AN AID TO CONVERT SPREADSHEETS TO HIGHER QUALITY PRESENTATIONS

A Thesis

by

WASIU OLANIYI OLAJIDE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2004

Major Subject: Computer Science

AN AID TO CONVERT SPREADSHEETS TO HIGHER QUALITY PRESENTATIONS

A Thesis

by

WASIU OLANIYI OLAJIDE

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Bart Childs
(Chair of Committee)

Mac Lively
(Member)

Rodger J. Koppa
(Member)

Valerie E. Taylor
(Head of Department)

May 2004

Major Subject: Computer Science

ABSTRACT

An Aid to Convert Spreadsheets
to Higher Quality Presentations. (May 2004)

Wasiu Olaniyi Olajide, B.S., University of Ibadan
Chair of Advisory Committee: Dr. Bart Childs

A table is often the preferred medium for presenting quantative information. In some cases the presentation of quantative information can be presented as textual data or graphics at a loss of precision and clarity. The subject of this thesis is to aid the extraction and production of quality tables from a common means of preparing data in tabular form, the spreadsheet.

Spreadsheet processors are in common use. Many tables are prepared by a range of users from the naïve users to experts in graphic arts. Spreadsheet data is also produced in automatic form from applications.

We will review the specification of tabular data, presentation formats, and the systems and their associated formats for storing and interchange of data. The goal of this research is the specification and development of a system to convert common spreadsheet data to a markup language that will allow for presentation of the data at a higher level of typographic excellence. The desired characteristics of this system will include

1. Robust importing of data from an array of commercial and open spreadsheet processors

2. Formatting decisions of the output specified by the user rather than taken from the spreadsheet

3. Development or identification of a canonical form that is robust, does not lose data, and allows for repeated automatic application of styles

4. Development of a program to convert this canonical form into a markup system.

To The Almighty, without whom nothing is possible.

ACKNOWLEDGMENTS

I wish to express my sincere gratitude to Dr. Bart Childs who not only agreed to be my advisor, but kept me focused on the directions and emphasis of this thesis work. I consider myself fortunate to have had the experience to work with him, and I appreciate his helpful and consistent guidance.

I thank Dr. Lively and Dr. Koppa for their invaluable inputs on this thesis work and my graduate school experience. I appreciate the opportunity to tap into their knowledge reservoir and for fitting me into their busy schedules. I thank Tunde for his willingness to help at the drop of the hat. He ensured all my paperwork were turned in on time.

I thank my parents and siblings for doing a very good job as my support system. The love seed has been sown, nurtured and will continue to flourish.

And finally, I thank Tairat for everything she is to me. I appreciate her insistence on, and support in the course of, pursuing my dreams.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A table is often the preferred medium for presenting quantitative information. In some cases the presentation of quantitative information can be presented as textual data or graphics at a loss of precision and clarity. The subject of this thesis is to aid the extraction and production of quality tables from a common means of preparing data in tabular form, the spreadsheet.

While the definition of a table might not be easy to reconcile across different sources, each definition emphasizes a table as being composed of rows and columns and it is easy to recognize a table in a document [1].

The definition of a table which is taken from [2], is an orderly arrangement of data, esp. one in columns and rows.

Tables are, in most cases, designed primarily to convey some kind of information and therefore, the ability of the table to convey the intended information is an important design consideration. The advent of computers has given rise to software tools for producing consistently high quality tables with spreadsheet software. The spreadsheet editing software provides a means to do significant mathematically related functions and lookup. Many tables are prepared by a range of users from naïve users to experts in graphic arts. Spreadsheet data is also produced in automatic from applications on the internet.

Historically, a spreadsheet is known as a “large sheet of paper with columns and rows that lays everything out about transactions for a business person to examine, It spreads or shows all of the costs, income, taxes, etc. on a single sheet of paper for a manager to look at when making a decision” [3].

A field that makes great use of tables is the accounting profession and initial spreadsheet processors were developed mainly as accounting aids [3].

The use of the spreadsheet processor has led to most items that are tabular in

The journal model is *IEEE Transactions on Automatic Control.*

nature to be prepared using these common systems. The most popular processor is undoubtedly the Microsoft Excel system. There are several others used at a significant level and are both proprietary and open types. Most of the systems do not suffer from the frequent changes in the Excel product which is apparently driven by desire to cause dependence on the vendor than to improve quality.

Most products can accept output from recent Excel systems. The latest version of these systems does not offer an open, portable means of producing graphically excellent tables for use with open systems. There is not a public standard for interchange of spreadsheet information for porting between different systems. The nearest to that is the fact that market dominance has led most systems having the capability of input of recent Excel formats.

This thesis work is a result of an investigation of the presentation capabilities and limitations of spreadsheets. A simple conversion method from most spreadsheets is investigated and a tool for converting these spreadsheets to high-quality presentation is described with a prototype presented. The desired characteristics of this system will include

1. robust importing of data from an array of commercial and open spreadsheet processors. Most spreadsheet processors will import from other processors but each has their preferred output format. Otherwise their output formats are quite limited, especially when compared to document preparation systems.

