

화학분자식의 3D 표현을 위한 XML 소스 생성기의 설계 및 연구

김재호*, 박사준**

요약

화학분자식(H_2O , CO_2)의 그래픽 표현을 위한 여러 교육용 응용프로그램들이 상용화되어 있다. 그러나 이들 응용 프로그램들의 데이터 형식은 자체 형식을 사용하고 있어 다른 응용 프로그램에서 활용할 수 없다. 또한 개방형 형식인 XML을 생성하는 경우에도 2D 표현을 위한 것이 대부분이다. 본 논문은 화학분자식을 개방형 형식으로 저장하면서 3D 구조로도 표현할 수 있는 도구를 개발하는 데 목적이 있다. 이를 위해 본 논문에서는 화학분자식을 XML 소스로 변환하여주는 소스 생성기를 설계하고자 한다. 이렇게 생성된 XML 소스는 2D/3D 화학분자식 관련 응용 프로그램에 표준 소스로 제공될 수 있을 것으로 기대된다.

A Study and Design on XML Source Generator for 3D Molecular Representation

Jae-Ho Kim*, Sa-Joon Park**

ABSTRACT

There are educational applications for graphic representation of chemical molecular expression(H_2O , CO_2). But the data formats of these applications use a self format. Therefore other applications can't use these formats. Also although the applications generate XML as open format, are mostly for 2D representation. The purpose of this paper is a development the tool making a molecular expression to open format, representing 3D structure. For the purpose, we try to design a source generator transforming a molecular expression to XML sources. We hope to serve the standard molecular source for 2D/3D molecular application with these XML sources.

Key Words : XML, CML, Molecular Chemistry, 3D Modelling, Education

* 강릉원주대학교 정보기술공학과(✉kimjaeho@gwnu.ac.kr)

** 대구한의대 모바일컨텐츠학부

· 제1저자(First Author) : 김재호 · 교신저자(Correspondent Author) : 박사준

· 접수일(2010년 7월 6일), 수정일(1차 : 2010년 8월 27일), 게재확정일(2010년 9월 1일)

I.

현재 데이터의 구조를 나타내는 표준적인 방법으로는 세계 표준안인 XML 관련 기술이 정립되어 가고 있다. 이러한 XML 기술은 각 응용 분야의 데이터 구조를 표현하는 데에 적용되어가고 있으며, 화학분야에서도 화학분자식을 표현을 위한 CML(Chemical Markup Language) 2.0이 제안되어 있다[1][2].

XML 기반의 CML은 마크업 언어로써 화학분자식의 표준으로 자리 잡고 있다. 이러한 CML 소스를 기반으로 화학분자식의 그래픽 표현을 위한 많은 시도가 있는데, 그 중 대표적인 것이 SVG(Scalable Vector Graphics) 이다[3][4]. SVG는 XML 그래픽 문법으로 다른 XML 언어들과 결합하여 다양한 응용프로그램들이 개발되고 있다. 그러나 SVG 소스는 2D 위주의 표현이기 때문에 화학분자식의 특징적인 3D의 구조를 표현하는 데에는 한계가 있다.

CML 소스를 이용한 3D 그래픽 표현 응용 프로그램[4]들도 존재하는데, 이는 CML 소스를 VRML(Virtual Reality Markup Language)과 같은 가상현실을 표현하는 소스로 변환하여 해당 응용 프로그램에서 렌더링 하여 표시하는 것이다. 2D 혹은 3D 그래픽 표현들 중에서 어떠한 방식을 사용 하던지 간에 이들 도구들은 CML 형태의 소스를 자체의 소스로 변환한 후 표시하는 방식을 취하고 있다. 즉 CML로 소스가 제공되어야지 만이 처리가 가능하다. H₂O와 같이 텍스트로 표시된 화학분자식을 XML 기반의 마크업 언어로 표현할 수 있는 생성기의 필요성이 대두된다. 반드시 CML 규격대로는 아니더라도 최소한 XML 기반의 표현으로 되어있어야 데이터의 호환이 가능하게 된다.

