4440 Project Proposal

Martin Ahrens
Peter Rosegger
Sovandy Hang
Yuan Liu

1. Motivation and Objectives

People across the world are increasingly accessing the web to find information on politics, sports, investments, and many other news topics. While much of this data is in traditional paragraph format, certain types of information are expressed in a tabular format instead. The general public is familiar with finding sports line scores and boxscores, as well as stock market data, in tables. From day to day, the formatting of these tables does not change, but the data inside them does (to reflect the previous day's games). Websites like ESPN.com maintain templates for, for example, a baseball boxscore, then simply fit the data inside the template on a daily basis. We propose to begin work on a system that would work in the opposite manner: starting with a template filled in with timely data, and extracting the data from the template so that it may be stored and manipulated later.

The first steps, our project for CS 4440, will focus on extracting player statistics from historical basketball box scores found on NBA.com. We will create a system that will find the source HTML from the pages that contain the box scores, parse the HTML, then store the statistical data from those box scores so that it may be manipulated and represented later. Although the specific statistical data that will result from our demonstration is already widely available, the driving ideas and methodology behind the work will be far more significant.

Ultimately, after work that will take far longer than the allotted time for this project, the system would be able to extract and aggregate data that is widely available on a day-to-day basis, but not as a large data set spanning several days (or even years), and make it publicly available. For example, a user could set up the system to extract weather.com's hourly weather data for Atlanta each day, store that data in a database, and then manipulate that data at a later date.

This ability would be particularly useful for several reasons, the first of which is that in many cases such data is not widely available in aggregate format. And even if it is available, actually finding it in a useful format adds another layer of complexity for the user. For these reasons, finding hourly weather data for Atlanta over the past year, for example, would be nearly impossible for an amateur. Our system will allow the end users of the data to decide what is important and what is not, eliminating much of the difficulty in finding and working with data sets.

2. Related work

Our work has three parts:

1. Extracting data from HTML websites
2. Repeating the process to build temporal data
3. Manipulating sports statistics

Repeatedly accessing web data and manipulating sports statistical data are well-known and mainstream ideas. Extracting data from HTML is a well-covered area in research, but extracting data from HTML in order to build temporal data in combination with these two other areas is what makes our project unique.
In 1998 Dayne Freitag of Carnegie Mellon University published Information Extraction from HTML: Application of a General Machine Learning Approach. But this research focused more on the artificial intelligence aspect than HTML extracting data from HTML.
Hiroshi Sakamoto, Hiroki Arimura, and Setsuo Arikawa published research in 2000 called Extracting Partial Structures From HTML Documents, but again, this work focuses more on deducing the type of data stored in the HTML than simply storing data from known structures.

3. Proposed work

The application is broken down into four main components. First Web page is converted into a file format readable to the parser. This can be done with existing open source tools, for instance, cURL and Snoopy. Second the parser reads data from the file, using predefined XML metadata as the guideline. Data is stored in a text file for it to be imported into database. Third data is imported into database. Forth is the data representation component. One of the main objective of this application is to be able not only to gather data, but to present it in a way easy to understand and beneficial to users. Statistical analysis of data will determine how useful the application from end-user perspective. It's our intention to cover as many as analysis variables as we can. To make it more user friendly, the result of analysis will be graph-oriented. For instance, pie charts, bar charts, and plots will be used in addition to numerical data. We figure that Scalable Vector Graphics (SVG) will be a good fit for our project. Search is one of the main aspects of data representation and will be embedded in the application. Our search will cover the variety of criteria related to individual player, team, etc... Architecture of the application showing the relationship between the four components is shown in Figure 1. Information flow diagram of the application is shown in Figure 2.