2. formatting decisions of the output specified by the user rather than taken from the spreadsheet. Spreadsheet creators are rarely graphic artists and the quality of the spreadsheet output is limited.

3. development or identification of a canonical form that is robust and does not lose data. Varying data formats and sizes should be easily accommodated. There have been several interchange formats available with varying levels of support in many systems but they have not been consistently updated or used as current standards. The StarOffice/OpenOffice system from SUN Microsystems spreadsheet processor uses `XML` as its standard storage format. This system is available for all common computer systems and is used as the conversion engine for spreadsheet input.

4. development of a program to convert this canonical form into a current formatter with acceptable quality of output. The output form was selected at the start of this project and is TEX/LATEX. This was chosen because it is an open system and a markup system. This can be easily modified for other document preparation systems.

CHAPTER II

LITERATURE REVIEW

Significant research has been done on tabular presentations, because of the importance of tables in information science. This chapter is a review of the more recent research works and spreadsheet systems available today.

The tool introduced in this research is in a domain that is not fully explored as an independent research area. Various spreadsheet editing tools exists currently, with varying degrees of support for exporting into a format which supports high quality presentations in a class like $\LaTeX$. A niché that is void is the ability to make formatting decisions during a conversion process. Research on the tabular data presentation and a brief summary of various spreadsheet editing tools follows.

A. Cognitive Research

For general data presentation, there is not a general consensus amongst various researchers on the efficacy of tables and other mediums (such as graphics or textual paragraphs) as information conveyors [4]. For example, while some [5, 6] report the superiority of the tabular format while others [7] report mixed results. In a study of tables, trees and formulas for decision analysis [8], the conclusion reached was that tables clearly outperform the competition for supplying the information necessary for quick, accurate decision analysis.

However, the best evidence of the effectiveness of tabular presentation is it's widespread use. The first choice for presentation of a data array is usually a tabular format. From baseball schedules to election results, table have consistently been proven to be the medium of choice for presenting various kinds of data.

Based on experiments [1], properties of table that make them effective tools for conveying information include the alignment of the data, the size of the fonts used. These factors are discussed in Chapter 4.

B. Abstract Research

The logical, structural and presentation attributes are often modeled independently. This is to enable focus on, and flexibility of, one model independently of the other. These models are then coupled together to derive what the table finally looks like. An approach which allows independent manipulation of the different models visually is discussed in [9]. An abstract table is defined as a set of finite set of labeled domains and a mapping to their possible values in [1].

C. Software Tools

This section reviews currently available software with respect to formatting features and ability to export/import spreadsheets from various formats. VisiCalc was the first spreadsheet software and was created by Daniel Bricklin in 1978 while he was at the Harvard Business School [3]. Since then, a lot of spreadsheets have been developed with varying levels of support for formatting and interoperability. The most common commercially available are Microsoft Excel and Lotus 1-2-3. While in the open source world, the most common are Gnome's Gnumeric and OppenOffice.org's CALC. Since this project is intended to be readily available, the only choices considered as the base format for conversions were open source. Gnumeric Gnome is not available on the Microsoft Windows platform, which is the predominant Operating System in the market today, so the choice was made to go with CALC. CALC is available on all common operating systems and has a well documented and easily accessible Java Application Programming Interface.