또한 대부분의 다른 응용 프로그램들은 자체로 화학분자식 포맷을 가지고 있어서 호환성에 취약

한 점이 있다. 이는 인터넷을 이용한 개방형 환경에서의 단점을 가지게 된다. 따라서 이러한 응용 프로그램에 호환성을 제공하기 위해 XML 기반의 화학분자식 소스를 제공할 수 있는 XML 기반 생성기가 필요하다.

본 연구에서는 목록선택 방식으로 구성된 화학분자식(예를 들어 H₂O)을 CML 형태의 소스로 변환하여주는 소스생성기를 설계하고자 한다. 이것을 일명 CML-G(Generator) 명명한다. 이를 이용하여 XML 기반의 화학분자식을 위한 그래픽 표현 응용 프로그램에 표준 화학분자식 소스를 제공할 수 있게 된다.

본 연구에서의 CML-G는 원소주기율표 1번에서 20번까지로 제한하고 있으며, 추후 많은 원소를 추가할 예정이다. 또한 이 결과를 교육 분야에 활용하기 위해 한국교육학술정보원에서 제안한 디지털 교과서[6]용 ETBK(E-TextBook) 데이터 스키마에 부합될 수 있는 구조로 제안할 예정이다.

본 연구에서는 화학분자식의 3D 그래픽 표현을 위한 CML 형태의 마크업 언어를 설계하여, 이를 기반으로 목록선택 방식으로 입력된 화학분자식을 CML 형태의 소스로 변환하는 생성기 시스템을 제안하고자 한다.

II.

화학분자식을 그래픽 및 3D 으로 표현하는 응용 프로그램들을 화학분자식 표기방식을 중심으로 구분해 보면 다음과 같은 3가지 유형으로 나누어 볼 수 있다.

① CML 소스를 다른 XML 기반 마크업(예를 들어 SVG)으로 변환하여 2D/3D 그래픽으로 표현하는 방식.

② CML 소스를 응용 프로그램 자체 포맷으로

변환하여 2D/3D 그래픽으로 표현하는 방식

③ 스텐실 Drag & Drop 방식으로 자체 인터페이스를 통해 입력한 후 2D/3D 그래픽으로 표현하는 방식.

표 1은 상용화된 화학분자식 응용프로그램들의 데이터 형식을 비교해 놓은 것이다.

표 1. 응용프로그램 비교
Table 1. Comparison among applications

응용프로그램	데이터 형식
ACD/ChemSketch	자체포맷 (sk2, chm, s3d, mol, mop)
BKchem	자체포맷, XML 형식 odf, png, svg, cdml
Jmol	자체포맷(spt), VRML, X3D
HyperChem	자체포맷(hin)
JME	자체포맷(smiles, mdl)
BS Contact	wrl, X3D, dae, pmlxml
ZVON	SVG

이중에서 ①번 방식은 미리 작성된 CML 소스를 화학분자식으로 하여 SVG 포맷으로 변환하는 방식을 채택하고 있다. CML 소스를 활용한다는 표준성이 있다. 이러한 방식을 사용한 시스템으로는 BScontact, ZVON[7] 등이 있으며 SVG 플러그인 설치되어야 그래픽으로의 표현이 가능하다.

②번 방식은 CML 소스를 사용하지만 사용하는 파일 포맷은 응용 프로그램 자체의 것을 사용한다. 이러한 방식 사용하는 시스템으로 BSContact, BKchem[8] 등이 있으며, 주로 stand-alone 시스템이다.

③번 방식은 상용 응용 프로그램들이 채택하고 있는 방식으로서 화학분자식 표기방식은 H₂O와 같은 텍스트 방식 또는 원소기호를 목록선택 방식으로 배치하는 방식이다. 그러나 소스가 응용 프로그램 자체 포맷으로써, XML 기반의 CML 방식이 아니기 때문에 표준 및 호환성에는 문제를 가지고 있다. 이러한 방식을 채택하고 있는 응용 프로그램들은 BKChem[8], ACD/ ChemSketch[9], Jmol[10], JME[11] 등이 있으며, 주로 Standalone 시스템이다.

