

General GeoGraphical Database (GGGD) System

野田 五十樹
産業技術総合研究所

Ver. -0.86

目次

| | |
|--|-----------|
| 第 1 章 Protocol | 5 |
| 1.1 Overview | 5 |
| 1.2 Register | 6 |
| 1.3 GetFeature | 8 |
| 1.4 Transaction | 12 |
| 1.4.1 Insert | 12 |
| 1.4.2 Update | 14 |
| 1.4.3 Delete | 15 |
| 1.5 GetCapabilities | 15 |
| 1.6 DescribeFeatureType | 17 |
| 1.7 XML Schema に対する制約 | 19 |
| 1.7.1 仕様上の制約 | 19 |
| 1.7.2 利用可能な XML Schema タグと属性 | 20 |
| 1.7.3 実装上の制約 | 20 |
| 1.8 データ型 | 21 |
| 1.8.1 プリミティブデータ型 | 21 |
| 1.8.2 文字列型の拡張 | 22 |
| 1.8.3 地物および時間表現データ型 | 22 |
| 1.8.4 マルチメディアデータ型 | 22 |
| 1.8.5 被覆データ型 | 23 |
| 1.9 GML 記述上の制約 | 25 |
| 1.9.1 仕様上の制約 | 25 |
| 1.9.2 実装上の制約 | 26 |
| 1.10 Filter 記述上の制約 | 26 |
| 第 2 章 ToDo | 29 |
| 第 3 章 Changes | 33 |

第1章 Protocol

1.1 Overview

GGGD はクライアント-サーバ型のサービスを提供する。GGGD のサービスを利用する場合、クライアントは特定の TCP/IP ポート (既定値は 5050) に対しソケットを開き、以下に述べるプロトコルに従い GGGDServer とやり取りを行う。サービスは、基本的にはまずクライアントがソケットを開いた後、要求を送り、サーバがそれに対する応答を返して 1 つのセッションが終了する。ソケットは 1 セッション毎に閉じられる。¹

GGGD のサービスで用いられるプロトコルおよびデータは全て XML で記述される。扱われるデータは各々木構造のデータ構造をもつものとして扱われ、その単位をデータエントリと呼ぶ。

データエントリには型によって分類され、1 つの型に属するデータエントリはすべて同じ木構造を持つものとする。この木構造は XML Schema [vdV03] により定義され、以下に述べる Register プロトコルによりシステムに登録される。

データベース中のデータエントリを操作するプロトコルとしては、WFS (Web Feature Service) [Ope02] をベースに独自の拡張をしたものを用いる。

GGGD は WFS で定義されているプロトコルのうち、以下のものをサポートする。

- **GetFeature:** データベースに登録されているデータエントリの検索を行う。
- **Transaction:**
 - **Insert:** データベースに、新たなデータエントリを追加記録する。
 - **Update:** データベースに登録されているデータエントリの内容を変更する。
 - **Delete:** データベースに登録されているデータエントリを削除する。
- **GetCapabilities:** データベースが提供するサービスに関する情報を知らせる。
- **DescribeFeatureType:** 登録されているデータの型に関する情報を、XML Schema [vdV03] の形式で知らせる。

さらに、データの型の登録のために、以下の拡張プロトコルをサポートする。

- **Register:** データベースに新たなデータの型を定義し、その型のエントリを扱うための設定などを行う。

¹keep-alive については検討課題である。

1.2 Register

データベースに新しいデータの型を登録するためには、以下の形式の文字列を GGGDServer に送る。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>version of the protocol</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <Register uri="URI for the new data type">
      <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:gml="http://www.opengis.net/gml">
        any XML schema definition.
      </xsd:schema>
    </Register>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ここで、‘URI for the new data type’ は、この登録を参照する際の ID として用いる。たとえばすでに登録されたデータ型を参照する場合は、以下のように XML Schema [vdV03] の include 機能を用いて呼び出すことができる。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <Register uri="bar">
      <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:gml="http://www.opengis.net/gml">
        <xsd:include schemaLocation="foo" />
        any XML schema definition.
      </xsd:schema>
    </Register>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

この例では、‘foo’ としてすでに登録されているスキーマを参照して新たなデータ型あるいはスキーマを定義する。この定義は、後で ‘bar’ として参照することができる。

注意しなければならないのは、この uri はデータの型の名前と異なることである。Register プロトコルでは、XML Schema を用いて複数のデータ型を同時に登録することができる。よって、Register タグの uri は同時に登録された複数のデータ型に対応することになる。

文法

```
<xsd:element name='Register' type='RegisterType' />
<xsd:complexType name='RegisterType'>
  <xsd:complexContent>
    <xsd:sequence>
      <xsd:element name='xsd:schema'
        type='xsd:schemaType'
        minOccurs='0' maxOccurs='unbounded' />
    </xsd:sequence>
    <xsd:attribute name='uri' type='uri' />
  </xsd:complexContent>
</xsd:complexType>
```

ただし、上記の `xsd:schemaType` は、1.7 節で述べる、簡単化された XML Schema である。

例 次の例は、道路のリンク (RoadLink) のデータ型を登録するためのプロトコルである。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGDD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <Register uri="RoadLink.xsd"
      xmlns:xi="http://www.w3.org/2001/XInclude">
      <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:gml="http://www.opengis.net/gml">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            Sample schema for DDT Geographical Information files.
          </xsd:documentation>
        </xsd:annotation>
        <xsd:include schemaLocation="common.xsd" />
        <xsd:element name="RoadLink" type="RoadLinkType"
          substitutionGroup="gml:_Feature" />
        <xsd:complexType name="RoadLinkType">
          <xsd:complexContent>
            <xsd:extension base="GeometryFeatureType">
              <xsd:sequence>
                <xsd:element name="representativePoint" type="geometryPropertyType"/>
                <xsd:element name="nodeList" type="nodeListType" />
                <xsd:element name="roadWidth" type="float" />
                <xsd:element name="nLanes" type="integer" />
                <xsd:element name="direction" type="string" />
              </xsd:sequence>
            </xsd:extension>
          </xsd:complexContent>
        </xsd:complexType>
        <xsd:complexType name="nodeListType">
          <xsd:complexContent>
            <xsd:sequence>
              <xsd:element name="beginId" type="IDType" />
              <xsd:element name="endId" type="IDType" />
            </xsd:sequence>
          </xsd:complexContent>
        </xsd:complexType>
      </xsd:schema>
    </Register>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

