

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Flexible Dialogues in Decision Support Systems

Flávio R. S. Oliveira and Fernando B. Lima Neto
University of Pernambuco
Brazil

1. Introduction

In addition to its obvious operational use, data contained in Information Systems can be also used to derive analytical models which may help decision makers to comprehend and better tackle semi-structured problems (Laudon & Laudon, 2004). This kind of problem is frequently characterized by having: (i) a large number of options to analyze, (ii) a reasonable level of uncertainty associated to available information about the problem and also, (iii) a high impact concerning the decision outcome (Turban, 1995). The most usual approach to solve semi-structured problems is to combine the expertise of a decision maker and analytical capabilities of Decision Support Systems (DSS), empowered by a model database. Despite the potential aid DSS can offer, the empirical study performed by Moreau (Moreau, 2006) suggests that when the supportive tool is not considered enriching to decision processes it will probably be abandoned. This is because decision processes are in most cases so complex and consuming that it is not viable to add extra cognitive effort – for example, a decision tool not aligned to decision maker's preferences and restrictions.

In most cases, design and construction of analytical models to be used in a DSS focuses only on providing the best possible accuracy. Other cases encompass more than one objective, but they are frequently related to attributes associated to model-data relation. The user is sometimes left out and this not very smart approach often leads to a very accurate tool but not user friendly.

Some research results suggest that it is of high importance to take into consideration the user goals, needs, preferences and restrictions. Johnson (Johnson, 2008) suggests that the concept of cognitive exhaustion can have multiple interpretations and that it is useful to model some of its characteristics to provide better suited supportive environments. According to Chakraborty, the cognitive stile of decision maker influences the perception of utility and ease information systems usage (Chakraborty et al., 2008). Djamassbi shown through an experiment that the positive mood of decision makers causes a positive impact in the acceptance of new decision support tools (Djamassbi, 2008).

Based on these research findings the strategy devised here is to avoid DSS abandonment and to improve user satisfaction by providing flexible decision dialogues, incorporating information of the user cognitive profile. The cognitive profile information was employed to allow DSS to provide a kind of support more closely adapted to each user, thus reducing unexpected behaviors.

In order to provide DSS with flexibility to various problem characteristics, a new intelligent approach was proposed (iDSS). In which, three elements deserve special attention: (i)

Source: Decision Support Systems, Advances in, Book edited by: Ger Devlin,
 ISBN 978-953-307-069-8, pp. 342, March 2010, INTECH, Croatia, downloaded from SCIYO.COM

creation of analytical models based on optimization of its attributes, (ii) hybrid system architecture and (iii) customized intelligent training method.

The validity and viability of ideas put forward here were tested in two proofs of concept. The first, concerns the adaptation of iDSS to its user. The second, regards the evolution of iDSS after initial training conclusion. In both (complementary) cases, attributes such as accuracy, decision tree height and cognitive appropriateness were noted down. Results suggest that the proposed approach would be viable for use in real world problems.

The remainder of this chapter is organized as follows:

- Section 2 includes relevant topics about DSS and Intelligent Computing techniques;
- Section 3 explains the proposed approach to provide flexible dialogues in DSS;
- Section 4 presents the two proofs of concept and includes discussion about obtained results;
- Section 5 includes the conclusion and final remarks.

2. Background

The following subsections present relevant information for comprehending details of this chapter. Subsection 2.1 recaps basic theoretical foundation about Decision Support Systems (DSS). Subsection 2.2 delves on DSS which employ Computational Intelligence Techniques as analytical models. Subsection 2.3 comments on Decision Trees which are used in the proofs of concept as means to parameterize decision dialogues. Finally, subsection 2.4 brings in important concepts of Genetic Algorithms and explains the customized version used here to generate Decision Trees.

2.1 Decision Support Systems (DSS)

The complexity and uncertainty involved in semi-structured problems prevent their complete specification in terms of information systems. Thus, the synergistic approach is used to tackle this kind of problem (which rely heavily on users expertise). A Decision Support System (DSS) is an interactive tool employed to improve analytical capabilities of a decision maker, in order to improve quality of decisions taken in semi-structured problems (Turban, 1995). Figure 1 shows an overview of a typical DSS, comprising five key elements: (i) access to external databases, (ii) a Database Manager, (iii) a Model Database Manager, (iv) a Dialogue Manager and (v) the Decision Maker.

