

XML을 이용한 DivxPlayer 사용자 인터페이스

기반 리모콘 컨트롤 설계 및 구현

송두재, 이상훈, 문승진
수원대학교 IT대학 컴퓨터학과
{snakehand, vkfka1, sjmoon}@suwon.ac.kr

A Design and Implementation of Remote Controller Using XML based on a DivX Player User Interface

Du-Jae Song, Sang-Hun Lee, Seung-jin Moon
Dept. of Computer Science, The University of Suwon

요 약

최근 VCR과 DVD 플레이어 이후에 많은 주목을 받고 있는 Divx Player는 수많은 종류의 상품들로 전성기를 맞이하고 있다. 이러한 Divx Player는 다양한 상품의 홍수 속에서 이미 충분한 하드웨어, 소프트웨어적 발전을 이루었다. 따라서 엇비슷한 성능의 상품들 속에서 소비자에게 돋보이기 위해서 차별화된 디자인과 편리한 UI(User Interface)가 필요하게 되었다. 이러한 UI 디자인을 구성하는 다양한 방법 중에 웹문서를 조작하듯이 쉽고 편리하게 UI를 수정할 수 있는 XML 파싱이 크게 각광받고 있다. 따라서 본 논문에서는 이러한 Divx Player의 UI를 구성하는 XML의 기능과 그에 관련한 리모콘 조작의 연관성과 그 동작에 관하여 설계 및 구현을 하였다.

1. 서 론

임베디드 기반의 Divx Player는 그 구성을 C/C++같은 프로그래밍 언어를 사용하여 작성되며 그것은 UI(User Interface) 또한 마찬가지이다. 여기서 문제는 Divx Player 개발과정에서 추가되는 기능에 따른 개선 및 수정 사항에 많은 어려움이 발생한다는 것이다.

이런 UI개발에 대한 문제를 해결하기 위한 대안으로 XML을 이용한 비교적 간단한 개선과 수정방법이 있는데, 본 논문에서는 이러한 XML을 이용한 UI 구성을 통해 Divx Player의 세부동작이 어떻게 이루어지는지 확인해 보고자 한다.

그 중에서 외부 입력 방식인 리모콘에 대하여 XML이 어떤 역할을 하는지, 어떤 동작이 이루어지는지에 대하여 알아보도록 하자.

본문의 2장에서는 XML을 이용하여 설계한 UI의 구성 및 장단점에 대하여 알아보고 3장에서는 이런 XML로 구성된 UI에 관련된 리모콘의 동작 및 구현에 대하여 설

명하고자 한다.

4장에서는 이에 대한 결론과 문제점을 서술하고자 한다.

2. XML을 이용한 UI의 구성 및 장단점.

먼저 리모콘을 이용한 외부입력방식에 대하여 알아보기 전에 그 기반인 임베디드 시스템과 XML의 상관관계에 대하여 정확히 알아둘 필요가 있다.

그렇다면 XML은 무엇인가, 간단하게 설명하자면 XML은 구조화된 문헌을 웹상에서 전송할 수 있도록 설계된 표준화된 텍스트 포맷이다. 그렇다면 기존의 웹문서인 HTML에 비하여 XML이 어떠한 장점을 지니고 있는지 임베디드 기반 하에서 어떠한 역할을 하는지 알아 보도록 하자.

먼저, XML의 장점은 장치와 시스템에 독립적이라는 것이다. 즉 다양한 소프트웨어와 하드웨어에서 이용할 수 있으며 어떠한 XML 소프트웨어에 의해서도 처리될

수 있도록 설계되었다. 또한, 특정 언어에 한정되어 있지 않다. XML은 ISO 10646 (Unicode)을 수용하고 있기 때문에 어떤 언어라도 마크업에 사용할 수 있다. 따라서 이는 XML이 언어나 국가에 대한 제한 없이 국제적으로 사용될 수 있게 하는 것이 특징이다. 이외에도 누구나 손쉽게 이해할 수 있는 간단한 문법이나, 확장성 등으로 사용자의 접근을 유도하는 것이다. 위의 예시와 같이 XML은 웹기반으로 널리 통용되고 있지만, 임베디드 시스템에도 많은 이점을 제공하고 있는데 대표적인 사례가 바로 손쉬운 설계와 포팅시간의 단축이다.