본 연구에서 구현된 CML-G는 표 1에 제시된 응용 프로그램들과는 달리 XML 형태의 개방형 파일 형식과 함께 3D 표현이 가능한 구조로 생성해 낸다.

III.

화학분자식을 3D로 표현하는 방법에서 가장 중요한 부분은 3D 표현 렌더링 엔진에 있다. 본 연구에서는 이러한 렌더링 방법 등에 대해서는 기존에 있는 렌더링 엔진을 활용을 한다고 가정한다.

본 연구의 주요 내용은 첫째, CML 형태의 마크업로 설계하여, 둘째, 입력된 화학분자식을 CML 소스로 변환한 후, 셋째, CML 형태의 소스가 필요한 방식에 활용하기 위한 CML 소스 자동생성기를 생성하는 것이다.

화학 분자식은 원자들이 상호 결합된 형태로 이루어져 있다. 분자식은 원소들의 결합을 통하여 다양한 형태로 확장이 가능하다. 따라서 분자식의 유연하고 확장 가능한 특성을 고려하여 XML을 기반으로 분자식을 기술하도록 CML 형태를 이용한다. 그러나 본 연구에서는 3D 표현에 필수적인 CML 요소들만 포함하여 구현하고 추후 연구를 통해 보강하고자 한다.

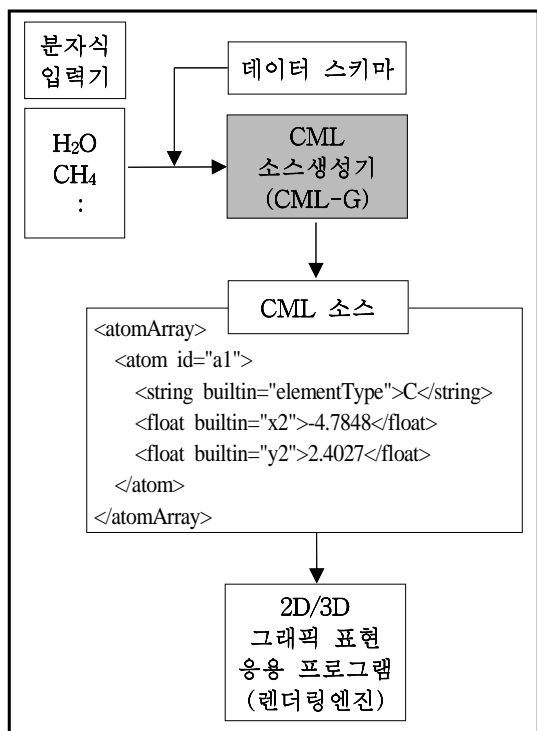


그림 1. 주요 연구내용

Fig 1. Main content of study

그림 1은 본 연구에서 수행되어질 내용들을 도식화하여 나타낸 것이다.

분자식입력기는 원자들을 이용하여 분자식을 생성하는 모듈로서 각 원자들을 목록선택 방식으로 연결한다. CML 소스 생성기 모듈은 조합된 분자 구조를 CML 형태의 소스로 생성해 내는 역할을 한다. 이렇게 해서 생성된 CML 소스는 렌더링 엔진에 전달되어 2D 혹은 3D 그래픽으로 표현된다.

텍스트 형태로 입력되거나, 목록선택 방식으로 입력된 화학분자식을 CML 소스로 변환하는 생성기를 설계하기 위해서는 우선 다음과 같이 원자 모델링 부분, 분자 모델링 부분으로 나누어서 마크업 언어를 설계할 필요가 있다.

다른 응용 프로그램과는 달리 CML-G를 통해서 는 CML 코드가 자동으로 생성되며, 이를 이용하

여 다른 응용 프로그램들이 활용할 수 있다.

