

**UNDERSTANDING, INTERPRETING AND QUERYING
WEB STATISTICAL TABLES**

by

Chi Hang (Philip) Leung
B.Sc., Simon Fraser University, 2000

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

In the School
of
Computing Science

© Chi Hang (Philip) Leung 2005

SIMON FRASER UNIVERSITY

Spring 2005

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without permission of the author.

APPROVAL

Name: Chi Hang (Philip) Leung
Degree: Master of Science
Title of Thesis: Understanding, Interpreting and Querying Web Statistical Tables

Examining Committee:
Chair: Dr. Fred Popowich

Dr. Wo-Shun Luk
Senior Supervisor

Dr. Jian Pei
Supervisor

Dr. Ke Wang
External Examiner

Date Defended/Approved:

April 19, 2005

SIMON FRASER UNIVERSITY



PARTIAL COPYRIGHT LICENCE

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

W. A. C. Bennett Library
Simon Fraser University
Burnaby, BC, Canada

ABSTRACT

Extraction of information from tables published on the Web is made less complicated because of easy identification of the text inside a table cell. In this thesis, we propose, and have implemented, a scheme which not only understands the contents in a statistical table, but is also able to convert them into a multidimensional database which can then be fed into an off-the-shelf system for querying and data integration. By carefully interpreting the intention of the table author via the visual cues embedded into the HTML text, and the layout design of multidimensional database modelling techniques, our system can successfully classify the keywords into semantically distinct dimension hierarchies, without any domain-specific knowledge, or machine learning. Experiments on a set of real-life statistical tables have confirmed the validity of this approach. Experiments on a set of real-life statistical tables have confirmed the validity of this approach.

Dedication

To

my dearest parents

and Joanne

for their love and support

ACKNOWLEDGEMENTS

I would like to sincerely thank my senior supervisor, Dr. Wo-Shun Luk, for his guidance, support, enthusiastic discussions and advice during my research. This thesis would not be made possible without him. I am thankful to Dr. Jian Pei who patiently read my thesis and provides valuable feedback and suggestions for improvement. My thanks also go to Dr. Ke Wang who graciously serves as the external examiner and provides comments for my work. I would like to thank Dr. Fred Popowich for being the Chair of my defence despite his tight schedule.

TABLE OF CONTENTS

Approval.....	ii
Abstract.....	iii
Dedication.....	iv
Acknowledgements.....	v
Table of Contents.....	vi
List of Figures.....	viii
List of Tables.....	ix
Chapter 1 Introduction.....	1
1.1 Table and Table Processing.....	1
1.2 Statistical Table.....	5
1.3 Table Processing – Current Approaches.....	7
1.4 Table as Multidimensional Object.....	8
1.5 A New Approach to Statistical Table Processing.....	11
1.6 Organization of the Thesis.....	12
Chapter 2 Representation Forms of a Table.....	13
2.1 Component Model.....	13
2.2 Multidimensional Data Model.....	16
2.3 Multidimensional Expression (MDX) & OLAP System.....	19
Chapter 3 Table Processing.....	22
3.1 Recognizing Table Components.....	23
3.2 Recognizing Dimension Hierarchies.....	27
3.2.1 Processing Column Heading.....	28
3.2.2 Processing Row Heading Region in a Table Body Section.....	29
3.3 Recognizing Table Body Sections.....	33
3.4 Table Integration.....	38
3.5 Linguistic Processing.....	39
3.6 Building Multidimensional Database.....	41
Chapter 4 System Design and Experimental Performance Evaluation.....	43
4.1 System Overview.....	43
4.2 System Architecture.....	44
4.2.1 Internal Structure of Web Table.....	45
4.2.2 Components.....	47
4.3 Performance Assessment.....	52
Chapter 5 Related Work.....	58
5.1 Statistical Table, Multidimensional Object, and OLAP.....	58
5.2 Web Tables.....	59
5.3 Recognition of Table Components.....	60

Chapter 6	Conclusion and Future Directions	62
6.1	Major Contributions.....	62
6.2	Future Directions.....	63
Bibliography.....		65

LIST OF FIGURES

Figure 1.1 – Examples of Non-Genuine Tables in Department of Computing Science web site of SFU 2

Figure 1.2 – Example of Genuine Table..... 3

Figure 1.3 – The Multidimensional View of Table 1..... 9

Figure 1.4 – Dimensions in an OLAP Database.....11

Figure 2.1 – Transformation of Table.....13

Figure 2.2 – Wang’s Physical Model of a Table14

Figure 2.3 – Our Component Model for Table Layout15

Figure 2.4 – Table 1 in Multidimensional Form17

Figure 2.5 - Multidimensional Form in 4 Dimensions19

Figure 3.1 - Relative Cell Positioning in Two Neighbouring Rows28

Figure 3.2 - Determining Hierarchical Relationship and Building Conceptual Columns33

Figure 3.3 - Merged Hierarchy from Row Headings35

Figure 3.4 - Two Hierarchies from Row Headings.....36

Figure 3.5 – Table with Drop-Down Boxes.....39

Figure 4.1 - System Overview (User Level)44

Figure 4.2 - Internal Architecture45

Figure 4.3 - A Simple Web Page Containing Table.....46

Figure 4.4 - Hierarchical Tag Tree of the Simple Web Page47

Figure 4.5 - Screen Shot of Web Table Miner48

Figure 4.6 - Simple Web Table Using CSS50

Figure 4.7 - A Statistical Table and Its Corresponding Virtual Grid.....51

Figure 4.8 - Star Schema of OLAP Data Cube.....52

Figure 4.9 - An Example in Which Visual Cues Fail to Work56

LIST OF TABLES

Table 1.1 – A Portion of the Statistical Table Shown in [PMWC2003].....	8
Table 2.1 – Table 1.1 in Component Form.....	16
Table 2.2 - A Table that Contains Dimensional Relation	18
Table 3.1 - A Table where Text is Part of the 'Slicer' Dimension.....	25
Table 3.2 - A Table whose second column consists of Row Headers	31
Table 3.3 - Another Table whose second column consists of Row Headers	30
Table 3.4 - An Example of Table with Multiple Table Body Sections.....	33
Table 3.5 - A Sample Statistical Table for Scenario 2	36
Table 3.6 - A Table with TBS's Having More Than One Hierarchy.....	37
Table 4.1 - Performance Evaluation of the Web Table Miner	55
Table 5.1 - A Table about Tour Packages.....	60
Table 5.2- A Table about Stock Quotation	60

CHAPTER 1 INTRODUCTION

Information extraction (IE) is about attaching words and numbers (values) in a document to some semantic labels (attributes), and storing the derived attribute-value pairs in a database [CM2004]. The document may be in the form of free text, or semi-structured data. Among various forms of the semi-structured data, table is very popular in IE research. Mining information from tables poses many interesting research problems due to the combined consideration of both language and layout [HN2000]. Lately, tables on web pages have become the object of choice for table mining researchers because of the ease in extracting text out of a cell. The focus of this thesis is on a special type of web table – statistical table that is regularly published by all levels of governments and publicly funded agencies in many countries.

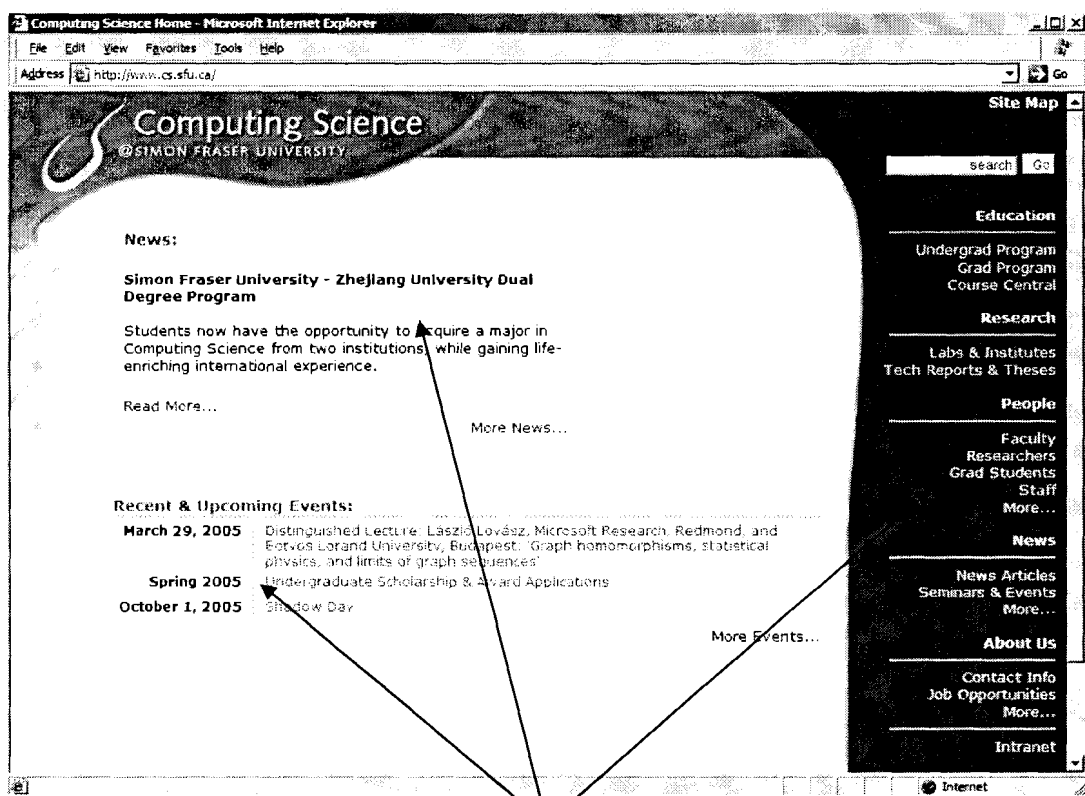
In this chapter, we discuss about the current and our approach to table processing, and in particular, statistical table processing.

1.1 Table and Table Processing

A table, according to the Oxford English Dictionary¹, is: “An arrangement of numbers, words, or items of any kind, in a definite and compact form, so as to exhibit some set of facts and relations in a distinct and comprehensive way, for convenience of study, reference or calculation”. It can also be defined in spatial terms, i.e., as a two dimensional cell assembly for presenting information. These definitions are sometimes too general for any meaningful table understanding. In [WH2002], some tables are

¹ J. A. Simpson (Editor) and Edmund S. Weiner, *Oxford English Dictionary*, Oxford University Press, 2nd edition, 1989.

considered to be “non-genuine”, because it is merely a mechanism for grouping contents into clusters for easy viewing, in contrast to “genuine” table which contains relational data, in the same sense as a table in a relational database. Figure 1.1 and 1.2 give examples of both table types. In Figure 1.1, tables are used to layout information on a web page in an organized way. In fact, one would not even notice the use of table because its borders and lines are made invisible. While in Figure 1.2, a “genuine” table is shown.



Non-Genuine Tables

Figure 1.1 – Examples of Non-Genuine Tables in Department of Computing Science web site of SFU ²

² <http://www.cs.sfu.ca>, April 2005

Symbol	Last	Change	Volume	Avg Vol	Bid	Ask	Name	Prev Cls	Open	52-wk Range	EPS (est)	P/E
NYJ	1,000.45	-25.55	N/A	N/A	N/A	N/A	NY JONES INDUSTRIALS	1,016.74	1,039.06	1,750.00 - 2,101.00	N/A	N/A
SPY	1,299.19	-29.07	N/A	N/A	N/A	N/A	S&P 500 COMPOSITE	1,299.19	1,296.29	1,139.02 - 1,411.00	N/A	N/A
SPYSE	9,277.12	-132.49	N/A	N/A	N/A	N/A	S&P/TSX COMPOSIT	9,409.51	9,411.63	8,098.06 - 9,968.41	N/A	N/A
YHOO	32.46	-1.09	27,053,032	20,946,181	30.45	34.47	YAHOO INC	33.46	32.95	25.01 - 39.79	0.58	57.69
NT	2.55	-0.05	27,313,200	17,073,090	N/A	N/A	NORTEL NETWKS CP H	2.60	2.56	2.59 - 5.91	0.08	32.50
NTIQ	3.18	-0.05	14,219,148	10,137,872	3.18	3.18	NORTEL NETWORKS C	3.23	3.20	3.05 - 8.00	0.204	15.83
MSFT	24.46	-0.38	100,580,272	66,938,772	23.46	25.47	MICROSOFT CP	24.84	24.55	23.82 - 30.20	0.92	27.00
GOOG	185.00	-6.45	11,584,509	10,388,000	0.01	9,000.00	GOOGLE	191.45	190.00	95.96 - 216.80	1.46	131.13
COGN	40.70	-9.33	832,538	803,954	0.01	9,000.00	COGNOS INC	41.03	40.75	28.90 - 47.40	1.47	27.91
BOJ	26.93	-1.17	512,675	813,500	0.01	9,000.00	BUSINESS OBJ SA A	28.10	27.25	17.15 - 29.59	0.52	54.04
INTC	22.12	-0.37	92,617,072	63,451,136	20.12	24.13	INTEL CP	22.49	22.15	19.64 - 29.01	1.16	19.39
AAPL	35.35	-1.91	61,798,220	30,561,090	35.34	36.37	APPLE COMPUTER	37.26	36.61	12.745 - 45.44	0.90	41.40
IBM	76.70	-6.94	27,934,400	4,878,590	N/A	N/A	INTL BUSINESS MAC	83.64	79.49	81.90 - 99.10	5.00	16.73
ADBE	60.55	-2.82	3,953,800	2,475,500	59.54	60.74	ADOBE SYSTEMS INC	63.48	62.75	39.32 - 68.95	1.92	33.06
SIRI	5.15	-0.15	27,721,100	43,929,000	4.14	6.25	SIRIUS SATELLITE	5.30	5.25	2.01 - 9.43	-0.57	N/A
TD.TO	50.27	-0.58	1,761,264	1,688,638	50.25	50.27	TORONTO-DOMINION	50.85	50.85	42.54 - 51.70	3.41	14.91
PIXR	94.39	-1.32	929,553	551,681	91.99	96.07	PIXAR	95.71	95.26	62.35 - 101.70	2.38	40.21
INTC	22.12	-0.37	92,617,072	63,451,136	20.12	24.13	INTEL CP	22.49	22.15	19.64 - 29.01	1.16	19.39
ORCL	11.70	-0.39	62,108,332	41,453,636	9.69	13.70	ORACLE CORP	12.09	11.93	9.78 - 14.87	0.54	22.39
CSCO	17.20	-0.51	87,018,136	57,473,909	17.05	18.19	CISCO SYS INC	17.81	17.48	17.13 - 24.20	0.79	22.54
SNE	36.78	-1.46	1,960,000	1,044,681	N/A	N/A	SONY CP ADR	38.24	37.34	32.35 - 43.16	1.776	21.53
IVGN	70.21	-0.11	539,270	823,853	68.99	71.20	INVITROGEN CORP	70.32	69.65	46.19 - 77.00	1.63	43.14
ATVI	15.42	+0.28	2,914,544	3,486,954	0.01	9,000.00	ACTIVISION INC	15.14	15.30	9.12 - 18.7125	0.6877	22.01

Figure 1.2 – Example of Genuine Table

In [DHQ1995], a table is defined so that it is just a “genuine” table. Other definitions put emphasis on the utility of a table. For example, a table is considered as one of the visualizations people use to search and compare data [ZBC2003]. It provides an indexing scheme which allows the reader to associate quickly row and column headers with cells located in the body of the table [ZBC2003]. Interestingly, a table is sometimes called “a database designed for the human eye” [PMWC2003].

Automated table processing, or simply table processing, has been included as part of the document analysis. Many documents contain tables, which are distinctly different in layout from free texts. Traditionally, a table understanding task can be considered as a sequence of two sub-tasks: table detection and table recognition ([HKLW2002]). Table detection is about locating tables in a document; while table

recognition is about identifying various components of a table, such as headers and cells in the table body.

The main application of document analysis is information storage and retrieval, or otherwise generally known as knowledge management. For example, it is desirable to classify documents by their contents, and store similar documents in a cluster for search purposes. Recently, tables are also analyzed as documents by themselves. A higher level of table analysis, which normally involves interpretation of the table representation, is necessary for the applications involving summarization, table translation, table content delivery on mobile devices or speech interface ([Hurst2001], [HKLW2000], [WH2002]).