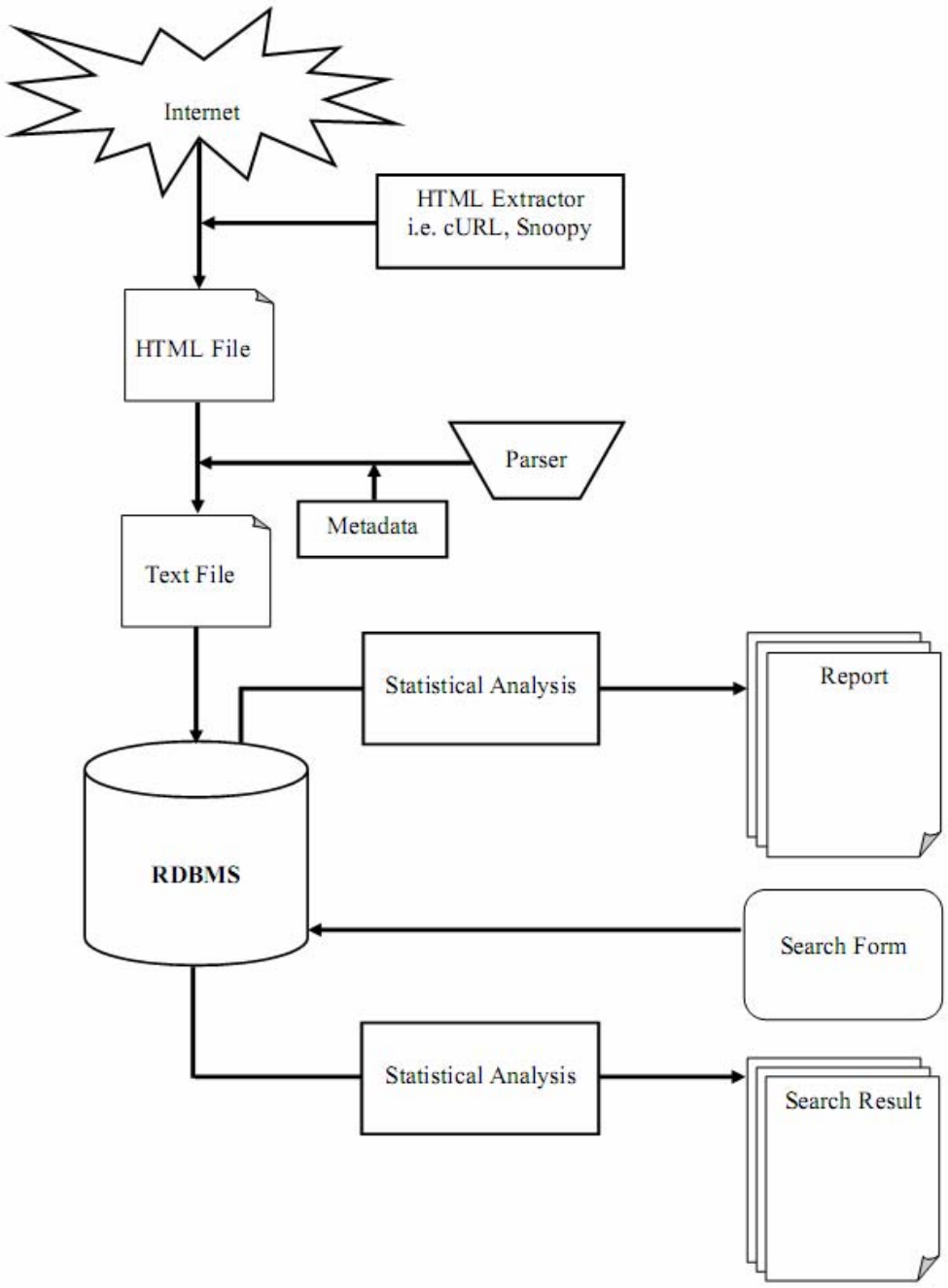

Figure 1. The architecture of the application, showing the relationship between components.

