Journal of Information Systems, Vol. 19, No. 1, Spring 2005, pp. 19-41

Financial Reporting and Auditing Agent with Net Knowledge (FRAANK) and eXtensible Business Reporting Language (XBRL)[1]

Matthew Bovee <bovee@bsad.uvm.edu>
The School of Business Administration
The University of Vermont - Burlington, VT 05405

Alexander Kogan <kogan@rutgers.edu>
Department of Accounting and Information Systems
Rutgers Business School – Newark and New Brunswick
Rutgers University
180 University Avenue, Newark, NJ 07102-1895

Rajendra P. Srivastava <rsrivastava@ku.edu>
School of Business, The University of Kansas
Lawrence, KS 66045

Miklos A. Vasarhelyi <miklosv@rutgers.edu>
Rutgers Business School – Newark and New Brunswick
Rutgers University
Faculty of Management, Rutgers University
180 University Avenue, Newark, NJ 07102-1895

Kay Nelson <nelson_k@cob.osu.edu>
Fisher College of Business, The Ohio State University
2100 Neil Avenue, Columbus, OH 43210

November 2004

[1] The authors express their gratitude to the Editor, Associate Editor, anonymous referees, Eric Cohen and Liv Watson whose valuable comments helped us to improve the paper. The authors also acknowledge significant help provided by many research assistants at the Ernst & Young Center for Auditing Research and Advanced Technology (E&Y CARAT) at the University of Kansas. This research was partially supported by the E&Y CARAT.

Financial Reporting and Auditing Agent with Net Knowledge (FRAANK) and eXtensible Business Reporting Language (XBRL)

ABSTRACT

This paper describes the development and applications of FRAANK – Financial Reporting and Auditing Agent with Net Knowledge. The prototype of FRAANK presented here provides automated access to, and understanding and integration of rapidly changing financial information available from various sources on the Internet. In particular, FRAANK implements intelligent parsing to extract accounting numbers from natural-text financial statements available from the SEC EDGAR repository. FRAANK develops an "understanding" of the accounting numbers by means of matching the line-item labels to synonyms of tags in an XBRL taxonomy. As a result, FRAANK converts the consolidated balance sheet, income statement, and statement of cash flows into XBRL-tagged format. Based on FRAANK, we propose an empirical approach towards the evaluation and improvement of XBRL taxonomies and for identifying and justifying needs for specialized taxonomies by assessing a taxonomy fit to the historical data, i.e., the quarterly and annual EDGAR filings. Using a test set of 10-K SEC filings, we evaluate FRAANK's performance by estimating its success rate in extracting and tagging the line items using the year 2000 C&I XBRL Taxonomy, Version 1. The evaluation results show that FRAANK is an advanced research prototype that can be useful in various practical applications. FRAANK also integrates the accounting numbers with other financial information publicly available on the Internet, such as timely stock quotes and analysts' forecasts of earnings, and calculates important financial ratios and other financial-analysis indicators.

Key words: Semantic Pattern Matching, SEC EDGAR, XBRL Taxonomy

I. Introduction

The accounting world has changed dramatically over recent years and the pace of change is accelerating. Major accounting firms have increasingly been using technology to obtain labor savings. They are progressively developing large knowledge-management services to support their professional activities. Input for such knowledge-management services comes from many valuable data sources that are publicly available on the Internet. The most notable one is the Securities and Exchange Commission's EDGAR (Electronic Data Gathering, Analysis, and Retrieval) Internet repository containing corporate filings with the SEC. Many Internet portals (e.g., Yahoo, Quicken) provide public access to financial markets data, analysts' forecasts, news feeds of business relevance, etc.

Numerous professionals depend critically on timely access to financial information. Complexity, cost, and latency of obtaining financial information in computer-understandable format create significant friction in the system. Although the SEC filings are available from EDGAR in a computer-readable format, they are difficult to process automatically because of great variations in the filing structure and terminology, which impede the automatic understanding of semantics of plain-text documents. A very promising approach to solving this problem is based on the development and wide acceptance of XML (eXtensible Markup Language). XBRL (eXtensible Business Reporting Language)[1] is a recent XML-derivative language aimed at enhancing and facilitating the business reporting and analysis process at the account level. Several other XML-derivative languages (such as ebXML), proposed in different industries or by associations, focus on the transaction level. While the development of XBRL shows great promise, there are significant barriers to wide acceptance of XBRL, including the costs associated with adopting XBRL and the absence of the SEC requirement to file financial statements in XBRL. Another

major impediment to widespread XBRL adoption is the difficulty of agreeing on common taxonomy standards. Before XBRL becomes the accepted standard for filing financial statements, intelligent software tools must be developed and utilized to automatically process the semi-structured natural-text financial documents that are publicly available on the Internet. Manual extraction of accounting numbers from the text of accounting statements available from EDGAR and manual conversion to the XBRL standard would be too cumbersome and expensive.

The objective of this paper is to describe the development, evaluation, and applications of FRAANK – Financial Reporting and Auditing Agent with Net Knowledge. The prototype of FRAANK presented here provides automated access to, and understanding and integration of rapidly changing financial information available from various sources on the Internet. In particular, FRAANK implements intelligent parsing to extract and understand accounting numbers from natural-text financial statements available from EDGAR, and converts the consolidated balance sheet, income statement, and statement of cash flows into XBRL-tagged format. It integrates the accounting numbers with other financial information publicly available on the Internet, such as timely stock quotes and analysts' forecasts of earnings, and calculates important financial ratios and other financial analysis indicators. FRAANK can potentially be utilized in an auditing practice, or used by investors and creditors in decision making.

More specifically, FRAANK is focused on the task of finding important information (e.g., accounting numbers) in semi-structured natural-text documents of accounting or auditing nature. The pioneering research in developing artificial-intelligence capabilities for analyzing financial statements and understanding accounting texts (Tanaka 1982) is described by Mui and McCarthy (1987) and O'Leary et al. (1988, 1991, 1992). The most comprehensive previously

reported implementation of the intelligent parsing of financial statements in the SEC EDGAR filings has been achieved in EdgarScan (Ferguson 1997; Steier 1995; Steier et al. 1997). A generic tool for semantic parsing of the SEC filings has been developed by Gerdes (2003). The predecessor of FRAANK, the so-called original Edgar agent (Nelson et al. 2000), was capable of retrieving quarterly filings (10-Q's) from the SEC EDGAR repository and parsing them to identify the most important accounting numbers. As compared with the new Edgar subagent of FRAANK, the original Edgar agent was much more limited in its capabilities and had a much simpler design. The original Edgar agent could process only the quarterly SEC filings and identify only a few of the most important accounting numbers. Its design was specially tuned for this limited task, and could not be scaled up to parse financial statements comprehensively. It interacted with a single online information source (the SEC EDGAR repository) only.

EdgarScan was created by PWC and made publicly available on the Internet in response to the need for automatic extraction of accounting numbers from the EDGAR filings.[2] Although EdgarScan is technologically very different from the original Edgar agent (it uses C and Prolog instead of Perl), its architecture and design have exactly the same limitations as those of the original Edgar agent. EdgarScan is designed to parse a predetermined limited set of accounting numbers, and therefore cannot be easily scaled up to understand the meaning of the financial statement labels that correspond to the lower levels of an XBRL taxonomy. Making EdgarScan (and other similar parsers) understand all the line items in the financial statements would require a total redesign of the architecture, similar to the effort reported in this paper, which involved transforming the original Edgar agent into the new version of FRAANK.

The new version of FRAANK is capable of parsing both the annual (10-K) and quarterly (10-Q) SEC filings and is designed to "understand" all the line items in the consolidated finan-

cial statements by matching them to the most appropriate tags in an XBRL taxonomy. The architecture of FRAANK facilitates learning over time since it makes it possible to improve FRAANK's accuracy by adding synonyms to the knowledge base whenever an exception is logged during execution. FRAANK's use of an XBRL taxonomy as the anchor for matching the line items makes it a natural tool for empirically evaluating various XBRL taxonomies, identifying their deficiencies for improvement, and justifying the need for additional industry taxonomies.[3] The utilization of FRAANK as a converter of regular SEC filings to XBRL can facilitate automation of various financial analyses that previously had to rely on human intervention. Such uses are further enhanced by FRAANK's capabilities of extracting additional financial information online, such as stock quotes and analysts' forecasts.

