

윈도우리스 액티브엑스 컨테이너의 구현

김아현*, 최상일*, 김영수*, 오금곤**, 김영동***

*조선대학교 제어계측공학과

** 조선대학교 전기과 교수

***조선대학교 정보제어계측공학과

Embodiment of Windowless ActiveX Container

*A-Hyun Kim, *Sang-il Choi, *Young-Su Kim, **Kum-Gon Oh, ***Young-Dong Kim

*Dept. of Control and Instrumentation Eng., Chosun University

**Dept. of Electrical Eng., Chosun University

***Dept. of Information, Control and Instrumentation Eng., Chosun University

ABSTRACT

The aim of this research is to bring flash-file that we have worked to place we want worked in flash and to design Active container that removed window frame in web like general movie player.

It being low price, Anyone using, we intend to design container which removed window frame by processing background of function.

Key word : Windowless ActiveX Container, OLE

1. 서 론

고사양의 PC등장과 다양한 초고속 인터넷망 개발 및 보급으로 많은 사람들이 강화된 그래픽 환경에서 인터넷 웹서핑을 하는 추세이며, 배너 광고 기법 및 응용프로그램의 UI는 가독성을 높이고 유저에게 보다 친근하게 발전하였다. 인터넷에서 콘텐츠와 응용프로그램의 로직이 성공을 좌우하는 만큼, 그것을 유저에게 표출해주는 UI나 프리젠테이션 기법은 더 더욱 중요한 위치를 차지한다. 인터넷이 활성화되는 초기에는 문자, 음성, 음악, 그림등의 단순한 콘텐츠를 담을 수 있는 HTTP 그리고 웹브라우저가 중요한 역할을 담당했다. 그러나 20세기 말을 기점으로 플래시 같은 멀티미디어 콘텐츠가 일반화되고 있어 최근에 플래시 무비를 활용한 콘텐츠가 빠져있는 사이트는 거의 볼 수 없다 해도 과언이 아니다. 플래시에서 지원하는 액션 스크립트를 활용해 게임, 영화, 교육콘텐츠, 광고 배너 등 다양한 콘텐츠들이 제

공된다.

최근에는 xml, soap 프로토콜과 기본 UI 컴포넌트를 제공하여 Rich Client Application을 개발할 수 있도록 변모하였다. 플래시는 콘텐츠를 제작할 때 .SWF에 .EXE등 몇가지 형태로 콘텐츠를 퍼블리싱 (publishing)할 수 있는데, 기본적으로 flash.ocx (ActiveX Control)가 미리 사용자 PC에 설치되어 있어야 하고, 이를 통해 퍼블리싱한 플래시무비를 전형적인 프레임형태로 모니터상에 보여줄 수 있다. 최근에는 응용프로그램에 스킨 개념을 도입해 다양한 형태의 UI로 사용자의 흥미와 프로그램의 기능을 쉽게 활용할 수 있게 발전하고 있다. Media Player와 Third-Party 웹브라우저에서 그러한 시도가 있으며, 이는 사용자들로부터 좋은 호응을 받고 있다. 하지만 이러한 시도들은 대부분 정적인 형태의 외형을 가지고 있다. 그러나 일정한 정해진 틀이 없이 표현될 경우 그 무비의 배경은 투명해질 수 있고, 일정한 사각형의 프레임이 없기 때문에 그 형태가 꼭 직사각형이 아니어도 되게 된다. 또한 실행을 시킬 때 프레임을 생성 할 필요가 없기 때문에 생성할 오브젝트의 크기나 생성 시간을 줄일수가 있다. 하지만 기존 웹기술에서는 프레임이 없이 플래시를 표현하는데는 몇가지 제약이 있었다. 따라서 세계적으로 3개의 업체에서만 이 윈도우 프레임이 제거된 플래시의 표현을 가능하게 한 컨테이너를 판매하고 있다. 이와 관련된 기술로는 자바에서 지원되는 새창 열기 기술이 있는데 이는 윈도우 프레임 안에서 구현된다. 따라서 웹브라우저상에서 자유롭게 구현 가능한 형태가 필요하다. 본 연구에서는 플래시에서 작업한 플래시 파일을 그대로 가져와서 일반 동영상 플레이어처럼 웹상에서 연동이 가능한 윈도우 프레임이 제거된 ActiveX 컨테이너를 구현한다. 저가로

누구나 쉽게 사용 가능하고, 기능상으로는 배경을 투명으로 처리하여 윈도우 프레임을 제거한 컨터이너를 구현하고자 한다.

