IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4930

A Framework for Real Time Decision Support

Chiu-Che Tseng
Department of Computer Science and Information Systems, Texas A&M University-Commerce, P.O. Box 3011, Commerce, TX 75429, chiu-che_tseng@tamu-commerce.edu

ABSTRACT

We describe our real time decision support framework; a system that provides decision support for various domains. The challenge of the decision support is that a large amount of diverse information can be potentially relevant to a decision, and that, frequently, the decisions have to be made in a timely manner. This presents the potential for better decision support, but poses the challenge of building decision support system for timely decision support. The decision models our system uses are implemented as influence diagrams. Using a suitable influence diagram, our system evaluates the influence diagram producing the decision recommendations.

INTRODUCTION

The goal of decision-making is to select an optimal action that satisfies the decision-maker's objective, or, in other words, to maximize the returns under the constraints given by decision-maker. Time is frequently an important factor in the real time domain. It is sometimes impossible for a decision-maker to utilize all the available information and come up with a decision in a timely fashion. Therefore, a control mechanism is needed to help the system balance between deliberation and timely decision-making.

Our system uses a decision model to produce investment recommendations. Our system is implemented with an Object Oriented Bayesian Knowledge Base (OOBKB)[16,23]. It contains the domain knowledge expressed in a set of classes hierarchically organized by the "subset" relation. The OOBKB can create a decision model, in this case an influence diagram, on the fly on different levels of detail. Our system uses the current model to compute which information sources should be accessed, deploys web agents for information gathering, solves the model for the optimal investment recommendation given the acquired information, and uses a user interface to communicate the result to the human user (See Figure 1).

We incorporate the notion of urgency into our system in order to determine how much detail the model should contain, and how much information we can gather. The system first assesses the urgency of the decision situation that the human investor currently is in, and then determines the right level of detail at which to instantiated the model. Based on the decision model and the urgency, our system then allocates the computational resources to perform the information gathering and to solve the influence diagram. In essence, then, the system uses the notion of urgency to trade off the value of computational time in urgent situation for the quality of the results obtained.

In the following sections of the paper, we first introduce our system's architecture and describe other components of our system. We then concentrate in detail on the OOBKB component, and show how the decision model can be constructed from the OOBKB. We follow by describing our definition of the notion of urgency and how it applies to our system. We give examples of how our system works under urgency, and describe how the resources are allocated to computation and information gathering. We end with conclusions and further research directions.

ARCHITECTURE OF THE FRAMEWORK

The architecture for the real-time decision support framework is based on the components shown in figure 1: an object-oriented Bayesian knowledge base, a decision model, a control module, an executor and an interface. These components work together to provide the basic functionality of our system. The components of the system are:

- Object Oriented Bayesian Knowledge Base - contains the object-oriented domain information, such as companies, information sources, users, etc. and the Bayesian information such as quanti tative, conceptual and structure information.
- Decision model - contains the influence diagrams created from the knowledge base; it represents the relevant factors of the investor decision model together with their probabilistic rela tionships.
- Control module - performs runtime control of our system
- Executor - performs actual information gathering actions by sending out web agents to gather the most valuable information from the available sources.
- Interface - provides communication with the human user.

  In the following sections, we will describe each of the components in detail.

INTERFACE AND EXECUTOR

If needed our system can send out information gathering agents to retrieve information relevant to decision-maker's situation.

The executor module contains the retrieval agents that are used by our system to get the information from the sources. The agents are implemented with AgentSoft's LiveAgent Pro toolkit. These web agents are responsible for generating the visual reports from their information gathering results. The executor module then sends the report generated from the retrieval agents to the interface module (see Figure 2). Apart from being displayed for the user, the gathered information is also by the system to provide an updated investment recommendation. Our system employs a myopic sequential information gathering strategy [25], according to which we rank our information sources by the value

Figure 1: Architecture of the framework.

