Vanden Berghe E., M. Brown, M.J. Costello, C. Heip, S. Levitus and P. Pissierssens (Eds). 2004. p. 187-193
Proceedings of 'The Colour of Ocean Data' Symposium, Brussels, 25-27 November 2002
IOC Workshop Report 188 (UNESCO, Paris). x + 308 pp.
– also published as VLIZ Special Publication 16

Use of XML technology in the Baltic Sea fisheries database

Peter Sandbeck, Brian James Cowan and Teunis Jansen

Danish Institute for Fisheries Research
Jægersborgvej 64-66, DK-2800 Lyngby, Denmark
E-mail: pes@dfu.min.dk

Abstract

Aggregated commercial catch sampling data is used to estimate the fish stocks of cod (*Gadus morhua)* in the Baltic Sea. The present system for data validation, uploading and downloading of data via the Internet is presented. XML technology is used to check the files before uploading. The way to configure different kinds of data validation and data cardinality in an XML schema file is discussed. XML schema files can be used to check for cardinality between record types, mandatory data, value ranges, string patterns and enumeration data consistency. The column separated data exchange format used in many present marine data exchange systems i.e. by the International Council for the Exploration of the Sea (ICES) is compared to the XML format and the benefits of using XML technology are discussed. Finally a model for a future system using XML technology and Webservices in a distributed Internet based datawarehouse solution is presented and discussed.

Keywords: XML schema; Baltcom; Validation; Webservice; Datawarehouse.

Introduction

In three projects financed by The European Commission, Directorate-General FISH, fisheries discard data from the fishery in the Baltic Sea in the period 1995-2001 has been sampled (Degel and Jansen, in prep.). All countries around the Baltic Sea participated in the projects. The overall aim of the projects was to improve the quality of the stock assessment of Baltic cod. In the second project a simple database system was developed, and in the third project a web based datawarehouse was developed in order to make the data handling much easier and the data validation more consistent. The datawarehouse named Baltcom was placed at the Danish Institute for Fisheries Research on a Windows 2000 server with Internet Information Server and Microsoft SQL Server software. The web based user interface was developed with .Net technology using Visual Basic and Active Server Pages (ASPX).

The data exchange format used during the projects was a column separated format with nested record types (one to many relations) as used by ICES. The participating countries all have programs, which request data from the national databases and write the data into files with the ICES format. In order to check the data in a consistent way before they are archived in the Baltcom database, it was decided to use XML technology to configure the data validation rules, thus the configuration was done in an XML schema file. XML is an international standard

developed by the W3 consortium (W3C). A program component which converts a file in ICES format into XML format was developed. By converting the ICES format into XML format we got the possibility of using a standard XML application programming interface (API), which is part of most modern programming software, to check and upload the data to the Baltcom datawarehouse. The W3C is still developing the XML standards and the XML schema semantics has been improved since the first version of the Baltcom datawarehouse application. The same is true for the standard XML API used in Visual Basic. Therefore a new version of the Baltcom datawarehouse application is under construction using the new version of XML schema semantics. This paper focusses on using the new version of XML schema technology in order to configure and perform various data validation checks in the Baltcom datawarehouse application.

Baltcom data exchange

The Baltcom data upload system is shown in Fig. 1. The data is archived in a common datawarehouse, which is accessed via the Internet. The program which uploads and downloads the data runs in an Internet browser. The national database manager makes a zipped file in the agreed data exchange format, which is an ICES format type. This file is then uploaded via the Internet based interface program. The upload program first converts the column based exchange format into an XML format file according to a predefined XML schema file. The data is then validated using the XML schema validation rules. If the data is erroneous the data is not uploaded, otherwise the data is archived in the Baltcom datawarehouse.

