GML Design for Moving Object Information of Probe Cars

Eunkyu Lee and Byung-Tae Jang
Telematics Research Division, ETRI
161 Gajeong-dong, Yuseong-gu, Daejeon 305-350, KOREA
{ekyulee, jbt}@etri.re.kr

Abstract: This paper refers and implements XML Web services, which supports networked interchanges of moving object information. XML Web services are expected to be fundamental building blocks in the move to distributed computing on the Internet. As an interchange format, GML encodes semantics, syntax, and schema of geospatial and geoprocessing-related information resources. It can give great benefits to utilize raw data easily as an XML encoding format does. This paper proposes a full coverage of interoperable location trajectory services consisting of 3 independent modules: a moving object database, a data processing server, and a web services interface module. For communications, SOAP protocols and WSDL documents are used, which can guarantee an interoperability of a system regardless of different platforms and service channels. This paper also designs a GML data format that represents location information of probe cars.

Keywords: Geography Information System (GIS), Geography Markup Language (GML), Moving Object (MO), Web Service.

1. Introduction

Intelligent Transportation System (ITS) technology has provided many governments and other organizations with an opportunity to introduce major improvements in traditional methods and procedures of providing services to citizens and customers. With traditional desktop-based ITS, it can be difficult to make your information available to all users. However, the recent XML Web services are making it possible to design and deploy very affordable and scalable applications using ITS visualization and analysis technologies. XML Web services expose useful functionality to web users through a standard web protocol such as Simple Object Access Protocol (SOAP), and provide a way to describe their interfaces in enough with Web Services Description Language (WSDL) and Universal Discovery Description and Integration (UDDI). In data presentation, Geography Markup Language (GML) becomes a new modeling language to encode semantics, syntax, and schema of geoprocessing-related information resources. It is an XML encoding for the transport and storage of geographic information.

This paper provides *a location trajectory service*. The proposed service system basically consists of three independent modules. A Moving Object (MO) database contains location information of probe cars and their history data. A data processing modules, provider and consumer, get location information from the database and handle them based on a few processing algorithms. A web services interface receives users' requests on web services architectures. This paper also designs a simple GML data format that simplifies location information of probe cars.

The rest of this paper is organized as follows. Section 2 and Section 3 review the current status and issues of related technologies, XML Web Services and GML. The next comes with the proposed schemes. It describes elements technologies and the architecture of the proposed GML Web services. Section 5 includes a service user taking advantage of the proposed moving object information services system. Finally, we conclude this paper in the last section.

2. XML Web Services

XML Web services are fundamental building blocks in the move to distributed computing on the Internet. Open standards and the focus on communication and collaboration among people and applications have created an environment where XML Web services are becoming the platform for application integration. Applications are constructed using multiple XML Web services from various sources that work together regardless of where they reside or how they were implemented. XML Web services expose useful functionality to Web users through a standard web protocol. In most cases, the protocol used is SOAP. XML Web services provide a way to describe their interfaces in enough detail to allow a user to build a client application to talk to them. This description is usually provided in an XML document called a WSDL document. XML Web services are registered so that potential users can find them easily. This is done with UDDI.

SOAP is a communication protocol for XML Web services. The SOAP specification defines the XML format for messages [5]. A simple SOAP message includes a well-formed XML fragment enclosed in a couple of SOAP elements. The SOAP specification also describes how to represent program data as XML and how to use SOAP to do Remote Procedure Calls. These optional parts of the specification are used to implement RPC-style applications, where a SOAP message containing a callable function and the parameters to pass to the function is sent from the client, and the server returns a message with results of the executed function.

WSDL describes a set of SOAP messages and how the messages are exchanged. It specifies what a request message must contain and what a response message will look like in unambiguous notation [8]. Because WSDL is an XML document, it is readable and editable in most cases and is generated and consumed by software. A notation that a WSDL file uses to describe message formats is based on a XML Schema standard. Therefore, WSDL is both programming-language neutral and standards-based, which makes it suitable for describing XML Web services interfaces that are accessible from a wide variety of platforms and programming languages.

