Corpora as Object-Oriented System. From UML-notation to Implementation

Serge A. Yablonsky
Petersburg Transport University, Computer Department, Russicon Company;
Kazanskaya str., 56, ap.2, St.-Petersburg, Russia, 190000
phone/fax: +7-812-3127213, *email:* root@russicon.spb.su, *URL:* www.russicon.ru;

Abstract

The paper descusses the complete process of building and managing a corpora warehouse, including case study involving the development of UML-specifications and patterns, architecture and examples of actual implementations of DBMS tools to support strategic corpora analysis.

1. Introduction

Documental systems with large-scale linguistic annotation are in service of a wide range of research and commercial applications. However, the research progress in this area is restricted by the lack of infrastructure for annotation technologies supported with special-purpose tools, making it difficult either to develop new coding systems or to place multiple annotations on the same source (Carletta, J., McKelvie, D., Isard, A., 2002).

For successful development of corpora invironment the needs in several inter-related areas are considered to be necessary (Ide, N. & Brew, C., 2000):

- Annotation and encoding formats;
- Data and tools architecture.

Reusability of linguistic resources is achieved by describtion of the data and its annotations using a common data model. For that purpose the XML and related standards such as the Resource Definition Framework (RDF) are widely used (Ide, N., Romary L., 2001). For example, the XML Corpus Encoding Standard (XCES, http://www.xml-ces.org) (Ide, *et al.,* 2000) is designed to be optimally suited for use in language engineering research and applications, in order to serve as a widely accepted set of encoding standards for corpus-based work in natural language processing applications. It has been widely adopted by the language processing community (a list of European and US projects using the CES is at http://www.cs.vassar.edu/CES/CES-P.html).

In the corpora case study we describe how XML linguistic annotation can be supported using XML as the data format. We provide a view on the corpora together with corpus-handling environment as object-oriented system. That suggests strategies for development of a widely reusable and extensible data and tools.

The paper descusses the complete process of building and managing a corpora warehouse based XML encoding, including the development of UML-specifications, architecture and examples of actual implementations of DBMS tools to support strategic corpora analysis. Open UML-specification of object-oriented corpora system could be expanded in future by community of language and speech software and resources developers. The set of different types of UML-specifications brings us to full three-level system, including data, business and user services.

We present the basic features of a prototype corpora knowledge management system under development intended to support linguists in their daily work. The system offers facilities to assist liguists and internet users as they search for relevant material, and then classify and annotate this material in a repository. In general the main features of corpora warehouse (Sullivan D., 2001) are implemeted using commercial DBMS Oracle9i.

2. Corpora System UML Notation

2.1. UML: A standard notation for object-oriented systems

Today Unified Modeling Language (UML) defines a standard notation for object-oriented systems (Booch G., Rumbaugh J., and Jacobson I., 1998). The objective of modeling is to complete a rigorous design with quality checks before we build a corpora system. The UML is an object-oriented methodology that standardizes modeling language and notation, not a particular method.

UML supports several different views of a system - class diagrams, behavior diagrams, use-case diagrams, and implementation diagrams. Use-case diagrams let UML users define how actors participate in an interaction with the system. UML users can capture the system's dynamic behavior by using activity diagrams, collaboration diagrams, sequence diagrams, and state diagrams. To document the lower-level details of a system, UML users can develop component diagrams and deployment diagrams. The UML is also extensible to support new modeling concepts by the use of stereotypes and patterns. The UML is a powerful solution for application object modeling.

Using UML enhances communication between linguistic experts, workflow specialists, software designers and other professionals with different backgrounds. UML can be used on a general level, which is intuitive for the users of workflow systems. In spite of this, UML symbols also have defined semantics, which means that the visual workflow description can be used as a software specification.

2.2. Mapping RDF/XML to UML

The RDF schema model itself is equivalent to a subset of the class model in UML. So RDF schema elements map directly into UML class model elements (*A Discussion…, 1998*). RDF schema uses a DLG (Directed Labeled Graph) model for describing schemas. Class schemas expressed in UML can be viewed also as DLGs.

To support mapping XML to UML, UML is extended with stereotypes for XML elements, element attribute lists, entities, and notations. Element attributes are like

UML class attributes. Stereotypes or tagged values represent XML operator symbols, sequence lists, choice lists, and element and element attribute multiplicity.