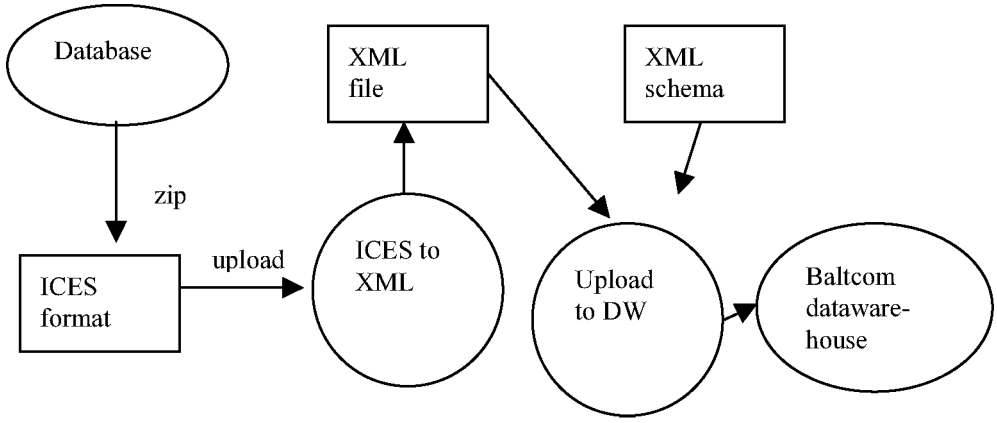

Fig. 1. The Baltcom data upload system.

The URL to the Baltcom system is http://www.Baltcom.org. A password to the system can be requested by mailing the administrator: administrator@baltcom.org.

ICES format to XML format conversion

The commercial catch sampling data used in Baltcom includes 4 different record types (EU-study 98/024). There is a one-to-many relation between record type 1 (haul information) and record type 2 (length frequency data). Record type 3 (sex-maturity-age-length keys) is related to

a journey not to a specific station and therefore it is not directly related to record type 1. Record type 4 (extended gear information) is not used in the present Baltcom datawarehouse. Each record type consists of a set of data fields. Each data field has a defined data type, and it might have a range of valid values, a specific char pattern or a set of valid enumerations. A part of the record type 1 data format specification is shown in Table I.

Table I. Part of an exchange format specification (from EU-study 98/024). All elements are mandatory. Type given as length (first digit) and alphanumeric/numeric (sec. digit)

Specifications for record type 1 (Haul information)				
Position	name	Type	Range	Comments
1 - 2	Record type	2A		Fixed value HH
3 - 5	Country	3A	See appendix I	ICES alpha code
6 - 7	Year	2N	00 to 99	
8 - 11	Journey	4N	1 to 9999	National coding system
12 - 14	Station No.	3N	1 to 999, 0	Seq. numb. by journey
15 - 16	Month	2N	1 to 12	
17 - 18	Day	2N	1 to 28/29/30/31	
19	Sampling type	1A	H, S	Harbour or sea sampling
20 - 22	Vessel length	3N	1 to 999	Overall length in m

In the XML format the haul information record type (record type 1) has been split into two record types, journey and station. In this way record type 3, the sex-maturity-age-length keys (SMALK) can be related to the journey record. Thus the structure of the data in the XML file is this:

```
<Journey>
        <Station>
                <Length frequency></Length frequency >
        </Station>
        <SMALK></SMALK>
</Journey>
```

There is a one-to-many relation between Journey and Station, between Station and Length Frequency, and between Journey and SMALK. These relationships are reflected in the XML schema file and hence in the XML data files. The ICES format files are converted to this XML format and then validated using the XML schema.

The advantages of using XML format for data exchange instead of column based formats are:

- data in XML format is more readable;
- data validation can be configured in an XML schema file;
- the format can easily be changed;

- uploading, downloading and data validation procedures can be programmed using the built in standard XML API.

XML schema validation

The ICES format specifications have all the information needed to build an XML schema file with data validation configuration. Each record type and each field type is defined as either a complex type or a simple type. The schema is defined in a top down manner starting with the topmost record types in the record tree and ending with the definitions of all the single field types. In XML terminology a schema file is denoted an XML namespace. Namespaces can be global like the main W3C XML namespace or private like the present Baltcom schema. The Baltcom schema could be made global by placing it on a website. The references to the XML namespaces used in an XML file are placed at the top of each XML file.

The following shows some examples of how different kinds of data validation are defined in the Baltcom XML schema file.

