
구조화된 벡터 그래픽 표현을 위한 SVG 편집시스템의 프로토타입 설계

김철순^{*} · 김창수^{*} · 정희경^{*}

^{*}배재대학교 컴퓨터공학과

Design of prototype that SVG editing system

Chul-Soon Kim^{*} · Chang-Su Kim^{*} · Hoe-Kyoung Jung^{*}

^{*}Dept. of Computer Engineering Paichai University

E-mail : {cskim, ddoja69, hkjung}@mail.pcu.ac.kr

요 약

인터넷의 급속한 발전은 표현의 고급화와 보편성을 제공하는 웹을 주축으로 하여 이루어져 왔으나, 웹 상에서 표현중심의 그래픽처리에는 비구조적인 방법에 따른 많은 문제점을 가지고 있다. 이러한 문제점을 보완하고자 W3C에서는 웹 상에서 벡터그래픽 표현의 효율적인 표현 및 저장, 공유를 가능하게 하기 위해 XML 어플리케이션인 SVG를 제정하였다. 이에 본 논문에서는 SVG 문서를 이용하여 그래픽 편집기 형태로 SVG의 구조적 편집 및 화면에 표현되는 그래픽 객체를 직접 편집할 수 있는 인터페이스를 통해 벡터그래픽을 편집하고 표현하기 위한 SVG 편집시스템을 설계하였다. 본 시스템은 지리정보시스템이나 모바일 서비스 환경 등에서 XML기반의 그래픽을 표현하는 효율적인 처리 등에 유용하게 사용할 수 있을 것이다.

ABSTRACT

Development of internet had centered on that offer universality of expression. But, graphic processing of expression center has many inconveniences because of method that is not structural in World Wide Web. Announced SVG that is XML application to permit efficient expression and storage of vector graphic expression, share in W3C to supplement these things. Therefore, because using SVG document in this treatise, edit vector graphic through interface that could edit expressed graphic object directly structural edit and screen that is offered in graphic edit system form and designed SVG edit system to express. This system may use usefully to efficient XML graphics processing on the GIS system and Mobile service environment and so on.

키워드

XML, SVG, 그래픽편집기, XML그래픽

I. 서 론

인터넷의 급속한 발전은 기존의 정적인 웹을 탈피하여 사용자들로 하여금 더욱더 동적이고 다양한 콘텐츠를 요구하는 형태로 바뀌어 가는 실정이다. 따라서 이러한 사용자들의 욕구를 충족시키기 위해 웹 상에서 표현의 고급화와 체계적이고 구조적인 방식의 콘텐츠를 생성하고 관리하기 위한 방법이 논의되었고, 이를 충족시키기 위한 솔루션이 요구되었다.

W3C에서는 웹 상에서 벡터그래픽 표현의 효율

적인 처리와 저장 및 공유를 가능하게 하는 XML[1]에서 파생된 어플리케이션인 SVG(Scalable Vector Graphics)[2]를 제정하였다. SVG는 기존의 웹에서 사용되던 비트맵 기반의 디스플레이보다 훨씬 풍부하고 인터랙티브한 그래픽 표현을 제공하기 때문에 기능이나 장치 호환성의 희생 없이도 완전하게 벡터 그래픽을 표현한다. 하지만 복잡한 SVG 구문을 자세히 알고 직접 작성할 수 있는 사용자가 많지 않기 때문에 일반 사용자

들이 편리하고 간단하게 SVG 그래픽을 구현할 수 있는 관련 저작도구가 요구되고 있는 실정이다. 이에 본 논문에서는 일반 사용자들이 손쉽게 그래픽 객체를 직접 저작함에 따라 복잡한 SVG 구문을 자동으로 생성해주기 위해 SVG 1.1을 기반으로 한 SVG 편집시스템의 프로토타입을 설계하였다.

