Interfacing Environmental Simulation Models with Databases using XML

T. Kokkonen, A. Jolma and H. Koivusalo

Water Resources Laboratory, Helsinki University of Technology, P.O.Box 5200, FIN-02015 HUT, Finland (tkokko@water.hut.fi)

Abstract: Typically in environmental management tasks one needs to examine and explore data from several sources, use simulation models, develop scenarios, assess impacts, and provide support for decision makers. Here we consider the XML (eXtensible Markup Language) standard in developing information transfer techniques between databases and simulation models. Suitability of XML as the agreed data transfer format is studied in a sample application, where two snow models of different complexity are linked with input data extracted from a relational database. The simple case study demonstrated that with free and easily accessible tools it was relatively straightforward to develop an XML interface between a meteorological data set and simulation models. We believe that XML is worth exploring on the track towards an open modelling framework. Such a framework would allow models developed by various expert groups to connect easily to a common information system.

Keywords: XML; Database; Information transfer; Modelling

1. INTRODUCTION

Environmental problems are typically of a multidisciplinary nature, and may affect broad regions, requiring the analyst to examine data from several sources and to apply computation and assessment tools provided by various expert groups. As pointed out by Huang and Xia [2001], difficulty in linking data sets and analysis tools is one of the barriers to be overcome in developing integrated water resources management techniques.

Consider assessing deterioration of a river habitat in response to processes occurring in the upstream part of the catchment. The analyst needs to take into account interactions between soil-vegetation-atmosphere processes and routing of water through the stream network, chemical processes affecting the water quality, and biological processes controlling the well-being of the habitat. Simulation tools and data may be available to address isolated parts of the above problem, but often they are scattered among numerous sources and exist in heterogeneous formats. Development of modelling tools and acquisition of data sets have often concentrated on relatively specific problems leading to troublesome model and data transferability. Also, the physical boundary of a catchment rarely coincides with political regions, which leads to variable documentation formats, and creates the need for methods catering for communication between many administrative entities such as provinces and countries [Voinov and Costanza, 1999]. Clearly, agreement of techniques on how to transfer information between data sources and models is one of the key issues in integrated management of complex environmental problems.

Integration of simulation models has been implemented in many environmental assessment studies. Aspinall and Pearson [2000] and Huang et al. [1999] developed integrated catchment assessment systems around specific GIS software packages. Booty et al. [2001] describe an environmental decision support system which runs on Windows platform, integrates data from multiple sources, and provides a selection of decision aiding tools. The modular design of their system allows for flexibility in adding components to the system to meet the demands of a particular application. Models can be incorporated into the system as executable binary codes by supplying an appropriate interface for the model input and output, or through conversion of the model source code to a programming language supported by the system. Cate et al. [1998] advocate an open modelling system which they define to be a system where software components running on any kind of hardware are easily combined. Ideally, the user can select the best combination of available modelling components for a particular problem at hand. For such a modelling framework to work, the semantics of the model input and output needs to be specified.

In recent years the accelerating rate at which information is transferred via the Internet has challenged service providers to adopt compatible techniques for information transfer. Such techniques need to have a widely accepted, standardised form, they must be universally accessible, and preferably they should be available cost-free. Compatibility and reliability of data transfer is of crucial importance for business applications servicing a large number of clients and requiring access to extensive data resources. Example applications include web-based shopping and banking services, flight reservation services, and browsing of library databases. The World Wide Web Consortium (W3C, http://www.w3.org) was established in 1994 to develop, accept and maintain interoperable technologies (specifications, guidelines, software, and tools) to service the Internet community. Existing, standardised techniques for information transfer provide an attractive alternative in environmental studies to link models and data sets in a compatible and readily comprehensible way. The eXtensible Markup Language (XML), which has established itself as the universal format for structured documents and data on the Web, provides one way to facilitate communication between presentation, application (model) and database components.

The aim of this study is to explore how one can move towards an open modelling system by adopting an agreed method that enables a simulation model to document and communicate its input requirements, and receive the input from the information system. Suitability of XML as the agreed data transfer format is studied in a sample application, where two snow models of different complexity are linked with input data extracted from a database.

