

Washington University in St. Louis

## Washington University Open Scholarship

---

All Computer Science and Engineering  
Research

Computer Science and Engineering

---

Report Number: WUCS-91-15

1991-02-01

### SAAM: The Strategic Asset Allocation Model

Judy Lewis, Todd Gamble, and John Tai

Asset Allocation has become a dominant factor for investment strategies in recent years. It has found that by holding a strategically diversified portfolio, a high total return on investments can be maintained while, at the same time, reducing portfolio volatility. With recent federal regulations mandating pension investment responsibilities, appropriate asset allocation has become more important than ever. SAAM is a software package specifically designed to be used as a tool to aid the investment professional in determining pension portfolio allocations. SAAM uses the expect system shell CLIPS, has a user-friendly interface, displays output graphically, and runs within the confines... [Read complete abstract on page 2.](#)

Follow this and additional works at: [https://openscholarship.wustl.edu/cse\\_research](https://openscholarship.wustl.edu/cse_research)



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

---

#### Recommended Citation

Lewis, Judy; Gamble, Todd; and Tai, John, "SAAM: The Strategic Asset Allocation Model" Report Number: WUCS-91-15 (1991). *All Computer Science and Engineering Research*.  
[https://openscholarship.wustl.edu/cse\\_research/634](https://openscholarship.wustl.edu/cse_research/634)

Department of Computer Science & Engineering - Washington University in St. Louis  
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

## **SAAM: The Strategic Asset Allocation Model**

Judy Lewis, Todd Gamble, and John Tai

### **Complete Abstract:**

Asset Allocation has become a dominant factor for investment strategies in recent years. It has found that by holding a strategically diversified portfolio, a high total return on investments can be maintained while, at the same time, reducing portfolio volatility. With recent federal regulations mandating pension investment responsibilities, appropriate asset allocation has become more important than ever. SAAM is a software package specifically designed to be used as a tool to aid the investment professional in determining pension portfolio allocations. SAAM uses the expect system shell CLIPS, has a user-friendly interface, displays output graphically, and runs within the confines of a 640K personal computer.

**SAAM: The Strategic Asset Allocation Model**

**Judith Lewis  
Todd Gamble  
John Tai**

**WUCS-91-15**

**February 1991**

**Department of Computer Science  
Washington University  
Campus Box 1045  
One Brookings Drive  
Saint Louis, MO 63130-4899**

**Table of Contents**

ABSTRACT .....	1
1. Introduction .....	1
2. Background .....	2
3. SAA Model .....	2
4. Knowledge Acquisition .....	3
5. Approach .....	4
6. Tool Selection .....	5
7. Implementation .....	6
7.1. Database Access .....	6
7.2. Historic Matching .....	6
7.3. Match Compositing .....	6
7.4. CLIPS Integration .....	6
7.5. User Interface .....	8
7.5.1. Forms Entry .....	8
7.5.2. Graphics Display .....	8
7.6. Additional Functionality and Future Potential .....	8
8. Lessons Learned .....	8
9. Validation and Evaluation .....	9
10. Conclusions .....	10
11. Acknowledgement .....	10
12. References .....	10
APPENDIX A - Figures .....	A
APPENDIX B - User Manual .....	B

# SAAM: The Strategic Asset Allocation Model

*Judith Lewis, Todd Gamble, John Tai*

Washington University

## ABSTRACT

Asset Allocation has become a dominant factor for investment strategies in recent years. It has been found that by holding a strategically diversified portfolio, a high total return on investments can be maintained while, at the same time, reducing portfolio volatility. With recent federal regulations mandating pension investment responsibilities, appropriate asset allocation has become more important than ever.

SAAM is a software package specifically designed to be used as a tool to aid the investment professional in determining pension portfolio allocations. SAAM uses the expert system shell CLIPS, has a user-friendly interface, displays output graphically, and runs within the confines of a 640K personal computer.

## 1. Introduction

Asset allocation has become a dominant factor in the performance of investment portfolios in recent times, and is expected to continue to do so in the future. A basic aspect of current investment strategy is to attempt to offset the the insidious effects of the inflation rate on the the "real" value of the investment return. The conventional solution is to find acceptable ways to cause the capital growth to outpace inflation's diminishing effects. Modern portfolio theory holds that market risk is the only risk that returns extra reward. However, greater levels of risk do not always reap greater rewards. It has been shown that a mix of different investment classes offers some defense against the downside of market corrections. By strategically distributing the portfolio assets among the investment classes of stocks, bonds, and cash, a portfolio manager can achieve a high total rate of return while keeping the portfolio volatility to a minimum. Such a process is referred to as a *strategic asset allocation*.

SAAM is a software package designed to be used as a tool to aid the investment professional in the process of recommending an asset allocation for a client. Specifically, this software is designed to be used in the advising of pension portfolio allocations for customers whose assets range between one million and ten million dollars.

The basis of the design was to make use of an expert system to analyze historical data in light of the current market indicators to determine similar periods in history. Using trends in that parallel period and the knowledge base acquired from interviews with the project team's expert, the software produces a suggested asset allocation which has been adjusted for risk. It's output on average was found to be within the acceptable performance range for an experienced investment advisor. While no strategy can be guaranteed optimal, this software seeks to approximate an optimal asset allocation. Further, the asset allocation suggested by SAAM can be tailored at runtime for a specific client's comfort level in the various asset classes. The output of SAAM is intended to be used as a tool which can be factored into the current portfolio recommendations made by a investment consultant.

The package was designed to be user-friendly in its interface with the professional and to display the output in graphical methods standard to the financial industry. The working of the expert system shell are intended to be transparent to the user, with all interface to the package being via forms windows or graphical windows. The total package was designed to operate within the confines of the the standard 640K personal computer running DOS.

## 2. Background

One of the dramatic lessons learned from the market collapse of Black Monday (October 1987) was the benefit of holding a diversified portfolio. This benefit is based on two key ideas:

- financial markets follow trends, but different classes don't respond in tandem
- different categories of investments have different rates of return

Hence, when a portfolio has a strategic mix of asset classes, downturns in one market area can be offset by gains in another. Further, empirical data demonstrate that, in the case of long-term investment performance, ninety-three percent of the total return was attributed to the asset allocation itself and only seven percent to the specific securities chosen.<sup>1</sup>

So, the important decision, from an investment performance standpoint, becomes the relative weighting given to each asset class in the portfolio. In the strategic asset allocation approach, this weighting is based on the the best risk/return expectation in each asset class. However, just determining the best risk/return ratios is not enough. A suitable mix must be suggested based on the financial objectives of the client and his risk tolerance level. And, it is this risk tolerance factor which is so very important to the asset allocation decision in the case of pension portfolios, the target client sector this software seeks to support.

In 1974, the Employee Retirement Income Security Act (ERISA) put into effect a set of federal guidelines specifying the standards of conduct for managers of employee benefits plans. By these standards, known as the fiduciary responsibility provisions of ERISA, any person with fiduciary responsibility for an employee plan is personally liable to make good any loss to the plan that it suffers due to any breach of these provisions. The provisions require a *fiduciary*, i.e. one who is able to control and make decisions about the plan, to act "with the care, skill, prudence, and diligence" necessary to maintain objectives of preserving the trust of the plan beneficiaries while attaining an adequate return.<sup>2</sup>

Section 404(a)(1)(C) of ERISA requires that the fiduciary diversify the investments of the plan with the express intent to "minimize the risk of large losses."<sup>3</sup> In the case of pension plans in the one to ten million dollar range, the fiduciary responsibility is often held by a sole-proprietor of a business or small board of trustees, the main professional focus of which is not pension management. The ERISA provisions provide the incentive for a diversified approach to their plan's investment strategies, but often existing software services to support such decisions are outside the operating expenses of their plan. This software seeks to fill that need.