UDDI is yellow pages of Web services. A UDDI directory entry describes a business and services it offers [6]. There are three parts to an entry in the UDDI directory. A white page describes the company offering the service: name, address, contacts, etc. A yellow page includes industrial categories of the services based on standard taxonomies. A green page describes interfaces to the service in enough detail for someone to write an application to use the Web services. The way services are defined is through a UDDI document called a Type Model or tModel. In many cases, the tModel contains a WSDL file that describes a SOAP interface to an XML Web services, but is flexible enough to describe almost any kind of services.

3. Geography Markup Language

A geographic feature, an abstraction of a real world phenomenon associated with a location relative to the Earth, has been a starting point for modeling geographic information. Recently, the *Geography Markup Language* becomes a new modeling language to encode semantics, syntax, and schema of geospatial and geoprocessing-related information resources [4]. GML is an XML encoding for the transport and storage of geographic information, including both geometry and properties of geographic features. GML utilizes the OpenGIS abstract specification geometry model [2]. Like any XML encodings, GML represents geographic information in text form. While a short while ago this might have been considered verboten in the world of spatial information systems, the idea is now gaining a lot of momentum. Text has a certain simplicity and visibility on its side. It is easy to inspect and change.

The GML approach is a great improvement over the historical reliance on simple GIF/JPG image maps for the following reasons: better quality maps, works on a browser, custom map styling, editable maps, more sophisticated linking capabilities, better query capability, control over content, service chaining, and so on. GML 3.0 released in 2003 provides a variety of kinds of objects for describing geography including features, coordinate reference systems, geometry, topology, time, units of measure, and generalized values. The Open Geospatial Consortium (OGC) defines a geographic feature as an abstraction of a real world phenomenon; it is a geographic feature if it is associated with a location relative to the Earth.

4. Proposed Scheme

This paper touches technologies described in previous sections to provide a location trajectory service, where data is encoded by GML format and SOAP communication protocols are used between a server and clients. Fig. 1 shows the overall architecture of the proposed service system.

In system design process, requirements that a server should support are summarized as follows. First of all, a service sequence should be sampled. A probe car connects to the server and requests to register a position data and related information of the specified MO to a database registry. When data is stored in database and a user requests MO information, then the server provides the information through web services interfaces. From the above scenario, protocol specifications and database structure should be considered. Protocols for data collection and data distribution should be distinguished. Especially when providing data to users, corresponding XML schemas and SOAP message formats should be supported.

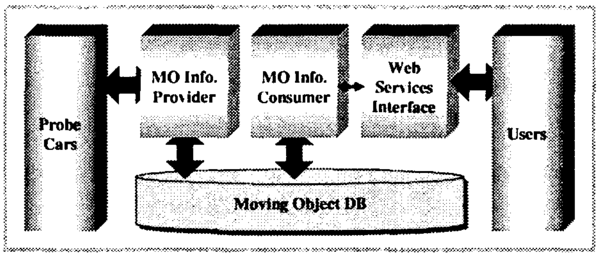

Fig. 1. Overall architecture of the proposed system.

1) MO Data Processing Modules

The MO data processing modules are directly connected with a database. A provider module communicates with a MO system administrating a target object. It receives location information from the MO periodically or randomly. Then it connects the database and stores the received data. A consumer module receives requests transmitted from a user through web services interfaces. Then it connects with the database, searches appropriated information, and then results in location data with additional information such as trajectory information of the requested moving object or its sets.

Table 1. Message format on storing MO information.

Length	Reference	Position			
		Longitude		Latitude	
		Length	Coordinates	Length	Coordinates

MO ID		Transmission Time						Direction	Velocity
Length	MO ID	Year	Month	Day	Hour	Min.	Sec.		

Table 1 shows a message format when storing MO information that is processed in the MO data provider module. The first *Length* is 5 byte-text string representing the whole size of the message. The *Reference* represents a type of referencing system of coordinates described in the message. The below parts represent information of the associated moving object such as its ID, directions, and velocity.

