FunBase: A Function-based Information Management System

Wafik M. Farag (farag@umich.edu)
Toby J. Teorey (teorey@umich.edu)

Software Systems Research Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122

Abstract

The volume of data required for decision making is growing fast. With technology advancement, decision makers not only face massive amounts of data to process, but also have to make their decisions quickly. This leaves current database technology for some domains limited in use. We propose a function-based system (FUNBASE) which automates information management for domains where current database technology does not scale up. The strength of FUNBASE stems from its three fold capability: first to share knowledge among users, second to use external criteria for accessing underlying data in a database, and third to automate the acquisition of data from flat files with short life spans. These features make FUNBASE a collaborative medium for information exchange as well as a vehicle for speeding the processing of results by integrating data and functions while keeping them modular for changes and updates.

1 Introduction:

Information management is usually driven by user needs, expectations, and technology. When the environments in which users operate change, a time lag follows until tools are created to reflect this change. We need to study the problems facing users, the available tools, and where research should be conducted to minimize such time lags (e.g., [French 90]). We propose a new system FUNBASE to address problems that occur in real world domains. In identifying those problems, we have found current tools to be inefficient, and we expect by introducing FUNBASE to shorten the time lag for such domains.

The concept behind FUNBASE can be clarified by a simple example. An individual wants to go for a relaxing vacation. A number of criteria define for him/her what is relaxing: dry temperatures between $70^o - 80^o$ F, clear blue sea water, and fine white sandy beaches. If the vacation seeker calls an airline company for reservations, the operator normally asks for a specific destination or no assistance is feasible. Left with no answer, the potential traveler seeks the knowledge of a travel agent or friends. A travel agent maps these criteria to specific data available in an airline database. But the criteria given by the potential traveler does not exist in an airline database. The mapping done by the travel agent therefore simulates a function which converts given criteria (hereafter called "external criteria") to data useful for extracting what is needed from the database. The would be traveler may need to check with a number of travel agents before arriving at a decision. Having functions (i.e., travel agents) integrated in a function base for querying on expedites the decision making process while achieving a more educated result. Similarly, FUNBASE extracts data from an underlying database by applying suitable functions from its function-base with external criteria as input arguments.

The above example is a typical scenario for what users in some domains experience. Integrating functions with data provides a faster way to achieve results. Meanwhile, we need to move from accessing data-by-data to accessing data via external criteria. FUNBASE leads us into the information age by providing a base of functions which manages a base of data (i.e., database) so users can use external criteria to extract information (i.e., output from function(s)). This elevates the neces-

CIKM '93 - 11/93/D.C., USA
© 1993 ACM 0-89791-626-3/93/0011 ....$1.50

sity for users to have a priori knowledge of the form or structure of the underlying data to extract what they want. When data is of a massive and changing nature FUNBASE becomes of a of a particular value for fast information access.

In this paper we begin with an overview of FUNBASE. Then we present High Yield markets in the domain of a financial market to identify characteristics of domains for which FUNBASE is useful. Next we examine how FUNBASE would aid in analyzing the High Yield market. Section five argues why relational and object oriented databases do not scale up. In conclusion, we highlight FUNBASE's main contributions.

2 FUNBASE: An Overview

FUNBASE's objective is to tie data with functions and produce more meaningful results. The need to have access to shared data has been well established; evolving from flat files, relational databases have provided an improved concept of data sharing. The same need is occurring now with respect to functions. Sharing functions is just as important as sharing data because in some cases data is useless without having the correct function. Similarly, knowing what functions are used in achieving a result or what set(s) of data are the constituents of a result is important. Thus FUNBASE is designed to tie together three components: data, functions, and results, as three layers integrating into a single architecture (see Figure 1.a).

