Intelligent Decisional Assistant that Facilitate the Choice of a Proper Computer System Applied in Business

Nicolae MĂRGINEAN
Finance-Accounting-IT Department, Bogdan-Vodă University, Cluj-Napoca, Romania
nicolae1976@yahoo.com

The choice of a proper computer system is not an easy task for a decider. One reason could be the present market development of computer systems applied in business. The big number of the Romanian market players determines a big number of computerized products, with a multitude of various properties. Our proposal tries to optimize and facilitate this decisional process within an e-shop where are sold IT packets applied in business, building an online decisional assistant, a special component conceived to facilitate the decision making needed for the selection of the pertinent IT package that fits the requirements of one certain business, described by the decider. The user interacts with the system as an online buyer that visit an e-shop where are sold IT package applied in economy.

***Keywords**: database, knowledge-base, decision tree, DSS, data mining, agents, reasoning, collaborative.*

1 Introduction

The fast changes of the business environment and the increasing complexity of the company activities requires a continuous adjustment, which often raises problems to the human decider. The computer systems applied in business were created as a solution to this challenge, due to their ability to process large volumes of data and aggregated information in order to optimize and get efficiency to the processes. Thus, nowadays, the computer systems are applied successfully in the field of production planning, in the management of customers and suppliers (or acquisitions and sales), in inventory management, human resources management, or in financial and accounts management and so on.

The desideratum of resources optimization represents a major objective of any manager of an organization. Because computer systems play an important role in achieving this objective, their choice and implementation is an acute necessity. The choice of a proper computer system is not an easy task for a decider. One reason could be the present market development of computer systems applied in business. Many specialists of the domain consider this market as powerful and mature. The big number of the Romanian market players determines a big number of computerized products, with various properties that must be adapted to the requirements of the certain business. We can tell that a human decider get at hand a large set of alternatives given by every computer system and an impressive number of analysis criteria for every alternative. His target is choosing one solution from the available alternatives so that the requirements of his business fit the properties of the chosen computer system. Unfortunately, the contemporary practice proves that the choice of a computer system applied in business is mainly based on subjective and non optimum reasons [1].

2 DSS as a collaborative system

Our proposal tries to eliminate these drawbacks, to optimize and to facilitate the decisional process, developing an online decision support system in order to choose the proper computer system for one business. This DSS will be integrated within an e-shop as an intelligent assistant for the buyers.

Decision support systems are information systems designed to interactively support all phases of a user's decision making process by means of analytical decision models, access to data warehouse and knowledge warehouse, using a set of normative and descriptive methods in order to choose the best decision or the satisfactory decisions.

We are of the opinion that the functions related to a decision-assisting computer systems may be easily detached starting from the stages of the decisional process it has to support. Thus, if we take as datum point the features and the stages of Simon's decisional process, the main functions of the decision support systems may be summarized as follows[2]:

- assistance offered to problem recognition stage;
- assistance offered to problem shaping stage;

- assistance offered to the stage of generating decisional alternatives;
- assistance in establishment of decisional criteria;
- assistance in selecting an optimal or acceptable solution;
- assistance in decision implementation;
- offering a user interface as friendly and natural as possible;
- easy access to the work components of the decision-assisting computer system and details about them;
- easy access to the stored information of an organization;
- enhanced flexibility and adaptability in order to cope with the rapid changes of the current business environment.

From the various number of decision support systems, we will focus on those where the management sub-system of certain large data volumes constitute the dominant technological component (according to the Power classification of the DSS). Also, the process of extracting new information from existing data collections within an organization will be achieved by data mining technology**.**

Blackboard system was the first attempt to integrate cooperating independent software modules. Using its three components, knowledge sources, blackboard and control component, the blackboard system tries to achieve a flexible and brainstorming style of a group of experts that solves a complex problem. The blackboard is the data structure that allows all knowledge sources to communicate with each other. Each knowledge source is an independent expert of certain aspects of the overall application. The control component leads the problem-solving process by allowing knowledge sources to respond to changes made to the blackboard.