2) Moving Object Database (MODB)

The MODB stores position information of MOs transmitted from the MO data provider module. It also contains other related data such as timestamps, and IDs. It manages the MO information and outputs appropriate data to the MO data consumer module.

Table 2. MO and state table schema.

Field Name	Type	Key	Default
oid	char(20)	NULL	NULL
x	double	NULL	NULL
y	double	NULL	NULL
t	char(20)	NULL	NULL
dir	int	NULL	NULL
speed	double	NULL	NULL
coType	char(1)	NULL	NULL

Field Name	Type	Key	Default
ID	char(1)	NULL	NULL
MaxTuplesCount	int	NULL	NULL
TimetoDelete	int	NULL	NULL

3) Web Services Interface

In order to support communications, a web services interface module provides protocols and service description messages. Fig. 2 shows SOAP messages transferred between servers and users. The bold typed variables can be replaced by real values in actual environments. Fig. 3 shows a part of WSDL document describing operations of the service. It describes messages, port types, and bindings.

```
POST /GNSS/WMOS.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.4s.re.kr/GNSS/GetGNSS"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelo
pe/">
  <soap:Body>
    <GetGNSS xmlns="http://www.4s.re.kr/GNSS/">
      <regionID>string</regionID>
      <probeCarID>int</probeCarID>
    </GetGNSS>
  </soap:Body>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelo
pe/">
  <soap:Body>
    <GetGNSSResponse
xmlns="http://www.4s.re.kr/GNSS/">
      <GetGNSSResult>string</GetGNSSResult>
    </GetGNSSResponse>
  </soap:Body>
```

Fig. 2. SOAP messages: Request and Response.

```
<message name="GetGNSSSoapIn">
  <part name="parameters" element="s0:GetGNSS" />
</message>
<message name="GetGNSSSoapOut">
  <part name="parameters" ele-
ment="s0:GetGNSSResponse" />
</message>
<portType name="WMOSSoap">
  <operation name="GetGNSS">
    <input message="s0:GetGNSSSoapIn" />
    <output message="s0:GetGNSSSoapOut" />
  </operation>
</portType>
<binding name="WMOSSoap" type="s0:WMOSSoap">
  <soap:binding trans-
port="http://schemas.xmlsoap.org/soap/http"
style="document" />
  <operation name="GetGNSS">
    <soap:operation soapAc-
tion="http://www.4s.re.kr/GNSS/GetGNSS"
style="document" />
    <input> <soap:body use="literal" />  </input>
    <output> <soap:body use="literal"
/>  </output>
  </operation>
```

Fig. 3. WSDL contents: description for the moving object information services.

5) GML Format for MO Information

This paper designs a simple GML format that constructs a type of MO information. Fig. 4 shows an example of the GML-encoded MO information. It simplifies the contained data: it has region ID, type of coordinates system, information of probe car including a set of coordinates, etc. The example shows a region in Seoul named 'Kangnam-gu' and the coordinates indicating a specific point with its ID of moving object.

```
<?xml version="1.0" encoding="euc-kr" ?>
<wts:FeatureCollection
xmlns:wts="http://www.opengis.net/gnss_etri"
xmlns:gml="http://www.opengis.net/gml" ... ...>
  <gml:description>XML Web Service for GNSS by
ETRI.</gml:description>
  <gml:boundedBy>
    <gml:Box srsName="WGS-84">
      <gml:coordinates>127.045014,37.494006
127.064189,37.499264</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <!-- = Kangnam-gu Seoul = -->
    <Probe rid="KN">
      <ProbeMember>
        <gml:Point gid="KNP1">
          <gml:coordinates>127.061597,37.498536
127.062458,37.498605</gml:coordinates>
        </gml:Point>
      </ProbeMember>
    </Probe>
  </gml:featureMember>
</wts:FeatureCollection>
```


Fig. 4. A simple GML message formation.