2. METHODS

2. 1 XML (eXtensible Markup Language)

Development of XML started in 1996 and it was adopted as a W3C standard in February 1998. Although XML as its own standard is fairly new, the methodology has a solid background in Standard Generalized Markup Language (SGML), which was developed in the early 1980s, became an ISO standard in 1986, and has been widely used for large documentation projects. The designers of XML adopted the best parts of SGML, guided by the experience gained with development of HyperText Markup Language (HTML), and produced a markup language whose performance matches that of SGML, but is more regular and simpler to use.

XML is a text-based markup language for describing what the data is, rather than just specifying how to display it, as is the case with HTML. Similar to HTML, markup is composed of a set of tags i.e. identifiers enclosed in angle brackets. The tag, its matching end tag, and the parsed character data (PCDATA) between the tags form an element of the XML data. An element can contain child elements allowing XML to represent hierarchical data structures. Additional information can be provided in an attribute, which is a qualifier within a tag. Figure 4 shows an XML document containing both element and attribute type of data. A Document Type Definition (DTD) can be used to define the structure of an XML document. A DTD lists the set of legal element names, specifies the hierarchy of the elements, and gives a listing of allowed attribute names (Figure 2). With a DTD different parties can agree on a format of the XML document for interchanging data. A validating parser can verify that all XML documents received conform to the structure defined in the DTD.

The following three features of XML have contributed to its increasing acceptance in web-based applications requiring standard data transfer formats. Firstly, the markup tags identify the information and break up the data into parts, and thus the same XML data document can be fed to different applications, which then can extract the relevant parts of the data from the document for further processing. Secondly, as a W3C technology XML is license-free and well-supported. There are freeware tools readily available, and a large community of developers can provide assistance for XML users over the Internet. And thirdly, as a plain text markup language XML is platform-independent meaning that application development is not tied to certain hard- and software vendors.

2.2 XML – DBMS (DataBase Management System)

XML-DBMS [Bourret, 2001] is a middleware for transferring data between XML documents and relational databases. It views the XML document as a hierarchical tree of data-specific objects and links these objects to a relational database using an XML-based mapping language. In this view, elements generally correspond to classes, and attributes and PCDATA correspond to properties.

Child elements are viewed as pointed-to classes, i.e. an interclass relationship exists between the classes corresponding to parent and child elements. XML-DBMS preserves the hierarchical structure of data in generating an XML document, and if requested, it also preserves the order in which the children elements at a given level in the hierarchy appear.

3. SAMPLE APPLICATION

Here an application of XML in linking simulation models and data is explored in a case study involving two snowmelt models. The models have different process descriptions resulting in different data requirements. The simpler of the models is based on the degree-day concept [e.g. Vehviläinen, 1992] and it needs only time-series of daily precipitation and air temperature as input data. The other snow model [Koivusalo et al., 2001] is based on the energy balance approach and is much more demanding in input data requirements. The energy balance model is driven by time series of precipitation, air temperature, relative humidity, wind speed, and downward short- and long-wave radiation. In addition to the data input, both models require parameter values to be prescribed and given as parameter input to the models prior to simulation. The degree-day model yields water equivalent and liquid water content of snow as simulation output, and the energy balance model produces both snow mass and energy state variables as simulation output. In this application the focus is on data transfer interfaces, and the models are "empty shells" [Argent and Grayson, 2001] not containing any computation procedures. The meteorological data were from two micrometeorological stations in Siuntio and Kirkkonummi, both of them located in southern Finland within 50 kilometres from Helsinki. The meteorological time-series and additional site and data specific information are stored in a PostgreSQL (http://www.postgresql.org) relational database.

