

SVG 파일 기반의 아바타 시스템을 위한 SVGStore의 설계 및 구현

김은주 김원중

순천대학교 컴퓨터학과

{heaven03, dbcho, kwj}@suncheon.ac.kr

The Design and Implementation of SVGStore for Avata System on SVG File

EunJu Kim LeeGi Cho WonJung Kim

Dept. of Computer Science, Suncheon National University

요 약

기존의 아바타 시스템은 GIF 파일 기반으로 한정된 아바타 캐릭터를 이용하여 다양한 표현이 제한적이었던 SVG 파일을 기반으로 하는 아바타 시스템의 경우 사용자가 능동적으로 아바타를 수정할 수 있어 다양한 아바타 캐릭터를 만들 수 있는 장점이 있다. 또한 기존 시스템은 아바타를 위한 GIF파일의 구성 정보만 유지하고 있었다면 SVG 파일 기반의 아바타 시스템의 경우 각 사용자별로 아바타를 표현하는 별도의 SVG 파일을 가지며, 사용자가 각 개인의 아바타를 수정함으로써 갱신 연산이 빈번하게 발생한다. 이에 효과적인 갱신 연산을 수행하는 SVG 파일을 저장하는 저장소가 필요하다. SVGStore는 각각의 아바타에 대응하는 SVG 파일을 저장하고 빈번한 갱신 연산을 효율적으로 제공한다.

1. 서 론

아바타(Avata)의 사전적인 의미는 분신(分身), 화신(化身)으로서, 인터넷 사용자들에게는 자신의 개성을 대신 표현하는 일종의 사이버 캐릭터이다. 특히 익명성이 강조되는 인터넷에서 또 다른 자기의 개성을 표출할 수 있기 때문에 사용자들의 호응이 좋아 대부분의 포털 사이트나 게임 사이트에서는 아바타 서비스를 제공하고 있다. 그러나 현재 서비스되고 있는 대부분의 아바타 시스템은 하나나 두개의 캐릭터에 단순하게 옷이나 장신구로 다양함을 표시하고 있으며 몇몇 사이트의 경우 캐릭터를 다양하게 제공하고 있으나 캐릭터에 대하여 사용자가 원하는 형태로의 변경을 불가능하여 제한적으로 운영되고 있다. 그래서 제안된 시스템이 SVG(Scalable Vector Graphic)파일을 기반으로 하는 아바타 시스템이다. SVG 파일 기반의 아바타 시스템은 사용자가 웹상에서 아바타를 직접 수정하므로써 다양한 캐릭터의 생성이 가능하기 때문에 기존 시스템에서 제공하지 못하는 캐릭터의 다양성 및 개성을 제공할 수 있다는 장점이 있다. 그러나, SVG기반의 아바타 시스템은 각 사용자별로 캐릭터를 위한 SVG파일이 별도로 존재하기 때문에 각 SVG 파일을 저장하기 위한 별도의 저장 공간이 필요하다[1].

2001년 순천대학교에서 개발한 SVG 기반의 아바타 시스템[1]은 SVG 파일 저장소로 MS SQL Server 2000을 사용하였다. MS SQL Server 2000의 XML 저장 기법을 이용하였으나 SVG 파일에 대한 갱신 연산이 자주 발생하는 경우 시스템의 성능이 저하되는 문제가 발생하였다. 이에 본 논문에서는 기존의 SVG 파일 저장소를 대신하여 SVG 파일을 관리하고 갱신 연산을 효과적으로 수행할 수 있는 SVGStore를 설계, 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구

로 SVG와 XML 파일을 저장하기 위한 기법에 대해 소개하고, 3장은 SVG 파일의 갱신 연산을 효과적으로 지원하는 SVGStore에 대하여 기술하며, 4장에서는 본 논문의 결론과 향후 연구과제를 제시하였다.

2. 관련연구

2.1 SVG

XML(eXtensible Graphic Language) 그래픽 표준인 SVG는 XML이 가지는 개방성(Openness), 상호 운용성(Interoperability)등의 장점을 수용한 벡터 기반의 그래픽 파일로서 SMIL(Synchronized Multimedia Integration Language), GML(Geography Markup Language), MathML 등 다른 XML 언어들과 유기적으로 결합하여 광고, 프로세스 컨트롤, 지리정보, 교육등 다양한 분야에 적용이 가능하다[2].

