XML based Virtual Catalogue Module in CoastBase

Wassili Kazakos, Andreas Schmidt and Heiko Paoli1

Abstract
CoastBase is a project that aims to device a system for European marine and coastal management, concerning both the search for and access to a broad range of information sources. In this paper, one central part of the overall system, the virtual catalogue module, will be presented. It facilitates the search through distributed and heterogeneous information sources by implementing an I3 Reference Architecture. In contrast to most other approaches of this type, the XML extreme approach we have chosen allows highest possible modularisation and configuration of the system almost wholly without programming. The XML extreme approach means that XML Schema is used to define the vocabulary, XSL-T is used for the transformations and the querying and visualisation of the results is also XML based. Within this service-oriented architecture, SOAP is used for communication.

1. Introduction
CoastBase started in January 2000 with 11 partners from 8 European countries. It aims to improve European marine and coastal management and by the date of the upcoming symposium, an prototype will be available. The Project is supported by the European Commission within the IST sub-division of 5th framework programme. See (kazakos et. al., 2000) in the index to this paper for more information about the project or visit the web site http://www.coastbase.org. This paper is focused on the virtual catalogue module as one of the major results of the project. It is designed to be flexible enough to be used not only in other coastal and marine applications, but also in other application domains. With the XML extreme approach, a crucial step towards XMLization of software development is taken. This approach not only uses XML for data exchange, but also for schema description, querying, transformation, visualisation and communication.

Chapter 2 of this paper provides an overview of the CoastBase architecture. Chapter 3 describes the catalogue service module in detail. Chapter 4 discusses the modularization of the domain model based on XML Schema and Chapter 5 explains

1 Forschungszentrum Informatik (FZI), Haid-und-Neu-Straße 10-14, D-76131 Karlsruhe, Germany, coastbase@fzi.de; http://www.fzi.de/dbs.html.
the XML-based query language, which has been developed within this project. Chapter 6 shows the configuration possibilities for the virtual catalogue model, and finally, in Chapter 7, we give a brief outlook of future developments.

2. **Overall Modular Architecture**

The CoastBase system is based on a 3-tier multi-server architecture. The three tiers are the CoastBase client tier, the CoastBase server tier and the data abstraction tier. Client and server are further divided into sub-components according to the functional decomposition of CoastBase. These are catalogue, data access and manipulation, and feedback. In the data abstraction tier, the wrappers for the inventory data, as well as for the local repositories are implemented.

When talking about the virtual catalogue of CoastBase, the search and feedback modules are meant. They cover three main functional units needed in every similar type scenario. The administration authenticates and authorises the user according to user groups. The query dispatching and retrieval of distributed inventory data is done by the catalogue server. The feedback server contains a update module, allowing the creation and actualisation of inventory data.

![Figure 1 CoastBase general architecture](image)

3. **The Virtual Catalogue Module**

One of the major design goals of the overall system and especially of the virtual catalogue component was the extensive use of XML and XML-based techniques, in order to achieve maximum interoperability with other systems and the reusability of several parts of the system.
3.1 Catalogue Client

The user interface for the catalogue client is dynamically generated HTML. In future XHTML might be generated. Components, like the query formulation module, requiring computation logic, are created by JSP templates. Data transformation oriented parts, like the result list presentation, are generated using XSL-T templates. In order to achieve a uniform and maintainable look-and-feel for the system, Cascading Style Sheets (CSS) are used to separate content from layout.

Multilinguality will be achieved primarily through the Java internationalization features. In the context of XSL-T style sheets, there will be a different style sheet for each language. Common rules will be imported from library style sheets, so that there is not much redundancy. For the query formulation external services can be used, e.g. a HTML thesaurus browser using the General Multilingual Environmental Thesaurus (GEMET) and a tool for geographic selection.

3.2 Catalogue Server

The catalogue server implements the main functionality of the virtual catalogue that is able to access distributed, but homogeneous sources with the same domain model.

