

이동통신에서의 SWF Player 설계 및 구현

서 정현*, 김 대규*, 강 영만**, 장 문석*

*순천대학교, **여수대학교

Design and Implementation of SWF Player on Mobile Communication

Joenghyun Seo*, Daekyu Kim*, Youngman Kang**, Moonsuk Jang*

*Sunchon National University, **Yosu National University

E-mail: jang@sunchon.ac.kr

요 약

무선이동통신의 발전과 더불어 다양한 형태의 무선 단말기에서 구동하기 위한 많은 어플리케이션들이 개발되었다. 본 논문은 이러한 환경에서 좀더 발전된 멀티미디어 환경을 제공하고자 시스템 의존도를 최소화한 미려한 Vector 방식 동영상 출력 모듈과 그래픽 알고리즘을 내장한 SWF 포맷을 지원하는 SWF Player를 설계하였다. 또한 다중 플랫폼 기반의 추상커널 계층을 이용하여 기기 종속적인 부분을 최소화하고, 일관적인 인터페이스를 지원할 수 있는 연구를 토대로 Mobile 환경에 적합한 SWF Player를 구현한다.

ABSTRACT

Many applications are developed on variety type of mobile terminal with mobile tele-communication improvement. We suggest that we offer more improve multimedia environment on that development. So we design SWF player which supports Macromedia SWF file format. It is application that animation and graphic module are independent of system. In addition to, we implement SWF player on mobile communication environment. For the purpose of it, we adapt AKL concept, which is designed for the consistent interface on heterogeneous operating system environment.

I. 서 론

현재 정보통신의 발전은 인터넷 이용자의 증가에 따라 급속히 발달하고 있다. 이런 배경으로 무선이동통신 또한 발전하여 다양한 형태의 무선 단말기에서 구동하기 위한 많은 어플리케이션들이 개발되었다. 고성능 이동정보통신기기(휴대전화, PDA, 스마트 폰, 노트북 PC)의 대중화와 무선 통신 기술의 발달로 인해, 사용자의 이동 통신에 대한 관심과 멀티미디어 서비스 요구가 날로 높아지고 있다.[1] 이러한 환경에서 좀더 발전된 멀티미디어 환경을 제공하고자 시스템 의존도를 최소화한 미려한 벡터 방식 동영상 출력 모듈과 그래픽 알고리즘을 내장한 Macromedia사의 SWF 포맷을 이용한 SWF 플레이어를 설계하였다.

본 논문에서는 2.5세대 이동 단말기에 고급화된 멀티미디어 서비스를 제공하기 위하여 SWF 파일 구조를 분석하고 이를 이동 단말에서 구동하기 위한 플레이어를 설계하기 위해 기존 SWF 플레이어 운영 방식을 연구하였다. 먼저 본문에서 SWF 파일 구조 및 특성을 설명하고 이를 각기 다른 하드웨어를 가진 이동단말기에 인터페이스를 지원하기 위한 해결방안으로써 다중 플랫폼

기반의 추상 커널 계층(AKL : Abstract Kernel Layer)을 이용하여 각종 운영체제를 지원하고, 기기 종속적인 부분을 최소화하여 일관적인 인터페이스를 지원할 수 있는 연구를 토대로 이동통신 환경에 적합한 SWF 플레이어를 구현하고자 한다.

II. SWF

1. SWF의 장점

현재 인터넷상에서 존재하는 웹사이트의 95% 가량이 Macromedia SWF 플러그 인을 탑재하여 사용자들에게 제공하고 있고, SWF 플레이어는 3억 이상의 사용자들이 그에 대한 연구 및 개발에 참여하고 있다. 이에 Macromedia사는 1998년 4월 이후부터 SWF 플레이어를 상업화 공개 배포하고 있다. 또한 SWF 파일 구조의 특성상 멀티미디어 부분에서 빠른 속도로 사용자들이 증가하고 있다.

