Data Update across Multi-scale databases

2004. 06. 09

HaeKyong Helen Kang
Multi-scale Databases

- A set of spatial databases.
- Covering the same geographic areas.
- Derivation from existing databases with constraints.

Real World

(Existing DBs)

1:500

1:1000

…

1:5000

1:25000

…

(Derived DBs)
Motivation

- Preservation of Derivation Constraints

Building Block

- Union if distance < 50
- Automatic update

Building

1. s1 s2
2. s3
3. s4 s5
4. s6
Derivation Of Multi-Scale Databases

- Functions for multi-scale databases

Source DB (building) → Multi-Scale Data Operators
(union if distance<50)
constraint

Multi-Scale DB Maintenance (Update) → Consistency

Multi-Scale DB (buildingBlock) → Functions for multi-scale DBs

(The focus of this paper)
Goal and Approach

- **Goal**
  - Preservation of derivation constraints by update propagation
  - Consistency maintenance of a multi-scale database.

- **Approach**
  - Incremental update of multi-scale database
A Multi-Scale Data Model

- FeatureClass
  - SourceFeatureClass
    - geo: SpatialObject
    - attribute1
  - MultiScaleFeatureClass
    - geo: spatialObject
    - derivedAtt
  - SpatialObject
    - derivedBy f()
    - derivation_information
  - Polygon
  - Point
  - Line

- Relation
  - ClassDirectory
    - srcClass
    - drvClass
    - derivationOperator
    - derivationPredicate
    - objectDirectory
  - ObjectDirectory
    - predicate
    - srcObjects
    - drvObject

- 1..0.*
- 0..1
- 1..*
Example of a Multi-Scale Data Model

Diagram showing the relationship between source files and building blocks, with GeoAggregation and derivation information.
Types of a Constraint and an update

- Constraints, based on semantic-abstraction
  - Attribute Specified
    - CLASSIFICATION LandUse TO ResidentialArea WITH ResidentialType;
  - Attribute-Value Specified
    - CLASSIFICATION LandUse TO ResidentialArea WITH ResidentialType;
  - Value Specified
    - GeoAggregation Building TO BuildingBlock With Distance < 50;
### Types of a Constraint and an update

#### Updates for the each constraint

<table>
<thead>
<tr>
<th>update constraint</th>
<th>Insert</th>
<th>Delete</th>
<th>Change Goem</th>
<th>Change Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td><img src="image1" alt="Insert" /></td>
<td><img src="image2" alt="Delete" /></td>
<td><img src="image3" alt="Change Goem" /></td>
<td><img src="image4" alt="Change Attribute" /></td>
</tr>
<tr>
<td>Attribute - Value</td>
<td><img src="image5" alt="Insert" /></td>
<td><img src="image6" alt="Delete" /></td>
<td><img src="image7" alt="Change Goem" /></td>
<td><img src="image8" alt="Change Attribute" /></td>
</tr>
<tr>
<td>Value</td>
<td><img src="image9" alt="Insert" /></td>
<td><img src="image10" alt="Delete" /></td>
<td><img src="image11" alt="Change Goem" /></td>
<td><img src="image12" alt="Change Attribute" /></td>
</tr>
</tbody>
</table>
# A Set of Rules and Algorithms

<table>
<thead>
<tr>
<th>Attribute Constraint</th>
<th>Insert</th>
<th>Delete</th>
<th>Change Goem</th>
<th>Change Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rule1</strong></td>
<td>Algorithm 1</td>
<td>Algorithm 2</td>
<td><strong>Rule3</strong></td>
<td>Algorithm 1,2</td>
</tr>
<tr>
<td><strong>Rule5</strong></td>
<td>Algorithm 1</td>
<td>Algorithm 2</td>
<td><strong>Rule7</strong></td>
<td>Algorithm 1,2</td>
</tr>
<tr>
<td><strong>Rule9</strong></td>
<td>Algorithm 4</td>
<td>Algorithm 5</td>
<td><strong>Rule11</strong></td>
<td>Algorithm 4,5</td>
</tr>
</tbody>
</table>

- **Rule1**: Rule for Insertion
- **Rule2**: Rule for Deletion
- **Rule3**: Rule for Changing Goem
- **Rule4**: Rule for Changing Attribute
- **Rule5**: Rule for Insertion
- **Rule6**: Rule for Deletion
- **Rule7**: Rule for Changing Goem
- **Rule8**: Rule for Changing Attribute
- **Rule9**: Rule for Insertion
- **Rule10**: Rule for Deletion
- **Rule11**: Rule for Changing Goem

*Notations:* 
- Insert: Rule1, Rule5, Rule9
- Delete: Rule2, Rule6, Rule10
- Change Goem: Rule3, Rule7
- Change Attribute: Rule4, Rule8

*Algorithms:* 
- Algorithm 1
- Algorithm 2
- Algorithm 3
- Algorithm 4
- Algorithm 5
- Algorithm 1,2
Prototype of A Multi-Scale Database Manager

- In Developing with a spatial library of ESRI(ArcObjects).
- Main Features of A Multi-Scale Database Manager
Example of Update Propagation Rule 11

- A source database and a multi-scale databases
  - Constraint type: value specified constraint (Distance < 50)
Example of Update Propagation Rule 11

- Update of the source database
  - fsrc3 is moved into the next of fsrc6.
  - Update type: change geometry