In order to build a scalable system that makes it easy to integrate new sources, a virtual integration approach was chosen. Except for a local repository into which products generated by CoastBase users are uploaded, there is no centralized database. In fact there is neither a centralized database for the data nor for the metadata. Instead, the existing Internet access points for the sources to be connected are used. The user queries are dispatched at run-time to the sources. In order to overcome the technical heterogeneity, wrappers that translate the queries into queries native to the source, adapt the access protocol, translate the results into XML and transform the structure to fit the CoastBase domain model are employed.

4. Domain Model and XML Schema

CoastBase should provide a uniform structure for the objects it makes accessible, especially for the metadata records describing these objects. This helps the user in formulating his query and processing the results. This type of global information model can be achieved by using one of two methods:

- The local schemas of the connected source can be integrated into a global schema using techniques developed in the field of schema integration.
- The field of application (here: data and information about coastal and marine environments) can be modelled (almost) independent of the source to which it will be connected. The result will be a so-called domain model.
Since CoastBase is supposed to be extensible and to be able to integrate many new sources in the future, the more flexible to changes to the underlying sources domain model approach has been chosen. The design of the domain model has several goals:

- to incorporate existing metadata standards such as the Global Environmental Information Locator Service (GELOS) and the European Directory of Marine Environmental Data (EDMED)
- to provide a framework for future extensions or applications to other environmental fields

In order to achieve these goals, a modular approach in conjunction with a semi-structured data model has been chosen. The domain model consists of different building blocks, such as spatial, subject and temporal coverage. Within those building blocks, several modules can be used to specify the building blocks in different ways. An example would be spatial coverage specified with modules such as bounding boxes, terms from a geographical thesaurus or even polygons. In each of the CoastBase independent building blocks, there is one module based on the GELOS and EDMED which represents the core attributes which should be available at the data sources. These attributes can be searched for via the standard user interface. Search functionality for additional attributes may be provided by an advanced search form.

\begin{center}
\begin{tabular}{|c|c|}
\hline
CoastBase independent & CoastBase dependent, e.g. \\
\hline
General & Oceanology \\
Temporal & Climatology \\
Spatial & Contaminants \\
Images/Maps & Fisheries \\
Literature & \\
Links & \\
Originator/Distributor & \\
\hline
\end{tabular}
\end{center}

In order to ensure flexibility and extensibility, the CoastBase domain model has been embedded into a modular framework. This framework is based on modularization techniques which are also used in XHTML. The domain model is organized into so-called building blocks which group attributes according to their semantics, such as temporal coverage, spatial coverage, information about originator and distributor, etc. New attributes for special application domains can be added easily without interfering with already existing building blocks. Of particular significance is the advantageous feature which makes it possible for old applications to simply ignore new building blocks.
Since there are many ways of specifying spatial coverage and other building blocks, the building blocks have modules. In reference to spatial coverage, building blocks of importance are bounding box, terms from a geographical thesaurus, polygons, etc. Within each building block, there is one core module which must be supported. This core module is based on GELOS and EDMED.

Several types of data are re-used within the domain model in a number of different attributes. This especially relates to the address information. Via XML-Schema, these complex types must only be specified once. If any changes or additions have to be made, they only have to be carried out once. Moreover, in the transformation routines for the presentation of the metadata (via XSL-T), this advantage can also be exploited. The transformation rules only have to be specified once.

The catalogue component of CoastBase is designed to be domain model independent as far as possible and to require only minimum changes (mainly to the user interface) for adaptation to new variants of the domain model. This is mainly achieved using XML based techniques.

5. A XML based Query language

According to the system requirements, CoastBase will support a boolean query language with AND and OR connectors on top of a semi-structured data model (XML). Since queries are only used to select certain metadata records which satisfy a certain query condition, the language does not have to contain a projection/construction clause or a grouping/sorting clause. Furthermore, joins between different information objects within a source or across different source are not required.

5.1 Query Atoms

Query atoms, which the boolean expressions are constructed from (like Title contains “CoastBase”), have the following conceptual structure:

Attribute. This specifies the attribute the atom refers to. The attribute is specified using nested XML elements, like in the domain model.