일반적으로 SWF파일 형식은 인터넷상에서 그래픽과 애니메이션을 제작하는데 매우 효과적으로 설계되었다. 장점으로는 컬러 형식의 비트맵, 애니메이션, 대화 버튼 등을 스크린에 빠르

게 출력한다. 또한 SWF 파일 형식은 태그를 사용함으로써 이전 버전의 플레이어를 호환하기 쉽고 새롭게 발전시키기에 용이한 형식이다. SWF 파일은 제한된 네트워크 예측할 수 없는 대역폭에서 파일을 압축하여 스트리밍을 빠르게 전송할 수 있다. SWF 파일은 그 형식이 단순하여 작은 크기로 플레이어를 구현할 수 있고, 운영체제 합수 집합에 아주 제한된 부분으로만 의존하므로 포팅이 용이하다. SWF 파일은 폰트와 같은 외부 자원에 의존 없이 디스플레이하며, O/S 합수 집합에 매우 제한적으로 의존하기 때문에 각기 다른 환경을 가지는 디스플레이 장치와 비트 depth를 가진 하드웨어에서도 잘 작동한다. SWF 파일은 빠른 속도와 높은 품질로 수행하도록 설계되었다.[2]

2. SWF의 구조

SWF 파일은 확장자 .swf로 저장된다. SWF 파일은 바이트 열로 저장되며 [그림 1]과 같은 헤더 블록과 각 기능별 태그로 구성되어 있다. 헤더 블록에는 'FWS'의 고정된 문자와 버전, 파일 전체의 길이(바이트 수), 프레임 크기, 1초당 프레임 처리율 그리고 프레임의 개수 정보가 담겨져 있다. 각 기능별 태그는 태그헤더와 내용으로 나누어져 있다.

구성	형태	내용
헤더 블록	Sign	*UI[8] 'F'
		UI[8] 'W'
		UI[8] 'S'
	UI[8]	버전
	UI[32]	파일 길이
	RECT	프레임 크기
	UI[16]	프레임 주사율
UI[16]	프레임 개수	
SWF 몸체	태그	[그림 2]참조 태그 번호, 태그 길이
	태그	지정된 형태 태그가 지시하는 내용

	태그	종료태그

[그림 1] SWF 컨테이너의 구조

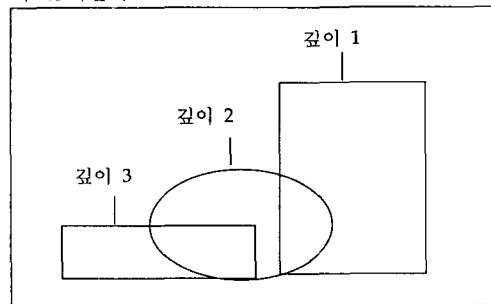
[그림 1]에서 *UI는 부호없는 정수를 뜻하며 대괄호 안의 숫자는 비트 수를 나타낸다. RECT는 프레임의 x, y축의 시작점과 끝점의 정보를 가진다. 태그는 Short 태그와 Long 태그가 있으며, Short 태그헤더는 태그의 길이가 62바이트 이하일 때 사용되고, Long 태그헤더는 그 길이가 63바이트 이상일 때 쓰인다. SWF 스트림은 하나의 헤더 블록, 태그의 반복 그리고 마지막으로 종료 태그가 오는 순서로 구성되어 있다.

종류	형태	Field
short tag	*UB[10]	태그 번호
	UB[6]	태그 길이
long tag	UB[10]	태그 번호
	UB[6]	'0x3F'
	UI[32]	태그 길이

[그림 2] 태그 헤더의 구조

[그림 2]에서 *UB는 부호없는 비트를 뜻하며, UI는 부호없는 정수로 대괄호 안의 숫자는 비트 수를 나타낸다.

SWF 파일의 디스플레이구조는 각각의 프레임 마다의 데이터로 구성되어 있으며 사운드, 이미지 등의 데이터가 프레임마다 분리되어 저장되어 있거나 데이터의 저장방식에 따라 해당 개체가 나타나기 시작하는 프레임에만 기록되기도 한다. 화면 좌표계는 트립(TWIP) 좌표계를 사용하고 있고, 1 트립은 1/20 픽셀로 표현된다. 때문에 미려한 그래픽 출력이 가능하다. 트립 좌표계는 논리적인 좌표계로 출력될 때는 화면의 픽셀 좌표계로 맵핑된다. 개념적으로, SWF 파일 형식은 정의와 제어 두 가지의 데이터 스트림으로 구성되어 있다. 정의 스트림은 사전에 캐릭터를 추가하고 제어 스트림은 사전에 정의되어진 캐릭터를 사용한 디스플레이 목록을 실행시킨다. 디스플레이 목록은 [그림 3]과 같이 특정 시간 시점의 화면 위의 오브젝트의 평면 목록이다. 또한 다른 구조에서는 자식 요소의 목록을 가지는 화면 위의 요소가 계층적인 목록으로 확장된다. 디스플레이 목록의 오브젝트는 깊이(Depth)에 의해 알 수 있는데, 깊이가 0일 경우 오브젝트는 스택의 맨 밑이다. 하나의 깊이가 주어질 때 오직 하나의 오브젝트가 존재한다.