또한 SVG는 텍스트 기반으로서 그래픽 파일에 대한 검색을 효율적으로 진행할 수 있으며 응용 프로그램들이 SVG 문서를 쉽게 사용할 수 있으며, DOM(Document Object Model) 인터페이스를 이용하여 선(Line), 폴리곤(Polygon), 텍스트(Text), 이미지(Image)등의 모든 그래픽 요소에 쉽게 접근할 수 있어 데이터베이스와 연동하여 웹 기반 그래픽 문서를 동적으로 생성할 수 있는 특징이 있다[2].

SVG의 장점은 변환이 자유로운 벡터 그래픽이며, 1,600만 이상의 색상들을 제대로 지원하며, XML, HTML4, XHTML과 호환이 가능하며, CSS, XSL, DOM등과 함께 사용이 가능하다. 단점으로는 웹 상에서 SVG 파일을 보려고 하는 경우 SVG Viewer를 설치해야 한다[3].

2.2 XML의 저장 방법

XML 데이터를 저장하기 위해서는 저장 시스템에 의한 방법과 저장 기법에 의한 방법으로 구분할 수 있다.

(1) 저장 시스템에 의한 방법

XML을 위한 저장 시스템은 크게 두가지 방법으로 구현할 수 있다. 첫 번째 방법은 Lore(Lightweight, Object Repository)[4]나 충남대학교의 XDMS처럼 새로운 데이터 모델을 지원하는 데이터베이스 시스템을 하부 저장 장치로부터 질의어 처리기까지 모두 만드는 것이다. XML은 기존의 관계형 데이터베이스의 릴레이션이나 객체지향 데이터베이스의 클래스와 다르게 가변적이며, 질의의 형태도 정규 경로식과 같이 기존 데이터베이스에서 사용되던 질의어와 다르므로 이런 특징을 지원하기 위하여 전용 데이터베이스 시스템을 구성하는 것으로서 새로운 모델을 부조화 없이 잘 표현하고 효과적인 질의가 가능하다[5].

두 번째 방법은 기존의 데이터베이스 시스템을 이용하는 방식이다. 이 방식은 XML 파일을 저장하기 위한 테이블이나 클래스를 생성한 다음 XML에 대한 질의를 데이터베이스에 대한 질의어로 변환하여 원하는 결과를 얻는 방법이다. 장점은 기존의 시스템이 가지는 장점을 활용할 수 있지만 단점으로는 RDBMS의 경우 검색시 다수의 테이블에 대한 고비용의 조인 연산을 수행해야 하며 검색 결과를 출력하는데 많은 노력이 필요하며, ODBMS의 경우 대용량의 데이터에 대한 복잡한 형태의 질의를 처리하는 능력이 미숙하다는 점이다[5].

(2) 저장 기법에 의한 방법

XML문서를 저장하는 방법으로는 분할 저장(Decomposition) 기법과 가상 분할(Virtual Fragmentation) 기법이 있다.

분할 저장 기법은 XML문서를 엘리먼트 단위로 쪼개어 저장하고 검색시 구조 정보를 참조하여 각각의 엘리먼트들을 조합하여 처리하는 방식이다.

분할 저장 기법의 장점은 삽입, 수정 또는 삭제하는 경우 해당 엘리먼트만 수정하면 되므로 문서의 편집 및 관리가 쉽고 동일한 내용을 가지는 노드를 공유할 수 있다. 단점으로는 각각의 엘리먼트의 내용을 조합하여 결과를 생성해야 하므로 검색 과정이 복잡하고 검색 시간이 오래 걸린다[6].

가상 분할 기법은 XML 문서를 하나의 BLOB형태로 데이터베이스에 저장한 후 문서 내에 있는 각각의 단말 노드에 대한 실제 문서상의 시작 오프셋과 종료 오프셋을 저장하는 방식이다.

