Real Time Decision Support System for Portfolio Management

Chiu-Che Tseng
Department of Computer Science Engineering
University of Texas at Arlington
300 Nedderman Hall
Arlington, TX. 76011
817-272-3399
tseng@cse.uta.edu

Piotr J. Gmytrasiewicz
Department of Computer Science
University of Illinois at Chicago
851 S. Morgan Street
Chicago, IL. 60607
312-335-1320
piotr@cs.uic.edu

Abstract

We describe our real time decision support system; a system that supports information gathering and managing of an investment portfolio. Our system uses the Object Oriented Bayesian Knowledge Base (OOBKB) design to create a decision model at the most suitable level of detail to guide the information gathering activities, and to produce investment recommendation within a reasonable time. To determine the suitable level of detail we define and use the notion of urgency, or the value of time. Using it, our system can trade off the quality of support the model provides versus the cost of using the model at a particular level of detail.

The decision models our system uses are implemented as influence diagrams. Using a suitable influence diagram, our system computes the value of consulting the various information sources available on the web, uses web agents to fetch the most valuable information, and evaluates the influence diagram producing the buy, sell and hold recommendations.

1. Introduction

The goal of investment decision-making is to select an optimal portfolio that satisfies the investor's objective, or, in other words, to maximize the investment returns under the constraints given by investor's constraints. The investment domain contains numerous and diverse information sources, such as expert opinions, news releases and economic figures, etc. This presents the potential for better decision support, but poses the challenge of building a decision support agent for accessing, filtering, evaluating, incorporating information from different sources, and for making final investment recommendations. Time is frequently an important factor in the investment domain. It is sometimes impossible for an investor 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 agent balance between deliberation and timely decision-making.

The investment domain, like many other domains, is a dynamically changing, stochastic and unpredictable environment. Since there is no way to correctly predict the movement of stock prices, we use a probabilistic prediction system. A good prediction system uses existing information about the stock market conditions in the past, and uses these to probabilistically predict the future market movement.

Our system uses a decision model to perform information gathering and to produce investment recommendations. Our system is implemented with an Object Oriented Bayesian Knowledge Base (OOBKB)[19,26]. 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.

Our design allows the system the advantage of making investment recommendations based on the newly available information in real time. For example, experts' opinions can be incorporated with their reliability parameters as they are posted on the web, as can the news of, say, interest rates, and how they reflect on the vulnerability of the stock price to the news item. Being able to incorporate real time information sources into our system increases its accuracy and responsiveness.

For portfolio management, there is related work by Sycara, et al. [25] that focused on using distributed agents to manage investment portfolios. Their system deployed a group of agents with different functionality and coordinated them under case-based situations. They modeled the user, task and situation as different cases, so their system activated the distributed agents for information gathering, filtering and processing based on the given case. Their approach mainly focused on portfolio monitoring issues and has no mechanism to deal with uncertainty and urgency factors. Our system on the other hand reacts to the real-time market situation and gathers the relevant information as needed. Other related research on portfolio selection problems has received considerable attention in both financial and statistics literature [1,3,4,5,6,23]. The research in both financial and statistics literature concentrated on selecting a portfolio using historic data and their method have no means to deal with real-time information. On the other hand, our work provides the mechanism to integrate the real-time information into our deliberating process in order to provide a better decision support.

Recent approaches for the value of information problem include Matheson (1990), Jensen and Dittmer (1997), and Jensen and Liang (1994). Their approaches concentrated on performing the most valuable test by calculating the value of performing the tests. Horvitz, et al. (1993), presented an approach that deals with the time critical information display for space shuttle control. Zilberstein and Lesser [30] used a non-myopic approach for gathering small amounts of information from the web to assist the human user in comparison-shopping. Our system uses a myopic information gathering approach which is similar to Zilberstein and Lesser, but adds the uncertainty of the information accuracy into consideration.