Multi-agent systems are another technology that can be used to build collaborating-software applications. The knowledge sources of blackboard were transformed in agents, which are software components placed in some environment that is capable of autonomous actions in this environment and that can collaborate between them, in order to meet the reactive and proactive objectives. In our opinion, all the knowledge source activations in the blackboard system use the same thread of the problem solving process and can be viewed as something like an agent that is created in response to an activation event and that remains alive until it has been executed. Multi-agent systems use the multithreading and combine more autonomous agents having a long life cycle; these agents are concurrently collaborating in a distributed environment [2].

Shortly, the proposed framework of our system can be summarized in some ideas (figure 1):

- the user interacts with the system as an online buyer that visit an e-shop where are sold IT package applied in economy.
- the template of his requirements is taken by an interface agent from a multi-agent system
- the system has a catalog in a database format, related to many IT packages that exists on the market. Here we find the most properties of the IT packages
- having the reminded database, an agents' community, creates an knowledge base using some methods of data mining. The knowledge base will be used to select the records that fit the user's requirements.
- having all transactions stored in the database of the e-shop, it can be obtained a special file that contain the profiles of the buyers together with the sold IT packets to them.
- the intersection of the variants provided by the clustering with the variants provided by the case-base reasoning, will give to user the decisional alternatives.
- using a software library with multi-criteria methods useful in making decision in the field of management, the user can make a decision through the support of the system.

According to the previous framework, the mechanism of proposed architecture gravitates to the virtual community of agents, which, in a collaborative manner, tries to solve the problems. We have identified a lot of agents, all connected to a facilitator, everyone having a specific role within the system (figure 2):

- the cluster agents – for a given k, identifies k clusters, analyzing the data from the database
- the selector of results provided by cluster agents, returning the k clusters with minimal variances
- decision trees agent – takes a cluster and analyze it, returning the main knowledge that characterize the cluster
- production rules agents – takes a decision tree and transform it in production rules.
- case-base reasoning agent- offers the closest cases to the requirements of the user
- interface agent – assures the interface between the agent's community and the user. Also, the same agent represents the interface between

agents' community and the e-shop from the Internet

- library agent – stores some quantitative methods of choosing a decision from a set of alternatives using multi-criteria analyze.
- monitor agent – used for management of the agents
- debug interface agent that allows us to debug or test the services provided by the other agents.