## 3. SAA Model

The focus of our project was to develop a software package that would incorporate a database of financial historical data for historical trend analysis and to use an expert system approach to making decisions about an asset allocation based on the knowledge obtained from the expert. The indicators chosen to model the financial realm are

- S&P500 Combined Income and Capital Return
- S&P500 Composite Stock Index
- US Treasury Bonds Total Return
- US Treasury Bonds Index
- Treasury Bill Total Return
- Treasury Bill Index
- Consumer Price Index Return

---

<sup>1</sup> "Putting an Asset Allocation Strategy into Practice", Arthur Zeikel, President, Merrill Lynch Asset Management, Inc., 1988.

<sup>2</sup> "Fiduciary Duties Under ERISA", Donald J. Myers and Michael A. Curto, "Compensation and Benefits Management," Spring 1988.

<sup>3</sup> *ibid.*

- Consumer Price Index
- Discount Rate
- 30-year Treasury Constant

Using these figures, the package calculates the return adjusted for risk for each asset class (cash, stocks, bonds) to develop a base asset allocation. Then, using the expert knowledge built into the system, this base allocation is modified to conform to the constraints of investment that are inherently prudent for the pension plan manager. Since the expert for the project is a "preservation of capital" money manager, these rules are inherently conservative in strategy. The resulting suggested asset allocation is, again, a tool for the pension advisor, and is expected to be further revised by an investment professional to conform to the the plan's current allocation goals and constraints, ie short-term cash needs.

The approach taken in the development of the software was patterned after the expert's own approach of historical search and analysis, and the heuristic modification of existing portfolio allocations. However, with the power of the computer and expert system rules, a much broader base can be sampled in each run than ever could be attempted by a sole investment practitioner for each customer.

#### 4. Knowledge Acquisition

The knowledge base used by the expert system shell was obtained through a series of personal interviews with a financial consultant who has been in pension portfolio management with Merrill Lynch of St. Louis for over eight years. One of the expert's admitted strengths is a conservative, long-range approach to client investing.

Clearly, the knowledge acquisition aspect of this project was a crucial factor in the development of this expert system. The process began with a few short, informal, but informative sessions early in the planning stage to discuss the nature of Computer Science and it's benefits to software support tools. Admittedly, many professionals outside of Computer Science are desirous of bringing the technological advantages of computers into their industry, but are bound by their lack of exposure to Computer Science, especially Artificial Intelligence, when considering what this field can contribute. So, the earliest meetings were learning sessions on the part of both the financial professional and the computer professional. From these early meetings, fundamental financial research material was provided by the expert covering various asset allocation strategies and investment goals. Gradually, out of these early meetings came the formulation of the appropriate set of fundamental issues which is expected to lead to an enduring financial tool. Given too many market variables, the software could not be developed to a successful level of operation by the end of a semester term. If too restricted in its knowledge base, the expert system would not have had enough data to make any worthwhile decisions. Hence, the proper combination of indicators needed to be chosen for SAAM.

Once the issue of market indicators had been resolved, the software professionals needed to become familiar with some of the workings of the market in order to be able to capture the asset allocation principles in code. The expert provided the financial reading material necessary for the software team to become familiar with the financial terms and principles necessary for this type of a project. This allowed the group to make better use of the expert's time during interviews, as the basic financial terminology had been acquired and the overall scenario in which this software would operate was understood.

After an initial working idea had been developed, the interview sessions became a software team effort. The first meeting was held in a tutorial session basis in which the financial expert presented a brief description of her financial background and interests, many of the financial terms and ideas that are important to the underlying basis of the model, and her ideas of what the software package should do to be of value to her. With the expert's permission, the interviews were taped for later review.

This meeting served to introduce the expert to the group and to introduce the group to the fundamental aspects of what the expert's approach to the asset allocation process involved.

Following the first meeting, the interview tapes were transcribed and those transcriptions and the additional financial material provided by the expert were studied. From this stage of research, a model of the software system was developed and specific questions for the next interview were developed.

The second interview was more structured and took the form of a question and answer forum in which specific, detailed questions about the calculations for the software were discussed, the database figures were

reviewed in detail, and some of the higher level rules for heuristic modification of the asset allocation began to take shape. Specific questions had been outlined in advance of the meeting and a detailed model in flowchart form had been prepared for use at that meeting as well. The model was explained to the expert in detail. Based on that high-level working model, the specific questions were asked regarding each aspect of the model.

Again, the second meeting was taped and transcribed. Following the review of the second transcription, the software development process began. To support specific questions that arose during the coding process, one person in the group was assigned to act as interface with the expert to both obtain the information needed to overcome short-term stumbling-blocks, and to keep the expert apprised of progress and problems in the implementation phase.

Because this project was the development of an expert system, the expert knowledge acquisition aspect was considered a very important aspect of the process. Every attempt was made to obtain the information necessary to proceed with the project, while at the same time not burdening the expert with unnecessary meetings. We were fortunate to have had a very enthusiastic and interested expert, as this phase was time-consuming for both sides of the project. The team needed to do outside reading and research to try to come to a better understanding of some of the financial matters attendant to such a decision, and the expert had to do outside preparation to develop the general rules needed by the team for configuration of the expert system shell. As is often the case with engineering and business professionals, knowledge is encapsulated by the professional in the form of personal and professional case histories. But, the software team tended to ask questions to elucidate general philosophies and techniques. Thanks to the interest and willingness of the expert, a number of interesting general-nature rules of financial strategy emerged. As the software group became more knowledgeable about the domain, the questions posed were more effective in obtaining specific rules. Clarifications for calculations using the data came about through simulation and role playing.

Finally, the working prototype was demonstrated to the expert. Her review of SAAM's performance was essential for the final modifications, and her suggestions for future functionality were quite encouraging. Additional functionality will not be discussed here, but the notion of the continuing thread of knowledge acquisition through the various stages of the project development is important to mention. At each stage in the development of SAAM, close communication with the expert played a crucial part in the success of this project. Additionally, the time and effort put into the planning and execution of successful interview techniques can not be overlooked. Successfully managing the relationship with the expert cannot be over-emphasized, since the continued enthusiasm and support given by an interested expert was crucial to the success of the final project.

## 5. Approach

From the knowledge acquisition step it was determined that the expert uses a collection of predictive rules to recommend adjustments to clients' asset allocations. These rules are applied in conjunction with evidence provided by historic cases which resemble the current market. The search for historic time periods is limited by historic data availability and constraints on the expert's time.

The indicators of asset performance in the market for the current time period, where a time period is a *window* which is defined to have a *size* attribute indicating the number of consecutive months contained in the window, are grouped into a reference frame called the *base window*. Since the current data is used for selection of historic periods which match the current market, the basis for selection is the *current base window*. The current base window which is compared with historic data contained in an *historic base window* selected from the historic database. For database reference, date windows also contain a *start* attribute indicating the date of the beginning of the window.

For each base window there is a corresponding *extended window* of the same size showing the performance of the indicators in the following time period. The object of the rule application to the current base window is to determine an optimal asset allocation for the extended window, i.e., the future time period. By searching the available historic data, a *match set* can be constructed which contains the dates of historic periods which closely correspond to the current base window. This allows a complete and impartial search of all available historic data for comparison, providing a more consistent model than that currently used by the expert for historic comparisons. These modifications to the expert's actions extend the functionality of the match step while preserving the basic principle of providing an historic basis for projection. The rules and constraints for selecting historic data for a match can be segregated from the more predictive and case based rules, since they are applied before the predictive rules. The matching rules are termed *rule set 1 (R1)*.



Given the historic data, R1 can be applied to produce the match set containing references to historic time periods similar to the current base window. Using the extended windows from the match set a single *composite window* can be made from all the match set items. The composite window is an approximation for the extended window of the current time period based on historic information. The optimal asset allocation calculated for this composite represents the optimal asset allocation for the extended window based solely on the application of R1.

A number of special case rules based on current (or near current) market information were acquired from the expert. These rules project modifications to current asset allocations from analysis of recent market trends. These rules are grouped into *rule set 2 (R2)*. In order for R2 to be used a base asset allocation is needed. The base asset allocation used is the optimal asset allocation for the current base window. This is easily determined by examining the data within the current base window and assigning assets (within constraint values) to those classes which performed best over the current base window. Once determined, the optimal asset allocation for the current base window can be used, along with recent historic data analysis to apply R2 and produce a suggested asset allocation for the extended window.