This paper is organized as follows. In the next section we outline the importance and various issues related to XBRL as the emerging business-reporting language and the need for intelligent agents to facilitate the adoption of XBRL. Then, in Section III we define the goals of FRAANK, and describe its function and architecture. In Section IV we give some technical details about the implementation and intelligent parsing in FRAANK. In Section V we evaluate the performance of FRAANK. We devote Section VI to describing various applications of FRAANK, and Section VII to discussing some of the key future developments in the FRAANK project and concluding remarks.

II. Background

XBRL has come a long way since the pioneering development of its predecessor XFMRL (eXtensible Financial Reporting Markup Language) in 1999. The XBRL Consortium, which includes numerous interested parties such as CPA firms, business-software vendors, and financial-

services companies, has been continuously working on developing and improving various components of this emerging standard. The main focus of XBRL has been on developing means for unambiguously identifying financially relevant information. As an XML-based standard, XBRL is designed to provide a set of textual tags for marking various parts of a document to identify accounting numbers relevant for external financial reporting. Adding such markup makes it possible to automate the processing of financial reports and facilitates the exchange of financial information between various computer systems.

To accommodate significantly diverging financial-reporting practices among different countries and various industries, XBRL uses a two-tiered design consisting of the XBRL specification and XBRL taxonomies. The XBRL specification provides a set of generic tags for expressing various facts, and as such, the XBRL specification is not tied to any particular set of accounting standards or practices, or even to accounting in general. To provide concrete tags for marking an actual financial report, the XBRL specification has to be complemented with an XBRL taxonomy. It is an XBRL taxonomy that defines the markup of various accounting concepts as well as the hierarchical and numerical relationships among these concepts. Therefore, while a single XBRL specification suffices, numerous taxonomies must be developed to accommodate the needs of different countries and/or industries.

Developing XBRL taxonomies is a difficult, laborious, and controversial process. A typical taxonomy has to define markup for several hundred concepts, and it has to be done in accordance with generally accepted accounting standards and practices. The standards are usually not formal enough and the practices are not uniform enough to allow for an unambiguous translation of them into taxonomies. On the one hand, it is important to develop a sufficient number of taxonomies to accommodate the true needs of all the constituents. On the other hand, it is important

to avoid the proliferation of different taxonomies to the extent possible, since the use of different taxonomies impedes the comparability of financial reports. It is therefore important to develop a methodology for evaluating proposed XBRL taxonomies, developing ways of improving them, and identifying and justifying needs for additional taxonomies.

An empirical approach towards the evaluation and improvement of XBRL taxonomies is to assess a taxonomy fit to the historical data, i.e., the quarterly and annual filings with the SEC that are available from the EDGAR repository. The FRAANK agent can be used to automate the evaluation and improvement of an XBRL taxonomy, which is utilized in FRAANK as the anchor in intelligent parsing of the natural text of financial statements. Most importantly, FRAANK facilitates the adoption of XBRL by providing the automatic conversion of financial statements to this emerging standard. Thus, FRAANK demonstrates the crucial importance of the intelligent-agent technology for the successful development and adoption of XBRL.

While software agents have been the focus of much attention both in popular press and in the research community, there is no commonly accepted definition of an agent.[4] Past experience with various artificial-intelligence applications has shown that to be successful in practice an application usually has to follow the "weak-AI" paradigm, i.e., to simulate intelligent behavior, and be sufficiently focused on a limited set of tasks.[5] This is the approach taken in FRAANK. The main subagent of FRAANK – Edgar – faces a difficult task of understanding very loosely formatted financial statements in the SEC filings, and therefore utilizes a tremendous amount of code to achieve performance worthy of the weak-AI qualification. On the other hand, the ticker subagent's task is much simpler – to choose the most likely ticker symbol for a given string representing a company name. While, strictly speaking, this task is ill defined, the performance of the ticker subagent is often comparable to that of a human being given the same task, and there-

fore quite useful even though it implements only a few simple heuristic rules (see the next section), and is an extreme example of the weak-AI approach.

III. Architecture of FRAANK

The current version of the Financial Reporting and Auditing Agent with Net Knowledge is capable of:

- analyzing semi-structured natural text in the financial domain,
- formally representing and using accounting knowledge, and
- interacting with a variety of online information sources.

The FRAANK prototype[6] can be viewed as the first step in gathering financial data on companies that are available on the Internet and using these data to provide value-added service. Once the agent retrieves financial information, these data can be processed and combined with other artificial-intelligence systems to provide knowledge for enhanced decision making in real time.

FRAANK communicates with its users over the World Wide Web through a simple interface. The user who is interested in a certain publicly traded company submits that company's name to the agent, and then FRAANK automatically:

- searches the SEC EDGAR database for the filings (10-Q and 10-K) by that company, intelligently parses, and tags using the US C&I XBRL taxonomy (Spec. 2) the filing selected by the user,
- queries the Yahoo ticker-search engine to identify the ticker of the company,
- uses the ticker to retrieve the most recent stock price from Quote.com,
- contacts Quicken.com to find the most current consensus forecast of earnings per share (EPS) provided by Zacks, and

- utilizes the obtained results to calculate various accounting ratios, and the Z-score (a discriminant measure of bankruptcy risk) (Altman 1968, 1983).

The multifaceted nature of FRAANK is reflected in its design. FRAANK consists of several subagents, corresponding to the Internet information sources that FRAANK utilizes in real time. These subagents are the Edgar agent, the ticker agent, the stock-quote agent, and the EPS agent. The architecture of FRAANK, as shown in Figure 1, allows the agent's logic to be clearly separated from the end-user interface on the one hand, and from the accounting knowledge source stored in a relational database on the other hand. Currently, the knowledge source in the database contains synonyms of accounting terms.

----- Insert Figure 1 here ------

Edgar Subagent

Edgar is the main subagent of FRAANK, since it is responsible for obtaining and processing the richest source of information – SEC filings of publicly traded companies available in the SEC EDGAR repository on the Web. The predecessor of the Edgar subagent, the original Edgar agent, was described in detail in Nelson et al. (2000). The current version of the Edgar subagent is completely redesigned, significantly more robust, and has much greater capabilities, which include:

- retrieving the company's recent SEC filings,
- finding and extracting from them consolidated financial statements,
- parsing the statements to identify all the line items, their balances, and their aggregation structure,
- matching the labels of the identified line items with the corresponding XBRL tags and tagging them, and

- identifying line items for which no XBRL tag in the taxonomy exists.

This subagent searches the SEC EDGAR repository for reports (10-Q and 10-K) of publicly traded companies that match the user-specified names and returns a list of available filings. These filings are sorted by date, with the most recent ones listed first. The user is prompted to choose a filing, which is then retrieved and analyzed. Since the Edgar subagent always contacts the SEC EDGAR database in real time over the Internet, it always provides access to the most recent filings. While this retrieval function is based on standard Internet technology, the other two functions of Edgar have to do with extremely difficult and highly nonstandard problems of analyzing natural text.

The current version of FRAANK finds in the retrieved 10-Q and 10-K filings the consolidated balance sheet, income statement, and statement of cash flows. Although a human accountant can easily locate a financial statement in the body of a 10-K filing, the same task is a challenge for a computer program because of the great variability in the structure of the filings and the language used. The location of the statements in the body of the filing varies significantly. While the SEC EDGAR filings are required to have some generic SGML tags such as the table and column tags, evidently there is no effective validation procedure in place to enforce compliance with these requirements, and the required tags can be missing or misplaced. While each statement is delimited with the table tags, there are usually many different tables in the body of the filing. Moreover, the caption can be both before and after the table tag, and the wording of the caption varies greatly. For example, the income statement can also be called the statement of income, the statement of changes in financial position, the profit and loss statement, or the statement of revenues and expenses, etc. Complicating the automatic location of statements

even further is the possibility of having the same keywords in the caption of some other tables in the body of the filing (e.g., due to the segment reporting requirements).

After a particular financial statement is extracted, it is parsed by the Edgar subagent to identify all the line items, their balances, and their aggregation structure. Line-item names can exhibit even greater variability than the table captions, as described in detail in Nelson et al. (2000). The intelligent parsing procedure in Edgar utilizes tables of line-item synonyms stored in a relational database (see Section IV for a more detailed description). It used to be possible to cross-validate certain line items with the SGML-tagged values presented in the Financial Data Schedule (FDS) appended to the 10-K and 10-Q filings. Unfortunately, the SEC abandoned the requirement to append the FDS as of early 2001, thus making this cross-validation impossible for newer filings. Therefore, for recent filings, only the aggregation structure (totals, subtotals, etc.) of the line numbers is used to improve the accuracy of parsing.