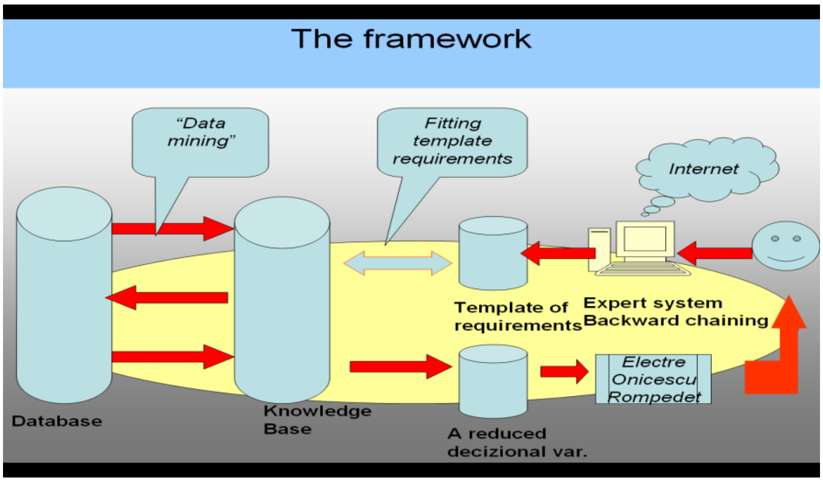


Fig. 1. The proposed framework

We have developed our system using Open Agent Architecture, a framework developed at the Artificial Intelligence Center of SRI International, an environment that allows creation of a flexible and dynamic community of software agents, providing a mean for integrating heterogeneous applications in a distributed infrastructure. We have chosen it due to its facilities for agent interaction based on delegation to a special agent named facilitator. This agent coordinates the agent community allowing parallel execution, error treatment, detection of conflicts that can occur in problem solving and so on. Also, the facilitator provides a global data store for its client agents, which allows them to adopt a blackboard style of interaction. A system configuration is not limited to a single facilitator. The facilitator maintains a knowledge base that contains all the capabilities of the agents community [3].

The client agents are usually specialized in providing a set of services and acts as a client of some facilitators. These services are named the capabilities of a certain agent. The declarations of the capabilities are referred as solvable. It is distinguished two types of solvable: procedure solvable and data solvable. A procedure solvable represents a procedure that acts in some manner whereas a data solvable offer a connection to a collection of data. A procedure solvable is handled by a declared handler whereas this is not necessary for a data solvable. Data solvable represent a dynamic collection of facts, which may be modified at runtime.

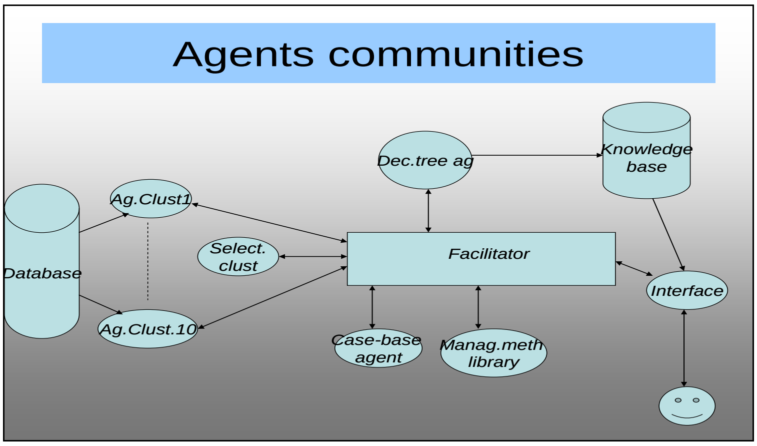


Fig. 2. The agents' community

Programming with delegation is very advantageous because reduces the dependencies among the agents of the community.

Also, meta-agents are very useful tools it the management of the community, its role being that of assisting the facilitator in coordinating the activities of the other agents [3].

In OAA, the agents can be developed using many languages as Prolog, C, Java, Visual Basic and so on. Every developed agent must incorporate in his code an agent library. For all reminded language is available a specific library.

Also, OAA provides a mechanism of triggering some actions or procedures when some set of conditions is met. Thus, using triggers, each agent can monitor either locally or remotely some events that can appear at its facilitator or peer agents. There are four types of triggers: communication triggers, data triggers, task triggers and time triggers.

The management of the agent community is achieved by two special agents[3]:

- the OAA monitor agent which have the next capabilities:
  - o the graphic visualization of agents community
  - o the visualization of all capabilities of an agent
  - o the deactivation of a certain agent
  - o the debugging of a user requirement
- the OAA debug interface agent that allows us to debug or test the services provided by the agents. The requirements can be introduce either using natural language or using the command "oaa_Solve".

The functionalities of our agents' community will be achieved using the multi-agent platform Open Agent Architecture, with one facilitator, with delegation style, and the language Sicstus Prolog for creating every agent.

