Abstract

Swoop takes contacts from multiple locations such as Facebook, LinkedIn, MySpace or your computer and gathers them in one location so users can easily obtain information of all their contacts.

System Overview

The system consists of a database, a parser, a connection manager and various front end programs that display, analyze and gather data. The database and parser work together to store and maintain data consistency. Together they make sure that the data is in the correct format and can be stored and displayed correctly. The connection manager makes sure the system is scalable. The user programs provide an attractive frontend interface for users. The front end is also in charge of doing analysis to directly gather and organize the data.

Database

The database stores all contact information gathered. Robustness was a major consideration during the design of the database, it had to add in new fields without requiring any changes to its basic structure. The data also has to be identified so they can be formatted properly when displayed and stored efficiently. To achieve this, all data are labeled as a specific field and each field has a data type. Data is stored as strings in the database.

Parser

The parser is in charge of converting data gathered online into a standardized format that can be efficiently stored into the database. Since each type of data has to be parsed in a different way, there is a specific parsing specification for each type. The specification makes sure all types, for example phone numbers, are parsed into the same format no matter how it is formatted when it is gathered. It also makes sure all the data to be displayed from the database is formatted into a standard format to display to the user. Thus keeping the whole user interface consistent.

Parser Assignment Details

Since information from the internet is very dynamic, we expect our field/type pair-ups and specifications to change fairly often. In order to not burden client programs with constant updates to keep their field/type pair-ups consistent with the server’s, only the field data is given to the clients. When the client sends the data in, the parser assigns the field to the most up-to-date type parser. In order to do this programmatically without many access to the database to find the newest specification, the assignment portion of the parser is automatically generated using code instead of manually written, this decreases the chances of human error since the field/type pairs only have to be changed in one location for it to work with the whole system.

Connection Manager

Because the traffic of the system is expected to be very high, a very fast and scalable connection manager was needed. The manager preloads multiple parsers into memory. When data is received, the manager assigns it to one of the parsers, when the load is high, the manager initiates additional managers so the amount of data can be processed very quickly. The amount of database read writes are minimized, so the only time the database is accessed is when the data is actually stored or loaded. Making the number of database read/writes per request to be exactly 1.

Linking Engine

The last part of the system is the Linking Engine. A link is a connection between two user objects. For example if Lloyd has an account in both LinkedIn and Facebook (two different social networks), the two different user accounts will create two user objects within the system. A link between the two accounts will tell the system that these two objects are actually the same person. Since many different sources of information are expected, it will be very tedious for links to be created manually. Thus an AI Linking Engine was created.