Once the agent has parsed the SEC filings and identified the line items and the corresponding values for the desired period, it matches the identified labels to the corresponding XBRL tags and then tags the line items accordingly. The current version of the agent uses the 2000 US C&I XBRL Taxonomy, Version 1 (see XBRL 2000). Whenever there are no matching XBRL tags, the agent uses generic tags for those line items.[7] As discussed later, this feature of the agent will be utilized in testing, validating, and improving the US C&I XBRL taxonomies.

Ticker Subagent

After the user selects a 10-K or 10-Q filing for analyses, FRAANK uses its ticker subagent to identify the ticker symbol of the company, which is needed for obtaining the stock quotes and EPS forecasts. There are a number of ticker-search engines publicly available on the

Internet. The ticker agent sends the company name to the Yahoo ticker engine, and retrieves a list of tickers matching the submitted name. The retrieved ticker list must then be analyzed to determine the most probable candidate, since this list will usually contain quite a few tickers. The ticker subagent utilizes several heuristic rules for identifying the best match, applying them in the following order:

- Since most probable matches should include user input as separate words in company names, items not containing all the exact words (e.g., containing super-words, i.e., words that contain the input words as proper substrings) are eliminated.
- Since most probable ticker symbols are short, the list is ordered by ticker-symbol length.
- Since most probable company names are short, items corresponding to the shortest tickers are ordered by the total length of the company name, and the first one is chosen as the best match. (Any tie is broken arbitrarily at this stage.)

Although these rules are quite simple, and thus the ticker subagent intelligence is rudimentary, very often the resulting best match is exactly the right ticker. This subagent illustrates the practicality of simple heuristics: certain seemingly complicated tasks can be accomplished by using quite simple means. Unfortunately, as the example of the Edgar subagent shows, this is not always the case (see Section IV for a description of our intelligent parsing technology).

Stock Quotes and EPS Subagents

The stock-quote subagent uses the ticker symbol identified by the ticker subagent to send the symbol to a public-stock-quote engine (mach.quote.com), and extracts from the response the most recent share price. On its Quicken Web site, Intuit provides free access to Zacks Investment Research data of earnings-per-share (EPS) forecasts. The EPS subagent uses the most probable ticker to retrieve the (diluted) EPS forecast for the next quarter from quicken.com.

Integration and Analysis

FRAANK integrates the subagents' responses (price from the stock-quote subagent, earnings from Edgar, and analyst forecasts) to compute historical and estimated important financial ratios showing this company's investment potential (like the price/earnings ratio), and to calculate a number of other financial ratios (e.g., quick ratio, current ratio). Examples of summarizing and integrating the information obtained by the subagents are detailed in Section VI.

IV. Design, Implementation and Intelligent Parsing in FRAANK

Since the most difficult task of FRAANK is the intelligent analysis of natural-text documents (i.e., 10-K and 10-Q filings), FRAANK needs a very strong pattern-matching capability. Therefore, the programming logic of FRAANK is implemented entirely in Perl (Wall et al. 1996) since, arguably, Perl provides one of the strongest pattern-matching functionalities. By far the largest part of FRAANK's code is devoted to the implementation of intelligent parsing of semi-structured natural text of the SEC filings.

Since FRAANK interacts with both its users and its external information sources over the Internet, the choice of Perl provides the benefits of very strong networking support. When FRAANK communicates with external information sources over the Internet (e.g., SEC, Yahoo, Quicken, etc.), FRAANK essentially acts as a specialized Web client. The Web client functionality of FRAANK is implemented using the high-level libwww-perl library (once called LWP, see also Wong 1997), which allows much simpler implementation of more sophisticated Web client functionality in the current version of FRAANK (as compared with the original Edgar agent, see Nelson et al. 2000).

End users also interact with FRAANK using HTTP. In this case, however, the user's Web client contacts FRAANK's Web server, which launches FRAANK using Common Gateway Interface (CGI). FRAANK relies on the CGI.pm module developed by Lincoln D. Stein (1999), which uses objects to create Web fill-out forms on the fly and to parse their contents. When executed by the HTTP server, the agent receives the user's input as CGI URL-encoded variables using the CGI.pm param() method.

The Edgar subagent of FRAANK has grown and transformed greatly since the original implementation (described in Nelson et al. 2000). It has been redesigned and modularized, and now includes almost two-dozen subroutines and about five-thousand lines of code. The major subroutines are designed for:

- extracting the consolidated balance sheet, income statement, and statement of cash flows,
- partitioning each statement into line items, identifying headings, and merging multiple lines corresponding to the same line item,
- splitting each line item into the label and values and identifying the column of values corresponding to the current period (as opposed to the previous ones),
- determining the aggregation structure of the values to identify totals among the line items,
- matching the labels to the synonyms of tags in an XBRL taxonomy and selecting the most appropriate XBRL tag for each line item,
- matching the headings to the identified totals, and selecting the most appropriate XBRL tags for the totals with empty labels.

Note that the first three tasks in the above list become much easier if the filings are presented in HTML (as opposed to plain text with a few SGML tags). Therefore, the Edgar subagent determines whether the filing is or is not in HTML, and chooses the parsing procedures accordingly. The parsing procedures in the Edgar subagent are heuristic algorithms that simulate the

results of information processing by an intelligent reader of the SEC filings. To achieve a high level of accuracy requires implementing a tremendous amount of parsing logic in the code of the Edgar subagent. For example, the extraction of tables of the three consolidated financial statements is based not only on locating the table tags and recognizing the captions corresponding to the synonyms of the statement titles stored in the knowledge base, but also on verifying that the beginning of the analyzed table corresponds to the expected structure. In the case of the balance sheet, the table is expected to start with assets, more precisely the most liquid assets (cash and cash equivalents). The income statement is expected to start with revenues or sales, while the statement of cash flows is expected to start with net income (for the statements prepared using the indirect method) or cash flows from operations. Additional logic is implemented to identify the end of the statement, which can be split into several parts (e.g., when it does not fit on a single page).

A special challenge in parsing the financial statements is the presence of headings and totals. Headings will usually have just the label and no values, while totals will often have just a trivial label (e.g., "total") and meaningful values. A heading may or may not have a corresponding total below in the statement, while a total may or may not have a corresponding heading above in the statement. The identification of totals cannot be based only on matching keywords (such as "total") and on locating (double) underlining since both can be missing. Therefore, the procedure for identifying the totals without meaningful labels is based on identifying the aggregation structure of the accounting numbers in the statement, under the assumption that spurious equalities are highly unlikely. The *totals procedure* is recursive. It starts from the top of the statement, adds up the values of contiguous segments of line items, and checks whether the sum equals the value of the following line item. If yes, then the following line item is identified as a

total and only this item is used instead of the corresponding segment for further identification of the nested totals. The procedure has to restart after each total is identified. Note that the maximum number of contiguous segments is only quadratic in the number of line items in the statement, and therefore the procedure is not very expensive computationally.

The totals procedure also incorporates special logic for identifying negative values, which is quite challenging since many filings do not follow the standard accounting convention of displaying negative values in parentheses, nor do they include any keywords (such as "less") that indicate negative values. Therefore, the procedure attempts to flip the sign of some values in the segment while calculating the total. Note that it would be prohibitively expensive computationally to attempt flipping the sign of every conceivable subset of values in the segment, and one can argue that in general it is computationally intractable to identify an arbitrary subset of negative values.[8] Therefore, it is assumed that the subset of the negative values will form a contiguous subsegment at the end of the considered segment (just before the total), and the procedure attempts to consider all such subsegments, which is computationally inexpensive since the number of such subsegments is at most $n - 1$, where n is the number of values in the segment.

To "understand the meaning" of the line items, the Edgar subagent has to map every line item (if possible) to a tag in an XBRL taxonomy. As illustrated in Figure 2, the Edgar subagent utilizes a knowledge source of accounting synonyms, stored in a relational database, to cope with the variation encountered in corporate use of terminology and subsequently identify and parse the appropriate substitute terms. The knowledge source is based on an XBRL taxonomy, with each taxonomy tag mapped to (possibly many) synonyms encountered in previously analyzed SEC filings. The connection to the database is established over ODBC, and the queries are implemented in standard SQL.