3.1

주기율표상의 1에서 20번까지의 원소들을 대상으로 하여 모델링할 것이지만, 향후 확장해 나갈 계획이다. 원자 모델링에 대한 XML의 기본 구조는 다음과 같다. 다음의 XML 구조에서 <elementArray>는 원자의 종류(타입)를 정의하는 엘리먼트이며 elementType 속성은 원자의 종류를 의미한다. 이러한 방법으로 elementType 부분에 1에서 20번까지의 원소들에 대한 명칭을 표시하여 XML 구조로 전개해 나간다.

```
<molecular>
<elementArray>
  <element id="e0" elementType="H" x="0"
    y="0" z="0" />
</elementArray>
<bondArray>
</bondArray>
<propertyArray>
  <property dataType="scalar"
    libraryRef="molecular_weight" />
  <property dataType="color"
    libraryRef="molecular_color" />
</propertyArray>
</molecular>
```

3.2

분자는 원자들이 모여서 구성된 형태로서, 원자들 간의 여러 관계에 의해서 그 형태가 결정된다. 주요 엘리먼트들은 <elementArray>, <bondArray>, <propertyArray>이 있으며, 분자 모델링에 대한 XML의 전체적인 기본 구조는 다음과 같다.

분자 모델링에서의 <elementArray>는 원자들이 결합되었을 때 나타나는 결합각도에 대한 정보들이 추가되어 3D로 표현되는데 활용이 된다.

```

<molecular>
<elementArray>
  <element id="e1" elementType="H" x="1.62"
    y="-0.03" z="0.84" />
  <element id="e2" elementType="H" x="1.32"
    y="0.80" z="-0.68" />
  <element id="e3" elementType="O" x="-1.19"
    y="0.01" z="-0.90" />
</elementArray>
<bondArray>
  <bond elementRef1="e1" elementRef1="e2"
    order="1" />
  <bond elementRef1="e2" elementRef1="e3"
    order="1" />
  <bond elementRef1="e1" elementRef1="e3"
    order="1" />
</bondArray>
<propertyArray>
  <property libraryRef="mml:molecular_weight"
    title="Molecular Weight"
    <scaler dataType="xsd:double"
      libraryRef="mml" molecular_weight />
  </property>
</propertyArray>
</molecular>
    
```

<elementArray>는 원자의 종류(타입)를 정의하는 엘리먼트이며 H 2개, O 1개로 구성된 H₂O를 의미한다. 아울러 속성 x, y, z은 각 원소들간의 위치를 의미한다.

```

<bondArray>
  <bond elementRef1="e1" elementRef1="e2"
    order="1" />
  <bond elementRef1="e2" elementRef1="e3"
    order="1" />
  <bond elementRef1="e1" elementRef1="e3"
    order="1" />
</bondArray>
    
```

<propertyArray>는 원자들의 질량을 설정하는 엘리먼트이다.

```

<propertyArray>
  <property
    libraryRef="mml:molecular_weight"
    title="Molecular Weight"
    <scaler dataType="xsd:double"
      libraryRef="mml" molecular_weight />
  </property>
</propertyArray>
    
```

3.3 CML

3.3.1

본 연구에서는 화학분자식이 목록선택 방식으로 입력이 되면 위에서 제시된 원자 모델링의 스키마를 이용하여 XML 기반의 화학분자식 소스를 설계하는 것을 목표로 한다. Microsoft Visual C++을 통해 개발하였으며, 선택된 원소들을 윈도우에 배치하기 위해 DirectX를 이용하였다.

이러한 소스 생성기를 CML-G로 명명하였으며, 다음과 같은 기능 모듈들로 설계된다. 각 모듈들은 위에서 정의된 원자, 분자 모델링 생성을 목표로 설계된다.

설계된 XML 기반 소스 생성기는 화학분자식 표기방식을 목록선택 방식을 이용한다.