가상 분할 방식의 장점은 문서를 한꺼번에 저장하므로 단말노드에 대한 검색의 경우 위치 정보를 이용한 구조적 검색이 가능하고 검색 효율이 상대적으로 우수하다. 단점으로는 XML 문서의 일부분에 대한 엘리먼트는 추가, 삭제되는 등의 갱신 연산이 발생하는 경우 문서내의 모든 엘리먼트에 대한 위치 정보를 수정해야 하며, 데이터의 일관성을 유지하기 위한 별도의 롤이 필요하다[6].

3. SVGStore

2001년 순천대학교에서 개발한 SVG기반의 아바타 시스템은 기존의 GIF 기반의 아바타 시스템에 비해 사용자가 직접 캐릭터를 수정할 수 있으므로 캐릭터를 표현하는 SVG 파일의 빈번한 갱신 연산이 발생한다. 기존의 시스템의 경우 SVG파일 저장소로 MS SQL Server 2000을 사용하였으며, 가상 분할 기법을 이용하여 저장함으로써 빈번한 갱신 연산시 시스템의 성능이 저하되는 문제점을 보여 왔다. 이에 본 논문에서는 아바타 시스템의 SVG파일의 갱신 연산을 효율적으로 지원하기 위하여 SVGStore를 설계, 구현하였다.

SVGStore는 SVG 파일의 갱신 연산의 성능을 향상시키기 위하여 <Face>, <Nose>, <Eye>등과 같이 의미적인 엘리먼트를 저장 단위로 분할하여 저장하는 분할 저장 기법을 사용하였다. 표 1은 아바타를 표현하기 위하여 사용된 SVG 파일의 DTD이며, 그림 1은 본 논문에서 설계, 구현한 SVGStore의 구조로서 Element Repository, Character XML File Template, User Character Temporary Repository, User Information Database, User Session Database, User Session Monitor로 구성된다.

표 1 아바타를 위한 SVG 파일의 DTD

```

<!ELEMENT CHARACTER (FACE, NOSE, EAR, MOUSE, EAR, HAIR, EYEBROWS)>
<!ELEMENT FACE (path)>
<!ELEMENT NOSE (path)>
<!ELEMENT EYE (path)>
<!ELEMENT EAR (path)>
<!ELEMENT HAIR (path)>
<!ELEMENT EYEBROWS (path)>
<!ELEMENT path EMPTY>
<!ATTLIST
    path
        id ID #REQUIRED
        class CDATA #IMPLIED
        d CDATA #REQUIRED
    >
    
```

3.1 Element Repository

Element Repository는 <FACE>, <NOSE>, <EYE>, <MOUSE>, <EAR>, <HAIR>, <EYEBROWS>등의 각각의 엘리먼트 단위로 XML파일을 분할하여 저장하는 것으로서 각각의 Element Repository는 사용자 아이디에 해당하는 UID로 인덱스되어 저장된다.

3.2 Character XML File Template

Character XML File Template는 표 1과 같이 구성되는 SVG 파일 중 공통적으로 구성되는 부분을 별도로 저장하고 있는 공간이다. 사용자의 요청이 발생하는 경우

