

DML(Diagram Markup Language)

시스템의 설계 및 구현

최종명, 윤태희, 신경희, 유재우

승실대학교 컴퓨터학과

jmchoi@it.soongsil.ac.kr

The Design and Implementation of DML System

Choi Jongmyung, Yoon Taehee, Shin Kyounghee, Yoo Chaewoo

Dept. of Computing, Soongsil Univ.

요약

컴퓨팅 환경에서 GUI의 사용이 일반화되면서 다이어그램을 이용하는 시스템이 많아지고 있는데 비해, 다이어그램을 이용하는 시스템을 개발하기 위해서는 많은 노력과 시간을 필요로 한다. 이것은 다이어그램의 각 그래픽 컴포넌트를 프로그램으로 개발하여야 하는 문제 때문이다. 이러한 문제를 해결하기 위해서 본 논문에서는 그래픽 편집기를 이용해서 다이어그램 컴포넌트를 기술하고, 이것을 XML 문서로 변환해서, 최종적으로 프로그래밍 코드를 자동적으로 생성하는 DML 시스템을 소개한다. 이 시스템은 다이어그램 컴포넌트를 그래픽 편집기를 이용하여 기술하기 때문에 쉽게 새로운 컴포넌트를 생성할 수 있으며, 생성된 컴포넌트는 XML 문서로 표현되고, 코드 생성기를 통해 프로그래밍 코드로 변환되기 때문에 프로그래밍 언어에 무관한 장점을 가지고 있다.

1. 서론

컴퓨팅 환경에서 다이어그램은 많이 사용되고 있으며 점차 중요성도 높아지고 있다. 다이어그램은 사용자의 이해를 돕고, 그룹 성원들간의 의사 소통을 원활히 하는 장점이 있기 때문에 단순한 문서화에서부터 시스템 분석, 설계 및 프로그래밍 등의 분야에서 널리 사용된다. 또한, 시각 프로그래밍이나 새로운 모델링을 위한 소프트웨어를 개발할 때 다이어그램은 필수적인 요소이다. 이러한 소프트웨어 시스템에서 사용되는 다이어그램들은 응용분야에 맞는 규칙과 의미를 가지고 있으며, 이러한 요구사항들을 만족시키기 위해서 다이어그램은 단순한 GIF나 JPEG 등의 이미지가 아닌 프로그램으로 작성되어야 한다. 다이어그램을 프로그램으로 구현하는 것은 많은 노력을 필요로 하는 작업이기 때문에 자동화된 방법을 요구한다.

XML[1]은 구조화된 문서를 표현하기 위한 표준 마크업 언어로서 확장성, 이식성 등의 특징 때문에 문서의 표현, SOAP[2], XML Query[3] 등의 프로그래밍, CML[4], MathML[5], VML[6] 등의 특수한 목적으로 사용되고 있다.

DML의 구현에 XML을 이용하는 가장 큰 이유는 XML을 이용하는 경우에 이식성이 뛰어나고, 기존의 표준화된 도구와 API를 사용할 수 있다는 것이다. 이러한 장점을 활용하여 다이어그램을 XML로 표현하고 표현된

다이어그램은 XSLT[7] 프로세서나 SAX[8] 및 DOM[9] 파서를 이용해서 원하는 프로그래밍 언어로 변경될 수 있다. 본 논문에서는 다이어그램 컴포넌트를 XML 문서로 표현하고, 이것을 DOM과 XSLT를 이용해서 원하는 프로그래밍 코드로 변환하는 방법을 사용한다.

본 논문은 2장에서 관련 연구를 소개하고 3장에서는 다이어그램을 XML 문서로 표현하기 위한 XML 응용프로그램인 DML을 소개한다. 4장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 Vector Markup Language(VML)

VML은 웹브라우저에서 벡터 그래픽을 표현하기 위해 개발된 XML, 응용프로그램이다. VML은 기본적인 도형을 제공하며, 이 도형들을 그룹으로 묶어서 사용할 수 있는 방법을 제공한다. VML은 웹브라우저에서 벡터 그래픽을 보여주기 때문에 그래픽 정보는 VML을 이용하고, 그림의 위치는 CSS2[10] 네임스페이스를 사용한다. VML은 웹 환경에서 그림을 간단하게 표현할 수 있는 방법을 제공하지만, 일반 프로그래밍 환경에서 다이어그램을 표현할 수 있는 방법을 지원하지 않는다. 또한 VML의 벡터 그래픽은 단순히 그림을 표현하기 위한 방법을 제시할 뿐 그림의 의미나 규칙[12]을 기술할 수는 없다.