본 논문의 구성은 다음과 같다. 본문에서는 SVG와 관련된 기술 및 특징에 관한 연구에 대해 알아보고, 전체 시스템의 구성과 각각의 구성요소 내에서 처리되는 모듈을 설계한 후, 마지막으로 결론 및 향후 연구과제를 제시한다.

II. 본 론

1. SVG 관련 연구

XML 어플리케이션 중의 하나인 SVG는 W3C에 의해 2001년에 SVG 1.0이 권고안으로 발표되었고, 현재는 SVG 1.1이 2003년 1월에 권고안으로 발표되었다. SVG는 2차원 그래픽을 표현하기 위해 XML을 기반으로 만들어진 XML 그래픽 표준이다. 실시간 데이터로부터 고품질의 다이나믹한 그래픽을 만들어 낼 수 있기 때문에 전자상거래, 지리정보, 교육, 광고 등의 분야에 적용될 수 있다.

SVG가 웹의 전반적인 기반구조에 적합한 이유는 첫째, XML 구문으로 되어있다는 것이다. 이는 SVG 역시 다른 XML 문서와 통합될 수 있다. 예를 들어 SVG문서를 CML(Chemical Markup Language)과 결합하여 화학식의 그래픽적인 표현을 가능하게 해준다. 둘째, SVG는 웹서비스에 있는 그래픽과 완벽하게 부합한다. 이것은 다른 XML 테크놀로지와의 통합과 상호운용성에 부합한다는 것이다. 셋째, SVG 역시 CSS를 사용하면 다른 HTML과 똑같은 방식으로 스타일을 바꿀 수 있다. 이는 표현으로부터 구조/내용을 분리한다는 것이다. 넷째, SVG는 SMIL(Synchronized Multimedia Integration Language) 애니메이션 구문을 사용한다. 이 구문은 독립적 SVG 파일에 유용할 뿐만 아니라 SVG 문서가 멀티미디어 SMIL 표현에 내장될 수 있도록 해주어 오디오 컴포넌트와 비디오 컴포넌트가 벡터 그래픽 애니메이션과 겹치도록 해준다. 마지막으로, JPEG 그리고 PNG 같은 래스터 이미지들을 위한 표준, XSLT를 이용한 SVG의 상호 변환, SMIL을 사용한 애니메이션, 칼라 프로파일을 사용한 색상 관리, 그리고 유니코드를 사용하는 국제적인 텍스트. Postscript, PDF, Java2D, Mac OS X Quartz에 사용되어 산업계가 인정한 그래픽 원리 위에 구축된 모든 것을 포함할 수 있다.

2. 시스템 설계

시스템의 전체 구성은 그림 1과 같다. 본 편집 시스템은 SVG 그래픽 객체를 처리하

는 객체편집기와 각각의 객체에 따른 SVG DOM(Document Object Model)[3]으로 생성된 SVG 원문을 처리하는 문서편집기로 구성된다.

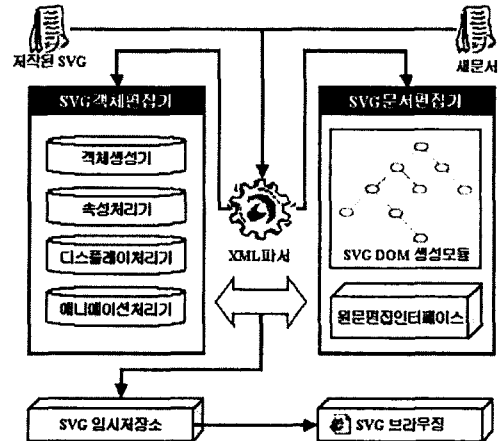


그림 1. 전체 시스템 구성도

2.1 SVG 객체 편집기

객체 편집기는 SVG 객체들의 편집과 디스플레이를 위해 각각을 객체생성기, 애니메이션처리기, 속성처리기, 디스플레이처리기 클래스로 구분한다. 각각의 클래스는 SVG 표현의 골격을 정의하는 프로토타입 클래스로부터의 상속을 받도록 설계하여 수정과 확장을 용이하게 한다. 그림 2는 객체 편집기의 클래스 구성도이다.