The access to External Databases is used to obtain data from other information systems. Raw data can be processed to be converted in information, which in turn can be stored in the Internal Database for later use. This data can also be used to derive analytical models which will be stored in the Model Database.

The Database Manager is responsible for acquiring external data and mediating the access to Internal Database. The Model Database Manager stores meta-data about analytical models contained in Model Database and is used to manipulate these models during decision making processes.

The Dialogue Manager is the interface layer that combines the expertise of Decision Makers and system analytical capabilities comprised both in Model and Internal Databases, providing the greatly needed interactivity to DSS. A common problem with DSS is related to designing the Dialogue Manager that can obfuscate system usefulness encouraging it to be sub-utilized or even abandoned.

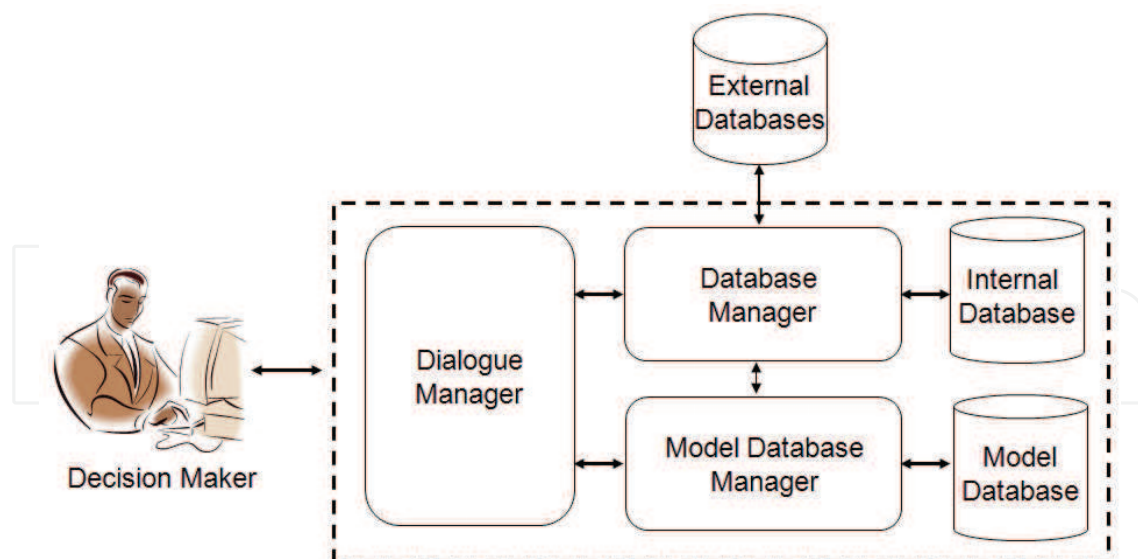


Fig. 1. Overview of a Decision Support System; adapted from (Laudon & Laudon, 2004).

A properly configured DSS can maximize the analytical capacity of a Decision Maker and, over time, is capable of improving the quality of decisions taken. The next subsection presents previous efforts to overcome some known limitations of DSS by employing Computational Intelligence Techniques.

2.2 Intelligent Decision Support Systems (iDSS)

Computational Intelligence Techniques are characterized by some distinguishing features such as capacities of learning, generalization and adaptation, all of them extremely useful in the domain of DSS.

Capacity of learning means that intelligent analytical models can be created based on examples, either input and output, related to phenomena involved in decision problems. This feature is important because real world problems sometimes cannot be easily formalized mathematically or statistically.

Capacity of generalization means that properly trained intelligent analytical models present coherent behaviors even when subject to patterns (*e.g.* decision problems) previously unseen.

Capacity of adaption means that intelligent analytical models are able to dynamically change its behavior to better deal with environmental circumstances. For example, some decision problems require fast response while others require a high accuracy. We stress that it is highly desirable that the DSS can switch its internal configuration to deal with both.