CHAPTER III

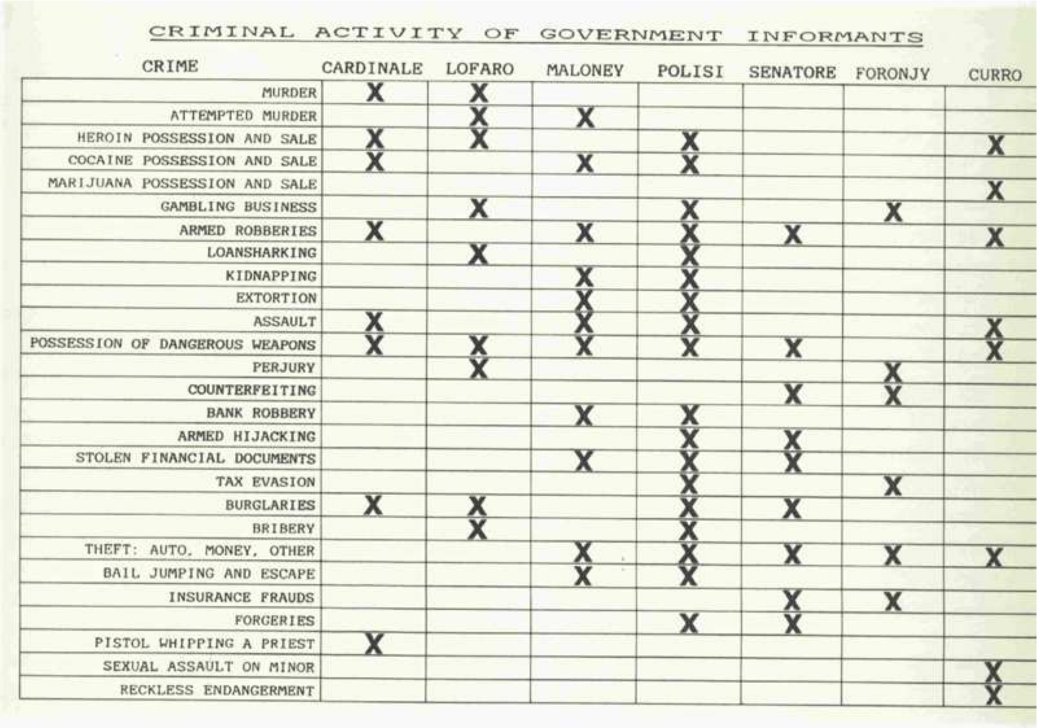
QUALITY PRESENTATIONS

A. Spreadsheets as Information Conveyors

Tables are a good way to show exact numerical values and are preferable to graphics for exact data sets [10]. The primary purpose of a table should be to convey specific information in an effective and efficient way. What is effective might vary based on the end goals and various types of tables have different presentation needs. To illustrate this, a table used by the defense in the case of The United States versus John Gotti *et al* [11]. The defense team made a dent in the prosecution's case by presenting a table which outlined the prior convictions of the government's witness to discredit their testimonies. The table, as shown in Fig. 1 was obviously crafted to convey the crookedness of the government's witness and nothing else. It does well in this regard and for it's intent, it would be a high quality table. For other intents, for example, aggregating the crimes committed by broad categorization which, for the purpose of the prosecution would be an outlining of crimes involving betrayal of trust, the table does a dismal job. The essence of tabular presentation is the ability to

- *convey* the intended information in a clear and concise manner to the reader
- make the reader *want* to read, in other words, not give the reader an excuse not to want to read, by being neatly and attractively formatted [12]
- have the reader put in a *minimal* amount of effort to comprehend
- leave a *lasting* impression on the reader, easily recollected.

A better version of the table presented in Fig. 1 would put the crimes committed by the witnesses more in perspective of betrayal of trust, trust being the essential element needed to justify credibility or otherwise of a witness. This however will not be useful for the defense's purposes.



CRIMINAL ACTIVITY OF GOVERNMENT INFORMANTS

CRIME	CARDINALE	LOFARO	MALONEY	POLISI	SENATORE	FORONJY	CURRO
MURDER	X	X					
ATTEMPTED MURDER		X	X				
HEROIN POSSESSION AND SALE	X	X		X			X
COCAINE POSSESSION AND SALE	X		X	X			
MARIJUANA POSSESSION AND SALE							X
GAMBLING BUSINESS		X		X		X	
ARMED ROBBERIES	X		X	X	X		X
LOANSHARKING		X		X			
KIDNAPPING			X	X			
EXTORTION			X	X			
ASSAULT	X		X	X			X
POSSESSION OF DANGEROUS WEAPONS	X	X	X	X	X		X
PERJURY		X				X	
COUNTERFEITING					X	X	
BANK ROBBERY			X	X			
ARMED HIJACKING				X	X		
STOLEN FINANCIAL DOCUMENTS			X	X	X		
TAX EVASION				X		X	
BURGLARIES	X	X		X	X		
BRIBERY		X		X			
THEFT: AUTO, MONEY, OTHER			X	X	X	X	X
BAIL JUMPING AND ESCAPE			X	X			
INSURANCE FRAUDS					X	X	
FORGERIES				X	X		
PISTOL WHIPPING A PRIEST	X						
SEXUAL ASSAULT ON MINOR							X
RECKLESS ENDANGERMENT							X

Fig. 1. Defense's Case on Government Witnesses in USA vs John Gotti

B. Summary

An important point to plan for when designing a good table is the presentation of the table itself. The formatting choices made in the course of preparing a tabular presentation are as important as the data being presented. This is the focus of this thesis work. We address questions such as

- what are the important things we need to look out for when formatting a table for presentation purposes?
- is it possible to model such attributes with a minimal property sets?
- can these property sets be openly accessible for conversion to and from standard spreadsheet software?
- can we provide a converter that converts exported spreadsheets to a format specific to a quality document preparation system (LaTeX).

CHAPTER IV

TABULAR STYLE STANDARDS AND ISSUES

The best practices for formatting data tables will be drawn from previous research work performed in this area and enumerated upon. A summary of the rules for formatting data tables that have been developed by psychologists, statistics and business information experts is given in [1]. The basic goal of these rules is to make the underlying information being communicated in a table be obvious with minimal instructions.