Since, two assessments of the optimal asset allocation for the extended window are produced (one from the composite window derived from application of R1, and one from the application of R2), they must be resolved to provide a single, strategic asset allocation for the extended window. The rules in R2 are special case rules derived from the expert's experience and domain knowledge. R1 produces results based on historical data searching. This implies that R2 can contain generalizations of information produced by R1, i.e., R2 is based on experience and R1 deals with historic information. Thus, the information from R1 and the rules in R2 may not be mutually exclusive. The two asset allocations may be averaged to form one asset allocation. The merger of the two asset allocations in this way takes advantage of the predictive capabilities of R2 while tempering its overestimation of special case properties with the results of R1. This method corresponds with the overall philosophy of strategic asset allocation by making recommendations based not only on reaction to current information, but also on a long term study of market performance. It is felt that this most closely corresponds the current workings of the expert. A set of working rules to identify special or extreme cases in the market are maintained by the expert (i.e., R2) and are used to make estimations based on these special cases. Since the use of these rules by themselves would in fact give the expert a kind of "tunnel vision", ignoring the more moderate aspects of the current data, historic data is also used. The major difference between the SAAM working model and the expert's procedure is that SAAM makes a more complete search for historic reference information.

A conceptual model of the system is presented in Figure 0. Note the separation of rule set applications and the data stream between the two. The model also includes features for graphic data display and hard copy output. Additionally, a submodule is included for data expansion so that the tables of market indicators may be kept up to date.

## 6. Tool Selection

After review of the knowledge acquisition and design steps it was determined that a rule based system coupled with some conventional database, graphics, and windowing software would be necessary. The selection of the particular tools to satisfy these needs was governed by their availability, cost, and system requirements.

Because of the expert's need for an economical, PC based application, the expert system shell **CLIPS**<sup>4</sup> was chosen to implement the rule based system. CLIPS is provided with complete source code and allows the integration of user defined functions making it an ideal choice for use in an integrated system. In addition the CLIPS system was designed to be embedded within other software so it is easy to lay a custom user interface and driver routines on top of the kernel. Rules and facts can be read from a file at runtime, allowing limited modification of the system without recompilation.

For the textual part of the user interface, including forms entry and choice selection, **PCcurses**<sup>5</sup> was chosen. It is in the public domain and emulates the UNIX character windowing package *curses*. PCcurses provides a standard windowing interface which is well documented and was assured not to interfere with any graphics packages in use because it used only the text facilities of the system.

---

<sup>4</sup> CLIPS is a product of the Mission Planning and Analysis Division's Artificial Intelligence Section, NASA, Johnson Space Flight Center.

<sup>5</sup> PCcurses was written by Bjorn Larsson.

To provide for the graphics aspects of the user interface the graphics libraries of Turbo C<sup>6</sup> were utilized. CLIPS and PCurses were both obtainable in source code form compatible with the ANSI C implementation in Turbo C. Turbo C's graphics library was also capable of directly addressing the graphics capabilities of the graphics card that was in use on the target machine.

The graphics library provided by Turbo C comes with ample documentation, supporting software, and many basic graphics routines for initialization, borders, bar charts, etc. This made it possible to develop the type of output that the package needed with a reasonable amount of time dedicated to the learning Turbo C's graphics commands. Although Turbo C graphics has its peculiarities, it integrated reasonably well with the other modules in SAAM.

## 7. Implementation

While SAAM is a rule based system developed using an expert system shell, it was impossible, or at least impractical, to implement all of the functionality of the system within CLIPS. Using the tools described above additional code was written to implement the user interface, graphics display and database access functions for SAAM.

### 7.1. Database Access

Since the historic data consisted of uniform records of market indicators cataloged for each month since 1960, the external database of historic facts was stored as a simple binary file. Each record in the file represents one month and holds the values of the indicators for that month. For file access the record location is easily computed by using the year and month as a key to the file. Using this file format, a set of C routines for database access were implemented.

### 7.2. Historic Matching

With the implementation of the historic database in place, routines were produced to implement the historic match phase of SAAM. The rules and constraints within R1 were coded in C, with the appropriate constraints being entered by the user at runtime or derived from default values. There are three indicators used in the search, one each for stocks, bonds, and cash. For each indicator an upper and lower bound for matching is defined to be an *envelope*. Envelope sizes constrain the search by restricting matches to only those values which fall within the envelope. Envelope bounds are described in terms of percent difference from the base value for the match. A normal default for an envelope is plus or minus ten percent. Additional input to the match phase includes the window size and a *roll rate*, which is the number of months to "skip" between start dates when searching the database for matches. The default roll rate is one (1), implying no skip, which searches the complete database attempting to match off every possible start month. The start and end dates for a search are, by default, the first and last entries in the database. The match phase generates a match set which is stored in a temporary file for use by subsequent computation steps.

### 7.3. Match Compositing

Once the matches are found their extended window information is collected from the database and averaged to form a composite extended window. From this composite an optimal asset allocation is computed. Once this allocation is computed the application of R1 is complete and the composite and corresponding asset allocation are retained for later merging with the results of R2.

### 7.4. CLIPS Integration

The application of R2 required the use of CLIPS. Each of the predictive rules provided by the expert for R2 were encoded in the Lisp-like syntax of CLIPS rules declarations and stored in an external file for loading at run time. Facts for the CLIPS system can be asserted internally by the result of rule firings or externally by calls to CLIPS access functions. The initial state of the SAAM system is set by external call by access functions to CLIPS. Values in the initial state include initial values for a base allocation, constraint bounds for asset classes, *trigger facts* for some of the special case rules. Trigger facts are facts which detail conditions of the recent market which must be calculated from the database and are asserted by functions external from CLIPS. Rules in CLIPS may, however, call these external functions in order to initiate the assertion of trigger facts for other rules. A set of housekeeping functions in SAAM initiate the execution of external functions which look for special case rule applications in

---

<sup>6</sup> Turbo C is marketed by Borland International, Scotts Valley, CA.

recent data. These external functions then assert any applicable trigger facts before the execution of the main rule set. What follows is an example case of a rule implementation which requires the use of external functions. The rule is called by the expert the "three steps and stumble rule."

English version of the rule:

If three or more successive increases in the discount rate occur within a six month period, then reduce the stock and bond allocations by five percent each moving the released funds to cash.

CLIPS version of the rule:

; rule 4

```
(defrule rule4 ; "3 steps & stumble, move stock & bond to cash"
```

```
  ?s <- (stock ?x)
  ?b <- (bond ?y)
  ?c <- (cash ?z)
  ?q <- (disc_r_rise_3)
  (stock_min ?a)
  (bond_min ?d)
  (cash_max ?e)
  (test (> ?e ?z))
```

=>

```
(retract ?s ?b ?c ?q)
(bind ?f (min (max (- ?x ?a) 0) 5)) ;get allowable stock, be sure stock>min
(bind ?g (min (max (- ?y ?d) 0) 5)) ;get allowable bond, be sure bond >min
(bind ?h (+ ?f ?g))
(bind ?k (- ?e ?z))
(if (>= ?k ?h)
  then
    (bind ?u (- ?x ?f))
    (bind ?v (- ?y ?g))
    (bind ?w (+ ?z ?h))
    (assert (stock ?u))
    (assert (bond ?v))
    (assert (cash ?w))
    (update_stock ?u)
    (update_bond ?v)
    (update_cash ?w)
  else
    (bind ?u (- ?x (* ?k (/ ?f ?h))))
    (bind ?v (- ?y (* ?k (/ ?g ?h))))
    (assert (stock ?u))
    (assert (bond ?v))
    (assert (cash ?e))
    (update_stock ?u)
    (update_bond ?v)
    (update_cash ?c)))
```

The CLIPS rule depends on the existence of the fact (`disc_rate_rise_3`) in order to fire. When the rule fires the constraints on asset bounds are checked before moving funds from one asset class to another. If the total recommended amount of funds to move would exceed a bound then only the amount within the bound constraint is moved. Once the rule fires the trigger fact is retracted from the CLIPS fact list to avoid repeated rule firings. The predicates such as (`update_cash ?x`) actually are externally declared user functions which update C variables in the main routine to keep a running report of modifications to the asset class allocations.