먼저, 임베디드 장비를 기반으로 둔, Divx Player UI 개발의 경우 C/C++에 대한 지식이 부족한 그래픽 디자이너들이 작업할 경우가 생기는데, 이러한 문제를 XML을 이용하면 홈페이지를 제작하듯이 손쉽게 UI의 디자인을 변경할 수 있게 된다.

또 포팅의 경우, 시스템의 개발 중 수정된 내용을 컴파일 하여 보드로 포팅 하는 작업이 필요한데, 이때 포팅에 걸리는 시간은 컴파일 될 파일의 용량에 비례한다. 따라서 컴파일 파일에 비하여 용량이 작은 XML문서를 이용하면 포팅시간이 단축되는 것이다.

그 외에도 XML파일만 수정하면 소스의 복잡하고 어려운 부분들을 수정할 필요 없이 비교적 쉽게 다른 임베디드 장비에도 적용 시킬 수 있다는 장점이 있다.[1]

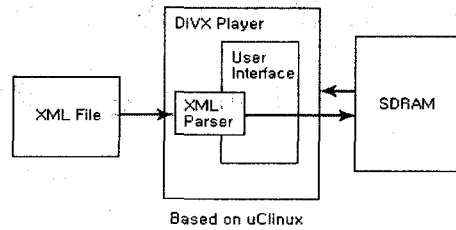
단, 이러한 XML에도 단점은 있다. 데이터로서의 XML이 일반적인 데이터에 비하여 단점으로 지적되는 사항은 데이터뿐만 아니라 태그들까지 저장되어야 하기 때문에 대용량의 자료를 저장하는 데에는 적합하지 않을 수 있다. 이것은 임베디드로 구현되는 시스템에도 마찬가지로 큰 걸림돌로 작용되는데, Divx Player 시작 시 XML Parsing 과정을 거치면서 XML파일의 내용 크기에 따라 프로그램 실행 속도에 영향을 주며 프로그램을 사용하는 내내 메모리에 상주하므로 리소스가 한정된 임베디드 시스템에 부담을 주게 되는 것이다.[2]

3. 임베디드 장비의 리모컨 구현.

3.1 임베디드 장비의 XML 파싱.

앞서 설명한 XML과 임베디드의 구조는 리모컨의 구현과 아주 밀접한 관계가 있다.

먼저 임베디드 기반에서의 XML파싱의 구조는 그림 1로 쉽게 이해할 수 있다. 그림 1에서 보듯이 작동은 XML문서가 파싱된 후 C/C++상에서 데이터 구조화되어 메모리에 상주하게 된다.



(그림 1) DivX Player의 XML파서 위치

3.2 리모컨과 관련한 UI의 동작

리모컨과 UI 구현관계를 살피기 위해선 먼저 UI를 구성하는 XML이 어떤 역할을 하는지 잘 알아야 한다.

```

<osdPage name="MediaLibrary"
  file="wivxplayer.png"
  selectedObject="AudioButton">
<buttonList>
<button name="AudioButton"
  file="audiobutton.png"
  x="165"
  y="105"
  width="376"
  height="60"
  keypad="UTMAButton">
<eventList>
<event name="OnActivate"
command=
"SHOW_AUDIO_PAGE_COMMAND" link="AudioListPage"/>
</eventList>
</button>
<button name="VideoButton"
  file="videobutton.png"
  x="165"
  y="172"
  width="376"
  height="60">
<eventList>
<event name="OnActivate"
command="SHOW_VIDEO_PAGE_COMMAND"
link="VideoListPage"/>
</eventList>
</button>
    
```

(표1) UI를 구성하는 XML파일의 예시

표1을 보면 Divx Player UI의 일부를 구성하는 XML 문서의 일부가 예시되어 있는데, <button>항목은 UI의 메뉴 하나하나를 구현하며, file항목에서 그 이미지를 불러온다. 그 외에 이미지의 구체적인 크기와 모양을 설정하는 항목들로 구성되어 있다. <event>항목은 <button>의 이미지를 리모컨으로 눌렀을 때 일어나는 동작을 지

시한 항목이다.

