

모바일 환경을 위한 대화형 언어의 정의

윤민홍^o 류은석 유혁
고려대학교 컴퓨터학과

{mhyun, esryu, hxy}@os.korea.ac.kr

Design of Mobile Interactive Language

Min-Hong Yun^o, Chuck Yoo
Dept. of Computer Science and Engineering, Korea University

요약

SMIL (Synchronized Multimedia Integration Language)은 멀티미디어(multimedia) 객체(object) 간의 동기화를 목표로 만들어진 markup language이다. SMIL은 시간과 사용자의 동작 포함한 많은 것들을 사건(event)으로 정의하여 사건의 발생에 따라 멀티미디어 객체들의 배열과 상영시간을 결정한다. 그렇지만 SMIL의 최대 목표는 동기화(synchronization)에 있기 때문에 대화형 언어(interactive language)에는 부족한 면이 있다. 본 논문에서는 SMIL에 대해서 간략히 살펴보고 SMIL을 기반으로 모바일(mobil) 환경을 위한 대화형 언어를 정의한다.

1. 서론

대화형 언어란 TV 방송과 같은 일방적인 제공 형태의 서비스가 아닌, 사용자의 반응에 따라 제공되는 콘텐츠(contents)의 내용이나 형식이 변화하는 서비스를 위한 언어를 말한다. 가정에서는 셋탑박스(set-top box)를 통해 TV로 체험하고 있는 것을 지원하기 위한 언어가 그것이다. 대화형 언어를 통해서 사용자들은 이미 결정된 스토리가 아닌 사용자가 직접 만들어가는 동적인 스토리를 구성해나갈 수 있는 것이다.

2001년 8월 World Wide Web Consortium (W3C)에 의해 SMIL 2.0이 발표되면서 인터넷 상의 대화형 언어가 자리잡아가고 있다. SMIL은 Synchronized Multimedia Integration Language의 약자로 객체들간의 시간적인 동기화를 중점으로 정의된 언어이다[1]. 리얼 플레이어(real player)에서 첫 화면이 바로 SMIL을 이용해서 멀티미디어 객체들간의 동기화를 맞춘 화면이다.

대화형 언어가 사용되는 곳들은 많지만 현재는

모바일(mobile)환경에서 사용되는 경향이 있다. 현재 국내의 모바일 환경에서의 멀티미디어는 동영상 플레이어(player)나 약간의 MMS (Multimedia Messaging Service)에 그쳤다. 그렇지만 2002년 Nokia에서 SMIL을 탑재한 휴대폰을 출시하면서 모바일 환경에서도 대화형 미디어를 원하고 있다. 하지만 SMIL의 주된 초점은 멀티미디어 객체의 동기화에 있기 때문에 동기화를 제외한 다른 측면에서는 대화형 미디어를 저작하는데 부족한 면이 있다. 이 논문에서는 모바일 환경에서 대화형 미디어를 지원하기 위한 대화형 언어를 정의한다.

2. 관련 연구

SMIL은 1998년 버전 1.0이 발표되고[2] 리얼 플레이어 G2, 퀵타임(Quick Time) 4.1, 인터넷 익스플로러(Internet Explorer) 5.5 등에 탑재되었고 2001년 많은 부분은 추가해 SMIL 2.0이 발표되었다[3].

- SMIL은 XML을 사용해 표현한다.
- timing과 synchronization grouping이 SMIL의 core를 담당한다.

이 논문에서는 SMIL의 소개를 위해 핵심부분인 구조적 시간과 시간적 동기화만 소개한다[4].

본 연구는 한국정보통신대학교 디지털미디어연구소의 정보통신연구개발사업의 연구비 지원에 의하여 수행되었음

2.1 구조적 시간 (Structural Timeline)

SMIL 은 크게 4 개의 구조적인 타임라인을 가지고 있다. <par>, <switch>, <seq>, <excl> 등 이다[4].

<par> (parallel)는 내부의 멀티미디어 객체들이 모두 동일한 시간에 시작한다는 가정을 가지고 병렬(parallel) 적으로 같은 시작 시간을 갖는다. 즉 모든 객체의 시작되는 시점이 같다는 것이다. 단지 그 하나의 시점으로부터 언제 객체가 화면에 렌더링(rendering) 될 것인가의 지연시간(delay)만 각 멀티미디어 객체들간에 차이가 있다(그림 1.a).

<switch>는 내부의 자식(children)들 중 테스트를 거쳐 적합한 자식을 렌더링 한다. 시스템(system)의 성능에 맞게 test 를 하는 것으로, 디스플레이의 크기나 네트워크 환경에 따라 적절한 자식 노드를 찾아서 렌더링 한다.

<seq> (sequence)는 내부의 모든 멀티미디어 객체들이 시간의 순서에 따라 렌더링 되게 한다. 하나의 멀티미디어 객체가 렌더링된 다음에야 다른 멀티미디어 객체가 렌더링 될 수 있다(그림 1.b).