Oliveira, Pacheco and Lima Neto dealt with Database and Model Database Managers, in Hybrid Intelligent Decision Suite (HIDS) (Oliveira & Lima Neto, 2008) and Multi-Objective Hybrid Intelligent Decision Suite (MO-HIDS) (Pacheco et al., 2008). Both suites combined different computational intelligent techniques to solve decision problems related to general purpose benchmark databases (Newman, 1998) and real world decision problem such as sugarcane harvest (Pacheco et al., 2008).

The contribution of this chapter encompasses the dialogue manager and employs a new architecture to provide iDSS with (i) adaption to user and (ii) flexibility when dealing with different problem characteristics, all not seen in both decision suites previously mentioned.

2.3 Decision trees

The graphical view of information in most cases can improve comprehension of relations of cause and effect in problems. Decision Trees are Intelligent Computing technique (Russel & Norvig, 2002) extensively employed in Data Mining tasks (Rud, 2001) and widely accepted for use in Decision Support Systems. This favourable acceptance can be related to inherent facts about Decision Trees: (i) their training algorithms are usually fast, (ii) their level of accuracy is satisfactory in a wide range of problems and (iii) it is possible to obtain explanations about how a DT gets to the conclusion by inspecting its structure. Figure 2 shows a DT which could be used to parameterize decisions D1-D4, by means of dialogical questions Q1-Q3.

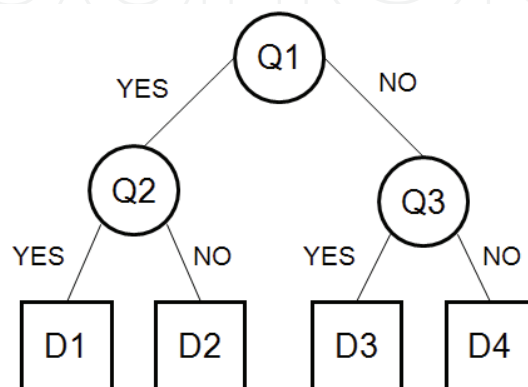


Fig. 2. Decision Tree used to parameterize a decision dialogue.

Below there is a hypothetical example of one decision dialogue parameterized by the decision tree presented in Figure 2:

1. DSS asks Question 1:
2. User answers NO:
3. DSS asks Question 3;
4. User answers YES;
5. DSS informs that the outcome is Decision 3.

In order to help in solving decision problems, the most common approach is to use problem related data to perform DT training. Some training algorithms always reach the same DT configuration for a given database; they often use metrics such as information gain and entropy (Quinlan, 1993)(Breiman et al., 1984).

In this work, Genetic Algorithms -GA (Haupt & Haupt, 2004) were employed as the training algorithm of DT. GA main contribution in this case is the possibility to create diverse models, and easily incorporating new metrics concerning DT and user cognitive profile.

2.4 Genetic algorithms

Decision processes (Chiavenato, 2004) presuppose a cycle in which candidate solutions are proposed, evaluated and selected. Genetic Algorithms (Haupt & Haupt, 2004) is a computational intelligence technique used primarily for optimization tasks. Figure 3 shows a typical GA cycle which is similar to a decision process cycle.

An initial population composed by candidate solutions is created and just after, according to problem specific criteria, it is evaluated. The fittest solutions are selected and combined in the crossover phase, aiming at new better solutions. A mutation phase occurs to provide an extra level of variability and also to avoid premature convergence. An evolved population