5. A Simple User of Moving Objects Information Services

This section describes a user module taking advantage of the proposed a location trajectory service system. Fig. 5 shows a traffic monitoring system that calls the proposed web services operations, receives location data of probe cars, and presents the traffic status of sections in a specific region. It also supports basic map application tools such as enlargement and POI notifications.

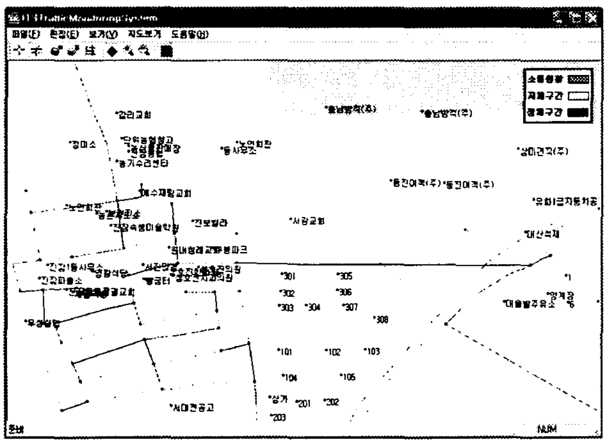

Fig. 1. A simple user: traffic monitoring system.

6. Conclusion and Further Works

Through this paper, we have proposed and implemented a MO information service system on web services architecture. It is an independent architecture for a complete service: it supports a moving object database, data processing server, a communication interface for XML web services, and a standard data format such as GML. The moving object database module contains location information for advanced traffic information services and trajectory information updated by real time. The data processing modules connect with the database and store or provide appropriate MO information. The web services interface module receives users' requests on web services architectures. For communications, SOAP protocols are used, which is able to guarantee an interoperability of a system regardless of different platforms and service channels. The research results are expected to be fundamental models onto a design of system architecture for interoperable information services.

We have a plan to implement the latest version 3.0 of the GML specification and to develop dedicated codec for the future work. Our future research also includes all the roles in the Web services architecture: a service provider, a service broker, and a service consumer.

References

[1] S. Jerome and S. Charles, 2003. OWS 1.2 SOAP Experiment Report, Open Geospatial Consortium.

[2] B. Kurt, 2003. OpenGIS Reference Model, Open Geospatial Consortium.

[3] J. Lieberman, 2003. OWS1 Web Service Architecture v0.3, Open Geospatial Consortium.

[4] C. Simon, D. Paul, L. Ron, P. Clemens, and W. Arliss, 2003. OpenGIS Geography Markup Language (GML) Implementation Specification version 3.0, Available at: Open Geospatial Consortium http://www.opengis.org/docs/..

[5] Simple Object Access Protocol (SOAP), 2003. World Wide Web Consortium (W3C). Available at: http://www.w3c.org/TR/soap/.

[6] Universal Discovery Description and Integration (UDDI). Available at: http://www.uddi.org/.

[7] WSA, Web Service Architecture, World Wide Web Consortium (W3C). Available at: http://www.w3c.org/TR/ws-arch/, 2004.

[8] Web Service Description Language (WSDL), 2004. World Wide Web Consortium (W3C). Available at: http://www.w3c.org/TR/wsdl20/.

[9] W. Michael and D. Matt, 2002. Integrating Spatio-thematic Information, *Proc. GIScience'2002*, LNCS 2478, Springer-Verlag, Berlin Heidelberg New York, pp. 346-361.

[10] B. P. Buttenfield, 2002. Transmitting Vector Geospatial Data across the Internet, *Proc. GIScience'2002*, LNCS 2478, Springer-Verlag, Berlin Heidelberg New York, pp. 51-64.

[11] E. Lee, M. Kim, B. Oh, and B. Jang, 2003. System Comparisons for GML Services, *Proc. ACRS' 2003 ISRS'*, pp. 1313-1315.

[12] B. Buttenfield, 1999. Sharing Vector Geospatial Data on the Internet, *Proc. the international Cartographic Association*, pp. 35-44.