2.1 FUNBASE's Layers

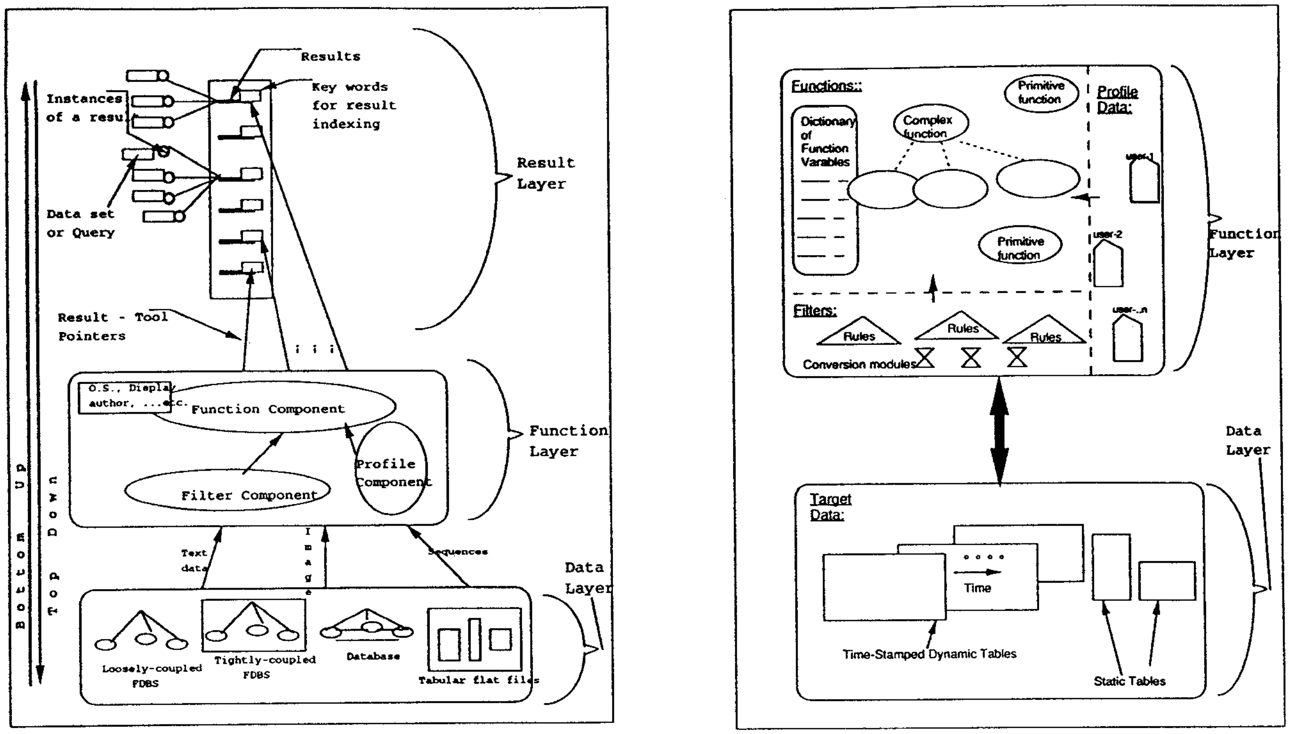
The contents of each layer is presented, followed by a description of how the three layers are integrated.

I. *Data Layer:*
Data can be of any form, simple textual data, remote-sensing data, genomic maps (i.e., spatial structure of chromosomes), etc. FUNBASE makes a distinction in the form in which data is presented and not in the structure or type of data itself. Data can either be presented via a database system or as a flat file. Database systems have query languages to extract and manipulate data, but at a cost: data entry in databases is a tedious and time consuming task. On the other hand, flat files are easy to connect to a system if there is support for automatic data extraction in the system. FUNBASE supports flat files that are of fixed format, columns and rows vs. plain text, etc. Minimal functionality for data extraction is provided by the system. Further data processing and manipulation is done by the function layer. The real world domains we encountered dealt more with data in flat files. Thus we chose flat files to be the form of data input for our testing. We refer to data in the data layer as *target data*, because it is basic stored data operated on by functions in the function layer.

II. *Function Layer:*
This layer contains the base of functions that extract and manipulate data and outputs results (i.e., information) into the result layer. Because functions are treated as black boxes, two other components are supplied – filters and profiles – which allow users to tune the function's performance according to their needs. A description of the three components – functions, filters, and profiles – follows.

Function Component: Users supply and use functions. Collaboration occurs when users supply functions in their area of expertise and allow others to access those functions. Functions are viewed as black boxes. FUNBASE keeps track of a function's attributes and not of a function's computational aspects. A set of keywords describing a function's computational behavior is integrated into the system as one of the function's attributes. These keywords are supplied by the function's author. The set of attributes identifying a function are stored in a dictionary-like fashion forming the integral part of a function model. Inputs, results, author, platform, display requirement, keywords, etc. are examples of a function's attributes. Querying on these attributes allows users to navigate and select a suitable function. Using the traveler example mentioned earlier, a query to the system could be to search for a function with *input(temperature, beach, sea_water)* and/or *result(city, nearest_airport, state)* from the set of functions in domain *vacations*.

Functions can be either primitive or complex. Primitive functions produce a computationally defined result with no side effects from input types. For example, an addition function accepts integers as input no matter what this input is, whether it is a temperature or a salary, as long as it is of the primitive type integer. If we want a function to add time as hours.minutes and time zones, then a complex function with knowledge about time is needed. Complex functions have capabilities like function overloading in programming languages, where a function name can have different results depending on the set of inputs. In Figure 1.b, complex functions that share common functionality forming a class are visually represented in a network-like fashion. Another class of complex functions we call *dimensional-functions* introduce capabilities to functions with input attributes or results that are time or space



(a) A three layered system

(b) A detail of two layers

Figure 1: FUNBASE : An architecture for an information system

dependent. This allows users to span the capabilities of existing functions over time or space. Dimensional-functions execute simple or complex functions iteratively with different data sets, computing output that is time or space dependent (e.g., time series analysis).

Functions are categorized according to their role: input functions, computational functions, and output functions. A user can query on what output functions can be used on the result of a computational function that is of interest. In the traveler example, an output function that takes *input(city, state)* as input and displays it on a world map enhances the visual aspect in decision making. This categorization allows users to discover new ways implemented by other users on how to present their output, or on types of input forms that are accessible.

Filter Component: Filters are of two types: rules and translate modules. Rules are used to massage input coming from the data layer according to the actions of a rule when its preconditions are satisfied. Massaging input data enables users to have control over what data a function receives since functions are treated as black boxes. "If-Then-Else" is the syntax used for rules. **AND**, **OR**, and **NOT** are operators used in rules specifying relations on attributes.