この例により登録される RoadLink のデータエントリの例は次のようになる。

```

<RoadLink>
  <gml:geometryProperty>
    <gml:LineString>
      <gml:coordinates>
        -11785.918203,-53652.534844 -11757.449384,-53576.404445
      </gml:coordinates>
    </gml:LineString>
  </gml:geometryProperty>
  <id>1</id>
  <representativePoint>
    <gml:Point>
      <gml:coordinates>-11785.918203,-53652.534844</gml:coordinates>
    </gml:Point>
  </representativePoint>
  <nodeList>
    <beginId>5960</beginId>
    <endId>6058</endId>
  </nodeList>
  <roadWidth>3.000000</roadWidth>
  <nLanes>1</nLanes>
  <direction>true</direction>
</RoadLink>

```

1.3 GetFeature

データベースに記録されているデータエントリーを条件にしたがって検索する時には以下の形式の文字列を GGGDServer に送る。

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>version of the protocol</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <GetFeature>
      <Query typeName="Name of Toplevel Element">
        <Filter>
          any conditions.
        </Filter>
      </Query>
    </GetFeature>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

”Name of Toplevel Element” には、Register プロトコル中の XML Schema で定義された、トップレベルの element の名前、すなわちデータの型を指定する。Filter タグの中には、OGC Filter Encoding Implementation Specification [Ope01] に従って条件を記述する。ただし、現状の実装では 1.10 節に示すような制約がある。

GGGD は GetFeature に対し、以下のような形式で検索結果を返す。


```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>version of the protocol</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <FeatureCollection xmlns:gml="http://www.opengis.net/gml">
      <ElapsedTime>
        <gml:beginPosition>Start Time</gml:beginPosition>
        <gml:endPosition>End Time</gml:endPosition>
      </ElapsedTime>
      <gml:featureMember>
        first element of retrieved data
      </gml:featureMember>
      <gml:featureMember>
        second element of retrieved data
      </gml:featureMember>
      ...
    </FeatureCollection>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ここで、*Start Time*, *End Time* は検索の開始および終了時刻を示す。このうち終了時刻については、次回の検索においてデータベースの差分情報を取りたい場合に、データの更新時刻の条件として用いることができる。

文法 GetFeature およびその応答の文法は、WFS (Web Feature Service) [Ope02] の GetFeature に従う。ただし、応答に関しては、以下の拡張・省略がある。

- WFS では FeatureCollection に BoundaryBox 要素が必要だが、GGGD では省略される。これは無駄な検索を避け処理を高速化するためである。
- GGGD では処理時間を示す ElapsedTime 要素が FeatureCollection に付加される。

例 次の例は、記録されている道路のリンク (RoadLink) を、地域および道幅の条件を指定して検索する場合のプロトコルである。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGDD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <GetFeature>
      <Query typeName="RoadLink">
        <Filter>
          <And>
            <BBox>
              <PropertyName>gml:geometryProperty</PropertyName>
              <gml:Box>
                <gml:coordinates>-11200.0,-51200.0 -11000.0,-51000.0</gml:coordinates>
              </gml:Box>
            </BBox>
            <PropertyIsGreaterThan>
              <PropertyName>roadWidth</PropertyName>
              <Literal>2.0</Literal>
            </PropertyIsGreaterThan>
          </And>
        </Filter>
      </Query>
    </GetFeature>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

これに対し、GGDDServer は以下のような応答を返す。

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGSD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <FeatureCollection xmlns:gml="http://www.opengis.net/gml">
      <ElapsedTime>
        <beginPosition>2004-08-14T00:53:26</beginPosition>
        <endPosition>2004-08-14T00:53:26</endPosition>
      </ElapsedTime>
      <gml:featureMember>
        <RoadLink>
          <gml:geometryProperty>
            <gml:LineString>
              <gml:coordinates>
                -10961.48356,-51103.930527 -11026.48803,-51089.745352
              </gml:coordinates>
            </gml:LineString>
          </gml:geometryProperty>
          <id>2143</id>
          <representativePoint>
            <gml:Point>
              <gml:coordinates>-10961.48356,-51103.930527</gml:coordinates>
            </gml:Point>
          </representativePoint>
          <nodeList>
            <beginId>9504</beginId>
            <endId>9172</endId>
          </nodeList>
          <roadWidth>3</roadWidth>
          <nLanes>1</nLanes>
          <direction>true</direction>
        </RoadLink>
      </gml:featureMember>
      <gml:featureMember>
        <RoadLink>
          <gml:geometryProperty>
            <gml:LineString>
              <gml:coordinates>
                -11178.713929,-51127.3582 -11240.947082,-51109.789681
              </gml:coordinates>
            </gml:LineString>
          </gml:geometryProperty>
          <id>2144</id>
          <representativePoint>
            <gml:Point>
              <gml:coordinates>-11178.713929,-51127.3582</gml:coordinates>
            </gml:Point>
          </representativePoint>
          <nodeList>
            <beginId>8524</beginId>
            <endId>8261</endId>
          </nodeList>
          <roadWidth>3</roadWidth>
          <nLanes>1</nLanes>
          <direction>true</direction>
        </RoadLink>
      </gml:featureMember>
      ...
    </FeatureCollection>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

1.4 Transaction

データエントリーの追加記録 (Insert)、更新 (Update) および削除 (Delete) には、WFS (Web Feature Service) [Ope02] の Transaction プロトコルを用いる。

Transaction の共通の形式は以下の通りである。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGSD'>
  <SOAP-ENV:Header>
    <protocolVersion>version of the protocol</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <Transaction>
      Insert or Update or Delete element
      Insert or Update or Delete element
      ...
    </Transaction>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1 つの Transaction 中に複数の Insert/Update/Delete が現れて良い。また、これらの複数の Insert/Update/Delete は異種のものが混ざっていてもよい。これら複数の Insert/Update/Delete は順に処理される。

Transaction プロトコルに対して、GGSDServer は以下のような返答を返す。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGSD'>
  <SOAP-ENV:Header>
    <protocolVersion>version of the protocol</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <TransactionResponse>
      <ElapsedTime>
        <gml:beginPosition>Start Time</gml:beginPosition>
        <gml:endPosition>End Time</gml:endPosition>
      </ElapsedTime>
      <TransactionResult>
        <Status>SUCCESS or FAIL</Status>
        [<Message>message from database system</Message>]?
        [<InsertCount>number of insert</InsertCount>]?
      </TransactionResult>
    </TransactionResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1.4.1 Insert

データエントリの追加登録には、以下のような Insert プロトコルを用いる。