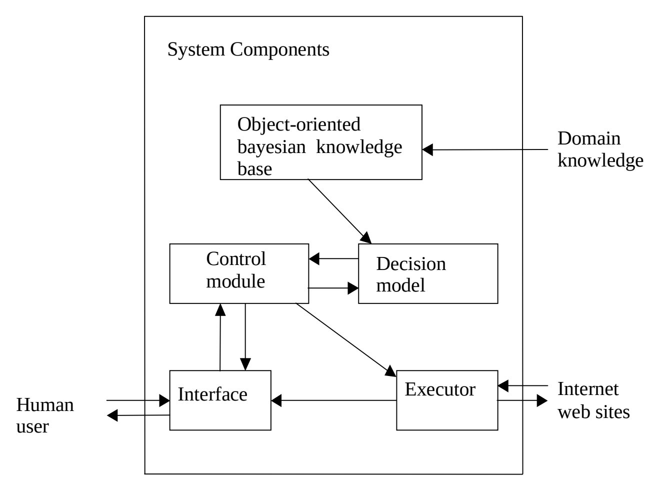
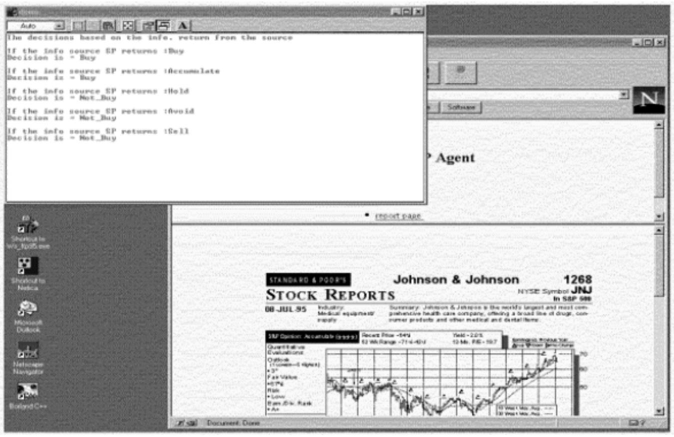


Figure 2: An executor retrieved a stock report from Standard



of information they can provide. By applying this strategy, we can ensure that our system is getting the most valuable information first, which in our domain is the information from the most reliable and informative information source.

The interface module handles the interaction between the human user and the system; the module displays the information gathered by the executor module, and displays the decision suggestion from the system.

OBJECT ORIENTED BAYESIAN KNOWLEDGE BASE

In real time decision support domain, problems are usually complex and incorporate many different relevant factors. To handle the complexity issue, we created a hierarchical Object Oriented Bayesian Knowledge Base (OOBKB) [16,23].

The Object Oriented Bayesian Knowledge Base (OOBKB) is the heart of our system - it stores and organizes the domain information. The domain information in the OOBKB is organized into hierarchy of classes, which represents the generalization to specialization of the concepts in our domain (see Figure 3). Since some of the values of the attributes of the instantiations of classes are not known with certainty, we use them as chance nodes an influence diagram. Thus, the OOBKB contains the probability and casual information (see Figure 4), from

Figure 3: The class hierarchy of the OOBKB in a simple financial domain.