2. 실행연구

2.1 OLE(Object Linking and Embedding)

1980년대 초, IBM과 애플은 데스크탑 컴퓨터에서 사용할 수 있는 운영체제를 발표하였다. 이러한 운영체제는 멀티태스킹이 가능한 것이었으므로 하나 이상의 어플리케이션을 동시에 실행할 수 있었으며, 시스템 규모의 클립보드 기능을 지원 하였다. 이러한 클립보드 기능을 사용하면 “잘라 붙이기” 방법으로 다른 어플리케이션과 데이터를 공유할 수 있었으며, 저장된 데이터가 다른 형식으로 남겨질 수 있다는 것도 중요한 점이였다. 이를 통해 소스 어플리케이션이 하나의 형식을 다른 어플리케이션에 다른 형식으로 전달하는 것이 가능해졌다. 어플리케이션간의 자료공유는 최근 더욱 발전하여 OLE(Object Linking and Embedding)이라는 기술이 윈도우에 첨가되었다.

마이크로소프트가 1991년에 발표한 새로운 컴포넌트 스펙으로, OLE 1.0은 기본적으로 하나의 도큐먼트에 텍스트, 그래픽, 비디오, 사운드 등의 여러 가지 포맷을 함께 넣어 저장하는 것인 복합 도큐먼트(compound documents)를 다루기 위한 방식이다. OLE 1.0은 어플리케이션 중심의 접근이 아닌 도큐먼트 혹은 데이터 중심의 접근 방식에 대한 진일보였다. 그러나 OLE 1.0의 엄청난 복잡성과 급곡한 학습곡선의 요구도에 의한 요인으로 대부분의 개발자들에게 외면을 당했다.

OLE 2.0스펙은 단순 복합 도큐먼트에 국한되지 않은 진정한 객체지향개념의 컴포넌트 즉 확장, 커스터마이징, 개량이 가능한 객체를 기반으로 한 완전한 서비스 구조를 가지고 있었고, 이서비스 구조의 하부는 컴포넌트 객체 모델(COM)이 든든하게 받치고 있다. OLE2.0의 구조를 통해구현이 가능한 서비스는 COM, 클립보드(Clipboard), 끌어놓기(Drag and drop), 포함(Embedding), 제자리 활성화(In-place activation), 연결(Linking), 모니터(Monikers), OLE 자동화(OLE Automation), OLE 컨트롤(OLE Controls), OLE 도큐먼트(OLE documents), 구조화 저장소(Structured storage), 단일 데이터 전송(Uniform data transfer; UDT) 등이 있다.

OLE에는 더 이상 버전번호를 붙이지 않는다. OLE는 확장이 가능한 구조이기 때문에 기본 제반 구조를 변경하지 않고도 개선과 확장이 가능하다.

OLE의 목적은 프로그램을 제작하는데 기존에 제작한 프로그램을 이용하여 프로그램 제작 시간을 단

축하고 경제성을 높이자는 것이다. 즉, 여러 어플리케이션에서 생성된 항목이나 “객체”들을 포함하는 문서를 만들고 편집할 수 있는 메커니즘이다. 복합 문서(Compound Document)라 불리는 OLE문서는 여러 종류의 데이터 컴포넌트들을 매끄럽게 통합해 준다. 사운드 클립, 스프레드 시트, 비트맵은 OLE문서를 찾을 수 있는 컴포넌트의 전형적인 예이다. 어플리케이션에서 OLE를 지원한다는 말은 다른 어플리케이션을 별도로 실행 시키지 않고도 OLE로 다른 구성요소를 불러올 수 있음을 의미한다. 즉 OLE는 엑셀 스프레드시트내의 값을 변경하여 워드문서내에 포함된 차트에 첨가하는 등의 작업을 가능하게 한 것이다. OLE는 데이터의 공유 뿐 아니라 자동화에도 기여하였다. OLE 자동화라는 기능은 하나의 어플리케이션이 다른 어플리케이션과 교류하고 이를 제어하는 과정을 가능하게 하였으며, OLE를 통해 제어되는 프로그램은 어플리케이션의 컴포넌트로 간주되었다. 이러한 기능은 여러 가지 특별한 어플리케이션의 개발을 통해 현실화되었다. 예를 들어 마이크로소프트의 워드의 매트릭스 기능을 통해 엑셀이 특정한 계산을 수행하는 등의 작업이 가능해 졌다. 따라서 OLE를 사용한 어플리케이션을 만들때 프로그램을 모듈별로 작성할 수 있으며 컴포넌트 소프트웨어 개발이 가능하다. 또한 플러그-인 개발방식이 가능한 장점이 있다.