Figure 1 shows an illustration of the linkage between the database and the snow models as implemented in the sample application. All input to the models and output from the models are directed through an XML interface. First the user decides which model is to be run. Each model comes with a DTD file, which specifies the required data input and its structure (Figure 2). This DTD file, along with user supplied information on what the requested variables are called in the database (DB Schema), are used to generate an XML-DBMS map file that describes the linkage between an XML document and the database. Figure 3 shows the map file for the degree-day model; the map file for the energy balance model (not shown) has a similar structure but is longer due to the greater number of input variables. The user selects the site and the modelling period and the XML data input file is created. The XML data file contains the relevant time-series required to drive the selected model, and site-specific metadata such as site name and units of the input variables (Figure 4). A separate XML-file is created for user specified model parameters. An XML parser embedded in the simulation model extracts from the input files information required in the model execution. In the current study Xerces Java Parser freely available from http://xml.apache.org is used. A model produces its output as an XML document whose contents can be stored in the database through the XML-DBMS interface.

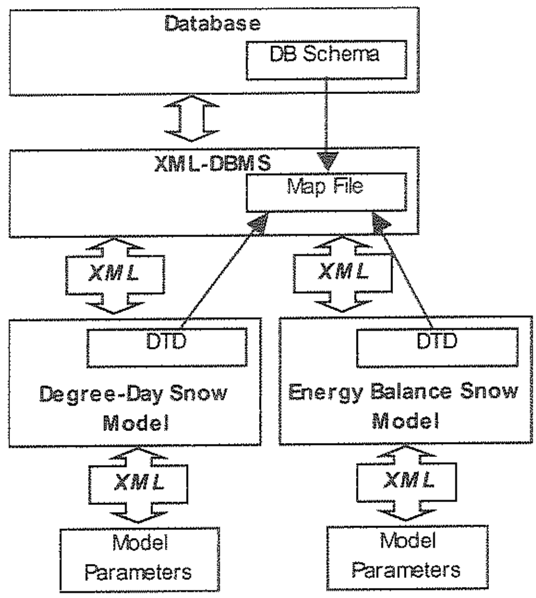

Figure 1. Schematic representation of the linkage between database and snow models.

4. DISCUSSION AND CONCLUSIONS

The simple case study demonstrated that with free and easily accessible tools it was relatively straightforward to develop an XML interface between a meteorological data set and environmental simulation models. The model input data stored in a relational database can be transferred to an XML document, and simulation output formulated as an XML document can be inserted into the database, through an XML-DBMS middleware. The linkage between an XML document structure and the database structure is described in a map file, which is combined from the information defining model input requirements and the database schema. A model capable of interpreting XML formatted input can be linked to the data source. From the data provider's point of view, there is no need to take into account differing model input data formats, but it suffices to receive a list of variables required for the modelling exercise. The modeller defines these requirements in a DTD file.

Clearly, standardised information transfer techniques only become warranted when a large number of data providers and modellers are involved. The strength of XML as a data transfer format lies in large-scale applications comprising several parties and extensive data resources. As an existing, platform-neutral, and well-supported data documentation format it is readily adoptable and thus offers a promising basis for compatible communication between environmental information systems and simulation tools. As an example of XML support, there exist freeware tools developed for validating the structure and authenticity of XML data.

a)

```
<!ELEMENT Site (Record+,AirTempUnit,
               PrecipUnit,SiteName)>
<!ELEMENT Record (AirTemp,Precip,Date)>
<!ELEMENT AirTemp (#PCDATA)>
<!ELEMENT Precip (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT AirTempUnit (#PCDATA)>
<!ELEMENT PrecipUnit (#PCDATA)>
<!ELEMENT SiteName (#PCDATA)>
<!ATTLIST Site SiteCode CDATA #REQUIRED>
<!ATTLIST Record RecordNumber CDATA #REQUIRED>
```

b)

```
<!ELEMENT Site (Record+,AirTempUnit,PrecipUnit,
               SWDownUnit,LWDownUnit,
               WindSpeedUnit,RHUnit,SiteName)>
<!ELEMENT Record (AirTemp,Precip,SWDown,
                 LWDown,WindSpeed,RH,Date)>
<!ELEMENT AirTemp (#PCDATA)>
<!ELEMENT Precip (#PCDATA)>
<!ELEMENT SWDown (#PCDATA)>
<!ELEMENT LWDown (#PCDATA)>
<!ELEMENT WindSpeed (#PCDATA)>
<!ELEMENT RH (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT AirTempUnit (#PCDATA)>
<!ELEMENT PrecipUnit (#PCDATA)>
<!ELEMENT SWDownUnit (#PCDATA)>
<!ELEMENT LWDownUnit (#PCDATA)>
<!ELEMENT WindSpeedUnit (#PCDATA)>
<!ELEMENT RHUnit (#PCDATA)>
<!ELEMENT SiteName (#PCDATA)>
<!ATTLIST Site SiteCode CDATA #REQUIRED>
<!ATTLIST Record RecordNumber CDATA #REQUIRED>
```

