Advanced Programming

DBLP Query Engine

Project Report

Subramanyam Dantu (2015103)
Tanmay Goyal (2015106)

Monday 28th November, 2016

1 Introduction:

DBLP is a computer science bibliography website. Starting in 1993 at the University of Trier, Germany, it grew from a small collection of HTML files and became an organization hosting a database and logic programming bibliography site. DBLP listed more than 3.4 million journal articles, conference papers, and other publications on computer science in July 2016, up from about 14,000 in 1995.

We designed a parsing mechanism to read the huge dataset and display different outputs based on the user's query inputs. We have tried to incorporate design patterns in our project.

2 Approach to the Project:

We began our project by trying to identify a suitable XML Parser in Java that could parse an XML file that was close to 2GB in size. After a proper analysis of the parsers that were available, we decided to move forward with the SAX Parser. The SAX (Simple API for XML) Parser is an event-based parser. Applications using SAX receive event notifications about the XML document being processed: an element and attribute, at a time in sequential order starting at the top of the document, and ending with the closing of the ROOT element.

After successfully being able to parse the huge dataset that was retrieved from the DBLP server. We started simultaneously designing the GUI and the back-end interface for each user query. We have used a wide variety of layouts to incorporate the design. These include BorderLayout, CardLayout, FlowLayout, etc. For the back-end design we made different classes for each query so that it simpler to access and parse the data based on the requirement.

3 Entity Resolution:

We have performed entity resolution by first finding out all the alias' of all authors from the 'www' tag. Then we stored it in an ArrayList, which contained all the alias' of a single author.

To use entity resolution in Query 1, we first matched the author input string with each element of the ArrayList data. Upon matching, we stored all the alias' of that author and then subsequently matched it in Query 1.

For Query 2, we equated all the alias' of an author in HashMap and added their respective publications.

4 Group Member Contribution:

- Subramanyam Dantu - Designed the front-end GUI, helped in integration of front-end and back-end, generated doxygen comments.
- Tanmay Goyal - File Parsing, back-end, entity resolution, and helped in integrating back-end to front-end.

5 References:

1. http://www.mkyong.com/java/how-to-read-xml-file-in-java-sax-parser/ - For explaining how to use SAX parser for parsing XML file.
2. http://stackoverflow.com/ - For all doubts that came up during designing the system.
3. http://www.stack.nl/ dimitri/doxygen/manual/docblocks.html
4. http://dblp.uni-trier.de/faq/What+do+I+find+in+dblp+xml