```
<Insert>
  Data Element
  Data Element
  ...
</Insert>
```

文法 Insert の文法は WFS (Web Feature Service) [Ope02] の仕様に従う。

例 以下は、RoadLink を 3 つ登録するためのプロトコルである。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <Transaction>
      <Insert>
        <RoadLink>
          <gml:geometryProperty>
            <gml:LineString>
              <gml:coordinates>
                -11785.918203,-53652.534844 -11757.449384,-53576.404445
              </gml:coordinates>
            </gml:LineString>
          </gml:geometryProperty>
          <id>1</id>
          <representativePoint>
            <gml:Point>
              <gml:coordinates>-11785.918203,-53652.534844</gml:coordinates>
            </gml:Point>
          </representativePoint>
          <nodeList>
            <beginId>5960</beginId>
            <endId>6058</endId>
          </nodeList>
          <roadWidth>3.000000</roadWidth>
          <nLanes>1</nLanes>
          <direction>true</direction>
        </RoadLink>
        <RoadLink>
          ...
        </RoadLink>
        <RoadLink>
          ...
        </RoadLink>
      </Insert>
    </Transaction>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

これに対し Server は以下のような応答を返す。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <TransactionResponse>
      <ElapsedTime>
        <gml:beginPosition>2004-09-04T02:49:04</gml:beginPosition>
        <gml:endPosition>2004-09-04T02:49:04</gml:endPosition>
      </ElapsedTime>
      <TransactionResult>
        <Status>SUCCESS</Status>
        <InsertCount>3</InsertCount>
      </TransactionResult>
    </TransactionResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1.4.2 Update

データエントリの変更には、以下のような Update プロトコルを用いる。

```
<Update typeName="Name of Data Element"
  [mode="preserve or override"]>
  <Property>
    <Name>XPath of the property to modify</Name>
    <Value>new value</Value>
  </Property>
  [<Property> ... </Property>]*
  <Filter>
    any conditions
  </Filter>
</Update>
```

ここで、<Update> タグ中の mode 属性の値は以下のような意味を持つ。

- mode の値が preserve である場合、元のデータエントリには updated フラグが付けられ、変更後の値は新しいデータエントリとして記録される。
- mode の値が override である場合、元のデータエントリの値が直接書き換えられる。

なお、mode 属性が省略された場合、preserve が指定されたものとして動作する。

文法 Update の文法は、mode 属性以外は WFS (Web Feature Service) [Ope02] の仕様に従う。

例 以下の例は、ある地域に存在する道幅が 2.0 のロードリンクのノードリンクをつなぎ変える操作を示すプロトコルである。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <Transaction>
      <Update typeName="RoadLink">
        <Property>
          <Name>nodeList/beginId</Name>
          <Value>12345</Value>
        </Property>
        <Filter>
          <And>
            <BBox>
              <PropertyName>gml:geometryProperty</PropertyName>
              <gml:Box>
                <gml:coordinates>-12200.0,-55100.0 -12000.0,-55000.0</gml:coordinates>
              </gml:Box>
            </BBox>
            <PropertyIsEqualTo>
              <PropertyName>roadWidth</PropertyName>
              <Literal>2.0</Literal>
            </PropertyIsEqualTo>
          </And>
        </Filter>
      </Update>
    </Transaction>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1.4.3 Delete

データエントリの削除には、以下のような Delete プロトコルを用いる。