In the current implementation of the Edgar subagent, a unified parsing subroutine is developed that parses all the line items in exactly the same way. The 552 elements of the C&I XBRL taxonomy, for the balance sheet, income statement, and statement of cash flows (see Table 1), are used as the key items in a growing database of (currently) approximately 3200 synonyms. This development is likely to ultimately both improve the accuracy of FRAANK's parsing and to identify possible deficiencies in, or necessary extensions to, the XBRL taxonomy. Indeed, the agent keeps track of the exceptions whenever the parsing subroutine fails, which can happen for two different principal reasons. The first one happens when no synonym in the database matches the line-item description. After human review of the exceptions log, this synonym will be added to the database with its paired corresponding key, thus eliminating this instance of failure in the future. The second reason for a parsing failure is the absence of the corresponding key item in the database of synonyms, which indicates a gap in the underlying XBRL taxonomy. The identification of such instances in the exceptions log should provide helpful information for improving the underlying XBRL taxonomy, or for identifying industries or firms where specific extensions to the core XBRL taxonomy or specialized XBRL taxonomies may be warranted.[9]

----- Insert Table 1 here -----

----- Insert Figure 2 here -----

All the synonyms for the tags corresponding to a particular financial statement are retrieved from the database and compared against parsed labels of the line items to determine all the complete or partial matches. Since the taxonomy contains hundreds of tags with several synonyms each, it is quite common to find multiple synonyms matching a single line-item label. The logic of the heuristics developed to resolve such matching conflicts is quite elaborate. The

preference is given to exact (as opposed to partial, or inexact) matches and to the longest matches among the partial ones. In those cases where multiple-exact (or only multiple equal-length partial) matches are found for a single-line item, it is assumed that the most general of the matched tags is the most likely correct match. Such a tag is found by determining the *closest common predecessor* of all the matched tags in the taxonomy hierarchy. This heuristic rule can fail, and is therefore verified using the position in the taxonomy hierarchy of the already assigned tags of neighboring line items. More specifically, it is verified that the already assigned tags of the neighboring five-line items below are not successors of the chosen closest common predecessor. Otherwise, the alternative matches are examined.

The closest-common-predecessor rule is also used in matching the headings to the totals (if they exist), and assigning tags to the totals without meaningful labels. If a financial statement has a heading and a matching total below, then they should be assigned the same XBRL tag – say <T> – and all the line items in between should be assigned XBRL tags that are successors of <T>. Moreover, no successors of <T> should be assigned to any other line items in the statement. All these rules are implemented in the parsing logic.

The Edgar subagent implements an elaborate logic for exception handling. Whenever the matching of XBRL tags to line-item labels fails, the agent records an exception in the log file for subsequent human analysis (see the next section).

V. Evaluation of FRAANK

In this section, we discuss the performance statistics of FRAANK in parsing and tagging the 10-K documents. While the current version of FRAANK has the capability of parsing and tagging both the 10-K and 10-Q documents, we provide here only the performance statistics for

the 10-Ks. While the basic structure of the 10-Q and the 10-K documents remains the same, the 10-Q documents are significantly shorter and their financial statements are usually simpler than those of the corresponding 10-K documents. Therefore, the parsing of the 10-Q documents is comparatively simpler, and the performance statistics for the 10-Ks provide a conservative estimate of the performance of FRAANK in parsing the 10-Q documents.

To evaluate the performance of FRAANK, we first utilized the training dataset consisting of the 10-Ks of seventy-eight public companies from twelve different industries (see Table 2-Panel B for details) for the year 1999. This dataset is part of the eighty companies that were selected originally for the study to validate the C&I XBRL Taxonomy, Version 1, by Bovee et al. (2002). The original set of eighty companies consisted of top five/seven Fortune 500 companies in twelve different industries. Since two companies' 10-K documents for that year did not contain the financial statements, our sample size for this study reduced to seventy-eight companies. The training dataset was used to improve the parsing and tagging logic of the agent. Next, we used a sample of fifty randomly selected publicly held companies to evaluate the performance of the agent. Once the names of these companies were selected, we used the most recent 10-K report that was in the plain-text (not "html") format to extract the financial statements and do the parsing and XBRL tagging.

The statistical analysis of FRAANK's performance consists of two parts. The first part deals with analyzing how successfully the parsing logic of FRAANK identified the accounting numbers and their labels in the body of the 10-K documents. While FRAANK's parsing logic has to "understand" the table captions and column headings to identify the correct tables of the financial statements and the columns corresponding to the current time period, the parsing logic does not deal with "understanding" the meaning of the identified accounting numbers. This latter

problem is dealt with in the tagging logic of FRAANK, which is the focus of the second part of the performance analysis. The tagging logic of FRAANK "understands" the identified accounting numbers by matching their labels to the synonyms of the tags of the C&I XBRL taxonomy, Version 1 (XBRL 2000). The performance analysis focuses on how successful the complicated tagging logic is, and therefore it disregards the tagging failures that are due to the deficiencies in the underlying XBRL taxonomy.

Performance Statistics on Parsing Logic

To analyze how well FRAANK parses 10-K documents, we break down the performance statistics to show the success rates in identifying the following: (1) tables of Balance Sheet (BS), Cash Flow Statement (CFS), and Income Statement (IS); (2) column of the table for the current time period; (3) multiple lines; (4) lines with values; and (5) all lines including headings and subheadings. Tables 2 and 3 provide the details of the performance statistics for the training dataset and test dataset, respectively.

----- Insert Tables 2 & 3 here -----

The performance accuracy of table-extraction logic is determined by the ratio of extracted tables to the total number of tables. Table-extraction logic fails due to a variety of reasons like inefficiency of the program logic, or errors in the company's statement. For example, the CFS of IBM was labeled wrongly as "Statement of stockholder's equity." Also, sometimes companies misspell words in the table caption, and the agent does not capture the entire table or captures additional information. These cases are treated as errors in estimating the accuracy. For the training dataset, the overall performance accuracy of the agent is 96.2 percent (see Table 2, Panel A). Table 2-Panel B shows the overall performance accuracy of the table-extraction logic by industry

for the training dataset, indicating that the agent has been trained to perform very well at the industry level too. For the test dataset, the overall accuracy of the agent for the fifty randomly selected companies is 91.3 percent (see Table 3), with the lower limit of 86.2 percent at the 95 percent confidence level.[10]

The identification of the table column corresponding to the requested time period is challenging because there is no simple pattern of how companies lay out the financial information in different columns. Some companies put the most recent information in the first column and some in the last column. Also, some companies put the date in one row and some in two rows; some spell out the month, date, and year, while some others put the date in a format like "1/29/00." The performance accuracy of this module is determined by the ratio of correctly identified columns of values to the total number of columns in the tables that are correctly extracted. For the training dataset, the overall performance accuracy is 99.6 percent (see Table 2), while the performance accuracy for the test dataset is 100 percent (see Table 3).

Financial statements usually contain line items that extend to several lines. This presents a challenging problem for parsing since the first line of such a multiple-line item does not contain any value, and neither do headings and subheadings. Thus, the challenge is in determining whether a line with no value is a heading, a subheading, or part of a multiple-line item. The performance accuracy of the multiple-lines parsing logic is determined by the ratio of correctly identified multiple lines to the total number of multiple lines. In calculating the number of multiple lines per statement we consider the set of lines that constitute a single multiple-line cluster to be a single entity. Similar approach is used to determine the number of multiple lines not correctly identified. The parsing fails in cases where the multiple-line patterns are not covered by the logic. The overall performance accuracy of multiple-line identification is 97.4 percent for the

training dataset and 93.1 percent for the test dataset (see Tables 2 and 3).

The performance accuracy of identifying values is determined by the ratio of correctly identified line-item values to the total number of line-item values. If a column is not identified correctly then none of the values will be identified correctly. Thus, in measuring this accuracy, we do not include the errors due to identifying the wrong column. The overall performance accuracy of identifying values is 99.7 percent for the training dataset and 99.8 percent for the test dataset.

The final parsing-performance accuracy is determined by the ratio of the number of lines for which labels and values are correctly identified to the total number of lines (excluding lines from the statements missed at the table-identification stage). The overall reliability for correctly capturing the line item and its value is 94.7 percent for the training dataset and 88.4 percent for the test dataset (see Tables 2 and 3).