Translate modules handle syntactic conversion of data; for example, temperature conversion from Celsius to Fahrenheit or visa-versa. These modules convert data to correctly fit a function's inputs, preventing data mismatch.

Profile Component: Different users may opt for different strategies to achieve their goal, so it is imperative for functions to be able to reflect such variations. The profile component provides users with the ability to tune a function's operation according to their needs. Each user or group of users may have a data file in this component. The data file contains function names and preference values for the corresponding input arguments with defaults. Functions should to be written putting into consideration kinds of input arguments. Input arguments that reflect a user's preference should have a default value. This allows FUNBASE to replace the default values for a function's inputs with the corresponding user preference values found in one's profile file. In case a preference value is not found for a particular input, or no profile file exists, FunBase executes the function with its default values. Data files appear in this layer because it is data for tuning functions. This is different from the target data in the data layer.

Having presented the different components of the function layer, we now present definitions of a function and a function's possible inputs as follows:

Definition 1: A function model F is a collection of sets $F = \{f_1, f_2, f_3, ..., f_n\}$ where f_i is either an input, output, or a computational function. Each function f_i is dependent on d inputs ($d \geq 0$) where $d \in D$. D is a set of inputs d of different sources.

Definition 2: For each f_i in F, input is a set $D = (d_1, d_2, d_3, ...d_n)$ of inputs; d_i may be any input of the following input sources:

1. *Target Data* (TD):– may or may not be filtered by rules or conversion modules from the filter component.
2. *Profile Data* (PD):– tunes functions according to user's preference.
3. *External Criteria* (EC):– the function's input arguments supplied by the user.
4. *Information* (I):– the output of another function f_i within the function-base.

III. *Result Layer:*
In the function layer, the function dictionary keeps track of the function's attributes: like input arguments, type of output, keywords describing the computational aspects of the function, etc.. Similarly, in the result layer, results are kept track of. Each time a function is executed, the user has the option of whether or not to save the result. When a new function is added to the function model the function's author is asked to supply keywords that would best describe a generalization of what such a function would output into the result layer. Later uses of the function by different users might result in adding new keywords that best describes each users' perspective. This helps to minimize the semantic problem by capturing different keywords for describing the same function.

When a result is obtained from applying more than one function in series, the result is tagged according to the last applied computational function with a cross reference to the last applied function. The last applied function would be an output function in this case. Each instance of a result (i.e., result from a function's run) stores information about that particular run. Stored information contains attributes such as user's name, time stamp of run, etc. in addition to a capsulation of the procedure. The capsulation of a procedure includes: the data used (especially if it is time varying), filters applied, data from profile file(s) called by various functions, and the sequence of functions applied in case of more than one function. Result capsulation preserve the steps taken during a run which becomes valuable for future studies on how results were obtained.

Results may appear similar to target data, especially if they are of the same type. However we do not include results in the data layer for of two reasons. First, results are processed data, meaning they are stored in this layer to capture the procedure used in producing them. Storing the results in the data layer will loose such *information* pertaining to their existence. The second reason is the complexity that might take place if a result is of a new type not currently existing in the underlying data layer. The simplicity of adding a new type depends on the data model it needs to be added to. Thus, adding results in the data layer is left to the user's judgment to copy them as needed.

With the three layers briefly described, we define an information system as follows:

Definition 3: A system $S(D, F, I)$ is an information system when output I is achieved by applying functions F to input D. Where I is a collection of sets $I = \{I_1, I_2, I_3, ..., I_n\}$, and each I_i is a set of data instances of type i. D is a collection of input data sets $D = \{D_1, D_2, D_3, ..., D_n\}$, where each D_k is a set of data instances of type k. A data type i of I_i may not necessarily be the same type k of $d_k \in D_k$ after applying function F.

We argue that for an information system to be functional, the three parts – data, functions, and results – ought to be shared and integrated. This allows a user to benefit from the system.

2.2 Access Paths in FunBase

We view FunBase as a two way street, with two paths for accessing information. Finding a result of interest to a user might lead to further accessing of the data set used and the function(s) applied. This path is a **top down** access method (see Figure 1.a). The second path is **bottom up**. In this path a user searches for a function to apply on a data set. Finding a function is similar to defining a concept on a set of attributes found in the data layer. This method helps discover concepts among data sets that other users have identified earlier.

Let us now explore a real-life example using the data and function layers in more detail. The result layer has yet to be implemented.

3 A Domain Example: High Yield Market

From a number of available applications encountered (e.g., [CIESIN 91]), we selected the High Yield bond (HY) domain from the financial market because it has diverse and challenging problems that still need to be addressed. Unlike the equity (stock) market where technical and fundamental information is available from a number of public databases, the HY or subinvestment grade market continues to be a new and evolving market with little information and insight in behavior. The HY market has a much lower volatility than the stock market, but because of its *contractual* nature, default risk is its primary concern [Markowitz 87, Merton 69].

