Astronomical Data Analysis Software and Systems IX
ASP Conference Series, Vol. 216, 2000
N. Manset, C. Veillet, and D. Crabtree, eds.

New Capabilities in the Astrophysics Multispectral Archive Search Engine

C. Y. Cheung

NASA Goddard Space Flight Center, Astrophysics Data Facility, Greenbelt, MD 20771

S. Kelley, N. Roussopoulos

University of Maryland, Institute of Advanced Computer Studies

Abstract. The Astrophysics Multispectral Archive Search Engine (AMASE) uses object-oriented database techniques to provide a uniform multi-mission and multi-spectral interface to search for data in the distributed archives. We describe our experience of porting AMASE from *Illustra* object-relational DBMS to the *Informix Universal Data Server*. New capabilities and utilities have been developed, including a spatial datablade that supports Nearest Neighbor queries.

1. Introduction

In the paper we presented at ADASS'98 (Cheung et al. 1999, Paper I), we described the fundamental principles used in the design of the Astrophysics Multispectral Archive Search Engine *(AMASE)*, that is based on Object-Oriented Data Base (OODB) methodology. *AMASE* is actually a *metadata* OODB that encapsulates: (1) the basic characteristics of astronomical objects; and (2) pointers to their observational data in distributed NASA data centers, into abstract objects. This enables searches for observational data from heterogeneous data sources to be done easily by scientific parameters. The URL for the *AMASE* homepage is `http://amase.gsfc.nasa.gov/`.

We started with a good data model to build *AMASE* for the scientific community. This proves to be invaluable when *Illustra*, the commercial Data Base Management System (DBMS) used in *AMASE* was acquired by another vendor and we needed to port the system to the *Informix Universal Data Server*. In this paper we describe our experience of porting *AMASE* into another DBMS.

In Paper I we also highlighted the complex procedures of loading the OODB knowledge base. Because many of data fields are hierarchical and interdependent in nature, a simple-minded approach of bulk loading new observations without any attempt to correlating them amongst themselves and/or cross-correlating them to objects already in the database will produce an unorganized mass of data that no indexing scheme can help. We spent the first two years of the AMASE project designing and implementing the core software for specialized loading, query and Web interface. The core software were also modified for the new DBMS. We shall describe new capabilities that result from this effort.

2. The *AMASE* Design

In the beginning, we decided that the *AMASE* database schema should model the structure of the scientific data as closely as possible and that the DBMS must have the following properties:

1. Multi-attribute Datatypes – named groupings of related fields that could be used to build complex types from basic scientific components.
2. Data Types with Repeating Fields – named collections of either distinct or non-distinct related values, i.e., sets or multi-sets.
3. Derived Types and Objects – the capability to define new types and objects by specifying additional types and access methods to previously defined ones, i.e., type/class inheritance.
4. User Defined Access Methods and Predicates – astrophysical aggregate, comparison, coordinate conversion and retrieval functions and procedures.
5. User friendly Query Language – an SQL-like language that would allow for fast query prototyping of ad hoc queries and reasonable query performance.

These requirements led us to choosing the object-relational DBMS, *Illustra*. Particularly because they provided server side support for extensible types (for performance as well as flexibility) and included a spatial type extension package ("data blade") as an add-on to their system. The spatial blade enabled us to create indexed spatial types from very beginning.

3. Illustra-to-Informix Porting

When Informix bought *Illustra* we viewed it as a very positive move for the longevity of *AMASE*. It is a much larger company and its relational DBMS engine has very good quality in terms of performance and reliability. They supported the *Illustra* product for one year while they developed a new version of their system called the *Informix Universal Server* (IUS), which integrates the object-oriented aspects of *Illustra* with their relational engine.

We discovered major differences between the two products when we ported our software. One important drawback was that Informix did not provide a spatial blade. But the University of Maryland Computer Science group took the opportunity to develop our own. In addition we extended the spatial capabilities beyond those formerly provided by the *Illustra* data blade. We extended the standard 2-dimensional types: point, line segment, rectangle and polygon to n-dimensions. We created these types and designed and wrote support functions which allow us to compare and index objects of the same dimensionality. As a result, we are able to support advanced spatial search capabilities in *AMASE* like the Nearest Neighbor query. This will allow users to search for objects by order of proximity to a given position.

There were also other surprises that required changes in our procedures. The syntax to bulk load objects into Informix is simpler but very different from that in *Illustra*. Sets and multi-sets are handled very differently, so are compound data types. *Oids* (object identifiers) and *poids* (parent object identifiers) are not supported in Informix. Several functions and SQL operators are also not supported by Informix.

We address these other porting issues in the following manner. We first redesigned the database schema to comply with Informix SQL syntax and to address the "oid, poid" and "set" issues. The syntax changes were easy. Informix supports all the types of objects the *Illustra* did, just in different ways. We added "oid" and "poid" fields to every object that had to be "joined" (correlated) with other objects to solve that problem. This impacted the loading software that is described below.

The difference in the implementation of sets between the two systems forced changes to every part of our code. *Illustra* implemented sets as relational tables subordinate to the main objects. The interface between them was transparent to the user (i.e. loading a parent object automatically loaded the underlying set.) But, for many queries it was necessary to access the sets directly, so their names were visible through SQL. Informix implements sets as multi-attribute fields (e.g., as arrays) within the objects themselves. Some of our sets have hundreds of elements so the *Illustra* approach was ideal for them, but others had only several so the Informix approach was ideal for them. Note that the Informix approach cannot work well for many elements. So, we examined all of our objects and created sub-objects out of those sets with many elements. This forced us to change a lot of both loading and query software. We had to change the parsing code for input data to take into account the differences in compound types. We either wrote or used alternative functions and SQL to replace the *Illustra* built-in functions and SQL.

The last major effort was modifying the loading software. Since the data to be loaded are generated by scripts and programs that process data from the NASA archives, we decided to not changes those. Instead, we wrotepre-processing programs that would convert *Illustra* readable data sets into Informix readable ones. We had to basically re-write the loading software itself, however, to accommodate the effects of our schema changes for sets, oids and poids.

4. Summary

The porting effort took approximately one year of one programmer's time to accomplish. Most of the time went into learning the Informix DBMS, then designing and implementing our own spatial blade. The Informix relational engine is much more flexible in terms of server and data placement and runs queries more quickly. So it is better for the long-term operation of *AMASE*. Informix provides good documentation and software support for designing and developing blades. The design of the advanced Spatial Data Blade with Nearest Neighbor query could not be fully realized without such support and documentation. The blade software can be made available on request to any group that would like to use it.

The *AMASE* project is supported by the NASA Applied Information Systems Research Program and is a joint effort between GSFC's Astrophysics Data Facility and the University of Maryland Institute of Advanced Computer Studies.

References

Cheung, C. Y., Roussopoulos, N., Kelley, S., & Blackwell, S. 1999 in ASP Conf. Ser., Vol. 172, Astronomical Data Analysis Software and Systems VIII, ed. David M. Mehringer, Raymond L. Plante, & Douglas A. Roberts (San Francisco: ASP), 213