2.2 DiaGen

DiaGen[11]은 다이어그램 편집기 생성기이다. DiaGen은 다이어그램 언어를 통해 각 컴포넌트의 정의, 리듀싱 규칙, 문법 및 의미를 기술하여, free-hand 편집기나 구분 지향 편집기를 생성할 수 있는 방법을 제공한다. DiaGen은 Hyperedge 문법으로 표현된 다이어그램의 명세를 스캐닝 단계, 리듀싱 단계, 파싱 단계, 의미 분석 단계를 거쳐 다이어그램 편집기를 생성한다. DiaGen은 문법과 의미를 기술할 수 있는 기능을 제공하지만, 다이어그램 명세를 표현하는 Hyperedge 문법이 복잡하다는 단점이 있다.

2.3 Graph Exchange Language(GXL)

GXL[13]은 XML을 기반으로 그래프로 표현되는 정보의 교환을 위해 각 노드에 대한 정의와 노드간의 연결을 정의할 수 있도록 제안된 Software Exchange Format(SEF)이다. 그래프의 문법은 Document Type Definition(DTD)를 이용하여 나타내며 속성(Attributes)과 연관성(Connectivity)은 스키마를 사용하여 나타낸다. 하지만, GXL은 그래프의 노드(Node)와 에지(edge)를 통한 연결성이라는 정보자체의 연관성을 표현할 수 있지만 본 논문이 지향하는 다이어그램 편집기에 필요로 하는 새로운 다이어그램 컴포넌트의 표현에 대한 유연성, Constraints와 Layout에 대한 고려가 부족하다.

3. DML 시스템

DML 시스템은 그림 1과 같이 그래픽 편집기, DML 문서 생성기, 코드 생성기 등으로 구성되어 있다.

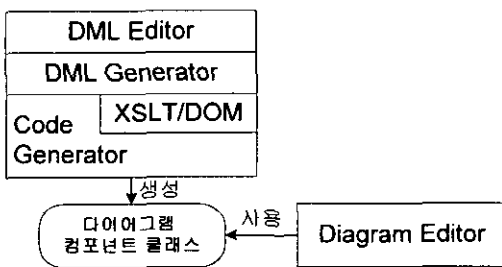


그림 1. DML 시스템 구조

사용자는 DML 그래픽 편집기를 이용하여 새롭게 정의하고자 하는 다이어그램 컴포넌트의 형태와 속성을 GUI를 이용하여 지정한다. 이렇게 DML 그래픽 편집기에서 기술된 형태와 속성은 DML 생성기를 통해서 DML 문서로 변환되고 이 DML 문서는 XSLT 처리기나 DOM 파서를 사용하는 코드 생성기에 의해 원하는 프로그래밍 언어의 클래스나 라이브러리로 변환된다. 생성된 다이어그램 컴포넌트 라이브러리는 응용프로그램에 맞는 다이어그램 편집기에서 사용된다. 그림 2는 DML 그래픽 편집기 모습이다.

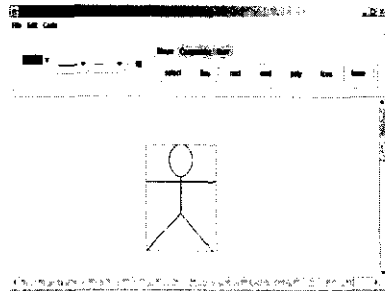


그림 2. DML 편집기

그림 3은 DML 그래픽 편집기를 통해 생성된 DML 문서의 예이다. 여기서 group은 하위 구조를 하나의 컴포넌트로 정의하고 shape은 이미 정의된 컴포넌트를 재정의하여 사용하는 최상위 엘리먼트이다. 각 컴포넌트의 속성은 표 1의 의미를 갖는다.

속 성	설 명
type	컴포넌트를 지정
x	컴포넌트의 초기 X 좌표값
y	컴포넌트의 초기 Y 좌표값
endX	1차원 컴포넌트의 최종 X 좌표값
endY	1차원 컴포넌트의 최종 Y 좌표값
width	2차원 컴포넌트들의 폭 값
height	2차원 컴포넌트들의 높이 값

표 1. 컴포넌트의 속성들

```

<?xml version="1.0" encoding="euc-kr" ?>
<group type="actor" targetNS="dml.ss.ssu.ac.kr">
  <shape type="oval" x="70" y="0" width="60" height="90"/>
  <shape type="line" x="0" y="100" endX="200" endY="100" />
  <shape type="line" x="100" y="90" endX="100" endY="200"/>
  <shape type="line" x="100" y="200" endX="0" endY="300"/>
  <shape type="line" x="100" y="200" endX="200" endY="300"/>
</group>
    
```

