

GML문서의 공간연산 처리를 위한 DOM 인터페이스의 확장

김호석*, 박순영, 정원일, 배해영

*인하대학교 전자계산공학과

{bluesnow, sunny, wncung}@dmlab.inha.ac.kr, hybae@inha.ac.kr

Extended DOM Interfaces for Spatial Operation in GML

Ho-Seok Kim*, Soon-Young Park, Wonil Chung, Hae-Young Bae

*Dept. of Computer Science & Engineering, Inha University

요 약

OGC(Open GIS Consortium)에서는 GIS 분야에서 표준포맷을 이용한 공간데이터의 저장 및 상호운용성을 활성화하기 위하여 공간데이터를 XML(eXtensible Markup Language)로 인코딩하는 GML(Geographic Markup Language) 표준 메커니즘을 제안하였다. GML의 공간데이터를 이용하여 공간연산을 하기 위해서는 GML 문서에 접근하여 문서 내에 있는 공간객체를 다룰 수 있는 기능이 필요하다. GML의 기반인 XML에는 이러한 기능을 가진 표준인터페이스인 DOM(Document Object Model)이 있으나, GML문서 내의 공간객체에 대한 공간연산을 위한 인터페이스는 지원하지 않는다. 그래서 본 논문에서는 GML 문서의 공간데이터에 대한 공간연산을 위해 XML의 DOM 인터페이스를 확장하는 방안을 제안한다.

1. 서론

인터넷의 활용도가 일상생활에까지 영향을 미치면서, 수많은 데이터들이 다수의 사용자에 의해서 인터넷상에서 공유되고 있다. 이렇게 다양한 포맷의 데이터 증가에 대처하기 위하여 문서의 표준 교환 포맷으로써 확장성과 유연성의 특징을 가진 XML을 기반으로 한 문서의 사용이 증가하고 있다. 이러한 데이터 중에서도 지리정보에 대한 활용도가 높아지면서 각종 포맷으로 구성되어진 공간데이터에 대한 상호운용성의 필요가 요구되어지고 있다.

OGC에서는 GIS 분야에서 표준포맷을 이용한 공간데이터의 저장 및 상호운용성을 활성화하기 위하여 공간데이터를 XML로 인코딩하는 GML[1] 표준 메커니즘을 제안하였다. GML의 공간데이터를 이용하여 공간연산을 하기 위해서는 GML 문서에 접근하여 문서 내에 있는 공간객체를 다룰 수 있는 기능이 필요하다. GML의 기반인 XML에는 이러한 기능을 가진 표준인터페이스인 DOM이 있으나, GML문서 내의 공간객체에 대한 공간연산을 위한 인터페이스는 지원하지 않는다. 그래서 본 논문¹⁾에서는 GML 문서의 공간데이터에 대한 공간연산을 위해 XML의 DOM 인터페이스[3]를 확장하는 방안을 제안한다.

공간연산을 위한 DOM 인터페이스의 확장은 크게 두 부

분으로 나누어 볼 수 있는데 DOMSpatialOperation 인터페이스와 DOMSpatialIndex 인터페이스이다. DOMSpatialOperation은 OGC의 Abstract Specification에서 제시하고 있는 공간연산자를 지원하며, DOMSpatialIndex는 공간색인을 지원하기 위한 인터페이스로서 공간데이터에 대한 여과-정제과정에 사용되어진다.

본 논문의 구성은 다음과 같다. 2장 관련 연구에서는 OGC에서 제안한 GML의 기본사항에 대하여 살펴보고, GML의 공간연산을 위하여 이용할 DOM에 대한 구성 및 인터페이스를 살펴본다. 3장에서는 기존의 XML DOM 인터페이스에 확장되는 두 가지의 공간 연산 인터페이스 구조를 설명하며, 4장에서는 공간연산 인터페이스가 추가된 DOM을 이용한 공간 연산 과정 예제를 설명한다. 마지막으로 5장에서는 본 논문의 결론 및 향후 연구과제에 대하여 기술한다.