2.2 CORBA(자바기반) & ActiveX비교

CORBA (Common Object Request Broker Architecture)는 네트워크에서 분산 프로그램 객체를 생성, 배포, 관리하기 위한 구조와 규격이며, 네트워크 상의 서로 다른 장소에 있고 여러 벤더들에 의해 개발된 프로그램들이 “인터페이스 브로커”를 통해 통신하도록 해준다. CORBA의 핵심개념은 ORB(Object Request Broker)이다. 이기종 컴퓨터들의 클라이언트와 서버 네트워크에 대한 ORB 지원이란, 클라이언트 프로그램이 분산 네트워크에서 서버가 어디 있는지, 또는 서버 프로그램이 어떤 인터페이스를 가질지 인식하지 않고서도, 서버 프로그램이나 객체로부터 서비스를 요구할 수 있다는 것을 의미한다. ORB들간의 요구와 응답을 성립시키기 위해 프로그램들은 GIOP(General Inter-ORB Protocol)를, 인터넷에서는 IIOP(Internet Inter-ORB Protocol)를 사용한다. IIOP는 GIOP의 요구와 응답을 각 컴퓨터의 인터넷 TCP(Transmission Control Protocol) 계층으로 대응시킨다.^[1]

CORBA의 경쟁대상은 독자적인 분산 객체 구조를 가진 마이크로소프트의 DCOM(Distributed Component Object Model)이다. 그러나 CORBA와 마이크로소프트는 게이트웨이를 통해 COM 클라이

엔트 객체가 CORBA 서버와 통신할 수 있도록 하는데 합의하였다. 마이크로소프트사가 책정한 객체 간 통신 규약. COM과 같은 종류의 객체 간 통신 규약에는 객체 관리 그룹(OMG)이 책정한 공동 객체 요구 매개자 구조(CORBA)가 있지만 COM에는 CORBA와 같은 객체 계층의 개념은 없다. 이 때문에 COM으로 객체의 재이용이라 말하는 객체 지향의 이점은 누릴 수 없다. ActiveX 기술은 마이크로소프트의 기본 객체기술인 COM을 기반으로 만들어진 프레임워크라고 할 수 있다. COM기술이 눈에 보이지 않는 아키텍처적인 성격이 강하다면 ActiveX기술은 COM기술을 비주요한 환경으로 구체화시킨 것이라고 말할 수 있다. CORBA는 모든 CORBA 서비스가 구현된 것은 아니라는 점, 사용자의 IDL, Language Mapping(IDL to C++/JAVA) 등 새로운 기술 습득이 필요하다는 단점을 가지고 있다.

2.3 마스킹(Masking) 기법

마스킹(Masking) 기법은 영상의 한 부분을 차단함으로써 스크린의 중첩비를 일시적으로 변형시키는 기법입니다. 예를 들어 오브젝트를 각각 하나씩 담고 있는 두 개의 레이어가 있다고 가정 했을때, 레이어의 특성상 위에 있는 레이어의 오브젝트와 아래에 있는 레이어의 오브젝트가 겹쳐지면 위에 있는 오브젝트가 아래 오브젝트를 가리게 됩니다. 이처럼 두 레이어의 오브젝트가 겹쳐 있는 상태에서 위쪽 레이어에 마스크를 설정하면 위에 있는 레이어의 오브젝트 영역만큼만 아래쪽 오브젝트가 투과되어 보이게 되는데, 이것이 마스킹 기법의 기본 원리입니다.

