Name Access Control Module

The Problem

Most of today's libraries have switched over from a paper record catalog to a digital version. There are many commercial systems available, but no comparable open source system has yet to be developed. The eXtensible Catalog system being developed at the University of Rochester is hoping to fill this vacant niche. The Name Access Control Module (NACM) is one of the first parts of this eXtensible Catalog. Its purpose is to help import large amounts of Bibliographic and Authoritative records from the older commercial system into the new open source system. It does this by finding and recording matches between books' listed authors and the official list of authors known in the system. A series of special actions are performed in order to better match differently recorded names that all should point to the same person. This matching will allow people to search for a person and find all books either written by or about that individual.

The Solution

The solution to the problem is a multi-step process by which bibliographic records are matched to authoritative records. These three steps are:

- Authorized Name Matching - The matching of a bibliographic record's authorized (or official) name to the authority records held in the database. String transformations are applied to perform this match.
- Alternative Name Matching - The matching of bibliographic record's alternative names (such as pen names) to the authority records held in the database. Again String transformations are applied to perform this match.
- Generated Name Matching - If no records are found in the existing authority records, create a new authority record and match to it. This allows future bibliographic records to be matched to the same person, even though an authority record did not previously exist.

The Product

Team MetaCat uses an iterative and evolutionary software design process. It starts with an initial requirements elicitation and analysis phase. This is followed by the creation of an overall design architecture. At this point the team moves into a loop; designing small aspects of the final system, implementing them, testing the results, and then moving on to new features as needed.

The overall architecture is decided after all of the project requirements are gathered and analyzed. An example GUI is built on top of a stable API framework. This API allows other components of the eXtensible Catalog to make use of the NACM with ease. The API consists mainly of two parts, an importer for processing given records, and a matcher for associating bibliographic records to authoritative ones. Both of these components interact with the core database via a DAO (Data Access Object).

The data stored in the project's persistent database is not overly extensive. The entire database focuses around three main tables. One table stores bibliographic information. Another table stores authoritative information. The final table stores matched links between the first two, indicating relationship and their associative information. There are a few other tables that represent complex data, such as the fields in a person's name that are used by the main three to accomplish their functions.

eXtensible Catalog