2. 관련연구

2.1 GML

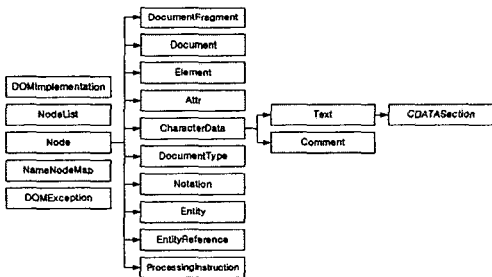
OGC에서는 XML 기술을 이용하여 GIS분야에서의 공간데이터의 표준적인 저장 및 상호 운용성을 활성화하기 위하여 공간 데이터를 XML로 인코딩하는 GML을 제안하였다. GML사항은 공간데이터와 비공간 데이터를 모두 포함한 지리정보의 저장 및 전송을 위한 XML 인코딩 메커니즘을 제공한다. GML 2.0 버전에서는 공간데이터를 GML

1) 본 연구는 정보통신부의 대학 S/W 연구센터 지원사업의 연구 결과임.

로 인코딩하기 위하여 Geometry 스키마(geometry.xsd)와 Feature 스키마(feature.xsd) 두 개의 스키마를 기본적으로 필요로 한다. Geometry 스키마는 GML 1.0에서 점, 선, 면과 같은 기하학적 요소를 나타내던 Geometry.dtd를 대체하기 위한 스키마이며, Feature 스키마는 GML 2.0에서 사용하는 Feature Collection 모델을 정의하기 위한 스키마이다. 아직 표준으로는 정해지지 않았지만 XLink 스키마는 링크 어트리뷰트를 제공하기 위해서 임시로 정의해놓은 스키마이다. 또한 어플리케이션 스키마는 다양한 지리 정보에 대한 요구를 충족시키기 위해 GML 사양에서 정의해 놓은 기본 스키마를 기반으로 하여 사용자가 직접 작성하여 참조하는 스키마이다.

2.2 DOM 인터페이스

DOM은 XML 문서의 내용을 나타내는 객체 모델이다. W3C의 DOM Level 1 Specification에서는 현재 DOM이 속성, 메소드, 이벤트로 제공해야 할 사항을 정의하고 있다. W3C의 DOM은 DOM을 구성하는 다른 객체에 대해 인터페이스는 정의하지만, 구현에 대해서는 어떤 사양도 제공하지 않는다. 개발자는 DOM안에 있는 객체를 이용하여 문서로부터 데이터를 읽기/변경/추가/삭제연산을 수행할 수 있다. 또한 문서 항해(navigation)를 위한 표준적인 기능 정의를 제공하고 XML 문서의 내용과 구조를 조작할 수 있는 기능을 제공한다. 다음 [그림 2-1]은 DOM을 구성하는 객체에 대한 클래스 계층을 보여준다.



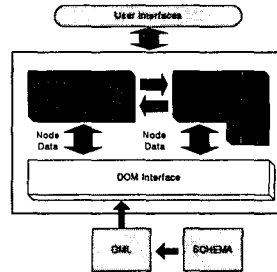
[그림 2-1] DOM을 구성하는 객체에 대한 클래스 계층

3. 공간연산을 위한 DOM 인터페이스의 확장

본 논문에서는 GML 문서에 저장되어 있는 공간 정보에 대한 공간 연산을 수행하기 위해서 기존의 XML 문서에 대한 접근 수단으로 사용되는 DOM 인터페이스를 확장하여 공간연산을 수행할 수 있는 공간 인터페이스의 추가한다.

DOM은 XML문서를 표현하는데 있어 기반 객체인 노드를 상속받아서 문서내용을 객체화하고, 각 노드를 조작할 수 있는 인터페이스를 제공한다. 이러한 기본적인 DOM의 각 노드와 인터페이스간의 API 구성에 공간데이터에 대한 연산을 하기 위한 노드객체로부터 상속받은 새로운 공간 노드를 정의하며, 이러한 노드들 간의 공간연산 인터페이스를 정의한다. 다음 [그림 3-1]은 확장된 DOM의 인터페

이스간의 구성도를 나타낸다. 기존의 DOM 인터페이스에 공간연산을 위한 DOMSpatialOperation 인터페이스와 공간인덱스구성을 위한 DOMSpatialIndex 인터페이스가 확장된 구조를 가지며, GML문서를 읽어서 공간객체를 저장하고 있는 노드를 이용하여 공간연산과 공간인덱스 구성을 수행한다.