설계된 핵심 모듈들은 ① 분자식 연결강도 처리 모듈, ② 분자식 생성 처리 모듈, ③ XML 파서 연결 모듈, 로 구성된다. 이외에도 원자 모델링 모듈, 원자들의 질량을 처리하는 모듈 등을 작성하여 전체 생성기 시스템을 구성한다.

핵심 모듈들에 대한 개략적인 설명을 다음과 같다.

3.3.2

원자들 간의 결합시 분자역학에 따른 연결강도를 산출하는 기능을 수행한다. 분자동역학 알고리즘이 적용되어 분자생성시 원자들 간의 결합각도에 대한 정보를 생성해 내는 역할을 한다.

MOLECULAR_BOND
DWORD dwRef1 DWORD dwRef2 BYTE btOrder
bool SetBond(DWORD dwRef1, DWORD dwRef2) DWORD GetRef() bool SetRef(DWORD dwRef) bool SetOrder(BYTE btOrder) BYTE GetOrder()

3.3.3

입력된 원자들 간의 조합에 따라 적합한 분자식을 생성하는 기능을 수행한다. 연결강도 처리 모듈의 정보를 이용하여 목록선택 방식으로 입력된 원자들의 연결을 CML 형태의 소스로 구성해 내는 역할을 한다.

MOLECULAR_MODEL
DWORD dwID std: :wstring elementString char elementType; D3DXVECTOR3 pos;
bool SetElementg(DWORD dwID, wstring) DWORD GetID(std: :wsting elementSting) bool SetString(std: :wsting elementSting) bool SetType(char* elementType) D2DXVECTOR3 GetPos(SWORD dwID)

3.3.4 XML

이 모듈은 입력 인터페이스와 XML 파서를 연결하는 모듈로서, 3D 화학분자식 XML 문서에 대해 파싱, 열기, 저장 등을 수행한다.

CXML
MSXML2::IXMLDOMDocument2Ptr pDoc MSXML2::IXMLDOMNodeListPtr pNodeList MSXML2::IXMLDOMNamedNodeMapPtr MSXML2::IXMLDOMNodePtr m_pNode
HRESULT Init(wstring file, wstring token) HRESULT Destroy(); HRESULT NodeParsing(IXMLDOMNodePtr Node) HRESULT Parsing(IXMLDOMNodePtr Node)

3.4 XML

본 연구에서 설계된 CML-G를 이용하여 분자식 XML 소스를 생성하는 예제를 다음과 같이 제시한다. 우선 CML-G의 초기화면은 다음과 같다.

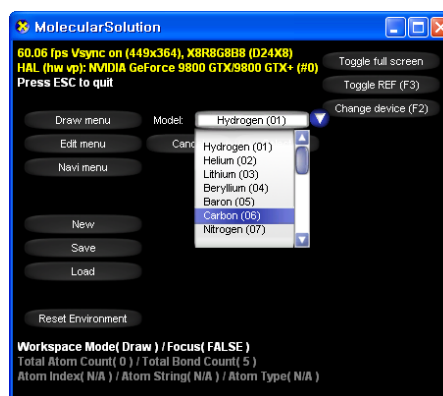


그림 2. CML-G 초기화면
Fig 2. CML-G Initial Window

위의 화면에서 Draw 버튼을 이용하여 원소 목록을 펼친후 원하는 원소를 선택한다. 선택된 원소들을 빈 공간에 배치하고, 연결한다. 원소들로 구성된 분자의 분자식을 생성하기 위해서는 Save 버튼을 클릭하면 된다.

