
벡터 그래픽 기반의 GML 문서 편집 및 변환 시스템

김창수* · 염성근** · 정희경***

GML Document Editing and Translation System based on Vector Graphic

Chang-Su Kim* · Sung-Keun Yeom · Hoe-Kyung Jung***

요 약

정보기술의 발달과 인터넷의 보편화에 따라 지리정보의 활용분야가 다양해지고 지리 정보를 효율적으로 관리하기 위해 다양한 지리정보 시스템(GIS : Geographic Information System)이 구축되었다. 하지만 다양한 형태의 지리정보 데이터는 서로 표준화 되지 않아 다양한 형태의 그래픽 저작도구에 의존하고 있다. 이에 OGC(Open Geospatial Consortium)는 상호운용 가능한 표준화된 지리정보 데이터를 기술하는 GML(Geography Markup Language)을 제안하였고 W3C에서는 벡터 기반의 SVG(Scalable Vector Graphics)를 제안하였다.

본 논문에서는 지리정보 데이터처리를 위한 XML 기반의 GML 데이터를 벡터 그래픽 객체로 생성하고 그래픽 객체들을 통해 GML 문서를 생성하는 코드 변환기를 정의하여 벡터 그래픽을 XML 기반의 논리구조로 변환하는 GML 문서 편집 및 변환 시스템을 설계 및 구현하였다.

ABSTRACT

According to development of Information Technology and generalization of internet, practical use field of geography information became various. Therefor various Geographic Information System (GIS) constructed to manage geography information efficiently. However, geography information data of various form is depending on graphic authorizing tool of various form being not normalized each other. So, OGC (Open Geospatial Consortium) proposed GML (Geography Markup Language) that describe normalized geography information data that can apply mutually and W3C proposed SVG (Scalable Vector Graphics) of vector base.

In this paper creates GML data of XML base for geography information data processing to vector graphic object. and design and implementation of GML document editing and translation system that define code converter that create GML document through created graphic objects and change vector graphic to logic structure of XML base.

키워드

지리정보, GIS, SVG, XML, GML, Translation

* 청운대학교 인터넷학과
** 배재대학교 컴퓨터공학과
*** 배재대학교 컴퓨터공학과(교신저자)

I. 서 론

정보기술의 발달과 인터넷의 보편화에 따라 지리정보의 활용분야가 다양해지고 지리 정보를 효율적으로 관리하기 위해 다양한 지리정보 시스템(GIS : Geographic Information System)이 구축되었다. 하지만 다양한 형태의 지리정보 데이터는 서로 표준화 되지 않아 다양한 형태의 그래픽 저작도구에 의존하고 있다. 이에 OGC(Open Geospatial Consortium)는 분산 환경에서 상호운용성을 지원하기 위해서 OpenGIS(Open Geodata Interoperability Specification)를 제시하였으나, 웹 환경을 고려하지 않았다는 단점이 있다. OpenGIS를 웹 환경에서 검증하고 활용하기 위해 상호운용 가능한 표준화된 지리정보 데이터를 기술하는 GML(Geography Markup Language)을 제안하였다.

W3C에서는 구조적인 방식의 콘텐츠를 생성하고, 관리하고 인터넷 상에서 벡터 그래픽 표현의 효율적인 처리와 저장 및 공유를 위한 제안W3C에서 1998년 제정되었다. SVG는 XML 문서이므로 XML 문서의 구조를 그대로 따른다. SVG 문서는 논리적 구조와 물리적 구조를 가지고 있다. 물리적으로 SVG 문서는 개체(Entity)라는 요소들로 이루어지며 개체는 다른 개체들을 참조해 그것들을 문서 안에 포함시킬 수 있다. 논리적으로 SVG 문서는 선언(Declaration), 엘리먼트(Element), 주석(Comment), 문자 참조(Character Reference), 그리고 처리 명령(Processing Instruction)으로 이루어지며 명시된 마크업에 의해 문서 안에 표시된다.

이에 본 논문에서는 지리정보 데이터처리를 위한 XML(eXtensible Markup Language) 기반의 GML 데이터를 벡터 그래픽 객체로 생성하고 그래픽 객체들을 통해 GML 문서를 생성하는 코드 변환기를 정의하여 벡터 그래픽을 XML 기반의 논리구조로 변환하는 GML 문서 편집 및 변환 시스템을 설계 및 구현하였다.