The input form in which a table is presented for programmatic processing is a vital factor not only in the techniques that are applied, but also in the kind of likely outcome. In the recent past, many researches focused on the extraction of low-level geometric information from scanned raster images of paper tables. Image processing techniques accounted for much of the success in detecting tables and recognizing various components of the detected tables. Since mid 90's, tables in electronic form have become prevalent, as part of the mega-trend of publishing documents on the Web. Since the physical representation of a web table is now available as a character string prior to its rendering, the emphasis of (automated) table processing of a web table should be on extracting more meaning from the HTML text. However, with a few exceptions ([YL2002], [YZ2001]), most of the researches on table processing has yet to take advantage of the visual cues embedded in the HTML tags in order to understand the table contents more thoroughly. Indeed, quite a few recent articles still stick to analysis of tables in free text.

Processing of unrestricted web tables is a difficult research problem for several seasons. First of all, a table embedded into a document is often not a self-contained

semantic unit, because the context, within which the table contents are interpreted, may lie outside the table grid. Secondly, given the diversity of how table can be defined, as indicated in the first paragraph of this section, one should not be surprised to find that tables in general do not share many similar characteristics because of the difference in contents, complexity of information embedded in the table, or choice of content layout by table authors. Thirdly, there are simply too many variations in which the same data may be presented in a web page. An example of the Common Data Set is cited in [TYM2004]. The Common Data Set (CDS) initiative is an effort to promote interoperability among the universities for publishing their own education-related data. However, “the tables that appear on the university web sites differ greatly in terms of HTML formatting, lexical variants of labels, extra or missing portions in those tables, etc.” [TYM2004]. Nonetheless, many researchers have been quite successful in extracting information from the table, by restricting to a certain kind of tables. One common approach is to process tables that are in the same domain of application, e.g., used car advertisements, with the a priori knowledge of the domain. Machine learning can also be a viable approach, if the tables share similar structures.

1.2 Statistical Table

In this thesis, we focus on a special kind of web tables: web statistical tables that are regularly published by all levels of governments and publicly funded agencies in many countries. For example, Statistical Canada publishes many statistical tables on the census data; these tables cover information in different area such as population and demography, economy, crime, and etc... These tables are different from tables that have been analyzed in the literature. They are meant to elucidate a massive amount of information in succinct way. The data are numeric in nature. They are mostly self-

contained, i.e., they are meant to be understood in isolation. Indeed, many tables are even downloadable as spreadsheets.

As mentioned above, data in statistical table are numeric in nature. Therefore, tables, that exhibit the structure of row and column headers as well as numerical data value in their corresponding data cell, are the ones that our research is interested in. There are other types of table published on the web which contain valuable data as well although they are not the focus in our research. In Chapter 5.2, we briefly discuss about them.

Statistical tables are important, if only one considers the huge amount of financial and human resources behind the efforts to procure them. Indeed, they are collectively one of prime information sources for decision makers in both public and private sectors. Consequently, most countries put out these tables on their governmental websites. Although the web is a much more user-friendly communication medium than print, it is still very difficult to search for specific information over a huge number of seemingly repetitive tables over the web. Moreover, due to the limited real estate on the screen, it is impossible to consider all aspects of the statistical data on one web page. A case in point is the census data published on the website of Statistics Denmark. A typical table has a title similar to this one: *Number of persons and course participants by area, educational area, highest education previously completed, age, national origin, sex and time* [SD]. The quantity “number of persons” can be qualified by values in 9 attributes, two of which have each more than 200 different values.

In practice, most agencies make the decision on the choice of the partial view(s) of the statistical dataset for display. While the selected tables usually present interesting and noteworthy trends, they do not always meet the specific needs of the reader, either in content or the style of visualization. Statistics Denmark does allow the reader to

submit his/her query on the web, but only tables with three dimensions are shown. Consequently, the entire statistical database must be segmented to fit into thousands of small tables! According to [ASSP2001], for complex analytical queries that typically require large amounts of data and processing, live access does not offer the level of interactivity, convenience, or processing efficiency that some users require. For these users, a highly preferable way to retrieve information from statistical tables is to build a system that is capable of developing a database schema for each statistical table, and populating the schema with data from the table, and then finally store the data in an off-the-shelf database system for ad hoc querying. The aim is to automate the entire process, transform web table as input into a database, together with its metadata, ready to be fed into the database system. Since statistical tables cover many different subject matters, the traditional ontology-based approach may not work. This thesis addresses the issues involved in building such a system.

1.3 Table Processing – Current Approaches

Up until now, one of the primary objectives of most work on IE from web tables (e.g., [CTT2000], [ELT2004], [LN1999], [SLN2002], [Pinto2002], [YL2002], [YTT2001]) is to accurately identify the attribute-value pairs contained in the table. An attribute may be column or row headers or combination of both; and the associated value, which is the content in the cell, that is associated column-wise and/or row-wise with the headers. There may be more than two keywords in an attribute. Take as an example as part of the statistical table described in [PMWC2003] (see Table 1.1), a typical attribute-value pair is [AreaPlanted-1997-Acres-Corp-Artichokes, 9300]. If one throws in the table caption as well – as is done in [Pinto2002] because it does contain relevant information, there may be even more keywords. While most work have reported good results in correctly identifying attribute-value pairs contained inside a table, it does pose problems

for the information retrieval system which must search these keywords to locate an answer to a given query. There are two problems in this regard. First, these attribute-pairs are not in a format that can be readily handled by a database system. The question answering system, QuASM [Pinto2002], for example, considers each attribute-value pair as a document for searching. The second problem is about the keywords contained in attribute, i.e., the metadata. It was reported in [PMWC2003] that the amount and quality of the metadata contained in the attribute were extremely important to the success of the system; but unfortunately, QuASM tended to “extract too much metadata”.

Corp	Area Planted			Area Harvested		
	1997	1998	1999	1997	1998	1999
	Acres					
Artichokes	9,300	9,700	9,800	9,300	9,700	9,800
Asparagus	79,530	77,730	79,590	74,030	74,430	75,890
Beans, Lima	2,700	3,000	3,200	2,500	2,000	2,900
Beans, Snap	90,260	94,700	98,700	82,660	87,800	90,600
Broccoli	130,800	134,300	137,400	130,800	134,300	137,300

Table 1.1 – A Portion of the Statistical Table Shown in [PMWC2003]

1.4 Table as Multidimensional Object

A new approach to solve these problems is introduced in this thesis. Instead of “too much metadata”, in our view, the problem is due to the lack of classification of the keywords in an attribute, which in turn makes it difficult to do any linguistic processing by the querying system. Our solution is the restoration of the table contents as a multidimensional object. Although some researchers have commented on the multidimensional nature of table, none has proposed a concrete way for the restoration. We conjecture that by reasoning of the layout of a well-designed statistical table, we can classify the keywords inside an attribute into semantically distinct groups, without resorting to any ontological knowledge. Below, we will attempt to show how we, as

human beings, discover the table as a multidimensional object. Later on, we will show how a program may mimic the reasoning of a human reader against the spatial layout and embedded visual cues in the statistical table.

We use Table 1.1 as the example. A human reader will have no trouble recognizing the column and row indexing structure, especially with the one-word row ('Acres') separating the column heading and the table body. The word 'Acres' in that row is seen as the unit for the value for the numeric data because the word 'Acres' spans over all numeric values in the table. The words, 'Corp' and 'Artichokes', should be put together as a group because 'Corp' is a label for all values in the first column. In addition, they are semantically distinct from column headers. The repetition of the strings '1997', '1998', and '1999' in the area of column heading, indicates a comparison is being made along the direction of these three words, and therefore they should form a group that is semantically distinct from the other two column headers, that is, 'Area Planted' and 'Area Harvested'. In total, three semantically distinct groups have been identified, which are mutually orthogonal, so that the numeric values inside the data cell may be viewed as a 3-dimensional cube, as shown in Figure 1.1.

Corp	Area		1997	1998	1999
	AreaPlanted	AreaHarvested			
Artichokes	9,300	9,300			
Asparagus	79,530	74,030			
Beans, Lima	2,700	2,500			
Beans, Snap	90,260	82,660			
Broccoli	130,800	130,800			

Figure 1.3 – The Multidimensional View of Table 1

In general, a dimension is often modelled as a hierarchy, with each node as an aggregate of entities immediately below them, or a node denotes a specialization of its parent. For example, 'Asparagus' is a specialization of 'Corp'. A dimension hierarchy is made up by the user, as part of the semantic analysis in data modelling. In the database terminology, the dimension hierarchies, together with the cube-like arrangement of data items is called OLAP (OnLine Analytic Processing) data model [C1993]. The connection between table generation and OLAP data model is first made in [WW1998]. It is pointed out also in [S1997] that statistical databases are very similar to the OLAP databases, because they share similar modelling methodology, and cater to applications that are analytic in nature.

The approach of extracting data from table data according to an OLAP, or OLAP like multidimensional data model, offers many significant advantages over the single dimensional model, i.e., the list of attribute-pairs. Clearly, with additional semantic information, the performance of a question answering system can only improve. Secondly, many querying systems are designed to query OLAP databases. MDX [M] is an example. More importantly, the OLAP data model greatly facilitates integration of similar tables. In a statistical database, data may be viewed from many dimensions, as is explained in the beginning of this paper. One way to reduce the number of dimensions in statistical tables is to 'slice' the cube along a dimension, that is, use a single value for the dimension and publish individual slices corresponding to each value for that dimension. For instance, in Figure 1.4, suppose for now the agricultural database used to produce Table 1.1, contains an additional dimension, say State. A 'slice' along the dimension "State" is a 3D table consisting of data pertaining to a particular state. Thus the original 4D OLAP database may be recovered by integrating

the 3D tables for all states. In this way, queries may be posed about the 4D database that could not be answered by considering only the published web tables.

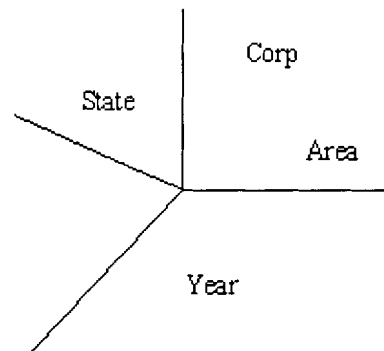


Figure 1.4 – Dimensions in an OLAP Database

1.5 A New Approach to Statistical Table Processing

Traditionally, table processing terminates when the table components are recognized. In this thesis, we begin with table component recognition, to be followed by two extra steps: the derivation of multidimensional model and linguistic processing. The derivation of multidimensional model means that our approach, based on human heuristics, transforms the original table into a multidimensional data cube, similar to OLAP data cube. And then linguistics processing is applied to the semantically related component in the multidimensional model. These will be discussed in more details in Chapter 3.

Another major feature of our approach is that it does not rely on any domain specific knowledge. In fact, the first two phases of our table processing, i.e., recognition of table components and derivation of the multidimensional table, depend entirely on heuristics regarding the spatial layout of the table. The basic rationale for our heuristics is that the table has been designed to make the reader comprehending with ease the deeply nested metadata. As a consequence, visual cues, such as indentation, font sizes, font style, and background and foreground colours are extensively employed to

delineate different types of headers, data cell, summary rows and columns and so on. Thanks to the HTML tags inside the text, our table processing system will pick up these visual cues, which allow the system to capture the essence of the visual image of the table as if the table is viewed by a human reader. These heuristics are applied in order to 'recover' the multidimensional structure of the original statistical database, or at least the relevant part of the database. In this regard, it is analogous to the reasoning behind David Waltz's approach to the understanding line drawings of 3 dimensional blocks [W1975]. Waltz's algorithm essentially traces the line segments that are thought to belong one single block, in a way that a normal human reader may do in the same situation. Of course, the line drawings are simple in comparison to web statistical tables. Perhaps due to this reason, the web statistical tables are adorned with visual cues for human visualization. Indeed, if the 3 dimensional blocks were uniquely coloured, it would be almost trivial for a program to recognize them.

1.6 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 is about modelling of statistical tables and the intermediate forms in which the table may be represented. We identify four forms in which a table may be represented: the input form (in HTML text), the component form (according to a component model), the multidimensional data form (according to a multidimensional data model), and database form (according to some off-the-shelf database system). The component and multidimensional data models are described in detail. A description of how a table may be queried in its final stage is also included. In chapter 3, we show how a table may be transformed into a component form and then into a multidimensional form. Then, a system called Web Table Miner is described in chapter 4 and experiment data are presented. In Chapter 5, we describe related work. And finally, chapter 6 concludes this thesis.

CHAPTER 2 REPRESENTATION FORMS OF A TABLE

In this chapter, we present an overview of the forms into which the table may be transformed, from the raw data to the ultimate query-able form:

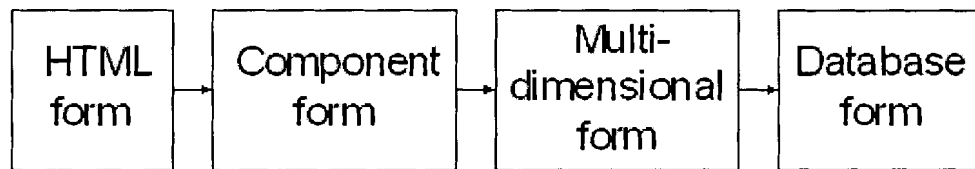


Figure 2.1 – Transformation of Table

In what follows, we first introduce our component model, and then the multidimensional data model, which define respectively the component and multidimensional forms. Finally, we briefly introduce the subject of querying a multidimensional database on an off-the-shelf database system.

2.1 Component Model

Wang in his Ph.D. thesis [W1996] defines a physical model of a table. The example and terms shown in Figure 2.2 are taken from Wang [WW1998], which in turn are based on the terminology from the Chicago Manual of Style [G1993]. The other entities that are normally associated with a table, such as title, footnotes, or source of the table are not shown.

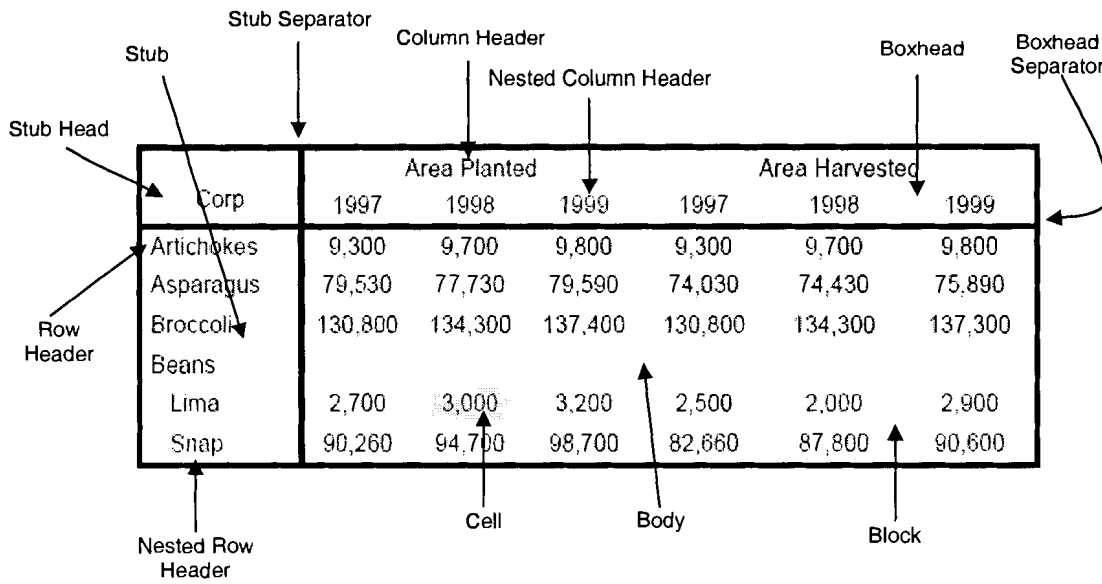


Figure 2.2 – Wang’s Physical Model of a Table

While our component model is, in many respects, similar to the “standard” model, we choose to define our own, together with our own terminology. Wang’s model is specified for the purpose of table generation, while our primary concern here is to understand a table that has already been generated.