```
<Delete typeName="Name of Data Element"
  [mode="preserve or override"]>
  <Filter>
    any conditions
  </Filter>
</Delete>
```

ここで、<Delete> タグ中の mode 属性の値は以下のような意味を持つ。

- mode の値が preserve である場合、元のデータエントリには deleted フラグが付けられるだけで、データベースからは消去されない。
- mode の値が override である場合、元のデータエントリはデータベースから消去される。

なお、mode 属性が省略された場合、preserve が指定されたものとして動作する。

文法 Delete の文法は WFS (Web Feature Service) [Ope02] の仕様に従う。

例 以下の例は、ある地域に存在する道幅が 2.0 のロードリンクを削除するプロトコルである。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <Transaction>
      <Delete typeName="RoadLink">
        <Filter>
          <And>
            <BBox>
              <PropertyName>gml:geometryProperty</PropertyName>
              <gml:Box>
                <gml:coordinates>-12200.0,-55100.0 -12000.0,-55000.0</gml:coordinates>
              </gml:Box>
            </BBox>
            <PropertyIsEqualTo>
              <PropertyName>roadWidth</PropertyName>
              <Literal>2.0</Literal>
            </PropertyIsEqualTo>
          </And>
        </Filter>
      </Delete>
    </Transaction>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1.5 GetCapabilities

GetCapabilities

GGGDServer が提供する WFS (Web Feature Service) [Ope02] のサービスの情報を取得するためには、以下の形式の文字列を GGGDServer に送る。

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>version of the protocol</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <GetCapabilities/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

GGGD は GetCapabilities に対し、以下のような形式でサービスの情報を回答する。

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>version of the protocol</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <WFS_Capabilities>
      <Capabilities>
        list of services
      </Capabilities>
      <FeatureTypeList>
        <Operations>
          list of operations available for all feature types
        </Operations>
        <FeatureType>
          <Name>a name of a feature type</Name>
        </FeatureType>
        <FeatureType>
          <Name>a name of a feature type</Name>
        </FeatureType>
        ...
      </FeatureTypeList>
    </WFS_Capabilities>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

list of services は利用可能なサービスのリストであり、以下の項目が列挙される。

- <GetCapabilities/>
- <DescribeFeatureType/>
- <Transaction/>
- <GetFeature/>
- <Register/>

list of operations available for all feature types にはすべてのデータ型に対して利用可能なデータベース操作のリストであり、以下の項目が列挙される。

- <Insert/>
- <Update/>
- <Delete/>
- <Query/>

文法 GetCapabilities の文法は WFS (Web Feature Service) [Ope02] の仕様に従う。

例 以下の例は、ある GGGDServer で利用可能なサービスの情報を取得した場合の応答の例である。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <WFS_Capabilities>
      <Capabilities>
        <GetCapabilities/>
        <DescribeFeatureType/>
        <Transaction/>
        <GetFeature/>
        <Register/>
      </Capabilities>
      <FeatureTypeList>
        <Operations>
          <Insert/>
          <Update/>
          <Delete/>
          <Query/>
        </Operations>
        <FeatureType>
          <Name>Node</Name>
        </FeatureType>
        <FeatureType>
          <Name>RoadLink</Name>
        </FeatureType>
        <FeatureType>
          <Name>Building</Name>
        </FeatureType>
      </FeatureTypeList>
    </WFS_Capabilities>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1.6 DescribeFeatureType

GGGDServer が扱っているあるデータ型の情報を取得するためには、以下の形式の文字列を GGGDServer に送る。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>version of the protocol</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <DescribeFeatureType>
      <TypeName>a name of a feature type</TypeName>
      <TypeName>a name of a feature type</TypeName>
      ...
    </DescribeFeatureType>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

これに対し、GGGDServer は以下のような応答を返す。

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGDD'>
  <SOAP-ENV:Header>
    <protocolVersion>version of the protocol</protocolVersion>
  </SOAP-ENV:Header>
  <xsd:schema xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
    schema definitions for the feature types
  </xsd:schema>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

schema definitions for the feature types の内容は、基本的には Register プロトコルによって登録されたものと同じである。

文法 DescribeFeatureType の文法は WFS (Web Feature Service) [Ope02] の仕様に従う。

例 以下の例は、道路 (RoadLink) のデータ構造の情報を取得するためのプロトコルの例である。

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGDD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <DescribeFeatureType>
      <TypeName>RoadLink</TypeName>
    </DescribeFeatureType>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

これに対し、GGGDSerVer は以下のような応答を返す。

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://www.carc.aist.go.jp/~noda/Rescue/GGGD'>
  <SOAP-ENV:Header>
    <protocolVersion>0.00</protocolVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <xsd:schema xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
      <!-- ----- RoadLink's definition ----- -->
      <xsd:annotation>
        <xsd:documentation xml:lang='en'>
          ...
        </xsd:documentation>
      </xsd:annotation>
      <xsd:include schemaLocation='common.xsd' />
      <xsd:complexType name='GeometryFeatureType'>
        <xsd:complexContent>
          <xsd:extension base='gml:AbstractFeatureType'>
            <xsd:sequence>
              <xsd:element maxOccurs='1' name='gml:geometryProperty' ... />
              <xsd:element maxOccurs='1' name='id' nillable='true' ... />
            </xsd:sequence>
          </xsd:extension>
        </xsd:complexContent>
      </xsd:complexType>
      ... <!-- --- schema's defined in 'common.xsd' here --- -->
      <xsd:element name='RoadLink' type='RoadLinkType' substitutionGroup='gml:_Feature' />
      <xsd:complexType name='RoadLinkType'>
        <xsd:complexContent>
          <xsd:extension base='GeometryFeatureType'>
            <xsd:sequence>
              <xsd:element name='representativePoint' type='geometryPropertyType' />
              <xsd:element name='nodeList' type='nodeListType' />
              <xsd:element name='roadWidth' type='float' />
              <xsd:element name='nLanes' type='integer' />
              <xsd:element name='direction' type='string' />
            </xsd:sequence>
          </xsd:extension>
        </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name='nodeListType'>
        <xsd:complexContent>
          <xsd:sequence>
            <xsd:element name='beginId' type='IDType' />
            <xsd:element name='endId' type='IDType' />
          </xsd:sequence>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:schema>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

1.7 XML Schema に対する制約

1.7.1 仕様上の制約

Register プロトコルでは、データ型およびスキーマを XML Schema [vdV03] により記述するが、XML Schema の冗長性を軽減するため、以下のような制約を設けている。

- 以下のものは、かならず schema タグの中のトップレベルで現れなければならない。

- xsd:include
- xsd:complexType
- xsd:simpleType
- xsd:group

すなわち、全ての type はトップレベルで定義されていなければならない、`xsd:element` の内部に直接 type を記述することはできない。

- restriction は単に base type を継承するのみで、制約は無視される。

1.7.2 利用可能な XML Schema タグと属性

GGGD で認識される XML Schema のタグおよび属性 (attribute) は以下のもののみである。

- `<xsd:schema [xmlns:prefix="URI"]*>`
- `<xsd:include schemaLocation="URI">`
- `<xsd:element name="Name of Element" type="Name of Type" [minOccurs="number"]? [maxOccurs="number"]?>`
- `<xsd:complexType name="Name of Type">`
- `<xsd:group name="Name of Group">`
- `<xsd:sequence>`
- `<xsd:choice>`
- `<xsd:all>`
- `<xsd:complexContent>`
- `<xsd:simpleContent>`
- `<xsd:extension base="Name of Type">`
- `<xsd:restriction base="Name of Type">`

1.7.3 実装上の制約

現在の GGGD の実装では、上記の制約に加え以下のような制約がある。これらは将来的には対応する予定となっている。

- `xsd:attribute` は定義できない (無視される)。すべてのデータは `element` として記述されなければならない。

1.8 データ型

この節では、GGGD において定義なしで用いることのできるプリミティブデータ型および既定の共通のデータ型について述べる。プリミティブデータ型についてはシステムの内部で直接扱われる。共通データ型については ‘common.xsd’ という URI で指定されるファイルに定義されている。

1.8.1 プリミティブデータ型

GGGD では、以下に示す型を組み込みのプリミティブデータ型として扱う。

- xsd:integer : 整数。10 進数で表記されることを仮定する。4byte 整数として扱われる。
- xsd:unsignedByte, unsignedLong, unsignedInt : 非負整数。いずれも 4byte 非負整数として扱われる。10 進数で表記されることを仮定する。²
- xsd:float : 実数。10 進数で表記されることを仮定する。
- xsd:string : 文字列。長さの制限は無視される。文字列の最大長は 65,535 byte である。
- xsd:SqlTinyBlob : 文字列。文字列の最大長は 255 byte である。
- xsd:SqlBlob : 文字列。文字列の最大長は 65,535 byte である。
- xsd:SqlMediumBlob : 文字列。文字列の最大長は 16Mbyte である。
- xsd:SqlLongBlob : 文字列。文字列の最大長は 4Gbyte である。
- gml:geometryPropertyType : 幾何形状。表記方法は GML 3.0 [Ope03] に準拠する。ただし、1.9 節で述べる制約がある。
- gml:TimePositionType : 点時刻。表記方法は ISO-8601 [Int04] に準拠する。ただし冗長性を省くため、年月日および時分秒の記述方法は

| *yyyy-mm-ddThh:mm:ss*

のみとする。

ただし、プリミティブデータ型とは、この型をそれ以上細かく分解せずに取り扱うことを示す。よって、特に幾何形状のようにその中に構造を持つ物でも、それがプリミティブ型であるなら、検索などでその内部構造を指定した条件を記述することはできない。

また文字列については、XML で許される文字列とする。

²8 進数、16 進数は検討課題。

1.8.2 文字列型の拡張

- IDType : 64 文字以下の文字列
- NameType : 256 文字以下の文字列
- URIType : 256 文字以下の文字列

1.8.3 地物および時間表現データ型

- GeometryFeatureType : 地物のベースタイプ。gml:geometryProperty (type=gml:geometryPropertyType) と id (type=IDType) を要素として持つ。
- TimeInstantType : GML の定義に従い、点時刻を表現する際に用いる。表記方法は以下の形式をとる。