3 Database

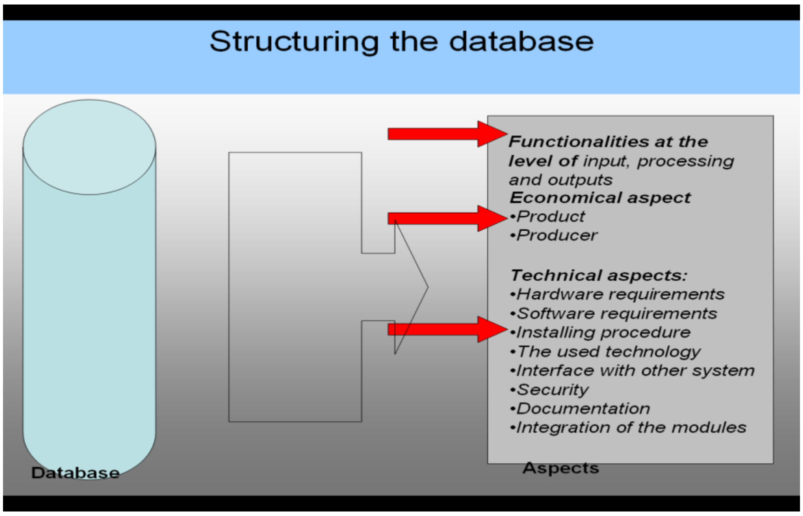
We will start building a database which will be used for data mining. First of all, we will try to identify the existent data source, beginning with Internet, going on the specialty literature and mass-media, and ending with direct contacts with the software producers, dealers and consulting firms in the field of computer system applied in business, in order to collect a big number of study aids. The installing kits (demo or full) of the computer system are very important to achieve an exhaustive analysis.

In order to build a proper database, we must have in attention the next important aspects (figure 3) [4]:

- ➢ identifying the functionalities provided by every analyzed computer system, at the level of:
  - ▪ inputs;
  - ▪ processing;
  - ▪ outputs;
- ➢ identifying the economical aspects of every computer system (especially the used price policies that directly influence the cost of the product implementation);
- ➢ identifying the technical and technological aspects of every computer system, as
  - ▪ the hardware requirements;
  - ▪ the software requirements;
  - ▪ the installing procedure;
  - ▪ the used technologies;
  - ▪ the interface with other systems;
  - ▪ the security of the system;
  - ▪ the documentation of the system;
- ➢ identifying the degree of the modules integration;
- ➢ identifying the properties related to the ergonomics of the products;
- ➢ economical and financial analysis of every software producer, that suppose:
  - ▪ storing economical and financial data;
  - ▪ calculation of the most important indicators;
  - ▪ evaluation of the trend of every firm;
  - ▪ interpretation of the economical and financial indicators by an expert system;
  - ▪ storing the results of the economical and financial data in order to make them available to the final user of decision support system.

Also, every software producer will be analyzed from the perspective of its economical and financial state, because, in our opinion, the state of a computer system is influenced by the state of its producer. Thus, we will collect all the possible economical and financial data related to every software producer into the database. The main source of data will be given by the balance sheets sent by every firm to the Ministry of Finance and Trade Registry Office. The data acquisition done, one agent that behaves like an expert system, will compute and interpret the main economical and financial indicators.

At this point, we can say that our database is ended and complete, so it can be used in making decision.



Structuring the database

Functionalities at the level of input, processing and outputs

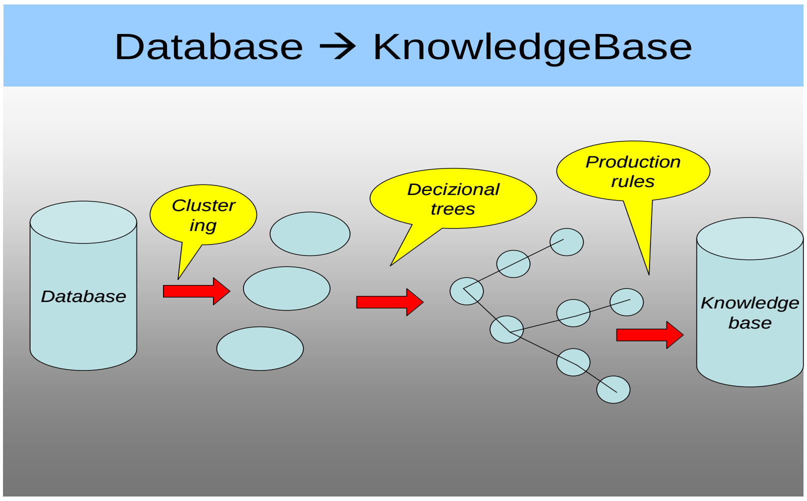
Economical aspect
- *Product*
- *Producer*