II. 관련연구

2.1 GML

OGC에서는 다양한 분야의 지리 정보를 손쉽게 상호 운용하기 위해 이질적인 환경의 지리 정보를 XML로 인코딩하여 다양한 분야에 사용 가능하도록 하는 GML 명세를 제시하였다. OGC에서는 1999년에 GML1.0을 발표한 이후, 2007년에 GML3.2.1을 발표하였다[2].

GML 명세에서는 OGC와 ISO/TC 211의 다양한 표준을 지원하기 위한 약 30여개의 GML 스키마들을 기술하고 있으며 응용 분야에서 사용할 수 있도록 다양한 객체도 정의하고 있다. 그림 1은 GML 클래스 구조를 보여주고 있다.

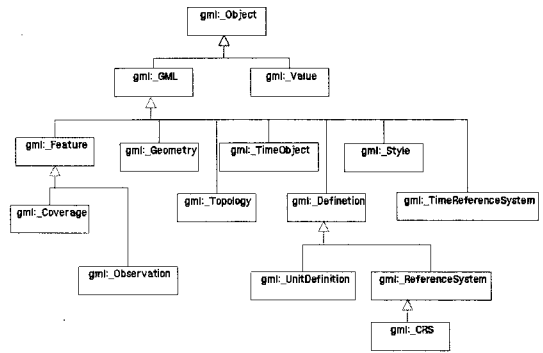


그림 1. GML 클래스 구조
Fig. 1 The architecture of GML class

2.2 SVG

SVG는 구조적인 방식의 콘텐츠를 생성하고, 관리하고 인터넷 상에서 벡터 그래픽 표현의 효율적인 처리와 저장 및 공유를 위한 제안W3C에서 1998년 제정되었다. SVG는 XML 문서이므로 XML 문서의 구조를 그대로 따른다. SVG 문서는 논리적 구조와 물리적 구조를 가지고 있다. 물리적으로 SVG 문서는 개체(Entity)라는 요소들로 이루어지며 개체는 다른 개체들을 참조해 그것들을 문서 안에 포함시킬 수 있다. 논리적으로 SVG 문서는 선언(Declaration), 엘리먼트(Element), 주석(Comment), 문자 참조(Character Reference), 그리고 처리 명령(Processing Instruction)으로 이루어지며 명시된 마크업에 의해 문서 안에 표시된다.

SVG 구성요소는 크게 4개의 구성요소가 있다. 모든 SVG 문서나 이미지에는 최상위 엘리먼트 <svg>를 포함하는 SVG 엘리먼트, 사용자의 뷰포트 영역에 사각형, 원, 선등과 같은 그래픽 객체들을 표현하기 위한 그래픽 엘리먼트, 여러 그래픽 엘리먼트들을 그룹화 하는 컨테이너 엘리먼트, 참조, 부연 설명이나 상태 처리 등에 사용되는 기타 엘리먼트들로 구성된다.

III. 시스템 설계

3.1 SVG/GML 편집기

GML과 벡터 그래픽 기반의 GML 문서 편집 및 변환 시스템은 그래픽 요소를 화면에 출력하기 위한 디스플레이와 각 그래픽 요소를 생성하고 관리하기 위한 그래픽

픽요소생성 및 관리, 편집한 내용을 저장하거나 외부 파일을 읽어오기 위한 입출력, 편집된 내용을 SVG 문서 객체로 반영하기 위한 DOM처리기와 벡터 그래픽 객체를 통해 GML 문서를 생성하는 변환기로 구성된다.

SVG/GML 문서편집기는 SVG/GML 편집기 객체가 실행되고 디스플레이의 제어관리부 객체를 통해 편집기의 기본 메뉴 및 이벤트를 처리한다. 제어관리부에서 로드하는 모듈을 통해 편집기를 구성하는 각 요소를 제어할 수 있는데, 본 시스템에서는 제어관리부에 각 모듈 목록을 저장하고, 저장된 목록을 파싱하여 해당 모듈을 로드하는 과정으로 편집기의 기본적인 구성을 실행하도록 하였다. 전체 시스템 구성은 그림 2와 같다.