## **7.5. User Interface**

Two of the main requests from the expert regarding package features were that SAAM be user-friendly, and the the output of the software be displayed in graphical form. The user interface for SAAM was implemented using the PC CURSES packages for entry forms management and the Turbo C graphics library for graphics display.

### **7.5.1. Forms Entry**

Forms were generated for the elicitation of user input of information pertaining to the user profile and constraint values for the system. PC CURSES interacts directly with the C environment and required very little integration with the other tools.

### **7.5.2. Graphics Display**

In an effort to display the data and results in a form familiar to the expert, bar charts were chosen as the preeminent display mechanism. Due to the widespread use of bar charts in the financial industry and their inherent ability to carry much information in a limited amount of space, the bar chart presentation form seemed the mechanism of choice.

Much effort went into planning and implementing the size of each view and the amount of data carried in each multiview window. A "rolling" option was even developed so that a user could roll forward looking at subsequent periods in the database using a roll rate selected by the viewer at run time.

The Turbo C graphics library includes primitives for line drawing and bar chart graphics and its documentation includes many routines for graphics initialization and basic text manipulation. Using these Turbo C features, SAAM's interface includes facilities for display of asset allocations (see Figure 5) and multiview window information. A multiview window graphs the market indicators from the database on a relative scale to the screen displaying each of the three different indicators in a separate graphics window. Each subwindow, or viewport, is indexed by the indicator values and the month of entry (see Figure 4). The height of the bar charts was carefully calculated and the index range modified through repeated trials to make each viewport large enough to adequately represent the data, while at the same time keeping the range narrow enough so that the month-to-month fluctuations in the index values could easily be determined by viewing the graph. An additional display shows a base window and its corresponding extended window side-by-side on the screen (see Figure 6). This is useful for displaying the current base window and the extended window generated from the match phase. The display windows were patterned after similar charts in the Wall Street Journal and leading money manager reports.

## **7.6. Additional Functionality and Future Potential**

In addition to running the asset allocation part of SAAM, the initial menu provides the user with options to scan the historic database and display selected historic periods graphically. This additional functionality was requested by the expert in early development stages for use in reviewing data and client presentation.

Since the development of the prototype, SAAM has been reviewed by the expert. Due to SAAM's success to this point, there are other interesting aspects which could be included if SAAM were expanded beyond this version. The expert has suggested that the display of the rate of return for the suggested asset allocation would be interesting from the standpoint of a customer determining what kind of return his money could experience given that asset mix.

## **8. Lessons Learned**

Thorough planning before coding and modularized approaches to programming responsibility were crucial to the success of this project. Initially, the group worked closely to develop the overall strategy for the entire project. But, the breadth of this project required that as soon as the plan was properly laid out, the responsibility for the code had to be divided among the team members. However, the detailed understanding of the overall project made for

smoother integration later, as the pieces were well designed to fit together in the end.

In SAAM's early versions, a debugging package was included that allowed error messages to be sent to a file called saam.log. These error messages proved to be quite helpful, especially for debugging modules which had previously been run separately. As a result, this trace mechanism was retained as a part of SAAM for later use.

## 9. Validation and Evaluation

The validation and evaluation of SAAM was conducted in a manner loosely conforming to the methodology proposed by O'Leary and others.<sup>7</sup> They define validation as the process of determining the accuracy of the representation of the expert's knowledge within the problem domain. Evaluation is defined as the examination of the effectiveness of the expert system to solve real-world problems, regardless of the expert's ability to function in the same problem situation.

Validation has two aspects -- verification and substantiation. Verification deals with determining if the expert system formulates the real problem in its entirety and produces from this formulation a credible solution within the problem domain. Substantiation deals with confirming that the implementation of the model demonstrates a satisfactory range of accuracy when used in normal application to the problem domain.

For validation the expert was asked to provide a set of real-world examples compiled from her own experience. These examples included her available data at the time and the recommendations that she made for her clients. Of this set, a subset was extracted of situations for which the results of the application of her recommendations were available. This subset of situations was fed into the system. The results were at or above the accuracy level of the expert under the same problem conditions.

Once performance was validated with the specific case examples, the expert was asked to review the conceptual and implementation diagrams of the system. A walkthrough was conducted with explanation of the systems architecture and process structures. The expert agreed that the system architecture and problem encapsulation was a correct and appropriate representation of her domain of concern.

For evaluation, a series of test cases from historical data were generated. For each test case an optimal asset allocation for the period in the future was compared with each of the results from R1, R2, and the results of their combination. The absolute difference of the results for each of the asset classes from the optimal was summed and used later to calculate a mean difference from the optimal. Below is a table representing the results of the evaluation process. The table entries represent the average deviation from optimal for each asset class and the corresponding result class. The result classes (from left to right) are R1 (the asset allocation for the historical composite), R2 (the asset allocation for the base data adjusted for special case observations), and R1R2 (the combination of R1 and R2). Note the variation in the results for the asset class cash. This result has been determined to arise from the fact that none of the special case rules depend on the value of cash performance, but do modify the cash allocation in order to affect the other asset classes. Also, some of the special case rules are allowed to modify the cash allocation beyond the usual constraint when the other asset classes have been performing erratically or have very low return values. The test was run for time period 1978 to 1988, with asset allocation predictions calculated for every quarter.

	R1	R2	R1R2
	-----	-----	-----
stock	6	6	6
bond	6	11	5
cash	13	10	7

Results of Evaluation for 1978-88 on quarter sampling.

---

<sup>7</sup> O'Leary, et. al., "Validating Expert Systems", *IEEE Expert*, (June 1990), 51-58.

## 10. Conclusions

SAAM as a system captures the general philosophy and working theory of the expert's problem domain. The objective of the project was to develop a working system in a limited amount of time. This required the limitation of the scope of the project to ten major indicators of financial market performance for historic reference. The expert tailored her responses and descriptions of decision making processes to cases which only dealt with those indicators. With these limitations in mind, the system performs well under evaluation by the expert. It also performs reasonably well in comparison with the optimal results generated for test data. Although, in this case, it is difficult to supply an appropriate evaluation metric because of the inherent interdependence of the asset classes in the allocation. Based on these initial results, SAAM demonstrates that it has the required basic functionality and working model in place to support additional enhancements and extensions. It is believed that enhancements to the application of R2, particularly in the addition of more market indicators, would improve overall performance.

## 11. Acknowledgement

The authors would like to thank Susan McCann, a financial consultant with Merrill Lynch, for her invaluable assistance to this project. Her interest and enthusiasm in the development of SAAM were, in many ways, as crucial as her expertise.

Support for graduate studies for Judith Lewis was provided by the James S. McDonnell Foundation under Grant 87-24 and the National Science Foundation under grant CDA-9000823.

## 12. References

1. Adeli, Hojjat, *Knowledge Engineering*, 1, McGraw-Hill Publishing Company, St. Louis, (1990).
2. Borland International, *Turbo C, Version 1.5 Reference Manual Set*, 1987.
3. Cromwell, Richard A. and Peters, Edgar E., "The Case for Tactical Asset Allocation", *Structured Investment Products Newsletter*, 16, Dec. 1988.
4. Irgon, Adam, et al, "Expert System Development, A Retrospective View of Five Systems", *IEEE Expert*, (June 1990), 25-40.
5. Lewis, Judith H., Gamble, M. Todd, and Tai, John, "SAAM: The Strategic Asset Allocation Model" submitted to the Midwest Artificial Intelligence and Cognitive Science Society Conference in Jan 1991.
6. Myers, Donald J. and Curto, Michael A., "Fiduciary Duties Under ERISA", *Compensation and Benefits Management*, (Spring 1988), 199-206.
7. O'Leary, Timothy J., "Validating Expert Systems", *IEEE Expert*, (June 1990), 51-58.
8. Zeikel, Arthur, "Putting an Asset Allocation Strategy into Practice", *An Investment Perspective from Merrill Lynch*.

