다. 또한 인터넷 익스플로러 이외의 브라우저에서 실행이 불가능하여 소프트웨어의 중속성이 발생한다[3].

2.3 코드재활용을 위한 컴파일 기술

2.3.1 LLVM(Low Level Virtual Machine)

LLVM은 컴파일러를 구성하는 각 모듈들이 잘 분리되어 있어 컴파일, 링크, 런타임 등 각 시점에서 컴파일 최적화 작업을 수행할 수 있는 컴파일러이다. 특히 JVM 처럼 가상머신 기반의 바이너리 코드를 생성할 수 있기 때문에 특정 CPU에 독립적인 코드의 생성이 가능하다. 하지만 JVM이 Java언어만을 지원하는데 비해 LLVM은 C, C++언어를 지원하고 있기 때문에 기존 C, C++로 작성된 많은 소프트웨어에 적용할 수 있다. 또한 x86, x86_64, PPC, ARM, MPIS 그리고 C 소스 Back-end 등 다양한 Back-end를 지원하고 있다.



(그림 2) LLVM의 컴파일 과정

(그림 2)와 같이 LLVM은 C, C++ 등의 Front-end단의 처리를 위해 GCC(GNU Compiler Collection)를 사용하지만 RTL(Register Transistor Logic)을 중간 결과물로 내보내는 대신 LLVM 가상머신의 규격대로 작성된 LLVM IR을 생성하는 것이 특징이다. 이 파일은 LLVM의 규격에 맞게 작성된 것으로 특정 CPU와는 독립적으로 작성된다. 그리고 마지막으로 시스템 CPU에 맞는 어셈블리어를 생성해주면 이후 실행파일 생성과정은 기존의 도구들을 사용하게 된다. 즉 LLVM의 핵심적인 기능은 GCC의 front-end에서 처리한 결과를 LLVM 가상머신을 위한 LLVM IR로 변환해주는 것이며 이와 같은 특징은 CPU 독립적인 패키지를 제작하는데 기반 기술이 된다[4].

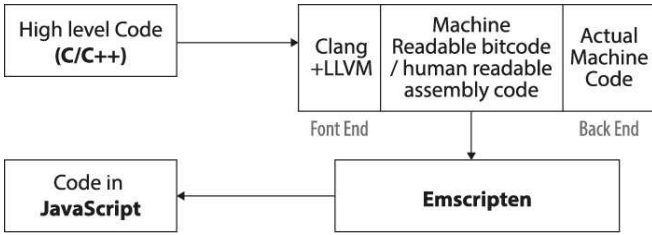
2.3.2 Emscripten

Emscripten은 (그림 3)과 같이 C/C++코드로 만들어진 소스를 LLVM Frontend가 IR로 바꾸면 이를 다시 자바스크립트로 변환하는 역할을 하는 것이다. 이것이 가능한 이유는 heap이 global typed array로 대체하고 쓰레드는 WebWorker로 변경하는 하는 식으로 바꾸어 구동되기 때문이다. 이런 연구는 주로 HTML5와 자바스크립트로 전통적인 애플리케이션 개발을 대체하려는 그룹에서 활발하게 추진되고 있으며, 현재 이쪽에서 가장 두각을 드러내고 있는 Backend는 모질라그룹(Firefox)의 Emscripten으로, C/C++을 IE, Firefox, Chrome 등에서 구동되는 자바스크립트와 HTML코드로 변환해준다.

이미 게임 쪽에서의 성과를 보이고 있다. 대부분의 브라우저에서 WebGL을 지원하고 있으므로 이를 이용해 OpenGL 기반의 PC나 모바일 게임들을 HTML5로 구동한

사례가 속속 공개되고 있다.

실제 언리얼 엔진이 데모 수준으로 웹 브라우저 상에서 구동된 적이 있으며(Epic Citadel), 이때 사용된 LLVM Backend도 Emscripten이다[5].