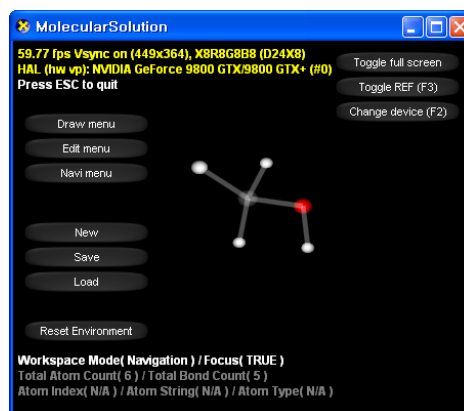


그림 3. CH₃OH 배치
Fig 3. CH₃OH Drawing

다음은 메탄올의 분자식인 CH_3OH 를 CML-G에서 구성한 후 XML 소스로 생성한 결과이다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<molecular>
<elementArray>
<element id="e3" emementType="H" x="-1.8"
y="-7.6" z="1.0"/>
<element id="e1" emementType="H" x="9.4"
y="32.9" z="-1.9"/>
<element id="e2" emementType="H" x="-14.3"
y="24.2" z="-50"/>
<element id="e0" emementType="C" x="1.8"
y="15.1" z="0.5"/>
<element id="e4" emementType="O" x="24.6"
y="11.0" z="-0.8"/>
<element id="e5" emementType="H" x="26.6"
y="-10.2" z="-0.7"/>
</elementArray>
<bondArray>
<bond elementRef1="e3" emementRef2="e0"
order="1"/>
<bond elementRef1="e1" emementRef2="e0"
order="1"/>
<bond elementRef1="e2" emementRef2="e0"
order="1"/>
<bond elementRef1="e0" emementRef2="e4"
order="1"/>
<bond elementRef1="e4" emementRef2="e5"
order="1"/>
</bondArray>
<propertyArray>
<property dataType="scalar"
libraryRef="molecular_weight"/>
<property dataType="color"
libraryRef="molecular_color"/>
</propertyArray>
</molecular>
```

다음은 복잡한 TNT의 분자식인 $\text{C}_7\text{H}_5\text{N}_3\text{O}_6$ 를 CML-G에서 구성한 후 XML 소스로 생성한 결과이다.

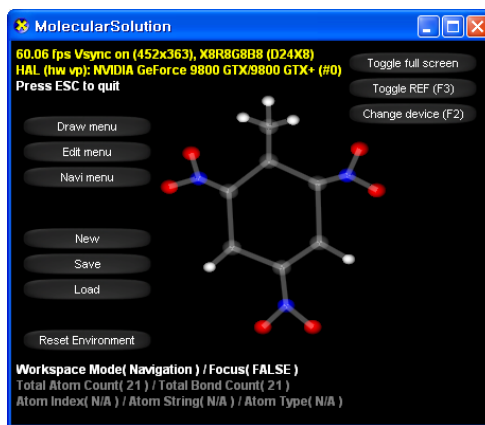


그림 4. $\text{C}_7\text{H}_5\text{N}_3\text{O}_6$ 배치
Fig 4. $\text{C}_7\text{H}_5\text{N}_3\text{O}_6$ Drawing

```
<?xml version="1.0" encoding="EUC-KR"?>
<molecular>
<elementArray>
<element id="e0" emementType="C" x="7.4"
y="38.8" z="-2.1"/>
<element id="e7" emementType="N" x="14.8"
y="-38.7" z="-2.4"/>
-----중략-----
<element id="e8" emementType="N" x="40.8"
y="30.8" z="-4.2"/>
<element id="e14" emementType="O" x="25.9"
y="-49.6" z="-5.3"/>
<element id="e15" emementType="C" x="7.3"
y="56.4" z="-5.5"/>
<element id="e20" emementType="H" x="17.5"
y="59.6" z="-6.8"/>
</elementArray>
<bondArray>
<bond elementRef1="e18" emementRef2="e15"
order="1"/>
<bond elementRef1="e19" emementRef2="e15"
order="1"/>
-----중략-----
<bond elementRef1="e9" emementRef2="e6"
order="1"/>
</bondArray>
<propertyArray>
<property dataType="scalar"
libraryRef="molecular_weight"/>
<property dataType="color"
libraryRef="molecular_color"/>
</propertyArray>
</molecular>
```