여기서의 SHOW_VIDEO_PAGE_COMMAND 라는 명령으로 링크된 VideoListPage의 내용을 불러오는 것이다.

```
<command name="SHOW_PAGE_COMMAND" id="10"/>
<command name="SHOW_AUDIO_PAGE_COMMAND" id="11"/>
<command name="SHOW_VIDEO_PAGE_COMMAND" id="12"/>
<command name="SHOW_PICTURES_PAGE_COMMAND" id="13"/>
<command name="SHOW_MESSAGE_COMMAND" id="14"/>
<command name="SHOW_LOADING_FILE_COMMAND" id="15"/>
<command name="SHOW_FILE_INFO_COMMAND" id="16"/>
<command name="SHOW_LOAD_PICTURE_ERROR_COMMAND" id="17"/>
<command name="SHOW_DVD_INFO_COMMAND" id="18"/>
<command name="SHOW_VCD_INFO_COMMAND" id="19"/>
<command name="SHOW_CDDA_INFO_COMMAND" id="20"/>
<command name="SHOW_DVD_TIME_SEARCH_COMMAND" id="21"/>
<command name="SHOW_VCD_TIME_SEARCH_COMMAND" id="22"/>
<command name="SHOW_FILE_TIME_SEARCH_COMMAND" id="23"/>
<command name="SHOW_WMVHD_TIME_SEARCH_COMMAND" id="24"/>
<command name="SHOW_WMVHD_INFO_COMMAND" id="25"/>
```

(표2)리모컨 명령에 관련된 XML 코드

<event>의 XML 명령어는 표2의 XML 코드를 통해 컴파일 소스와 연동된다.

이 명령코드들이 원하는 동작을 하기 위해선 컴파일 소스에 그 명령어들이 정의되어 있어야 하며 표3의 코드들이 바로 XML문서에 있는 명령어들을 정의한 내용이다.

표3의 각 명령어는 다른 프로그래밍 파일에 더욱 자세한 세부동작이 정의되며 명령에 대한 호출은 명령어에 할당된 고유번호에 의해 이루어진다.

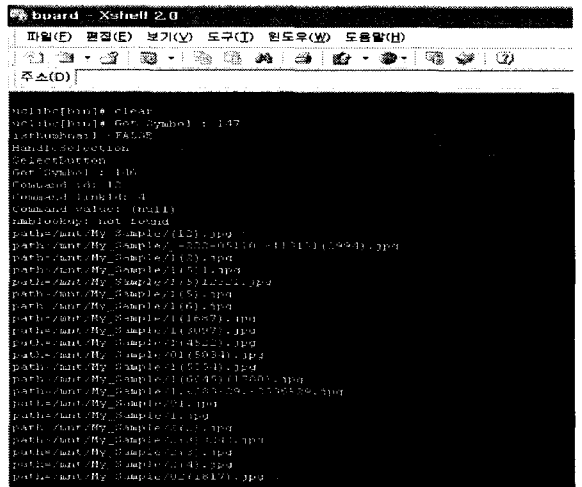
이처럼 XML파일은 누구나 쉽게 알아보기 쉬운 구조를 바탕으로 리모컨 동작과 같은 복잡한 프로그래밍 소스의 구동도 매우 쉽게 이해하고 수정하는 것이 가능한 것이다.

만약, 개발자가 UI의 불필요한 기능을 삭제하고자 하면, 컴파일 소스를 손댈 필요 없이 표1과 표2의 XML문서부분을 수정해주면 된다.