Performance Reliability of XBRL Tagging

The success of FRAANK in matching the line items in the 10-K financial statements to the C&I XBRL taxonomy tags is measured by: (1) the fraction of the number of lines correctly tagged, (2) the total dollar amount correctly tagged in relation to the total dollar amount that should have been tagged, and (3) the number of lines correctly tagged at various levels of the XBRL taxonomy. These analyses are presented in Tables 4 and 5, and are discussed below.

----- Insert Tables 4 & 5 here -----

Since there are many problems with the C&I XBRL Taxonomy, Version 1, as pointed out by Bovee et al. (2002), we excluded those lines for which there were no appropriate tags from counting the lines to be tagged, and counted as errors only those lines that were either not properly tagged due to programming errors or for which the tags were in doubt. We considered as errors the tagged lines that we were not sure were tagged correctly in order to get a conservative estimate of FRAANK's accuracy.

The second column of Table 4-Panel A presents the performance statistics for tagging line items broken down by the financial statement and overall for the training dataset. The overall accuracy is 88.4 percent, with 89.0 percent for BS, 88.3 percent for CFS, and 87.9 percent for IS. The second column of Table 5-Panel A provides the performance statistics by statement and overall for the testing sample of fifty randomly selected firms. FRAANK's performance in tagging line items is the highest, 87.3 percent for BS and the lowest, 70.4 percent for IS, with the overall accuracy of 80.4 percent. FRAANK's performance for BS is higher than for CFS or IS because we have spent much more efforts in improving the logic and collecting XBRL tag synonyms for BS than for CFS or IS.

Columns 3 – 9 of Table 4-Panel A and columns 4 – 10 of Table 5-Panel A provide the breakdown of line-item tagging errors by type for the training and testing datasets respectively. Note that the errors due to those line items for which tags were in doubt are not significant either on training or on testing. Errors due to programming issues increase on testing as compared with training, but not drastically. Errors due to XBRL taxonomy problems include the case of missing tags in the taxonomy and the case of missing synonyms for existing taxonomy tags in our synonym database. The significant increase in taxonomy problem errors on testing as compared to training is due to the latter case, and can be easily remedied by adding synonyms to FRAANK's database.

The performance statistics of tagging dollar values is determined by the ratio of the sum of the absolute dollar amounts correctly tagged by FRAANK in relation to the total sum of the absolute dollar amounts that should have been tagged. FRAANK is able to correctly tag 86.2 percent of dollar values on the training dataset (see the last column of Table 4-Panel A) and 85.5 percent for the test dataset (see the last two columns of Table 5-Panel A). We use the ratio estimation technique as described by Cochran (1977) to determine the lower limit of this performance reliability at the 95 percent confidence level.[11]

FRAANK's accuracy of tagging by the XBRL taxonomy level is reported in Table 4-Panel B (for the training dataset) and Table 5-Panel B (for the test dataset). Note that the highest level of the XBRL taxonomy corresponding to specific line items in the financial statements is Level 3, while Level 1 is the tag for "statement" and Level 2 tags determine the type of statement such as BS, CFS, and IS. The BS of companies in the training and test datasets have line items corresponding up to eight levels of the XBRL taxonomy, whereas the CFS have ten levels (one company had one line item at the tenth level) for the training dataset but nine levels in the test

dataset. The IS have thirteen levels for both datasets. The high accuracy of XBRL tagging persists even after being broken down by taxonomy level, and seems to be higher for the topmost levels of the taxonomy, which is explained by the proportionately larger number of synonyms collected for the topmost level XBRL tags (see Table 1).

VI. FRAANK's Applications

In this section we describe possible uses of FRAANK that can facilitate making decisions involving financial information, such as investment, financial health, and creditworthiness evaluation; analytical review procedures in auditing, and corporate benchmarking. A decision maker can use FRAANK from any place where Internet access is available, and the information will always be as timely as the data sources that are available online. If online continuous financial reporting were available, FRAANK would be able to incorporate it as the source of financial information to provide the timeliest analysis.

Internet-based online trading has recently become an important force in the financial markets and investment decision-making. An Internet trader may have to make a fast decision about either taking or liquidating a certain equity position. While watching how the trading in that security develops, this trader can simultaneously use FRAANK online to automatically find, retrieve and integrate relevant information, and compute various ratios. Savvy investors and financial analysts typically do such analyses manually. Since FRAANK performs these functions automatically, without explicit instructions from the user, it relieves users from the relatively routine and time-consuming part of their tasks, and expeditiously allows them to concentrate on the more intelligent aspects of the decision-making process.

A creditor can use FRAANK to facilitate a credit-granting decision, especially the quantitative analysis portion. For example, a loan officer will typically want to review the liquidity position of the company and to ensure that the company is not in financial distress. In such a scenario, the loan officer could instantly obtain through FRAANK the net income, the quick and current ratios, and the Z-score for the company.

The Z-score (Altman 1968, 1983) is a measure of bankruptcy risk that combines five ratios characterizing corporate liquidity (working capital/total assets), solvency (retained earnings/total assets), profitability (earnings before interest and taxes/total assets), leverage (market value of equity/book value of total debt), and activity (sales/total assets). To calculate the Z-score, FRAANK obtains financial numbers not only from the parsed 10-K or 10-Q statements but also from additional online information sources (e.g., a real-time stock-quote server). Therefore, this example demonstrates the importance of FRAANK's capability of integrating financial information obtained from various online sources.

Auditors are required to perform various analytical procedures (AICPA 1998, AU 329.05) that involve identifying unusual trends in account balances and their relationships by performing ratio analysis such as quick-and-current ratios, inventory turnover, gross-margin ratio, etc., and comparing these ratios with both industry standards, and with those of peers and competitors, as well as with past performance. FRAANK can help the auditor by automatically gathering online the relevant financial information about the client's peers, calculating the appropriate ratios, and presenting them to the auditor. This frees auditors from mundane tasks, enabling them to utilize the information provided by FRAANK and concentrate on more important and difficult judgments, such as making complex going-concern decisions or audit-risk assessments.

Management also compares financial ratios with industry averages and with those of peers, suppliers, and competitors to monitor and evaluate company performance and identify competitive threats and advantages. As described above, FRAANK can be used by management as an invaluable online tool for such benchmarking purposes. Using FRAANK, companies can quickly compare themselves to competitors in their industry sector or market across a variety of standard financial ratios.

VII. Concluding Remarks and Future Developments

This paper has described the design and implementation of FRAANK, the intelligent agent that gathers financial information about publicly traded companies over the Internet and then processes it to help various decision makers (e.g., investors, creditors, auditors, and managers). More specifically, FRAANK implements intelligent parsing to extract accounting numbers from the natural text of quarterly and annual financial statements available from the SEC EDGAR repository. FRAANK develops an "understanding" of the accounting numbers by matching the line-item labels to synonyms of tags in an XBRL taxonomy. As a result, FRAANK converts the consolidated balance sheet, income statement, and statement of cash flows into XBRL-tagged format. Based on FRAANK, we propose an empirical approach towards evaluating and improving XBRL taxonomies and for identifying and justifying needs for specialized taxonomies by assessing a taxonomy fit to the historical data, i.e., the EDGAR filings.

We evaluate FRAANK's performance in processing the 10-K SEC filings in two ways on two different datasets: the training dataset of seventy-eight companies and the test dataset of fifty randomly selected companies. First, we evaluate the success rates of FRAANK in identifying and extracting the following: (1) Tables of Balance Sheet (BS), Cash Flow Statement (CFS), and

Income Statement (IS), (2) Column of the table for the desired time period, (3) Multiple lines, (4) Lines with values, and (5) All lines including headings and subheadings. These success rates are reasonably high for both the training dataset and test dataset. Second, we evaluate FRAANK's success rate in tagging the line items using the C&I XBRL Taxonomy, Version 1.

The evaluation results show that FRAANK is an advanced research prototype that can be useful in various practical applications. FRAANK also integrates the accounting numbers with other financial information publicly available on the Internet, such as timely stock quotes and analysts' forecasts of earnings, and calculates important financial ratios and financial health indicators (such as Altman's Z-score).

The FRAANK agent lays down the foundation for additional research in the area of online financial reporting and auditing. Direct research resulting from this study will extend and refine the FRAANK agent's capabilities for gathering, isolating, and analyzing key financial and nonfinancial data from the Edgar database and other information sources. We discuss below several important areas in which FRAANK's capabilities can be significantly enhanced.