A. Types of Tables

Broadly speaking, there are two types of tables, formal and informal tables [13].

1. Informal Tables

An Informal Table is generally a continuation of a paragraph of text and does not necessarily have titles or captions. It is usually a short paragraph that is formatted on tab stops and is useful in presenting pieces of information which would be better conveyed if it is separated from the main paragraph text.

2. Formal Tables

A Formal Table has the following properties

1. A title to formally state the information the table is meant to convey
2. Horizontal and vertical rules as appropriate.

As illustrated by Table I, while an informal table is more of an continuation of a paragraph and therefore does not need all the elements a formal table would need which include titles, captions and footnotes.

Table I. Formal and Informal Tables, Formal Table Version

Table Type	One Line Description
Formal Table	Formal Elements: Title, caption, rules, longer.
Informal Tables	Formal Elements optional.

Description based on various style manuals.

B. Graphical Excellence

Graphical excellence consists of complex ideas communicated with clarity, precision and efficiency [10]. What is sought is the presentation of tables to communicate the information desired with minimal effort on the reader's side.

C. Best Practice Presentation Objectives

Detailed descriptions of formatting suggestions for tables are discussed in [1, 13, 14, 15]. These descriptions are focused on here, and are aggregated from style manuals and research works. Note also that these specifications are based on majority of audiences and special adjustments might be required for special classes of people, for example, the visually impaired.

To be effective, font sizes should range between 8 and 12 point, 8 being the minimal that will ensure readability and anything greater than 12 points places a greater strain on the reader in terms on putting the items in the table together coherently in a minimal sweep across the table.

Typographic cues like typefaces (**bold**, *italics*, etc.) should be used to distinguish information that is intended to be highlighted and should be used sparingly to avoid creating a "ransom note" [12]. A table should have at least two columns to necessitate presenting the data as a table and items should be separated and grouped using spaces or rules.

Different items have different alignment requirements. Text is usually recommended to be aligned left. Numbers should be horizontally aligned on decimal points if they contain decimals or right aligned otherwise. They could also be aligned on other characters (e.g. `%, =, <, >,` and other symbols and mathematical operators).

CHAPTER V

SOLUTION

This chapter describes the solution proposed and developed in this thesis work.

A. The System

The system proposed and developed was a system that will allow the user to construct a high-quality presentation in the following manner

1. The source is originally done through the use of a spreadsheet.
2. The source is imported into OpenOffice.org spreadsheet software.
3. The contents are then scanned and imported to a canonical form and stored in XML. This canonical form is completely independent of the source spreadsheet software and any spreadsheet software can be imported into this form. Once in this form, this allows for the other tasks from this point onwards to be done without the need for the origin spreadsheet. Also, other converters can be utilized to convert to this form from any other spreadsheet.
4. The formatting decisions reflected in the canonical form of the spreadsheet can be replaced or edited. These formatting decisions include
    - aids for adjustment of column widths, row heights, and other characteristics can be easily adjusted or input
    - aids for the creation of column and row headings and handling of multicolumn headings
    - aids for included for inserting vertical and horizontal rules of varying thickness to aid readability of the resulting output.

B. The Abstract Table

The table modelled is based on the properties that were determined to be of importance to increasing the presentation quality of a table. The attributes, which work like on and off buttons are set on the table, column, row and cell levels. 0 means the property is off and a number greater than 0 means the property is on. For some attributes, there are varying degrees of *"on"*ness. An example is a vertical rule after a row. 0 means off and a value greater than 0 specifies the thickness of the rule. The units of measurements are parametrizable.

1. Table Level Properties

The properties modeled on the table level are listed below:

***numRows*:** Number, specifies the number of rows in the table

***numCols*:** Number, specifies the number of columns in the table.

***caption*:** String, specifies the caption for the table

***footnote*:** String, specifies the footnote to be placed at the bottom of the table

***label*:** String, LaTeX specific property. Specifies the reference to be used in the exported source for easy cross referencing

2. Column Level Properties

The properties modeled on the column level are listed below:

***IsHeader*:** Number, specifies if this column is a header column. This is useful in making some formatting decisions. An example is a column that has been specified to be aligned on the decimal point. The header column in this case should not be aligned with the other cells in that column.

***IsBold*:** Number, specifies if this column is formatted bold typeface.

***DefDataType*:** Number, specifies the default data type for this column.This data type is applied to any cell in the column which does not have a recognized data type assigned to it.

***SepBefore*:** String, specifies the separation character to be put before this column. This is usually a vertical bar used to draw a vertical rule across the table.

***SepAfter*:** String, specifies the separation character to be placed after this column. Like the sepb4, this is usually a vertical bar.

***column*:** Number, the number of the column to which this set of attributes would be applied.

***width*:** Number, the width of this column.

***vrulebefore*:** Number, specifies if there is a vertical rule before this column. A number greater than zero specifies the thickness of the rule. Can be used in conjunction with the Sepb4 attribute to generate various column separation patterns.

