

OpenVG 엔진 기반의 UI 위젯 컴포넌트 시스템 설계 및 개발

김재형, 최윤호, 김종진, 이정화, 장진근, 손진현
한양대학교 컴퓨터공학과

e-mail:jhkim@cse.hanyang.ac.kr

OpenVG Engine-Based UI Widget Component System Design and Development

Jae Hyung Kim, Yun Ho Choi, Jong Jin Kim, Jeong Wha Lee,
Jinkun Jang, Jin Hyun Son
Dept. of Computer Science & Engineering, Hanyang University

요 약

OpenVG는 낮은 수준의 하드웨어 가속 기능과 이를 바탕으로 하는 2D 벡터 그래픽 렌더링 API를 제공함으로써 모바일 환경에서 높은 화질의 그래픽을 구현할 수 있다. 이러한 장점에도 불구하고 OpenVG 엔진을 이용한 실제 사용자 인터페이스(User Interface)의 구현은 쉬운 일이 아니다. 그 이유는 OpenVG API에서는 기존의 개발 환경에서 제공하는 기본적인 위젯 컴포넌트(Widget Component)를 제공하지 않기 때문이다.

본 논문에서는 OpenVG를 용이하게 사용하여 효율적인 UI 구현을 위한 위젯 컴포넌트 시스템을 제시하고자 한다. 우선 OpenVG 엔진을 기반으로 위젯 컴포넌트와 그들 사이의 계층 구조와 상관 관계를 설계하고, 시스템이 구동되는 방식을 기술한다.

1. 서론

1)최신의 사용자 인터페이스는 동적이면서 화려함을 모토로 발달하고 있다. 이러한 기조가 모바일 환경에서도 예외가 아니다. 모바일 환경에서 그래픽 성능이 갈수록 중요시되면서, 주목 받고 있는 것이 벡터 그래픽 분야다. 벡터 그래픽을 이용한 사용자 인터페이스는 화면의 크기에 관계없이 일정한 해상도를 표현할 수 있으며, 동적인 변화에도 부드러운 처리를 지원한다.

이러한 벡터 그래픽 분야의 활성화와 맞물려 국제 소프트웨어 표준화 기구인 ‘크로노스 그룹

(Khronos Group)’은 벡터 그래픽 분야의 표준을 표방하는 OpenVG를 발표했다. 벡터 그래픽에 대해서 데스크탑 환경에서는 플래시(Flash)가 표준으로 쓰이고 있지만, 모바일 환경에서는 그렇지 못한 것이 현실이다. 모바일 환경의 그래픽 보드가 플래시를 위한 가속(Acceleration)을 제공하지 못하기 때문이다. 따라서 모바일 환경 등의 임베디드 시스템을 타겟으로하는 OpenVG는 업계의 표준이 될 것으로 확신하고 있다.

이처럼 벡터 그래픽이 사용될 곳은 무궁무진하지만 그 중에서도 가장 필수적으로 사용되는 곳이 사용자 인터페이스다. 그런데 OpenVG에서는 이를 위한 응용 프로그램 인터페이스를 제공하지 않고 있다. 본 논문에서는 OpenVG를 이용해 사용자 인터페이스를 개발하는 개발자의 작업능률을 향상시킬

1) 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원 사업(IITA-2006-C1090-0603-0031)의 연구결과로 수행되었음

수 있는 위젯 컴포넌트 시스템을 설계 및 제안하고자 한다.

2. 관련 연구

2장에서 살펴볼 내용은 벡터 그래픽의 특징과 임베디드 환경에서의 필요성을 기술하도록 한다. 이는 본 논문에서 설계한 위젯 컴포넌트 시스템의 구성을 이해하는데 배경지식으로 유용하게 쓰일 수 있다.

벡터그래픽은 주어진 2차원이나 3차원 공간에 선이나 형상을 배치하는데 있어 일련의 명령들이나 수학적 표현을 통해 디지털 이미지를 만든다. 벡터 그래픽 파일에는 선을 그리기 위해 각 비트들이 저장되어 있는 대신에, 연결될 일련의 점의 위치가 들어 있다. 그로 인해 파일 크기가 작아지는 결과를 가져온다. 벡터 그래픽을 화면에 표시하기 위해서는 임베디드 시스템의 특성상 제약이 발생할 수밖에 없다. 가장 문제되는 것은 저전력성이다. 기존의 그래픽 라이브러리에서는 벡터 그래픽을 표현하기 위해서는 소프트웨어 엔진을 통한 가속을 활용해 벡터 그래픽의 부드럽고 자연스러운 확장 축소작업을 처리해왔다.

3. OpenVG 위젯 컴포넌트 시스템 설계 및 개발

본 논문에서 제안하는 시스템에 대한 설계와 구체적인 개발 내용을 소개하도록 한다.