The HY market contains companies with bonds rated BB, B, or CCC or lower, which makes them riskier than investment grade securities with ratings of BBB and above. As with any financial system, the goal is to maximize the return on investment and minimize risk [Markowitz 52]. Financial analysts study external factors affecting the HY market to formulate a strategy for selecting a small set of companies expected to satisfy their goal.

Studying external factors affecting HY market's performance (e.g., economical, regulatory, prevailing capital market conditions, demographical shifts, etc.) and formulating a strategy to address companies issuing these bonds which come from a diversity of industries, makes it hard if not impossible to have a reliable big picture of the HY market investigated by a single person. Currently, each financial analyst is assigned to analyze a number of companies, according to the securities they issue, in one or more industries (e.g., healthcare, airlines, food, etc.) which together form the HY market. After studying the factors affecting these industries, the financial analyst formulates a strategy on how the market (i.e., economic outlook) is expected to perform resulting in two sets of companies, a set with the best returns and a set with worst. This portfolio management strategy is based on a *quantitative portfolio management*, which is an analysis using technical information on top of financial data. The following example is drawn from steps based on quantitative portfolio management.

In this method a top down approach of portfolio management is illustrated. This approach views the major micro and macro economic outlook which determines the investment process leading to the selection of securities that fit such a profile. From a number of significant performance factors affecting securities, a multi-factor model has been identified (e.g., liquidity, rating, seniority, industry, price/yield, etc.).

An analyst starts with a table of companies ranked alphabetically by name, from which a number of tables are generated according to rankings by other criteria. Figure 2.a shows a sample of data from one of the generated tables indexing companies according to "Moody's" and "Standard and Poor's" (S & P) rating. See `Rating MDY/S+P` column for ranking. Figure 2.b shows a sample of data ranked by industry, which is one of the factors in the multi-factor model. This indexing is based on external knowledge of each company's service or product, thus categorizing each company accordingly. An analyst then uses the multi-factor model to map the external factors into a set of basic factors affecting company's selection (i.e., liquidity, rating, seniority, industry, price/yield, etc.).

For example, an analyst's strategy expects there will be a good economy and a strong market. Using the multi-factor model, the analyst maps a good economy to be a preference for cyclical companies. Similarly, the analyst maps a strong market to be a preference for higher yielding securities. Cyclical industries are greatly affected by the state of the economy and are preferred over defensive industries in a strong to recovering economy. The Air Transport industry, for example, is cyclical because of the strong correlation between air travel use as an option for vacation travelers, who greatly affect such an industry, and who are themselves affected the state of the economy. Industry ranking by type, whether cyclical or defensive, is based on general knowledge of the industries' behavior in varying conditions. Another approach in categorizing industry by type is the historical correlation of that industry with the Gross Domestic Product (GDP). Security categorization according to the multi-factor model helps in the selection process. Let's say an analyst has two industries designated: Transportation and Utility. Applying the first basic factor, Transportation, a cyclical industry, is chosen for further examination (see Figure 2.b), while Utility, a defensive industry, is dropped. Tables like Figure 2.b makes it easy for the analyst to select from the set of companies that appear only in the Transportation industry and which fit the profile.

Price/yield is another basic factor used for company selection satisfying the second expectation, which is a strong market. Higher yielding securities is a fuzzy term which depends on how aggressive the analyst's profile is in dividing the scale. Dividing the total HY market scale into five quartiles causes the highest yield securities to lie in the first quartile, the higher yield to become the second quartile, etc. For example, if you add the Yield to Worst (see Figure 2.b, Yield to Worst column) and divide that by five you will get five quartiles of the market with the first quartile containing companies with the highest yield and so forth. Dividing the HY market into ten quartiles reflects the profile of an analyst who is more aggressive because s/he chooses a smaller number of companies in a single quartile, thus being more selective. In our example, examining the