For example, Rational Rose (Rational Rose Enterprise Edition, 2001) represents XML DTD elements as stereotyped classes. The list below shows the primary stereotypes for XML DTDs:

- <<DTDElement>> represents all elements in the DTD;
- <<DTDGroup>> represents the grouping type for a specific element;
- <<DTDEntity>> represents the entities declared in an XML DTD;
- <<DTDNotation>> represents the notations declared in an XML DTD.

XML DTDs declare the elements and structural relationship between elements. XML DTD stereotypes graphically depict the DTD elements and the structural relationship between elements using UML constructs, such as multiplicity and association relationships. Multiplicity is the number of occurrences for a given instance of the XML element. Because the relationship between elements in XML DTDs is structural, XML elements map to association relationships. In XML, DOCTYPE name is the root element of the DTD. If XML documents are represented as a tree structure, then elements in the document are nodes on the tree. Each node on the tree contains actual data in the document. UML notation of XML Corpus Encoding Standard DTD is shown on the figure 1 (fragment).

2.3. XML structured data representation

The database research community has been actively investigating XML (see, for example, Didier M., *et al.,* 2000).

Much of the effort has been directed at using XML as a database wrapper and mediation medium, storing and indexing XML in traditional database systems, understanding the interaction of DTDs with constraint mechanisms, and designing query languages for XML.

When database systems are used for XML, data structuring is systematic and explicitly defined by a database schema.

One of the main questions in the XML/SGML markup technologies for working with linguistic data and paticular with corpora is how to map XML/SGML to database system.

We descuss four mappings of XML/SGML to databases:

- a table-based mapping;
- an object-relational (object-based) mapping;
- direct XML maping;
- hybrid XML mapping.

2.3.1. Table-based and object based XML mappings

Two first mappings model the data in XML documents rather than the documents themselves. This makes these mappings good for data-centric documents and poor for document-centric documents.

