

Issue Statement: Storing Linguistic Data in Databases

S. A. Miller, SIL International
Steve_Miller@sil.org

One of the most enduring challenges in creating linguistic software is the question of where to store or persist linguistic data. What database system should be used for linguistic applications?

Linguistic data lends itself better to an object-oriented approach to software engineering than to a relational approach. For instance, a text can be adequately stored in a relational database, but finding a morpheme within the string for analysis can prove to be a frustrating experience. The SQL language is not known for its string manipulation. In contrast, texts, words, and morphemes can be elegantly modeled as classes.

However, the most successful database management systems to date have been relational, not object-oriented. Microsoft SQL Server and Oracle are household names within the software industry, but object-oriented databases are virtually unknown. This forces software application developers to choose one of three options: 1) Mapping classes onto a relational database, to make use of solid software with strong vendors and a wide support base. 2) Finding an object-oriented database to suit their needs. 3) Finding an XML database to suit their needs.

All of these solutions have difficulties.

Mapping Classes onto a Relational Database

Database expert [C. J. Date](#) has argued that using relational databases can reasonably be used to store or persist object data, if the design is right. However, an experience known as the [object-relational impedance mismatch](#) has been common in practice. Object-oriented and relational systems are similar, but frustrations come in their subtle differences, a problem that [Scott Ambler has written](#) extensively about. Software developers repeatedly have difficulties making the leap from the paradigm they know to the other.

Moreover, relational databases are made for terabytes of data, while most linguists do their research on a laptop. Microsoft's [MSDE](#) (now [Express Edition](#)) has been a nice answer for this problem, and Oracle just announced its competitor [Oracle XE](#). However, both of these products are proprietary solutions. No one knows how long they will remain free, and many believe the open source model better enables scholars and linguists to do their work. Viable open source options that have been considered are [Firebird](#) and [PostgreSQL](#). For smaller applications, MySQL or SQL Lite may be alternatives.

Finding an Object-Oriented Database

A list of object-oriented databases can be found [here](#). Of the lot, [Progress ObjectStore](#) might be the best known, but it's proprietary. db4objects [became open source](#) a year or so ago, and [shows promise](#), but its query language is said to be difficult to use. None of the products have a wide support base.

XML Databases

Over [thirty native XML products](#) exist. [Berkley DB XML](#) has a lot to like, and is perhaps best known. However the XML layer sits on top of the Berkley engine. That layering is similar to the object-relational layering already discussed. Furthermore, “[a document is the engine’s atom of persistence](#); you cannot store or otherwise manipulate pieces of documents.” While this atom of persistence might be good for linguistic texts, most linguistic data has finer grain. Performance problems may well be an issue.