A HTML table has all of its contents confined to the table grid, except the caption, which is the text associated with the *<caption>* tag. Table header and footer are respectively the top and bottom rows of the table grid. They are usually text strings associated with *<thead>* and *<tfoot>* tags, and take up the whole row as one single cell. As a pre-processing step, all single-cell rows from the top and the bottom of the table grid are stripped off. While the caption, table header and table footer may be candidates for interesting linguistic processing (see Chapter 3.5), our focus here is on the table core, which is made of three types of region: *Upper Left Corner (ULC)*, *Column Heading*, and *Table Body*, as shown in Figure 2.3. The ULC and Column Heading share the same rows, and are horizontally separated from the *Table Body* by a line called *Column*

Heading Divider. The Table Body consists of one or more *Table Body Sections (TBS's)*. A TBS, together with the ULC and Column Heading could be viewed a table by itself. To us, this is how we segment a table into smaller units. It represents a major departure from the Wang's model which includes blocks of data cells, and the component model defined in [LN2000], which defines nested tables.

A TBS may be subdivided into three regions: *TBS header*, *Row Heading* and *Data Cells*. Row Headings of all TBS's, as well as ULC, are aligned vertically on a vertical line, which is called *Row Heading Divider*.

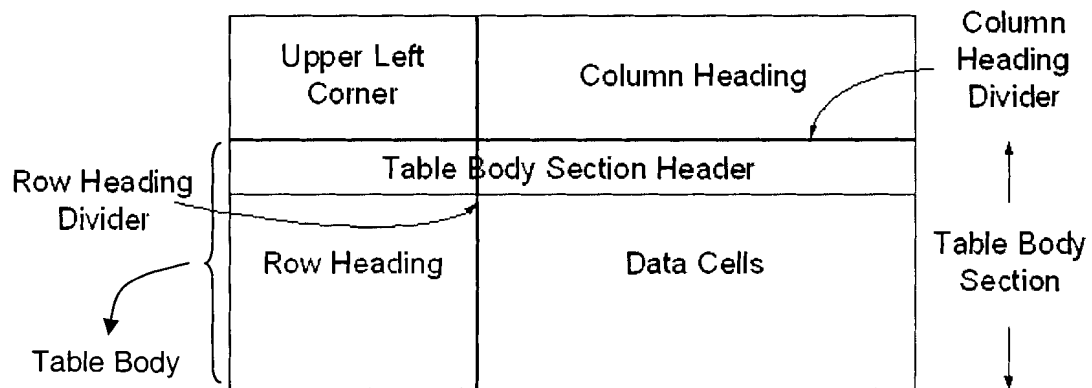
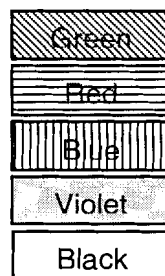


Figure 2.3 – Our Component Model for Table Layout

As an example, consider the Table 1.1 in chapter 1, which is reproduced in Table 2.1. The colouring scheme of the texts inside the cells and the shading pattern of the cells show the regions that the cells belong to:

- ULC
- Column Heading
- Row Heading
- Table Body Section Header
- Data Cells



Corp	Area Planted			Area Harvested		
	1997	1998	1999	1997	1998	1999
	Acres					
Artichokes	9,300	9,700	9,800	9,300	9,700	9,800
Asparagus	79,530	77,730	79,590	74,030	74,430	75,890
Beans, Lima	2,700	3,000	3,200	2,500	2,000	2,900
Beans, Snap	90,260	94,700	98,700	82,660	87,800	90,600
Broccoli	130,800	134,300	137,400	130,800	134,300	137,300

Table 2.1 – Table 1.1 in Component Form

2.2 Multidimensional Data Model

Multidimensional Data Model is adapted from the Star Schema model for OLAP databases [S1997]. It consists of three main components: dimension hierarchy, dimension relation and multidimensional dataset. Most tables embody only dimension hierarchies and multidimensional dataset. (Note that a relation in this thesis is meant to be a table in the sense of a relational DBMS.)

A *dimension hierarchy* is a generalization hierarchy, in the object-oriented data modelling parlance. A node in the hierarchy is called a member of the hierarchy. All members in the hierarchy are related to each other through a set of parent-child relationships. There are two types of dimension hierarchies: concept hierarchy and measure hierarchy, the latter of which has a count of one. We first consider the concept dimensional hierarchy (Figure 2.4 (i) and (ii) are examples of concept hierarchies). Two types of member are found: *concept* and *aggregate* members. A concept member denotes a concept in the domain covered by the table. This concept is a specialization of the concept donated by its parent. For example, broccoli is a specialized form of corp. An aggregate member denotes a statistical view of all concept members under the same parent. For example, an aggregate member can be labelled as 'total' or 'median', associated with a well-known aggregation function. Or it can be a specially calculated

member, whose value may be computed by a formula. Usually, the formula is given as part of the header, such as “% change from 2000 to 2001”.

The *measure hierarchy* is distinguished from dimension hierarchies because the former is not semantically related to the domains covered by the table. It is about the quantity associated with the numeric value stored in the data cells of the table. Each member in the measure hierarchy has two parts: the label which denotes the semantic meaning of the quantity, and the unit in which the quantity is measured. The Area hierarchy in Figure 2.4 (iii) is an example of a measure hierarchy. The measure hierarchy does not have any aggregate members.

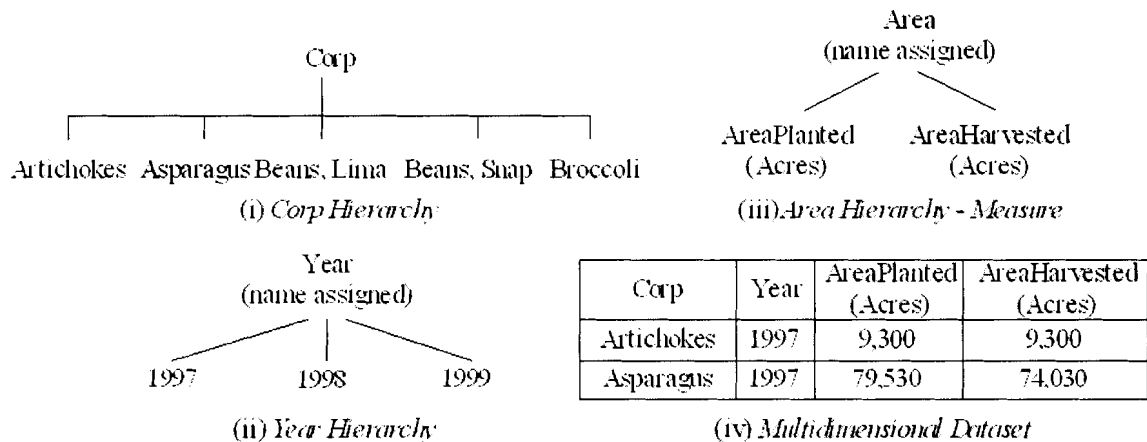


Figure 2.4 – Table 1 in Multidimensional Form

A dimensional hierarchy contains only the names (or IDs) of the members. If extra information about the members is available, it is stored in a dimension relation. A dimensional relation may exist as an extension of a dimensional hierarchy. Each tuple in the relation contains information pertaining to a certain member of the corresponding dimensional hierarchy. For example, tuples in the dimension relation associated with Crop may contain descriptions of various crop items. The table from Statistic Canada is a real-life example (Table 2.1), where column header ‘Type’ for the second column

apparently contains values that are associated with the values in the first column 'Name'.

Thus the dimension relation has two attributes: Name and Type.

Name	Type ¹	Population			Total private dwellings, 2001
		2001	1996	% change	
Canada †		30,907,094	28,846,761	4.0	12,548,598
British Columbia †		3,907,738	3,724,500	4.9	1,643,969
Tzeetzi Lake 131	R	688	575	-198.8	017
Ucluelet	DM	1,559	1,658	-6.0	705
Ulkatcho 13	R	0	29	-100.0	1
Uncha Lake 13A	R	0	0	07.5	4
Union Bay 4	R	0	0	0	0
Unnamed 10	R	17	15	13.3	13
Upper Hat Creek 1	R	28	31	-9.7	10
Upper Nepa 6	R	5	11	-54.5	4
Upper Sumas 6	R	175	136	28.7	62
Valemount	VL	1,195	1,303	-8.3	523
Vancouver	C	545,671	514,008	6.2	248,081
Vanderhoof	DM	4,390	4,401	-0.2	1,663
Vernon	C	33,494	32,165	4.1	15,288
Victoria	C	74,125	73,504	0.8	42,359
View Royal	T	7,271	6,441	12.9	3,166
Village Island 1	R	10	11	-9.1	4
Waiwakum 14	R	91	100	-9.0	26
Warfield	VL	1,739	1,788	-2.7	798
Wells	DM	235	249	-5.6	175
West Moberly Lake 168A	R	52	69	-24.6	16
West Vancouver	DM	41,421	40,882	1.3	17,299
Whispering Pines 4	R	60	43	39.5	22
Whistler	DM	8,896	7,172	24.0	8,410
White Rock	C	18,250	17,210	6.0	9,397

Source: Statistics Canada's Internet Site, <http://www12.statcan.ca/english/census01/products/standard/popdwelt/>, December 2004

Table 2.2 - A Table that Contains Dimensional Relation

Dimensional relations are in general not very visible in statistical tables. Space limitation may be the primary reason. Important information about the member is usually shown as part of the text string labelling the member. Still, in tables associated with commercial applications, a member is sometimes associated with a hyperlink that points to another web page. For instance, a table that advertises used cars may contain the ID

of the used car, the model, age, mileage and price. Clicking of the ID will lead to another web page giving more details about the used car [ETL2004].

The multidimensional dataset is an n -ary relation. Each tuple of the relation is in fact equivalent to an attribute-value pair, as we define it in the introductory chapter. All tuples however have a fixed length, n , which is calculated as follows. A tuple has an attribute from each concept dimension hierarchy and as many attributes as the leaf-level members of the measure hierarchy. See Figure 2.5 (iv) for an example of a partial multidimensional dataset. Only two out of a total of 30 tuples are shown.

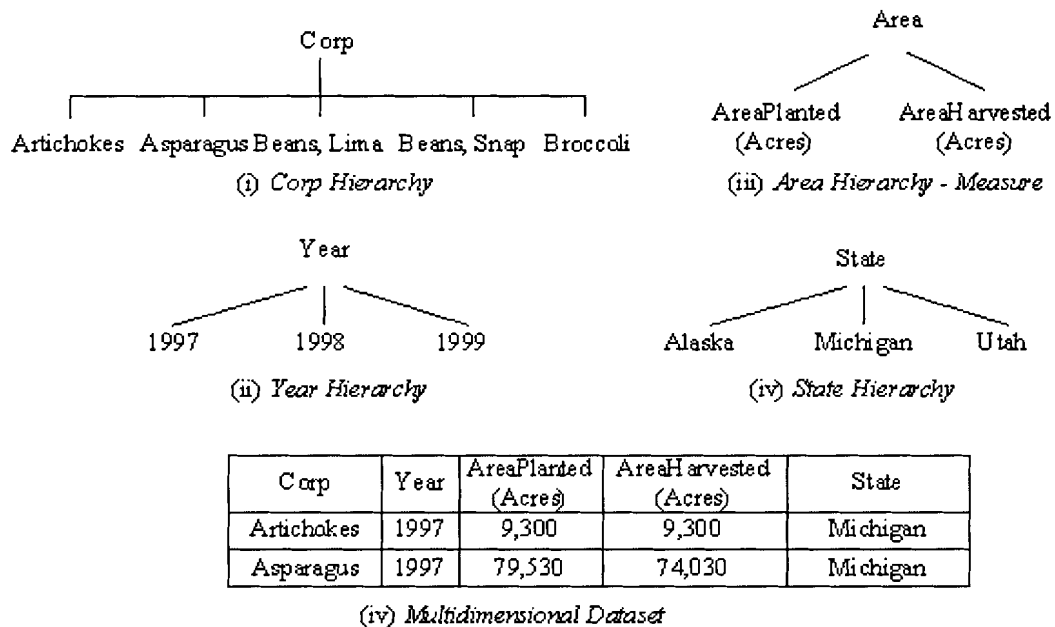


Figure 2.5 - Multidimensional Form in 4 Dimensions

2.3 Multidimensional Expression (MDX) & OLAP System

One major attraction of the multidimensional data model is that it fits nicely with off-the-self database systems. The output, i.e., the dimension hierarchies and multidimensional dataset, can be input directly into an OLAP database. From there, tables in different formats may be generated using a standard query language, i.e.,

Multidimensional Expression, or MDX [M]. The language is very powerful, but its complexity prohibits us from giving even a simple introduction. However, we will show how the table in Table 1.1 may be generated its multidimensional form using the general concepts of MDX, as determined by the multidimensional data model. We use the data model shown in Figure 2.5 as the example, which is essentially Figure 2.4, augmented by an additional dimensional hierarchy, State.

The first task in table generation is to determine which dimensions are chosen to be presented in the table, and additionally, which chosen dimensions are on the column axis and row axis respectively. For the slicer dimensions, i.e., those dimensions that are not chosen, values of the attributes for those dimensions in the multidimensional dataset are specified. In our example, we have the following categorization of dimension hierarchies:

- On the column axis: Year, Area
- On the row axis: Corp
- Slicer dimension: State, where State = 'Michigan'

The next task is to configure the dimensions on both axes. For the column axis, we use the *cross-join* operation, which is the Cartesian product of two sets. The first set consists of members from the Area Hierarchy, i.e., [AreaPlanted] and [AreaHarvested]; the second set, [1997], [1998], and [1999]. For the row axis, we use the following specification: ([Corp], [Corp].descendants). There is an option of the cross-join, such as the empty columns/rows as a result of cross-join are suppressed.

To populate the table cells, the first step is to select all tuples with value 'Michigan' under the attribute [State]. The values under the attributes [AreaPlanted] and [AreaHarvested] in each tuple are chosen and entered into the corresponding cells in the

table, as indicated by the values under the other two attributes, i.e., [Corp] and [Year]. The outcome is Table 1.1, minus the 'Acres' row.

Once the multidimensional database is derived, it is ready to be fed into an OLAP system for querying purposes. Most OLAP systems accept a star schema and a fact table as the input. The schema in our case consists of the dimension hierarchies and the relational schema for the multidimensional dataset, while the fact table is just the multidimensional dataset.

CHAPTER 3 TABLE PROCESSING

In devising a strategy for restoration of multidimensional object from a table, we rely heavily on the presumption that the table designer is experienced in presenting statistical data to viewers on the Web. For example, various visual cues are deployed to highlight the neighbouring rows (or columns) that donate different things. The concepts in parent-child relationship are presented in the table either in the top-down order (vertically) or the left-right order (horizontally). Structurally, we presume that the table is generated by software that follows the similar principles of MDX. For example, a dimension hierarchy is not split into parts located in row and column heading regions respectively. Operations similar to cross-join are used to pair members from different dimensions. If the table designer does not deploy any clear layout structure or visual cues in the table, our presumption would be violated and our strategy would not be successful. Based on our observations on multiple statistical websites, such as Statistics Canada and Statistics Denmark, our presumption is believed to be valid.

We begin processing a table by recognizing various components of the physical model for the given table, which is described in Chapter 3.1. This task is carried out by recognizing the column and row heading dividers. From the column and row headings, we deduce, in Chapter 3.2, the dimensional hierarchies that are embodied in the table. In Chapter 3.3, we show how Table Body Sections (TBS) are recognized. At this point, the hierarchies have already been derived, and they are subject to linguistic processing, which relies on domain-independent information for more in-depth analysis of non-concept hierarchies. Only temporal and measure dimensional hierarchies are processed, using generic knowledge bases about time, and about various types of

measurements. In Chapter 3.5, we show an instance of table integration. Finally, in Chapter 3.6, we show how to build a relational schema for the multidimensional dataset and to extract data from the table cells to populate the dataset.

