# OpenMMS: An Ontology Driven Architecture for Macromolecular Structure

Douglas S. Greer
*University of California, San Diego*

John D. Westbrook
*Rutgers University*

Philip E. Bourne
*University of California, San Diego*

An object metamodel based on a standard scientific ontology has been developed and used to generate a CORBA interface, an SQL schema and an XML representation for macromolecular structure (MMS) data. In addition to the interface and schema definitions, the metamodel was also used to generate the core elements of a CORBA reference server and a JDBC database loader. The Java source code which implements this metamodel, the CORBA server, database loader and XML converter are part of the soon to be released OpenMMS toolkit.

The CORBA, SQL and XML expressions of the MMS data were designed to address the needs of a wide range of applications and to provide efficient, high performance access for several common use cases. These cases include many types of services and applications that require fast and flexible access to molecular structure information provided by the Protein Data Bank (PDB) [1]. Many types of inquires involve calculations based on the spatial position of atoms or residues while others involve extensive string searches through complex data types. Although these inquiries cannot be anticipated in advance, a major goal of this work is to provide an easy-to-use, interfaces that can provide for specialized computations with near-optimal performance.

A major milestone in this work was passed in February 2001 when the Board of Directors of the Object Management Group (OMG) voted to adopt the CORBA IDL Macromolecular Structure specification derived from this metamodel. This specification [2] provides an open, standardized object-oriented application-programming interface (API) that allows direct access by remote programs to the binary data structures of the PDB.

The overall view of the OpenMMS toolkit is shown in Figure 1. In the upper portion of this figure, the flow of information that describes the structure of the MMS data is shown using thin black arrows. In the lower portion, the flow of actual MMS data is shown using thick gray arrows.

The mmCIF ontology metamodel is derived from the scientific ontology [3] developed by the International Union of Crystallography (IUCr) and based on the macromolecular Crystallographic Information File (mmCIF) standard, which encompasses both the mmCIF dictionary and the mmCIF data files [4]. The mmCIF dictionary, which provides a technical definition of the data fields along with their representations and relationships, supplies most of the information used by the metamodel framework. The solid scientific basis provided by the dictionary helps promote program interoperability and insure correct numerical results. Although the mmCIF dictionary was written in STAR format [5], the MMS ontology metamodel extracts the needed information from the dictionary such that the ontology and its derivations are independent of STAR or any other particular file format.