이 마스킹(Masking) 기법으로 스킨 처리를 할 경우 마스킹 이미지 작업이 필수적이다. 또한 애니메이션이 불가능하며, 초당24[fps] 즉, 24장의 마스킹 비트맵이 필요하게 된다. AlphaBlending은 불가능하며 일반적인 틀이 없는 실정이다.

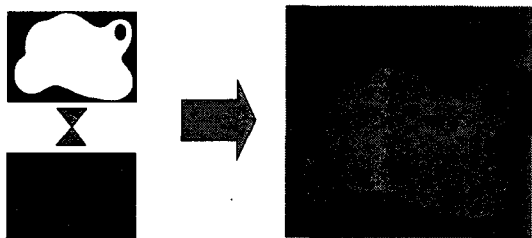


그림 1 알파블렌딩 프로세스
Fig. 1 Alpha Blending Process

2.4 알파블렌딩(Alpha Blending) 기법

알파블렌딩이란 화면에 찍을 픽셀과 이전에 찍혀

있던 픽셀을 서로 섞어 투명한 효과를 내는 것인데 단어의 뜻으로 알수 있듯이 알파값 가지고 섞는다는 의미입니다.

여기서 알파값이란 투명한 정도를 의미하며 예를 들어 도우넛 모양의 그래픽 파일을 프리젠티에 삽입하는 경우 해당 파일이 알파채널을 가지고 있다면 도우넛의 가운데 빈 공간이 프리젠티에서 설정한 배경화면으로 나오게 된다.

보통 알파값을 정수로 쓸경우 0~255사이의 값을 쓰는데 값이 클수록 해당 픽셀의 비율이 높아지고 낮을수록 비율이 낮아진다.

알파 블렌딩은 투명한 물체를 표현하기 위한 기법으로 블렌딩 순서에 또는 목적에 따라 다양한 블렌딩 연산이 존재하지만 일반적으로 다음과 같은 연산을 통해 수행된다.^[2]

$$CO = CS \cdot \alpha + (1 - \alpha) \cdot CD$$

(CO: 결과값, CS: 블렌딩 하려는 투명한 물체의 색깔값, CD: 투명한 물체가 덮어 씌울 이미 그려진 물체의 색깔 값, α : 블렌딩 하려는 물체의 투명도)

그림 1 에서 알 수 있듯이 물체가 렌더링 된 순서에 따라서 화면에 나타나는 영상이 달라지기 때문에 알파 블렌딩 유닛에 들어올 처리될 물체들의 순서를 실제 블렌딩 될 순서대로 유지하는 것은 매우 중요하다.

알파 블렌딩의 순서는 물체가 먼저 보이는 쪽부터 처리되거나, 가장 뒤쪽으로부터 처리되는 등의 두가지 경우가 있다. 물론 물체들이 블렌딩 유닛으로 들어오는 순서가 블렌딩 될 순서를 항상 유지하고 있다면 가장 이상적인 경우가 된다. 하지만 일반적인 경우 연관성이 없는 물체들이 무작위로 늘어서 있게 되므로 렌더링 할 물체들의 정보를 미리 트리구조로 만들어 놓고 정렬을 통해 앞에서 뒤, 또는 뒤에서 앞으로 정렬 작업을 통해 블렌딩 순서를 유지해 주어야 한다.

. 0x00RRGGBB ~ 0xFFRRGGBB (0x00=완전투명, 0x7F=투명도 50%, 0xFF=불투명)의 범위안에서 색상이 지원이 된다. 알파블렌딩 역시 일반적인 틀이 없는 실정이다. PSD, PNG, TIFF, SWF등의 형태로 콘텐츠를 publishing할 수 있다.

3. Windowless ActiveX Container

Windows2000이상에서는 윈도우를 투명처리 할 수 있는 UpdateLayeredWindow라는 함수가 지원된다. 하지만 CPU리소스를 많이 점유하므로, 재갱신되는 윈도우 영역을 최소화하여 성능을 향상시켜야 하

고, Window 9x, NT에서는 이러한 기능의 함수가 지원되지 않으므로 알파블렌딩 DIB포맷의 출력된 내용으로 영역값을 추출하여 영역처리를 수행한다.

또한 성능향상이 필요한 투명처리부는 인라인 어셈블리(inline assembler)로 코딩하여 성능을 향상시켰다.