Technical aspects:
- *Hardware requirements*
- *Software requirements*
- *Installing procedure*
- *The used technology*
- *Interface with other system*
- *Security*
- *Documentation*
- *Integration of the modules*

Fig. 3. Structuring the database

4 Knowledge base

The knowledge base will be obtained applying two methods of data mining against the database presented previously (figure 4). Data mining is a technology of exploration of data existing in data bases, with the aim of discovering new aspects of the activity it is carried out, as well as correlations among events, associations among some deeds, behavioral patterns etc. By retrospective analysis of data, applying different methods of data mining technology we can discover some hidden laws which are difficult to be discovered.



Database → KnowledgeBase

Fig. 4. Obtaining knowledge from database using data mining

The first step will be the determining of the main groups that can be discovered from the existent data, using K-means clustering method. Every discovered cluster will be analyzed using decision tree method, discovering the main characteristics of these. The result of this step will be a set of decision trees. The last step will be represented by the conversion of the decision trees in production rules. The decision tree obtained in such a way transforms itself in production rules by the already known method (root →leaf).

It will be respected all the phases of the data exploration: problem definition, identification and selection of data, data preparation, model building and its evaluation followed by its adoption as an analysis tool. The clustering method (K-means) will identify the main groups of computer systems existing on the market. The rules that govern every group will be discovered using decision trees algorithms. The generated decision trees through ID3, C45 or C50 algorithms will be transformed in production rules, stored in the knowledge base.

Clustering method, a member of data mining group, allows us to identify existent groups in a volume of analyzed data. It is a technique that has capacity to discover hidden characteristics of a dataset records.

The main steps of k-means clustering are not difficult. At the beginning, it is determined the number of clusters K which became the centroids or centers of these clusters. We can take any random objects as the initial centroids. Then the K means algorithm will follow the next pseudo code until convergence is achieved[5].

```
take K random objects
Iterate until no object is moved from the certain group:
1. Determine the centroids using the averages of the cluster's members
2. Determine the distance of each object to the centroids
3. Group the object based on minimum distance (find the closest centroid).
```

K is the number of the obtained clusters but is generated in a random manner, so it's possible to find out other valid clusters. In order to determine a good value for the K, we propose to generate clusters for different values of K. In other words, for the same data grouping algorithm will be applied for different value of K. The best value of K will correspond to that situation where clusters have a minimal variance. The variance of a cluster will be computed using the sum of squared differences between every element and the average value from the cluster.

Another method which can be applied would be the method of using the decision trees.

Discovering the knowledge with the help of the decision trees represents one of the most used practical inductive inference method. In addition, the possibility of their easy way to be transposed under the form of some production rules contributed to the success of the adoption of this method.

In the specialty literature several learning algorithms imposed, by using decision trees. Generally speaking, all these algorithms have a common structure: they deal with the issue in a top-down manner and the search in the area of states is done in a greedy manner.