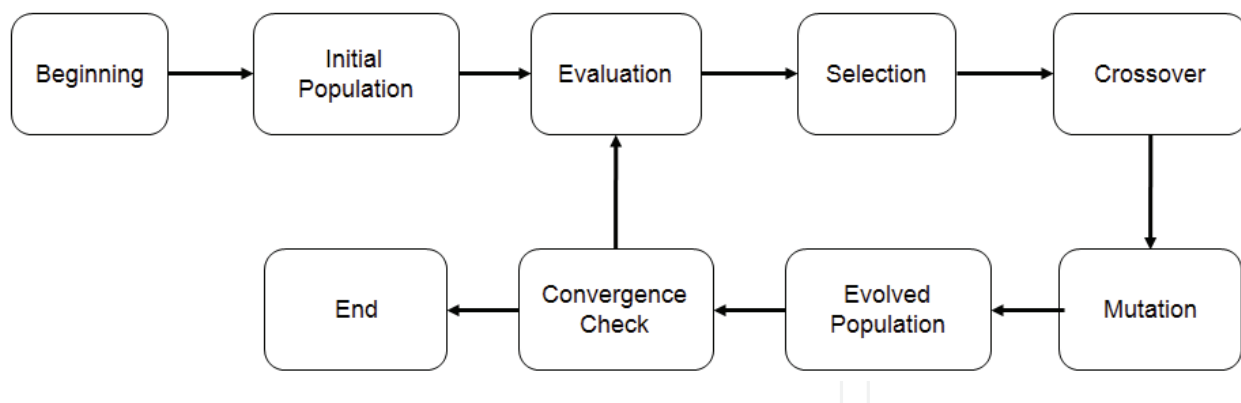


Fig. 3. Evolutive cycle of a basic Genetic Algorithm.

emerges from the mutation phase and it will be checked for convergence, concluding the evolutionary cycle or starting a new one. Figure 4 shows a special kind of GA specially designed to create DT. Input variables db stands for a database of patterns for training, p stands for a list of parameters and the output variable dt is the resulting decision tree.

The Aitkenhead algorithm (Aitkenhead, 2008) is composed of a main loop repeated until the stop criterion is met (line 2). A candidate decision tree dt is evaluated, suffers mutations to its original predictions (line 4) and questions (line 7). When these mutations lead to evolution (i.e. improvements in fitness) the decision tree must be stored. When the iterations are over, the best decision tree dt will be available for use.

This presented algorithm can automatically select the most relevant attributes and also, was customized to employ specific metrics in the fitness evaluation. More details can be seen in section 4.

```

1.  GenerateDecisionTree(  $db, p$  ):  $dt$ 
2.    While stop criterion is not met
3.      Evaluate fitness of  $dt$  e attribute it to  $F1$ 
4.      Performe  $x$  mutations to Question Nodes
5.      Store  $F1$  into  $F2$ 
6.      While  $j <$  number of mutations to Prediction Nodes
7.        Select and mutate a node
8.        Evaluate fitness  $F3$  of mutated tree
9.        If  $F3 > F2$ 
10.         Accept mutated tree
11.         Attribute  $F3$  to  $F2$ 
12.        End-If
13.      End-While
14.      If  $F2 > F1$ 
15.        Attribute resulting tree to  $dt$ 
16.        Attribute  $F2$  to  $F1$ 
17.      End-If
18.    End-While
19.  Return  $dt$ 
  
```

Fig. 4. Decision Tree training algorithm; adapted from (Aitkenhead, 2008).

3. An approach to provide flexible dialogues in decision support systems

The following subsections present our proposal to employ user cognitive profile information into analytical model creation, the system architecture and also the training method.

3.1 Intelligent model creation using cognitive profile information

The use of cognitive profile information was included in this approach to better bridge the gap between what the user expects and what the iDSS can offer. Also, by incorporating cognitive profile information it is easier to create an identification effect, making the user to perceive his own characteristics in the system, improving his confidence in the supportive tool.

User cognitive profile (CP) for purposes of this approach can be determined as shown in Equation 1 as a combination of Preferences (P) and Restrictions (R).

$$CP = (P, R) \quad (1)$$

In order to achieve flexible dialogues in an iDSS, it will be necessary to put forward specific metrics, and design a method to create intelligent models considering more than just accuracy. For this purpose, two classes of metrics were determined: model centric metrics (MCM) and user centric metrics (UCM).

Model centric metrics are those which can be determined by inspection of a model and are independent of user. Examples of MCM are accuracy or height of a Decision Tree.

User centric metrics are those which must be derived by user-model interaction and are thus subjective. Example of UCM is the satisfaction of a user with a given model when solving a decision problem.

Considering that an analytical model can be evaluated in terms of both classes of metrics, it is possible to think of model training as an optimization process such as Equation 2.

$$\text{Trained Model} = \text{Max}(\text{MCM}, \text{UCM}) \quad (2)$$

When dealing with multiple objectives, it is important to observe at least two situations: (i) when there is no conflict among objectives and (ii) when there is conflict among objectives.