3.1 대상 환경

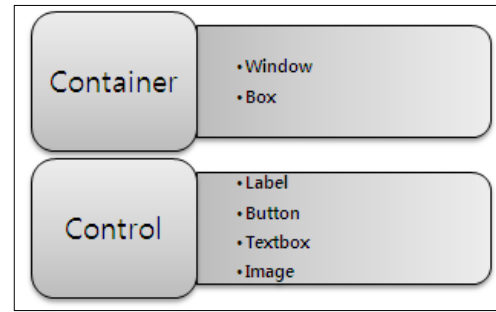
본 논문에서 제안하는 시스템을 설계하는 데 있어서 지정한 대상 환경은 임베디드 시스템 중에서도 최근 주목받고 있는 스마트폰이다. 스마트폰의 운영체제(Operating System)는 윈도우즈 CE가 사용된다. 또한 입력 장치로 키패드 입력만을 지원하고 있다. 따라서 본 논문에서 제안하는 시스템은 윈도우즈 CE의 환경을 어느 정도 고려하였고, OpenVG 응용 프로그램 인터페이스를 기반으로 설계되었음을 밝혀둔다. 또한 C 언어를 기반으로 설계 및 구현되었다.

3.2 OpenVG 위젯 컴포넌트 계층 설계

일반적인 데스크탑 환경에서 제공하는 위젯 컴포넌트들의 종류를 대폭 축소시켜 CPU나 메모리에 제약이 있는 임베디드 시스템 환경에 적합하고, 스마트폰 등의 모바일 환경의 작은 화면에 적합한 위젯들로만 구성을 이룰 수 있도록 디자인 되었다.

본 시스템에서 제공하는 위젯 컴포넌트의 종류는

(그림 1)과 같다. 기본적으로 다른 위젯들을 포함할 수 있는 위젯들을 컨테이너(Container)로 명명하고

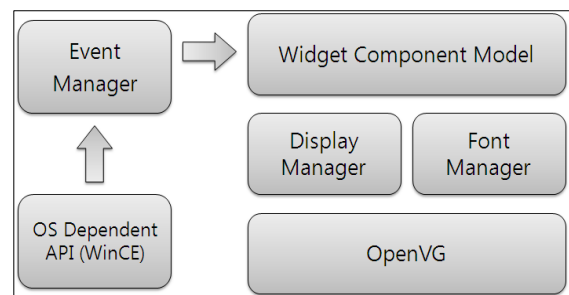


(그림 1) Widget Component Model

위와 같이 분류하였다. 레이블(Label)과 버튼(Button)처럼 화면 표현의 개념 상 컨테이너 내부에 포함되면서 화면에 표현되는 위젯들을 컨트롤(Control)로 명명하였다. 박스의 경우 또 다른 박스의 내부에 포함이 가능하지만 기본적으로 컨테이너에 속하므로 이에 따라 분류하였다.