In the field of model refinement, there are several approaches. The value of modeling was first addressed by Watson and Brown (1978) and Nickerson and Boyd (1980). Chang and Fung (1990) have considered the problem of dynamically refining and coarsening of the state variables in Bayesian networks. However, the value and cost of performing the operations were not addressed. Control of reasoning and rational decision making under resource constraints, using analyses of the expected value of computation and consideration of decisions on the use of alternative strategies and allocations of effort, has been explored by Horvitz (1988) and Russell and Wefald (1991). Poh and Horvitz (1993) explored the concept of expected value of refinement and applied it to structural, conceptual and quantitative refinements. Their work concentrated on providing a computational method for the criteria to perform the refinement. However, their work did not address the need of a guided algorithm to perform the node refinement throughout the network. In our work, we used a guided method to perform the conceptual refinement for our model. We see significant performance improvement of the model after applying the refinement algorithm.

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.

2. Investment agent architecture

Our system consists of four different sub-components; the object oriented Bayesian knowledge base, the decision model, the interface and the executor. The architecture of our system is depicted in Figure 1.

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 quantitative, 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 relationships.
- 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.

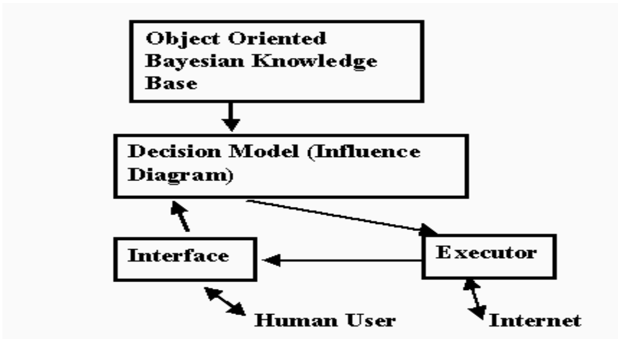

Figure 1. Investment agent's architecture.

Our system run constantly on a user's machine and send out web agents to gather the information only when our system think it is worthwhile to do so. Our system takes the cost of the information gathering into consideration. The cost including the monetary cost (cost of accessing the information) and cost of time (urgency). The monetary cost is the fee for the web agent to access certain information site, for example to obtain a company's financial Analysis & ratios from Wall Street Research Net (WSRN.COM) will cost you $49.95. The cost of time will be discussed in detail in section 7.

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 [28], according to which we rank our information sources by the value 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.

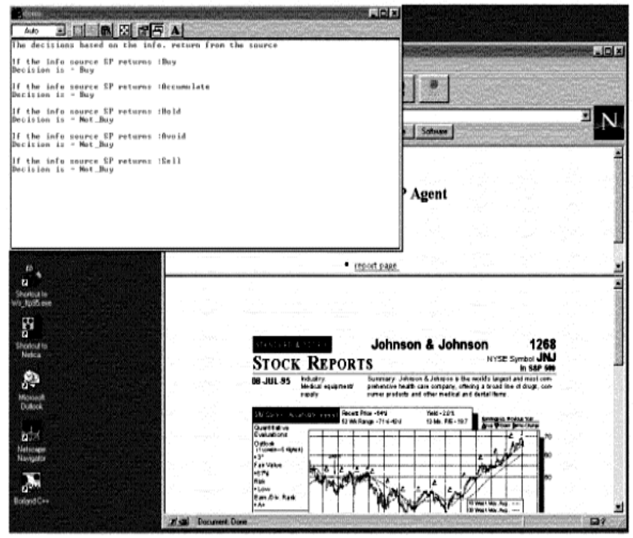

Figure 2. An executor retrieved a stock report from Standard and Poor stock report.

3. Object oriented bayesian knowledge base

There are more than 2000 stocks to pick from in today's market. It would be computationally infeasible to create a huge network that incorporates all the companies and the information sources in order to provide the investor with a portfolio recommendation. To handle the complexity issue, we created a hierarchical Object Oriented Bayesian Knowledge Base (OOBKB) [18,25].

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 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 5.

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.