Company	Coupon	Maturity	Rating MDY/S+P	Amt Out ($MM)	Price	Yield to Worst	Yield Chg (B.P.)	Actual Return	YTD Actual Return	Current Sprd over Tsy (B.P.)	Change in Sprd (B.P.)	Start Prem over Tsy	Ending Prem over Tsy
WAINOCO OIL CORP	12.000	08/01/02	B1 B-	100	102.50	11.46 W	-22	1.92	4.13	557	4	90.2	94.7
WAXMAN INDUSTRIES INC	13.750	06/01/99	B2 B-	100	108.00	11.21 R	-7	.99	14.17	741	28	171.8	195.3
WOLVERINE TUBE INC	10.125	09/01/02	B2 B	100	100.75	9.98 W	-14	1.56	1.73	409	11	64.9	69.6
WOODWARD & LOTHROP INC	14.750	08/15/95	B3 B-	95	50.00	48.93 M	68	2.33	6.75	4468	109	935.7	1,052.7
Par Averages ---		Subtot Par Val --		35474		10.95	-19	.78	14.92	609	15	124.6	138.4
Mkt Averages ---		Subtot Mkt Val --		35789		10.66	-23	.81	14.07	583	11	121.4	134.4
Split B													
BALLY'S PARK PLACE FUNDI	11.875	08/15/99	B3 CCC-	350	103.25	11.00 W	-21	1.65	18.54	568	6	100.6	107.2
CABLEVISION INDUSTRIES C	11.250	02/15/02	B2 CCC+	150	100.88	10.52 W	-24	1.14	13.39	745	16	210.6	243.8
COAST S & L ASSOCIATION	16.000	10/01/94	CAA B-	100	88.50	23.58 M	28	1.36	13.06	1978	64	461.3	521.3
COAST S & L ASSOCIATION	15.750	05/15/00	CAA B-	125	83.00	20.20 M	3	1.44	13.93	1431	28	228.6	243.3
EPIC HEALTHCARE GROUP	15.000	02/01/01	B3 CCC+	115	110.50	10.33 R	51	.18	8.86	726	91	183.6	237.6
EPIC HOLDINGS INC.	12.000	03/15/02	B3 CCC+	250	59.25	12.07 M	-10	1.72	6.04	571	16	83.9	89.8
EXIDE CORP	12.875	06/15/97	CAA B-	135	105.00	10.83 W	-48	1.19	25.81	790	5	226.8	269.6
KASH N' KARRY FOOD STORE	14.000	02/01/01	B3 CCC+	105	103.50	13.05 W	-1	1.07	16.67	774	27	133.8	145.8
LEVITZ FURNITURE CORP	12.875	05/15/97	B3 CCC+	93	97.00	13.77 M	227	-5.52	29.46	845	123	168.8	159.3
LOEHMANN'S	13.750	02/15/99	CAA B-	108	99.50	13.86 M	27	.11	7.93	797	-134	217.7	135.5
MCCAW CELLULAR COMMUNIC	14.000	06/15/98	B3 CCC+	400	108.25	8.36 W	-209	2.16	13.03	543	-155	201.8	185.3
MESA CAPITAL CORP	12.000	08/01/96	B3 CCC-	300	92.00	15.05 M	-169	6.16	16.47	1108	-137	291.2	279.9
MESA CAPITAL CORP	13.500	05/01/99	B3 CCC-	300	97.00	14.21 M	-124	6.29	17.32	832	-99	151.7	141.5
PA HOLDINGS CORP	13.750	07/15/99	CAA B-	157	108.25	11.02 W	-50	1.46	16.52	796	110	147.3	260.2
SULLIVAN GRAPHICS INC	15.000	02/01/00	B3 CCC-	100	90.25	17.39 M	-6	1.59	45.22	1150	20	184.2	195.5
SUPERMARKETS GENL HOLDG	14.500	09/15/97	B2 CCC+	475	106.25	11.06 W	-23	1.05	11.95	800	18	226.0	261.6
SYBRON ACQUISITION CO	13.250	08/15/98	B3 CCC+	160	107.00	8.02 C	-8	.99	11.99	523	36	150.9	187.9
Par Averages ---		Subtot Par Val --		3423		12.34	-52	2.03	15.47	809	-20	191.1	205.3
Mkt Averages ---		Subtot Mkt Val --		3364		12.14	-54	1.93	15.29	800	-21	192.3	207.5
CCC/Split CCC													
AMSTAR CORP	11.375	02/15/97	CAA CCC-	195	98.00	11.97 M	0	.93	43.43	800	32	179.7	202.1
CULLUM CO	14.000	12/01/98	CAA CCC+	100	109.50	9.29 R	-24	1.00	24.69	622	16	175.1	203.6
DR PEPPER/SEVEN-UP COS I	15.500	10/01/98	CAA CCC+	420	101.00	11.83 M	-23	2.02	11.60	652	60	96.6	122.9
ENVIROSOURCE INC	14.000	04/01/98	CAA CCC-	160	106.00	12.40 W	-3	1.04	58.38	815	1	190.4	192.1
INTERNATIONAL CONTROLS C	12.750	08/01/01	CAA CC	130	88.00	15.26 M	-195	11.10	85.92	889	-170	160.1	139.9