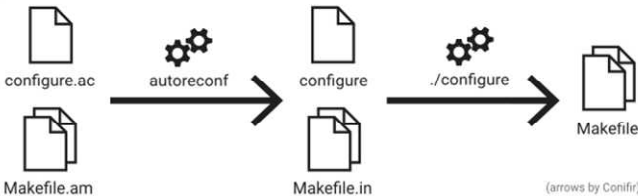


(그림 3) Emscripten 흐름도

2.3.3 GNU 빌드 시스템

GNU 빌드 시스템, 또는 간단히 Autotools는 소스 코드 패키지를 수많은 유닉스 계열 운영 체제로 포팅할 수 있게 도와주는 프로그래밍 도구 제품군이다.

소프트웨어를 여러 플랫폼에서 사용할 수 있게 만드는 것은 어려울 수 있고 C 컴파일러는 시스템마다 다르다. 특정한 라이브러리 함수는 일부 시스템에 존재하지 않고 헤더 파일들은 각기 이름이 다를 수 있다. 이를 관리하는 한 가지 방법은 #ifdef 등의 코드를 이용하여 조건식 코드를 기록하는 것이다. 그러나 다양한 빌드 환경 때문에 이러한 접근은 관리를 불가능하게 만들어가고 있다. (그림 4)와 같이 GNU 빌드 시스템은 이러한 문제를 더 관리할 수 있도록 하고 있다.



(그림 4) GNU 빌드 시스템 구성도

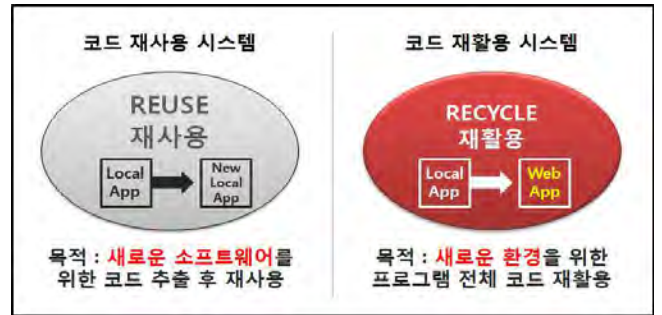
GNU 빌드 시스템은 GNU 툴 체인의 일부이며 수많은 자유 소프트웨어 및 오픈 소스 패키지에 널리 쓰인다. GNU 빌드 시스템을 이루는 도구들은 사유 소프트웨어에 GNU 빌드 시스템의 이용을 허가하는 특별한 라이선스 예외와 더불어 GNU 일반 공중 사용 허가서 하의 자유 소프트웨어로 라이선스 된다[6].

2장 관련연구에서는 코드 재사용을 위한 코드 재사용 시스템을 알아보고 문제점을 분석하여 코드 재사용이 아닌 코드 재활용 시스템을 위한 기술을 제안하기 위해 필요한 기술을 연구하였다.

3. C/C++ 코드 재활용 시스템 설계

3.1 C/C++ 코드 재활용 시스템 개요

본 논문에서는 ActiveX의 DLL(C/C++ 컴포넌트)를 HTML5환경에 적용시키기 위해서는 기존의 코드 재사용 시스템으로는 목적이 다르기 때문에 재사용이 불가능하다. 결국 새로운 개념인 코드 재활용 시스템이 연구되어야 하며 그 개념은 (그림 5)와 같이 새로운 환경을 위한 프로그램 전체 코드를 재활용 하는 것이 코드 재활용 시스템이다.



(그림 8) 코드 재사용과 코드 재활용 시스템의 차이

3.2 C/C++ 코드 재활용 시스템 구조

Emscripten은 C/C++코드를 자바스크립트 코드로 컴파일 해주는 컴파일러로서 C/C++ 코드를 쉽게 자바스크립트 코드로 컴파일 해준다. 하지만 ActiveX의 DLL(C/C++ 컴포넌트)는 경우에 따라 수많은 소스코드 파일로 구성되어 있다. 그렇기 때문에 소스코드 파일 간 코드 의존성이 발생하여 Emscripten만으로는 코드를 재활용하기 힘들다.

따라서 (그림 6)과 같이 코드 의존성 문제를 해결해주는 GNU 빌드 시스템과 통합 빌드 시스템을 도입하여 코드 의존성 문제를 해결하고 LLVM과 Emscripten을 조합하여 코드 재활용 시스템 구조를 설계하였다.