# Appendix A

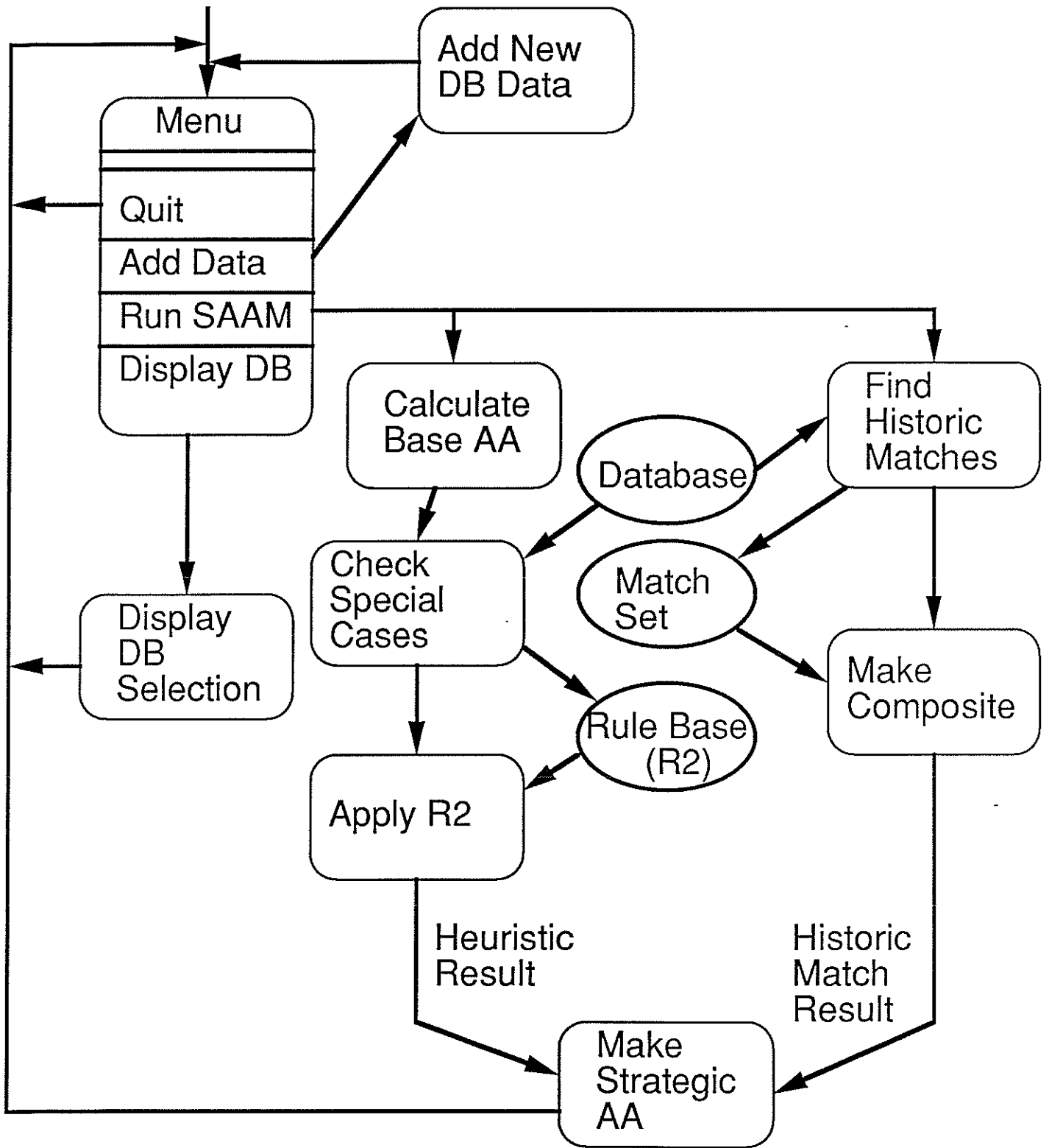


Figure 0



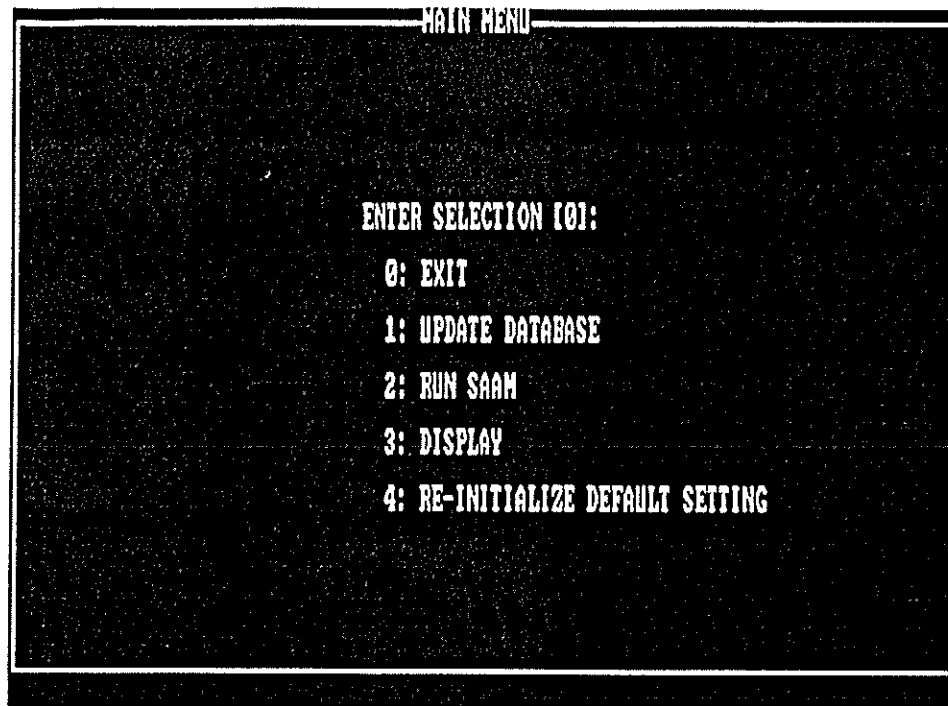


Figure 1

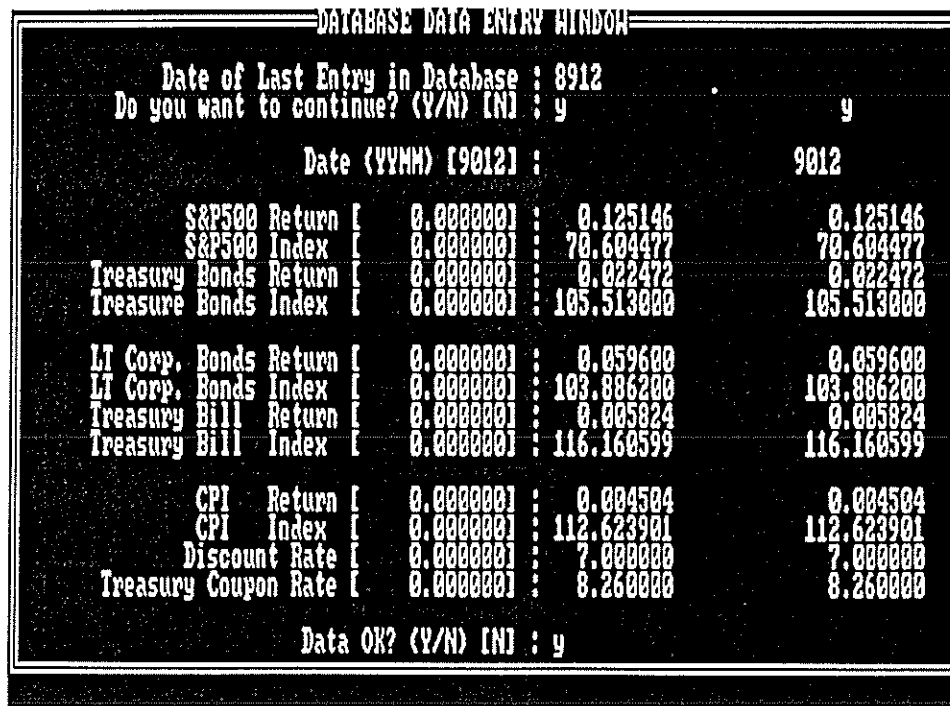


Figure 2

```

SAAM EXECUTION
Current Date (YYMM) [9012] : 8408
Rolling Window Size (No. of month) [ 12] : 12
Comfort Allocation - Stock Max (%) [ 70] : 70
                  - Stock Min (%) [ 30] : 30
Comfort Allocation - Bond Max (%) [ 70] : 70
                  - Bond Min (%) [ 30] : 30
Comfort Allocation - Cash Max (%) [ 30] : 30
                  - Cash Min (%) [ 0] : 0

Envelope Size - Stock Above [0.1000] : 0.100000
               - Stock Below [0.1000] : 0.100000
Envelope Size - Bond Above [0.1000] : 0.100000
               - Bond Below [0.1000] : 0.100000
Envelope Size - Cash Above [0.1000] : 0.100000
               - Cash Below [0.1000] : 0.100000

Data OK? (Y/N) [N] : y