***vruleafter*:** Number, specifies if there is a vertical rule after this column. A number greater than zero specifies the thickness of the rule. Can be used in conjunction with the SepAfter attribute to generate various column separation patterns.

***alignment*:** Number, specifies how this column is aligned, left, right or center.

***is_italic*:** Number, specifies if this column is italicized

***decalign*:** Number, specifies if this column is aligned on decimal points, for number columns.

3. Row Level Properties

The properties modelled at the row level as listed below:

***isHeader*:** Number, specifies if this row is a header row.

***isBold*:** Number, specifies if this Row is to be in bold typeface.

***rowSep*:** Number, separation between t his row and the next.

***row*:** Number, number of the row to which these attributes are applied.

***height*:** Number, height of the row.

***hrulebefore*:** Number, specifies if there is an horizontal rule before this row. A number greater than zero specifies the thickness of the rule.

***hruleafter*:** Number, specifies if there is an horizontal rule after this row. A number greater than zero specifies the thickness of the rule.

***spaceafter*:** Number, specifies the amount of vertical space after this row.

***is_italic*:** Number, specifies if this row is italicized.

Row formats supercedes settings specified at the column level. Thus, if the same attribute is specified differently for a column and a row, the cell at the intersection of the column and the row will have the setting specified at the row level. For instance, if the ***is_italic*** is specified to be 0 for column 1 and specified to be 1 for row 2, The cell at column 1, row 2 would be italicized. Also, cell level settings take ultimate precedence over row and column level settings.

C. Subsystems Description

The subsystems of the application are outlined as follows:

- Conversion Engine
- Canonical Data Storage
- Inference Rules
- Rules Engine

1. Conversion Engine

The conversion from other spreadsheets format to OpenOffice is done by the OpenOffice CALC software. The conversion from the OpenOffice format is achieved by connecting to a factory instance of OpenOffice using the Java Application Programming Interface. The contents of the spreadsheet are extracted passed on to the Data Storage module.

2. Canonical Data Storage

The converted data is stored in XML [16] format using the Document Object Model [17]. XML provides for definition of schemas that specifies the markups we can use for the data store. Each cell of the spreadsheet is stored along with some attributes that can be set at the column and row levels using the appropriate format settings document.

Each cell consists of the following elements

***Data*:** String, contains the data this cell holds row.

***Row*:** Number, specifies the row this cell belongs to.

***Column*:** Number, specifies the column this cell belongs to.

***Italic*:** Number, specifies if this cell is italicized.

***Bold*:** Number, specifies if this cell is typeset in bold typeface.

***ODataType*:** Number, specifies the original data type for this cell. This is kept in case the data type is reset at the row or column level.

***RowSpan*:** Number, specifies the number of rows this cell spans.

***ColSpan*:** Number, specifies the number of columns this cell spans.

***Precision*:** Number, specifies the precision of this data value in this cell if it is of number data type.

***Length*:** Number, specifies the length of this cell.

***SepBefore*:** String, specifies the separation before this cell, same as in the row and column settings.

***SepAfter*:** String, specifies the separation after this cell, same as in the row and column settings.

The choice of XML was driven by the following:

Standards XML has emerged as the standard format for information interchange between diverse systems.

Availability Readily available XML editors can be used to modify row and column attributes. The XML Editor used is the XMLEditPro [18].

Interface Attributes can also be set via a programmatic interface using readily available and free parsers. The parser used in this project is the Oracle XML Parser [19].

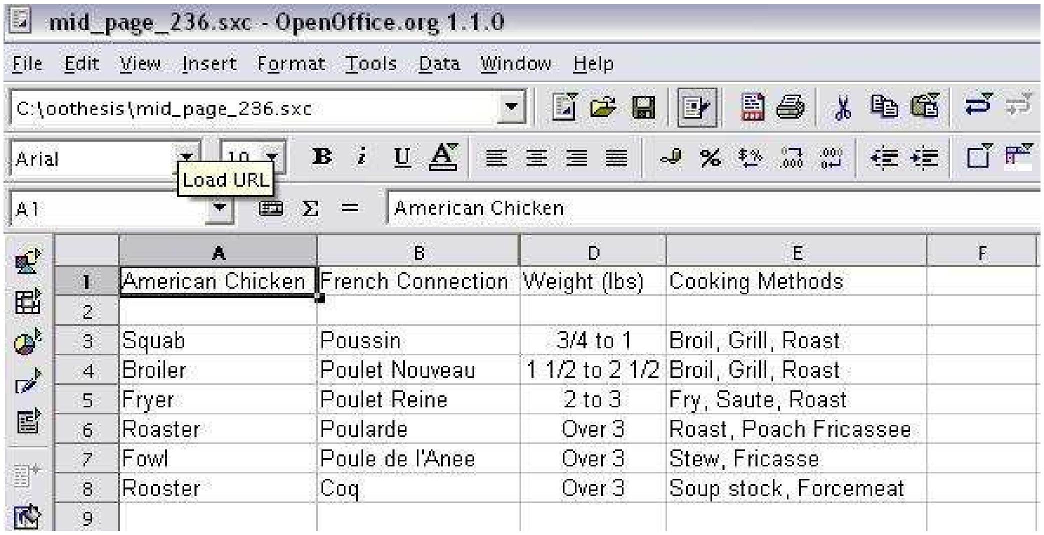
Modularity Allows for a modular design since the TEX conversion routine need not cater for different input data formats.