```
<timeInstantElement>
  <gml:timePosition>a data of gml:TimePositionType</gml:timePosition>
</timeInstantElement>
```

以下は (阪神大震災の本震時刻を表す) 点時刻の例である。

```
<firstImpactTime>
  <gml:timePosition>1995-01-17T05:46:00</gml:timePosition>
</firstImpactTime>
```

- TimePeriodType : GML の定義に従い、特定の時間区間の開始および終了時刻を表現するのに用いる。表記方法は以下の形式をとる。

```
<timePeriodElement>
  <gml:beginPosition>a data of gml:TimePositionType</gml:beginPosition>
  <gml:endPosition>a data of gml:TimePositionType</gml:endPosition>
</timePeriodElement>
```

以下は、(2004 年 11 月 10 日の水戸の昼間の時間を表す) 時間区間の例である。

```
<dayTime>
  <gml:beginPosition>2004-11-10T05:41:00</gml:beginPosition>
  <gml:endPosition>2004-11-10T17:09:00</gml:endPosition>
</dayTime>
```

1.8.4 マルチメディアデータ型

- MultiMediaDataEntityType : 画像、音声などのデータの実体を格納する際に用いる。具体的には URI、MIMEType、Encoding、Data の 4 つのエレメントからなる。

```
<multiMediaDataEntity>
  <URI>URI for the data</URI>
  <MIMEType>MIME type of the data</MIMEType>
  <Encoding>encoding type</Encoding>
  <Data>encoded data string</Data>
</multiMediaDataEntity>
```

URI for the data にはそのデータを同定する URI が入る。MIME type of the data はデータのタイプ (画像・音声等のフォーマット) を MIME type の記述に法って文字列で記述す

る。*encoding type* は、none または base64 で、データがどの形式でエンコードされているかを記述する。³*encoded data string* には実際のデータを記述する。

- MultiMediaDataReferenceType : 画像、音声などのデータの参照を指定する際に用いる。具体的には URI と MIMEType からなる。

```
<multiMediaDataReference>
  <URI>URI for the data</URI>
  <MIMEType>MIME type of the data</MIMEType>
</multiMediaDataReference>
```

URI for the data にはそのデータを同定する URI が入る。*MIME type of the data* はデータのタイプ (画像・音声等のフォーマット) を MIME type の記述に法って文字列で記述する。

1.8.5 被覆データ型

ラスタデータやメッシュデータなどの被覆データ型については、GML 3.1 の RectifiedGridCoverageType の記述方法に準拠する。ただし、同 Coverage Model の記述方法にはいくつかの冗長性があるため、以下に示す形式のみを扱うものとする。

```
<gml:RectifiedGridCoverage>
  <gml:rectifiedGridDomain>
    <gml:RectifiedGrid dimension="coverage dimension">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>list of lower bounds of each axis</gml:low>
          <gml:high>list of upper bounds of each axis</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisName>name of the first axis</gml:axisName>
      <gml:axisName>name of the second axis</gml:axisName>
      ...
      <gml:axisName>name of the last axis</gml:axisName>
      <gml:origin>
        <gml:Point>
          <gml:coordinates>position of the origin</gml:coordinates>
        </gml:Point>
      </gml:origin>
      <gml:offsetVector>offset vector for the first axis</gml:offsetVector>
      <gml:offsetVector>offset vector for the second axis</gml:offsetVector>
      ...
      <gml:offsetVector>offset vector for the last axis</gml:offsetVector>
    </gml:RectifiedGrid>
  </gml:rectifiedGridDomain>
  <gml:rangeSet>
    data part
  </gml:rangeSet>
</gml:RectifiedGridCoverage>
```

coverage dimension は、グリッドの次元を表す。例えば、直線上に並んだデータ列の場合は 1 を、2 次元地図上にマップされる場合は 2 を、3 次元空間を埋める場合には 3 を指定する。

list of {lower/upper} bounds of each axis には、グリッドの各次元の下限・上限を示す。*name of the {first/second/.../last} axis* にはグリッドの次元の名前を記述する。*position of the origin* には原点の座標を指定する。*offset vector for the {first/second/.../last} axis* には、グリッドの各次元

³uuencode をサポートすべきが検討事項。Windows で対応できるのか？

の方向ベクトルを記述する⁴。よって、グリッドの各点の座標 p は、dimension=3、原点が o 、グリッドの各次元の方向ベクトルが u 、 v 、 w とすれば、

$$p_{ijk} = o + iu + jv + kw$$

となる。ただし、添字の i, j, k は *list of {lower/upper} bounds of each axis* に指定された範囲を定義域とする。

data part は、gml:File もしくは gml:DataBlock 形式のいずれかをとるものとする。

gml:File 形式は以下の通りである。⁵

