CMSC424. Class Project

Phase 1 (DB design) due: April 1, 2008
Final project due: May 1, 2008

Overview

For this project you will have to develop a database for storing professional information about a group of people. The system will maintain information typically found in a resume for a set of people, together with information connecting these people to each other (either by co-authorship or mentorship relationships). One use of this system would be to mine these data for different types of information, such as co-authorship distance between different people, impact of a certain researchers (e.g. publications by current and past advisees), etc. Also, this system should be able to create a resume in one of several specified formats for a selected person stored in the database. The system you develop should be able to import information from external data sources, such as citations stored in BibTeX format.

Data to be stored

You will need to store the following types of information:

1. **Publications.** You should be able to store any type of publication and associated information that can be represented in a BibTeX record (for more details check BibTeX documentation). Author information, journal/conference/book, date of publication, etc. should be stored in your database.
2. **Skills.** These represent different proficiencies a person might have, whether a specific programming language, software system, or foreign language.
3. **Employment history.** The database should allow you to store a person's job history, including employment dates and a brief summary of job responsibilities and achievements.
4. **Education.** You should store education history: schools, degrees, majors, GPAs, as well as dates attended.
5. **Mentoring.** You need to be able to store a list of people mentored at some point in time (both current and past), including dates and additional information about the actual project.
6. **Honors.** Store information about any honors and awards a person might have received, including dates, name of award, and description of what the award means.
7. **Personal information.** This should be generic information that you might want to add, such as sports, statement of goals, etc.

Additional requirements

Interaction. The system you develop should allow a user to add information to the database (e.g. import a new file) as well as to add or edit individual records for selected datatypes (e.g. skills, employment history, CV format) using a simple form.

Annotations. It is useful to allow every object in your database to be annotated with additional information that can be displayed when requested. For example, for a particular job, you might want to report specific job responsibilities, or simply list the job title, depending on specific CV formatting requirements.

Categories. For all the datatypes described above you also need to store information about one or more categories they belong to. For example a publication could be distinguished based on whether it's in a conference or journal, whether it is an invited talk or poster presentation, etc. These categories will be used to decide which objects will be printed in the report and how they will be formatted. Some CV formats may require, for example, all publications in one single list, while others might request that you separate out the refereed publications from the unrefereed ones. Note that the best way to implement such categories is through the use of a controlled vocabulary: store a separate table containing all possible category names for publications, service, etc. and use DB constraints to ensure that the values associated with individual objects correspond to entries in this table.

CV formats. Your database needs to store information about how different types of CVs/resumes need to be formatted. A possible implementation would involve storing information about the order in which individual categories are reported and specific formatting requirements (e.g. heading, font, citation format, # of items, or # of years to report, etc.). Note that there are many ways to display citations - you will need to implement several formatting rules.

External data. Your program needs to be able to import BibTeX files containing publication information, or other data in XML format.

Global reports/Data mining. You should be able to construct several types of reports about the data stored in the database. These include:

- Histogram of #publications/year.
- Minimum, maximum, average time spent employed.
- Gaps in employment
- Amount of "in-breeding" in a person's publication lists: level of overlap between the set of a person's co-authors and the set of their co-authors' co-authors.
- Competitors/fans: the list of authors on papers that cite a researcher's papers but that are not their co-authors. This list should be ranked by number of publications - those who frequently cite the work are either competitors or huge fans.
- Citation network: find authors that are a certain co-authorship distance from other authors.
- Impact: find number of publications, or number of advisees, of all advisees of a certain person.

Deliverables

The project will be run in two phases:

Phase 1, due April 1, 2008 (40 pts). You need to provide us with an E-R diagram and a relational schema describing your database. In addition, you will need to create the necessary tables in a database system, and populate an initial version of the database with data we will provide.

Associated with this should be a report describing the specific assumptions you make about the data, and the various constraints you might want to enforce. Also describe how you would retrieve the types of information required above from this database. Finally, provide a brief description of the user interface you envisage.

Phase 2, due May 1, 2008 (60 pts). You need to provide us with a fully implemented system, together with a report describing your implementation. You will be asked to demo this system to Sharath.

Note that a week before the project is due we will provide you with a couple additional queries your system will have to support. If you've done a good implementation job you should be able to incorporate these queries without much trouble. Make sure you design an extensible system.

Notes:

- Your reports for the two phases should clearly indicate the contribution of each team member.
- You can use any programming language you like, however you need to make sure that your software works on the Grace machines.

Submission information: Submit your project and reports by e-mail to me (mpop@umiacs.umd.edu) and Sharath (sharath@cs.umd.edu). The reports should be in PDF format. The software should be either in a .zip file or in a .tar.gz file.

Grading criteria

The grading will primarily focus on the following broad requirements:

- Good relational design
- Good documentation
- Code clearly written and well commented
- Good user interface
- System is extensible
- Creative design

Skills

The goal of this project is to teach you to design and implement a database according to an incomplete/imperfect specification. If anything is unclear, please ask questions. Frequent communication between the developer and customer is key to successful design.

Among the skills I hope you will acquire during this project are:

- ability to interact with the customer and team members during the design and implementation of a database system
- development of parsers that can be used to populate a database with data from different types of flat files (BibTeX, XML, etc.)
- ability to design a database system that can be easily extended as user requirements change
- development of simple form-based interfaces for interacting with a database

Additional information

We will provide additional information on the website that can help you in your project: links to documentation describing the different file formats, test data, etc. Please be inquisitive and not rely exclusively on these. There are many online resources that can be useful to your project. Please ask questions if anything is unclear.