[그림 3-1] 공간 연산을 위한 DOM의 확장 인터페이스

3.1 DOMSpatialOperation

DOMSpatialOperation 인터페이스는 각 공간객체의 좌표값이 저장되어 있는 노드를 이용하여 공간연산 메소드에서 해당 공간 연산을 수행하게 된다. 다음[표 3-1]은 DOMSpatialOperation 인터페이스의 구성을 보여준다.

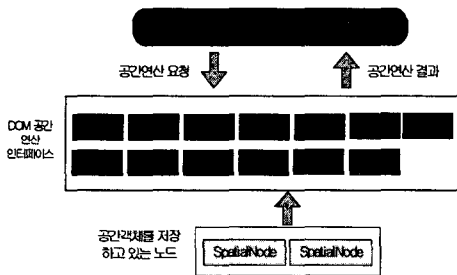
속성	설명
int spatialObjectType	해당 노드의 공간객체의 타입을 나타낸다.(Point, Linestring, Polygon, MultiPoint, MultiLinestring, MultiPolygon)
int spatialOperationType	노드에 저장되어 있는 공간객체를 이용한 공간연산의 타입을 나타낸다.

메소드	설명
boolean Intersects(node left, node right)	두 개의 노드에 대한 Intersect 관계를 가지면 TRUE, 아니면 FALSE를 반환한다.
boolean Disjoint(node left, node right)	두 개의 노드가 Disjoint 관계를 가지면 TRUE, 아니면 FALSE를 반환한다.
boolean Contains(node left, node right)	두 개의 노드가 Contain 관계를 가지면 TRUE, 아니면 FALSE를 반환한다.
boolean Equal(node left, node right)	두 개의 노드가 Equal 관계를 가지면 TRUE, 아니면 FALSE를 반환한다.
boolean Touches(node left, node right)	두 개의 노드가 Touch 관계를 가지면 TRUE, 아니면 FALSE를 반환한다.
boolean Within(node left, node right)	두 개의 노드가 Within 관계를 가지면 TRUE, 아니면 FALSE를 반환한다.
boolean Overlaps(node left, node right)	두 개의 노드가 Overlap 관계를 가지면 TRUE, 아니면 FALSE를 반환한다.
boolean Crosses(node left, node right)	두 개의 노드가 Cross 관계를 가지면 TRUE, 아니면 FALSE를 반환한다.

메소드	설 명
double buffer(node spatialNode)	공간객체의 일정한 거리 내에 있는 지역을 반환한다.
double Distance(node left, node right)	두 객체 점 사이의 최단거리를 반환한다.
node ConvexHull (node spatialNode)	공간객체의 convex hull 관계의 공간 객체를 반환한다.
node Intersection (node left, node right)	두 객체 사이의 intersect 되는 공간객체를 반환한다.
node Union (node left, node right)	두 객체 사이의 union 관계의 공간객체를 반환한다.
node Difference (node left, node right)	두 객체 사이의 difference 관계의 공간객체를 반환한다.

[표 3-1] DOMSpatialOperation 인터페이스

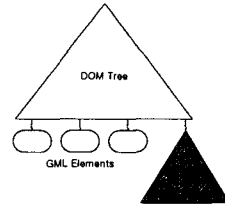
DOMSpatialOperation 인터페이스는 두 개의 공간객체에 대한 위상관계 연산을 지원할 수 있도록 Intersects, Disjoint, Contains, Equals, Touches, Within, Overlaps, Crosses를 정의하며, 공간객체의 분석을 위한 Distance, Buffer, ConvexHull, Intersection, Union, Difference의 공간연산자를 정의한다. 입력 값으로는 DOM 트리에서 실제로 공간좌표값을 저장하고 있는 Spatial Node가 주어지고, 이 노드를 이용하여 공간객체의 상호연관 관계를 연산하게 된다. [그림 3-2]는 이러한 공간연산의 과정을 도식화한 것이다. 사용자의 요청에 의해서 확장된 DOMSpatialOperation의 API를 이용하여 해당 공간객체의 위상관계 연산 및 분석을 수행한다.