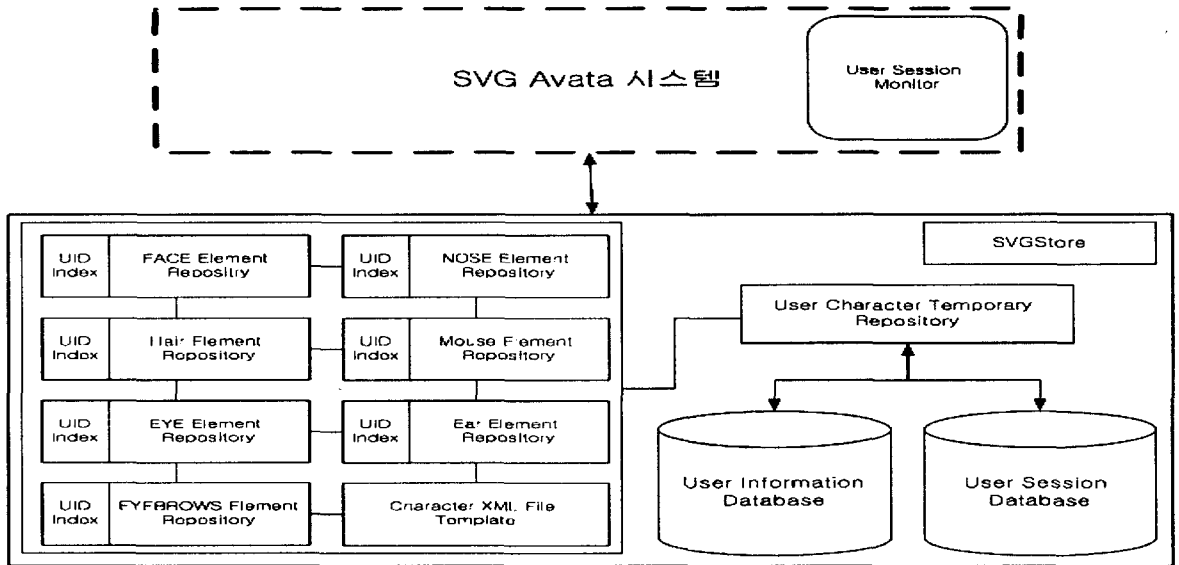


그림 1 SVGStore의 구조

각각의 엘리먼트와 결합되어 사용자의 아바타를 표현하기 위한 SVG로 파일로 변환된다.

3.3 User Character Temporary Repository

XML파일을 분할하여 저장하는 경우 갱신 연산에 비해 검색 연산의 성능은 떨어진다. 이런 문제를 해결하기 위하여 User Character Temporary Repository는 사용자가 로그인하여 세션 정보를 유지하는 동안 아바타를 표현하는 SVG파일을 저장하며 사용자가 SVG 파일을 요청할 때마다 바로 제공하는 역할을 수행한다. 사용자 세션 정보가 종료되는 경우 해당 사용자의 SVG 파일은 제거된다.

3.4 User Information Database와 User Session Database

User Information Database는 사용자의 개인 정보를 저장하는 곳으로서 사용자가 가입하게 되면 Element Repository를 위한 사용자가 만든 사용자 아이디 외에 UID가 자동적으로 할당되어 저장된다. User Session Database는 아바타 시스템을 사용하는 경우 해당 사용자의 세션 정보가 저장된다.

3.5 User Session Monitor

User Session Monitor는 기존의 아바타 시스템에 추가된 모듈로서 사용자가 세션 정보가 종료되는 경우 해당 사용자의 아바타를 표현하는 SVG파일을 User Character Temporary Repository에서 제거하는 역할을 수행한다.

4. 결 론

기존의 아바타 시스템의 경우 캐릭터가 고정되어 있어 수정이 불가능하다면 SVG기반의 아바타 시스템의 경우 사용자가 웹 상에서 직접 수정이 가능하다는 장점이 있다. 그러나, 각 사용자별로 별도의 SVG 파일이 존재하므로 SVG파일을 저장하는 저장소의 갱신 연산 성능이 중요하다. 본 논문에서 구현한 SVGStore를 사용하는 경우 기존의 MS SQL Server를 이용한 것보다 30% 나은 갱신 연산 성능을 제공하였다.

앞으로의 연구 과제는 효율적인 검색을 위한 XML-QL, XQL등과 같은 표준화된 질의 언어를 지원하도록 연구되어야 할 것이다.

참고문헌

- [1] 장미화, "XML/SVG를 이용한 확장 가능한 Web Character 구현", 순천대학교 석사 논문, 2001.12.
- [2] 나방현, 심규찬, 이종연, "XML 그래픽 입문", 21세기사, 2001.7.15
- [3] http://network.hanbitbook.co.kr/view_news_print.htm?serial=575
- [4] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom, "Lore: A Database Management System for Semistructured Data," SIGMOD Record 26(3), pp.54-66, 1997.
- [5] 박상원, 정재욱, 정태선, 김형주, "XML과 데이터베이스", 정보과학회지 제19권 제1호, 2001.1.
- [6] 김훈, 한상용, 홍의경, "XML 문서 저장 시스템", 데이터베이스 연구회지, 16권 2호, 2000.12.