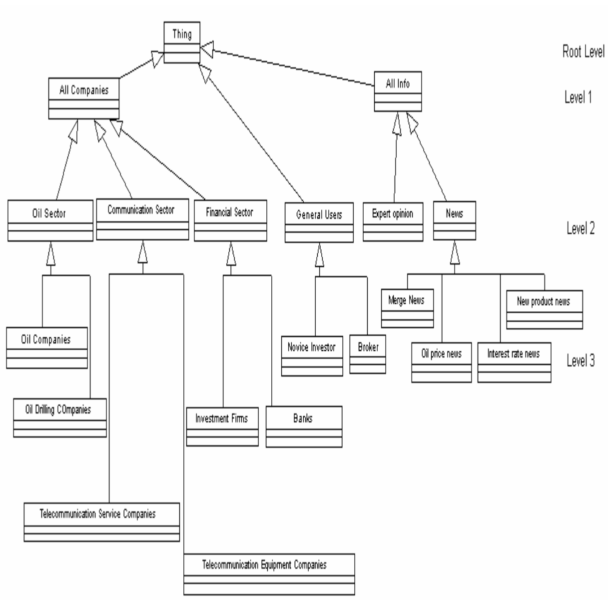


Figure 4: The classes instantiated detail from OOBKB

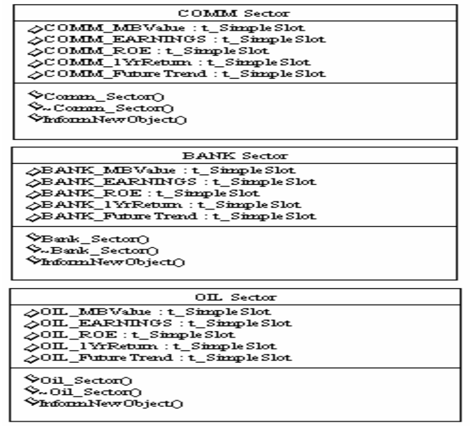
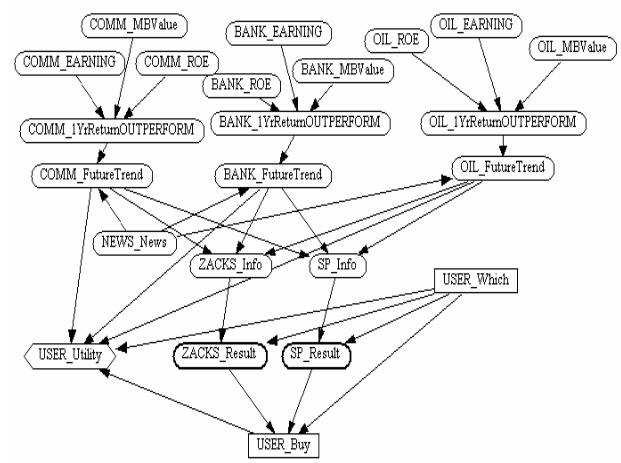


Figure 5: Decision model created from level two classes.



which we can derive and create influence diagram on the fly. Since our OOBKB organizes the classes in a hierarchical order, we are able to create influence diagrams on different levels. The different level of instantiation represents the decision model from abstract to detailed. The more detailed the decision model the more nodes are explicitly represented within the influence diagram.

The level of detail for the decision model is controlled by the urgency factor. That takes into account the computational cost and the information gathering cost. Briefly speaking, the system first calculates the urgency based on the current information, and then uses that information to decide how detailed the decision model should be. An example instantiation using the classes on the second level of abstraction in our investment domain is depicted in Figure 5. We will describe the urgency calculation in more detail in section 6.

The OOBKB can be created and updated offline to provide up to date representation of the domain. This can take the computational burden out of runtime, thus increasing the performance of our system. The learning process can include updating of the conditional probability tables (CPTs) and prior distributions in each class.

DECISION MODEL – INFLUENCE DIAGRAMS

Influence diagrams are directed acyclic graphs with three types of nodes – chance nodes, decision nodes and utility nodes. Chance nodes, usually shown as ovals, represent random variables in the environment. The decision nodes, usually shown as squares, represent the choices available to the decision-maker. The utility nodes, usually of diamond or flattened hexagon shape, represent the usefulness of the consequences of the decisions measured on a numerical utility scale. The arcs in the graph have different meanings based on their destinations. Dependency arcs are the arcs that point to utility or chance nodes representing probability or functional dependence. Informational arcs are the arcs that point to the decision nodes implying that the pointing nodes will be known to the decision-maker before the decision is made.

The decision model coordinates with the control module in order to provide the sequential information-gathering plan for the executor to implement if needed.

CONTROL MODULE

The control module produces the sequential information gathering strategy and performs the runtime control of our system. The system employs a myopic sequential information gathering strategy, through which we rank the information sources by the value of information they can provide. By applying this strategy, we can ensure that the system is getting the most valuable information, based on the reliability of the information sources. The system takes the cost of the information gathering into consideration. The cost includes both the monetary cost (cost of accessing the information) and cost of time. The monetary cost is the fee for the web agent to access certain information site. The runtime control function of the control module is used as an action controller of our system.

URGENCY

In real time domains, timing is a critical element when making decisions. Using up valuable time on creating a more detailed model and rendering decision from it might not be worth it because the opportunity might have already passed. More succinctly, the probability of losses due to inaction creates urgency.

We defined the urgency or the value of time as the following:

Definition: The urgency, URG(t), is the value of one time unit and is defined as the difference between situation values from two time periods. A situation value can be defined differently for different domain problems. Usually represents using expected utilities or some other measurements.

Take stock market for example; in this case we defined the urgency using the overall market movement and our portfolio's movement at time t:

$$URG(t)=\max(0, overall\_stock\_trend_{t1}) - our\_portfolio\_trend_{t1} \qquad \text{- (1)}$$

where zero represents the riskless asset (usually cash, assuming no inflation).