Cardinality

The attributes "minOccurs" and "maxOccurs" are used to define the cardinality between records. The default values are 1 for minOccurs and 1 for maxOccurs. In the example below, a journey record may hold from 1 to an infinite number of "Station" records and zero or one "SMALK" record.

```
<xsd:complexType name="JType">
        <xsd:sequence>
                <xsd:element name="DateStart" type="xsd:date"/>
                <xsd:element name="DateEnd" type="xsd:date"/>
<xsd:element name="Station" type="SType" maxOccurs="unbounded"/>
<xsd:element name="SMALKs" type="SMALKsType" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="Country" type="tCountry" use="required"/>
        <xsd:attribute name="Journey" type="tJourney" use="required"/>
        <xsd:attribute name="Year" type="tYear" use="required"/>
</xsd:complexType>
```

Primary key fields

The fields which make up the primary key for a record type are defined as attributes rather than normal fields (see example above). In this way the primary key data in the XML file will be shown in the header target of each record. The "use=required" attribute makes the data mandatory.

Range check

Range checks are defined with the "minInclusive" and "maxInclusive" attributes.

```
<xsd:simpleType name="tYear">
        <xsd:restriction base="xsd:int">
```

```
                <xsd:minInclusive value="1900"/>
                <xsd:maxInclusive value="3000"/>
        </xsd:restriction>
</xsd:simpleType>
```

String pattern

String patterns can be defined with the "pattern" attribute.

```
<xsd:simpleType name="tRectangle">
        <xsd:restriction base="xsd:string">
                <xsd:pattern value="[0-9]{2}[A-Z]{1}[0-9]{1}"/>
        </xsd:restriction>
</xsd:simpleType>
```

In this example a geographical rectangle is defined as two digits followed by one letter followed by one digit.

Enumeration

Some data may only have a defined set of values. These values are defined as enumerations.

```
<xsd:simpleType name="tSamplingType">
        <xsd:restriction base="xsd:string">
                <xsd:enumeration value="H"/>
                <xsd:enumeration value="S"/>
                <xsd:enumeration value="M"/>
        </xsd:restriction>
</xsd:simpleType>
```

What is missing is the possibility to check for data dependencies between data fields. For example in the journey record, it is not possible to check that the end date is bigger than or the same as the start date. This feature will properly be included in future XML schema semantics. However today this kind of data validation has to be done by the upload program in a traditional way.

Distributed internet based databases

The present Baltcom datawarehouse solution operates with one website and one datawarehouse, which all partners have to use. However new Internet technologies make it possible to build more advanced database solutions. A proposal for a distributed database system is presented in Fig. 3.

Here each institution has a database with a structure, which is different from the other institutions databases, which is usually the case. When a common data exchange format is defined in an XML schema, each institution builds a datawarehouse based on that data structure

and loads the data from the database into the datawarehouse. Thus the datawarehouses at all the institutions participating in the project have the same structure, and data can easily be transferred to another datawarehouse in XML files. The request for data is done directly by calling a webservice at the source website, which makes the data request, sends the data back to the calling webservice, which validates and uploads the data in the target datawarehouse. A webservice is a program on a website, which can be called from another program at another website. There are other technologies, which can do the same i.e. Corba and BizTalk technology.

The advantages of this solution are that data can be pulled to the requesting institution at any time. When common data have to be used the data access is fast because data are stored locally. Finally data at each site can automatically be updated when they are updated at the source database. Thus this kind of solution will probably be seen more in future projects.