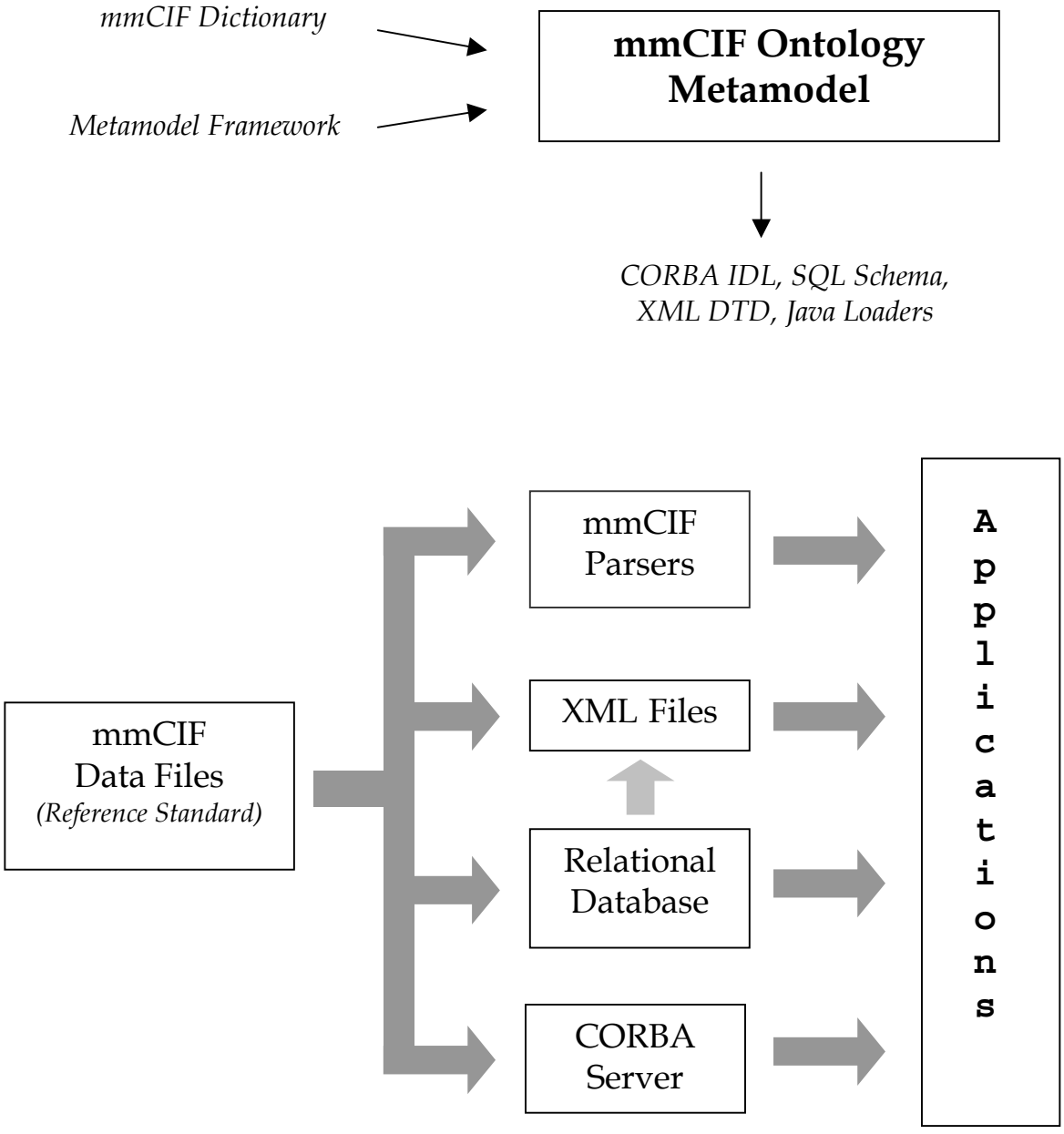


*Figure 1. The OpenMMS Metamodel and Dataflow*

The metamodel itself is structured as a directed acyclic graph, which is created from a hierarchy with some well-defined interconnections between the nodes. The nodes are instances of metamodel classes such as interfaces, structs, lists, constants and fields. A framework that determines which mmCIF categories and items are to be used and how they are partitioned into modules defines the overall structure of the metamodel. The XML, SQL and CORBA expressions of the mmCIF ontology and several of the key Java source code files

are generated from the metamodel using the “Visitor” design pattern [6]. The Visitor base class defines the process of traversing the metamodel hierarchy and various Visitor subclasses generate specific files such as the CORBA IDL definition, the SQL schema and Java source code for the JDBC loader. Using the Visitor design pattern allows the code that generates each of these to be cleanly divided into a separate class. Furthermore the subclasses only need to override methods in the base class that correspond to metamodel elements they are interested in.

The “glue” that ties together the CORBA, XML and SQL representations of the MMS data is the mmCIF data files. Any errors or discrepancies in these expressed forms are resolved by referring back to this standard reference. The Entry object, which models a single structure, is a central object defined in the metamodel. In the XML, mmCIF, and PDB representations an Entry usually corresponds to a single flat file. With any of these flat files representations users are required to retrieve and parse the complete file in order to use even a small portion of it. However implementations of the CORBA and SQL interfaces allow applications to retrieve a single MMS data element from a remote server and import it for local use. Moreover the data is already in binary form, so the time consuming parsing and ASCII to binary conversions that are performed repeatedly by each application, are not necessary.