Figure 2. DTD files specifying data input requirements for: (a) degree-day; and (b) energy balance snow models.

```
<?xml version="1.0" ?>
<!DOCTYPE XMLToDBMS SYSTEM "../xmldbms.dtd">
<XMLToDBMS Version="1.0">
<Maps>
 <ClassMap>
   <ElementType Name="Site"/>
   <ToRootTable>
    <Table Name="sites"/>
   </ToRootTable>
   <PropertyMap>
    <Attribute Name="SiteCode"/>
    <ToColumn>
      <Column Name="site_id"/>
    </ToColumn>
   </PropertyMap>
   <PropertyMap>
    <ElementType Name="SiteName"/>
    <ToColumn>
      <Column Name="site_name"/>
    </ToColumn>
   </PropertyMap>
   <PropertyMap>
    <ElementType Name="PrecipUnit"/>
    <ToColumn>
      <Column Name="precip_unit"/>
    </ToColumn>
   </PropertyMap>
   ...
   <RelatedClass KeyInParentTable="Candidate">
    <ElementType Name="Record"/>
    <CandidateKey Generate="No">
      <Column Name="site_id"/>
    </CandidateKey>
    <ForeignKey>
      <Column Name="site_id"/>
    </ForeignKey>
    <OrderColumn Name="rec_order" Generate="No"/>
   </RelatedClass>
 </ClassMap>
 <ClassMap>
   <ElementType Name="Record"/>
   <ToClassTable>
    <Table Name="siuntio_series"/>
   </ToClassTable>
   <PropertyMap>
    <Attribute Name="RecordNumber"/>
    <ToColumn>
      <Column Name="rec_order"/>
    </ToColumn>
   </PropertyMap>
   <PropertyMap>
    <ElementType Name="Date"/>
    <ToColumn>
      <Column Name="date"/>
    </ToColumn>
   </PropertyMap>
   <PropertyMap>
    <ElementType Name="Precip"/>
    <ToColumn>
      <Column Name="precip"/>
    </ToColumn>
   </PropertyMap>
   ...
 </ClassMap>
</Maps>
</XMLToDBMS>
```

Figure 3. A map file for linking database structure with XML data input file structure (degree-day snow model).

a)
```
<?xml version="1.0" encoding="UTF-8"?>
    <Site SiteCode="1">
        <Record RecordNumber="1">
            <AirTemp>0.39</AirTemp>
            <Precip>2.5</Precip>
            <Date>Jan 1, 1997</Date>
        </Record>
        <Record RecordNumber="2">
          ...
        </Record>
        <AirTempUnit>C</AirTempUnit>
        <PrecipUnit>mm/d</PrecipUnit>
        <SiteName>Siuntio</SiteName>
    </Site>
```
b)
```
<?xml version="1.0" encoding="UTF-8"?>
    <Site SiteCode="1">
        <Record RecordNumber="1">
            <AirTemp>0.39</AirTemp>
            <Precip>2.5</Precip>
            <SWDown>0.2</SWDown>
            <LWDown>25.0 </LWDown>
            <WindSpeed>0.3</WindSpeed>
            <RH>95</RH>
            <Date>Jan 1, 1997</Date>
        </Record>
        <Record RecordNumber="2">
          ...
        </Record>
         <AirTempUnit>C</AirTempUnit>
        <PrecipUnit>mm/d</PrecipUnit>
        <SWDownUnit>MJ/m2/d</SWDownUnit>
        <LWDownUnit>MJ/m2/d</LWDownUnit>
        <WindSpeedUnit>m/s</WindSpeedUnit>
        <RHUnit>%</RHUnit>
        <SiteName>Siuntio</SiteName>
    </Site>
```

Figure 4. Data input files in XML format for: (a) degree-day; and (b) energy balance snow models.