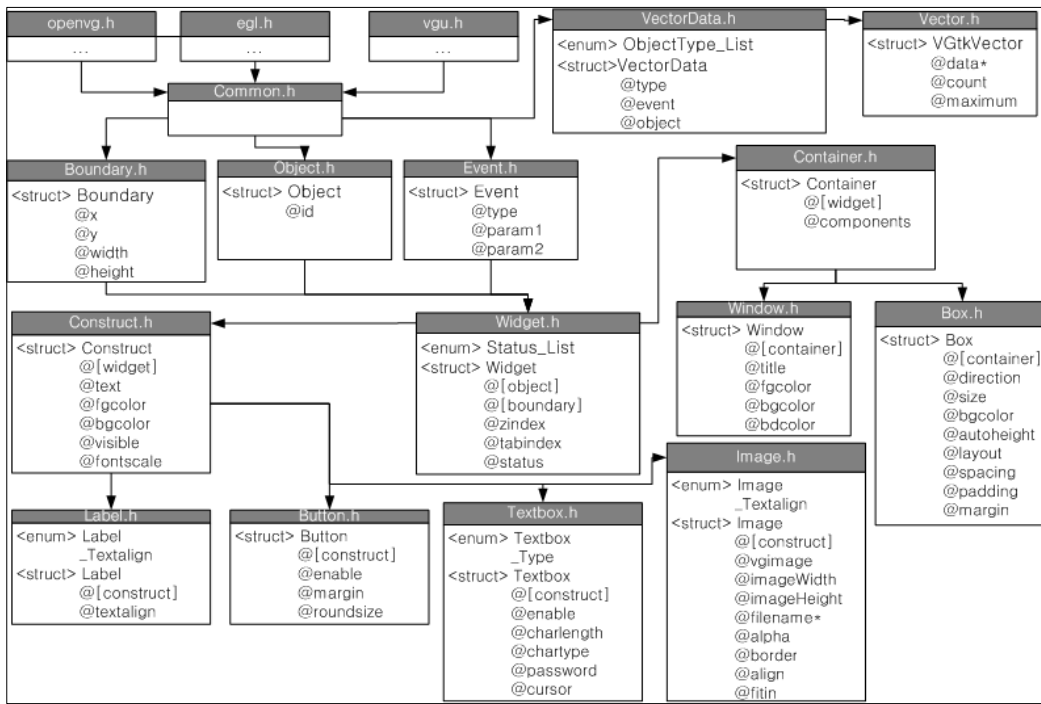
3.3 시스템 구성

OpenVG 기반의 위젯 컴포넌트를 사용하기 위해 제공되는 총체적인 시스템을 도식화하면 그림 2와 같다.



(그림 2) System Architecture

기본적으로 OpenVG 엔진을 기반으로 그 위에 디스플레이 매니저(DisplayManager)와 폰트 매니저(FontManager)가 상위 계층을 형성하고 있다. 디스플레이 매니저는 OpenVG API의 라이브러리를 바탕으로 실제로 표현될 화면을 구성하는 역할을 담당하고 있다. 독립적 구성을 통해 디스플레이 매니저의 교체를 통해 화면 구성에 변화를 주는 테마 기능을 구현할 수 있다. 폰트 매니저는 텍스트(Text) 정보를 얻어서 화면에 표현해주는 역할을 하는데, 디스플레이 매니저에서 텍스트 정보와 표현될 위치 정보를 인자로써 전달하며 호출되는 방식을 취한다.



(그림 3) Widget Component Model Structure

디스플레이 매니저를 통해 표현될 위젯 컴포넌트들의 속성 정보를 보유하고 있는 부분이 위젯 컴포넌트 모델(Widget Component Model) 부분이다. 각 위젯 컴포넌트에 대한 속성 정보를 객체의 형태로 유지하고 있으며 객체를 생성하는 것으로 화면에 출력하기 위한 기반 작업이 처리된다.

본 시스템은 3장 1절에서 언급한 바대로 환경에 독립적이지 않다. 따라서 어느 정도의 의존성을 가질 수 밖에 없는데 그 부분이 바로 이벤트 핸들러(EventHandler) 부분이다. 본 시스템이 대상으로 하는 환경은 윈도우즈 CE로서, 윈도우즈 CE에서 발생하는 이벤트를 캐치(Catch)하여 본 시스템 고유의 형태로 이벤트를 재가공한다. 이를 통해 적절한 이벤트를 각 위젯 컴포넌트에 전달할 수 있고, 이벤트 핸들링 모듈을 시스템 내부로 이양할 수 있다.

3.4 기능부 DLL 구성

본 시스템에서는 각 기능부를 DLL로 구성한다. DLL을 구성할 때, 객체 지향적인 디자인을 위해 각 기능부는 독립적으로 구성되었다. 간략히 설명하자면, 각 기능부 내부에서 사용될 함수를 또 다른 DLL에서 호출해서 사용하는 방식을 지양한다는 이야기이다. 본 논문에서 제시하는 시스템은 다음과 같은 DLL 형태의 기능부로 구성되어 있다. 각 기능부에 대한 상세 설명은 다음과 같다.

3.4.1 Model.dll

본 시스템에서 제공하는 위젯들의 기반에 해당하는 것으로 각각의 위젯들이 객체 지향적인 설계에

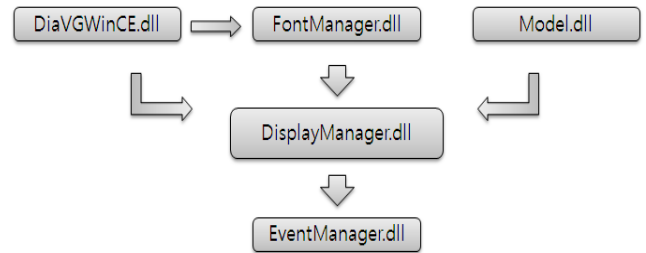


그림 4 DLL System Flow Diagram

따라 구조체로 구성되어 있는 것을 (그림 3)에서 확인할 수 있다. 여기에는 위젯을 사용하기 위한 메소드(Method)에서부터 위젯의 표현 양식 등을 결정짓는 속성들을 정의하고 있다. 또한 시스템의 기반을 이루는 자료구조에 대한 정의도 포함된다. 이 DLL에서 정의내린 데이터들을 기반으로 디스플레이 매니저가 화면에 위젯을 표현하게 된다. 각 구조체들은 객체 지향(Object-Oriented) 설계 원리에 따라 설계된 것으로 일련 직선으로 이루어진 관계들은 모두 부모-자식(Parent-Child) 관계로 파악할 수 있다. 이는 객체 지향의 상속(Inheritance) 관계를 나타내는 것으로 각각이 객체 지향의 캡슐화(Encapsulation)를 따르면서 어느 정도의 다형성(Polymorphism)을 구현하는 것이 가능하다. 즉, 추가적인 위젯 컴포넌트를 제작할 경우 상위 객체의 속성(attribute)만을 상속받게 되면 제작에 들어가는