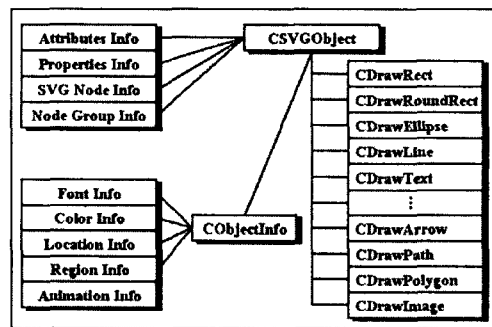


그림 2. 객체편집기의 클래스 구성도

2.1.1 객체 생성기 설계

객체 생성기는 사용자로부터 입력받은 SVG 문서를 XML파서를 통해 메모리 상에 DOM 다큐먼트로 상주하게 되고 SVG DOM 노드를 탐색하면서 각각의 엘리먼트 별로 해당 그래픽 표현을 위해 메모리에 COBList 클래스를 상속받은 m_List로 노드를 분류하여 각 객체를 저장한다.

그림 3은 객체생성기의 구성도를 보여준다. DOM 노드를 탐색하는 도중에 객체에 속성과 프로퍼티가 존재할 경우 렌더링을 위해 속성처리기와 애니메이션 처리기로 데이터를 보내고, 각각의 객체들이 사용자 영역에 렌더링 되기 위해 디스플레이 처리기로 객체들의 정보를 전달한다.

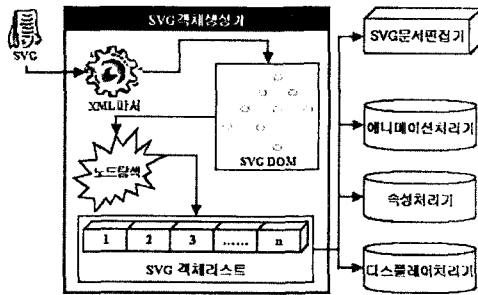


그림 3. 객체생성기의 구성도

2.1.2 속성 처리기 설계

객체 생성기로부터 전달받은 각각의 노드에 대한 속성 및 프로퍼티 정보들은 속성 처리기에 의해 CList 클래스를 상속받은 m_attList로 각 객체의 포인터와 속성 및 프로퍼티 정보들을 구조체 형식으로 저장한다. 저장된 정보들은 사용자 편집에 의해 특정 객체의 정보가 수정될 때마다 SVG DOM 노드를 변경한다. 그림 4는 속성처리기의 구성도를 보여준다.

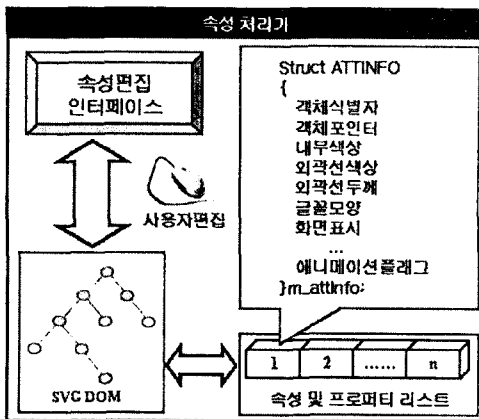


그림 4. 속성처리기의 구성도

속성 및 프로퍼티 리스트에 저장된 정보들을 토대로 CListCtrl을 상속받은 속성편집 인터페이스를 통해 사용자에게 해당 객체의 정보를 수정할 수 있도록 해준다. 사용자 편집은 ICC(International Color Consortium)[4]칼라 편집

기, 유니코드를 포함한 글꼴편집기 등의 인터페이스를 통해 사용자에게 동적으로 제공되며, 수정된 정보들은 곧바로 디스플레이 처리기로 전달되어 사용자 영역에 렌더링 될 수 있게 한다.