[그림 3] 디스플레이 목록 구조

SWF의 이미지는 크게 Shape, Bitmap, Button, Text/Font로 구성되어 있으며 Shape는 일반적으로 PC 벡터 그래픽 형식으로 Fill과 Line Style 그리고 라인곡선의 연속을 가진 데이터 구조로 정의된다. Bitmap은 JPEG와 Lossless 압축을 사용하는 GIF 및 PNG 형식을 지원한다. Buttons는 SWF 파일에서 상호 작용을 위해 제공되는데, 특정상태 변화에 따른 행동을 지시한다. Text/Font는 내부적으로 True Type 폰트 출력 방식을 처리하는 것과 마찬가지로 하나의 그래픽 개체이다. 각 글자는 벡터로 표현되며 한 글자 당 무수히 많은 직선, 곡선으로 이루어진다. 사운드 형식은 WAV 형식의 일종인 ADPCM과 RAW, MP3의 세 가지 형식을 지원한다.[3] 마지막으로 SWF 파일은 간단한 Action model을 포함하고 있는데, 일반적으로 Button의 상태변화를 통해 해

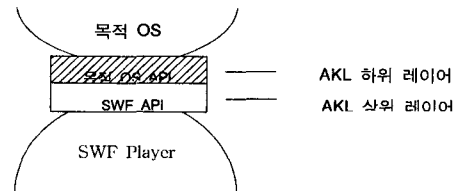
당 목록으로의 분기를 통해 관련 Action의 처리를 담당한다.

III. 추상 커널 계층

II장에서는 SWF 파일 구조 및 특성을 설명하였다. 이를 각기 다른 특성을 가지는 다양한 이동 단말에 구현하기 위해서는 기기 특성으로부터 영향을 최소화하는 방향으로 시스템이 설계되어야 한다. 이를 위한 방안으로써 본 논문에서는 다중 플랫폼 기반의 추상커널 계층 구조를 이용하여 각종 운영체제를 지원하고, 기기 종속적인 부분을 최소화하여 일관적인 인터페이스를 지원하도록 설계하였다. 본 추상커널 계층 구조는 이동통신 환경에 적합한 SWF 플레이어 구현하고자 무선 단말에서의 WAP 브라우저 지원을 위한 AKL 설계와 구현에 관한 연구에 기초한 것이다. 본 연구에서 이용하는 추상커널 계층 구조는 마이크로 브라우저의 개발에 적용하여 상용화된 기술이다.

1. 개념 모델

추상커널 계층은 다중 플랫폼 지원을 위한 구조로써 [그림 4]와 같이 목적 운영체제 커널(Target OS kernel)에서 동작되는 별도의 커널을 두는 개념과 같다.[4] 따라서, 추상커널 계층 구조에서 구현된 응용프로그램은 어느 플랫폼에 이식하든지 동일한 기능을 발휘할 수 있다[5].



[그림 4] AKL 개념 모델

2 추상커널 계층 구성 요소

일반적으로 커널에서 지원하고 있는 태스크는 독립된 코드, 힙, 그리고 스택 영역으로 구성되어 있다. 본 연구에서는 추상커널 계층을 설계하여 이를 지원하는 태스크를 가상 태스크(Virtual task)로 명명한다. 이를 기반으로 구성된 추상커널 계층 구성 요소는 6가지로 나누어진다.

1) Dispatcher

메시지 큐로부터 메시지를 추상커널 계층에서 정의한 API 중 하나인 **receive**로 수신하여 대상 VTask를 결정하고 정의된 인자로 호출한다.

2) 메시지 큐

FIFO 큐로서 **send**와 **receive**를 이용하여 접근할 수 있다. 모든 VTask는 **send**를 이용하며 Dispatcher만이 **receive**를 이용할 수 있다.

3) 타이머 태스크

유닉스의 델타-큐 방식을 도입하였다. 각

VTask에서 타이머가 필요할 때, **GetTime** API를 이용하고, 취소는 **DeleteTime** API를 이용한다.