Among the four methods for obtaining MMS data that are shown in Figure 1, the mmCIF parsers are the most general purpose and also provide the most detailed access to the underlying data. Parsers and access libraries in several programming languages have been developed to support mmCIF [7]. In addition, the OpenMMS toolkit contains a Java parser that is used to generate the XML, SQL and CORBA data representations, but can also be used directly by application software to load specific application defined data structures. Rather than force an application to use a data structure defined by the parser, the Java parser uses the “Builder” design pattern [6] to pass data directly into an application. Using this approach, the application creates a subclass of the parser’s Builder class and overrides a few methods with code that examines the parameterized data types, and stores the data values of interest into a data structure that it defines.

A relational database that supports an SQL-92 compatible interface provides an appropriate API for many applications, particularly ones that require extensive string searches. The mmCIF ontology maps onto a relational database schema in natural way in that categories in mmCIF correspond to entities or tables and items in mmCIF correspond to attributes or columns.

XML is a simple, powerful and widely used standard for interchanging data. However the use of opening and closing tags around each data value results in a file size approximately ten times larger that the corresponding mmCIF data file which uses a single white space separator between data values in a loop structure. Due to the large number of atoms in a typical structure, these large files quickly overwhelm most general-purpose XML software. CML [6] and other XML formats have proposed grouping the data items in rows or columns surrounded by single tags, but such schemes lose some of the advantages gained by using XML. However in either case XML files can still serve as the lingua franca between applications provided they are able to efficiently handle the large amount of data involved.

In situations where the XML will ultimately be displayed by a users web browser, a system design that generates a potentially much smaller XML document based on a query to a relational database may be more appropriate. This data path is shown in Figure 1. Software for generating XML documents based on SQL or XML queries is not provided by the OpenMMS toolkit, but is available from many relational database companies and third-party software vendors.

The CORBA server potentially provides the highest performance access to MMS data. The object-oriented interface is used to define structures independent of platform and programming language, and yet may be optimized to copy binary data quickly and efficiently across the network. Thus by the appropriate caching of the binary MMS data, a CORBA server can provide ideal support for extensive scientific calculations performed in a large multiprocessor environment.

Applications are always limited by the computational power and I/O bandwidth of the available hardware and the time available for software development and optimization. The appropriate use of metamodels can present the scientific data in a number of ways that reduce the burden on software developers while at the same time increasing the effective functionality and performance.

The PDB is managed by the Research Collaboratory for Structural Bioinformatics (RCSB), a nonprofit consortium comprised of Rutgers University, the San Diego Supercomputer Center and the National Institute of Standards and Technology.

## References


1. H.M. Berman, J.D. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne, (2000) The Protein Data Bank. *Nucleic Acid Research* 28(1), 235-242.
2. ftp://ftp.omg.org/pub/docs/lifesci/00-11-01.pdf
3. J.D. Westbrook and P.E. Bourne, (2000) STAR/mmCIF: An Extensive Ontology for Macromolecular Structure and Beyond. *Bioinformatics* 16(2) 159-168
4. P.E. Bourne, H.M. Berman, B. McMahon, K. Watenpaugh, J. Westbrook., and P.M.D. Fitzgerald, (1997) The Macromolecular CIF Dictionary. *Methods in Enzymology*. 1997 227, 571-590.
5. S.R. Hall, (1991) The STAR file: A new format for electronic data transfer and archiving. *J. Chem Inf. Comput. Sci.*, 31, 326-333
6. E. Gamma, R. Helm, R. Johnson, J. Vlissides. (1994) *Design Patterns, Elements of Reusable Object-Oriented Software.* Addison-Wesley, Reading MA.
7. http://www.rcsb.org/pdb, http://pdb.sdsc.edu, http://pdb.rutgers.edu/mmcif, http://www.iucr.ac.uk/iucr-top/cif/index.html


## Authors Contact Address


Douglas S. Greer
San Diego Supercomputer Center
University of California, San Diego
9500 Gilman Drive
La Jolla, CA, 92093-0527, USA
Email: dsg@sdsc.edu