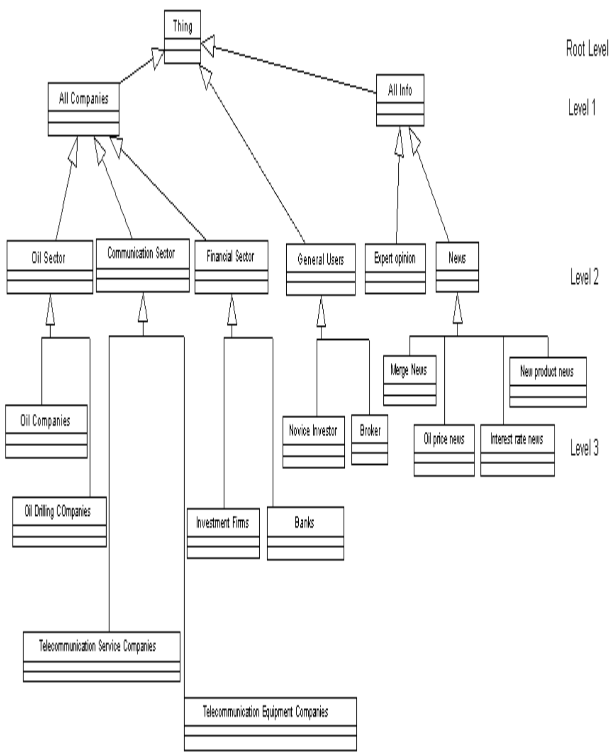

Figure 3. The class hierarchy of the OOBKB in a simple investment domain.

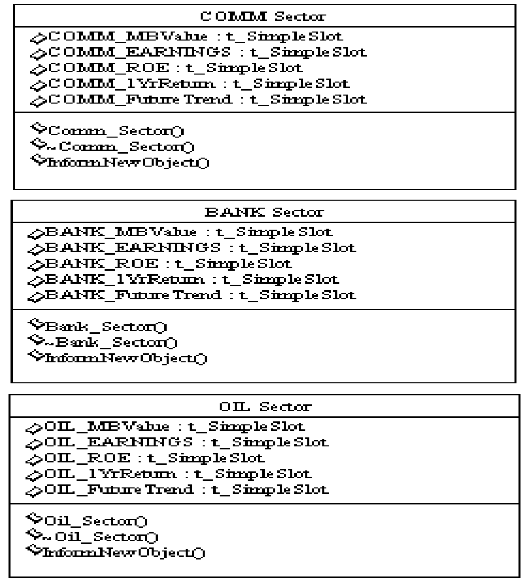

Figure 4. The classes instantiated detail from OOBKB.

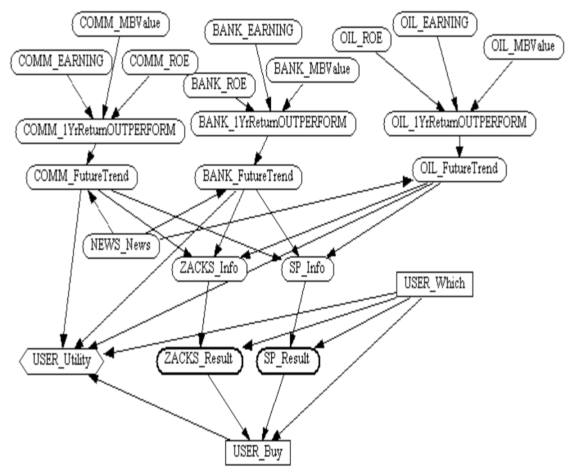

Figure 5. Decision model created from level two classes.

4. 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.

There are several methods for determining the optimal decision policy from an influence diagram. The first is by Howard and Matheson (1981)[14]. Their method consists of converting the influence diagram to a decision tree and solving for the optimal policy within the tree, using the exp-max labeling process. The disadvantage of this method is the enormous space required to store the tree. The second method is by Shachter (1986)[24]. His method consists of eliminating nodes from the diagram through a series of value preserving transformations. Each transformation leaves the expected utility intact, and at every step, the modified graph is still an influence diagram. Shachter proved that the transformation does not affect the optimal decision policy and the expected value of the optimal policy. However, it still requires a large amount of space to support the transformation steps. Pearl (1984)[21] has suggested a hybrid method using branch and bound techniques to prune the search space of the converted decision tree from the influence diagram. The disadvantage of this method is the trading off time for space, so it will requires more time to obtain the optimal policy. Our implementation uses Shachter's method, implemented within the Netica® Bayesian reasoning package.