A well known algorithm would be the so-called ID3. ID3, for each addition of a new node to the tree, applies a certain statistic formula for all the attributes, according to the already existing examples, in order to establish which one of these best classifies the multitude of situations. In such a way the best attribute is added to the tree structure. For each possible value of the current attribute it is determined the new node in the tree, having in mind only the examples associated to each node. The above mentioned procedure is applied recursively for each node-leaf, till the moment when one of the two following conditions is fulfilled: either all the attributes were included in the decision tree, or all the values of the objective attribute, associated to the current node, are identical (the entropy is void). In order to represent the above mentioned statistical formula, the informational entropy has to be mentioned. This represents an indicator of the informational theory that characterizes the degree of disorder or the impurity of an arbitrary collection of examples. The bigger the value leads to the greater the informational disorder. Informational entropy is calculated according to the formula[6]:

$$Entropia(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

where "S" is the set of examples, c is the number of possible values of the objective attribute and p is a coefficient smaller than 1, equal to the ratio between the number of the examples that are afferent to a value of the objective attribute and the total number of examples (the cardinal number of examples $|S|$).

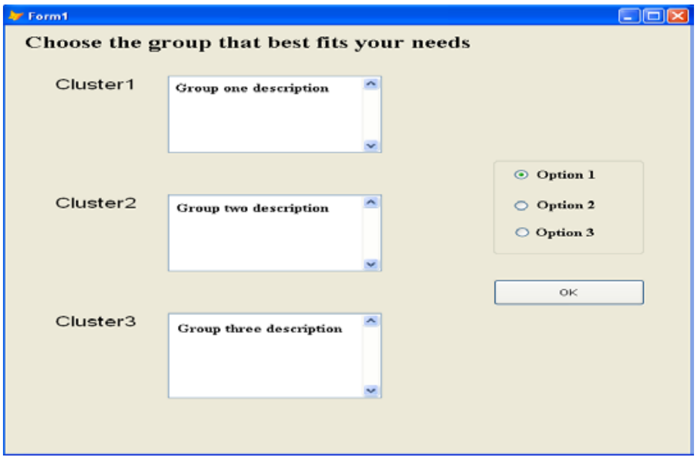
With the help of this indicator we can calculate for each of the attribute in the collection of examples the so-called informational gain, through which we surprise the effectiveness of an attribute in classifying the multitude of examples. In other words, it reflects the manner in which the informational entropy of a collection is been reduced, as a result of partitioning the examples with the help of an attribute. The calculus of this indicator is made after the formula[6]:

$$Castig(S, A) = Entropia(S) - \sum_{v \in valorile(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

where S is the collection of examples, A is the attribute for which the indicator is calculated, $|S|$ the cardinal of the set S, $|S_v|$ the cardinal of Sv set, and v is a value of the attribute A.

For the decision trees determined by ID3, the so-called issue of "overfitting" might appear. In other words, the decision tree which is formed on the basis of a certain set of data will not classify as good the new set of data, the classification being valid only in a small part. It is proved the fact that starting with dimensions exceeding 25 nodes, the decision trees represent weaker and weaker the new sets of examples. That is why one prefers the trees with less than 25 nodes [7]. Yet, in order to prevent this trouble, for trees with over than 30 nodes, we can use the well known algorithm of Quinlan, $C_{4.5}$.

The new knowledge base represents the main way to generate decisional alternatives. For every cluster, the interface agent will display next window with the main properties of the selected clusters in the previous steps (figure 5). In fact, the main properties are the discovered production rules expressed in a natural manner. The users will decide what cluster is the best solution for his requirement.



Choose the group that best fits your needs

Cluster1 Group one description

Cluster2 Group two description

Cluster3 Group three description

● Option 1
○ Option 2
○ Option 3

OK

Fig. 5. User interface – choosing the proper cluster

After the proper choice of the user, it can be identified all records from database that correspond to the selected cluster. The number of the resulted records is big enough to propose as decisional alternatives so it must conceive a new filter in order to reduce the number of them.