4) 가상 태스크 제어 테이블

각 VTask의 현재 진입점을 갖고 있는 목록이다. 전이는 **setStateTransition**을 이용한다.

5) VTask

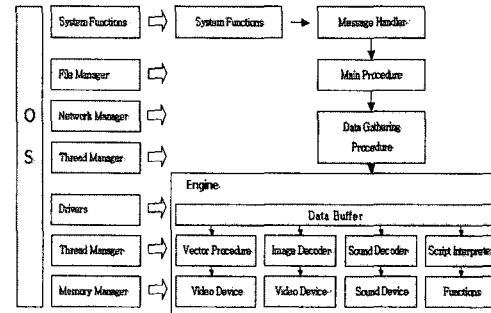
VTask는 제어권에 관해 수동적이라는 점에서 일반 서브루틴의 개념과 동일하나, 상태전이를 기반으로 구현해야 한다는 점이 다르다.

6) 기타 API

- 상호배제에 해당하는 기능
- 소켓(UDP)에 해당하는 기능
- Put 픽셀에 해당하는 기능

IV. SWF Player의 설계와 구현

1. SWF 플레이어의 설계



[그림 5] SWF 플레이어 구조

SWF player는 [그림 5]와 같이 7가지의 시스템 구조를 가지고 있으며 이는 운영체제의 특성에 따라 이식성을 고려한 구조로 설계되어 있다. 최초로 읽어 들인 데이터에는 SWF 파일의 전반적인 정보가 있는 헤더가 포함되어 있으며 이 헤더에서 파일의 전체 크기를 얻을 수 있다. 여기에서 얻어지는 파일 크기정보를 바탕으로 메모리 할당을 시도하고 할당된 메모리에 파일의 나머지 부분인 실제 데이터를 일종의 큐 형태로 저장한다. 데이터는 파일 시스템으로부터 읽어 들여지고 파일 끝에 도달할 때까지 반복된다. SWF 플레이어는 데이터를 모두 읽어 들인 후 디코딩을 시작하고 네트워크를 통한 데이터 수신은 로컬 파일 시스템과 달리 새로이 생성된 쓰레드에서 수신을 담당하고 데이터가 수신될 때마다 수신된 데이터의 크기만큼 힙에서 메모리 할당을 시도하며, 할당된 데이터 포인터와 함께 주 쓰레드로 쓰기 메시지를 전송한다. 메시지 핸들러는 시스템에서 발생하는 외부 환경의 변화 및 쓰레드간의 통신에서 통지를 받기 위한 구조이다.

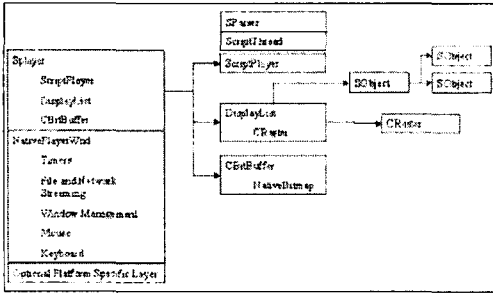
2. SWF 플레이어의 구현

본 연구에서는 SWF 플레이어의 구현을 위해

ANSI C를 이용하는데, 현재 이동 단말의 특성 상 객체지향 개념을 지원하는 C++이나 Java의 전체 기능을 수용하는 것은 메모리나 CPU 등의 자원의 제약으로 어렵기 때문이다.

그러나 SWF Player 구현시 객체지향 개념이 도입되면 특성을 자연스럽게 표현할 수 있으며, 상속관계 등을 통하여 다양한 기능을 구현할 수 있다. [그림 5]에서는 구현된 모듈의 클래스 단위의 상속관계를 표현한다. 본 연구에서는 이러한 객체 지향적 특성을 ANSI C로 표현할 수 있도록 객체 지향적 API를 설계하여 구현하고 이를 이용하여 SWF 플레이어를 구현하였다. 본 연구의 객체 지향적 API의 모든 클래스는 오브젝트라는 최상위 클래스로부터 기본적인 속성(멤버변수, 멤버함수)을 상속받는다. 따라서 모든 클래스는 이 오브젝트라는 클래스로부터 파생된 클래스들이다. 상속은 단일 상속만을 지원하도록 구현하였으며, 다중상속의 개념이 필요한 경우에는 Java의 경우와 같이 인터페이스를 통하여 구현하였다.