<excl> (exclusive)에서는 동시에 2 개 이상의 객체가 렌더링 될 수 없다. <seq>와는 달리 시간에 따른 스케줄링이 아닌 이벤트에 따라 스케줄링이 된다. 이것은 이벤트(event)에 기반한 멀티미디어 렌더링 스케줄링을 하는데 사용할 수 있다(그림 1.c).

것인지를 스케줄하는 비결정적(non-deterministic) 특성을 지니고 있기 때문이다.

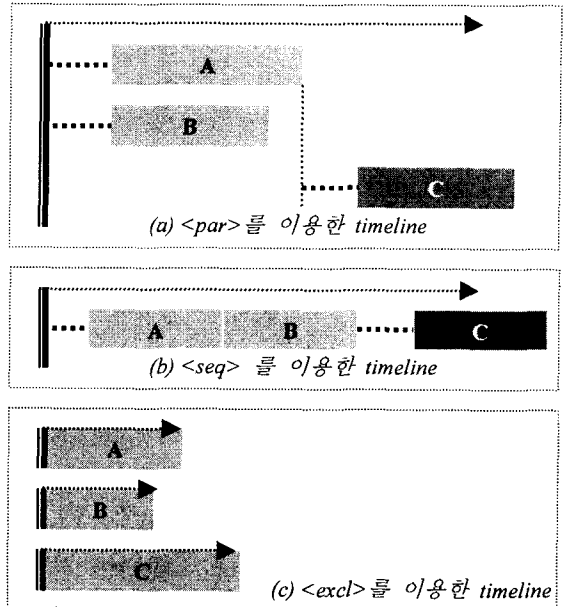


그림 1. Structural Timeline
(a) A, B, C 가 parallel 하게 렌더링 된다. C 는 A 가 끝나고 임의의 시간이 지난 후 렌더링 (b) A, B, C 가 순차적으로 렌더링된다. B 는 A 가 종료된 직후, C 는 B 가 종료되고 임의의 시간이 지난 후 렌더링 (c) A, B, C 가 각각 독립되게 렌더링된다. 단, 한번에 하나의 멀티미디어 객체만 렌더링

2.2 시간과 동기화 (Time and Synchronization)

가장 간단한 시간제어(timing control)는 <begin>, <end>, <dur>이다. 객체가 parent node 의 시작시각을 중심으로 언제 시작할 것 이며 언제 끝낼 것 인지 그리고 얼마나 지속시킬 것인지를 정의하게 된다. 그 외에도 <endsync>는 자식 노드들 중 어떤 노드의 렌더링이 종료되면 현 노드를 종료할 지를 결정한다. <fill>은 렌더링이 끝나고 남은 시간동안 렌더링하던 것을 종료할 것인지 렌더링의 가장 마지막 장면 또는 가장 마지막 소리를 들려줄 것인지를 결정하게된다. <repeatCount>와 <repeatDur>은 각각 횟수와 시간으로 멀티미디어 객체를 렌더링할 한계를 결정한다. 뿐만 아니라 SMIL 은 시간 구현의 오차를 허용하기 위해 <syncBehavior>, <syncTolerance>, <syncMaster> 등 의 정의도 가지고 있다[5].

3. Vector Graphics

SMIL 이 가장 중요하게 생각했던 것들은 객체들간의 동기화 이다. SMIL 을 대화형 언어라 부르는 까닭은 시간적 동기화에만 신경을 쓴 것이 아니고, 사용자의 반응을 이벤트로 정의하고 이 이벤트에 따라 어떤 멀티미디어 객체를 렌더링 할

사용자의 반응을 받아서 대응하기 때문에 대화형 미디어의 특성을 가지고는 있지만, SMIL 은 추상화 시키면 객체들의 동기화를 하는 언어이다. 현재 대화형 언어의 큰 흐름을 이루고 있는 것은 매크로미디어(Macromedia)사의 플래쉬(Flash)이다. 플래쉬도 객체간의 동기화를 맞추어 주고는 있지만 플래쉬의 또 다른 가장 큰 특징은 벡터 그래픽스(vector graphics)를 사용하고 있다는 점이다. 벡터 그래픽스는 래스터 이미지 (raster image)와는 달리 확대하거나 축소해도 화질의 변화가 없다는 장점을 가지고 있다.

모바일 환경을 위한 대화형 미디어가 벡터 그래픽스를 사용하게 될 것이라는 것은 당연하다. 간단한 'next' 버튼을 만들더라도 래스터 이미지 보다는 벡터 그래픽스를 사용하는 것이 훨씬 더 작은 크기에도 고화질을 보여줄 수 있기 때문이다. 따라서 이 논문에서 제안하는 모바일 환경을 위한 대화형 언어는 벡터 그래픽스를 포함한다. 대화형 미디어는 사용자에게 많은 반응을 받기 위해서 작은 크기의 많은 이미지들이 필요할 것인데 이것들을

모두 외부에 위치한 멀티미디어 객체로 다루기 보다는 대화형 언어가 가지고 있는 내부의 객체로 다루는 것이 응답속도나 외부의 멀티미디어 객체를 받는데 소요되는 컨택선에 따르는 비용을 고려하면 효과적이다.