3.1 구현설명

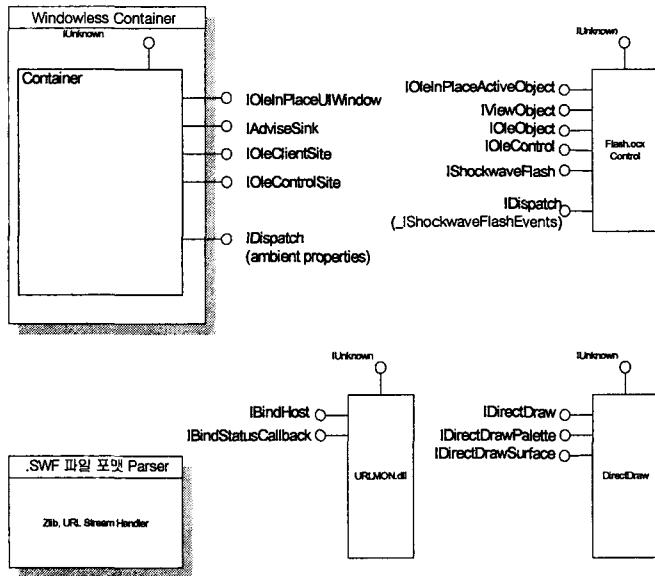


그림 2 클래스 구성
Fig. 2 Class Structure

그림 2의 windowless ActiveX Container를 구동하는 순서도는 다음과 같다.

- Container를 초기화한다.
- Control을 Windowless모드로 Activation한다.
- 무비클립이 Binding되면 포맷 분석기를 통해서 Document Size와 flash버전 등을 얻는다.(추후에 초기 윈도우 화면 크기를 퍼블리싱 된 DocumentSize로 화면에 출력한다)
- 무비클립이 Play되면 매 Frame마다 Container에 Event를 발생시킨다.
- 컨테이너는 투명처리작업을 수행하고 Draw를 통해서 화면에 출력한다. Windows 9x인 경우 윈도우를 알파블렌딩할 수 있는 함수 지원이 안되므로 영역처리를 수행하고, Windows 2000 이상에서 투명 윈도우를 출력한다. 9x, 2000 계열 모두 화면출력 시 Update된 영역만을 처리하여 성능을 향상시켰다.
- 무비클립에서 지정한 fs-command Event를 받으면 해당 작업을 수행한다.

3.2 특징

그림 3은 Windowless ActiveX Container의 개념도를 나타낸 것이다. 그리고 표1은 Windowless ActiveX Container의 Spec을 나타낸 것이다.

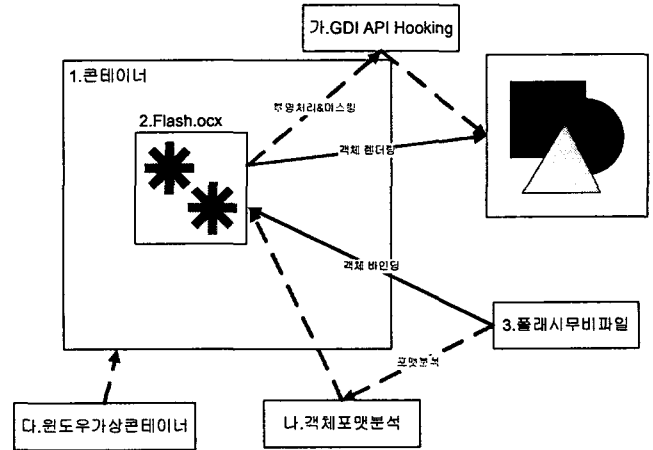


그림 3 개념도
Fig. 3 General idea picture

표 1 Construction Spec.
Table 1 Construction Spec.