(a) Sample of Data Ranked by Rating

Company	Coupon	Maturity	Rating MDY/S+P	Amt Out ($MM)	Price	Yield to Worst	Yield Chg (B.P.)	Actual Return	YTD Actual Return	Current Sprd over Tsy (B.P.)	Change in Sprd (B.P.)	Start Prem over Tsy	Ending Prem over Tsy
SOUTHLAND CORP	4.500	06/15/04	B2 NR	206	55.75	11.46 M	-62	5.76	34.18	510	-36	82.6	80.2
SPECIALTY RETAILERS INC	14.625	08/15/99	B2 B	100	112.00	9.53 R	-124	2.86	24.22	573	-88	159.4	151.0
SUPERMARKETS GENL HOLDG	11.625	06/15/02	B3 B-	200	104.50	10.74 W	-19	1.82	8.64	485	6	78.2	82.6
WOODWARD & LOTHROP INC	14.750	08/15/95	B3 B-	95	50.00	48.93 M	68	2.33	6.75	4468	109	935.7	1,052.7
ZALE CORP	13.125	06/01/07	CA D	200	7.00	19.00 M	11	.00	-22.22	1264	37	185.5	198.7
ZALE CREDIT CORP	11.900	06/01/97	CA D	75	43.00	19.00 M	-21	2.38	72.00	1368	7	243.6	257.8
Par Averages ---		Subtot Par Val --		4420		11.80	-3	-7.20	-14.95	644	5	133.2	135.9
Mkt Averages ---		Subtot Mkt Val --		2586		11.09	3	.36	-1.06	584	8	123.5	124.1
Transportation													
AP INDUSTRIES INC	12.375	06/15/01	CA D	88	9.00	29.72 M	29	.00	5.88	2336	55	344.8	367.3
AUBURN HILLS TRUST	16.875	05/01/20	B2 B+	1100	129.00	12.98 M	11	.26	67.96	562	17	73.6	76.5
CHRYSLER CORP	12.000	11/15/15	B2 B+	800	103.00	11.55 W	22	-.49	55.62	518	47	71.2	81.5
EXIDE CORP	12.875	06/15/97	CAA B-	135	105.00	10.83 W	-48	1.19	25.81	790	5	226.8	269.6
FEDERAL-MOGUL CORP	8.375	10/01/93	BA1 BB+	100	101.50	6.77 M	-60	1.12	8.73	371	-19	112.9	121.3
FLEXI-VAN LEASING INC	13.500	05/15/98	B3 B-	100	108.75	8.22 R	-49	.96	15.73	528	-8	160.0	180.4
GREYHOUND LINES INC	10.000	07/31/01	B2 NR	165	92.00	11.36 W	9	.33	19.40	500	35	70.3	78.6
HERTZ PENSKE TRUCK LEASI	11.500	07/15/97	BA3 BB+	100	105.25	8.00 W	-59	1.10	10.18	494	-18	147.9	161.5
HYSTER-YALE MTLS HNDLING	12.375	08/01/99	B1 B+	200	108.25	9.73 W	-12	1.14	14.09	576	20	130.2	145.5
INTERNATIONAL CONTROLS C	12.750	08/01/01	CAA CC	130	88.00	15.26 M	-195	11.10	85.92	889	-170	160.1	139.9
KELSEY-HAYES CO	13.250	11/15/94	NR BB	213	100.50	6.86 R	-7	1.02	10.16	407	38	114.5	146.0
LEAR SEATING CORP	11.250	07/15/00	B3 B-	125	102.50	10.56 R	25	-.09	4.59	525	53	84.5	99.0
LEASEWAY TRANSPORTATION	13.250	08/01/02	CA D	193	35.00	10.99 M	9	.00	89.19	462	35	64.6	72.7
NAVISTAR FINANCIAL CORP	11.950	12/01/95	B1 BB+	92	92.50	15.01 M	636	-6.56	1.49	1076	534	167.9	253.5
NORTHERN PACIFIC CORP	13.750	06/15/97	CAA NR	200	10.00	140.09 M	71	8.41	-21.00	13478	99	2,393.9	2,538.5
RIO GRANDE INDUSTRIES IN	13.625	05/15/95	B2 B-	65	101.00	13.14 M	-1	1.04	16.79	889	41	182.2	209.6
SEALED POWER TECHNOLOGIE	14.500	05/15/99	B3 B-	100	110.00	10.15 W	-218	3.27	18.19	721	47	120.6	246.2
SPTC HOLDING INC	12.125	10/01/00	B2 B-	200	97.00	12.73 M	21	-.02	28.36	684	46	104.0	116.4
UNIROYAL GOODRICH TIRE C	14.125	06/15/98	BA3 NR	250	110.00	6.64 R	-66	1.00	12.64	370	-13	110.8	126.4
VARITY CORP	11.375	11/15/98	B1 BB	150	104.00	10.15 R	-31	1.80	20.69	619	1	144.6	156.3
Par Averages ---		Subtot Par Val --		4506		17.25	4	.95	38.75	1181	32	211.2	232.8
Mkt Averages ---		Subtot Mkt Val --		4506		11.79	-0	.59	36.47	625	28	110.9	126.6
Utility													
ARKLA INC	8.875	07/15/99	BA1 BB+	200	100.00	8.87 M	-0	.71	1.87	298	25	44.5	50.7
CTC MANSFIELD FUNDING CO	11.125	09/30/16	BAA3BB+	593	112.75	6.89 W	-107	.59	6.13	382	53	70.8	125.2

(b) Sample of Data Ranked by Industry

Figure 2: Sample of Data used by HY Analysts (Source of First Boston Corp.)

companies satisfying the first criteria which lie in the Transportation industry in Figure 2.b, we find that four companies fall within the higher yield criteria range: International Controls with a yield of 15.26, Auburn Hills Trust with a yield of 12.98, SPTC Holding Inc. with a yield of 12.73, and Greyhound Lines Inc. with a yield of 11.36. Examining Figure 3.b, SPTC Holding Inc. does not appear in the list of Best Returns while the three other companies do appear with rankings that reflect their corresponding yield. Because our example is simplified by taking into consideration two basic factors only: industry and yield, it does not provide a complete analysis. Applying the multi-factor model results in Figure 3.B with the list of Best/Worst companies.