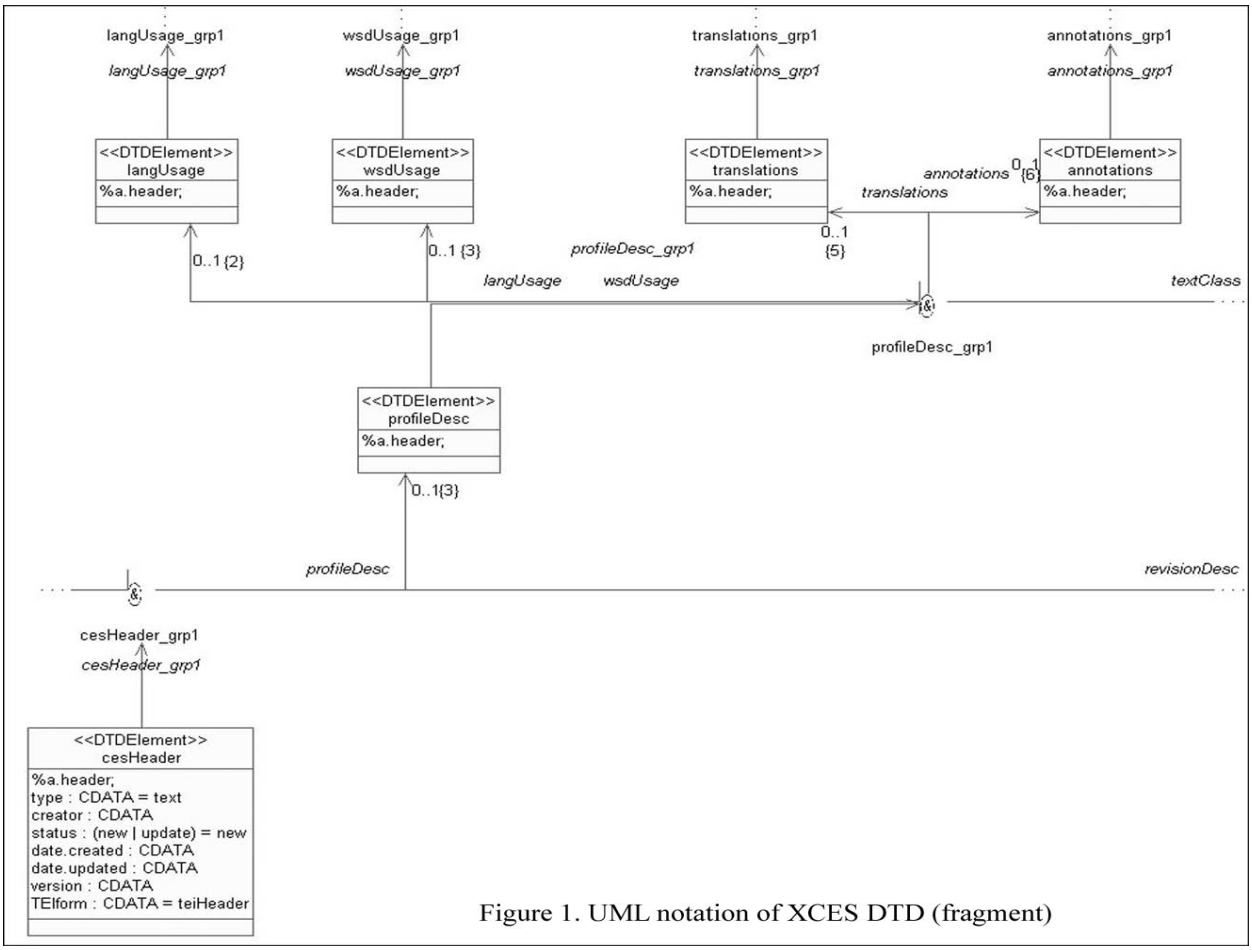

Figure 1. UML notation of XCES DTD (fragment)

Both mappings are commonly used as the basis for software that bidirectionally transfers data between XML documents and databases: from XML documents to the database and from the database to XML documents. In this approach a relational or object-oriented database system is extended to support XML data management. All current commercial database systems provide XML support of that kind. For example, Oracle's XML SQL Utility (Oracle9i Database Online Documentation, 2002) and IBM's DB2 XML Extender (Bertino, E., Castano, S., Ferrari, E., and Mesiti, M., 2001). For the sake of discussion, we consider Oracle XML SQL Utility as representative of the many systems following this approach.

In Oracle 9i complex XML documents can be stored as object-relational instances and indexed efficiently. Such instances fully capture and express the nesting and list semantics of XML. With Oracle's extensibility infrastructure, new types of indixes, such as path indixes, can be created for faster searching through XML documents. XML SQL Utility (XSU) stores XML and converts SQL query results into XML. XSU provides the means to store an XML document by mapping it to the underlying object-relational storage, and conversely, provides the ability retrieve the object-relational data as an XML document. XSU converts the result of an SQL query into XML by mapping the query alias or column names into the element tag names and preserving the nesting of object types. The result can be in text or a DOM (Document Object Model) tree. The generation of the latter avoids the overhead of parsing the text and directly realizes the DOM tree.

2.3.2. Direct (composed) XML maping

The *third mapping* is likely to be used as a direct mapping of *any* XML document in database system as if XMLType is the database data type and is good for managing document-centric documents.

Oracle9i stores XML data in its special XMLType or CLOB/BLOB columns and use XMLType functions or Oracle Text indexing to search these documents (Oracle9i Database Online Documentation, 2002). XMLType or CLOBs/BLOBs are used to store XML documents if the incoming XML documents do not conform to one particular structure. Storing an intact XML document in a CLOB or BLOB is a good strategy if the XML document contains static content that will only be updated by replacing the entire document. Corpora include written text such as articles, advertisements, books, legal contracts, and so on. Documents of this nature are known as document-centric and are delivered from the database as a whole. Storing this kind of document intact within database gives the advantages of an industry-proven database and its reliability over file system storage.

Oracle Text (interMedia Text) indexing enables fine grain searching of XML element content. Oracle allows the creation of Oracle Text (interMedia Text) indexes on LOB columns, in addition to URLs that point to external documents. This indexing mechanism works for XML data as well. Oracle9i recognize XML tags, and section and sub-section text searching within XML elements' content. The result is that queries can be posed on unstructured data and restricted to certain sections or elements within a document.

XMLType or CLOBs/BLOBs storage are ideal if the structure of the XML document is unknown or dynamic, but much of the SQL functionality on object-relational columns cannot be exploited. Concurrency of certain operations such as updates may be reduced. However, the exact copy of the document is retained.