그림 3. 생성된 DML 내용

생성된 DML 내용은 코드 생성기에 의해 원하는 프로그래밍 언어로 변환된다. 예를 들어 원하는 코드가 자바 언어이면, 자바 언어를 위한 XSLT 스타일시트를 작성해서 자바 언어로 변환한다. 한번 작성된 XSLT는 다시 사용할 수 있기 때문에 편리하게 원하는 코드 형태로 변환할 수 있다. XSLT로 변환이 불가능하거나 복잡한 경우에는 DOM 파서를 이용해서 코드를 변환할 수 있다. 그림 4는 생성된 자바 코드의 일부분이다.

```
package dml.ss.ssu.ac.kr;
...
public class Actor implements Drawable {
    Drawable next, prev;
    Vector children = new Vector();
    ....
    public void paint(Graphics g) {
        int n = children.size();
        for(int i = 0 ; i < n ; i++){
            Drawable v = (Drawable)
                children.elementAt(i);
            v.paint(g);
        }
    }
    ....
}
```

그림 4. 생성된 자바 코드

그림 5는 위의 과정을 통해 새롭게 생성된 다이어그램 컴포넌트를 다른 다이어그램 편집기에서 사용하는 예이다.

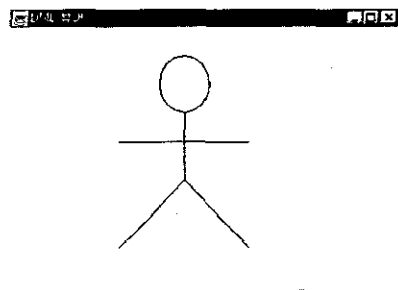


그림 5. 생성된 컴포넌트를 사용한 예

4. 결론 및 향후 연구방향

다이어그램은 컴퓨팅 환경의 발전으로 많은 분야에서 사용되고 있으며 그 중요성도 점차 증가하고 있지만 다이어그램을 이용하는 소프트웨어를 개발하기 위해서는 다이어그램 컴포넌트의 개발에 많은 노력과 시간을 소비해야 한다.

이러한 문제점을 해결하고자, 본 논문에서는 DML 그래픽 편집기에서 그래픽으로 기술한 다이어그램 컴포넌

트를 XML을 기반으로 한 응용프로그램인 DML 문서로 표현하고, DML 문서로 표현된 내용을 XSLT 프로세서나 DOM 파서를 이용해서 자동으로 해당 프로그램 소스를 생성하여 새로운 컴포넌트를 다이어그램 에디터에서 사용하는 방법을 소개하고 있다. 다이어그램 컴포넌트를 DML로 표현하는 것은 컴포넌트가 프로그래밍 언어에 종속되지 않아 프로그램간의 정보 교환이 용이하고 정보의 표현이 통일된다는 장점을 갖으며 개발 시간과 비용을 절감하는 장점을 갖는다.

추후로는 DML에 DiaGen과 같이 각 컴포넌트의 의미를 기술하기 위한 방법, 컴포넌트간에 문법적인 관계, Constraints와 Layout을 지정하는 방법 등을 연구할 예정이다.

5. 참고 문헌

- [1] Extensible Markup Language (XML) 1.0, <http://www.w3.org/TR/2000/REC-xml-20001006>
- [2] Simple Object Access Protocol (SOAP) 1.1, 2000. <http://www.w3.org/TR/SOAP/>
- [3] XML Query, <http://www.w3.org/XML/Query>
- [4] Chemical Markup Language, <http://www.xml-cml.org/>
- [5] Mathematical Markup Language, <http://www.w3.org/Math/>
- [6] Vector Markup Language, <http://www.w3.org/TR/NOTE-VML>, 1998
- [7] XSL Transformations, <http://www.w3.org/TR/xslt>
- [8] The Simple API for XML, <http://www.megginson.com/SAX/index.html>, 2000
- [9] Document Object Model, <http://www.w3.org/DOM/>
- [10] Cascading Style Sheets, <http://www.w3.org/Style/CSS/>
- [11] M. Minas, O. Koth. "Generating Diagram Editors with DiaGen". in *Proc. of the Int'l Workshop on Applications of Graph Transformation with Industrial Relevance*, pp 433-440, Sep., 1999.
- [12] G. Viehstaedt M. Minas, "Graphical representation and manipulation of complex structures based on a formal model", in *Proc. 7th International Conference on Advanced Information Systems Engineering (CAISE'95)*, LNCS 932, Springer-Verlag pub. pp. 243-254, Jun., 1995.
- [13] R. C. Holt, A. Winter, A. Schurr. "GXL: Towards a Standard Exchange Format". in *Proc. 7th Working Conference on Reverse Engineering*, available <http://www.uni-koblenz.de/~winter/publications.html>, 2000.