3.1 Recognizing Table Components

To a human reader, the first attempt to recognize the contents of table is often to identify the two dividers or equivalently the ULC, so that data cells could be associated with its corresponding column and row headers. This too is the first task for our table recognizer.

Detection of the Column (or Row) Heading Divider of a table in print medium is often quite straightforward, because the typesetter would use a thicker line to denote this divider. While HTML does have this provision, it is rarely used by table authors. Instead, the following rules seem to work with most statistical tables. If there is more than one Table Body Section (TBS) in the table body, the rule is to visually separate one TBS from the neighbouring ones, e.g., a blank line. It follows logically that there should be also a visually separation between the first TBS and the Column Heading. If there is only one TBS, the table is considerably simpler and the need to visually separate the TBS from the Column Heading is less pressing. Occasionally, colouring and/or shading may be applied to the same effect, but it cannot be counted on as a reliable rule. A more reliable one is about the predominantly numeric data within the table body of the TBS, in contrast to predominantly alphanumeric texts in the Column Heading.

Let us state in more formal manner how to recognize the Column Heading Divider. The processing begins with the first two rows of the table, after the title of the table, if any, has been stripped off. If we can't find at least one of the following conditions is satisfied, the first row is abandoned, and the next pair of rows are

considered, and so on. For every pair of rows being considered, the Column Heading Divider is said to be found between the two rows if one or more of the following conditions:

- The two rows are visually different, due to different shading pattern or background colour.
- The second row is either a blank row, or has significantly fewer cells than the first one.
- The two rows have the same number of cells, which are identified aligned, with the second row having significantly more cells with numeric data

Recognizing the Row Heading Divider can be done either directly, i.e., by recognizing the Row Heading; or indirectly, i.e., by recognizing the Upper Left Corner (ULC). We begin with latter, because we have already recognized rows where the cells of ULC may reside.

The easiest way to recognize ULC is to look for empty cells, starting from the very corner to the right side of the table. If there is at least one empty cell, the ULC will then be the collection of all these empty cells. If the first cell in that corner is not empty, then we need to investigate the text therein. The interesting point about this text is that it may be associated with either the column headers, or the row headers, or none at all. Generally the text is perceived by a human reader as a row header, but the table author may alter the reader's perception, by dressing up the appearance of the text in the ULC. In other words, with appropriate visual cues, the text could be made to appear to be part of the row heading, or unrelated to both the row and column heading. For example, the text in the ULC can be made visually different from texts in all other cells in the same row, or the text in the ULC is followed by a punctuation at the end of the text, such as ':' to indicate the texts in the remaining row are continuation of the text in the ULC. Another exception is when the text is aligned in a different way as a signal that it is not to

be considered a row header. The table shown in Table 3.1 is an example. The ULC there spans three rows in the leftmost column and the text 'Ontario' appears in the top cell. This is a clear signal to the reader that this text is not intended to be part of row or column heading, and therefore should be included into table title. Incidentally, this table is one of a series of tables for all provinces and territories in Canada, and the text is just a value for the 'slicer' dimension.

Ontario	2001			
	Youths charged		Adults charged	
	rate per 100,000 population	% change from 2000 to 2001	rate per 100,000 population	% change from 2000 to 2001
All incidents	4,855.9	0.3	2,087.3	2.1
Criminal Code offences (excluding traffic offences)	4,332.6	0.4	1,536.0	3.5
Crimes of violence	1,063.1	-0.7	480.5	3.8
Homicide	1.2	0.2	1.1	10.5
Attempted murder	3.9	29.8	2.2	0.9
Assaults (level 1 to 31)	757.8	-3.0	385.6	4.1
Sexual assault	71.5	-12.8	31.2	-7.0
Other sexual offences	4.2	-12.8	2.1	-19.5
Robbery	174.7	15.5	23.5	12.5
Other crimes of violence ²	49.7	6.9	34.6	7.3
Property crimes	1,801.1	-1.2	440.3	1.2
Breaking and entering	380.8	-7.3	62.6	-4.8
Motor vehicle theft	205.6	8.3	24.8	0.4
Theft over \$5,000	10.9	-8.9	8.0	11.4
Theft \$5,000 and under	823.2	-2.6	183.6	5.5
Possession of stolen goods	289.1	3.6	70.8	1.7
Frauds	91.4	5.1	90.7	-3.2
Other Criminal Code offences	1,468.4	3.3	615.2	5.0
Criminal Code offences (traffic offences)	0.0	**	314.7	2.0
Impaired driving	0.0	**	250.5	1.5
Other c.c traffic	0.0	***	64.2	4.1

offences3				
Federal statutes	523.3	-0.4	236.6	-5.8
Drugs	382.1	2.3	206.5	-2.4
Other federal statutes	141.2	6.3	30.2	-23.8

Table 3.1 - A Table where Text is Part of the 'Slicer' Dimension

In summary, we assert that if the text in the (single) cell has the same appearance as other column headers in the same row, in terms of the font style, font size and alignment within the cell, then it is most likely a column header. Otherwise, other rules may be applied to determine the nature of the text, such as the ones just cited.

Scenario 1: The Column Heading has only one single row

- A The leftmost cell of the row is empty → this is an empty ULC
- B The leftmost cell of the row is non-empty: the text contained therein is row header unless one of the following situations happens, in which case the text is a column header:
 - i The cell and/or the text is visually different from all other cells in the row
 - ii There is punctuation at the end of the text, such as ':'

Scenario 2: Column Heading has two or more rows

- A The leftmost cell of the first row spans the entire Column Heading region: it is visually a single cell.
 - i The cell is empty → this is an empty ULC
 - ii The cell is not empty: the text contained therein is then the column label for the column of row headers in the Table Body.
- B The leftmost column of the Column Heading has more than one cell:
 - i The top cell is empty: all of the cells in the column of the Column Heading are considered to be one cell and processed as in Scenario 2.A
 - ii The top cell is not empty: the text is not likely connected to the Row or Column Heading regions, because of lack of spanning row-wise or column-wise. It is considered as part of the table header.

By now, with decisions already made on the locations of ULC and Column Heading, the Row Heading Divider would have been determined as well. However, this

decision should be re-affirmed by the following rule: scanning the columns from left to right, we determine the Row Heading Divider if most of the texts contained in the cells in current column change to numeric values in the next column.

3.2 Recognizing Dimension Hierarchies

Column Heading and Row Heading are the main sources where information about the number of dimensions, the members inside a dimension hierarchy, and the child-parent relationships between the members in the same hierarchy are discovered. The first rule we make is that no hierarchy may span two heading regions, because it would be confusing to the human reader.

We observe that the Column and Row Headings are asymmetric in design because of the layout of a web page. In short, the space of a web page is limited in width, but is quite expandable in length. The table author usually includes members of one to two small hierarchies that are short and has few members, into the Column Heading. A favourite dimension is the time dimension, with the purpose of showing trends along the timeline. Another dimension that is commonly located in the Column Heading is the measure hierarchy, because it makes less sense to label the data row by row. Consequently, we reason that the measurement unit for numeric values in the Table Body is found either in the Column Heading, or in the header of a Table Body Section.

The recognition of dimension hierarchies in the two heading regions does have one thing in common: it is about how to extract headers when there is more than one row or column in the Column or Row Heading respectively. When two headers are present in the same column, or same row, we use essentially the same procedure to

determine the relationship, if any, between the two. We first consider the Column Heading.

3.2.1 Processing Column Heading

When two column headers are found in the column, the following situation could happen, as shown in Figure 3.1:

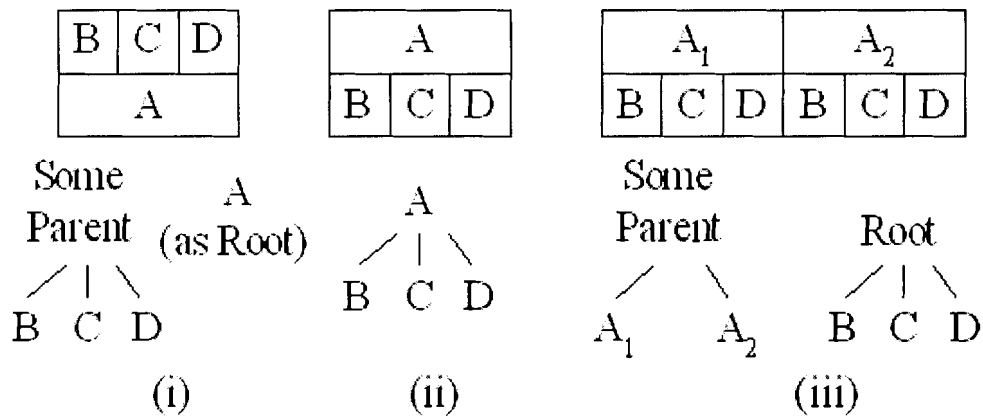


Figure 3.1 - Relative Cell Positioning in Two Neighbouring Rows

- (i) The column headers are drawn from two different dimensions. Usually, it does not make much sense to associate several members of one dimension with one single member from other hierarchy, except in the situation where the single member from the measure hierarchy.
- (ii) The column headers are drawn from one dimension hierarchy, but two neighbouring levels.
- (iii) Two sets of column headers are drawn from two different dimensions to form a Cartesian product of the two, as outcome of the cross-join operation of these two sets. This situation is commonly seen when data are presented for comparison purposes. Since empty columns or rows may be suppressed, for example, the column header (A₂, C) may be

eliminated because it is not applicable, we should not always look for exact replication of one set of column headers.

These rules apply to the processing of row headers in the Row Heading, with the exception that the situation (i) no longer applies because members of the measure hierarchy seldom appear in the Row Heading region. The following procedure shows how the rules are actually applied.

- For every two consecutive rows starting the top of the Column Heading:
 - Is there a column-spanned cell in the top row?
 - **No.** Is there a column spanned cell in the bottom row?
 - **Yes.** For each spanned cell, the text is a measurement unit for all headers in the columns of the top row that are spanned by the cell, which are considered to be members of a hierarchy at the same level. For example, in Figure 3.1 (i), A is the bottom row is to be a measurement unit, while headers B, C, and D in the top row are members of a concept hierarchy. This conjecture can be confirmed by linguistic processing (see Chapter 4.3).
 - **No.** All headers in the top row are members of a hierarchy. Headers in the same column but at different level are considered to be in parent-child relationship in the dimension hierarchy.
 - **Yes.** Repeating sub-sequences of headers?
 - **No.** All the headers are members of the same hierarchy, and at the same level; for example, B, C, and D in Figure 3.1 (ii).
 - **Yes.** The headers in the sub-sequence are new members of a different hierarchy; for instance, B, C, and D form repeating sub-sequence in Figure 3.1 (iii).

3.2.2 Processing Row Heading Region in a Table Body Section

The Row Heading Region consists of the non-numeric portion of all Table Body Sections (TBS) inside the Table Body. Here, we are concerned only with the Row

Heading Region confined to a single TBS. In Chapter 3.3, we will consider how the results derived for each TBS may be integrated.

Let us first suppose that there are two or more columns inside Row Heading region. Like the processing of rows in Column Heading, we always consider two neighbouring columns (if any) together, beginning from the first two leftmost columns, and moving one column to the left at a time. For a specific pair of columns, there are two possible situations. One situation is that the second column is there to provide more information about the row headers in the first column, as in the case of Table 2.2 cited above. As part of table processing, the dimension relation will be extracted from the table. For example, 'Vancouver' area belongs to Type 'C', where the Type is an attribute in the relation tuple of 'Vancouver'. The other situation is that the second column consists of row headers, as in the table in Table 3.2 and 3.3 below. Consequently, we may apply the same reasoning to these two columns as what we do to two neighbouring rows in the Row Heading region.

Age Group/Sex		Marital status				Total
		Never married	Now married	Widowed	Divorced/separated	
15 - 19	Male	21.8	58.1	-	100.0	21.9
	Female	18.6	38.4	-	23.3	18.7
	Sub-total	20.2	44.9	-	37.8	20.4
20 - 24	Male	74.3	93.3	66.7	82.0	75.2
	Female	75.8	66.3	69.1	74.7	74.8
	Sub-total	75.1	73.8	68.8	76.9	75.0
25 - 29	Male	92.0	95.0	40.0	86.9	92.7
	Female	94.1	72.9	66.8	73.5	85.6
	Sub-total	93.0	80.4	63.2	78.2	88.8
30 - 34	Male	91.5	96.2	87.0	89.8	94.0

Female	92.8	67.0	60.2	73.9	75.2
Sub-total	92.1	78.5	63.4	79.1	83.5

Source: Census and Statistics Department of HKSAR, <http://www.info.gov.hk/censtatd/eng>, July 2003

Table 3.2 - A Table whose second column consists of Row Headers

Family situation	Number of parents with earnings	All children	Children in low income ¹
		%	
Children living in couple families ²	None	2.6	12.3
	One	16.2	20.8
	Both	55.9	16.6
	Total	74.8	49.7
Children living in lone-parent families ³	None	4.0	20.2
	One	10.4	19.2
	Total	14.4	39.4
Total	...	100.0	99.9
1. Children living below the low-income cut-offs (see the explanation in the methodology).			
2. All children under 18, except those living in the Yukon, Northwest Territories, Nunavut, on Indian reserves and in institutions.			
3. Children living in single-family households with no additional persons, e.g., grandparents, uncles and aunts, etc.			
... not applicable			

Source: Statistics Canada, <http://www.statcan.ca>, July 2003

Table 3.3 - Another Table whose second column consists of Row Headers

Suppose now only a single column is present, which is more likely than not. For this case, it becomes necessary for the table author to signal as clearly as possible the type of relationship between the headers in two consecutive rows. We rely on visual cues such as indentation, column spanning, font type, font style, font size, font colour, and background colour to ascertain the relationship between different rows. If they are visually identical, then the headers with same visual cue represent members on the same level of the same hierarchy. Otherwise, they are either members at different levels of the same hierarchy, or members from different hierarchies. In the former case, the headers are copied to the same “conceptual” column; otherwise, they are copied to

different “conceptual” columns (see Figure 3.2). After processing all rows in this manner, we end up with a number of conceptual columns. Then we can use the same procedure describe in Chapter 3.2.1 to derive the dimension hierarchies embedded in the Row Heading region.

The rule of thumb is that the row headers sharing the same visual cue are considered to be at the same hierarchical level. Based on observation, the visualization cues used effectively are background colour, indentation, font style like bolded font, and font size. The following procedure shows how we determine the hierarchical relationship of the row headers based on these visual cues. Each call to this procedure finds all the rows in the same level. By recursively calling this procedure, hierarchical relationship can be determined. Figure 3.2 also illustrates this procedure.

- Iterate over the rows from first (start) row to last (end) row
 - If it is the start row?
 - **Yes.** And remember the visualization cues of the start row.
 - **No.** Check whether the background colour, indentation, font style and size are the same as start row
 - **Yes.** Mark it as same level as the start row and examine next row.
 - **No.** This is an unmarked row which is related to the previous marked row. Examine next row.
- For each marked row which relates to unmarked row(s), recursively apply this procedure to the unmarked rows.

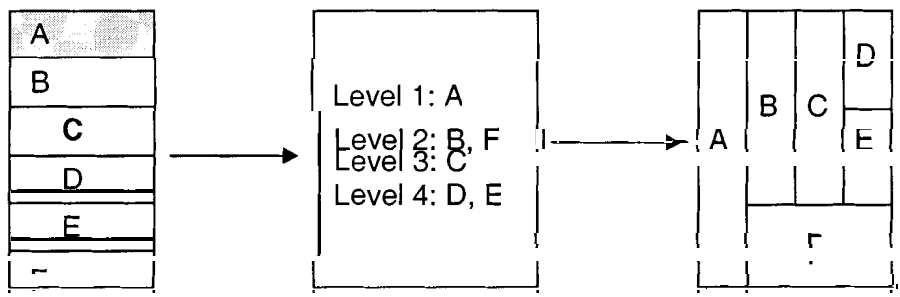


Figure 3.2 - Determining Hierarchical Relationship and Building Conceptual Columns

3.3 Recognizing Table Body Sections

A Table Body section (TBS), coupled with the Column Heading region, could be a table by itself. Multiple TBS's are integrated into the table because they are too small to be meaningful if they are made individual tables and they are tightly related to each other by virtue of sharing the same Column Heading. In many instances where there are multiple TBS's, one is often seen as the summary, and the rest, the details, so that the reader may drill down from the summary to details, as shown in Table 3.4.