2.1.3 디스플레이 처리기 설계

디스플레이 처리기는 CSVGRender라는 이름의 기본 클래스에 정의되며, 사용자 뷰영역을 설정하고 각각의 SVG DOM 노드들이 갖는 여러 디스플레이 정보들을 수집하여 해당 노드가 표현하고자 하는 내용을 사용자 뷰영역에 보여준다. 그림 5는 디스플레이 처리기의 처리과정을 보여준다.

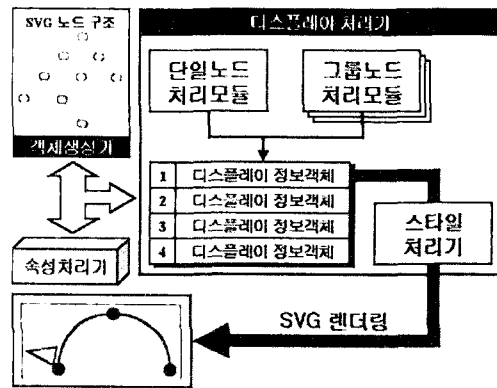


그림 5. 디스플레이 처리기의 처리과정

객체 생성기와 속성 처리기로부터 전달받거나 수정된 객체의 정보들은 단일노드 표현인지, 그룹노드 표현인지를 인지한 후 스타일 처리기를 통하여 사용자 뷰영역에 전달된다. 스타일 처리기는 CObjectInfo클래스에서 상속된 확장클래스에 정의되며, SVG문서 내에 존재하는 캔버스의 크기와 각 노드별 속성 및 프로퍼티 정보를 토대로 화면의 실제 캔버스 상에 SVG 그래픽을 표현한다.

캔버스에 객체를 그릴 때, 실제 뷰영역의 좌표는 논리적인 캔버스의 좌표로 변경한 후 해당 객체가 갖는 위치 설정을 변경하여 SVG DOM에 저장한 다음 그려주고, 렌더링 속도의 향상과 매끄러운 렌더링을 위해서 그려질 객체의 정보를 CMemDC 클래스에 저장하여 미리 메모리 상에 가상으로 그려준 후 화면이 갱신될 때 한꺼번에 그려지게 되며, 사용자로 인해 정보가 수정된 객체는 사용자 뷰영역에서 해당 객체의 위치만을 갱신시켜준다.

2.1.4 애니메이션 처리기

애니메이션 처리기는 SVG의 animate 엘리먼트에 기술된 시간, 속성, 반복횟수, 타입 등의 정보를 토대로 애니메이션 사용자 인터페이스를 통해 처리된다. 애니메이션 처리기는 CObjectInfo 클래스

스를 상속받는 CSVGAnimate 클래스에 정의된다. 속성 처리기에 저장된 애니메이션 플래그를 통해 애니메이션 여부를 판단하고, 이 속성의 값으로 TRUE를 가질 경우에는 애니메이션 처리기를 활성화한다. 사용자에게 의해 설정된 타임라인을 통해 해당 시간만큼 객체를 이동시키거나 확대하는 등의 변형을 주고, 변형된 정보는 다시 SVG DOM의 해당 노드 정보를 수정한다.

2.2 SVG 문서편집기

SVG 문서편집기는 사용자로부터 입력받은 SVG문서나 새로이 작성된 SVG문서의 DOM 다큐먼트를 편집하기 위해 일반적인 텍스트 편집기 형식의 인터페이스로 제공되며, CSVGSourceView 클래스에 정의된다. 그림 6은 SVG 문서편집기의 처리과정을 보여준다.