2.3.3. Hybrid XML maping

Hybrid XML mapping in many cases gives better control of the mapping granularity. For example, when mapping a text document, such as a book, in XML, it may not be want for every single element to be expanded and stored as object-relational. Storing the font and paragraph information for such documents in an object-relational format may not be useful with respect to querying. On the other hand, storing the whole text document in a CLOB reduces the effective SQL queriability on the entire document. The alternative is to have user-defined granularity for such storage. The metadata for the document in this case can be stored in object-relational tables in the server for fast indexing and access.

Figure 2 shows schematically how a hybrid approach to storing XML documents in the database, provides finer granularity for ease of data searches. In the book example, it may not be wanted the following: to query on top-level elements such as chapter, section, title, and so on. These elements can be stored in object relational tables. To query the book's contents in each section. These sections can be stored in a CLOB. The granularity of mapping is specified at table definition time.

The advantages of the hybrid XML mapping are the following:

- it gives the flexibility of storing useful and queryable information in object-relational format while not decomposing the entire document;
- it saves time in reconstructing the document, since the entire document is not broken down.

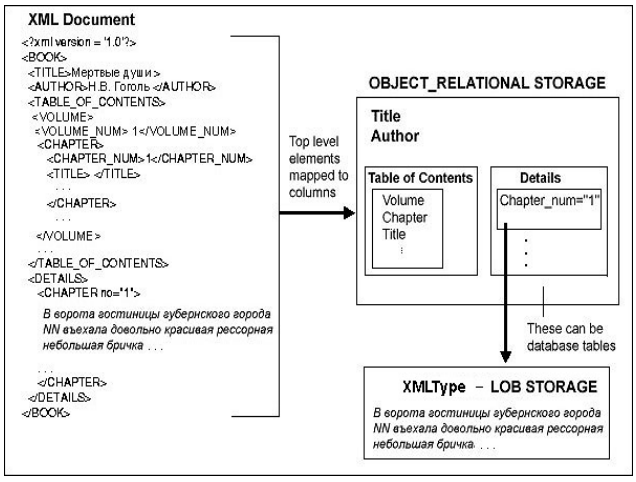

Figure 2. Hybrid XML mapping

2.3.4. UML specifications of three-level corpora system

We present the open UML-specification of object-oriented corpora system that could be expanded in future by community of language and speech software and resources developers. The set of different types of UML-

specifications brings us to full three-level system, including user, business and data services.

UML specification of data services should use standard XML DTD (or its part) as a basis. In the previous sections we descussed the ways of selection of the type of UML/XML mapping in database system.

UML specification of business services also should use standard UML notations of standard linguistic annotation and corpora manipulation procedures. Reusability of linguistic and corpora manipulation business services could be achieved by usage of a widely accepted set of UML notation standards for corpus-based work in natural language processing applications.

3. Corpora Management with DBMS XML-Enabled Technologies

3.1. Customizing Presentation of Corpora

XML is increasingly used to enable customized presentation of Corpora for different browsers and users. By using XML documents along with XSL stylesheets on either the client or server, XML data could be transform, organize, and present tailored to individual users for a variety of client devices. Using XML and XSL also makes it easier to create and manage dynamic Corpora Web sites. The corpora look is changed simply by changing the XSL stylesheet, without having to modify the underlying business logic or database code of corpora management system. For new looks design new XSL stylesheets is needed. This is illustrated in Figure 3.

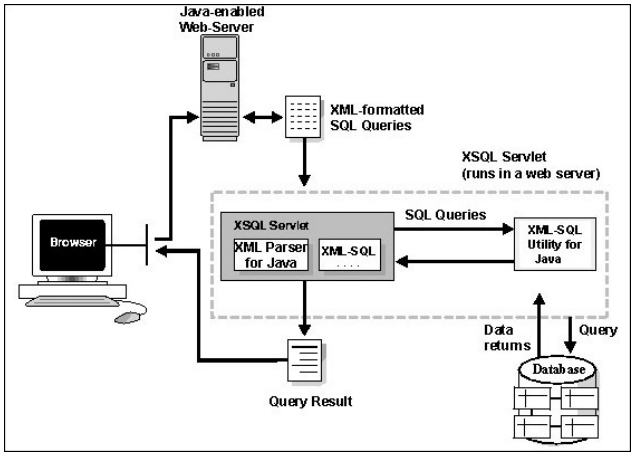

Figure 3. Customizing Corpora Presentation

3.2. Publishing Composite Corpora Documents

Corpora has numerous document repositories of SGML and XML marked up text fragments and annotations. Composite documents must be published dynamically. After data modeling and design guidelines object views can more readily be created against the data. For example, in Oracle DBMS documents are stored in in XML format using XMLType, where the relational data is updatable. Oracle9i's Internet File System (9iFS) as the data repository interface and helps implement XML data repository management and administration tasks.