A TBS header mainly serves as a divider between neighbouring sections. At times, other information may be included if it is applied only to the particular TBS. A common example is the measurement unit, as shown in Table 3.4.

	2001			
	Canada ¹	Atlantic Region	Quebec	Ontario
Household characteristics				
Estimated number of households	11,767,180	897,190	3,041,380	4,379,490
Average number of persons per household	2.57	2.58	2.39	2.68
		\$		
Weekly expenditure detail²				
Total weekly food expenditure	123.76	108.76	118.47	125.90

Food purchased from restaurants	37.52	27.40	33.76	38.93
Food purchased from stores	86.24	81.36	84.71	86.97
While on trips overnight or longer	2.56	2.25	2.24	1.93
Locally and on day trips	83.68	79.11	82.47	85.04
Meat	17.34	17.15	17.32	18.26
Fish and other marine products	2.81	2.52	2.70	3.11
Dairy products and eggs	12.68	11.86	13.28	12.55
Bakery and other cereal products	12.51	12.36	12.44	12.42
Fruits and nuts	9.82	7.81	9.11	10.46
Vegetables	8.67	7.28	8.81	8.88
Condiments, spices and vinegar	2.56	2.36	2.53	2.40
Sugar and sugar preparations	3.02	3.06	2.74	2.88
Coffee and tea	1.41	1.08	1.40	1.36
Fats and oils	1.03	1.35	0.97	0.92
Other foods, materials and food preparations	8.45	8.89	7.80	8.12
Non-alcoholic beverages	3.39	3.40	3.37	3.67
1. Excludes the Yukon, the Northwest Territories and Nunavut.				
2. Average weekly expenditure data refer to all households.				
Source: Statistics Canada, Income Statistics Division, Catalogue no. 62-554-XIE.				
Last modified: February 21, 2003.				

Source: Statistics Canada, <http://www.statcan.ca/english/Pgdb/famil27a.htm>, July 2003

Table 3.4 - An Example of Table with Multiple Table Body Sections

The usual visual cues, such as font style and size, text alignment, and background colour, are the means by which the reader are able to differentiate one TBS from another. A blank line is often inserted as a readability enhancement. Consideration should be given to the possibility that one TBS may be visually distinguished from the rest because it is the summary section. In the case where all TBS's look alike, the following rules may be applied to recognize the summary section:

The first section is the summary section, that is, if a summary section exists.

The summary section has fewer rows.

The text in the summary section contains some special words, such as 'total'

Finally, row headings of all TBS's must be integrated. By now, the row heading of each TBS is in the form of one or more dimension hierarchies, including the measure hierarchy. The measure hierarchies of all TBS's will be merged together, but the

concept hierarchies must be carefully considered. Three different scenarios are considered:

Scenario 1: One concept hierarchy per TBS: non-identical

Some hierarchies are different from others: all hierarchies are merged together, such that the TBS headers (minus the measure, if any) become the members at the second level of the merged hierarchy. For example, the merged hierarchy of row headings of the two TBS's of Table 3.4 is shown in Figure 3.3:

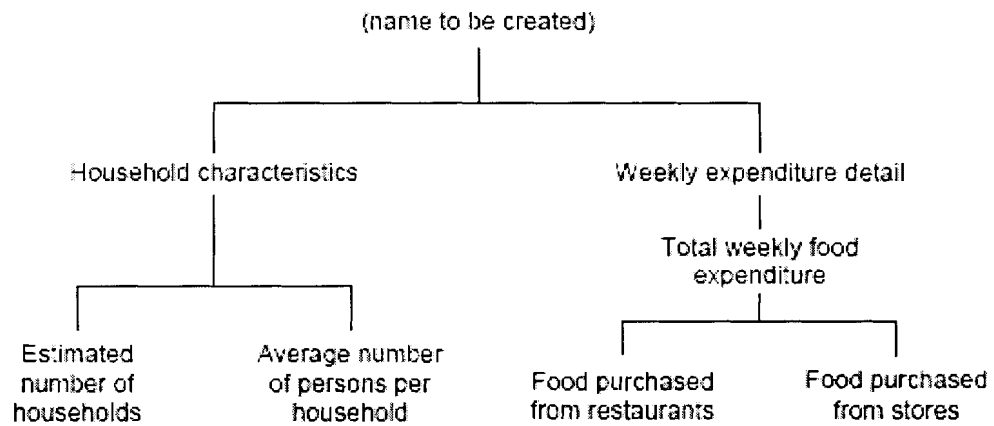


Figure 3.3 - Merged Hierarchy from Row Headings

Scenario 2: One concept hierarchy per TBS: identical

This concept hierarchy will then be a hierarchy for the entire table, together with another hierarchy that is made of all TBS headers at the second level. For example, the Row Heading region under the label 'age, sex, race/identity' would have the following hierarchies for Table 3.5, as illustrated in Figure 3.4:

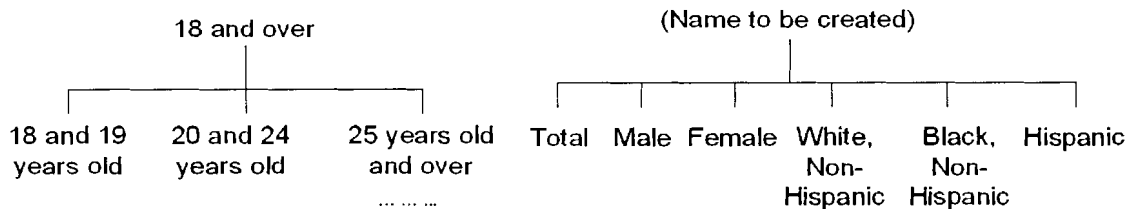


Figure 3.4 - Two Hierarchies from Row Headings

Table 9.—Highest level of education attained by persons age 18 and over, by age, sex, and race/ethnicity: March 2001
(In thousands)

Age, sex, and race/ethnicity	Total	Elementary level		High school			College					
		Less than 7 years	7 or 8 years	1 to 3 years	4 years	Completer	Some college	Associate	Bachelor's	Master's	Professional	Doctorate
1	2	3	4	5	6	7	8	9	10	11	12	13
Total												
18 and over	203,998	8,647	6,054	18,267	2,901	85,439	40,070	15,703	33,067	10,749	2,611	2,060
18 and 19 years old	8,131	74	72	2,376	465	2,322	2,225	63	1	1	1	1
20 to 24 years old	18,836	341	227	1,806	395	5,768	6,911	1,204	2,160	105	26	7
25 years old and over	177,022	6,231	5,759	12,625	2,141	67,749	30,334	14,445	30,844	10,644	2,686	2,053
25 to 29 years old	17,806	438	219	1,294	239	5,212	9,780	1,526	4,151	736	173	61
30 to 34 years old	19,536	624	272	1,221	248	5,810	3,562	1,920	4,515	1,000	283	192
35 to 39 years old	21,945	598	267	1,457	248	7,106	3,905	2,110	4,409	1,270	357	253
40 to 49 years old	43,172	1,901	622	2,525	475	14,191	7,820	4,309	8,060	2,360	656	482
50 to 59 years old	30,900	935	794	1,893	319	10,083	5,069	2,479	5,150	2,555	619	525
60 to 64 years old	10,447	514	448	1,046	95	3,803	1,915	569	1,309	638	153	152
65 years old and over	26,376	2,134	3,123	4,226	518	11,572	12,419	1,514	3,278	1,349	344	326
Male												
18 and over	98,112	3,342	2,918	8,977	1,539	31,042	18,964	6,827	16,177	5,213	1,719	1,467
18 and 19 years old	4,121	41	38	1,293	253	1,074	1,048	22	4	1	1	1
20 to 24 years old	9,396	207	122	942	240	3,112	3,271	573	843	26	13	7
25 years old and over	84,637	3,005	2,758	6,329	1,031	26,458	14,645	6,233	15,231	5,103	1,707	1,480
25 to 29 years old	8,816	249	122	679	110	2,265	1,846	342	1,901	311	62	32
30 to 34 years old	9,533	301	123	638	126	3,096	1,647	825	2,152	479	148	136
35 to 39 years old	10,932	229	155	735	124	3,752	1,820	861	2,124	547	192	164
40 to 49 years old	21,072	557	152	1,242	279	6,986	3,741	1,901	3,947	1,389	460	333
50 to 59 years old	14,967	424	392	901	149	4,341	2,818	1,127	2,730	1,275	440	362
60 to 64 years old	4,847	233	229	449	46	1,566	741	229	724	370	119	136
65 years old and over	14,170	1,302	1,415	1,714	199	4,256	1,399	567	1,753	727	277	280
Female												
18 and over	105,876	3,305	3,136	9,290	1,461	24,797	21,108	8,876	16,890	5,536	890	564
18 and 19 years old	4,010	34	34	1,300	186	1,248	1,176	31	1	1	1	1
20 to 24 years old	9,490	134	100	754	166	2,646	3,641	632	1,317	79	13	10
25 years old and over	92,386	3,137	3,002	7,306	1,110	30,493	16,290	8,213	15,513	6,451	879	594
25 to 29 years old	6,897	187	98	614	129	2,348	1,935	369	2,230	424	112	29
30 to 34 years old	10,003	223	149	613	122	2,721	1,914	1,004	2,363	611	136	168
35 to 39 years old	11,050	229	155	735	124	3,354	2,085	1,257	2,277	627	188	160
40 to 49 years old	20,490	534	160	1,262	197	6,234	4,345	2,318	4,174	1,577	476	358
50 to 59 years old	16,034	511	402	902	171	5,712	2,953	1,352	2,420	1,280	369	283
60 to 64 years old	5,500	280	276	597	51	2,235	868	359	565	314	35	57
65 years old and over	18,799	1,331	1,709	2,506	319	7,319	2,586	957	1,525	622	67	69
White, non-Hispanic												
18 and over	146,752	1,519	3,890	10,554	1,597	48,010	30,347	12,261	26,524	8,346	2,164	1,670
18 and 19 years old	5,221	14	35	3,700	165	3,344	1,685	43	1	1	1	1
20 to 24 years old	12,313	14	10	759	155	3,705	4,905	893	1,703	AD	7	3
25 years old and over	131,220	1,491	3,722	8,375	1,196	43,711	29,958	11,395	24,801	8,893	2,157	1,867
25 to 29 years old	11,450	46	26	653	93	3,280	2,568	1,073	3,079	544	121	39
30 to 34 years old	13,051	37	97	605	107	3,707	2,434	1,417	3,479	847	196	140
35 to 39 years old	15,965	57	69	771	115	4,393	2,782	1,563	2,455	991	287	182
40 to 49 years old	31,400	150	300	1,370	261	10,456	5,935	2,431	6,520	2,525	666	389
50 to 59 years old	24,077	193	475	1,096	160	7,975	4,666	2,045	4,323	2,226	632	457
60 to 64 years old	8,168	137	288	720	59	3,153	1,366	469	1,091	588	133	162
65 years old and over	27,336	871	2,453	3,270	399	10,178	4,150	1,315	2,866	1,193	322	299
Black, non-Hispanic												
18 and over	23,596	810	666	3,373	584	8,447	4,961	1,703	2,422	728	158	76
18 and 19 years old	1,189	2	2	564	65	316	237	1	1	1	1	1
20 to 24 years old	2,678	11	10	390	111	943	970	144	174	4	14	4
25 years old and over	19,790	599	641	2,415	408	7,187	3,793	1,559	2,247	723	145	71
25 to 29 years old	2,436	8	16	239	54	899	686	210	378	34	18	10
30 to 34 years old	2,523	25	28	144	34	1,012	565	214	619	62	21	98
35 to 39 years old	2,790	30	22	128	44	1,046	586	224	368	109	17	5
40 to 49 years old	5,134	62	44	514	89	2,020	1,089	487	563	200	44	31
50 to 59 years old	3,144	66	93	453	73	1,120	666	234	284	105	31	20
60 to 64 years old	1,391	29	75	165	23	524	149	60	89	42	5	2
65 years old and over	2,742	81	364	693	71	735	213	60	149	114	8	13
Hispanic												
18 and over	21,830	4,092	1,292	3,422	652	6,113	3,160	1,029	1,525	361	106	71
18 and 19 years old	1,276	66	37	536	102	304	228	7	1	1	1	1
20 to 24 years old	2,922	308	117	493	125	941	767	115	90	4	2	1
25 years old and over	17,686	3,725	1,138	2,393	425	4,867	2,225	807	1,442	357	105	71
25 to 29 years old	2,798	373	127	499	79	836	420	163	255	28	18	9
30 to 34 years old	2,573	457	148	417	89	551	402	169	265	45	19	8
35 to 39 years old	2,722	498	184	419	65	1,179	359	164	243	54	12	9
40 to 49 years old	4,074	797	229	532	97	1,121	555	229	365	43	32	18
50 to 59 years old	2,536	608	165	282	53	656	302	122	175	66	17	16
60 to 64 years old	1,311	272	71	101	19	234	112	34	29	12	7	4
65 years old and over	1,842	719	238	213	29	598	116	55	77	28	2	7

*Rounds to zero.
NOTE: Total includes other racial/ethnic groups not shown separately. Although cells with fewer than 75,000 weighted persons are subject to relatively wide sampling variation they are included in the table to permit various types of aggregations. Detail may not sum to totals due to rounding.
SOURCE: U.S. Department of Commerce, Bureau of the Census, Current Population Survey, unpublished data. (This table was prepared September 2002.)

Source: US National Center for Education Statistics, <http://nces.ed.gov/programs/digest/d02/tables/PDF/table9.pdf>, 2003

Table 3.5 - A Sample Statistical Table for Scenario 2

Scenario 3: At least one TBS has more than one dimension hierarchy

A combination of the rules for the above two scenarios is adopted here. All identical hierarchies are singled out and become hierarchies for the Row Heading region, in a manner similar to scenario 2. The remaining hierarchies in each TBS are merged together into one single hierarchy, and then these single hierarchies in the TBS's are merged together into one single hierarchy, in a manner similar to scenario 1. For example, consider the more complicated table in Table 3.6. There are in fact only two TBS's. Here, each TBS has one common hierarchy, which has two members: 'Total persons in households' and 'Average number of persons in household'. Based on our strategy, this hierarchy is singled out and the other hierarchy in the TBS's would be merged together. Another hierarchy is composed of 'Total households' (from the first TBS), 'Single-detached house', 'Apartment, five or more storeys', 'Movable dwelling', and 'Other dwelling' (from the second TBS).

Definitions and notes	2001				
	Saskatoon	Calgary	Edmonton	Vancouver	Victoria
	number				
Total households	88,945	356,370	356,515	758,715	135,605
Total persons in households	222,155	941,635	924,230	1,963,645	305,295
Average number of persons in household	2.5	2.6	2.6	2.6	2.3
Single-detached house	54,870	218,630	211,670	327,650	68,840
Total persons in households	158,690	667,395	632,410	1,051,580	184,275
Average number of persons in household	2.9	3.1	3.0	3.2	2.7
Apartment, five or more storeys	4,695	23,925	23,145	89,780	6,820
Total persons in households	6,295	36,300	33,225	147,675	9,485
Average number of persons in household	1.3	1.5	1.4	1.6	1.4
Movable dwelling	760	1,970	5,730	5,230	1,685
Total persons in households	1,910	4,260	14,265	9,485	3,020
Average number of persons in household	2.5	2.2	2.5	1.8	1.8

Other dwelling	28,620	111,845	115,980	336,055	58,255
Total persons in households	55,265	233,690	244,330	754,905	108,515
Average number of persons in household	1.9	2.1	2.1	2.2	1.9

Source: Statistics Canada, Census of Population.

Last modified: 2005-01-18.

Source: Statistics Canada, <http://www.statcan.ca/english/Pqdb/famil27a.htm>, 2003 July

Table 3.6 - A Table with TBS's Having More Than One Hierarchy

3.4 Table Integration

One of the main advantages of our approach, that is, recovery of the table as a multidimensional object, is the ability of integrating similar tables seamlessly. Here we consider one of the most obvious kinds of table integration: tables with drop-down boxes, as shown in Figure 3.5.