When there are not conflicting objectives, they can be combined by an aggregative function and dealt with a single objective optimization technique such as Genetic Algorithms (Haupt & Haupt, 2004) or Particle Swarm Optimization (Eberhart & Kennedy, 1995).

When there are conflicting objectives, they must be optimized separately and literature suggests employing Multi-Objective Evolutionary Algorithms. SPEA2 (Zitzler et al., 2001) and NSGAII (Deb et al., 2000) have been widely used with success to generate Pareto fronts in conflicting multi-objective problems.

3.2 Proposed architecture

The architecture put forward here is composed of four main components: (i) a User Interface Module which concentrates on receiving inputs and emitting outputs, (ii) a Dialogue Manager responsible for selecting an appropriate analytical model and also responsible for exchanging information with decision maker, (iii) a System Memory where all relevant information about user and interactions are stored and (iv) a Model Manager, used to store and create new analytical models. Figure 5 shows the architecture employed to provide iDSS with flexibility to different problems and adaption to different user cognitive profiles.

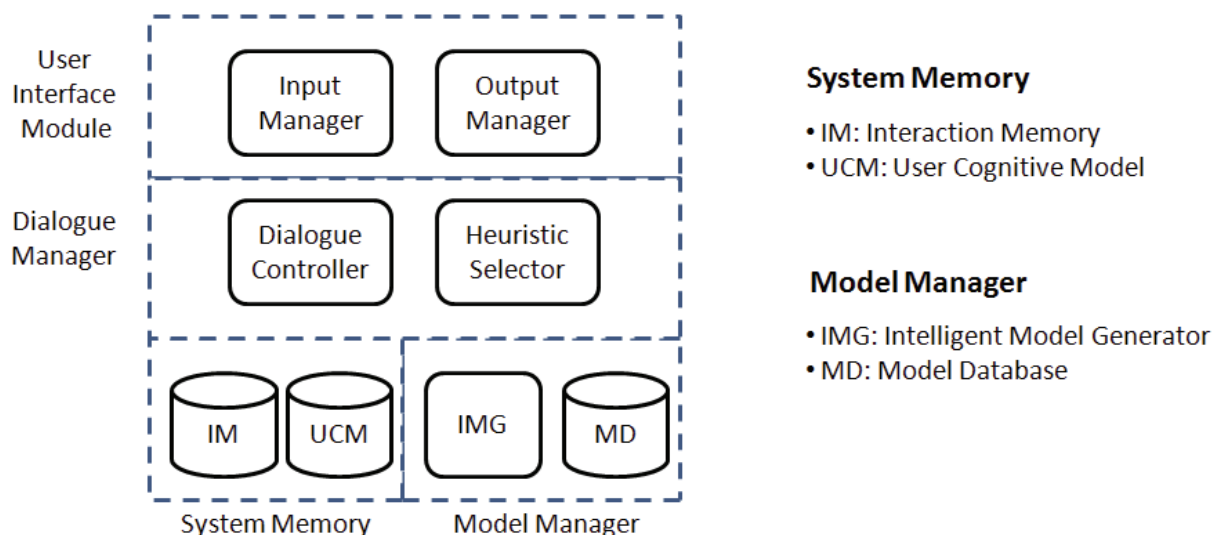


Fig. 5. Proposed Intelligent Decision Support System Architecture

Next we provide bottom-up explanations about architectural functioning:

1. Interaction Memory (IM) stores statistics about user interaction and feedbacks. Its content is useful for improving system performance over time;
2. User Cognitive Model (UCM) stores information about user, such as his preferences and restrictions. Its content is useful for creating analytical models which are appropriate according to user cognitive profile;
3. Intelligent Model Generator (IMG) is an active entity responsible for creating by demand models, employing information about each problem and the User Cognitive Model. The Decision Tree training algorithm presented in section 2.4 could be employed in this generator;
4. Model Database (MD) is a repository that contains all intelligent analytical models created;
5. Heuristic Selector is responsible for selecting an appropriate model according to problem specificities and according to User Cognitive Model and Interaction Manager. If a suitable model cannot be found, Intelligent Model Generator will be requested to create a new one;
6. Dialogue Controller, according to the selected model, requests and sends information to User Interface Module;
7. Input Manager is used to deal with user inputs, converting them to a suitable format for use by the adjacent Dialogue Controller. For example, Input Manager could receive inputs through checklists, plain text or even voice, converting them to a specific format required by Dialogue Controller;
8. Output Manager is used to put decision results into the best format according to each user. A decision to harvest some plots in a sugarcane harvest problem, for example, could be shown as textual lists, by abstract maps, or as geo-referenced map.