3. Inference Rules

Stored in relational table form in a database. The inference rules are based on the formatting attributes which have been specified on the row and column levels.

4. Rules Engine

This subsystem takes the data stored in the canonical format and, based on a chosen format specification, converts the input data into a properly formatted table in LATEX source. The format specification contains the rules for formatting a data table to suit that type of formatting. An example of a format specification would be *professional*. The header in a *professionally* formatted data table might be centered and bold. The main purpose of the rules engine is to apply flexible, well defined and data formatting rules, which have been proven to be the main determinants of high-quality


mid_page_236.sxc - OpenOffice.org 1.1.0
File Edit View Insert Format Tools Data Window Help
C:\oothesis\mid_page_236.sxc
Arial
10
Load URL
A1

American Chicken

	A	B	D	E	F
1	American Chicken	French Connection	Weight (lbs)	Cooking Methods	
2					
3	Squab	Poussin	3/4 to 1	Broil, Grill, Roast	
4	Broiler	Poulet Nouveau	1 1/2 to 2 1/2	Broil, Grill, Roast	
5	Fryer	Poulet Reine	2 to 3	Fry, Saute, Roast	
6	Roaster	Poularde	Over 3	Roast, Poach Fricassee	
7	Fowl	Poule de l'Anee	Over 3	Stew, Fricasse	
8	Rooster	Coq	Over 3	Soup stock, Forcemeat	
9					

Fig. 2. Spreadsheet in OpenOffice

presentations, to produce a well-formatted and readable data table that is suitable for the intended audience.

D. Sample Transformations

To illustrate the formatting options available with this system, a sample table imported from OpenOffice is shown in Table II. This table is taken from [20] which cites Beck, Berthole, and Child, *Mastering the Art of French Cooking* (New York: Knopf, 1961) and is slightly modified to suit our purposes. The original spreadsheet document is as shown in Fig. 2.

The formatting process is straightforward. We simply specify a style we would like to apply to the table by picking from pre-defined style templates using the format settings defined earlier and shown in Table III.

By applying format settings 1, 2 and 3, we obtain the results specified in Table IV.

Applying format settings 4 and 5, we obtain the results specified in Table V.

As these examples show, it is quite easy to set the template formats for fairly

Table II. Imported Table with Default Formatting

American Chicken	French Connection	Weight (lbs)	Cooking Methods
Squab	Poussin	3/4 to 1	Broil, Grill, Roast
Broiler	Poulet Nouveau	1 1/2 to 2 1/2	Broil, Grill, Roast
Fryer	Poulet Reine	2 to 3	Fry, Saute, Roast
Roaster	Poularde	Over 3	Roast, Poach Fricassee
Fowl	Poule de l'Anee	Over 3	Stew, Fricasse
Rooster	Coq	Over 3	Soup stock, Forcemeat

Table III. Format Settings for Sample Table

No.	**Desired Format Setting**	**What to Set**	**Where Set**
1.	First row italicized	*IsBold* set to 1	Row
2.	First column bold and left aligned	*IsBold* set to 1 and alignment property to LEFT_ALIGN	Column
3.	Second column italicized	*Is_italic* set to 1	Row
4.	Fractions in third column should be appropriately formatted	*DataType* set to Number	Column
5.	Footnote to acknowledge source	*Footnote* set to The TEXbook	Global

Table IV. Sample Table with Format Settings Applied—1

American Chicken	*French Connection*	*Weight (lbs)*	*Cooking Methods*
Squab	*Poussin*	3/4 to 1	Broil, Grill, Roast
Broiler	*Poulet Nouveau*	1 1/2 to 2 1/2	Broil, Grill, Roast
Fryer	*Poulet Reine*	2 to 3	Fry, Saute, Roast
Roaster	*Poularde*	Over 3	Roast, Poach Fricassee
Fowl	*Poule de l'Anee*	Over 3	Stew, Fricasse
Rooster	*Coq*	Over 3	Soup stock, Forcemeat

Table V. Sample Table with Format Settings Applied—2

American Chicken	*French Connection*	*Weight (lbs)*	*Cooking Methods*
Squab	*Poussin*	$\frac{3}{4}$ to 1	Broil, Grill, Roast
Broiler	*Poulet Nouveau*	$1\frac{1}{2}$ to $2\frac{1}{2}$	Broil, Grill, Roast
Fryer	*Poulet Reine*	2 to 3	Fry, Saute, Roast
Roaster	*Poularde*	Over 3	Roast, Poach Fricassee
Fowl	*Poule de l'Anee*	Over 3	Stew, Fricasse
Rooster	*Coq*	Over 3	Soup stock, Forcemeat

Source: The TEXbook.

sized tables. The transformations from the original table in Table II to the final setup in Table V involve only a simple change in the values in the format settings file. These changes could also been have been made programmatically since the storage is in open format.