In examining the HTML source of the example table, one may find a list of character strings associated with each drop-down box. The top drop-down box in the table of Figure 3.5 is associated with names of the country, provinces, and territories, while the one lower is associated with the two years 1996 and 2001. For each pair of strings from each list will correspond to a specific table for a specific region and a specific year.

To integrate tables which are thus parameterized, new hierarchies are created, one for each drop-down box, in addition to the dimension hierarchies that are created as a result of the table processing described in the previous sections.

The screenshot shows the Statistics Canada website interface. At the top, there are logos for Statistics Canada and the word 'Canada'. Below these are navigation links: Français, Contact Us, Help, Search, and Canada Site. A secondary row of links includes The Daily, Canadian, Community, Our products, and Home. A third row includes Census, Statistics, Profiles, and services, with an 'Other links' link below. A breadcrumb trail reads: Topic-based Tabulations > Age and Sex > 97F0003XCB2001001. The main heading is 'Age (123) and Sex (3) for Population, for Canada, Provinces, Territories, Census Metropolitan Areas 1 and Census Agglomerations, 1996 and 2001 Censuses - 100% Data'. Below this is a drop-down menu for 'Select another geographic area for this product:' with 'Vancouver' selected. The table title is 'Vancouver'. Below the title is another drop-down menu for 'Select another dimension for this product:' with 'census year (2)' selected and '2001' chosen. A 'Refresh' button is present. The table has columns for 'TITLE', 'Total - Sex', 'Male', and 'Female'. The data rows are as follows:

TITLE	Sex (3)		
	Total - Sex	Male	Female
Total - Age	1,986,965	972,730	1,014,235
0-4	104,810	53,850	50,960
Under 1	20,420	10,545	9,875
1	20,615	10,565	10,055
2	20,935	10,780	10,160
3	20,935	10,630	10,310
4	21,900	11,335	10,565
5-9	117,975	60,800	57,175
5	23,140	11,880	11,260
6	23,575	12,285	11,290
7	23,560	12,145	11,410
8	23,805	12,190	11,615
9	23,900	12,300	11,600
10-14	122,610	62,805	59,805
10	24,195	12,325	11,870

Source: Statistics Canada, <http://www12.statcan.ca/english/census01/Products/standard/themes/DataProducts.cfm>, December 2004

Figure 3.5 – Table with Drop-Down Boxes

3.5 Linguistic Processing

Linguistic processing is about analyzing the texts having been identified as metadata, column and row headers, for refinements of the dimension hierarchies in the multidimensional data model. However, its targets are very specific; concept hierarchies are not involved at all, except in a very general way. We have already mentioned one example: the task to ascertain a Table Body Section (TBS) as the summary section. The processing is quite straightforward: scanning for specific words such as 'total',

'average' and 'median'. In general, linguistic processing techniques are applied to three groups of entities: aggregates, measurements and temporal entities. To this end, we have built *pattern group* for each group of entities. Reasoning is applied to headers in various hierarchies, aided by these pattern groups, during or after the formation of the dimension hierarchies.

The measurement pattern group consists of words for units and corresponding symbols associated with measurements of all kinds of quantity commonly used in statistical tables, in terms of area, length, volume, money, time and so on. Special keywords and symbols such as “%”, “thousand” and “million” are also included. The main purpose for this measurement pattern group is ascertain the true value of the numeric data contained in the data cells, so that they can be subject to proper statistical functions, such as summation. For example, if the data value is 10.9 and the unit is found to be “million\$”, then it will be modified to read \$10,900,000. More importantly, we can recognize a hierarchy to be a measurement hierarchy through linguistic processing. Consider Table 1.1 in the introductory chapter. By applying the heuristics as described in Chapter 3.2.1, the headers 'Area Planted' and 'Area Harvested' are found to be members of a hierarchy. 'Acres' is the lone member of another hierarchy. None of the two hierarchies are explicitly labelled as the measure hierarchy. When we match the members with words inside the measure pattern group, we can deduce that these are some measures related to area as a quantity. This deduction may be further strengthened by locating the header “Acres” as a unit of measurement for area. Thus we are able to merge these hitherto separate hierarchies together and make it the measure hierarchy.

Processing temporal entities aims at identification of temporal entities, such as day, date, month, and year. Since many statistical tables have a temporal dimension,

identification of this dimension is essential to cross-table processing. This information is useful also for question-answer information retrieval systems. For example, it can be used to answer temporal range queries such as the total crop in area harvested from 1997 to 1999, in the context of Table 1.1.

Identification of aggregate members inside a dimension hierarchy is an important task. The most common aggregate function by far, is total. An aggregate pattern group contains all sorts of variations of the word. The aggregate member ‘total’ may be confirmed by adding the values associated with primary members at the same level of the dimension hierarchy. There are, however, other functions that are not so easily recognized, particularly those that are defined by a formula. Fortunately, these formulae are often defined as part of the header, which contains some special keywords such as total or median.

There are other opportunities for linguistic processing that we are still looking into. The caption, header and footer are fertile grounds for additional semantic meaning. For example, the table’s title *“Individuals living in low income, by age, Canada, 1980, 1990 and 2000”* contains information about the names of the measures and the dimensions in the table. Obviously, natural language processing techniques are required to extract the useful information from the text.

3.6 Building Multidimensional Database

Once all dimension hierarchies have been derived, a relation for the multidimensional dataset is built and populated by data extracted from data region in each DBS. Structured as a relation in a relational database system, this dataset has a column for every hierarchy, plus a column for each leaf member in the measure hierarchy. Each row in the relation is generated by a data cell, which is associated with

at least one member of each hierarchy, including the measure hierarchy. For this data cell, the name of the member associated with a concept hierarchy is entered as a row under the appropriate column, and the value of the data cell is entered under the appropriate member of the measure hierarchy in the same row, as shown in Figure 2.5 (iv) as an example. In case where two members from the same hierarchy are associated with the cell, for instance, Corp and Asparagus, we may include both of them as the entry in row under the column of the relation. Alternatively, an ID is created for member at the lowest level, with a pointer connecting it to the appropriate hierarchy. An example is shown in Chapter 4.2.2.7.

CHAPTER 4 SYSTEM DESIGN AND EXPERIMENTAL PERFORMANCE EVALUATION

As a proof of concepts, a research prototype, called Web Table Miner, has been built to implement the essence of the table processing described in Chapter 3. A description of the system is presented in Chapter 4.1 and 4.2. We have run the system against a sample of 150 tables from Statistics Canada to provide some evidence on how our table processing algorithms work out against real life data. The experimental results are tabulated in Chapter 4.3.

4.1 System Overview

The Web Table Miner is a semi-automatic system which requires some interactions with the user to process a web table. Figure 4.1 illustrates the entire process of how the system works. Firstly, user submits the URL of the web page containing meaningful HTML tables and the Web Table Miner loads up the web page by using Microsoft IE browser as a COM component. With the help of the browser object (MS IE COM component), the web page is parsed. The parsed web page is then handled by the table recognizer to filter out uninteresting information and determine the table which potentially contains OLAP multidimensional data. With this semi-automatic system, user would be notified the potential multidimensional data table in the web page and decide which table(s) to process. Next, the multidimensional table miner takes over to process the selected tables and build an OLAP cube. The last step of the process is that the schema generator flattens the cube and produces the OLAP star schema representing the cube data. The output is in text format in which the dimensions, their

hierarchies, and members of the OLAP data are printed. The measure of the data is also printed if the table parser is able to detect it. The flatten cube data is also outputted in comma separated format such that it can be easily read by spreadsheet program like Excel. The future direction of this system is to have the capability of multiple data table integration given that they are similar. Moreover, this system would be more useful if it is able to transfer the found data over to an OLAP server like Analysis Service of Microsoft.

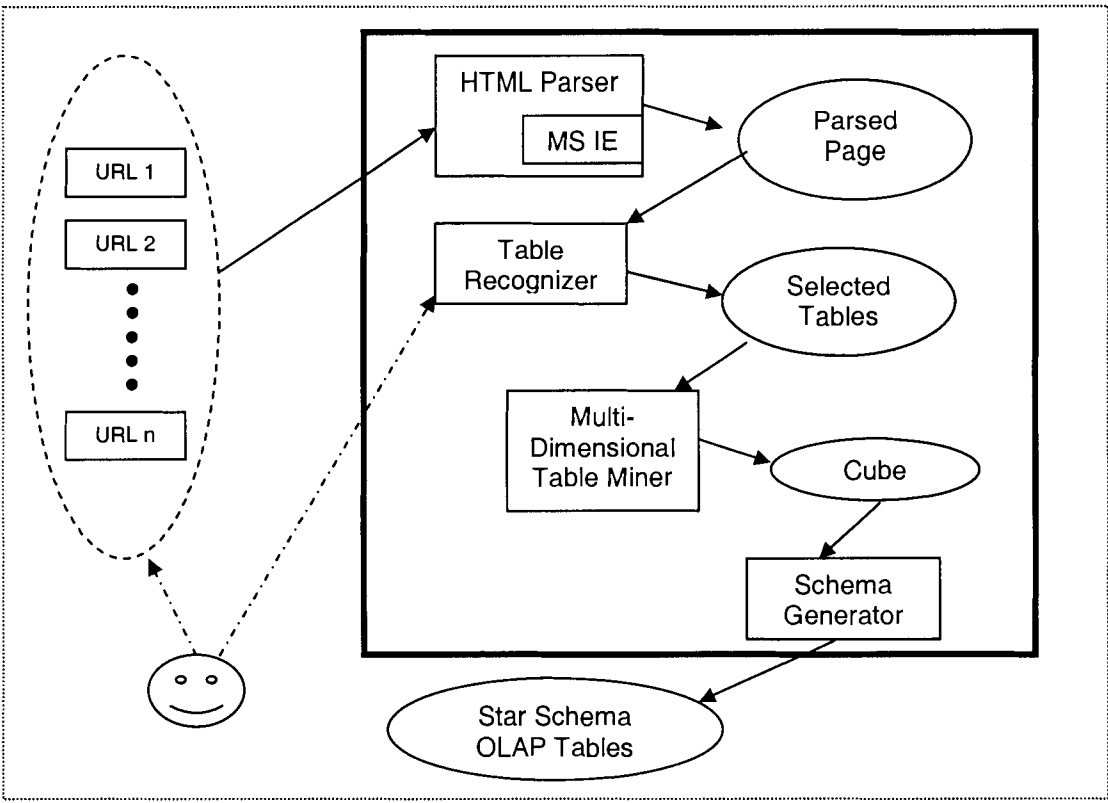


Figure 4.1 - System Overview (User Level)

4.2 System Architecture

In this section, we discuss about the internal structure of the system. We first describe the internal structure that represents the web table and then the overview of various components in the system.

4.2.1 Internal Structure of Web Table

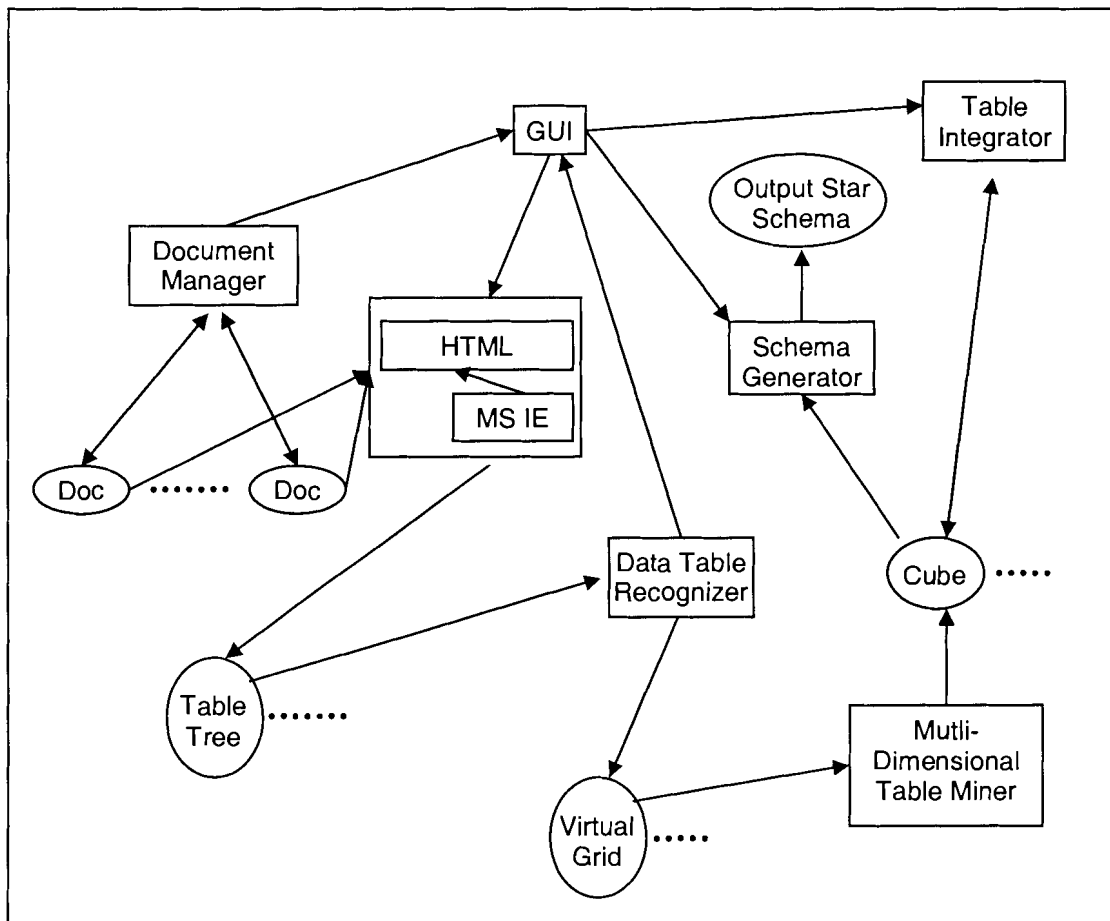
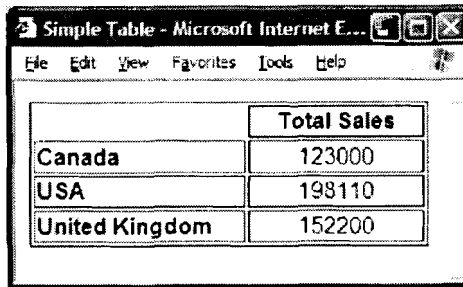


Figure 4.2 - Internal Architecture

A HTML web document consists of markup language used to describe the page. HTML tags are used in order to do so. These tags are organized in a hierarchical format inside the document, as illustrated in Figure 4.3 and 4.4 below.