Corpora management system can use XSL stylesheets to assemble the document sections (fragments) or/and annotations and deliver the composite documents to users. One suggested solution is to use Arbortext and EPIC (http://www.arbortext.com) for single sourcing and authoring or multichannel publishing. Multichannel publishing facilitates producing the same document in many different formats, such as HTML, PDF, WORD, ASCII text, SGML, and Framemaker.

These are the main tasks involved in such solution:

- Design of XML DTD and database.
- Storing document's sections or fragments in XMLType columns in CLOBs in the database.
- Create XSL Stylesheets to render the sections or fragments into complete documents.

For Oracle such XML components are used

- XML Parser with XSLT ;
- XSQL Servlet and XSU to move sections or fragments into and out of the database

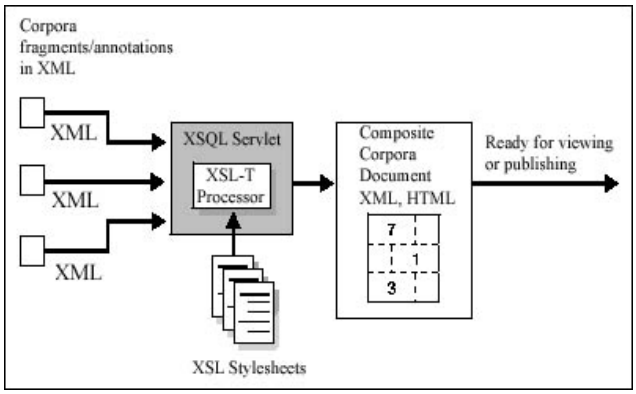

Figure 4. Using XSL to Create and Publish Composite Corpora Documents

3.3. Pilot Corpora Management System

A developer of a large corpora receives data from various news sources. This data must be stored in a database and sent to all linguists and users on demand so that they can view (annotate) specific and customized parts of new corpora. The developer uses XSL to normalize and store the documents in a database. The stored data is used to back several internet/intranet nodes. These nodes receive HTTP requests from various wired and unwired clients.

We use XSL stylesheets with the XSQL Servlet to dynamically deliver appropriate rendering to the requesting service. See Figure 5.

The Corpora management system application uses Oracle XML platform components together with the Oracle9i database to build a web-based news service. It combines Java, XML, XSL, HTML, and Oracle9i.

The powefull search engine of prototype system particular uses advantages of Oracle Text search and services and includes:

- *Content-based retrieval on free text with both literal (word) predicates and thematic predicates*. It includes: a comprehensive range of operators and index preferences (e.g. Boolean, exact phrase match, proximity, section searching, fuzzy, stemming, wildcard, thesaurus, stopwords, case sensitivity, and search scoring), "about" search, structured search, broad document

format support and multi-language support. For example, texts can be searched for stems of word e.g "teach" would return "teaching", "taught" etc. Fuzzy match can be used if you are not sure of the spelling of a word. A search can be done to find words which are close to each other within a word. Documents can be searched on what a document is about as opposed to the existence of specific words. Gists or Theme Summaries can be produced which produce a summary of what a document is about (using themes).

- *For XML framework XPath searching enables sophisticated queries which can reference and leverage the embedded structure of XML documents* – instead of using a text query to find documents, you use a document to find queries. XML path searching is able to perform sophisticated section searches: doctype disambiguation, attribute value searching, automatic section indexing, and more.