벡터 그래픽스를 사용하기 위해 이 논문은 모든 벡터 그래픽스 객체를 언어의 앞부분에 정의하도록 제안한다. 많은 벡터 그래픽스 객체들이 재사용되기 때문이다. 이 객체들은 각기 ID를 가지고 있으며 이 ID를 사용해 객체를 접근 하는 것이다. 하나의 벡터 그래픽스 객체는 여러 개의 작은 객체들로 구성될 수 있는데, 벡터 그래픽스 객체의 불필요한 제정의 불필요를 피하기 위해 벡터 그래픽스 객체의 일부분만 변경되는 경우에는 변경된 부분만 대체하는 방법을 사용한다.

4. Binary Format

벡터 그래픽스를 내포한다는 것은 파일의 크기가 커진다는 것을 의미한다. 플래쉬로 이루어진 애니메이션(animation)의 경우 해당하는 플래쉬 파일의 크기가 상당히 커서 전체를 인터넷을 통해 전체를 다운로드(download)하는 데에만 상당한 시간이 걸린다. 만약 이것들이 XML 형태의 텍스트(text)로 되어 있다면 파일의 크기는 더욱 커지기 때문에 다운로드하는 데에 걸리는 시간은 더욱 늘어날 것이다.

바이너리(binary)형태를 사용하는 데에는 몇 가지 장점이 있다. 앞서 말한 XML 이라는 텍스트로 이루어진 파일을 압축할 수 있다. 또 바이너리 형태로 만들면 대화형 미디어를 저작하는 과정에서 발생할 수 있는 ID의 충돌문제와 가독성을 위해 ID에 많은 비트(bit)를 할당하는 문제를 줄일 수 있다. 대화형 미디어 플레이어에서는 XML 파서(parser)를 필요로 하지 않기 때문에 플레이어의 크기가 작아질 수 있다.

그렇지만 플래쉬의 경우 비트 연산(bit operation)이 많기 때문에 렌더링을 위한 처리과정에 비싼 연산을 수행하게 된다. 따라서 이 논문에서는 바이트 단위로 할당(aligned)해 모바일 단말기상의 플레이어의 부담을 줄이는 언어를 제안한다. 이 논문이 제안하는 언어는 SMIL 문서를 파싱하고 코딩하는 과정을 통한 바이너리 형태의 생성이 아닌 저작과정에서 대화형 언어가 바이너리 형태를 지니기 때문에 언어의 특성을 더욱 많이 반영하기 때문에 더 높은 압축률을 얻을 수 있다[6].

5. 결론 및 향후 연구과제

대화형 미디어는 단방향 방식의 방송의 단점인 사용자의 의견이 멀티미디어 콘텐츠에 전혀 반영

되지 않는다는 점을 극복한 미디어이다. 지금 나와있는 대표적인 대화형 미디어를 위한 언어에는 W3C에서 발표한 SMIL이 있는데, 이것은 멀티미디어 객체들간의 동기화에 초점을 맞춘 언어이다. SMIL을 모바일 환경에 적합하게 하기위해 이 논문은 벡터 그래픽스와 바이너리 형태를 제안했다. 벡터 그래픽스를 내장함으로써 응답속도와 컨택선에 소요되는 비용에 대한 이익을 얻을 수 있고, 바이너리 형태를 채택함으로써 파일 사이즈를 작게 하고 ID의 충돌을 제거하는 등의 이익을 얻을 수 있다.

이 논문이 제안하는 것은 모바일 환경에서 대화형 멀티미디어를 위한 언어이다. 벡터 그래픽스를 사용하고 바이너리 형태를 취했지만 이 논문은 스트리밍에 대한 고려를 하지 않았다. 스트리밍을 할 경우 스트리밍을 책임지는 부분이 멀티미디어 콘텐츠 디코더에 존재할 것인지 아니면 대화형 언어에 존재할 것인지에 대한 연구가 필요하고, 실제 구현을 통해 기존의 SMIL과 비교해서 어떤 차이를 보이든지도 연구해야 할 것이다.

참고 문헌

- [1] Rutledge, L., "SMIL 2.0: XML for Web multimedia", *Internet Computing, IEEE vol.5, Sep/Oct 2001*, 78-84
- [2] "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", W3C Recommendation, 15 June, 1998, <http://www.w3.org/TR/REC-smil/>
- [3] "Synchronized Multimedia Integration Language (SMIL 2.0)", W3C Recommendation, 7 August, 2001, <http://www.w3.org/TR/smil20/>
- [4] Bulterman, D.C.A., "SMIL 2.0 Part1: Overview, Concepts, and Structure", *Multimedia, IEEE vol.8, Oct-Dec 2001*, pp. 82-88
- [5] Bulterman, D.C.A., "Examples and comparisons", *Multimedia, IEEE vol.9, Jan-Mar 2002*, pp. 74-84
- [6] Chia-Yuan Teng, "Compression of SMIL Documents", *Data Compression Conference, 2000. Proceedings. DCC 2000*, 2000, p572