```

Figure 3

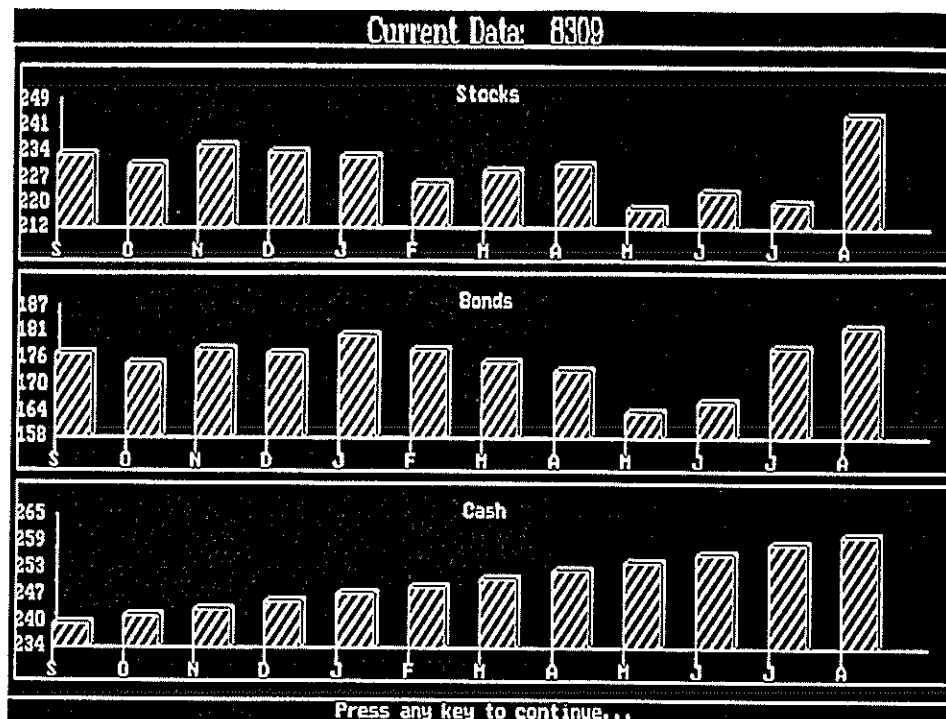


Figure 4

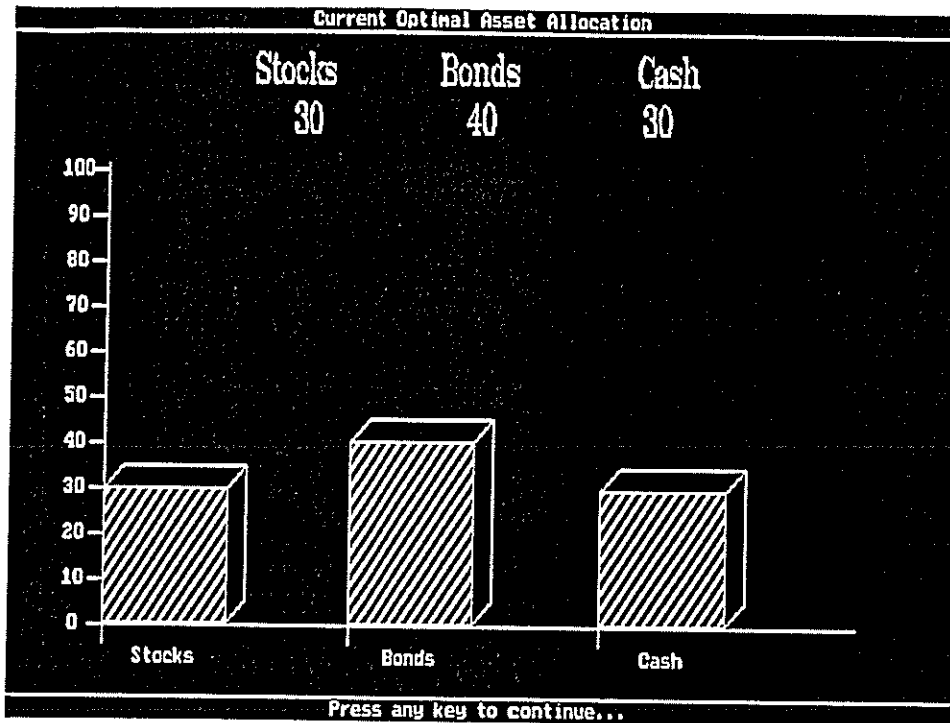


Figure 5

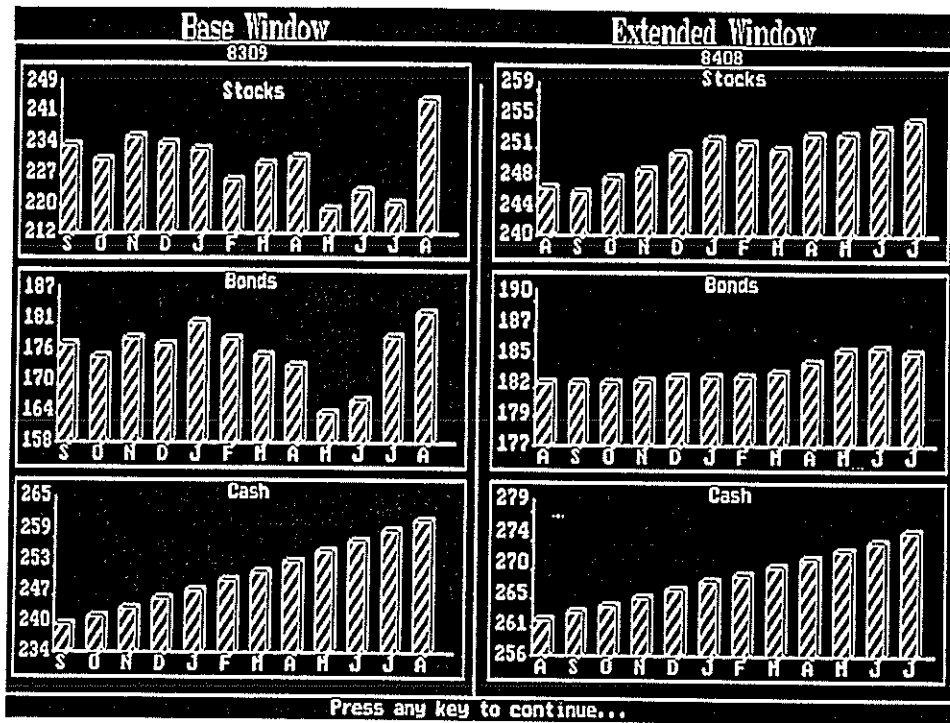


Figure 6

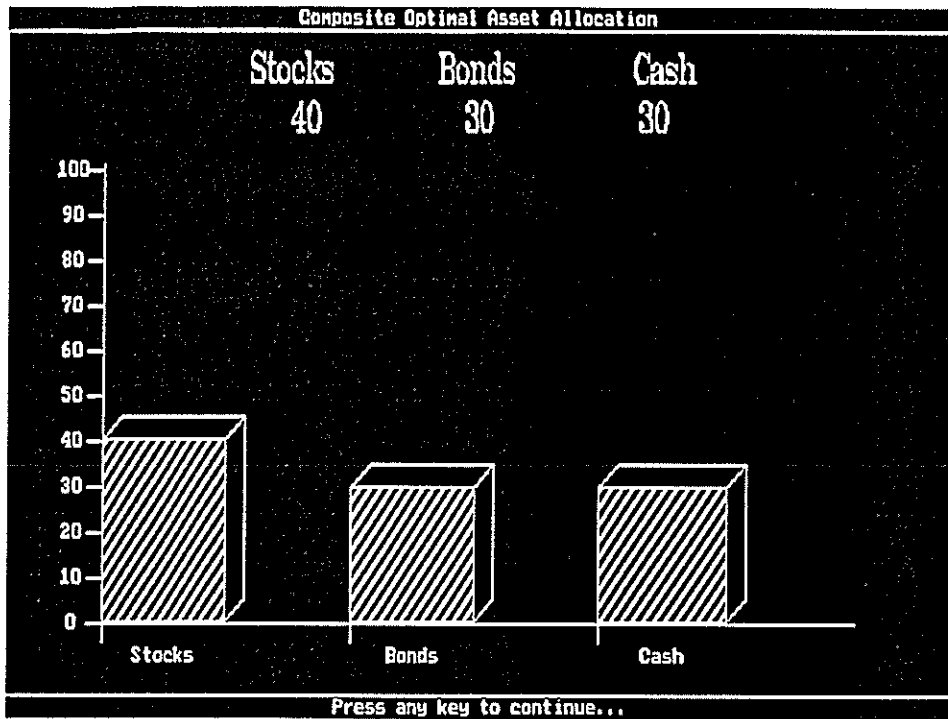


Figure 7

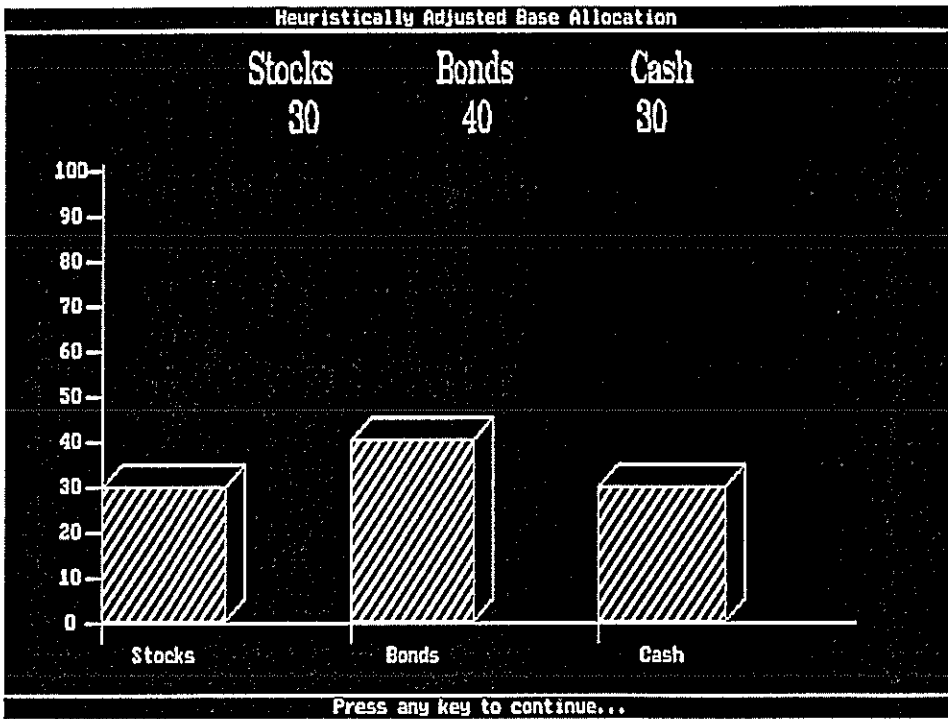


Figure 8

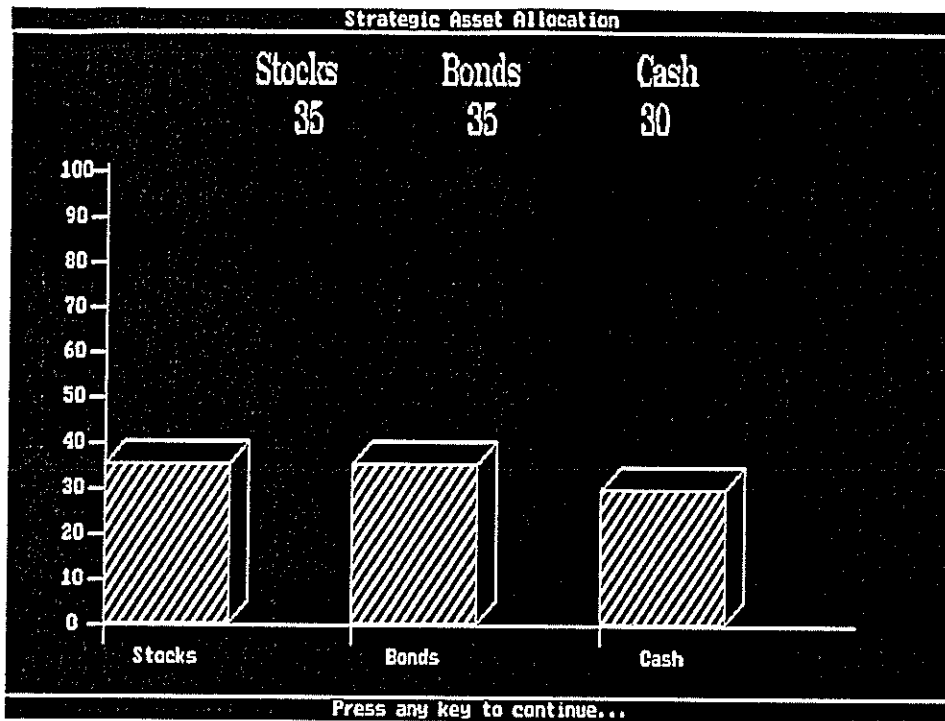