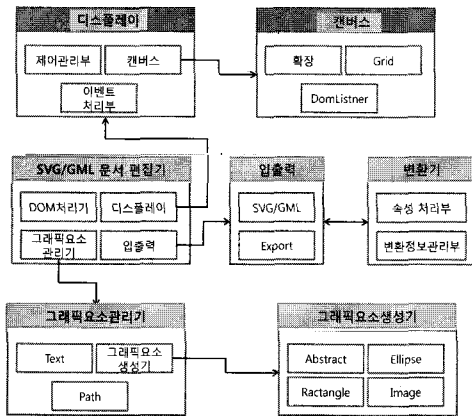


그림 2. 전체 시스템 구성
Fig. 2 The Diagram of System

3.1.1 DOM 처리기

SVG 캔버스에서 표현되는 SVG 객체들에 대한 사용자의 이벤트에 대한 SVG DOM 문서객체에 대한 수정이 이뤄지는 부분이다. 사용자가 특정 객체에 대해 정의할 수 있는 이벤트에는 정렬, 크기조절, 중간 맞춤, 순서, 회전 등이 있다. 이러한 이벤트가 발생했을 때, 해당 변경 사항에 대한 처리를 담당하는 부분이 DOM 처리기이다.

기본적으로 DOM 처리기는 XML DOM을 사용하며, 편집기 객체 내부에 정의된 SVG XML문서 객체의 내용을 이벤트 발생 시점에 즉각 처리하여 업데이트하고, 업데이트된 내용은 편집기의 화면에 반영된다. 따라서

DOM 처리기는 SVG 캔버스에서 처리되는 모든 이벤트 처리부와 연동되어 정보를 주고받을 수 있도록 설계되어 있다.

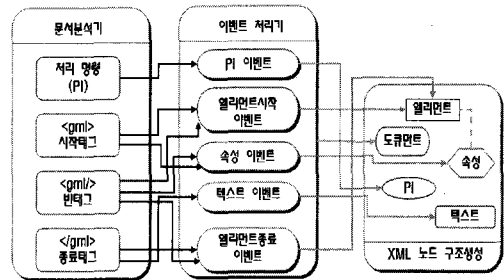


그림 3. DOM 처리기에서 노드생성
Fig. 3 Node Creation in DOM Processor

그림 3은 DOM 처리기를 통해 XML 문서가 파싱되어 XML 노드구조가 생성되는 모습을 보여준다. XML 파서 모듈은 크게 XML 문서의 여러 요소들을 DOM 형태로 정의하기 위한 XML 노드 클래스와 이를 관리하고 처리하기 위한 인터페이스를 제공하는 클래스로 구분된다. XML 노드를 구조화하기 위해 SAX 같은 이벤트기반 방식을 통해 파싱 단계를 거친다.

3.2 디스플레이

편집기에서 그래픽 요소를 화면에 출력하기 위한 부분으로 주요 구성으로는 SVG 캔버스, 이벤트 처리부, 제어관리부가 있다.

3.2.1 캔버스

SVG 파일을 화면에 표현하기 위한 처리를 담당한다. SVG 캔버스의 역할은 크게 SVG 객체를 표현하기 위한 것으로 캔버스 크기 조절, grid 관리, 확대-축소(Zoom)와 같은 캔버스 설정 및 SVG 객체를 그리고 편집할 수 있는 인터페이스를 제공한다. 이때 제어관리부를 사용해서 세부적인 문서 저장이 가능하다.

캔버스 영역은 GRID, BOTTOM, SELECTION, DRAW, TOP의 5개 계층(Layer)로 구성된다. 캔버스, 즉 편집 창을 새로 생성하면 5개 계층이 자동으로 생성된다. 이 위에 사용자 이벤트에 따라 SVG 객체를 표현함과 동시에 해당 객체에 대한 정보를 DOM처리기와 공유하여 SVG파일을 생성할 수 있도록 한다.

디스플레이에서 메모리 리스트에 저장된 그래픽 객체들을 객체를 이용하여 사용자 뷰포트에 렌더링 시키기 위한 처리 흐름을 그림 4에 나타내고 있다.

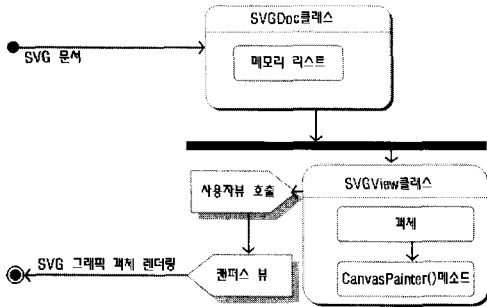


그림 4. 디스플레이 처리기의 처리 흐름도
Fig. 4 The process flow of Display handler

사용자로부터 입력된 SVG 문서는 SVGDoc 클래스의 메모리 리스트에 그래픽 객체 형태로 저장되고 저장된 객체들은 렌더링을 위해 모든 사용자 뷰를 갱신하는 함수를 호출한다. 이 함수에 의해 SVGView 클래스의 멤버인 CanvasPainter() 함수가 자동으로 호출되어 사용자 뷰포트에 SVG 그래픽 객체를 메모리 리스트에 저장된 순서대로 그려준다.