The decision model coordinates with the strategy module in order to provide the sequential information-gathering plan for the executor to implement. The model includes the chance nodes that represent the results obtained from the investment advisor sources such as the First Call, Zacks Investment Inc., etc. that could be accessed by the system on the Internet. The model also contains the external variables of the domain, the Future trend node that represents the future price movement of the stock. The other part of the model to be elicited was the utility model, which is used to compare possible outcomes as a function of the decisions. The utility was expressed as the monetary gain or loss to the stock investor, and it is determined by the future trend, the buy/sell decision, and the information selection decision. To represent our model in Howard canonical form [12] we used a group of deterministic nodes such as Zacks Result, etc. to represent the information query results, and the information selection decision node. The conditional probability tables, each associated with an information query result node, represent the accuracy of each information source. They can be obtained from the historic accuracy data for each source, and are stored in the OOBKB. The conditional probability tables of the external financial factors such as beta, ROE etc. can also be obtained from historical data, and are stored in the OOBKB.

5. Model refinement

Mutual information is one of the most commonly used measures for ranking information sources. Here we apply this to our nodes in the network in order to provide guidance for refinement. Mutual information is based on the assumption that the uncertainty regarding any variable X represented by a probability distribution P(x) can be represented by the entropy function.

$$H(X) = -\sum_{x} P(x)\log P(x) \qquad -(1)$$

Assume that the target hypothesis is H and we want to know the uncertainty of H given X is instantiated to x, can be written as:

$$H(H|x) = -\sum_{h} P(h \mid x)\log P(h \mid x) \qquad -(2)$$

summing over all the possible outcome of x, we got

$$H(H|X) = -\sum_{x}\sum_{h} P(h,x)\log P(h \mid x) \qquad -(3)$$

where x, h are the possible values of the variable X and hypothesis H.

Then when we subtract H(H|X) from the original uncertainty in H prior to consulting X H(T), we can obtain the uncertainty reduction of H given X. This reduction is called Shannon’s mutual information.

$$(H|X) = H(H) - H(H|X) = -\sum_{x}\sum_{h} P(h,x)\log\frac{P(h,x)}{P(h)P(x)} \qquad -(4)$$

The value of refinement is defined as the difference between the model performance before and after the refinement. The performance of the model is measure by the average expected utility of the model run on the test cases. The current model performance MP(C) is represented as:

$$MP(C) = avg_n \sum_{n} EU(a \mid xn, C) \qquad -(5)$$

where xn is the test case input to the model.

The performance of the model after the refinement MP(R) is represented as:

$$MP(R) = avg_n \sum_{n} EU(a \mid xn, R) \qquad -(6)$$

Then the value of the refinement VR is:

$$VR = MP(R) - MP(C) \qquad -(7)$$

We applied the value of refinement to increase the values in the nodes and to increase the number of nodes in the decision model. To illustrate how the refinement works, we use our sample OOBKB in figure 3. We first create a model from the level 2 classes and applied the refinement algorithm to increase the value in the nodes. When the stopping criteria for the algorithm met: $VR \leq 0$, we have the refined model for the level 2 classes. At this point, the algorithm traverse down the subclasses of the level 2 classes to create mode detailed nodes for the model. Depends on the VR, only the subclass that increases our model's performance will be instantiated. Eventually the algorithm will stop and create a refined level 3 model, and so on. The heuristic algorithm we used is the following:

Heuristic Guided Refinement Algorithm

For all nodes except the target node

1. Calculates the mutual information value based on the target hypothesis.

2. Ranking the nodes based on the mutual information.

3. Calculates the current model's performance MP(C).

4. Refines the highest-ranking node by doubling the values in the node.

5. Calculates the refined model's performance MP(R).

6. Calculates the VR = MP(R) – MP(C).

7. If VR >0 then repeat the step 4 to 6.

8. Else if VR < 0 then go to the next highest ranking node and repeat the step 4 to 6.

9. Stop when going through all the nodes.

10. For all next level subclasses, instantiated one at a time.

11. Go to step 1 to 9 for all current nodes

By performing the model refinement, we could create different detailed level of models (figure 6 and 7) from our OOBKB and also increase our model's precision given the historic data. We could also update the OOBKB with the refined value for the node. For example, for the COMM_MBValue slot for the COMM sector in the figure 4, we initially have only 2 values. After the model refinement, we increase to 8 values for the slot. The update will help keeping our OOBKB more realistically and accurately reflecting the real world environment.

6. Urgency

In the investment domain, timing may be the 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.