Figure 9

DISPLAY DATA

---

Display Starting Date (YYMM) [9012] :	7500
Roll Rate (no of month) [ 3] :	3
Data OK? (Y/N) [N] : y	

Figure 10

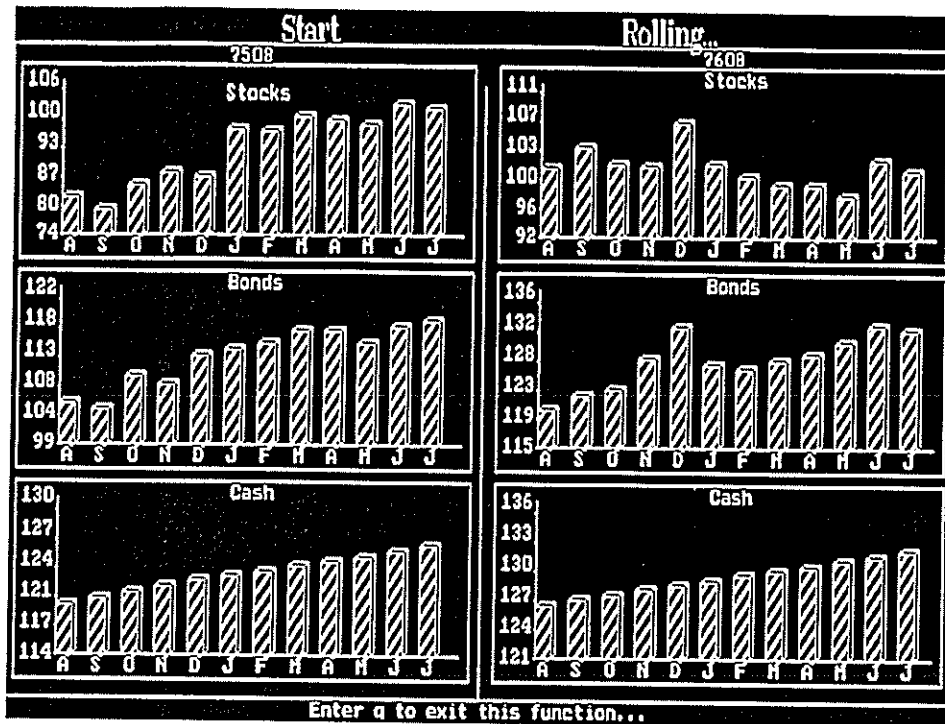


Figure 11

# Appendix B

## USER MANUAL

The following is a description of the functionality of SAAM and is intended to be a tutorial of its use.