3.2.2 이벤트처리부

이벤트처리부는 사용자가 SVG 객체를 선택했을 때 발생하는 이벤트 처리부이다. 특정 객체를 선택하면 객체에 대한 크기 조절 모드로 전환되고, 다시 클릭 이벤트를 주면 회전 모드로 전환된다. 이벤트처리부는 단일 객체 혹은 다중 객체 선택 모드를 분류해서 처리하며, 각 객체에 주어지는 이벤트를 받기 위한 SelectionListener를 포함한다.

3.2.3 제어관리부

제어관리부는 편집기에서 사용하는 다양한 제어부를 관리하는 역할을 한다. 그 중 제어관리부는 SVG 캔버스에 위치한 SVG객체의 속성 정보를 제어하거나 속성 값을 구하는 등의 SVG에 대한 처리와 실제 svg파일을 로딩 했을 때, 파일에 저장된 svg 데이터를 핸들링 하는 역할을 한다. 또한 SVG 객체를 처리하는 여러 스레드를 관리한다. 그 외 SVG 엘리먼트, 리소스에 대한 제어를 제

어관리부에 의해 관리된다.

3.3 입출력

본 시스템에서 지원하는 파일을 읽고 쓰는 기능을 제공한다. 이미지 타입의 JPEG, PNG, BMP, PDF 로 export 할 수 있다. 또한 변환부에서 GML 문서와 SVG 문서를 변환될 결과를 저장할 수 있다.

3.4 변환기

변환기는 GML 문서와 SVG 문서간의 변환을 위해 각 문서의 스키마 문서 정보를 통해 매핑 정보를 정의한다. 문서의 변환을 위해 원하는 형식의 정보를 변환할 수 있도록 변환에 필요한 기본적인 정보들과 다양한 변환 방법을 제공 하게 된다.

변환기는 크게 각 문서의 노드들의 정보를 표현하기 위한 속성처리부와 변환 정보를 관리하는 변환 정보부로 크게 구성된다.

3.4.1 속성처리부

변환 시에 GML스키마에 정의된 형식이나 속성의 특성을 참조해야 한다. 변환될 GML문서의 노드가 최상위 엘리먼트인지 여부와 속성 여부를 구분하고 노드의 위치 정보를 XPath 구문으로 변환하여 구문 검색기를 통해 해당 위치의 노드 객체를 추출한다. 이 객체를 속성 추출기를 통해 해당 노드에 정보를 제공한다. 이 정보는 GML 문서 노드의 변환 처리에 반영된다. 속성 처리부는 그림 5와 같다.

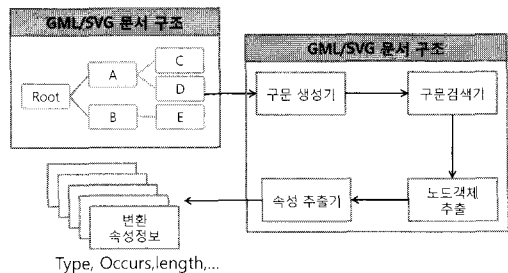


그림 5. 속성 처리부
Fig. 5 Property Process Module

3.4.2 변환정보부

GML 스키마와 SVG 문서 간의 변환 관계를 정의 및

변환 정보를 관리한다. 기본적인 변환 형태는 변환을 위한 노드의 데이터 타입과 같은 노드 정보를 비교하여 유효한 변환 규칙인지를 판단한다. 유효한 변환 규칙이면 변환 정보부에서 변환 정보 구조에 추가하여 변환 정보 객체로 관리한다. 그림 6은 변환 정보의 흐름도이다.

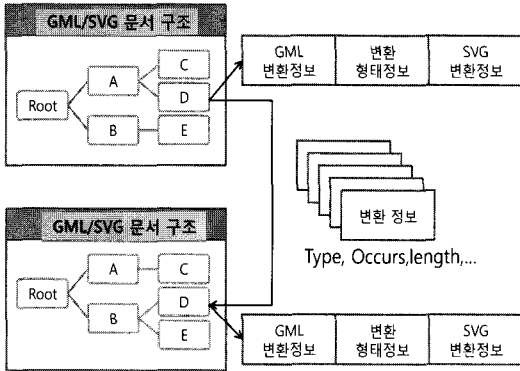


그림 6. 변환 정보의 흐름도
Fig.6 Flowchart of Translation Information

3.5 그래픽 요소 관리 및 생성기

Canvas에 표현할 수 있는 객체에 대한 정보를 구현한 부분이다. 본 편집기에서는 기본 도형과 경로(path), 텍스트를 지원하며, 기본도형은 **Ellipse, Rectangle, Image**, 등으로 구성된다. 각 도형은 크게 사용자의 마우스이벤트를 처리하는 부분과 해당 이벤트를 처리하는 처리부로 나뉘는데, 주요 이벤트로는 선택, 크기 변환, 페인트, 회전 등이 있다.