그림2는 리모컨으로 위의 명령코드를 실제로 실행할 때 컴파일 화면에 경과가 출력되는 모습이며, 그림3은 Divx Player의 동작 모습이다.

```
#define SHOW_PAGE_COMMAND 10
#define SHOW_AUDIO_PAGE_COMMAND 11
#define SHOW_VIDEO_PAGE_COMMAND 12
#define SHOW_PICTURES_PAGE_COMMAND 13
#define SHOW_MESSAGE_COMMAND 14
#define SHOW_LOADING_FILE_COMMAND 15
#define SHOW_FILE_INFO_COMMAND 16
#define SHOW_LOAD_PICTURE_ERROR_COMMAND 17
#define SHOW_DVD_INFO_COMMAND 18
#define SHOW_VCD_INFO_COMMAND 19
#define SHOW_CDDA_INFO_COMMAND 20
#define SHOW_DVD_TIME_SEARCH_COMMAND 21
#define SHOW_VCD_TIME_SEARCH_COMMAND 22
#define SHOW_FILE_TIME_SEARCH_COMMAND 23
#define SHOW_WMVHD_TIME_SEARCH_COMMAND 24
#define SHOW_WMVHD_INFO_COMMAND 25
```

(표3)XML명령과 연동된 컴파일 소스 코드



(그림2) 리모컨 명령실행 결과



(그림3) Divx Player에서 리모컨으로 메뉴를 선택하는 모습

4. 결론 및 연구과제

본 논문에서는 Divx Player에 관한 리모컨 동작과 UI 구성에 대하여 설명하였다.

본문에서는 Divx Player UI의 구현과 리모컨에 대하여 서술하기 전에 XML에 대하여 많은 언급을 하였고, 실제로 리모컨 구현에 관하여서도 XML을 빼놓을 수 없었는데, 그것은 XML이 중요한 역할을 맡는다는 것을 알려주는 것이다.

만약 Divx Player가 XML문서를 전혀 사용하지 않았거나, 본문에서 XML을 제외하고 구체적인 설명을 했다면 상당히 이해하기 어렵고, 긴 설명의 내용이 되었을 것이다. XML을 이용한 임베디드 인터페이스의 구현은 XML을 사용하지 않던 형식에 비하여 파서의 구현 등 추가적인 작업이 요구되나, UI수정이나 본문과 같은 리모컨의 설계 및 작업을 프로그래밍에 전문적인 지식을 갖추지 못한 사람도 쉽고 간단하게 수행할 수 있게 하였다. 또 수정 후 Divx Player의 소스파일을 컴파일 할 필요 없이 적은용량의 XML 파일만 포팅 해도 되기 때문에 작업속도가 빨라져 많은 편의를 제공할 것이다.

이와 같은 XML을 이용한 작업의 편의성 덕분에 본문에서 제시한 리모컨뿐만 아니라 다른 분야에서도 XML을 활용하면 많은 도움이 되리라고 예상된다.

하지만 시스템의 리소스가 충분치 않은 임베디드 시스템의 특성상 XML 파싱을 통해 데이터들이 메모리에 상주에 있다는 것은 많은 부담으로 작용한다. 특히 최근 HD급 영상인 WMV9까지 지원하는 DiVX Player에서 메모리 관리는 SYSTEM의 안정성 측면에서 매우 중요한 부분인데, DiVX Player의 많은 부분을 XML파일에서 파싱된 데이터에 의존한다면 정상적인 동작을 기대하기 힘들 것이다. 그러므로 임베디드 시스템에서의 XML과 다 적용은 피해야 할 것이다.[3]

참 고 문 헌

[1] <http://www.w3.org/TR/REC-xml/>
(W3C standard: eXtensible Markup Language(XML))

[2] <http://white.chungbuk.ac.kr/>
(XML의 장단점 및 응용분야)

[3] "Korea Embedded System for Linux",
(<http://www.ksel.co.kr>)