The current version of FRAANK analyzes line items of SEC filings using a separate program code and a table of synonyms for each line item analyzed. Although this approach allows FRAANK to achieve a high level of accuracy in identifying line items, it is very difficult to make this process 100 percent accurate without a tremendous amount of human involvement in the expansion of the synonyms database. We therefore plan to research the possibility of using machine-learning techniques for automated acquisition of new synonyms of accounting terms found in the text of financial statements, using the structure of the statements. The current version of FRAANK builds a foundation for this future development by identifying for every exceptional

line item the closest correctly matched tags above and below, and thus selecting a segment of the XBRL taxonomy where the matching tag should be located. This development should result in automatically expanding the table of synonyms of accounting terms whenever FRAANK encounters a new synonym.

Users of FRAANK will be interested in obtaining other types of relevant information not currently available from FRAANK. This includes intelligent digests of the general economic and company-specific news. Such information can be found on various information portals like Yahoo or Quicken. Future research and developments of FRAANK in this direction will add news subagents and intelligent digesting mechanisms to achieve this goal.

Although most companies' Web sites currently limit presented financial information to what is already available from the Edgar filings, they still contain useful additional information such as press releases. Moreover, future developments in online business reporting will make companies' Web sites an even more essential source of relevant information for FRAANK. However, obtaining that information presents a major challenge for the current FRAANK technology, since there are many companies whose corporate Web sites, subject to change without notice, vary greatly in structure, content, and format.. It will therefore be extremely important to research the possibility of developing new technology for finding important financial and nonfinancial information in companies' Web sites of arbitrary structure. This technology will probably utilize formal representation and learning of the structure of Web information sources (see, e.g., Perkowitz et al. 1997).

The intelligence capabilities of FRAANK can be enhanced by integrating it with some of the existing AI systems for accounting and auditing such as neural networks, rule-based systems,

or belief networks. For example, one can integrate FRAANK with artificial neural networks developed for predicting bankruptcies (Fanning and Cogger 1994), for recognizing potentials for management fraud (Fanning et al. 1995), and for making going-concern judgments (Biggs et al. 1992). This integration will result in providing automated timely expert advice about companies and industries.

References

Altman, E. I. 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance* 23 (4) (September): 589-609.

_____. 1983. *Corporate Financial Distress: A Complete Guide to Predicting, Avoiding and Dealing with Bankruptcy.* New York, NY: John Wiley & Sons.

American Institute of Certified Public Accountants (AICPA). 1998. AICPA Professional Standards, Vol. 1. AU Section 329.

Biggs, S. F., M. Selfridge, and G. R. Krupka. 1992. A computational model of auditor knowledge and reasoning process in the going-concern judgment. *Auditing: A Journal of Practice and Theory* (Supplement): 82-112.

Bovee, M., M. Ettredge, R. P. Srivastava, and M. A. Vasarhelyi. 2002. Does the Year 2000 XBRL Taxonomy accommodate current business financial reporting practice? *Journal of Information Systems* 16 (2): 165-182.

Cochran, W. G. 1977. *Sampling Techniques.* New York, NY: John Wiley & Sons.

EDGAR Database, U.S. Securities and Exchange Commission. 2002. http://www.sec.gov/edgarhp.htm.

EdgarScan, PwC Global Technology Centre. 2001. Available at: http://edgarscan.pwcglobal.com/servlets/edgarscan.

Fanning, K., and K. Cogger. 1994. A comparison analysis of artificial neural networks for financial distress. *International Journal of Intelligent Systems in Accounting, Finance and Management* 3 (4): 241-252.

Fanning, K., K. Cogger, and R. Srivastava. 1995. Detection of management fraud: A neural network approach. *International Journal of Intelligent Systems in Accounting, Finance and Management* 4: 113-126.

Ferguson, D. 1997. Parsing financial statements efficiently and accurately using C and Prolog. In *The Fifth International Conference and Exhibition on the Practical Applications of Prolog.* London, UK.

Franklin, S., and A. Graesser. 1997. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Intelligent Agents III. Agent Theories, Architectures, and Languages*, edited by J.P. Müller, M.J. Wooldridge, N.R. Jennings. *Lecture Notes in Computer Science* 1193. http://www.msci.memphis.edu/~franklin/AgentProg.html.

Garey, M. R., and D. S. Johnson. 1979. *Computers and Intractability—A Guide to the Theory of NP-Completeness.* San Francisco, CA: Freeman.

Gerdes, J. Jr. 2003. EDGAR-Analyzer: Automating the analysis of corporate data contained in the SEC's EDGAR database. *Decision Support Systems* 35 (1): 7-29.

libwww-perl Library. 2001. Available at: http://www.linpro.no/lwp/.

Mui, C., and W. E. McCarthy. 1987. FSA: Applying AI techniques to the familiarization phase of financial decision making. *IEEE Expert* 2 (3): 33-41.

Nelson, K. M., A. Kogan, R. P. Srivastava, M. A. Vasarhelyi, and H. Lu. 2000. Virtual auditing agents: The EDGAR Agent challenge. *Decision Support Systems* 28 (3): 241-253.

O'Leary, D. E., and T. Munakata. 1988. Developing consolidated financial statements using a prototype expert system. In *Applied Expert Systems*, edited by E. Turban and P.R. Watkins. Amsterdam: Elsevier Science Publishers. 143-157

_____ and T. Eis. 1991. A knowledge-based system for cash management: With implications for structuring DSS and with extensions for system learning and knowledge acquisition. *Advances in Working Capital Management* 2: JAI Press. 197-209

_____ and N. Kandelin. 1992. ACCOUNTANT: A domain dependent accounting language processing system. In *Expert Systems in Finance*, edited by D. E. O'Leary and P. R. Watkins. Amsterdam: Elsevier Science Publishers. 253-267.

Perkowitz, M., R. B. Doorenbos, O. Etzioni, and D. S. Weld. 1997. Learning to understand information on the Internet: An example-based approach. *Journal of Intelligent Information Systems* 8 (2): 133-153.

Searle, J. R. 1984. *Minds, Brains and Action.* Cambridge, MA: Harvard University Press.

Stein, L. D. 1999. CGI.pm - a Perl5 CGI Library. http://stein.cshl.org/WWW/software/CGI/cgi_docs.html.

Steier, D. 1995. Comparable datasets in performance benchmarking. In *Working Notes of the 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, edited by C. Knoblock & A. Levy.

_____, S. B. Huffman, and D. I. Kalish. 1997. Beyond full-text search: AI-based technology to support the knowledge cycle. *AAAI Spring Symposium on AI in Knowledge Management*, 161-167.

Tanaka, S. 1982. *The Structure of Accounting Language.* Tokyo, Japan: Chuo University Press.

Wall, L., T. Christiansen, and R. L. Schwartz. 1996. *Programming Perl*, 2d ed. Sebastopol, CA: O'Reilly and Associates.

Wong, C. 1997. *Web Client Programming with Perl: Automating Tasks on the Web*. Sebastopol, CA: O'Reilly and Associates.

XBRL. 2000. *XBRL Taxonomy: Financial Reporting for Commercial and Industrial Companies, US GAAP. 2000-07-31*, edited by S. de la Fe, Jr., C. Hoffman, and E, Huh. XBRL.Org. http://www.xbrl.org/Taxonomy/us-gaap-ci-2000-07-31.pdf.