```
<gml:File>
  <gml:rangeParameters>
    <gml:CompositeValue>
      <gml:valueComponent>
        <user-defined tag for the first parameter
          [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
      <gml:valueComponent>
        <user-defined tag for the second parameter
          [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
      ...
      <gml:valueComponent>
        <user-defined tag for the last parameter
          [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
    </gml:CompositeValue>
  </gml:rangeParameters>
  <gml:fileName>URI for the data file</gml:fileName>
  <gml:fileStructure>Record Interleaved</gml:fileStructure>
</gml:File>
```

gml:DataBlock 形式は以下の通りである。

```
<gml:DataBlock>
  <gml:rangeParameters>
    <gml:CompositeValue>
      <gml:valueComponent>
        <user-defined tag for the first parameter
          [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
      <gml:valueComponent>
        <user-defined tag for the second parameter
          [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
      ...
      <gml:valueComponent>
        <user-defined tag for the last parameter
          [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
    </gml:CompositeValue>
  </gml:rangeParameters>
  <gml:tupletList>list of parameters in CSV style</gml:tupletList>
</gml:DataBlock>
```

以下は、gml:DataBlock を用いた、平均気温・気圧のメッシュ上のデータの記述例である。

⁴方向ベクトルは gml:Pos 形式、すなわち空白区切りの数字列で表記される。

⁵gml:fileStructure については、GML 3.1 では Record Interleaved のみ正式サポートされることになっている。


```

<app:AverageTempPressuer>
  <gml:rectifiedGridDomain>
    <gml:RectifiedGrid dimension="2">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>1 1</gml:low>
          <gml:high>9 6</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisName>u</gml:axisName>
      <gml:axisName>v</gml:axisName>
      <gml:origin>
        <gml:Point>
          <gml:coordinates>2.0,1.0</gml:coordinates>
        </gml:Point>
      </gml:origin>
      <gml:offsetVector>1.0 0.2</gml:offsetVector>
      <gml:offsetVector>-0.2 1.0</gml:offsetVector>
    </gml:RectifiedGrid>
  </gml:rectifiedGridDomain>
  <gml:rangeSet>
    <gml:DataBlock>
      <gml:rangeParameters>
        <gml:CompositeValue>
          <gml:valueComponent>
            <app:Temperature uom="urn:x-si:v1999:uom:degreesC"/>
          </gml:valueComponent>
          <gml:valueComponent>
            <app:Pressuer uom="urn:x-si:v1999:uom:kPA"/>
          </gml:valueComponent>
        </gml:CompositeValue>
      </gml:rangeParameters>
      <gml:tupleList>2.0,101.2 5.0,101.3 7.0,101.4 11.0,101.5
                    13.0,101.6 17.0,101.7 19.0,101.7 23.0,101.8 29.0,101.9
                    ...
                    ...</gml:tupleList>
    </gml:DataBlock>
  </gml:rangeSet>
</app:AverageTempPressuer>

```

上記のデータ型は、gml:RectifiedGridCoverageType という名前で ‘common.xsd’ に登録されている。よって、例えば上記の app:AverageTempPressuer というデータ型を扱えるようにするためには、下のような定義を Register プロトコルにより登録すればよい。

```

<element name="AverageTempPressuer" type="gml:RectifiedGridCoverageType"/>

```

1.9 GML 記述上の制約

1.9.1 仕様上の制約

各プロトコルでは、座標、幾何形状、および時刻・時間情報を GML[Ope03] の GeometryType および TimeType により記述するが、冗長性の軽減等のため GML 3.1 のそれには対応しない。具体的には以下のような制約がある。

- GGGD で扱うことのできる GML の幾何形状は以下の通りである。
 - gml:Point (点)

- gml:LineString (曲線)
- gml:LinearRing (閉曲線)
- gml:Polygon (多角形)
- gml:GeometryCollection (幾何形状集合)
- GML では、gml:Polygon の外枠の閉曲線および内部の虫食い部分を表現する方法として、gml:outerBoundaryIs/gml:innerBoundaryIs タグによるものと、gml:exterior/gml:interior タグによるものがあるが、GGGD では gml:outerBoundaryIs/innerBoundaryIs のみを用いる。
- GML では座標の表記に gml:coordinates ・ gml:pos ・ gml:coord の 3 通りの表現方法があるが、GGGD ではこれらのうち gml:coordinates のみを扱う。
- gml:coordinates では小数点 (default で ‘.’)、1 つの点座標内の各次元の数値の区切り (default で ‘,’)、および座標間の区切り (default で半角空白) の文字を変更できるが、GGGD では default の文字を用いるものと仮定しており、これらの文字の変更はできない。
- TimePeriodType は GML では開始および終了時刻の表記にいくつかのバリエーションが存在するが、GGGD では以下の形式のみを扱う。