It should be noted, that the present sample application is limited and is insufficient in demonstrating feasibility of XML in complex environmental assessment studies. It does not reveal problems that would only appear in large-scale applications. For example, an extra step in transferring data from a database to the model increases the total turnaround time of the model execution.

Despite the shortcomings listed above, the current study suggests that XML is worth exploring on the track towards an open modelling framework. Such a framework would allow models developed by various expert groups to connect easily to a common information system. Figure 5 shows a schematic of an open modelling framework where data transfer is implemented with XML. Issues that require addressing in further development of the system described in this paper include: 1) currently only one database can be accessed in generating a XML data input file, 2) user interface has not yet been implemented with the system being command-line driven, and 3) XML tools, such as XML Schema (http://www.w3.org), are presently not used for validating the data input.

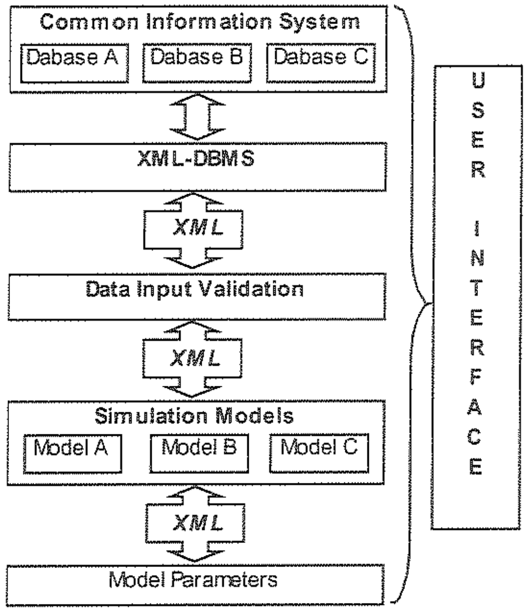

Figure 5. Schematic representation of an open modelling framework.

5. ACKNOWLEDGEMENTS

This study was funded by the Academy of Finland and by the Ministry of Agriculture and Forestry of Finland. We greatly appreciate support provided by Prof. Pertti Vakkilainen.

6. REFERENCES

Argent, R.M., and R.B. Grayson, Design of information systems for environmental managers: an example using interface prototyping, *Environmental Modelling & Software*, 16, 433-438, 2001.

Aspinall, R., and D. Pearson, Integrated geographical assessment of environmental condition in water catchments: Linking landscape ecology, environmental modelling and GIS, *Journal of Environmental Management*, 59, 299-319, 2000.

Booty, W.G., D.C.L. Lam, I.W.S. Wong, and P. Siconolfi, Design and implementation of an environmental decision support system, *Environmental Modelling & Software*, 16, 453-458, 2001.

Bourret, R., XML-DBMS: Middleware for Transferring Data between XML Documents and Relational Databases, http://www.rpbourret.com/xmldbms/, August 8, 2001.

Cate, H.H.ten, H.X. Lin, and A.E. Mynett, A case study on integrating software packages, In. Babovic, V., and L.C. Larsen (eds.) *HYDROINFORMATICS '98*, A.A. Balkema, pp. 457-464, Copenhagen, 1998.

Huang, G.H., L. Liu, A. Chakma, X.H. Wang, and Y.Y. Yin, A hybrid GIS-supported watershed modeling system, *Hydrological Science Journal*, 44, 597-610, 1999.

Huang, G.H., and J. Xia, Barriers to sustainable water-quality management, *Journal of Environmental Management*, 61, 1-23, 2001.

Koivusalo, H., M. Heikinheimo, and T. Karvonen, Test of a simple two-layer parameterisation to simulate energy balance and temperature of a snowpack, *Theoretical and Applied Climatology (in press)*, 2001.

Vehviläinen, B., Snow cover models in operational watershed forecasting, National Board of Waters and the Environment, Helsinki, pp. 112, 1992.

Voinov, A., and R. Costanza, Watershed management and the Web, *Journal of Environmental Management*, 56 (4), 231-245, 1999.