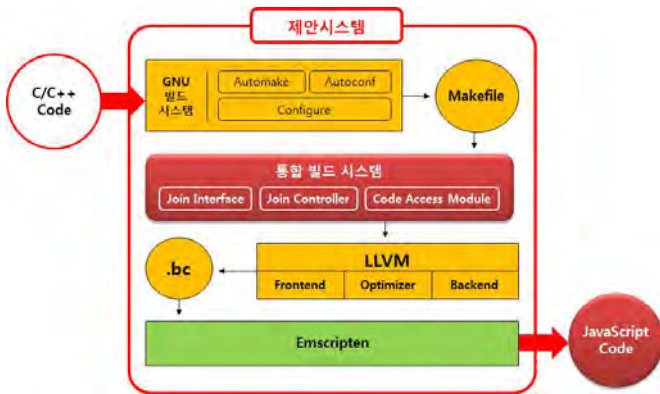
(그림 6) 코드 재활용 시스템 구조

3.3 C/C++ 코드 재활용 시스템 기능블록도

본 논문에서 제안하는 C/C++ 코드 재활용 시스템의 기능 블록 다이어그램은 (그림 7)과 같다. Automake와 Autoconf와 Configure로 구성된 GNU 빌드 시스템을 사용하여 Makefile을 생성한 뒤 Join Interface Join Controller Code Access Module로 구성된 통합 빌드 시스템으로 의존성 문제를 해결하였다. 그리고 통합 빌드 시스템을 통해 하나의 파일로 결합한 뒤 LLVM으로 .bc를 생성하고 마지막으로 Emscripten을 통해 JavaScript 코드로

컴파일 한다.

참고문헌



(그림 7) C/C++ 코드 재활용 시스템 기능블록도

[1] 140114조간 [보도] 미래부, “2013년 웹 호환성 실태조사 발표 자료”, 2013
 [2] 이경환, “객체지향과 소프트웨어의 재사용”, 정보학회지, 1990
 [3] 김재현, “ActiveX 컨트롤의 보안모델 설계”, 석사학위논문, 2006
 [4] 남현우, 김수현, “CPU 독립적인 리눅스 패키지의 설계 및 구현”, 한국컴퓨터종합학술대회, 2010
 [5] Javascript LLVM, 그리고 emscripten, http://social.gabia.com/index.php?document_srl=11970#sthash.Lwr5lGJ8.dpuf
 [6] http://ko.wikipedia.org/wiki/GNU_빌드_시스템

4. 결론

ActiveX는 DLL(C/C++ 컴포넌트) 연동과 로컬 리소스를 사용하여 웹의 한계를 극복할 수 있는 웹 애플리케이션을 기술이다. 최근 까지 국내에서 널리 활용되어 왔지만 호환성과 보안문제로 인해 ActiveX는 폐지될 예정이며 이를 대체하기 위해 노력중이다.

현재 HTML5가 ActiveX를 대체할 대안으로 고려되고 있지만 HTML5가 지원하는 API 안에서만 자유롭고 DLL(C/C++컴포넌트) 호출이 가능하기 때문에 ActiveX를 완벽하게 대체하기는 사실상 불가능하다.

따라서 본 논문에서는 기존의 ActiveX의 C/C++ 코드를 재활용하기 위해 GNU Build System과 통합 빌드 시스템을 통해 코드 의존성을 분석하고 LLVM을 통해 Bitcode로 컴파일한 후 최종적으로 Emscripten을 통해 Javascript 코드로 재활용하는 시스템을 설계 하였다.

향후 본 논문의 C/C++ 코드재활용 시스템의 구현과 기존에 개발되어진 ActiveX기반 공인인증서 및 보안관련 애플리케이션을 재활용해야 하며 재활용된 자바스크립트 코드의 유일한 약점인 속도 문제를 향상시키는 자바스크립트 최적화 연구가 필요할 것으로 사료된다.

사사의 글

본 연구는 미래창조과학부의 2015년 고용계약형 SW석사과정 지원 사업(과제번호:H0116-15-1003)으로부터 지원받아 수행한 결과입니다.