The stock trend is defined as the overall rate of the stock market movment:

$$Overall\_stock\_trend_{t1} = \frac{(overall\_stock\_index_{t1} - overall\_stock\_index_{t0})}{(t1 - t0)} \qquad \text{- (2)}$$

And our portfolio trend is defined as our current portfolio's overall movement:

$$\frac{(our\_portfolio\_index_{t1} - our\_portfolio\_index_{t0})}{(t1 - t0)} \qquad \text{-(3)}$$

Thus, if our current portfolio consists of cash only then the trend is zero.

For instance, if the overall market is going up at time t but our portfolio exhibits a downward trend, the urgency, URG(t) will be a large number indicating that the investor has to act fast in order to prevent further losses. But if the overall market is going down at time t and our portfolio is going down as well at a lesser rate, the URG(t) will be the difference between the riskless asset (cash) and our portfolio's trend at time t. In this case, even though our portfolio is better off than the overall market, we are still facing an urgency to adjust our portfolio and to convert to cash as quickly as possible.

Clearly, the fact that the time is valuable forces agents to be time effective in executing external actions such as information gathering, and crucially impacts the viability of non-physical actions such as creating and computing the model. The most important non-physical action that the urgency of the situation could make ill advised is, of course, the agent's reasoning, and, in particular, modeling.

TRADING OFF TIME FOR DETAIL DURING MODELING

We use a simple investment portfolio example to demonstrate how our system trades off computational time for details included in the decision model. Our example OOBKB in Figure 3 contains the domain information consisting of three industrial sectors, user information and external information sources. The three industrial sectors have sub-classes denoting different companies within each sector. The user class is further derived into two sub classes: expert and novice user. Each class contains specific information about the user, such as risk preference, etc. The external information class is divided into two sub classes: news and expert opinions. News represents the market news, such as inflation, and economical figures released by the government, etc. The expert opinions represent the opinions on the stocks from different investment firms' experts that are posted on the web.

We first calculate the urgency for the current situation using the formula (1) defined in the previous section. We then apply the urgency result to compare the benefits of using the more detailed model to the cost of time required to run it.

For example, if the value of time, i.e., the urgency, is high, then creating an abstract level decision model (see Figure 6) is preferable. In this case, the system provides the investor with abstract advice, like to buy or sell certain sectors. The investor was given not very detailed advice, since it was important to make a decision fast.

If the situation is not as urgent, then creating a more detailed decision model (see Figure 7) is preferable. In this case, the decision model will contain more information than the abstract model. The model contains extra information about different type of investors, individual company information and different type of news information. From which the system will provide more refined and detailed recommendations.

We tested our system on actual stock market data. For experimental runs, we selected 12 companies from SP500 company listing. We divided the companies into three sectors, communication, banking, and oil production sector. We used the companies' financial ratio data from 1993 to 1996 as our training data set and the 1997 data as our test set.

We first calculated the urgency of the investor's situation by using the stock price data from January 6, 1997. We obtained the open, close, high and low price for SP500 index at that date. For our example, we use the differences between the opening and the closing price of that date and divided with the number of seconds within the trading day to obtain

Figure 6: Abstract decision model creates from level 2 classes in OOBKB.