A vital task analysts need to monitor is testing their premises by examining their multi-factor model of the market. In the above example, one has to test if higher yielding companies continue to have good returns in a good market scenario. Another test is for industry type, to see if an industry continues to adhere to a particular type, whether cyclical or defensive. These tests are verified over time. When new data of a HY market comes in, the analyst tests his/her expectations to check if the basic factors applied for a scenario still hold, and to see that no shift in the market took place.

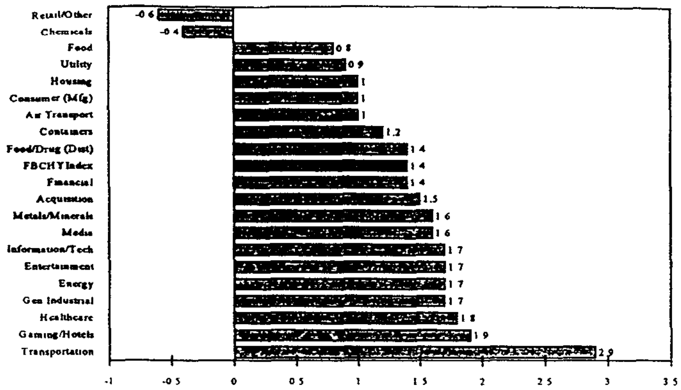
4 FUNBASE as a Financial Analyst

Using the above example we show how FUNBASE would play a vital role when used as a tool in HY market analysis.

Data in Figure 3.a and Figure 3.b are considered dynamic data, as they change frequently to reflect the moment by moment changes in price and yield. Files having infrequently changing data are identified as static files. A data file containing securities names and the industry they belong to is an example of a static file. Another example is the categorization of industries according to their type, cyclical or defensive. An industry's type can change, but only after a long period of testing its securities's pattern of performance. In the function layer, analysts define functions to simulate their strategy under different economic outlooks (i.e. scenarios) for the industries they are assigned to. A function *Eco_Mkt* with input arguments *input* (*economy_status*, *market_status*, *industries*, *yield* := 5) and output *result* (*company_name*, *coupon*, *maturity*, *price*, *yield_to_worst*) is a sample of a function defined in the above example. Such a function when applied to the basic table of companies ranked by name would extract the set of four transportation companies mentioned earlier. Applying a function which takes into consideration more factors, only three of the four companies will result as appears in Figure 3.b. Data found in a users's profile file might also alter such a selection. An entry *Eco_Mkt* : *yield* = 6 in a user profile file would tune that function to divide the market into six quartiles instead of the default value of five. Another way of altering the final result is to use filters. Let's assume that new information just came in about a company in default, filing chapter 11 with deteriorating conditions. A rule such as: `IF MDY_Rating=D THEN Remove security` filters it from consideration. In this case we are hypothetically assuming that the rating factor is not taken into consideration in the used function. Rules help to filter input before reaching a function in order to easily reflect the effect of new criteria that were not taken into consideration while writing the function. Having a function model analysts can define different functions with the same intended result. Meanwhile, each function captures a different profile of an analyst satisfying each one's perspective. This allows analysts to share the outcome without the necessity of revealing sensitive information on how they decided on opting for such a profile.

Although the above example is simplified for clarity and space, it represents the complex nature of the industry. Each analyst in the research effort has a number of employees to assist him or her by manually generating a number of charts for each cycle, done on a monthly basis. In addition, mid-year and end-of-year reports to analyze and project the market's directions are required. The problem becomes more critical when an overall picture of the entire market is needed. This requires a number of meetings among all analysts. FUNBASE would decrease the human workload by putting each analyst's knowledge in the system as functions, filters, or profile data. In this way, a single individual would be able to quickly reach a decision. Even for a single analyst who generates data files, as it is currently done, sorting companies by rating (Figure 2.a), by industry (Figure 2.b), etc., it is time consuming. FUNBASE can automatically extract the required data by external criteria inputs via functions manipulating dynamic and static files in the data layer. This obviates the need for an analyst to examine data, but allows the analyst to concentrate on identifying a strategy (e.g., good economy and strong market) which when used will directly achieve the result intended.

The benefits of FUNBASE become more evident in carrying out real scenarios with large numbers of interacting basic factors and inter-dependencies among securities across different industries. In HY markets, information gets reflected every moment in current prices and corresponding yields leading to new selections. Having a responsive methodology for security selection according to pre-existing market outlooks, FUNBASE can rapidly have securities bought and sold accordingly. A more simplified method is currently in use



(a) Actual return by industry

	Company	Coupon	Maturity	Price	Yield To Worst	Actual Return
		BEST RETURNS				
1	Ampex Group Inc	13.250	08/01/99	42.00	NA	13.45
2	International Controls	12.750	08/01/01	80 00	NA	8.11
3	GTech Corporation	14.000	08/01/00	119.00	6.68	7.31
4	Specialty Retailers Inc	14.625	08/15/99	110.00	10.77	7.28
5	Auburn Hills Trust	16 875	05/01/20	130.00	12.88	6.63
6	JPS Textile Group Inc	9.250	06/01/99	93.00	10.73	6.46
7	Greyhound Lines Inc.	10 000	07/31/01	92.50	11.27	6.08
8	Reliance Group Holdings	11.000	11/15/96	94 00	12 89	5.35
9	Trans World Airlines	15 000	07/31/94	80 00	NA	5.26
		WORST RETURNS				
1	Trans World Airlines	14.250	02/01/96	11 00	NA	-42.11
2	Gordon Jewelry Corp	13.750	12/01/96	25 00	NA	-41.86
3	Trans World Airlines	12.000	09/30/08	3 50	NA	-22.22
4	Trans World Airlines	12.000	12/31/01	3 50	NA	-22.22
5	Macy (R.H.) & Co Inc	16.500	11/15/06	6 50	NA	-18 75
6	Insilco Corp	16 750	01/15/01	18.00	NA	-14 29
7	SPI Holding Inc.	14 750	10/01/02	72 00	NA	-12.20
8	Restaurant Enterprises	12 250	12/15/96	83 00	NA	-8 47