- Thus, Rule 11 is applied to this situation
## Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$FC$</td>
<td>Feature Class</td>
<td>a set of features; $FCSRC$: a set of source features; $FC_{DRV}$: a set of derived features from $FCSRC$</td>
</tr>
<tr>
<td>$f$</td>
<td>A set of features in $FC$</td>
<td>$f_i$: $i$th feature, $f$.<em>geo: a spatial shape of a feature $f</em>{ins}$: a set of inserted features</td>
</tr>
<tr>
<td>$R_{CDRY}$</td>
<td>Class Directory Class</td>
<td>a set of relations between $FCSRC$ and $FC_{DRV}$</td>
</tr>
<tr>
<td>$R_{ODRY}$</td>
<td>Object Directory Class</td>
<td>a set of relations between $f_{SRC}$ and $f_{DRV}$</td>
</tr>
<tr>
<td>$att$</td>
<td>Attribute</td>
<td>$FC.att$: an attribute of $FC$; $f.att_j$: $j$th attribute of a feature $f$</td>
</tr>
<tr>
<td>$dom(att)$</td>
<td>Domain of $att$</td>
<td>the ranges of attribute values of $att$</td>
</tr>
<tr>
<td>$v$</td>
<td>An attribute value</td>
<td>$f.a = v$ : $v$ is an attribute value of $f.a$</td>
</tr>
</tbody>
</table>
Example of Update Propagation Rule 11

- Rule 11

ChangeGEOM \( f_{ch} \) in \( FC_{SRC} \)

- \( f_{drv_i} \) in \( FC_{DRV} \) derived from \( f_{ch} \), to be modified

  INSERT \( f_{new1} \)
  INTO \( FC_{DRV} \)
  BY \( \text{union}(f_{src} - f_{ch}) \),
  WHERE \( f_{src} - f_{ch} \) satisfies with \( v \)

  DELETE \( f_{drv_i} \)
  FROM \( FC_{DRV} \)

- \( f_{drv_j} \) in \( FC_{DRV} \), not derived from \( f_{ch} \), to be modified.

  INSERT \( f_{new2} \)
  INTO \( FC_{DRV} \)
  BY \( \text{union}(f_{ch} \cup f_{drv_j}) \)
  WHERE \( f_{ch} \cup f_{drv_j} \) satisfies with \( v \)

  DELETE \( f_{drv_j} \)
  FROM \( FC_{DRV} \)

Algorithm 4 (Insert)

Algorithm 5 (Delete)
Example of Update Propagation Rule 11

Algorithm 4 : Insertion

Input: $F_{C_{SRC}}, F_{C_{DRV}}, R_{CDR_Y}, R_{ODR_Y}$
Output: $F_{C_{DRV}}$, updated

Method:

1: Let $ListUpdateObj$ be a list of objects($f_{ins}$) in $F_{C_{SRC}}$ of which geometry is changed.

2: Let $f_{drv}$ be all objects of $F_{C_{DRV}}$.

3: $constraint \leftarrow R_{CDR_Y}.derivationPredicate$ <50

4: While ($ListUpdateObj \neq \emptyset$) Do

5: Get $f_{ins}$ from $ListUpdateObj$

6: Get $f_{drv}$ satisfies a constraint with $f_{ins}$

7: $F_{C_{DRV}} \leftarrow union(f_{ins} \cup f_{drv})$,

where ($f_{chk} f_{ins} \cup f_{drv}$ satisfies with $constraint$

8: End while
Example of Update Propagation Rule 11

**Algorithm 5: Deletion**

**Input:** $FC_{SRC}$, $FC_{DRV}$, $R_{CDRV}$, $R_{ODRV}$

**Output:** $FC_{DRV}$, updated

**Method:**

1. Let $ListUpdateObj$ be a list of objects ($f_{del}$) deleted from $FC_{SRC}$.
2. Let $ListSrcObjs$ and $ListObjs$ be a list of objects each.
3. Let $f_{drv}$ be a subset of $FC_{DRV}$ where $f_{drv}$ is derived from $f_{del}$.
4. **While** ($ListUpdateObj \neq \emptyset$) **Do**
   5. Get $f_{del}$ from $ListUpdateObj$.
   6. Get $f_{drv}$ from $R_{ODRV}.drvObject$ of $f_{del}$.
   7. $ListSrcObjs \leftarrow R_{ODRV}.srcObjects$ of $f_{drv}$.
   8. $ListObjs \leftarrow \{ListSrcObjs\} - \{ListUpdateObj\}$.
   9. $ListUpdateObj \leftarrow \{ListUpdateObj\} - \{ListSrcObjs\}$.
10. **call** GeoAggretation($ListObjs$).
11. delete $f_{drv}$.
12. **End while**
Example: derivation of building block
Example: update of building block

Diagram showing the process involving Source DB, Modeling Tool, Directory DB, Update Rules, Update Propagation Manager, and Multi-Scale DB (Derived). Additionally, a screenshot of a user interface is shown, which includes options for selecting UPDATE Type (INSERT Data, DELETE Data, CHANGE Geometry, CHANGE Attribute) and importing an update file.
Conclusion

- Consistency between a source and its derived multi-scale databases during updates.
- Update Propagation Rules and Algorithms
  - for types of an update
    - insert,
    - delete,
    - change geometry,
    - change attribute)
  - and for types of a constraint
    - Attribute specified
    - attribute-value specified
    - Value specified
- Guarantee the integrity of a multi-scale database