The screenshot shows a browser window titled "Simple Table - Microsoft Internet E...". The menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The main content area displays a table with the following data:

	Total Sales
Canada	123000
USA	198110
United Kingdom	152200

```
<html>
<head>
<title>Simple Table</title>
</head>

<body>
<table border="1" bordercolor="#111111">
  <tr>
    <td> </td>
    <td align="center"><b>Total Sales</b></td>
  </tr>
  <tr>
    <td><b>Canada</b></td>
    <td align="center">123000</td>
  </tr>
  <tr>
    <td><b>USA</b></td>
    <td align="center">198110</td>
  </tr>
  <tr>
    <td><b>United Kingdom</b></td>
    <td align="center">152200</td>
  </tr>
</table>
</body>
</html>
```

Figure 4.3 - A Simple Web Page Containing Table

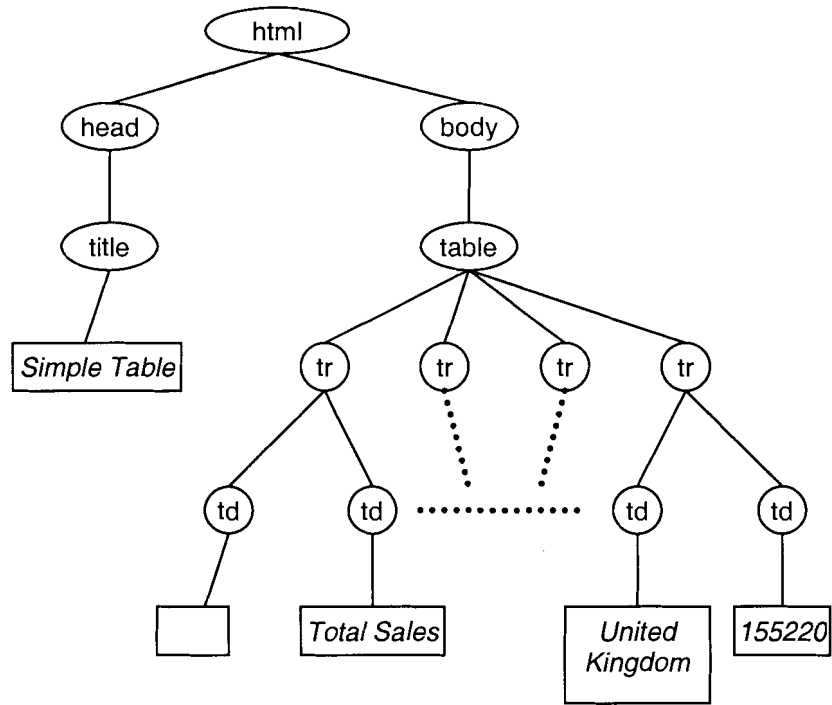


Figure 4.4 - Hierarchical Tag Tree of the Simple Web Page

Our system is only interested in the web table which contains multidimensional data. After parsing the HTML page, we build up the table structure for each table found inside. The web table is mapped onto a grid. By doing so, each table cell in the table can be easily referred by its corresponding row and column indexes. It is common that a table cell spans multiple rows and columns. Such cell is represented by multiple grid cells internally.

4.2.2 Components

Internally, the Web Table Miner consists of seven major components, namely, the GUI, document manager, HTML parser, table recognizer, multidimensional table miner, table integrator and schema generator. The focus of our research lies primarily in the table recognizer and multidimensional table miner.

4.2.2.1 GUI Manager

GUI (Graphic User Interface) Manager – Provides an interaction layer between the user and the other components in the system. It provides view to the web page that the user submits. Essentially the view is a browser as our system takes advantage of the MS Internet Explorer COM component which parses and renders the page. MS IE COM component also exposes a DOM (Document Object Model) for programmer to access or modify the loaded web page. Multiple views can be opened to view different web pages at the same time. Figure 4.5 below displays a screen shot of the system.



Figure 4.5 - Screen Shot of Web Table Miner

4.2.2.2 Document Manager

Document Manager – Simply maintains information on all open web pages. Each web page is referred to as a document. Once a web page has been rendered by the Internet Explorer object, a document is created in the system to hold information

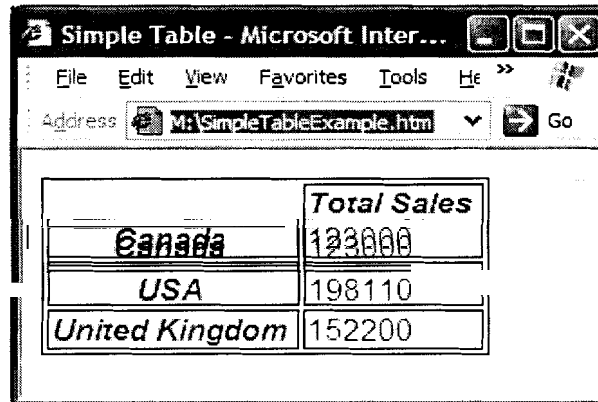
regarding the parsed web page, such as a reference to its DOM, reference to the table tree and user specified value (tables selected to be processed) applicable to the document.

4.2.2.3 HTML Parser

HTML Parser – Initiates the building of the table tree of a parsed web page. It searches through the HTML's tag tree via the DOM and builds up a tree of tables in the web page. In addition, it checks whether external or header cascading style sheet is used to format the web page. The reason is that style sheet contains information about the visualization of the web page and thus contains valuable information to our table mining algorithm. If so, it retrieves information possibly related to table formatting, such as font properties (type, style, weight and size), colour, text properties (indentation), etc... Figure 4.6 displays an example.

4.2.2.4 Data Table Recognizer

Data Table Recognizer – Processes each table in the table tree and determines whether it is a genuine data table which we are interested in. Identification of multidimensional statistical table is based on a couple of simple heuristics on table's structure and visualization cues. In short, we filter out non-data tables that contain only hyperlinks, forms or images. Another heuristic the recognizer based on is the fact that statistical table has a structure layout which the row and column header areas contain mostly texts and the remaining data area contains mostly numerical data. The Data Table Recognizer also builds a virtual grid which maps onto the web table for easy access to the web table. As mentioned earlier, we could access any cell of the table by using row and column indexes. Figure 4.7 displays an example of virtual grid.



```
<html>
<head>
<title>Simple Table</title>
<style type="text/css">
table
{
  font-family: arial;
  color: red;
}
td.label
{
  text-align: center;
  font-style: italic;
  font-weight: bold;
}
</style>
</head>

<body>
<table border="1" bordercolor="#111111">
  <tr>
    <td> </td>
    <td class=label>Total Sales</td>
  </tr>
  <tr>
    <td class=label>Canada</td>
    <td>123000</td>
  </tr>
  <tr>
    <td class=label>USA</td>
    <td>198110</td>
  </tr>
  <tr>
    <td class=label>United Kingdom</td>
    <td>152200</td>
  </tr>
</table>
</body>
</html>
```

Figure 4.6 - Simple Web Table Using CSS

	<i>Total Sales</i>	
	<i>1997</i>	<i>1998</i>
	<i>\$ millions</i>	
<i>Canada</i>	61300	61000
<i>USA</i>	99110	99000
<i>United Kingdom</i>	82200	70000

	<i>Total Sales</i>	
	<i>1997</i>	<i>1998</i>
	<i>\$ millions</i>	
<i>Canada</i>	61300	61000
<i>USA</i>	99110	99000
<i>United Kingdom</i>	82200	70000

Figure 4.7 - A Statistical Table and Its Corresponding Virtual Grid

4.2.2.5 Multidimensional Table Miner

Multidimensional Table Miner – Regarded as the most important component of our system. As its name implies, it digs deep into the web table and searches for multiple dimensional data such as dimensions, measure and the data. The details of the methodology are discussed in chapter 3. Briefly, by understanding the layout and structure of the web table, the miner discovers the core components of a multiple dimensional data cube.

4.2.2.6 Table Integrator

Table Integrator – Integrates multiple web tables which contain similar data context. Data presented on web page are limited to two dimensions. As a result, data from cube needs to be sliced into multiple tables during presentation. This components aims to integrate these sliced tables together. This component is not yet completed and is a future research direction.

4.2.2.7 Schema Generator

Schema Generator – Converts the data cube and generates a star schema for the cube. Last but not least, a star schema for the web table is outputted on a comma separated value (csv) format file. The output contains dimension table for each

dimension discovered in the web table, and the fact table which contains the numerical values and measures, as shown in Figure 4.8.

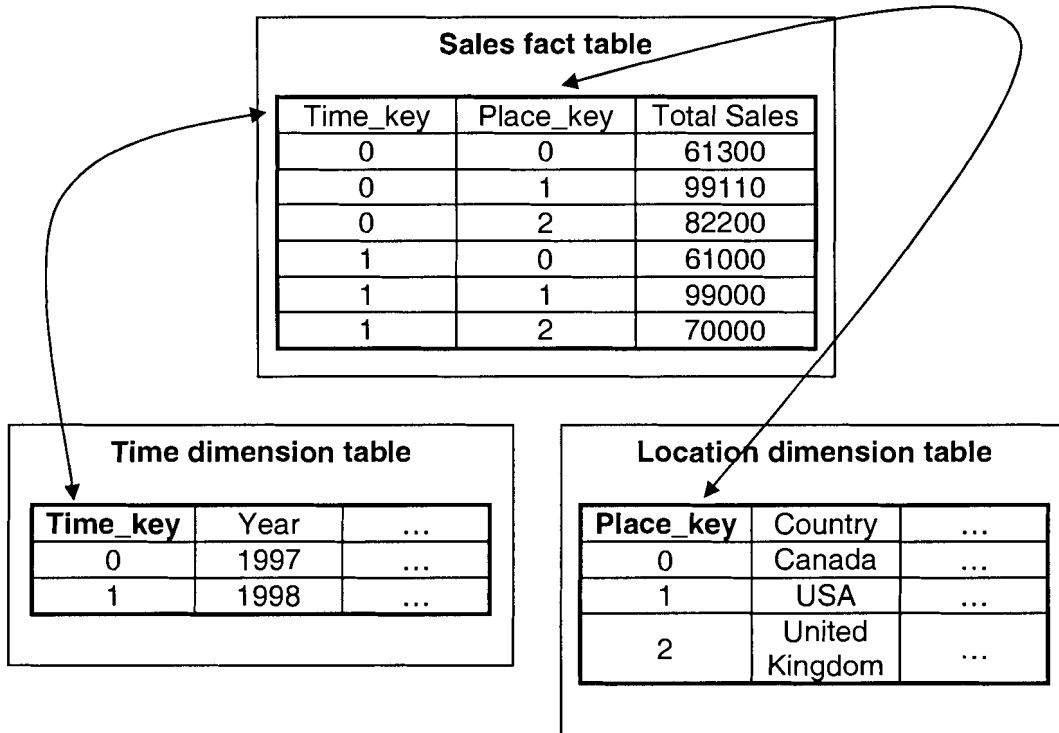


Figure 4.8 - Star Schema of OLAP Data Cube

4.3 Performance Assessment

Having examined the hundreds of web tables from government agencies in many countries, we are confident that our approach will work for a very large majority of the statistical tables. Unfortunately, many tables are published in PDF format, which are raster images. Though we have looked into some commercially available conversion tools, we could not find any that are capable of converting tables in PDF format to ones in HTML format that are normally produced by most authoring tools. We have chosen the website of Statistics Canada as the source for our experimental data because most of the tables are in both HTML and PDF formats. Note that many example tables cited

above that are taken from websites of governmental agencies of non-Canadian countries, and some of them are actually in print or PDF format.

We work with a set of 150 randomly selected tables from Statistics Canada web site. We evaluate the performance of our system on how well it locates (i) dimension hierarchies, (ii) metadata components, (iii) measure hierarchy and measurement units, and (iv) data components. For (i) dimension, we are interested in how well the system locates all the dimension hierarchies in a table. Moreover, (ii) metadata components include row and column headers and the parent-child relationships of these headers within a dimension hierarchy, and name of dimension. When measuring the system's performance on metadata components recognition, we check how well we can recognize the parent-child relationships of the headers. It is difficult for our system to locate name of dimensions as the name does not usually appear in the table and the system does not contain any ontology or domain specific information except temporal information. Then, we evaluate the system's ability in locating the (iii) measure hierarchy and any measuring unit associated with the data. Lastly, the performance on retrieving the (iv) data components and relating them to their corresponding dimensions' members is evaluated. Take table given in Figure 4.7 as an example, the table has two dimension hierarchies: location and time. The location dimension contains metadata components of 'Canada', 'USA', and 'United Kingdom'. All three components lie on the same level of hierarchy. The time dimension contains members of '1997' and '1998'. 'Sales' is the measure of this table and its data has measure unit of '\$ million'. Data '61300' is associated with the dimension's member '1997' of time dimension and 'Canada' of location dimension.

In order to measure the effectiveness of our system, we used three standard Information Retrieval (IR) and Information Extraction (IE) measures: *Recall*, *Precision*,

and *F-measure*. *Recall (R)* is the percentage of the pieces of information for which the system extracted is correct. Therefore, recall represents how much information the system correctly extracted and identified. *Precision (P)* is another useful measure which represents the reliability of the system as it refers to the percentage of the correctly extracted pieces of information out of all available pieces of information. For both measures, R and P, the higher these values are (both measures have value from 0 to 1 inclusive), the better our system is. These measures are defined as follows:

$$\mathbf{R} = (\# \text{ correct recognized information}) \div (\# \text{ information system proposed})$$

$$\mathbf{P} = (\# \text{ correct recognized information}) \div (\# \text{ information available in samples})$$

The third measure, *F-measure*, is a weighted harmonic mean³ of P and R. *F-measure* is designed to combine P and R with an equal weight and provide a single measurement for measuring IR and IE system performance and combine. Similarly to P and R, *F-measure* lies between 0 and 1. By the same token, a high value would represent an effective system. The *F-measure* is defined below:

$$\text{F-measure} = (2 \times P \times R) \div (P + R)$$

Table 4.1 shows the performance evaluation of our Web Table Miner. As we can see, the system performs very well on recognizing dimension hierarchies and metadata components as it achieves 0.970 and 0.978 *F-measure* respectively. These are the indications that the system is accurate and reliable in these tasks. Since our system does not rely on domain knowledge, it fails to recognize nested dimension when the nested dimension in a table does not exhibit the repeated pattern as in Figure 3.1 (iii).

³ According to <http://mathworld.wolfram.com/HarmonicMean.html>, the harmonic mean $H(x_1, \dots, x_n)$ of n numbers x_i (where $i = 1, \dots, n$) is the number H defined by:

$$\frac{1}{H} \equiv \frac{1}{n} \sum_{i=1}^n \frac{1}{x_i}$$

By the same token, our system could mistakenly treat measure dimension as ordinary dimension. Occasionally, the system fails to recognize the relationship between metadata components in the headers because they have the same visualization cue and thus are treated as same members in a hierarchy. The table in Figure 4.9 is an example. Both 'Canada' and 'British Columbia' headers share the same visual cues. Consequently, our system would not be able to determine Canada is the parent of 'British Columbia' in the dimension hierarchy. In terms of data component correctness, our system does even better in extracting the data with correct association to the dimension hierarchy and members. It has a very high F-measure value of 0.995. The case in which the system fails to extracting data components correctly is when the table's structure is not expected by the system.

Performance Indicators	Dimension Hierarchy	Metadata Components	Measure Hierarchy & Measurement Units	Data Components
Total Number Available in Samples	390	3560	212	13205
Total Number Proposed by System	387	3615	181	13190
Total Number Recognized Correctly	377	3510	168	13137
Precision	0.974	0.971	0.928	0.996
Recall	0.967	0.986	0.792	0.995
F-measure	0.970	0.978	0.855	0.995

Table 4.1 - Performance Evaluation of the Web Table Miner

However, clearly our system does not locate measure-related metadata as accurately as the other three components. The system occasionally fails to distinguish measure from ordinary dimensions due to our system's minimal linguistic processing and

lack of domain knowledge. We hope that in the future, more sophisticated linguistic processing will significantly improve the system performance. From the gathered results, we notice that the system's accuracy ($P = 0.928$) in measure metadata recognition is higher than its reliability ($R = 0.792$). When a measuring unit exists, it is easier for our system to detect it than the measure of the underlying OLAP cube because of the unit's pattern can be recognized. Measure is not always explicitly stated in the web statistical table because it could be stated outside the table like in the caption when all the data of a table belong to one single measure.

<u>Name</u> ▼ ▲	<u>Population</u>			<u>Total private dwellings, 2001</u>
	<u>2001</u>	<u>1996</u>	<u>% change</u>	
Canada †	30,007,094	28,846,761	4.0	12,548,588
British Columbia †	3,907,738	3,724,500	4.9	1,643,969
Burnaby - Douglas	119,998	113,409	5.8	49,317
Cariboo - Chilcotin †	80,469	81,881	-1.7	37,827
Delta - South Richmond	124,881	122,064	2.3	43,204
Dewdney - Alouette	121,477	111,692	8.8	45,437
Esquimalt - Juan de Fuca †	110,909	107,111	3.5	45,790
Fraser Valley	129,828	121,901	6.5	51,094
Kamloops, Thompson and Highland Valleys	100,452	99,356	1.1	43,121

Source: Statistics Canada's Internet Site, <http://www12.statcan.ca/english/census01/products/standard/popdwell/>, December 2004

Figure 4.9 - An Example in Which Visual Cues Fail to Work

Since our experiment was done on datasets from Statistics Canada only, one may argue that our system may over-fit the tables generated by this particular agency. Although no actual experiment was done on other websites, we did observe that statistical tables, published by other agencies like Statistics Denmark, Census and Statistics Department of Hong Kong SAR, and U.S. Census Bureau, do have clear layout and rich visual cues in them similarly to what we have seen in Statistics Canada. There may be slight differences between tables from different sites because they may

have different conventions in generating tables. For instance, in table 3.5, there is a row inserted in between the column heading and the table body which does not have specific meaning to the data in the table. This row is simply used to index the table's columns. We may need to adjust our heuristics to take into account of slight syntactic difference in tables from different websites. In the case of table 3.5, we should discard the indexing purpose row in our table processing. Most importantly, statistical tables from other websites do have the characteristics of clear layout and rich visual cues despite minor differences. Therefore, we believe our approach would be successful when it is applied to other datasets as well. In fact, examining statistical tables from other websites is listed as one of our future directions in Chapter 6.2.