CHAPTER VI

SOLUTION SUMMARY AND FUTURE WORK

A. Solution Summary

The system introduced in this thesis work is a non-interactive system that takes a spreadsheet prepared using the OpenOffice CALC Spreadsheet and connects to an instance of the Openoffice Application Programming Interface [21] to convert the spreadsheet into a canonical XML format. The system then takes a style sheet as input, and applies it to the XML file and produces a LaTeX rendition of the spreadsheet. An example of an application of this system would be a summary presentation based on a table whose source is updated daily. A standard style sheet can be created for this document and when the data changes, the new spreadsheet is exported with the standard style sheet specified. This produces a consistent, high quality TeX output in each run.

The main focus of this work was to investigate the presentational capabilities of current spreadsheet software with regards to the outcome of previous research on the attributes of high quality tabular presentations.

The formatting attributes that contribute to high quality presentation were modeled on the cell, row, column and global levels. These models were defined in XML schemas. XML was chosen because of it's extensibility and open format. Also, the tool is developed in Java and thus would be able to run on most operating systems given the installation of the Java Virtual Machine.

The solution is designed to be programmable and easily extensible and is well suited for exporting and formatting fairly sized tables to the LaTeX format.

B. Future Work

There is opportunity for improvement in the area of spreadsheet formatting and portability. Most of the spreadsheets available in the market today have restrictions when it comes to portability among themselves. Most will import spreadsheets from other formats, for instance, Microsoft Excel will import most spreadsheets available in the market today. The file format for the Microsoft Excel documentation by Microsoft Corporation itself is not openly available. The closest to this is published by the OpenOffice group. Because of these types of restrictions, an open form of spreadsheet storage would be a valuable step towards easier integration of spreadsheet software output.

The system developed is a step in this direction. There are other advancements that could be made with this tool within the larger goal of easier conversion from different spreadsheet formats and export into a format that encourages and enables quality presentations.

1. Modifications to The System

Slight modifications could be made to the system in forms of integration and extensions. This will, with minimal effort, allow the system to provide additional and convenient features. Each of these possibilities are described below.

Integration The tool can be integrated with other spreadsheet software to ease formatting of documents that are produced regularly. For example, an hyperlink can be provided that takes the current spreadsheet being worked on and converts it using a specified style sheet to TEX format.

Extension Add-ons can be supplied for the system to extend its functionality. The data used by each subsystem is easily accessible and can be manipulated using other software. An utility can be added that manipulates either the style sheet or the converted data based on other decision trees.

2. Directions

The System itself could be modified to provide more features and also make it more user friendly. While these features would involve some effort, they may be easily added on to the system because of the well defined, extensible architecture of the application. The suggested modifications are described below.

An Interactive System The current version is non-interactive, in that the formatting properties to be specified are edited directly in the format settings XML file. However, this sets a bar on the expertise required for using the system to be higher than an entry level computer user. An interactive system that establishes a dialogue with the user and stores the settings based on the user's response would lower the required operating expertise.

Large Tables The system could be extended to handle larger tables. Handling tables that span more than one page wide or tall are not as straightforward in LaTeX and the system will require some more functionality to be able handle such tables.

More Flexible Alignments One of the formatting attributes that makes a good presentation is the alignment of the numeric columns. Currently, the system allows for alignment on decimal points by specifying the *decalign* property on the column level. This could be extended to allow for alignment on any character which might include `%, =, <, >,` and other symbols and mathematical operators). Paragraph cells, which need special alignment can also be provided.

It is appropriate to end this thesis with a quote from Edward Tufte [10], a legend in the field of visual information presentation:

> What is to be sought in designs for the display of information is the clear portrayal of complexity. Not the complication of the simple; rather the task of the designer is to give visual access to the subtle and the difficult that is, the revelation of the complex.

REFERENCES

[1] Xinxin Wang. *Tabular abstraction, editing, and formatting.* PhD Dissertation, University of Waterloo, Ontario Canada, 1996.

[2] Houghton Mifflin Company. *The American Heritage College Dictionary, Third Edition.* Massachusetts: Houghton Mifflin Company, 1997.

[3] D.J. Power. A brief history of spreadsheets [online]. 2003. Version 3.5 http://www.dssresources.com/history/sshistory.html.