Definition: The urgency, URG(t), is the value of one time unit and is defined as the difference between the overall market movement and our portfolio's movement at time t:

$$URG(t)=\max(0,\text{overall\_stock\_trend}_{t1})-\text{our\_portfolio\_trend}_{t1} \quad - (8)$$

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

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

$$\text{Overall\_stock\_trend}_{t1}=\frac{(\text{overall\_stock\_index}_{t1}-\text{overall\_stock\_index}_{t0})}{(t1-t0)} \quad - (9)$$

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

$$\text{Our\_portfolio\_trend}_{t1}=\frac{(\text{our\_portfolio\_index}_{t1}-\text{our\_portfolio\_index}_{t0})}{(t1-t0)} \quad -(10)$$

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 risk-less 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.

7. Trading off time for detail during modeling

We use a sample 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 subclasses 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.

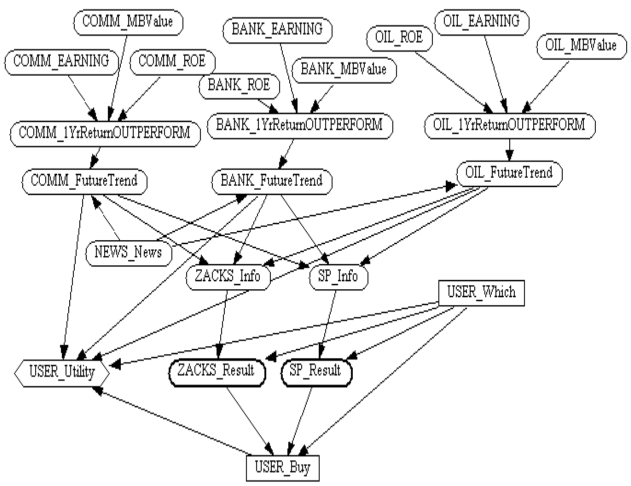

Figure 6. Abstract decision model creates from level 2 classes in OOBKB.

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.

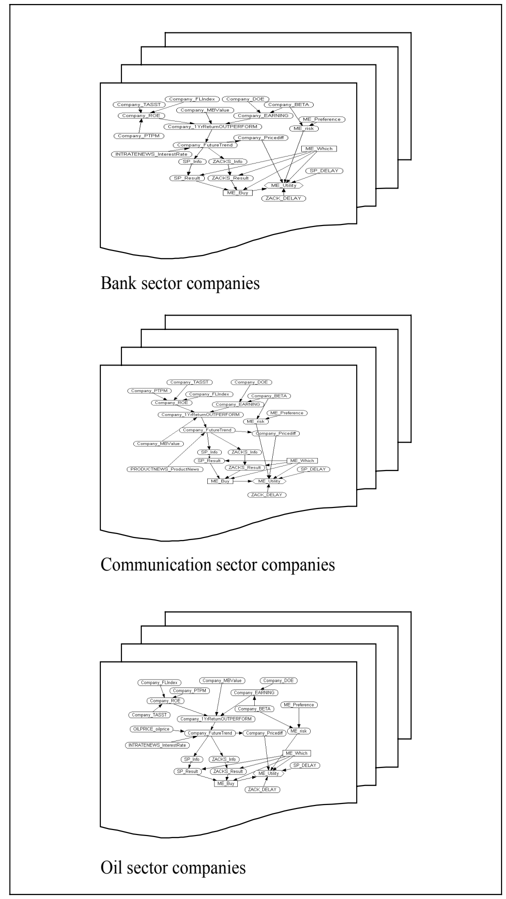

Figure 7. Detailed decision model creates from level 3 classes in OOBKB.

8. Experiments

Our system ran continuously in a background mode on investor's machine. It will activate itself every two minutes to assess the urgency of current situation and decides which detail level of the model it should create. The two minutes interval is chosen based on research [2,7] on the public information arrival in stock market and its impact to stock market returns. Berry and Howe [2] they measured the public information flow to financial markets and used it to document the patterns of information arrival on an intra-day flow. From their research, they measured a mean of 39.12 numbers of news stories in an hour within the trading hours of a day. This measure translates to a news story every 1.53 minutes, which is why we pick to activate our system every two minutes. This activation interval will keep our system up to date with the current information and response accordingly.