```
<timePeriodElement>
  <gml:beginPosition>a data of gml:TimePositionType</gml:beginPosition>
  <gml:endPosition>a data of gml:TimePositionType</gml:endPosition>
</timePeriodElement>
```

1.9.2 実装上の制約

現状の GGGD の実装では、以下のような制約がある。

- 時刻の表記において、ISO-8601 [Int04] ではタイムゾーンの指定および秒以下の表記を許しているが、GGGD ではこれらがあってはいけない。タイムゾーンは JST であると仮定する。

これらの制約は将来解除される予定である。

1.10 Filter 記述上の制約

GetFeature および Transaction の Update ・ Delete プロトコルでは、検索条件を OGC の Filter Encoding Implementation [Ope01] に従って記述するが、現状の GGGD の実装では同規格に完全には対応していない。具体的には以下のような制約がある。

- 空間オペレーション (Spatial Operators) では、図形の包含関係などを表すが、現状では外接長方形による同等の空間オペレーションで近似されている。このため、厳密な検索を行うためには、検索結果に対し再度幾何学計算を行う必要がある。
- WFS の Filter 定義では、複数回現れる element に対する条件の記述は定められていない。このような element に対する条件記述では以下のタイプの論理オペレータを追加する必要がある。

- for-all (\forall): 全ての element で条件成立
- exists (\exists): どれか 1 つの element で条件成立
- nth: n 番目の element で条件成立

以上の制約は将来解消される予定である。

第2章 ToDo

- transaction に mode attribute をつける。mode は append か override のどちらか。append の場合は、もとのデータに delete flag をつけ、新たに登録。override はもとのデータを直接書き換え。
- cleanup コマンドがいるのではない？ delete flag がついたものを全て消去する。
- ロボットによる画像撮影情報の表現。例えば以下のような形式。
 - 情報の種類:
 - * 撮影主体
 - * 撮影地点 (x,y,z)
 - * 撮影時刻 (t)
 - * 撮影比高 (h) : 地面あるいは床からの高さ
 - * 画角 : 水平方向、垂直方向
 - * 視野角 : 水平方向分、垂直方向分
 - * 対象領域 : 画像に写っている範囲を記述。無い場合は撮影地点・比高・画角・視野角から計算
 - * カメラの性能 : URI で指定する。URI と実際の性能のテーブルは別途用意
 - * 画像のタイプ : MIME type
 - * 画像データ : URI で指定する。URI と実際のデータの対応は別途用意
 - * 備考情報 :
 - XML 表現

```

<!-- 撮影情報など -->
<SensedImageInfo>
  <URI>URI-for-this-entry</URI>
  <Agent>name-of-device</Agent>
  <SensedLocation>
    sensed location point in GML
  </SensedLocation>
  <SensedTime>
    sensed time in GML
  </SensedTime>
  <SensedHight>sensed hight</SensedHight>
  <SensedAngle>
    <horizontal>sensed horizontal direction</horizontal>
    <vertical>sensed vertical direction</vertical>
  </SensedAngle>
  <ViewConeAngle>
    <horizontal>horizontal view angle</horizontal>
    <vertical>vertical view angle</vertical>
  </ViewConeAngle>
  <SensedArea>
    sensed area in GML
  </SensedArea>
  <SensorSpec>URI for a sensor spec</SensorSpec>
  <SensedData>
    <MIMEType>a MIME type</MIMEType>
    <URI>URI for data</URI>
  </SensedData>
</SensedImageInfo>
<!-- 実際のデータ -->
<SensedDataEntity>
  <URI>URI for data</URI>
  <MIMEType>a MIME type</MIMEType>
  <Encoding>plain, base64 or uuencode</Encoding>
  <Data>
    actual encoded data
  </Data>
</SensedDataEntity>

```

具体例：