[4] Richard A. Coll, Joan H. Coll, and Ganesh Thakur. Graphs and tables: a four-factor experiment. *Communications of the ACM*, 37(4):76–86, 1994.

[5] J.A. Ghani. *The effects of information representation and modification of decision performance.* PhD Dissertation, University of Pennsylvania, Philadelphia, 1981.

[6] Wainer H. and Raiser M. Assessing the efficacy of visual displays. in *Proceedings of the American Statistical Association*, pages 89–92. ASA, 1976.

[7] I.H Nawrocki. Alphanumeric versus graphical displays in a problem solving task. Virginia: U.S.Army Behaviour and Systems Research Lab., 1972.

[8] Shailendra C. Palvia and Steven R. Gordon. Tables, trees and formulas in decision analysis. *Communications of the ACM*, 35(10):104–113, 1992.

[9] Horst Silberhorn. Tabulamagica: an integrated approach to manage complex tables. in *Proceedings of the 2001 ACM Symposium on Document engineering*, pages 68–75. ACM Press, 2001.

[10] Edward R. Tufte. *The Visual Display of Quantitative Information.* Connecticut: Graphics Press, 1983.

[11] United States Court Of Appeals. United states of america vs. john a. gotti et al [online]. 2003.
http://www.ipsn.org/US_v_Gotti.htm.

[12] Bart Childs. Papers, Abstracts, and Résumés [online]. 2003. http://courses.cs.tamu.edu/bart/cpsc481/Talks/docs_talk.pdf.

[13] The New York Public Library. *The New York Public Library Writer's Guide To Style and Usage—First Edition*. New York: Stonesong Press, 1983.

[14] The University of Chicago Press. *The Chicago Manual of Style–14th Edition*. Illinois: The University of Chicago Press, 1993.

[15] Franklin Covey. *Style Guide For Business and Technical Communication—Third Edition*. Utah: Franklin Covey Co., 1997.

[16] World Wide Web Consortium. Extensible Markup Language (XML) [online]. 2003. http://www.w3.org/XML.

[17] World Wide Web Consortium. Document Object Model (DOM) [online]. 2003. http://www.w3.org/DOM.

[18] David Levinson. XMLEdit Pro. [online]. 2003. http://www.daveswebsite.com.

[19] Oracle Corporation. Oracle XML Parser for Java [online]. 2003. http://otn.oracle.com/tech/xml/index.html.

[20] Donald E. Knuth. *The TEXbook*. Massachusetts: Addison Wesley Publishing Company, 1990.

[21] Sun Microsystems Inc. Openoffice Documentation [online]. 2003. http://www.openoffice.org/documentation.html.

APPENDIX A

CONSTANTS DEFINITIONS

```
//ROWS
public static final int HEADER_ROW = 1;
public static final int NORMAL_ROW = 2;


//COLUMN PROPERTIES
public static final int LEADER_COLUMN = 1;
public static final int STRING = 2;
public static final int INT = 3;
public static final int FLOAT_MAIN = 4;
public static final int FLOAT_FRACTION = 5;
public static final int DOUBLE= 6;
public static final int PARA = 7;
public static final int NULL = 8;//Paragraphs


//ALIGNMENTS
public static final int LEFT_ALLIGN = 1;
public static final int RIGHT_ALLIGN = 2;
public static final int CENTER_ALLIGN = 3;


//BOLD/NO_BOLD
public static final int IS_BOLD = 1;
public static final int NO_BOLD = 0;


//ITALIC/NO_ITALIC
public static final int ITALIC = 1;
public static final int NO_ITALIC = 0;


//LAYOUT
public static final int LAYOUT_LANDSCAPE = 1;
public static final int LAYOUT_POTRAIT = 2;
```

APPENDIX B

CELL DEFINITION

```
public class Cell{


int Italic ;
int Bold ;
int Allignment;
int ODataType;//original data type that was determined
              // during initial read..
int DataType;//data type overriden...
int ColSpan ;
int RowSpan ;
int Precision;
int Length;
String Sepb4; //seperator that comes before data
String SepAfter;//seperator that comes after data
String Data;


public Cell (){
Italic = utils.NO_ITALIC;
Bold = utils.NO_BOLD;
Allignment = utils.LEFT_ALLIGN;
ColSpan = 1;
RowSpan = 1;
Data = "";
DataType = utils.STRING;
Length = 5;
Sepb4 ="";
SepAfter = "";
DataType = utils.STRING;
Precision = 0;


}
```

VITA

Wasiu Olajide received his Bachelor of Science in computer science from The University of Ibadan, Ibadan, Nigeria in January 1999. He joined the graduate program at The Texas A&M University, in January 2002. His research has been in document formatting and database systems. He may be contacted through the Department of Computer Science, Texas A&M University, College Station, TX 77843-3112.

The typist for this thesis was Wasiu Olajide.