Predicate. This specifies the type of comparison. Typical predicates for textual data are equality (=) and inequality (!=), but also truncation predicates (contains, starts with, ends with). For numerical data, the typical predicates are <, >, <=, >=. For the use of hierarchical thesauri, additional predicates need to be added which specify that child terms or parent terms are also valid (equal or parent, equal or child).

Value. This specifies the value that is the basis of comparison.

In the case of multi-valued attributes, the query atom is satisfied if one value satisfies the atom.
5.2 Complex Query Expressions

Complex query expressions are formed using the connectors AND and OR with common semantics. The connectors are considered n-ary, i.e. \( x_1 \ \text{AND} \ x_2 \ \text{AND} \ x_3 \) need not be expressed as \( ((x_1 \ \text{AND} \ x_2) \ \text{AND} \ x_3) \). If required, the query language can be extended to support negated atoms also (NOT operator).

5.3 Incomplete Information

In heterogeneous environments and especially when using a semi-structured data model, the case will come up in which certain query atom refers to an attribute that is not available at the source. Nonetheless, the result of this type of query must be clearly-defined. There are two options to achieve such a clearly-defined query result: either the query atom which refers to the missing information portion must be evaluated as false (guarantee that the user does not receive any result object that does not satisfy the condition) or the atom must be ignored (which does not exclude any result object that may satisfy the condition although it cannot be guaranteed). The choice between these two options is essentially a trade-off between result quality (strict evaluation) and result completeness (loose evaluation).

5.3.1 Strict Evaluation

At the evaluation level of the query condition, a strict evaluation strategy means that for those attributes for which no mapping to source attributes is possible (i.e., no corresponding attribute is available), the query atom is evaluated as false.

5.3.2 Loose Evaluation

Formally, the option of ignoring query atoms that refer to an attribute which is not available at the source can be expressed as a three-valued logic with the truth values \( T, F \) and \( ? \). The query atom that cannot be evaluated due to incomplete information is evaluated as \( ? \). The evaluation function is then defined as the following:
\[
\begin{align*}
\text{eval}(T \ \text{AND} \ ?) &= T & \text{eval}(F \ \text{AND} \ ?) &= ? \\
\text{eval}(T \ \text{OR} \ ?) &= T & \text{eval}(F \ \text{OR} \ ?) &= ?
\end{align*}
\]

If the whole expression is evaluated as \( ? \), then treat it like \( T \).

If NOT expressions are allowed, then this evaluation function can be extended in the following manner:
\[
\text{eval}(\text{NOT} \ ?) = ?
\]
5.4 Representation In XML

Since there is no well-established query language for XML and the requirements for CoastBase are very modest (only query conditions are required, no "joins"), a simple, proprietary language is used which is basically an XML-representation of a boolean expression tree with query atoms as leaves. In future extensions of CoastBase, this simplified query language can be easily migrated to an XML query standard if required.

Using the inclusion mechanism of the XML Schema standard, it is possible to validate the query against the domain model specification.

6. Virtual Catalogue Configuration

One of the major achievements of the virtual catalogue module is the configuration possibility provided by the XML extreme approach. The goal is to be able to install a new instance of the virtual catalogue module just by changing text files and almost wholly without programming, as shown in following picture.

7. Outlook and Future Work

In this paper, the XML extreme approach for accessing coastal and marine inventory data was presented. The modularisation and configuration of the system was one of the major design goals and the results prove the clear advantages of the XML approach compared with proprietary configuration mechanisms. The virtual catalogue of CoastBase is the starting point for a generic framework on the XML-based intelligent integration of information (XI3) currently under development at FZI. Within XI3, catalogue searching will be combined with knowledge management features, like support for ontologies and access to semantic relations.
By adding workflow support and context-aware searches, XI3 will become a full fledges framework of XML based integration tools and methods.

8. Acknowledgements

The authors greatly appreciate the cooperation with all CoastBase partners (see http://www.coastbase.org section partners). Financial support of the European Commission under contract IST-1999-11406 is gratefully acknowledged.

9. References


Copyright 2001 Metropolis Verlag, Marburg, ISBN: 3-89518-370-9