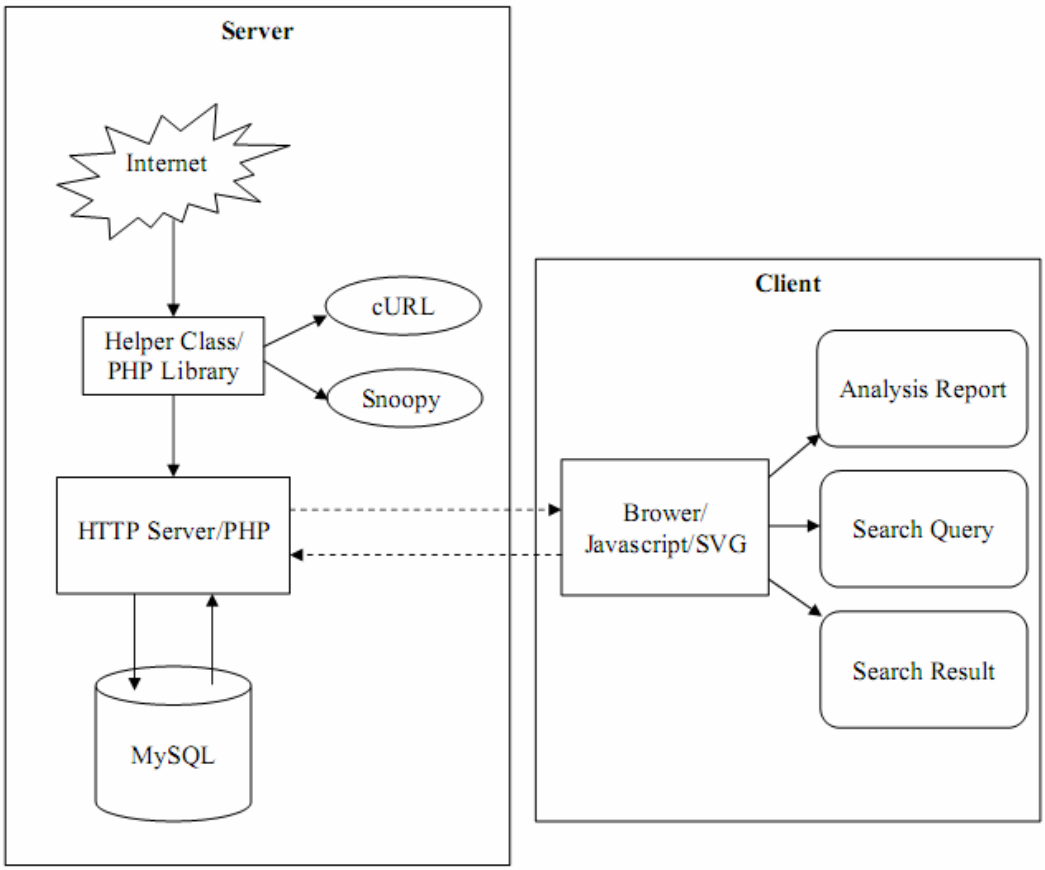

Figure 2. Information flow diagram of the application.

4. Plan of action
    1. Brief description of project progress
        1. The project will be mainly separated to four major sections: The preparation section, The Database construction section, the production/coding section, and The Finalization/Documentation section.
        2. How weeks are been planned:
            1. Week 1 - software preparation, make sure everyone can start
            2. Week 2 - Database paper works and implementation.
            3. Week 3 - 7 Production/Coding period
            4. Week 8 - Finalization/Documentation
    2. Software/Hardware Requirement:
        1. Software Requirements:
            1. A Web host that supports PHP/MySQL. www.1and1.com is a perfect host for this project
            2. PHP. This is provided by the Web host
            3. MySQL. This is provided by the Web Host
            4. Komodo Edit 4.2. This is the software to write html/php code, can be obtained freely at http://www.activestate.com/Products/komodo_edit/. Platfo rm supported: Windows, Linux, Mac OS (Peter may likes it, because he uses a Mac laptop).
            5. A website that publishes sports statistic data. www.nba.com www.espn.com.
        2. Hardware Requirement:
            1. A personal Computer. The performance of this computer is equivalent to the machines in the College of Computing.
    3. Weekly Schedule
            1. Week 1 (Oct 1st - Oct 5th):
                1. Setup PHP/MySQL, make sure everyone has access to database.
                2. user/login implemented
                3. Early Database assessment done
            2. Week 2 (Oct 8th - Oct 12th):
                1. Database ER diagrams implemented
                2. Database Model implemented
            3. Week 3 (Oct 15th - Oct 19th):
                1. User Interface implemented
                2. HTML parser for PHP implemented
                3. Extract data from HTML to local file
                4. Continues access website to pull data
            4. Week 4 & 5(Oct 22th - Nov 2nd)
                1. Player statistic query processing implemented
                2. Player statistic analysis implemented

5. Week 6 & 7(Nov 5th - Nov 16)
    1. Player stats graphical representation implemented
    2. Player stats output implemented
6. Week 8 (Nov 19th - Nov 23th)
    1. Final Documentation done
    2. Prepare for Presentation

5. Evaluation and Testing Method
    - Extracting data from HTML sources of websites
    - Enumeration of relevant HTML pages for mining will be an important step of the process
        - Examination by-hand of generated links may be validated when performing initial sweeps across a website
        - Testers will generate relevant page listings and compare suggested URLs for mining
        - When a sufficient percentage or all of the relevant pages are generated, this step can be completed

    - Our algorithm will extract relevant HTML source data as an initial step
        - Algorithm to handle individual pages will be tested with erroneously formatted pages in a variety of forms
        - Several pages will be calculated by hand and compared to data aggregated by the application
        - The effect of rescanning pages multiple times will not introduce errors into the data
        - The capability of handling pages with changing URLs will need to be handled in an as-needed basis for websites

    - Building temporal data
    - There will be a speedier way to test temporal data acquisition than sitting at a computer for 5 hours updating data
        - Test data and processes will be developed to simulate time passing for a temporal mining to simulate hours in seconds
        - Having configured data for running tests in a timely manner may require the addition of a control file for configuring the tests
    - Data integrity will again be compared by hand for an initial set of data to the algorithms output

    - Manipulating sports statistics
    - The presentation of graphs, some minimally complex aggregate queries, etc. will be implemented and verified by hand for several cases

6. Bibliography

Freitag, Dayne. (1998). *Information Extraction from HTML: Application of a General Machine Learning Approach*.
Sakamoto, Hiroshi. Arimura, Hiroki. Arikawa, Setsuo. (2000). *Extracting Partial Structures From HTML Documents*.