(b) Best/Worst Returns

Figure 3: HY market aggregate analysis (Source of First Boston Corp.)

in the equity market in the form of program trading. A mathematical model identifying a simpler two factor relationship. The multi-factor model as in the HY market adds a level of complexity. Here FunBase 's versatility becomes evident in letting users design their own functions, capturing their approach of the multi-factor model and integrating them in FunBase 's function model. This allows users to view FunBase as a corporate resource where data, functions, and results are integrated and accessible.

5 FunBase vs. current DB Systems

In relational databases, two features come in handy for applications like HY markets: tabular form of data and simplicity of use. Unfortunately, the database design cycle and data entry is a time consuming process. Time is a rare commodity in an application like HY market. Meanwhile, the results expected from an analyst require functionality not possible to implement using a relational query. This is due to lack of computational completeness [Atkinson 92]. The flexibility needed in the HY market to extract data via external criteria using functions, while being able to tune them, makes FunBase more favorable for such applications.

One of the important features that gives OODB systems an edge over the relational model is the encapsulation of both program and data. While this has proven to be very useful for CAD/CAM applications, other applications like HY markets need greater flexibility. FunBase shares the idea of integrating functions and data with OODB, but we differ in the ability to query on functions and access them as well as data. In FunBase a user can apply a function of interest to a data set of interest. In OODB, a function (i.e., method) is designed for a particular data only and the concept sharing of knowledge by using that function on other data is not possible. Furthermore, updates and time series analyses are not very clear if implemented in OODB [Snodgrass 90]. Another factor is the form of data used. Although the object model is appealing in some cases, the tabular form is still needed in a variety of applications.

We see FUNBASE as a system not meant to subsume the two other systems, but to address unsatisfied needs for a sector of applications. We also view FUNBASE as a function-based system integrating data and results and not simply a database system.

6 Conclusion

A new type of system, FUNBASE, has been proposed. This system would integrate different levels of information (i.e., data, functions, and results). An example of a sector of domains has been presented, identifying needs that are not met by current systems, but are addressed by FUNBASE. FUNBASE does the following:

- Integrates data with functions and results.
- Provides the facility to query on functions using a function model.
- Accesses data via external criteria.
- Acts as a collaborative medium for function and information.
- Acts as a data acquisition tool for tabular flat files.

The function model in the function layer is currently being implemented. Users can check in new functions and query about existing ones.

We view FUNBASE base as a corporate resource for information access and sharing rather than a data resource for data sharing only, as in databases. Integrating functions and results with available data automates the information availability and advances us to the information era.

References

[Atkinson 92] Atkinson, Malcom., *et al.* (1992). "The Object-Oriented Database System Manifesto." *Building an Object-Oriented Database System - The story of* O_2, Morgan Kaufmann Pub., San Mateo, California, pp. 3-20, 1992.

[CIESIN 91] Consortium for International Earth Science Information Network, (1991). "CIESIN Global Environment Directory: Phase 1, Concept Demonstration Design Document." Technical Report, CIESIN, September 1991.

[Flierl 90] Flierl, Glen, (1990). "Scientific Database Workshop Position Paper." *Computer Science Technical Report No. 90-22*, University of Virginia, Charlottesville, Virginia 22903, August 1990.

[French 90] French, James C., Jones, Anita K., and Pfaltz, John L., (1990). "Scientific Database Management." *Computer Science Technical Report No. 90-22*, University of Virginia, Charlottesville, Virginia 22903.

[Markowitz 52] Markowitz, Harry, (1952). "Portfolio Selection." *Journal of Finance*, 7(1), March pp.77-91.

[Markowitz 87] Markowitz, Harry, (1987). "Mean Value Analysis in Portfolio Choice and Capital Markets." Basil Blackwell, 432 Park Ave. South, Suite 5103, New York, New York 10016.

[Merton 69] Merton R. C., (1969). "Life Time Portfolio Selection under Uncertainty, the Continuous Time Case." *The Review of Economical Statistics*, pp. 247-259, August 1969.

[Snodgrass 90] Snodgrass, Richard (1990). "Temporal Databases - Status and Research Directions." *SIGMOD record*, Vol. 19, No. 4, pp. 82-89, December 1990.

[Zemankova 90] Zemankova, Maria, (1990). "Scientific Database Management and Conduct of Research". *Computer Science Technical Report No. 90-22*, University of Virginia, Charlottesville, Virginia 22903, August 1990.