IV. 시스템 구현

4.1 GML 문서 편집 순서도

GML과 벡터그래픽 기반의 GML 문서 편집 및 변환 시스템에서 사용자가 특정 이벤트를 요청했을 때 처리되는 과정의 시퀀스 다이어그램을 그림 7에 나타냈다.

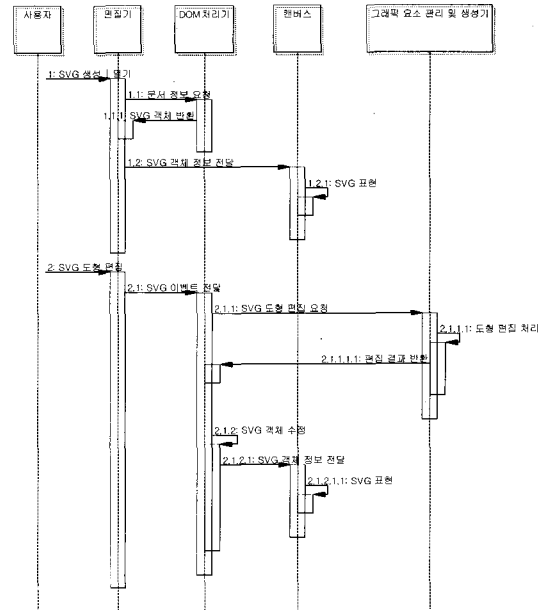


그림 7. GML 문서 편집 및 변환기 순서도
Fig. 7 Sequence Diagram of GML Document Editing and Translation System

이벤트는 SVG 문서 생성 및 열기와 SVG도형을 편집하는 두 가지로 나타났다.

먼저 사용자가 처음 SVG문서를 생성하거나, 기존에 저장한 문서를 열어서 재편집을 원할 경우이다. 사용자로부터 발생한 이벤트에 따라 편집기는 해당 SVG문서의 정보를 DOM처리부에 요청한다. DOM처리부는 요청된 문서를 DOM파서를 사용해서 파싱 한 후 SVG문서의 정보에 해당하는 DOM객체정보를 편집기에 전달한다. 편집기는 전달받은 DOM객체정보를 캔버스에 보내 해당 문서에 내용을 사용자가 확인할 수 있도록 화면에 표현해 준다.

SVG 도형편집 이벤트는 사용자 발생시킨 이벤트를 편집기가 받아서 DOM처리기로 전달한다. DOM 처리기는 이벤트를 분석하여 해당 처리를 위해 그래픽요소관리기로 전달한다. 그래픽요소관리기는 특정 도형에 대한 처리를 담당하는데, DOM처리기에서 지정한 도형을 생성하거나 편집하는 등의 처리를 한다. 이렇게 처리된 정보는 다시 DOM처리기로 전달되고, 전체 SVG문서 객체를 수정하는 과정을 거쳐, 캔버스를 통해 화면에 표현된다.

4.2 GML 문서 편집 시스템

그림 8은 시스템의 화면 구성을 보여준다. 전체 화면은 일반적인 그래픽 툴의 형태로 이루어지며, 자식(Child) 윈도우의 중앙에 GML 캔버스를 위치시킴으로써 저작되는 그래픽 객체들을 한눈에 볼 수 있도록 인터페이스를 사용자 중심으로 지향하였다. 특정 객체에 애니메이션이 추가될 경우 화면의 하단에 위치한 애니메이션 컨트롤에 등록되어 사용자가 쉽게 GML 그래픽 객체에 애니메이션을 추가하도록 구현하였다.