[그림 3-2] DOMSpatialOperation 인터페이스를 이용한 공간연산 수행 과정

3.2 DOMSpatialIndex

DOM은 객체간의 검색을 위한 정렬이나 색인 구조에 대한 인터페이스를 제공하지 않는다. 따라서 원하는 공간객체를 찾기 위해서 모든 노드를 순회하면서 순차적인 검색을 한다는 것은 공간연산의 효율성을 떨어뜨리게 된다. 본 논문에서는 효율적인 공간 질의에 대한 처리를 위하여 가장 널리 이용되는 공간 색인인 R 트리 계열의 색인 기법을 이용하여 DOM 인터페이스에 적용하고자 한다.



[그림 3-3] R 트리를 이용한 공간색인 인터페이스 모델

R 트리를 위한 공간인덱스 노드의 구성은 다음과 같다. R 트리의 인덱스노드는 MBR과 자식노드에 대한 포인터 값으로 구성되어지며, R 트리 리프노드는 MBR값과 실제 공간객체의 정보가 저장되어 있는 DOM트리의 노드에 대한 포인터 값으로 구성된다.

다음 [표 3-2]는 공간인덱스를 위한 DOMSpatialIndex 인터페이스의 구성을 보여준다.

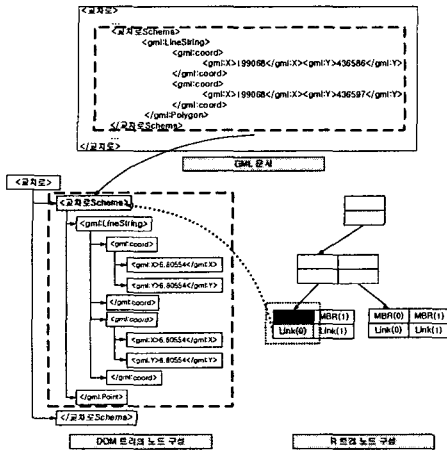
프로퍼티	설 명
int indexType	DOM에 구성되어지는 공간인덱스의 종류를 나타낸다.
DOMMBR keyValue	R 트리의 키 값으로 사용되는 객체의 MBR값을 나타낸다.
IndexNode leafNode	R 트리의 리프노드
IndexNode indexHeadNode	R 트리의 인덱스노드의 헤더

메소드	설 명
BOOL DOMSI_Create()	공간인덱스를 생성, 초기화 한다.
BOOL DOMSI_Insert()	공간인덱스 리프노드를 삽입한다.
BOOL DOMSI_Delete()	공간인덱스 노드를 삭제한다.
BOOL DOMSI_Update()	공간인덱스 노드를 갱신한다.

[표 3-2] DOMSpatialIndex 인터페이스

GML 문서를 읽어서 DOM을 구성할 때, 공간객체 데이터를 삽입한 후 해당 공간객체를 이용하여 R트리에 공간데이터의 MBR을 계산하여, MBR을 키 값으로 하는 새로운 노드를 R 트리에 추가한다.

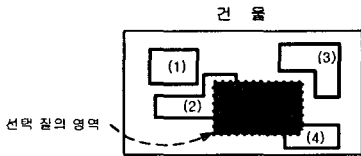
[그림 3-4]는 GML 문서가 DOM에 로딩되어질 때 동시에 공간인덱스를 구성하는 것을 나타내고 있다. <교차로 Schema>는 LineString타입의 공간객체로서 DOM에 해당 공간객체의 좌표값을 이용하여 키 값으로 사용할 MBR을 계산하여 R 트리에 삽입을 하며, <교차로Schema>노드에 링크를 함으로써 MBR을 이용하여 검색 후에 실제 좌표값을 가진 공간객체의 정보를 저장하고 있는 노드를 얻을 수 있다.



[그림 3-4] DOM 트리의 구성 및 R 트리 인덱스 구성

4. GML문서에 대한 공간 연산 처리 과정

다음 [그림 4-1]는 GML문서에 대한 공간연산을 수행하기 위해서 본 논문에서 제안하고 있는 확장된 DOM의 공간연산 인터페이스를 이용한 예제를 보여준다.