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 pervious two minutes of SP500 index data to calculate the urgency of the situation. We used the delta of the index within the two minutes to calculate the current moving trend of the index. For our example, we use the differences between the prices and divided with the number of seconds within the trading day to obtain the overall market trend in seconds and we assumed our current portfolio consists of cash only. Based on these assumptions, we then calculate the urgency using formula 8:

$$URG(t) = \max(0, (748.03-747.65)/120) - 0$$
$$= 3.17 \times 10^{-3} \text{ point/per second}$$

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 Netica as our inference engine (see Table 1).

Table 1. Runtime of the two models

Decision Model	Average Run time (10 runs)
Abstract decision model	0.26 seconds
Detailed decision model	2.03 seconds

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 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%. We then converted those numbers into points by multiple with the number of point of the SP500 when year started. Here is the comparison of the performance using one-year total return as criteria (see Table 2).

Table 2. Performance of the two decision models

Decision Model	One year total return in points
Abstract decision model	198.9
Detailed decision model	395.18

The annualized returns above, converted to return obtained per unit time (two minutes between possible trades, in our example) yield 6.02×10^{-3} and 1.198×10^{-2} points gains, 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, cost of time is 8.242×10^{-4} points, and for detailed model it is 6.44×10^{-3} points. Subtracting the cost from gain figures results in 5.2×10^{-3} and 5.5545×10^{-3} 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 90 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.

We also performed a more extensive experiment on our system using the S&P500 companies' data from year 1997 and 2000. In this setting, our system created a detailed model and selected companies to invest from the 500 companies of the S&P500. For the year 1997 our system obtained an average one year return of 38.26% from the companies our system selected. For 1997 the S&P500 index produced a return of 24.21% and the leading index fund Vanguard Index 500 produced a 32% return. And for year 2000 our system obtained an average one year return of 12.23%. For that year the S&P500 index produced a –9% return and the Vanguard index 500 produced a –8& return. Our system in both cases outperforms the S&P500 index and the leading mutual fund.

9. 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.

10. References

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

[2] T.D. Berry and K.M. Howe, "Public Information Arrival", *Journal of Finance*, vol. 49, issue 4, 1994, pp. 1331-1346.

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

[4] T.M. Cover, "Universal portfolios", *Mathematical Finance*, 1(1)(1991), pp. 1-29.

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

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

[7] L. H. Ederington and J.H. Lee, "How Markets Process Information: News Releases and Volatility", *Journal of Finance*, vol. 48, issue 4, 1993, pp. 1161-1191.

[8] G. Gorry and G. Barnett, "Experience with a model of sequential diagnosis", *Computers and Biomedical Research*, 1968.

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

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

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

[12] R.A. Howard, "Information value theory", *IEEE Transactions on Systems Science and Cybernetics*, 1966.

[13] R.A. Howard, "From Influence to Relevance to Knowledge", *Influence Diagrams, Belief Nets and Decision Analysis*(1990), pp. 3-23.

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

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

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

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

[18] J. E. Matheson, "Using Influence Diagrams to Value Information and Control". *Influence Diagrams, Belief Nets and Decision Analysis*, 1990, pp. 25-48.

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

[20] J.E. Matheson, "Using Influence Diagrams to Value Information and Control", *Influence Diagrams, Belief Nets and Decision Analysis(1990)*, pp. 25-48.

[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference Revised Second Printing*, Morgan Kaufmann, 1997.

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

[23] P.A. Samuelson, "Lifetime portfolio selection by dynamic stochastic programming", *Review Econom. Statist*. 51 (1060), pp. 239-246.

[24] R.D. Shachter, "Evaluating Influence Diagram", *Operations Research*, 1987, pp. 871-872.

[25] K.P. Sycara and K. Decker,."Intelligent Agents in Portfolio Management", *Agent Technology*, Springer-Verlag, 1997.

[26] D. Suryadi and P.J. Gmytrasiewicz, "Learning Models of Other Agents using Influence Diagrams", *In Proceedings of User Modeling: the Seventh International Conference*, 1999.

[27] R.R. Trippi and J.K. Lee, *Artificial Intelligence in Finance and Investing*, Irwin, 1996.

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

[29] S. L. Dittmer and F. V. Jensen. "Myopic value of information for influence diagrams", *In Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, 1997.

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