Table 1: Numbers of XBRL Tags and Knowledge base Synonyms by XBRL Taxonomy Level[12]

Statement	XBRL Level	# of Tags	# of Syno-nyms
Balance Sheet			
	3	2	16
	4	8	45
	5	48	495
	6	81	385
	7	110	287
	8	35	84
	9	4	4
Total Count		**288**	**1316**
Income Statement			
	3	2	26
	4	6	65
	5	9	121
	6	17	185
	7	18	51
	8	5	69
	9	4	28
	10	19	151
	11	27	241
	12	11	69
	13	19	125
Total Count		**137**	**1131**
Cash Flow Statement			
	3	2	56
	4	7	129
	5	9	138
	6	10	118
	7	42	668
	8	44	507
	9	28	365
	10	10	10
Total Count		**152**	**1991**

Table 2: Performance Statistics of FRAANK for Parsing Logic for <u>Training Dataset</u> of 78 Companies

Panel A: Performance Statistics by Financial Statements															
Statement/Industry	**Table Extraction Logic**			**Column Identification Logic**			**Multiple Lines Parsing Logic**			**Parsing Logic for Capturing Value in Each Line**			**Combined Parsing Logic for Capturing Lines and Their Values including Multiple Lines**		
	Total No. of Tables	No. of Tables Captured	Reliability (%)	Total No. of Columns	No. of Columns Captured	Reliability (%)	Total No. of Multiple Lines	No. of Multiple Lines Captured	Reliability (%)	Total No. of Lines with Values	Value of Each Line Correctly Captured	Reliability (%)	Total No. of Lines	Lines Correctly Captured	Overall Reliability (%)
Balance Sheet (BS)	78	76	97.4	76	76	100	315	304	96.5	2961	2951	99.7	3045	2940	96.6
Cash Flow Statement (CFS)	78	74	94.9	74	74	100	427	422	98.8	2964	2958	99.8	3129	2953	94.4
Income Statement (IS)	78	75	96.2	75	74	98.7	192	184	95.8	1989	1983	99.7	2115	1960	92.7
Overall	234	225	96.2	225	224	99.6	934	910	97.4	7914	7892	99.7	8289	7853	94.7
Panel B: Performance Statistics by Industry															
Airlines	21	20	95.2	20	20	100	129	122	94.6	918	917	99.9	960	910	94.8
Automobile	21	21	100	21	21	100	135	135	100	733	726	99.1	733	726	99.1
Beverages	21	20	95.2	20	20	100	63	60	95.2	681	681	100	724	678	93.7
Computer Software	24	24	100	24	24	100	99	96	97	757	749	98.9	757	746	98.6
Computer and Office	18	17	94.4	17	17	100	88	88	100	569	568	99.8	607	568	93.6
Entertainment	21	21	100	21	21	100	113	112	99.1	826	824	99.8	826	823	99.6
Food and Drug Stores	21	19	90.5	19	19	100	58	58	100	690	689	99.9	777	689	88.7
Food Services	12	12	100	12	12	100	17	17	100	400	399	99.8	400	399	99.8
General Merchandisers	21	21	100	21	20	95.2	76	69	90.8	652	652	100	652	630	96.6
Motor and Vehicle Parts	12	12	100	12	12	100	48	48	100	413	413	100	413	413	100
Petroleum	21	17	81.0	17	17	100	43	41	95.4	599	599	100	764	597	78.1
Pharmaceuticals	21	21	100	21	21	100	65	64	98.5	676	675	99.6	676	674	99.7
Industry Mean Reliability			96.4			99.6			97.6			99.7			95.2
Industry Standard Error			1.67			0.40			0.86			0.11			1.85
95% Confidence Precision			3.68			0.88			1.90			0.23			4.07

Table 3: Performance Statistics of FRAANK for Parsing Logic for <u>Test Dataset</u> of 50 Companies

Panel A: Performance Statistics by Financial Statements																				
Statement/Industry	**Table Extraction Logic**				**Column Identification Logic**				**Multiple Lines Parsing Logic**				**Parsing Logic for Capturing Value in Each Line**				**Combined Parsing Logic for Capturing Lines and Their Values including Multiple Lines**			
	Tables		**Reliability (%)**		**Columns**		**Reliability (%)**		**Multiple Lines**		**Reliability (%)**		**Values Lines**		**Reliability (%)**		**Single + Multiple Lines**		**Reliability (%)**	
	Total Number	No. of Tables Captured	Sample Result (%)	Lower Limit at 95% Confidence Level	Total No. of Columns	No. of Columns Captured	Sample Result (%)	Lower Limit at 95% Confidence Level	Total No. of Multiple Lines	No. of Multiple Lines Captured	Sample Result (%)	Lower Limit at 95% Confidence Level	Total No. of Lines with Values	Value of Each Line Correctly Captured	Sample Result (%)	Lower Limit at 95% Confidence Level	Total No. of Lines	Lines Correctly Captured	Overall Sample Result (%)	Lower Limit at 95% Confidence Level
Balance Sheet (BS)	50	46	92.0	81.7	46	46	100	93.5	156	148	94.9	90.7	1582	1582	100	99.8	1727	1574	91.1	90.0
Cash Flow Statement (CFS)	50	44	88.0	76.3	44	44	100	93.2	204	190	93.1	89.3	1725	1722	99.8	99.6	1984	1708	86.1	84.8
Income Statement (IS)	50	47	94.0	84.5	47	47	100	93.6	89	80	89.9	82.4	1087	1082	99.5	99.0	1175	1073	91.3	90.0
Overall	**150**	**137**	**91.3**	**86.2**	**137**	**137**	**100**	**97.8**	**449**	**418**	**93.1**	**90.7**	**4394**	**4386**	**99.8**	**99.7**	**4886**	**4355**	**89.1**	**88.4**

Table 4: Performance Statistics of FRAANK for Tagging using C&I Taxonomy, Version 1 Tags for <u>Training Dataset</u>

Panel A: Tagging Reliability by Statement and by Error Type

Statement	Tagging Reliability (%) in Terms of Number of Lines Correctly Tagged*	Breakdown of Errors by Type							Tagging Reliability (%) in Terms of Total Dollar Value of All Lines**
		A: Total Number of Lines to be Tagged	B: Number of Lines with Taxonomy Error (No Matching Taxonomy Tag or Synonym)	C: Taxonomy Error in Percentage (100*B/A)	D: Number of Lines with Programming Errors	E: Percentage of Programming Errors (100*D/(A-B)	F: Number of Lines in Doubt whether They are correctly Tagged	G: Percentage of Lines in Doubt (100*F/(A-B)	
BS	89.0	2243	100	4.5	206	9.6	30	1.4	96.2
CFS	88.3	2316	65	2.8	263	11.7	1	0	85.2
IS	87.9	1619	73	4.5	180	11.6	7	0.5	71.7
Overall	**88.4**	**6178**	**238**	**3.9**	**649**	**10.9**	**38**	**0.6**	**86.2**

*Errors include programming errors and tags in doubt. This provides a conservative estimate of reliability.
**This reliability is defined to be equal to (Total dollar value of lines tagged correctly)/(Total dollar value of lines tagged correctly + total dollar value of lines tagged wrongly because of programming errors). Absolute value of each line item is taken while calculating the totals.

Panel B: Tagging Reliability by XBRL Taxonomy Level [#]

Statement/Level	3	4	5	6	7	8	9	10	11	12	13	**Total**
BS	100	97.9	92.9	92.5	72.1	92.9	N/A	N/A	N/A	N/A	N/A	**90.3**
CFS	98.7	94.9	98.2	58.2	91.5	89.6	84.7	0	N/A	N/A	N/A	**88.3**
IS	92.9	91.7	96.8	87.8	100	97.9	98.5	91.9	79	82.2	78.7	**88.3**
Overall	**99**	**95.6**	**94.5**	**84.6**	**85.3**	**91.6**	**86.7**	**91.6**	**79**	**82.2**	**78.7**	**89**

[#]This reliability is calculated based on the total number of lines tagged correctly per level in comparison to the total number of lines that had XBRL tags.

Table 5: Performance Statistics of FRAANK for Tagging using C&I Taxonomy, Version 1 Tags for <u>Test Dataset</u> of 50 Companies

Panel A: Tagging Reliability by Statement and by Error Type

Statement	Tagging Reliability (%) in Terms of Number of Lines Correctly Tagged		Breakdown of Errors by Type							Tagging Reliability in Terms of Total Dollar Value of All Lines	
	Sample Mean (%)*	Lower Limit at 95% Confidence Level	A: Total Number of Lines to be Tagged	B: Number of Lines with Taxonomy Error (No Matching Taxonomy Tag or Synonym)	C: Taxonomy Error in Percentage (100*B/A)	D: Number of Lines with Programming Errors	E: Percentage of Programming Errors (100*D/(A-B)	F: Number of Lines in Doubt whether They are correctly Tagged	G: Percentage of Lines in Doubt (100*F/(A-B)	Sample Mean (%)	Lower Limit at 95% Confidence Level
BS	87.3	85.5	1147	265	23.1	97	11.0	15	1.7	90.9	87.4
CFS	80.2	78.0	1190	355	29.8	155	18.6	10	1.2	85.9	76.4
IS	70.4	67.3	867	280	32.3	168	28.6	6	1.0	68.2	60.8
Overall	**80.4**	**79.1**	**3204**	**900**	**28.1**	**420**	**18.2**	**31**	**1.3**	**85.5**	**79.6**

*Errors include programming errors and tags in doubt. This provides a conservative estimate of reliability.