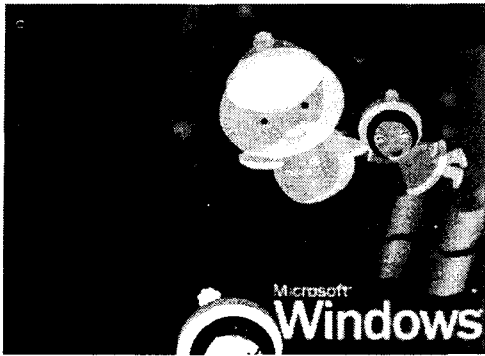
O/S	Windows 9x, NT 4.0 서비스팩 3이상, Windows 2000, Windows XP
플래시 지원	flash.ocx(4.0이상 지원) 별도의 컨테이너(Web Browser) 없이 투명한 윈도우 구현
사이즈	100kb 이하 경쟁제품과 비교시 성능 우수
Sprite	Sprite기능 지원(CPU 리소스 최소화)

Window9x 계열에서의 Window 영역처리를 지원하고, 경쟁제품 (IceProjector, IcePlayer)과 호환성 확보하였다. Customize Context Menu, Player Window Handling, Desktop Window Coordinate, DisplayChange/Quit Event Handling 등 디자이너를 위한 다양한 FS-COMMAND지원한다. Media player/web-browser control을 플래시 안의 윈도우에 내장가능하고, 플래시와 연동하여 다양한 효과 구현할 수 있다. 동영상은 별도의 스트림 서버에서 전송 받음으로 SWF 파일 사이즈 최소화하며, Animation-gif지원한다.

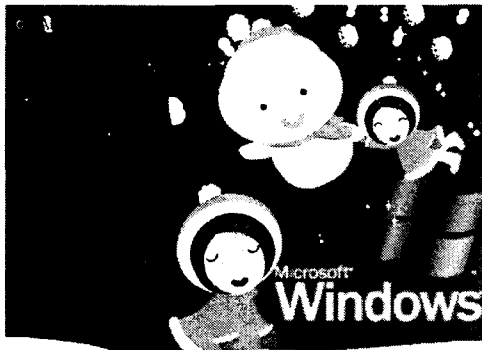
3.3 적용

그림 4에서 보듯이 windows 9x인 그림 (b)의 경우 updatelayeredwindow와 같은 기능의 함수가 존재하지 않으므로 alpha값이 50% 이하인 경우에는

투명으로 alpha값이 50% 이상은 불투명으로 인식하고 영역처리를 수행한다.



(a) 2000&xp



(b) windows 9x

그림 4 각 운영체제에서의 비교
Fig. 4 Class Structure

그러나 실제로는 alpha값이 50% 이하인 경우에는 투명으로, 50% 이상은 불투명으로 인식하고 영역처리를 수행한다. 하지만 realtime으로 cpu점유를 최소화하면서 영역처리를 매프레임마다 자동으로 수행하는 것이 의미가 있다.

5. 결론 및 향후 과제

본 논문에서는 플래시 무비를 OLE와 ActiveX로 일정한 틀이 없는 윈도우상에서 구현하였다. 상용화된 제품인 Projector의 경우 Publishing 할 때 .exe 형태로 만들어지고 Windowless가 지원되지 않았다. 메신저나 광고용으로 제작된 플래시 무비 구현 시 기존의 전용 플레이어를 사용 할 경우 별도의 소스 라이선스가 필요하며 사이즈가 커지고 호환성이 좋지 않는 문제가 있었다. 그러나 별도의 어플리케이션을 ActiveX로 구현 시 소스 라이선스를 필요로 하지 않으며 별도의 컨테이너 없이 투명한 윈도우를 구현할 수 있었다. 또한 다양한 O/S를 위해 별도의

컨테이너 구조를 통해 프로그램의 우수성을 확인하였다.

참 고 문 헌

- [1] J. Siegel, "CORBA Fundamentals and Programming", John Wiley & Sons Inc., pp. 196-204, 1996.
- [2] Tomas Akenine-Moller, Eric Haines, "Real-Time Rendering," Second Edition, AK Peters, p.102, 2002 .
- [3] 유정근, "웹 환경에서 ActiveX를 이용한 DB 어플리케이션의 설계 및 구현", 광운대학교 전산대학원, 석사학위논문, 1997.
- [4] 장진곤, "Web상에서 ActiveX 기법을 이용한 CAI", 계명대학교 교육대학원, 석사 학위논문, 1998.
- [5] 김명미, "플래시애니메이션을 이용한 동영상광고에 대한 연구 : 포털사이트를 중심으로", 중앙대학교 대학원, 석사학위논문, 2002.