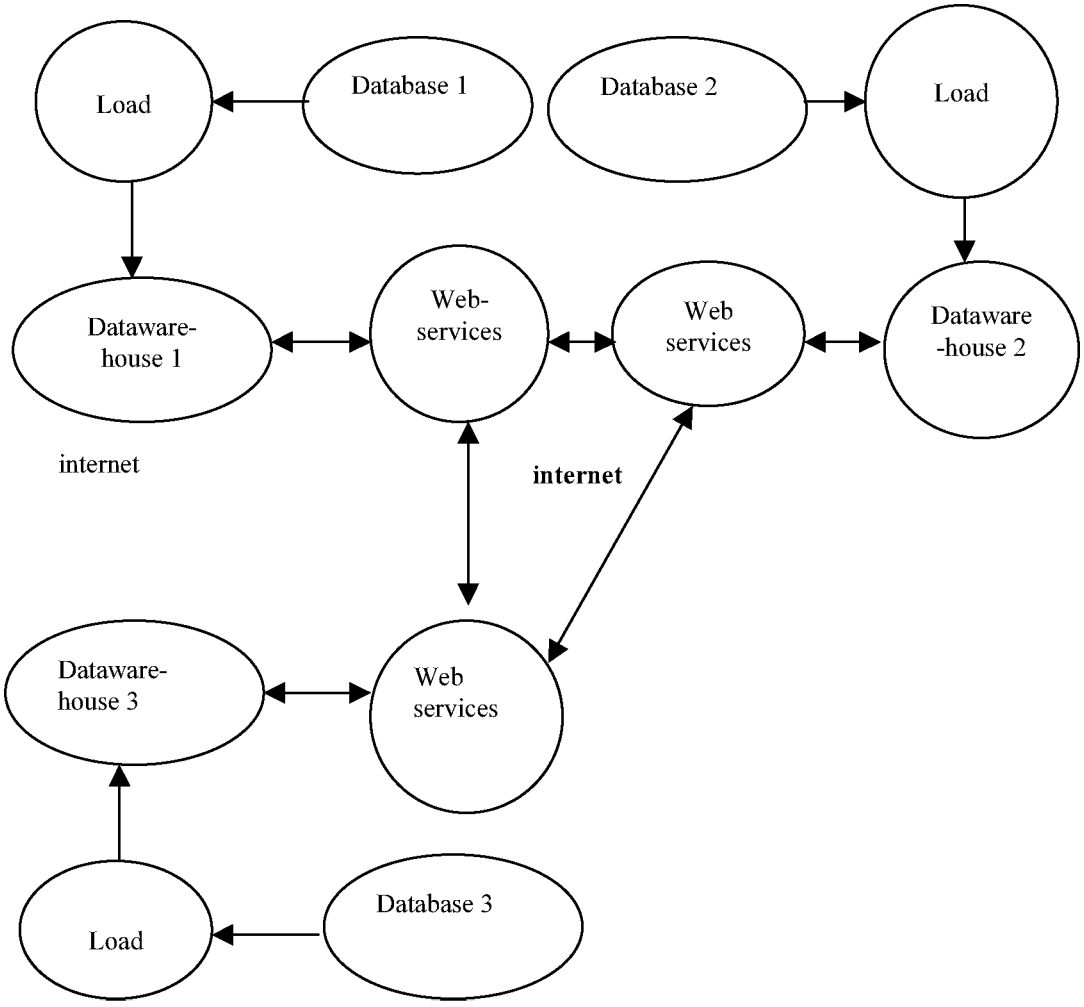

Fig. 2. Distributed internet based database system.

Conclusion

In the Baltic Sea Fisheries database XML technology has been used for data exchange and data validation. The project has revealed some general advantages of using XML for these purposes. Column based data exchange formats like ICES formats can easily be converted into XML formats with nested record types. Data in XML format is more readable and the files can be handled directly by most modern application programming interfaces.

The XML data exchange format for a specific project can be defined in an XML schema, thus ensuring the consistency of the XML exchange format used. The XML schema can also be used to configure the validation of data. Several kinds of checks can be performed like cardinality between record types, mandatory data, value ranges, string patterns and enumerations. However some kinds of data validation like comparisons of data values between fields can still not be performed.

Modern Internet technologies use XML as a standard data interface and configuration format. XML is an integrated part of new technologies like .Net, webservices and others. With these technologies one can build advanced solutions like distributed database systems where data is pulled from a source database on one website to a target database on another website by an application.

References

Degel H. and T. Jansen. (in prep.). BaltCom database. An internet based data warehouse for fisheries data in Baltic fisheries. ICES Working Paper. Baltic Fisheries Assessment Working Group.

EU-study 98/024. EU-study project No. 98/024. Version 7 of 14/6-01. Livingston, David: Essential XML for web professionals, Prentice Hall 2001 - W3C: http://www.w3c.org