### 1. Getting Started

Create a new directory for the SAAM modules. Copy the modules from your floppy disk to the hard drive. Enter SAAM at the DOS prompt, and "Welcome to SAAM!"

You'll be greeted by a short set of graphics screens, and then be prompted to enter into the functional portion. One press of the return key will remove the graphics and introductory messages and you will find yourself in the main menu. (See figure 1.)

The main menu options are clearly marked by their identification index. This manual will cover each option in the order in which they are found on that screen.

SAAM's interface was specifically designed to be as understanding as possible. In test runs, all manner of uncommon data were entered and SAAM responds quite reasonably. In most cases, inappropriate data (like entering your name for the date) will be ignored and the default value will be used. Not so obvious errors will be accepted by SAAM, but you are able to return to the entry of data as often as you like to correct an undesirable input. Just enter the letter "n" or "N", or merely press the return key, at the final prompt, and you will be returned to the top of the screen. SAAM will remember what you typed during the most recent interactive session, so you'll need only correct those fields that you determine to be incorrect. All previously entered values will be retained as the default value, and will be displayed by simply pressing the carriage return key.

Once you are satisfied with your input, enter a "y" or "Y" and SAAM will automatically present you with the next screen for the current function.

So, now let's discuss each option separately. Unless otherwise indicated, each menu will operate in the same manner.

#### 1.1. SAAM's Menu System

##### 0: EXIT

Entering the number for the exit option will terminate the current session with SAAM, and return you to your usual DOS prompt. Thanks for joining us!

##### 1: UPDATE THE DATABASE

This screen is used to enter the indices to the database to keep SAAM's fundamental indicator knowledge up to date. It is designed to handle numerical input for one month's data. The values are intended to be entered as real numbers with upto 6 decimal places. (See figure 2.) The maximum value permitted is 99999.999999 or -9999.999999. Again, SAAM has a good memory; so, if an indicator is mistyped the first time, repeat the entry process before continuing to the next screen. To maintain the integrity of your database, you'll have to be careful about this screen. Your visual inspection is the only data error checking in this version. Once the data is entered, SAAM will remember it that way. Therefore, if the data is entered incorrectly, the incorrect values will be used in processing. There is currently no built-in correction facility. It is suggested that you keep periodic backups of database information upon which you rely. Also, it's important that you enter the data in chronological order and do not leave any gaps between entered months. This screen offers an easy exit feature if you decide that you do not want to do data update. Simply enter "n" or "N" at the continue-decision prompt, and enter "y" or "Y" for correct data. SAAM will return you to the main menu without having altered the database.

##### 2: RUN SAAM

The following option contains most of the functionality of the asset allocation services of SAAM. (See figure 3.) The following is a description of the input required:



- **date** : either today's date for a current run or a date in the database for historical review. Of course, today's date is set as the default. Note that SAAM accepts dates in the following format, YYMM. Therefore, January 1989 is entered 8901. Acceptable dates are strictly a month followed by year, hence they are four digit integers.
- **rolling size** : sets the granularity of SAAM's search through the database in the matching phase. This input is intended to be an integer value in months (maximum value, 12).
- **comfort levels** : indicate to SAAM your maximum and minimum allowances for allocations in each asset class. The final allocations suggested will be calculated based on these ranges.
- **envelope sizes** : these values are intended to be entered as real numbers less than one. They represent how closely historical indicators must come to the current values before SAAM will consider them a "match". For example, +5%,-5% for the stock envelope is entered: stock above 0.05 and stock below 0.05. Too small a value will result in too few matches, conversely too wide an envelope (ie, a value too close to 1) will result in too many matches. In either case, SAAM will prompt you about the condition. You can return to the menu and modify your original input. SAAM, of course, will have remembered your last input value for you.

If you are satisfied with your input, enter a "y" or "Y" and let SAAM go to work!

The first output from SAAM will be a graphical depiction of the values for the stocks, bonds, and cash indices over the past base period. (See figure 4.) You stipulated the size in months of this base period when you entered your choice for the rolling size in the menu for this option. These values are displayed in bar chart form. Their absolute values are displayed along the vertical axis. Months are abbreviated with the first letter of the month's name and appear along the horizontal axis. Pressing enter when ready will allow you to proceed to the next stage of SAAM's analysis.

SAAM will display in bar chart form what would have actually been the optimal asset allocation for the base time period in this screen. (See figure 5.) The base time period was displayed in the first graph. Again, pressing enter allows you to proceed to the next screen.

SAAM then displays an informative screen to let you know that the historical data base is currently being processed for periods in history which match the current data. A return value is considered a match if it falls within the range you specified for the index in menu 2 (Figure 3). These informative screens may appear periodically during execution. As discussed earlier in this manual, these informative screens will let you know about certain stages in SAAM's processing or changes you need to make to your selections in order to obtain more informative output.

From the matches found, SAAM makes a composite by averaging of the extended time periods found and creates an extended (predictive) view based on the trends established during those matching periods. The base window is then displayed in the left hand side of the multiview window; the right hand side represents what SAAM expects the following period to look like for those asset classes, based on historical trends established by the data-base search and processing. (See figure 6.)

A composite asset allocation is then calculated by SAAM and displayed for you in graphical, bar-chart form. (See figure 7.) This is the expected optimal allocation based on the historical rule search only.

The next screen depicts the asset allocation as heuristically determined from the current base window. This represents the expert rule set modification of the base asset allocation. (See figure 8.) Again, to continue to the next screen, simply press the enter key.

The final graph, labeled the "Strategic Asset Allocation", is the final suggested allocation based on the combined information gleaned from the database search combined with the expert knowledge built into the system. (See figure 9.) It is this final allocation which is expected to most closely match the market's performance.

### 3: DISPLAY

This option is intended to provide a multiview window for the purpose of tracking long stretches of time across the database. (See figure 10.) As it was expected to be used as a comparison feature, two time periods are

displayed in the multiview format. (See figure 11.) The time period on the left is the current time period, a twelve-month display. The view on the right half of the screen will "roll" at the rate you set. The default value is again quarters (3 months), but you can watch any rate between 1 and 12 months by so setting that integer figure in the roll rate field of menu 3. The rolling will continue displaying the next roll period until you enter a lowercase 'q' (to "quit") followed by pressing the enter key. (Note: A rolling display that starts within 12 months of the last entry in the database will clearly not have any data to display on the right side of the multiview. Hence, SAAM will indicate this by displaying empty graphs with the empty "9999" date.)

Attempts were made to be as agreeable as possible in this rolling function. If you enter a start date that is prior to the earliest date in the date base, SAAM will begin the graphical display at the first database entry (currently 6001). If you enter a date that is not yet entered in the database, SAAM will deposit you back at the main menu. Any date in between will roll at the desired rate until you reach the end of the data base. As you approach the end of the data, you will notice that fewer than twelve months appear in the multiview on the right. When the last of the data is displayed, SAAM will return you to the main menu.

#### **4: REINITIALIZATION**

This final option is a convenience feature that will automatically return SAAM's values to their default state. If after a few trial runs you want to return to the original default values, enter 4. The menu will not change, but SAAM's values will have been reinitialized.

Again, to terminate the session, enter 0 or simply press the return key.