**Report # 1**

**(May 23rd-June 1st, 2007)**

**Raghav Krishna Padmanabhan**

**TANGO,**

**Doc Lab,**

**Rensselaer Polytechnic Institute.**

It is a well accepted notion now that to realize Semantic Web, a repository of ontological data is the first step. And since creating an ontological repository manually is time intensive, we attempt to create ontologies automatically. We begin the task with the data stored in tables. To manipulate and retrieve the data, we need to convert all the information we gather into a standard notation.  The standard notation chosen in our case is Wang's notation [1]. I started with reading the report "Interactive Wang Notation Tool for Web Tables" by Piyushee Jha and Dr.Nagy which describes the development of a tool which converts an HTML table into Wang's notation automatically. The interactive tool has been created in Matlab to obtain Wang's notation. As the tool interacts with the users, it is required that the users sufficiently understand and recognize table structures and patterns and differentiate between the category/sub-category cells and the content cells. In tables which are not completely 'intuitive' (which Piyushee calls foreign tables) , there might be some patterns in the structures like merged cells which delineate the category cells from the content cells. There are a few other aspects like learning and further automation which needs to be worked upon to make the tool more robust.

What is Wang's Notation ?

Wang's notation is a two part notation of a table:

(a) The C notation which is a finite set of labeled domains (Headers, Sub headers etc.)

(b) S (delta) notation which represents the set of individual values corresponding to C .

Then I studied the evolution of the Wang's Notation tool converter. The Version 1 required a lot of manual intervention with the user having to copy and paste the table into an Excel spreadsheet and having to type in the responses in the Matlab command window.This version also had its limitations with respect to the levels of the table.

The Version 2 of the tool improved upon its first version by automating the delta notation. This was a step that greatly reduced the amount of time in generating the delta notation automatically. This was realized by the introduction of 'Symmetric Tables'. Symmetric tables are the tables in which every delta cell has all the categories it is related to either in the same row or same column. The delta notation was generated automatically by a program written by Piyushee which checked all the delta cells and

found out the category or sub category cells by just traversing the same row and the entire column.Then the GUI (Graphical User Interface) part was added to the program which enabled the user to click the cells of the table generated rather than typing the answers to the questions raised by the program to the user.

The third version of the tool makes use of the concept of 'Trees'. Since i was not really conversant with the terminology, i went through some online material about trees and binary trees and their traversals[2](I found the concept of trees very intriguing and i plan to peruse more material on them in future). The table categories and sub categories are converted into trees and the traversal of the trees in pre order was similar to the Wang's notation order for categories. For this, the table is represented in indented notation and later converted into a Table of Contents representation as it is easier to manipulate them in Matlab. It is easier to manipulate binary trees than general trees and hence the general trees are converted into binary trees. The notation followed to label the nodes in the trees was a little difficult to fathom and not entirely clear to me. It needs to be seen if there can be another method for identifying the position of the node in the tree more easily. Piyushee's Matlab program performs a pre order traversal of the converted binary tree and also generates the notation with the curly braces and brackets. also, further improvements were made in the program like

(a) reduction of clicks by sweeping all the cells within the clicked cells,

(b) making it more 'user-friendly' by a color coding for clicked cells (red for clicked cells and blue for cells identified by the system.)

(c) implementing a post-editing tool which aids the user in correcting the generated indented notation.

The system proceeds to create the category and delta notation once the user gives the 'GO' signal.

In the next version of the Wang's Notation tool (Version 3.5), the input file being fed into the MATLAB tool was changed from the Excel spreadsheet to a file in ASCII format. To convert the input HTML file into an ASCII file, the open source code was provided by Cui Tao of Professor Embley's lab. It made use of the W3c packages for DOM tree parser (org.w3c.dom and org.w3c.tidy). This program written in Java converted the input file into a Document Node which was processed to generate the ASCII versions of it. However, for tables with spanned headers, the input to the Matlab code (i.e. the output of the Java program) had to be modified with the attributes of the header specified.

To correct this problem, I perused material on the Document Object Model. According to [3], the DOM is an API (Application Programming Interface) for valid HTML and XML documents which defines the logical structure of the documents and the way they are accessed and manipulated. In DOM ,the tables are converted into a tree with node and then manipulated. I managed to successfully implement this change and add the required attributes to the output file. In addition, the program had to be 'built' and the resulting jar file had to be saved in the file system as a text file using the 'java-jar' command. I was able to reduce one step by directly obtaining the output file into the file system by introducing an output stream and printing the data into the stream as and when the program printed the output in the console.

While trying to learn about DOM trees, I started reading the research paper by Dr.Embley and Cui Tao of BYU - "Table Interpretation by Sibling Page Comparison" [4]. The paper started off with explaining what tables are, how information is stored in them and the complex table structures like Conjoined tables, Nested tables. Sometimes they maybe arranged to fit the space available and labels may not always be placed on the top or left or they may span across columns. All these forms of tables might be easily interpreted by a human mind. But this aspect of flexibility in table formation and arrangement makes it a challenge for automated table interpretation. This paper explored the problem of automatic table interpretation of complex table structures using the concept of Sibling Tables. Sibling Pages are pages from the same website and with similar structures. The tables in the sibling pages are called sibling tables. This paper also makes use of Wang's notation to interpret the table information. The paper then talks about how to identify tables in a web-page using the HTML table tags and then proceeds to the aspect of matching tables in sibling pages by converting them into DOM trees.

I have identified a few technical papers from Dr. Nagy's homepage which I plan to read and analyze this week, which I think will give me a better understanding about tables, their interpretation and processing.

1. A Tabular Survey of Table Processing

2. Table Processing Paradigms- A Research Survey

3. A discussion on Ontologies by Yuji Tijerino

I also plan to complete reading the WNT report prepared by Piyushee in this week.