[그림 4-1] 공간객체에 대한 공간질의

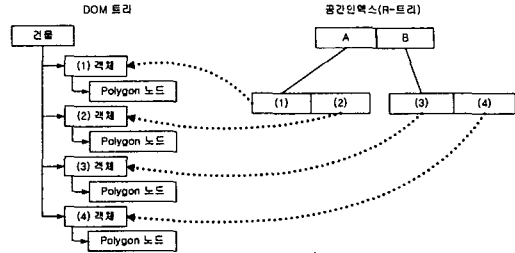
위 그림의 (1)번 건물을 GML문서로 나타내면 다음과 같다

```

<BuildingSchemaMember>
<Building>
<gml:Polygon><gml:outerBoundaryIs><gml:LinearRing>
<gml:coord>
<gml:X>30</gml:X><gml:Y>25</gml:Y>
</gml:coord>
<gml:coord>
<gml:X>110</gml:X><gml:Y>25</gml:Y>
</gml:coord>
<gml:coord>
<gml:X>110</gml:X><gml:Y>84</gml:Y>
</gml:coord>
<gml:coord>
<gml:X>30</gml:X><gml:Y>84</gml:Y>
</gml:coord>
<gml:coord>
<gml:X>30</gml:X><gml:Y>25</gml:Y>
</gml:coord>
</gml:LinearRing></gml:outerBoundaryIs></gml:Polygon>
</Building>
</BuildingSchemaMember>
    
```

나머지 (2), (3), (4) 건물도 같은 방식으로 GML 문서로 표현할 수 있다. 이와 같은 GML 문서는 확장된 DOM 인

터페이스를 통하여 DOM 트리로 구성되는데, GML의 공간객체 노드가 DOM 트리에 삽입되면서 동시에 R 트리로 [그림 4-2]와 같이 구성된다. R 트리의 리프노드는 해당객체의 MBR과 DOM의 공간객체노드의 포인터값이 저장된다.



[그림 4-2] DOM의 공간노드와 공간인덱스 구성

“Building”에 대한 4개의 Polygon타입의 공간객체에 일정 영역의 질의영역이 주어지고, 해당 질의 영역과 겹치는 모든 공간객체를 검색하라는 질의를 수행하는 과정에서 R 트리의 여과과정[4]을 통하여 (2), (3), (4) 번 객체가 후보 객체로 선정된다. DOMSpatialOperation 인터페이스의 Intersection() API를 이용하여 (2), (4) 번 객체가 질의결과에 해당됨을 확인할 수 있다.

5. 결론 및 향후 연구과제

본 논문에서는 GML 문서에 공간 연산을 수행하기 위해서 기존의 XML 문서에 대한 접근 수단으로 사용되는 DOM의 인터페이스를 확장하여 DOMSpatialOperation과 DOMSpatialIndex 인터페이스를 추가하였다.

DOMSpatialOperation은 공간객체간의 위상관계를 연산할 수 있는 공간연산 API를 지원하며, DOMSpatialIndex는 DOM 트리의 공간 노드를 검색방법을 제공하기 위해서 R 트리를 이용한 공간인덱스 API를 지원한다.

향후 연구로는 DOM의 확장된 공간연산 인터페이스의 구현 및 DOM 인터페이스 내에서의 효율적인 공간인덱스 처리 지원하기 위한 메모리기반의 공간인덱스를 제공하는 것이다.

참고문헌

[1] OpenGIS Geography Markup Language (GML) Implementation Specification version 2.1.1 <http://www.opengis.org/techno/implementation.htm>
 [2] OpenGIS Consortium, Inc. OpenGIS Simple Features Specification Revision 1.1 <http://www.opengis.org/techno/implementation.htm>
 [3] DOM, Tree Structure-based API and was issued as a W3C Recommendation in October, 1998[3] <http://www.w3c.org/TR/REC-DOM-Level-1/>
 [4] Guttman, A., R-Trees: An Dynamic Index Structure for Spatial Searching, Proc. Of the ACM SIGMOD Conf on Management of Data, pp.47-57, 1984[4]
 [5] J.E Corcoles, P.Gonzalea, A Specification of a Spatial Query Language over GML, GIS'01 November 9-10, 2001