CHAPTER 5 RELATED WORK

5.1 Statistical Table, Multidimensional Object, and OLAP

While this research is not about table generation per se, generation and understanding of tables is closely linked because there is little doubt that most, if not all, statistical tables are generated by some software systems, proprietary or otherwise, from statistical databases. Wang [WW1998] in his Ph.D. thesis presents a conceptual model of table for table generation, where a table is a map, with the unordered Cartesian product of hierarchies as its domain, and some universe of entries as its range. The number of hierarchies involved is the dimension of the table. Let's take Table 1.1 as the example. An attribute, according to Wang's model, is a member of the Cartesian product of three sets of labels: {AreaPlanted, AreaHarvested}, {1997, 1998, 1999}, and {Corp.Atrichokes, Corp.Asparagus, Corp.Beans-Lima, Corp.Beans-Snap, Corp.Broccoli}. Each label set corresponds to a dimension hierarchy, in our terminology, and the '.' that separate the two strings is indicative of a parent-child relationship inside a hierarchy. Our multidimensional data model can be similarly defined, but the range is the set of numeric data, which brings it closer to the OLAP data model. [S1997] remarks that OLAP and statistical databases serve the same purpose, statistical analysis, but its main emphasis is on comparing database modelling techniques between the two types of database. It is suggested in [GBLP1996] that a data cube, such as one shown in Figure 1.3, can be conveniently depicted as a statistical table in a 2-dimensional medium, such as a paper or screen. Of course, MDX, the multidimensional query language does exactly the same task, with many powerful reporting features.

To the best of our knowledge, there has not been any research reported in the literature on any scheme to recover the perceived multiple dimensionality of the data in a table, with one minor exception. As reported in [DHQ1995], a research project is described whose aim is to extract relevant information from tables embedded in some construction industry specifications in a plain text format. Beginning with a set of concept hierarchies describing terms commonly used in construction industry, the authors discuss how information, that could be used to generate a table, is stored in a relation, which is similar to the multidimensional dataset in our model back in Figure 2.4. This relation is defined as the canonical layout. The table processing begins with table detection. After a table has been successfully detected from the specification, it is processed in two steps. First, the “primary information”, which is the locations of the column and row headers inside the concept hierarchies, is extracted by analysis of the table layout and keyword matching. With this information, the canonical form, and hence the relation, is ‘reconstituted’, by applying natural language processing techniques. It is unclear from the paper how the table layout is analyzed, nor how successfully their table processing techniques are when they are applied to real-life data. From we have learnt from the paper, the targeted tables are very simple ones, and the contents contained in these tables are domain-specific, i.e., building construction industry. Indeed, the concept hierarchies are available in advance to assist the information extraction task.

5.2 Web Tables

Tables posted on the web do not in general conform to the generic model shown in Figure 2.3, because of the special characteristics of Web as communications medium, in comparison to those of print medium. Some authors prefer to pack all related information together as a table, regardless of the row and column indexing principle of a table. The following example, shown in Table 5.1, taken from [CTT2000], is actually an

aggregation of two tables, according to the generic table model: a table with the first two lines and another table with the rest of the lines.

Tour Code		DP9LAX01AB		
Valid		1999.04.01–2000.03.31		
Class/Extension		Economic Class	Extension	
Adult	P r i c e	Single Room	35,450	2,510
		Double Room	32,500	1,430
		Extra Bed	30,550	720
Child		Occupation	25,800	1,430
		Extra Bed	23,850	72
		No Occupation	22,900	360

Table 5.1 - A Table about Tour Packages

As Web becomes more popular, not only on desktop, but also on handheld devices, tables are created to cater to busy viewers who just need some tidbit of information as it breaks out, such as headline news. Examples of these tables are online stock quotes and weather forecast (see Table 5.2).

Last: 56.45	Change: +0.12	Open: 55.98	High: 56.97	Low: 55.428	Volume: 30,884,700
Percent Change: +0.21%		Yield: n/a	P/E Ratio: 49.96	52 Week Range: 47.50 to 76.15	
After Hours Trading 4/4/2002 6:27:00 PM		Last: 56.45	Change: UNCH	Volume: 534,000	

Table 5.2- A Table about Stock Quotation

Solutions have been proposed to process these types of web table ([YL2002], [CTT2000]). It should be noted that our processing scheme as presented here is not able to understand these types of table properly.

5.3 Recognition of Table Components

Recognition of table components without any advanced knowledge of the table contents is in general very difficult. Most of the research papers on table understanding rely on some sort of ontologies to recognize the meanings of the headers and how they

are related to each other. The TANGO (Table ANalysis for Generating Ontologies) system, which aims to generate ontologies from a statistical table ([ETL2004]), does rely on auxiliary information such as dictionaries, lexical data, and data frames (e.g. a data frame for country names) to understand tables published in CIA World Factbook.

Nonetheless, there have been attempts to understand tables without any auxiliary information. In [YZ2001], it is proposed that objects inside a web page, which includes tables, may be compared for similarities solely on the basis of the visual cues associated with the example objects that have already been recognized by means of machine learning. For tables in plain texts, a variant of the Markov chain model is used to predict which type of row the next row in the table most likely is, based on the type of the current row [PMWC2003]. However, the parameters of the model are obtained by machine learning. Using a similar approach as ours, that is, reasoning against the spatial layout of the table, TINTIN (Table INformation-based Text INquery) system is able to distinguish table captions from table data rows for tables ([PC1997]), but their component recognition rules apply to only free texts, and they do not appear to work for complex tables for data analysis.

CHAPTER 6 CONCLUSION AND FUTURE DIRECTIONS

6.1 Major Contributions

New approach to processing web tables: Table processing for unrestricted tables is a very difficult task. To be convinced, one needs only a glance to the diverse definitions of a table given in the beginning of this paper. Web as a communication medium for tables has both negative and positive implications for table processing. On one hand, the wide availability of authoring tools leads to greater variability of table formats, as the authors become more creative in publishing tables. The lack of grid-like regularity in cell assembly and the use of table for holding un-related pieces of information together within a grid have made understanding of web tables a harder problem than tables in plain text. On the other hand, the table author is more adept in employing all kinds of visual cues to get across to the reader how the table contents should be understood. This is particularly so for tables which contain complex information. The traditional techniques in table understanding, for example, natural languages, ontology, and machine learning, have not taken full advantages of the visual cues that are embedded in the HTML text, to deal with web tables. The approach we have presented is customized for complex tables published on the web.

New component model of table: Most of the models in the published literature are about the kinds of components of which the table is made up, and their geometric layout. Wang's model is one of the most popular ones. These general-purpose models are not well-suited for web tables that are analytic in nature, that is, statistical tables, which tend to be narrow and long. As a result, the table body is

normally segmented into sections, which, though all related, are potentially smaller tables by themselves when they are coupled with the column headers.

Restoration of table as a multidimensional object: Despite of several references to table as a multidimensional object, there has not been any serious attempt to interpret the table contents so as to establish the table as a multidimensional object. While the restoration is interesting on its own right, its practical significance is far greater. With row and column headers classified into multiple hierarchies, the keywords related to time and measure dimensions are easily identified, which greatly facilitates further processing to recognize those keywords.

Querying of statistical tables via OLAP systems: The main advantage of procuring a table as a multidimensional object is the ability to feed the table contents to an off-the-shelf OLAP system. With the contents in the OLAP system, not only one may be to query the table via a standard multidimensional query language like MDX, but also to integrate related tables together. With querying, one may transform the table into a format that is more suitable for the user.

6.2 Future Directions

This research confirms the viability of developing techniques for understanding a restricted class of the web tables. Also, we have shown in Chapter 5 that while our techniques work well for statistical tables, it fails to cope with other popular web tables that are different in format. As an alternative to pursuit of a general purpose table understanding technique that are able to cope with all types of web table, one may develop a method that may properly classify web tables, and apply appropriate table processing table algorithms. For example, the Table 5.1 is actually an aggregate of two tables. If this table is properly decomposed into two tables, then the table contents would be understood. In fact, our processing algorithm would generate a 5-dimensional table

object (Tour-Code, Valid, Class/Extension, Adult/Child, Price), where Price is the measure dimension.

The more immediate task for us is the re-writing of the current system into a more modular system, where individual rules are more easily modified without affecting other rules. More tables from more countries should be analyzed, in order to validate our theoretical framework for processing web statistical tables.

There is obvious advantage in applying linguistic processing in order to assist the recognition of measure dimension. We observed that the caption of the table, which is often located on top of or underneath the table, contains information that may be helpful in finding the measure dimension of the table.

BIBLIOGRAPHY

- [ASSP2001] J. Ambite, C. Shahabi, R. Schmidt and A. Philpot, "Fast Approximate Evaluation of OLAP Queries for Integrated Statistical Data", National Conf. for Digital Government Research, Los Angeles, 2001.
- [C1993] E. Codd, "Providing OLAP: An IT Mandate", Unpublished Manuscript, E.F. Codd and Associates, 1993.
- [CSDHK] Census and Statistics Department, HKSAR, <http://www.info.gov.hk/censtatd/eng>
- [CM2004] A. Culotta, and A. McCullum, "Confidence Estimation for Information Extraction", Proc. of Human Language Technologies Conference, 2004.
- [CTT2000] H. Chen, S. Tsai, and J. Tasi, "Mining Tables from Large Scale HTML Texts", In Proc. 18th International Conference on Computational Linguistics, Saarbrücken, Germany, July 2000
- [DHQ1995] S. Douglas, M. Hurst and D. Quinn, "Using Natural Language Processing for Identifying and Interpreting Tables in Plain Text", In Proc. of the 4th Annual Sym. on Document Analysis and Information Retrieval, Las Vegas, 1995.
- [ETL2004] D. Embley, C. Tao, and S. Liddle, "Automating the Extraction of Data from Tables with Unknown Structure", Data & Knowledge Engineering, 2004 (to appear).
- [G1993] J. Grossman (editor), "Chicago Manual of Style", Ch. 12 (Tables). University of Chicago Press, 14th edition, 1993.
- [GBLP1996] J. Gray, A. Bosworth, A. Layman, H. Prahesh, "Data Cube: A Relational Aggregation Operator Generalizing group-BY, Cross-Tabs, and Sub-Totals", Proc. of ICDE '96, New Orleans, February, 1996.
- [HKLW2000] J. Hu, R. Kashi, D. Lopresti and G. Wilfong "A System for Understanding and Reformulating Tables" , Fourth ICPR Workshop on Document Analysis Systems (DAS'2000), Rio De Janeiro, Brazil, December 2000.
- [HKLW2002] J. Hu, R. Kashi, D. Lopresti and G. Wilfong, "Evaluating the performance of table processing algorithms" International Journal on Document Analysis and Recognition, Vol. 4, No. 3, March 2002.
- [HN2000] M. Hurst, and T. Nasukawa, "Layout and Language: Integration Spatial and Linguistic Knowledge for Layout Understanding Tasks", In Proc. of 18th International Conference on Computational Linguistics (COLING 2000), 2000.
- [LN1999] S. Lim and Y. Ng, "An Automated Approach for Retrieving Hierarchical Data from HTML Tables". In Proceedings of ACM CIKM'99, pp. 466-474, Kansas City, MO, USA, November 1999.

- [LN2000] D. Lopresti and G. Nagy, "A Tabular Survey of Table Processing", In: A. Chhabra and D. Dori (eds): "Graphics Recognition – Recent Advances", LNCS 1941, Springer Verlag, 2000
- [LNY2002] S. Lim, Y. Ng, and X. Yang, "Integrating HTML Tables Using Semantic Hierarchies and Meta-Data Sets", Proc. of International Database Engineering and Application Symposium (IDEAS'02), pp. 160-169, IEEE Computer Society, Edmonton, Canada, July 17-19, 2002.
- [M] Microsoft Corp. MDX, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agmdxbasics_04qg.asp
- [NCES] National Center for Education Statistics, USA, <http://nces.ed.gov>
- [Pande2002] A. Pande, "Table Understanding for Information Retrieval", M. Sc. Thesis, Virginia Poly. Inst. and State University, Va., USA, 2002.
- [PC1997] P. Pyreddy and W. Croft, "TinTin: A System for Retrieval in Text Tables", In Proc. 2nd Int. Conf. Digital Libraries, 1997.
- [Pinto2002] D. Pinto, W. Croft, M. Branstein, R. Coleman, M. King, W. Li, and X. Wei, "Quasm: A System for Question Answering Using Semi-structured Data", In Proc. of the JCDL 2002 Joint Conference on Digital Libraries, pages 46–55, 2002.
- [PMWC2003] D. Pinto, A. McCallum, X. Wei, and B. Croft, "Table Extraction Using Condition Random Fields", Proc. of ACM SIGIR, 2003.
- [S1997] A. Shoshani, "OLAP and Statistical Databases: Similarities and Differences", Tutorial of PODS 1997.
- [SC] Statistics Canada, <http://www.statcan.ca>
- [SD] Statistics Denmark: *Number of persons and course participants by area, educational area, highest education previously completed, age, national origin, sex and time.*
<http://www.statistikbanken.dk/statbank5a/SelectVarVal/Define.asp?Maintable=VEU21&PLanguage=1>
- [T2004] Y. Tijerino, D. Embley, D. Lonsdale, Y. Ding, and G. Nagy, "Towards Ontology Generation from Tables", submitted, April 2004 (<http://www.deg.byu.edu>).
- [TYM2004] A. Tengli, Y. Yang and N. Ma, "Learning Table Extraction from Examples", In Proc. of 22nd International Conference on Computational Linguistics (COLING 2004), 2004.
- [W1975] D. Waltz, "Understanding Line Drawings of Scenes with Shadows", in P.H. Winston (ed.) The Psychology of Computer Vision, McGraw-Hill, 1975.
- [WH2002] Y. Wang and J. Hu, "A Machine learning based approach for Table Detection on The Web", The 11th International World Wide Web Conference (WWW2002), Honolulu, Hawaii, USA, May 2002.

- [W1996] X. Wang, "Tabular Abstraction, Editing and Formatting", PhD thesis, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1996, Available as Research Report CS-96-09, Department of Computer Science, University of Waterloo.
- [WW1998] X. Wang, and D. Wood, "A Conceptual Model for Tables", In Proc. of Principles of Digital Document Processing, 4th International Workshop, PODDP'98, Saint Malo, France, March, 1998, Lecture Notes in Computer Science, Vol. 1481.
- [YL2002] Y. Yang and W. Luk, A Framework for Web Table Mining, ACM Workshop in Web Information and Data Management (ACM WIDM'02), McLean, Virginia, USA, November, 2002.
- [YTT2001] M. Yoshida, K. Torisawa, and J. Tsujii, "A Method to Integrate Tables of the World Wide Web, In Proc. 1st International Workshop on Web Document Analysis", Seattle, WA, USA, September 2001, pp. 31-34.
- [YZ2001] Y. Yang and H. Zhang, "HTML Page Analysis Based on Visual Cues", 6th Int. Conf. on Document Analysis and Recognition (ICDAR '01), Seattle, 2001.
- [ZBC2003] R. Zanibbi, D. Blostein, and J. Cordy, "A Survey of Table Recognition: Models, Observations, Transformations, and Inferences", International Journal of Document Analysis and Recognition, 2004.