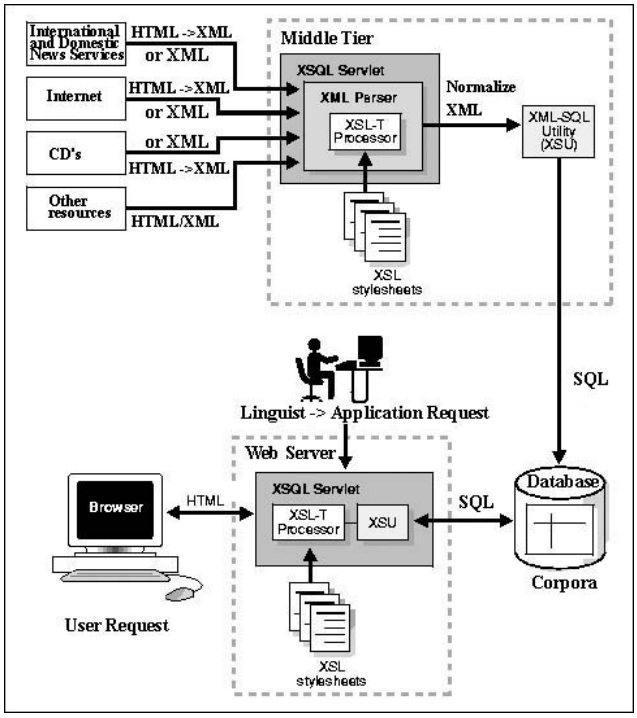

Figure 5. Pilot Corpora Management System

In addition to the search capabilities, a number of other features are provided to simplify application development.

- *Corpora format support.* In order to index documents stored in a variety of native formats, such as Word, Excel, PowerPoint, WordPerfect, HTML, and Acrobat/PDF, system supplies a broad variety of "filters" that allow documents stored in their native formats to be indexed. Support for more then 150 file formats in order to index files in a large range of formats including Word, Acrobat, HTML, WordPerfect, Powerpoint, Excel Flexible Storage Location - documents can be stored and indexed in the database, in a location pointed to by a URL or in an external file.
- *Corpora viewing and highlighting.* System services can convert any supported document format to either plain text or formatted text (an HTML approximation retaining as much as possible of the original formatting; available for all formats except PDF). Both plain text and HTML versions may be viewed in a standard browser.
- *Text manager.* System supplies an administration tool through which all major text maintenance and administration functions may be performed.

Today we use corpora system for Russian language corpora (Yablonsky S.A., 2000). We plan to establish a suitable portal for Web access language resources using existing software.

4. References

Bertino, E., Castano, S., Ferrari, E., and Mesiti, M., 2001. Controlled access and dissemination of XML documents. In DB2 Universal Database XML Extender, XML Extender Administration and Programming. http://www-4.ibm.com/software/data/db2/extenders/ xmlext/ docs/v71wrk/english/index.htm, retrieved August 2001.

Booch, G., Rumbaugh, J., and Jacobson, I., 1998. The Unified Modeling Language user guide, Addison-Wesley.

Carletta, J., McKelvie, D., Isard, A., 2002. Supporting linguistic annotation using XML and stylesheets. In G. Sampson and D. McCarthy (eds.) *Readings in Corpus Linguistics,* , London and NY.

A Discussion of the Relationship Between RDF-Schema and UML. W3C Note 04-Aug-1998, http://www.w3.org/TR/NOTE-rdf-uml/.

Didier M., *et al.,* 2000. Professional XML. Wrox Press Ltd.

Ide, N., Romary L., 2001. XML Support for Annotated Language Resources. In: *Linguistic Exploration,* Workshop on Web-Based Language Documentation and Description, Dec 12 - Dec 15, 2000, University of Pennsylvania Philadelphia, Pennsylvania, USA.

Ide, N. & Brew, C., 2000. Requirements, Tools, and Architectures for Annotated Corpora. In: *Proceedings of the EAGLES/ISLE Workshop on Meta-Descriptions and Annotation Schemas for Multimodal/ MultimediaLanguage Resources and Data Architectures and Software Support for Large Corpora*. Paris: European Language Resources Association.

Oracle9i Database Documentation (Release 9.0.1), 2002.

Rational Rose Enterprise Edition 2001, Documentation.

Sullivan D., 2001. Document Warehousing and Text Mining, John Wiley & Sons, 542 p.

Yablonsky S.A., 2000. Russian Monitor Corpora: Composition, Linguistic Encoding and Internet Publication. Proceedings Second International Conference on Language Resources & Evaluation, Athens, Greece, 2000.