시간이 상당히 줄어들 뿐만 아니라 중복적인 코드의 사용도 방지할 수 있다

3.4.2 DiaVGWinCE.dll

이 기능부에서는 OpenVG 엔진이 할당되게 된다. DiaVGWinCE.dll에 존재하는 함수들이 폰트 매니저와 디스플레이 매니저로부터 호출되는데, 이를 그림 4의 화살표를 통해서 확인할 수 있다.

3.4.4 DisplayManager.dll

화면에 각 위젯들을 출력시켜주는 역할을 하는 DLL에 해당한다. (그림 3)의 모델에서 정의된 위젯들의 구조체들을 기반으로 OpenVG 응용 프로그램 인터페이스를 이용한다. 이를 통해, 화면상에 각 위젯들을 실체화 시키는 것이 디스플레이 매니저의 목적이다. 또한 OpenVG 엔진을 사용하는 개발자의 입장에서 OpenVG 응용 프로그램 인터페이스만을 활용해서 화면 구성에 대한 변화를 쉽게 줄 수 있는 테마 기능을 제공한다.

3.4.3 FontManager.dll

디자인 부분에서의 폰트 매니저를 담당하는 것으로 텍스트를 표현하기 위한 라이브러리를 의미한다. 따라서 디스플레이 매니저를 통해서 각 위젯이 표현될 때 각 위젯 내부에 포함되는 텍스트를 그리기 위한 부분을 이 기능부에서 호출한다는 것을 그림 4의 화살표를 통해 알 수 있다.

3.4.5 EventManager.dll

위에서 언급한 기능부들이 기본적으로 그려지는 기반 시스템 호출 이벤트 부분에서부터 사용자의 입력에 따른 데이터들의 조작과 그 변화에 따른 결과물을 출력시키기까지 일련의 모든 과정들을 처리하는 부분에 해당한다. 또한 입력 이벤트에 대한 사용자가 직접 제작한 처리 과정을 적용하기 위한 사용자 정의 함수(User Definable Function)이 위에서 기술한 모델 부분에서 정의되어 있어 이에 대한 호출도 담당한다.

4. 시스템 구현

본 논문에서 제시한 설계를 바탕으로 구현한 시스템을 스마트폰에 탑재하였다. 본 논문에서 제시한 위젯 컴포넌트를 모두 사용하였고, 각 기능부의 동작 처리를 확인하기 위해 간단한 로그인 폼을 구성하였다.

기본적으로 시스템에서 제공하는 각 위젯 컴포넌

트 간의 이동을 위한 포커스(Focus) 처리에서부터 각 위젯 컴포넌트에 대해 사용자 정의 함수를 연결시킬 수 있도록 제작되었다. 이를 통해 사용자 인터페이스에서 통상적으로 사용되는 화면 전환에 대한 처리를 확인할 수 있다. 텍스트 박스에서 입력받은 텍스트를 표현하기 위한 처리를 확인할 수 있다.



(그림 5) 시스템 구동

또한 본 시스템의 가장 큰 특징인 테마 기능도 탑재된 시스템의 디스플레이 매니저를 교체함으로써 확인이 가능하다.

5. 결론

기존의 개발 환경에서 제공하는 그래픽 라이브러리에는 개발의 용이성을 위해서 기본적으로 제공되는 위젯 컴포넌트가 포함되어 있었다. 기본적인 위젯 컴포넌트를 이용해서 개발자는 직접 위젯 컴포넌트를 제작할 필요가 없이 좀 더 효율적으로 개발에 임할 수 있으며 그래픽 API의 활용성을 높일 수가 있다. 본 논문에서 제안하는 시스템을 통해 벡터 그래픽을 위한 OpenVG 엔진의 활용성 및 확장성을 극대화시킬 수 있다.

참고문헌

- [1] <http://www.khronos.org/>
- [2] 김지용, 광지영, 설동명, 안성호, 이은령, 이경희, 김두현 “임베디드 그래픽 사용자 인터페이스 기술 동향”
- [3] <http://terms.co.kr/>
- [4] 김재명, 박태준, 양만석, 권기규, 임동선 “The Current Status of S/W Platform for Advanced Embedded Systems”
- [5] Stéphane Chatty, Stéphane Sire, Jean-Luc Vinot, Patrick Lecoanet, Alexandre Lemort, Christophe Mertz “Revisiting visual interface programming: creating GUI tools for designers and programmers”