또한 SWF 플레이어는 현존하는 그래픽 알고리즘들을 상당량 내장하였고 순수하게 장치 독립적인 코드만을 사용하여 작성되어 있다. 메모리로 할당받은 버퍼에 내장된 그래픽 알고리즘으로 도형을 그려나가며 한 프레임이 완성될 때마다 화면 장치로 버퍼를 전송한다.



[그림 6] SWF 플레이어 구성 모듈

버퍼의 생성 및 출력은 장치 비 의존적으로 설계되었고, 다만 화면장치로의 전송을 위해서는 장치 의존 코드 작성을 필요로 한다. 내장된 스크립트를 통해 외부와 인터페이스가 가능하며, 내장된 스크립트 인터프리터를 통해 분석되고 수행되어진다. 스크립트로 내장된 네트워크 모듈을 통해 데이터의 전송과 수신이 가능하다. 상호간에 전달되는 데이터는 아스키 기반의 데이터만을 허용한다. 엔진 부분은 하나의 모듈에 모두 포함되며 네트워크, 채널서비스, 그 밖의 추가 지원 기능들은 별도의 라이브러리로 작성하는 것을 원칙으로 하여 최소한의 자원 투자만으로 기능확장이 가능하도록 구현하였다.

SWF 플레이어는 Splayer, NativePlayerWnd, SParser, ScriptThread, DisplayList, CBitBuffer 등으로 구성되고 각각의 기능은 다음과 같다. Splayer : OS에 독립적인 부분으로 ScriptPlayer,

DisplayList 와 CBitBuffer를 제어한다.

NativePlayerWnd : OS 의존적인 부분이며 시간 제어, 파일과 네트워크 흐름 제어, 창 관리를 하며, 마우스, 키보드와의 인터페이스를 담당한다.

SParser : SWF File의 데이터를 분석한다.

ScriptThread : 각각의 타임라인의 폰트, 사운드와 오브젝트들을 제어한다.

ScriptPlayer : 데이터 스트림과 내부 라이브러리를 제어를 담당한다.

DisplayList : Object의 출력과 출력영역에 대한 관리 작업을 수행한다.

CBitBuffer : SParser에서 분석한 데이터를 저장하고, 그 데이터를 SPlayer에게 전달한다.

5. 결론

이동통신의 보급에 따라 무선 인터넷 인구가 날로 급증하고 있으며, 이에 따라 이동 단말기 기능 및 서비스도 급격하게 발전하고 있다. 본 연구는 이런 급변하는 이동통신 시장에서 좀더 발전된 멀티미디어 서비스를 제공하기 위하여 SWF 플레이어를 구현하기 위한 연구를 수행하였다. 인터넷상에서 그래픽과 애니메이션을 제작하는데 매우 효과적으로 설계되어진 SWF 파일을 이동단말에서 제공함으로써 이동통신상에서 빠르고 미려한 애니메이션, 그래픽과 사운드 등의 멀티미디어 서비스를 이용할 수 있도록 한다. 이를 통하여 이동단말 상에서 사용자 서비스의 질을 향상하고, 본 연구의 결과물인 SWF 플레이어는 무선 인터넷 전화기에 이식하는 작업을 진행하고 있다. 앞으로 무선전화기 외에 PDA 등에도 SWF 플레이어를 이식하고 운영하는 연구를 계속할 예정이며, 속도와 메모리 사용량 등에 관한 최적화 방안도 병행으로 연구해야 할 분야이다.

참고 문헌

- [1] WAP Forum, "Wireless Application Environment Specification," April, 1998.
- [2] <http://openswf.org>, "Introduction to SWF"
- [3] <http://openswf.org>, "SWF File Format Specification"
- [4] WAP Forum, "Wireless Application Protocol Architecture Specification," April, 1998.
- [5] 김대규 등, "WAP 브라우저 지원을 위한 AKL 설계와 구현," 해양정보통신학회 춘계학술대회 초록집 5. 18. 2001.
- [6] QUALCOMM, "Real-Time Executive System(REX)," QUALCOMM Inc., 1999.
- [7] 이석호, "C로 쓴 자료 구조론," 최홍당, 1997.
- [8] David Pogue, "Palm Pilot : The Ultimate Guide," 2nd Edition, O'Reilly, 1999.