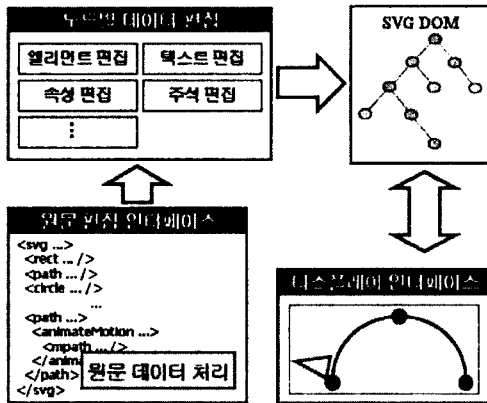


그림 6. 문서편집기의 처리과정

이 클래스에서는 사용자 입력으로 생성된 SVG DOM 객체 또는, 새문서 작성으로 생성된 DOM 객체를 문자열로 읽어들이어 텍스트 편집기에 보여주고 객체편집기에서 수정된 내용에 정밀한 수정이 필요할 때 이 문자열을 편집함으로써, 좀더 정확한 표현을 할 수 있게 된다. 이 클래스를 통해 사용자는 현재 문서의 내용에 대한 가독성을 높일 수 있고, 원하는 부분의 문자열을 수정함으로써, SVG DOM 객체의 정보를 수정할 수 있게 된다. 수정된 문자열은 다시 DOM 객체로 재정의되며, 이를 통해 객체편집기에 정보를 전달함으로써 SVG 객체의 표현을 수정하게 한다.

2.3 SVG 편집시스템 모듈 시험

본 시스템은 Windows 2000 Server 환경에서 Microsoft Visual C++ 6.0으로 설계하였고, 사용된 파서는 Microsoft MSXML4.0을 사용하였다.

그림 7은 SVG 편집시스템의 프로토타입에 대한 검증을 위해 모듈 시험용으로 구현된 객체편집 테스트용 버전이다.

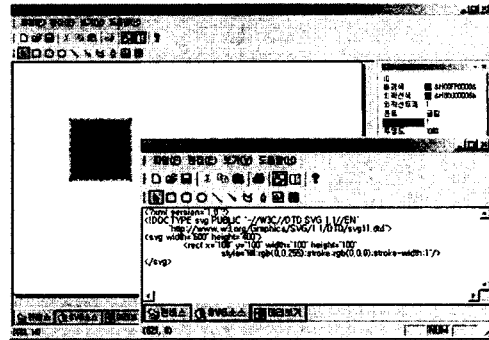


그림 7. SVG 편집시스템 모듈 시험

III. 결 론

기존의 웹에서 사용된 표현중심의 그래픽처리에는 비구조적인 방법에 따른 많은 문제점을 가지고 있었다. 이러한 문제점이 대두되기 시작하면서 W3C에서는 구조적으로 벡터그래픽 처리를 하기 위한 SVG를 제정하였고, 이에 따른 여러 관련 어플리케이션이 요구되었다. 현재 SVG 1.0 버전을 기준으로 나온 몇몇 어플리케이션[5]들이 존재하지만 이들이 처리하고 있는 범위는 SVG의 일부분에 국한되어 있는 실정이다. 이러한 문제점을 보완하고자 본 논문에서는 SVG 1.1버전을 기반으로 하여 벡터그래픽을 편집하고 표현하기 위한 SVG 편집시스템의 프로토타입을 설계하였다. 본 연구에서는 SVG 그래픽 표현을 위한 요건들을 제시하였고, SVG의 구조가 표현하는 그래픽 데이터 처리에 관한 방법을 제안하였다.

본 연구를 토대로 시스템 구현이 완전하게 이루어지면 지리정보시스템이나 모바일서비스 환경 등에서 XML기반의 그래픽을 표현하는 효율적인 처리 등에 유용하게 사용할 수 있을 것으로 사료된다.

참고문헌

- [1] W3C's eXtensible Markup Language(XML) Version 1.0, <http://www.w3.org/TR/REC-xml>, Feb. 10, 1998
- [2] W3C's SVG 1.1 Spacification Homepage, <http://www.w3.org/TR/SVG11>
- [3] W3C's Document Object Model, <http://www.w3.org/DOM>
- [4] International Color Consortium Homepage, <http://www.color.org>
- [5] W3C's SVG Implementations Homepage, <http://www.w3.org/Graphics/SVG/SVG-Implemmentations.htm8>