**This reliability is defined to be the ratio of the total dollar value of lines tagged correctly and the sum of the total dollar value of lines tagged correctly and the total dollar value of lines tagged wrongly because of programming errors. Absolute value of each line item is taken while calculating the totals.

Panel B: Tagging Reliability by XBRL Taxonomy Level [#]

Statement/Level	3	4	5	6	7	8	9	10	11	12	13	Total
BS	100	93.8	92	92.3	65.8	68.4	N/A	N/A	N/A	N/A	N/A	**88.8**
CFS	97.6	98.5	95.5	96.7	66.1	79.6	72.5	N/A	N/A	N/A	N/A	**81.2**
IS	83.3	58.3	84	34.5	83.3	100	89.7	67.9	62.8	75.5	33.3	**71.1**
Overall	**98.1**	**89.8**	**91.5**	**85.5**	**66.3**	**85.5**	**75.1**	**67.9**	**62.8**	**75.5**	**33.3**	**81.5**

[#]This reliability is calculated based on the total number of lines tagged correctly per level plus the total number of lines tagged wrongly because of programming errors per level.

Figure 1: The Architecture of FRAANK

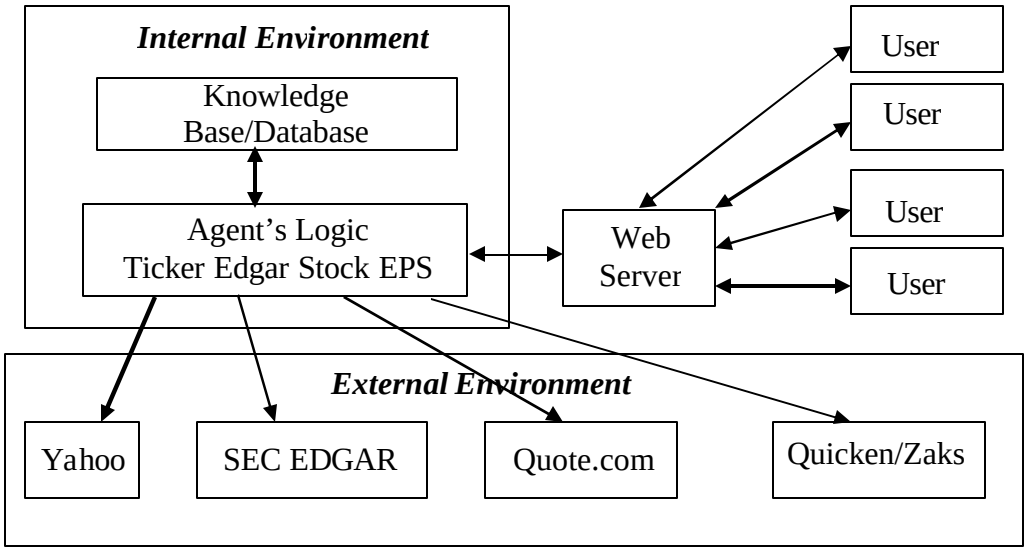
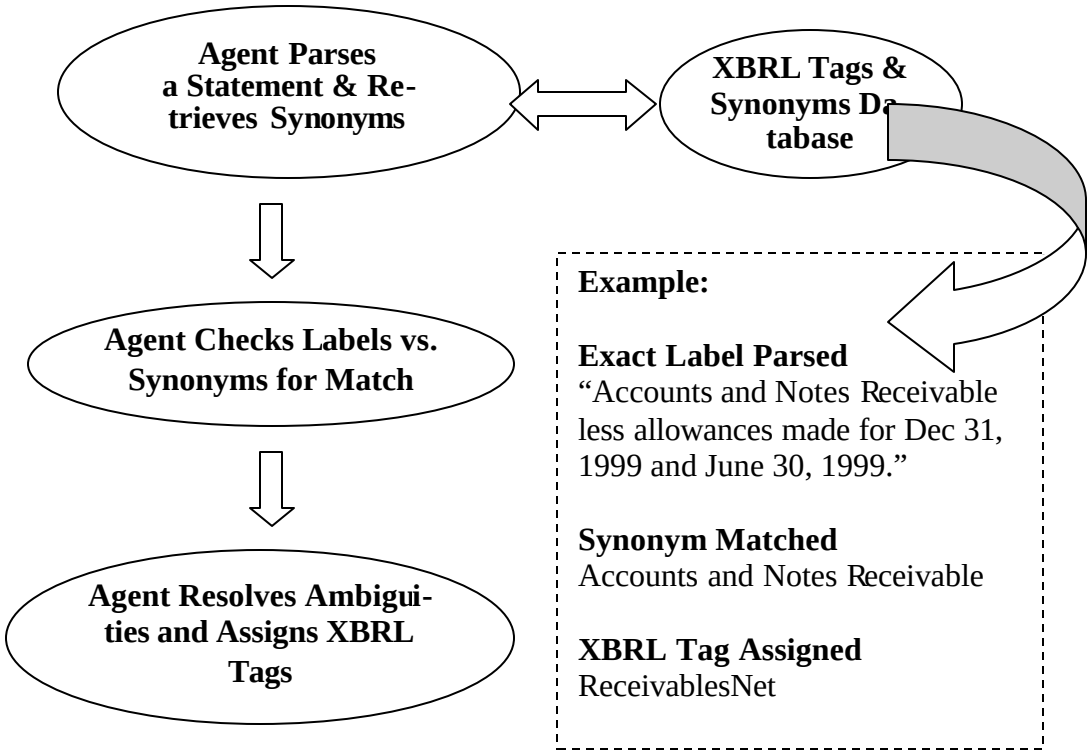


Figure 2: Synonym Matching in FRAANK



Endnotes

[1] See http://www.xbrl.org.

[2] EdgarScan is a Web-based interface to the SEC EDGAR filings, which is driven by a back-end engine that regularly retrieves EDGAR filings from the SEC servers and automatically parses them to find a subset of key accounting numbers, normalizes them to a common format, and stores the results in its own database. (See the homepage of EdgarScan at http://edgarscan.pwcglobal.com/servlets/edgarscan.)

[3] While the possibility of using FRAANK to evaluate XBRL taxonomies is a "side benefit" of FRAANK's architecture, this capability will become more important as the development of various XBRL taxonomies comes to the forefront of financial reporting.

[4] For an extensive discussion of what an agent is, see Franklin and Graesser (1997).

[5] There is an ongoing debate in the AI literature between the proponents of the so-called "strong AI" versus "weak AI" (see, e.g., Searle 1984). The strong-AI proponents claim that a "true" AI program has to be a universal model of a human-thinking process. The proponents of weak AI argue that it is more important and productive to concentrate on designing programs that only simulate intelligent behavior, i.e., produce results similar to what a human thinker would have done. While the weak-AI approach emphasizes that seemingly intelligent behavior may result from a seemingly trivial code, in most cases achieving useful results requires implementing very complex logic.

[6] The prototypes of the FRAANK agent which process the 10-Q and 10-K filings respectively can be found at:

http://www.fraank.eycarat.ukans.edu/cgi-bin/10Q_PAPER/10Q.HTM
http://www.fraank.eycarat.ukans.edu/cgi-bin/10K_PAPER/10K.HTM

[7] For example, the balance sheet of Compaq in its 2001 10-K filing has a line item labeled "Leases and other accounts receivable." Since the XBRL taxonomy does not have an appropriate tag, the agent tags this item with a generic tag "KU:unknown" and labels it with the correct description.

[8] Such problems are known as "NP-hard" in the theory of computational complexity (see, e.g., Garey and Johnson 1979).

[9] The need for industry-specific extensions has already been demonstrated (Bovee et al. 2002).

[10] In general, the Poisson distribution has been used for determining the lower limit at the 95 percent confidence level. However, for sufficiently large sample sizes the Normal distribution has been used.

[11] The tagging reliability in terms of dollar value is defined as: $R = \sum_i y_i \Big/ \sum_i x_i$ where y_i is the dollar amount on a statement correctly tagged and x_i is the dollar amount correctly tagged plus the dollar amount incorrectly tagged due to programming errors. The standard error of R for confidence interval is determined by Equation 2.47 in Cochran (1977): $s(R) = (n/\sum x_i)\sqrt{(\sum y_i^2 - 2R\sum y_i x_i + R^2 \sum x_i^2)/n(n-1)}$.

[12] XBRL level 1 is the tag for "statement" and Level 2 tags determine the type of statement such as BS, CFS, and IS.