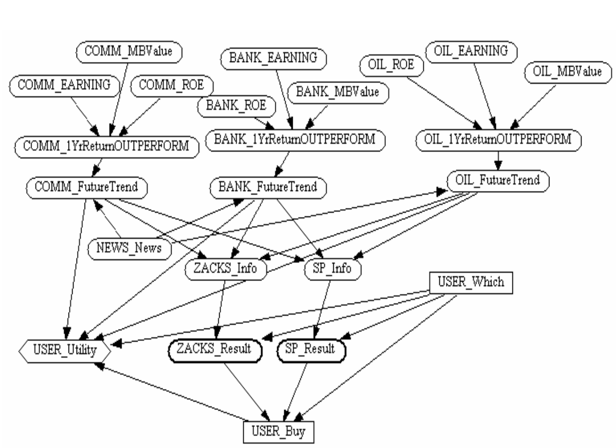
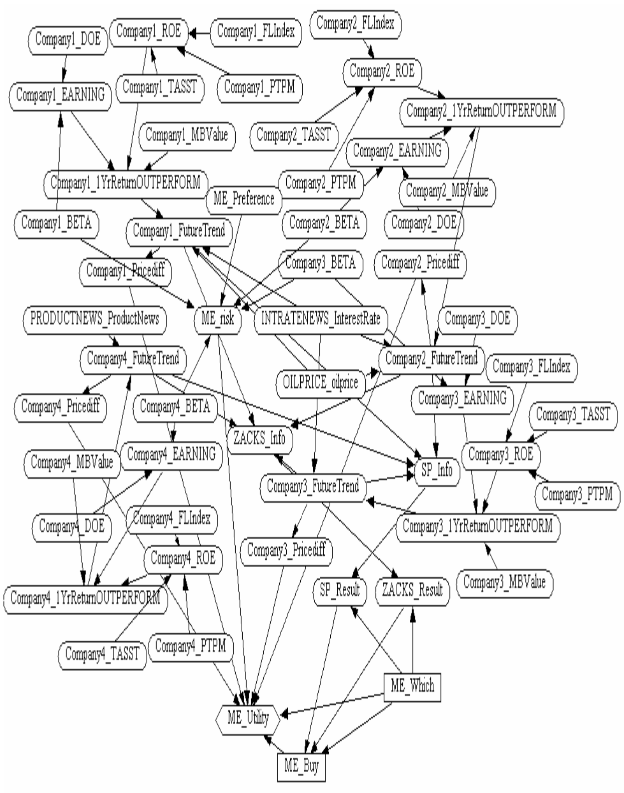


Figure 7: Detailed decision model creates from level 3 classes in OOBKB.



the overall market trend in seconds and our current portfolio consists of cash only. Based on these assumptions, we then calculate the urgency using formula 1:

$$\begin{aligned} URG(t) &= \max(0,\ (748.03-747.65)/25200) - 0 \\ &= 1.5 \times 10^{-5} \text{ point/per second} \end{aligned}$$

The above figure is the value of time (it would also be called the opportunity cost in economics literature in this case) in points per second, for investor being fully invested in cash while the overall market is going up.

We now need to evaluate the cost of running different models in terms of run time, and in terms of points. During our example runs, we calculated the average runtime of two models (See Figure 6 and 7) created on the second and third level of the OOBKB hierarchy, respectively. As expected, the demands of the more detailed model required more computational time. Here, the runtime is measured on an Intel Pentium II 400MHz machine using Neticaâð as our inference engine (see Table 1).

On the abstract decision model (see Figure 6), our system recommended not to consult any external information source and selected the communication sector as the one to invest in. We averaged the one-year total return on the four companies within the sector and obtained the average return of 26.59%. The detailed decision model, in Figure 7, returned the recommendation of not getting any external information source either, and not buying the first company out of four available in

Table 1: Runtime of the two models.

Decision Model	Average Run time (10 runs)
Abstract decision model	0.25 seconds
Detailed decision model	0.83 seconds

Table 2: Performance of the two decision models.

Decision Model	One year total return
Abstract decision model	26.59
Detailed decision model	52.83

this sector. From this more detailed recommendation, we assume that the investor purchased the other three companies in the communication sector and obtained an average return of 52.38%. Here is the comparison of the performance using one-year total return as criteria (see Table 2).

The annualized returns above, converted to return obtained per unit time (second, in our example) yield 3.94×10^{-5} and 7.83×10^{-5}, for abstract and detailed decision models, respectively. From the URG(t) and the runtime of the models, we calculate the loss due to the computational time used for each model. For the abstract model, loss per second is 3.7×10^{-6} points, and for detailed model it is 1.24×10^{-5} points. Subtracting the cost and gain figures results in 3.57×10^{-5} and 6.59×10^{-5} for abstract and detailed models. Thus, our example computation suggests that the more detailed model is more beneficial, and it is worth the computational time given the urgency of the situation in this case. But, if another computing platform were to be used (say a Pentium II system), the computational time for the more detailed model would make it less preferable, and the system would choose to deliver a faster but more abstract investment recommendation.

CONCLUSION AND FUTURE WORK

In this report, we have presented a framework for using Object Oriented Bayesian Knowledge Base to aid the investor in a time critical situation. In our approach, the agent's knowledge is represented as an influence diagram created from the different levels of the OOBKB. The agent can use this model to gather extra information and make decision recommendations to the investor.