```

<!-- 撮影情報など -->
<SensedImageInfo>
  <URI>URI-for-this-entry</URI>
  <Agent>ロボット 1-1</Agent>
  <SensedLocation>
    <gml:Point>
      <gml:coordinates>-100.0,10.5,2.0</gml:coordinates>
    </gml:Point>
  </SensedLocation>
  <SensedTime>
    <gml:timePosition>2004-10-17T01:23:00</gml:timePosition>
  </SensedTime>
  <SensedHight>2.5</SensedHight>
  <SensedAngle>
    <horizontal>60.0</horizontal>
    <vertical>-10.0</vertical>
  </SensedAngle>
  <ViewConeAngle>
    <horizontal>30.0</horizontal>
    <vertical>20.0</vertical>
  </ViewConeAngle>
  <SensedArea>
    <gml:Polygon>
      <gml:outerBoundaryIs>
        <gml:LinearRing>
          <gml:coordinates>
            -11767.0,-55247.0 -11767.0,-55256.0
            -11772.0,-55258.0 -11773.0,-55249.0
            -11767.0,-55247.0
          </gml:coordinates>
        </gml:LinearRing>
      </gml:outerBoundaryIs>
    </gml:Polygon>
  </SensedArea>
  <SensorSpec>URI-for-a-sensor-spec</SensorSpec>
  <SensedData>
    <MimeType>a-MIME-type</MimeType>
    <URI>URI-for-data</URI>
  </SensedData>
</SensedImageInfo>
<!-- 実際のデータ -->
<SensedDataEntity>
  <URI>URI-for-data</URI>
  <MimeType>a-MIME-type</MimeType>
  <Encoding>base64</Encoding>
  <Data>
    <!-- base64 でエンコードされたデータがここにくる。 -->
  </Data>
</SensedDataEntity>

```

- Coverage 関係。

- Coverage のための共通データ型を定義したが、正しいかどうかまだ検証していない。
- RawXMLData 型の実装。この部分は XML の解釈をせずにそのまま文字列として SQL に格納。
- offsetVector を正しく解釈すべきか？(現状は単なる string として解釈)

第3章 Changes

- [2005-01-11:Ver.-0.86:I.Noda]
 - Changes の形式を整理。
 - Coverage について、common.xsd を直すとともに、todo 追加。
- [2005-01-07:Ver.-0.87:I.Noda]
 - 被覆の形式について追加。
 - データ型の各記述を subsection に格上げ。
- [2004-11-30:Ver.-0.88:I.Noda]
 - この changes を逆順にソート
 - ロボットによる撮影画像の情報格納について、todo に記述追加。
- [2004-11-10:Ver.-0.89:I.Noda]
 - コードの実例のフォントサイズを小さいもので統一。
 - セクションわけの整理。
- [2004-11-09:Ver.-0.90:I.Noda]
 - システム名を XGD から GGGD に変更。
 - 制約について、仕様による制約と、実装による制約を分けて記述。
 - さらに、繰り返し現れる element について許容。(システムの実装)
- [2004-11-05:Ver.-0.91:I.Noda]
 - エンベロープを SOAP Envelope に準拠。
 - Transaction の Update と Delete に mode 属性を追加。(システムには未実装)
- [2004-10-26:Ver.-0.92:I.Noda]
 - 竹内さんの指摘に従い、WFS 仕様 への参照を増やす。
 - 応答メッセージの中で、SQL に依存した記述を修正。
- [2004-10-21:Ver.-0.93:I.Noda]
 - エンベロープの導入。(システムには未実装)
- [2004-10-21:Ver.-0.94:I.Noda]
 - latex エラーの fix。マクロの修正。
- [2004-10-19:Ver.-0.95:I.Noda]
 - ToDo の追加。目次と索引の追加。参考文献の追加。
- [2004-10-15:Ver.-0.96:I.Noda]
 - Multi-Media データの扱いの説明追加。
- [2004-10-?:Ver.-0.97:I.Noda]

- Multi-Media データの扱いについて規定。
- [2004-10-??:Ver.-0.98:I.Noda]
 - GetCapabilities、DescribeFeatureType プロトコルの拡張。
- [2004-09-04:Ver.-0.99:I.Noda]
 - 本文書作成

関連図書

- [Int04] International Organization for Standardization. *ISO 8601:2000 : Data elements and interchange formats – Information interchange – Representation of dates and times*, 2 edition, Jun. 2004. [http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=2% 6780](http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=2%206780).
- [Ope01] Open GIS Consortium, Inc. *Filter Encoding Implementation Specification (OGC 02-059)*, ver. 1.0.0 edition, Sep. 2001. <http://www.opengeospatial.org/docs/02-059.pdf>.
- [Ope02] Open GIS Consortium, Inc. *Web Feature Service Implementation Specification (OGC 02-058)*, ver. 1.0.0 edition, May. 2002. https://portal.opengeospatial.org/files/?artifact_id=7176.
- [Ope03] Open GIS Consortium, Inc. *OpenGIS Geography Markup Language (GML) Implementation Specification (OGC-02-023r4)*, ver. 3.00 edition, Jan. 2003. <http://www.opengis.org/docs/02-023r4.pdf>.
- [vdV03] Eric van der Vlist. *XML Schema*. O'REILLY, 3月 2003.

索引

- attribute, 20
- base type, 20
- base64, 23
- BoundaryBox, 9
- common.xsd, 21
- Delete, 5, 12, 15, 26
- DescribeFeatureType, 5, 17
- ElapsedTime, 9
- element, 8, 20
- Encoding, Data, 22
- FeatureCollection, 9
- Filter, 8, 26
- Filter Encoding Implementation, 26
- GeometryFeatureType, 22
- GeometryType, 25
- GetCapabilities, 5, 15, 16
- GetFeature, 5, 8, 26
- GGGD, 5
- GGGD Server, 5
- GML, 21, 25
- GML 3.1, 25
- gml:coord, 26
- gml:coordinates, 26
- gml:DataBlock, 24
- gml:exterior, 26
- gml:File, 24
- gml:GeometryCollection, 26
- gml:geometryProperty, 22
- gml:geometryPropertyType, 21, 22
- gml:innerBoundaryIs, 26
- gml:interior, 26
- gml:LinearRing, 26
- gml:LineString, 26
- gml:outerBoundaryIs, 26
- gml:Point, 25
- gml:Polygon, 26
- gml:pos, 26
- gml:TimePositionType, 21
- id, 22
- IDType, 22
- include, 6
- Insert, 5, 12
- ISO-8601, 21, 26
- MIME type, 22, 23
- MIMETYPE, 22, 23
- MultiMediaDataEntityType, 22
- MultiMediaDataReferenceType, 23
- NameType, 22
- none, 23
- OGC, 26
- OGC Filter Encoding Implementation Specification, 8
- RectifiedGridCoverageType, 23
- Register, 5, 6, 18
- restriction, 20
- RoadLink, 7, 9, 13, 18
- schema, 19
- Spacial Operators, 26
- TCP/IP ポート, 5

- TimeInstantType, 22
- TimePeriodType, 22, 26
- TimeType, 25
- Transaction, 5, 12, 26
- type, 20
- unsignedInt, 21
- unsignedLong, 21
- Update, 5, 12, 14, 26
- URI, 21–23
- uri, 6
- URIType, 22
- uuencode, 23
- Web Feature Service, 5, 9, 12, 14, 15, 17, 18
- WFS, 5, 9, 12, 14, 15, 17, 18
- XML, 5, 21
- XML Schema, 5–8, 19, 20
- xsd:attribute, 20
- xsd:float, 21
- xsd:integer, 21
- xsd:SqlBlob, 21
- xsd:SqlLongBlob, 21
- xsd:SqlMediumBlob, 21
- xsd:SqlTinyBlob, 21
- xsd:string, 21
- xsd:unsignedByte, 21
- 音声, 22
- 開始時刻, 22
- 外接長方形, 26
- 画像, 22
- 型, 5
- 幾何形状, 21, 25
- 幾何形状集合, 26
- 曲線, 26
- 空間オペレーション, 26
- 組み込み, 21
- 検索, 21
- 更新, 12
- サービス, 16
- 削除, 12
- 座標, 25
- 時間区間, 22
- 時間情報, 25
- 時間表現データ型, 22
- 時刻, 25
- 実数, 21
- 地物型, 22
- 終了時刻, 22
- 小数点, 26
- 整数, 21
- 属性, 20
- 外枠, 26
- 多角形, 26
- 追加記録, 12
- データエントリ, 5
- データ型, 16
- データの型, 8
- 点, 25
- 点時刻, 21, 22
- トップレベル, 8, 19, 20
- 内部, 26
- ノードリンク, 14
- 被覆データ型, 23
- 非負整数, 21
- プリミティブデータ型, 21
- 閉曲線, 26
- 包含関係, 26

マルチメディアデータ型, 22

虫食い部分, 26

メッシュデータ, 23

文字列, 21

文字列型の拡張, 22

ラスタデータ, 23

ロードリンク, 14, 15