5. Obtaining the final decisional alternatives

In our opinion, case-base reasoning is one of the possible solutions. The case-base reasoning has as main idea the identification of cases that are similar to the case which has to be analyzed, in order to study their results for their further application for the current case. For our situation, we try to identify the records from the database that are closer to the requirements of the user. In order to improve the confidence degree of the results and reduce the decisional alternatives, it will be applied case-based reasoning on the special file obtained through the processing of the e-shop database. Here, we can distinguish two important tables: the transactions and the customers. In the transaction tables we have stored all the effective transactions of the e-shop. Also, we know that in every online transaction every buyer it is authenticated in the e-shop system. In this sense, at the beginning of the transaction, the buyer gives to e-shop system all the details about his business. All these details will be stored in the customers' catalog. Joining these two tables, we obtain a file where is present all the details of the customers together with the name of the IT packet sold to them. Applying case-base reasoning against this file, we can find out which are the IT packets sold to customers that have similar characteristics with the current buyer. Thus, it will be identified the nearest cases to the current situation (figure 6).

Finally, the intersection between the records provided by the case-base reasoning and the records corresponding to the selected cluster will represent the final decisional alternatives that will be proposed to user. These records will be displayed using the next window offered by the interface agent (figure 7).

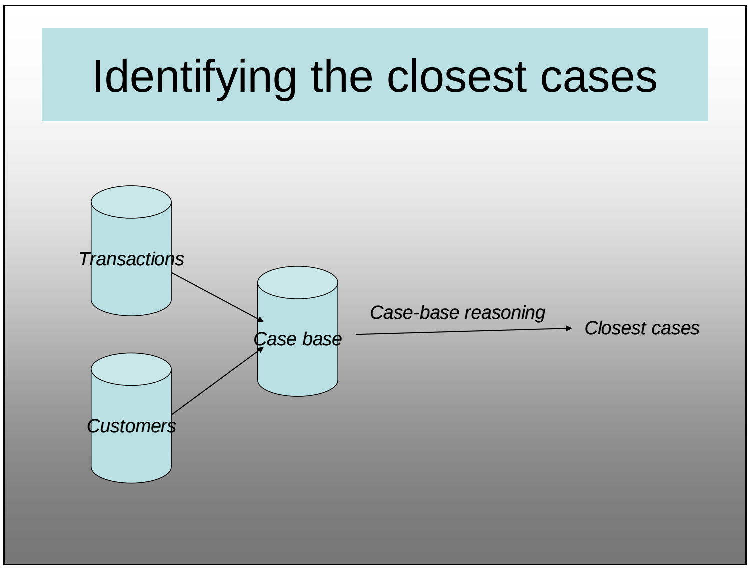


Fig. 6. Identifying the closest cases

Having a set of decisional alternatives, it will be made available to the user a software library with 3 managerial methods for the choice of the solution among a set of alternatives (Electre, Rompedet, Onicescu). The user has to choose one of these options, apply it and find out the final solution. It is very useful to use the multi attribute decision making methods because allow partially or completely rank the alternatives.

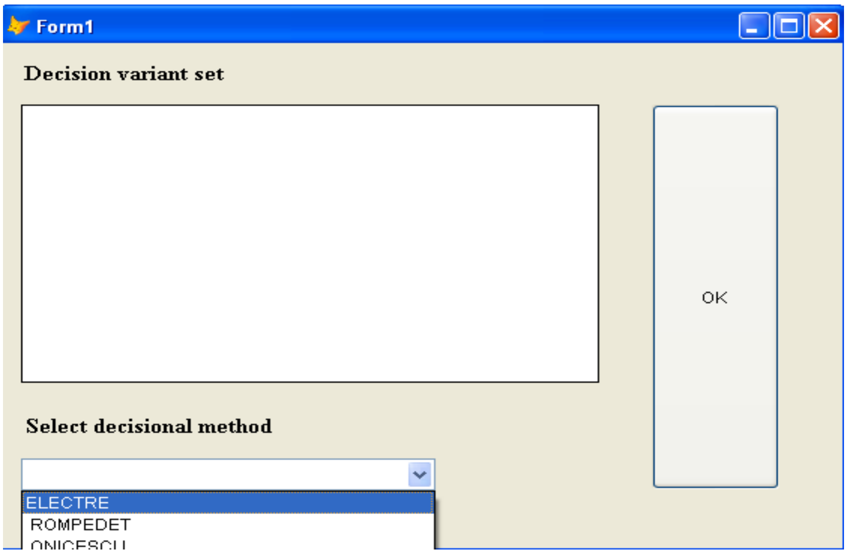


Fig. 7. User's interface – displaying decisional alternatives

As an example, Electre method, using concordance and discordance indices, is used to build a partial ranking in order to identify a set of promising alternatives.