We showed how the important notion of urgency arises and can be used in our approach. Urgency is the value of time, and has the intuitive property of favoring immediate actions, sometimes making computational actions, such as expanding the model and information gathering, ill advised.

In our future work, we will refine the urgency definition to include more realistic factors for the investment domain and the information value definition for other types of information sources. We will also develop a suitable learning process for the OOBKB concentrating on the model refinement and sensitivity analysis.

REFERENCE

Algoet, P. H. and Cover, T.M. Asymptotic optimality and asymptotic equipartition properties of log-optimum investment. The Annals of Probability, 16(1)(1988), 876-898.

Campbell, J.Y., Lo, A.W. and Mackinlay, C. The Econometrics of Financial Markets. Princeton University Press, 1968.

Cover, T.M. Universal portfolios. Mathematical Finance, 1(1)(1991), 1-29.

Cover, T.M. and Gluss, D.H. Empirical Bayes stock market portfolios. Advances in Applied Mathematics, 7, (1986), 170-181.

Cover, T.M. and Ordentlich, E. Universal portfolio with side information. IEEE Transactions on Information Theory, 42(2)(March 1996), 348-363.

Gorry, G. and Barnett, G. Experience with a model of sequential diagnosis, Computers and Biomedical Research, 1968.

Gmytrasiewicz, P.J. and Durfee, E.H. Elements of a Utilitarian Theory of Knowledge and Action, IJCAI, 1993, pp. 396-402.

Heckerman, D., Horvitz, E. and Middleton, B. An Approximate Nonmyopic Computation for Value of Information, IEEE Transaction of Pattern Analysis and Machine Intelligence, 1993.

Horvitz, E. and Barry, M. Display of Information for Time-Critical Decision Making, Proceedings of the Eleventh conference on Uncertainty in Artificial Intelligence(1995), Morgan Kaufmann, 296-305.

Howard, R.A. Information value theory, IEEE Transactions on Systems Science and Cybernetics, 1966.

Howard, R.A. From Influence to Relevance to Knowledge, Influence Diagrams, Belief Nets and Decision Analysis(1990), 3-23.

Howard, R.A. and Matheson, J.E. Influence Diagrams, The Principles and Applications of Decision Analysis: Vol. II (1984), Strategic Decisions Group.

Jensen, F.V. An Introduction to Bayesian Networks. Springer-Verlag, New York, NY., 1996.

Jensen, F.V. and Liang, J. drHugin A system for value of information in Bayesian networks, Proceedings of the 1994 Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems(1994), 178-183.

Kalagnanam, J. and Henrion, M."A comparison of decision analysis and expert rules for sequential diagnosis", Uncertainty in Artificial Intelligence 4, 1990, pp. 271-281.

Koller, D. and Pfeffer, A. Object-Oriented Bayesian Networks. In Proceedings of the 13th Annual Conference on Uncertainty in AI (UAI 97), (August 1997).

Matheson, J.E. Using Influence Diagrams to Value Information and Control, Influence Diagrams, Belief Nets and Decision Analysis(1990), 25-48.

Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference Revised Second Printing (1997), Morgan Kaufmann.

Russell, S.J. and Norvig, P. Artificial Intelligence, A Modern Approach, Prentice Hall, 1995.

Samuelson, P.A. Lifetime portfolio selection by dynamic stochastic programming. Review Econom. Statist. 51 (1060), 239-246.

Shachter, R.D. Evaluating Influence Diagram, Operations Research, 1987, 871-872.

Sycara, K.P. and Decker, K. Intelligent Agents in Portfolio Management. Agent Technology, Springer-Verlag, 1997.

Suryadi, D. and Gmytrasiewicz, P.J. Learning Models of Other Agents using Influence Diagrams, In Proceedings of User Modeling: the Seventh International Conference, (1999).

Trippi, R.R. and Lee, J.K. Artificial Intelligence in Finance and Investing. Irwin, 1996.

Tseng, C.C. and Gmytrasiewicz, P.J. Time Sensitive Sequential Myopic Information Gathering. In Proceeding of Hawaii International Conference of System Science (HICSS32)(1999).

Zilberstein, S. and Lesser, V. "Intelligent information gathering using decision models", Technical Report 96-35, Computer Science Department, University of Massachusetts, 1996.

Related Content