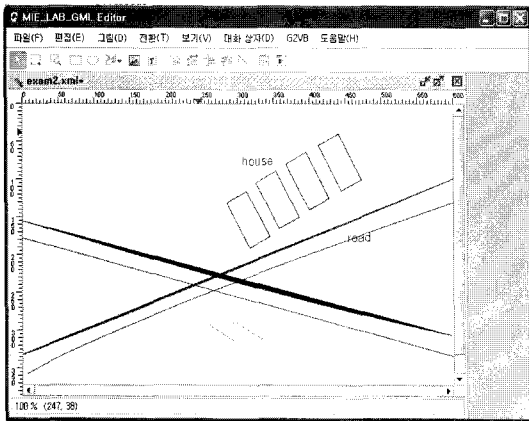


그림 8. GML 문서 저작 시스템의 화면 구성
Fig. 8 The process flow of Display handler

GML 속성 창과 리소스 창은 기존의 다른 시스템과는 달리 지지부진한 윈도우들을 줄이기 위해 팝업(Popup) 다이얼로그로 구현하여 사용자가 확인하고자 할 때만 보일 수 있도록 하였다.

V. 고찰 및 결론

OpenGIS를 웹 환경에서 활용하기 위해서는 지리정보를 표현하기 위한 표준으로 OGC는 XML에 기반한 GML 제안하였다. 사용자들이 웹 환경에서 지리정보를 쉽게 이용하기 위해서는 GML 문서를 저작할 수 있는 GML 문서 편집기가 필요하다.

이에 본 논문에서는 GML 문서를 쉽게 분석, 생성, 디스플레이, 편집할 수 있고, 또한 SVG 문서 변환하여 저장할 수 있는 GML 문서 편집 및 변환 시스템을 설계 및 구현하였다.

본 논문에서 구현한 시스템은 GML의 렌더링 처리에 대한 구성을 기반으로 구현함으로써 표준화에 따른 처리 시스템으로써 변환에 능동적으로 대처 가능하다. 또한 시스템을 모듈화 하여 부분적인 수정 및 대체와 이식이 가능하게 하여 사용자가 쉽게 GML 문서를 저작할 수 있다.

향후 연구 과제로는 본 논문에서 제시한 GML 문서 작성 시스템의 기능을 향상시키기 위해 사용자의 요구를 다양하게 반영한 사용자 인터페이스의 개발이 필요하고, 다양한 지리 정보를 표현하고 있는 여러 GML 문서를 간편하게 하나의 문서로 통합할 수 있는 방안과 GML 문서를 필요에 따라 다양한 XML 문서로 변환해 줄 수 있는 방안에 대한 연구가 필요할 것이다.

참고문헌

- [1] OpenGIS Consortium. Inc OpenGIS Simple Features Specification for OLE/COM Revision 1.1 1999
- [2] OpenGIS Consortium, Inc, Geography Markup Language(GML) Implementation Specification 3.2.1 2007
- [3] W3C, Scalable Vector Graphic(SVG) Version 1.1 <http://www.w3.org/TR/SVG11>, Jan. 14 2003
- [4] W3C, Cascading Style Sheets, level 2, <http://www.w3.org/TR/TEC-CSS2/>.May 12, 1 998
- [5] Java SimpleGMLReader package, <http://www.vividsolutions.com/JUMP/javadoc/com/vividsolutions/jump/util/io/SimpleGMLReader.htm>
- [6] W3C, SVG Transformation Tool, <http://www.w3.org/2002/05/svg2stuff.html>
- [7] W3C, SVG Implementations, <http://www.w3.org/Graphics/SVG/SVGImplementations.html>

저자소개



김창수(Chang-Su Kim)

1996년 배재대학교 전자계산학과
(이학사)

1998년 배재대학교 전자계산학과
(이학석사)

2002년 배재대학교 컴퓨터공학과(공학박사)

2001년~2004년 배재대학교 IT교육센터 책임강사

2005년~현재 청운대학교 인터넷학과

※ 관심분야: 멀티미디어 문서정보처리, XML, Web Services, Semantic web,



염성근(Sung-Kun Yeom)

2008년 배재대학교
컴퓨터공학과(공학사)

2008년~현재 배재대학교
컴퓨터공학과(석사과정)

※ 관심분야: 멀티미디어 문서정보처리, XML, USN



정희경(Hoe-Kyung Jung)

1985년 광운대학교 컴퓨터공학과
(공학사)

1987년 광운대학교 컴퓨터공학과
(공학석사)

1993년 광운대학교 컴퓨터공학과(공학박사)

1994년~현재 배재대학교 컴퓨터공학과 교수

※ 관심분야: 멀티미디어 문서정보처리, XML, SVG,
Web Services, Semantic Web, MPEG-21, Ubiquitous
Computing, USN