6. Conclusions and perspectives

We have tried to issue a lot of proposals on building an intelligent decisional assistant, part of an e-shop, beginning with a conceptual framework, which emphasizes the main components of proposed system, and continuing with detailing the properties, the roles and functionalities of every component. We could observe that the mechanism of proposed architecture gravitates to the virtual community of agents, which, in a collaborative manner, tries to solve the problems. We have identified many agents, everyone having a specific role within the system.

An intelligent assistant like that presented in our study can make the difference between a common e-shop and our e-shop. Buying the proper IT package for an organization is not an easy task. It is an investment for a long term, can be expensive, it is responsible for many processes of an organization, so that a failed choice can affect in a negative manner the life of a business. As a result of the provided support by an intelligent assistant, the customers' satisfaction and confidence are increasing.

After a thorough analysis of the information given above, we'll reach the conclusion that the future DSS will be mostly built by use of multi-agent technologies and machine-learning. Specific manner of multi-agent systems for decomposition of the complex problems management confronts with in sub-problems, followed by a strong relationship of collaboration between independent modules and proactive modules bearing cognitive capacities, capable of handling knowledge from different places and electronic spaces represents the most proper manner of creating decision support systems.

In our knowledge-based society, very complex and unpredictable, intelligent DSS will become indispensable tools in making decisions.

References

[1] N. Mărginean, "A framework of an intelligent DSS that facilitate the proper choice of an ERP", In Proc. of *Economia de piata si statul de drept in Romania – membra a UE*, UBV, May 2008.

[2] N. Mărginean, *Sisteme inteligente de asistare a deciziei*, Editura Risoprint, Cluj-Napoca, 2006,

[3] D. Martin, A. Cheyer and D. Moran, The open agent architecture: A framework for building distributed Software Systems, Available online: http://www.ai.sri.com/oaa

[4] N. Mărginean, *Gestiunea informatică a organizatiilo*r, Ed. Risoprint, Cluj-Napoca, 2008.

[5] N. Mărginean and J. Sirbu, "Anayzing a day of stock exchange using clustering", *Rev Analele Universitatii Maritime Constanta*, 2008.

[6] T. Mitchel, *Machine Learning*, McGraw-Hill Companies, Boston, 1997.

[7] D. Zaharie, F. Albescu, I. Bojan, V. Ivancenco and C. Vasilescu, S*isteme informatice pentru asistarea deciziei – Data warehouse, OLAP, Data-mining,* Editura DualTech, Bucureşti, 2001.

Nicolae MĂRGINEAN has graduated in 1999 the Faculty of Economical Sciences, Babes-Bolyai University from Cluj-Napoca, specialization "Computer Science in Economy". Beginning with the year 2007, he holds a PhD diploma in Economical Cybernetics and Statistics, having as the title of the thesis "Intelligent Decision Support Systems". He joins the stuff of Bogdan Voda University from Cluj-Napoca, where, currently, is the lecturer in the field of "Computer Science in Economy", teaching disciplines as "Algorithms and Data Structure", "Artificial Intelligence", "Management of IT resources within an organizations", "ERP". He is the author and co-author of 6 books and many articles published in the journals and proceedings of national and international conferences, symposiums etc. related to domains as decision support systems, artificial intelligence, management of IT resources, WEB programming. Also, as an Authorized Physical Person with CAEN 6201, he is the software programmer within a team that develops an ERP named "INFCOM".