3.3 System training method

As an Intelligent System, iDSS must be subject to an effective training in order to work properly. Figure 6 shows five phases involved in training a iDSS created according to the architecture proposed in subsection 3.2.

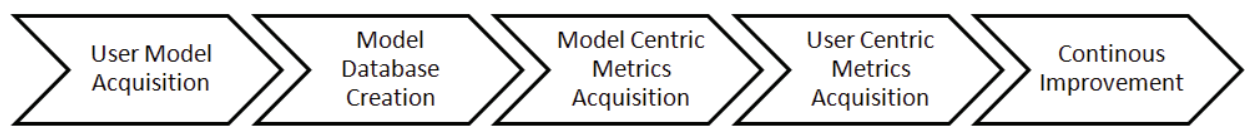


Fig. 6. Bird-eye view of phases of the new iDSS training.

Next we provide further explanations regarding training phases:

1. The first step is to store the user cognitive model, comprising preferences and restrictions. This can be done explicitly, requesting the user to inform his preferences and restrictions directly, or implicitly, by asking some questions and deriving preferences and restrictions according to user answers;
2. Using Intelligent Model Generator and User Cognitive Model, a Model Database must be created, employing the approach informed in section 3.1;
3. After defining the model centric metrics relevant to iDSS operation, it is important to evaluate Model Database to extract statistics for each model, storing them into the System Memory;
4. At this point, iDSS could be put into work. However its user centric metrics would not be calibrated, since it wasn't yet used by real decision makers. Instead of overburdening decision makers with iDSS training, it is better to employ the User Cognitive Model to simulate various user-system interactions. Feedback offered by the simulated decision maker must be stored in Interaction Memory;
5. Continuous Improvement phase was included to accommodate the dynamism of users and environment. Since both tend to change at least slightly over time, the four previous phases should be performed again over specific (desirably, large) periods of time, or according to reductions in the system performance.

4. Experiments and results

Two proofs of concept were implemented in order to test the validity and viability of ideas presented in section 3 into a controlled environment. Their details and results are presented in subsections 4.1 and 4.2. Table 1 shows information about each database employed in both proofs of concept. These databases were obtained from a benchmark repository used in Data Mining tasks (Rud, 2001). Each one has different attributes, presenting various levels of difficulty for iDSS, in this case used to help in solving classification problems.

Database	# of Patterns	# of Attributes	# of Classes
Breast	569	30	2
Contraceptive	1682	9	3
Glass	214	9	5
Heart	297	13	2
Wine	178	13	3

Table 1. Databases employed in proofs of concept.

Two model centric metrics were evaluated: (i) accuracy and (ii) tree height. Also, two user centric measures were evaluated: (i) cognitive appropriateness and (ii) satisfaction. The cognitive appropriateness is proportional to how many preferences were respected in decision tree. In this experiment, the preferences were modeled as problem attributes

considered important by a user and should be used whenever possible. The only restriction considered was the maximum tree size.

In order to evaluate user satisfaction, three kinds of cognitive profiles were created: (i) accuracy oriented, (ii) similarity oriented and (iii) speed oriented. Each cognitive profile has a Primary Satisfaction Criteria (PSC) and a Secondary Satisfaction Criteria(SSC). In order of importance, these criteria were:

- 1. Accuracy oriented: Accuracy and Cognitive Appropriateness;
- 2. Similarity oriented: Cognitive Appropriateness and Accuracy;
- 3. Speed oriented: Speed and Accuracy;

The level of satisfaction related to each measure, was as follows:

- 1. Satisfaction with Cognitive Appropriateness (CA):
 - a. If CA is bigger than 70%, then satisfaction is high;
 - b. If CA is smaller than 40%, then satisfaction is low;
 - c. Otherwise, then satisfaction is medium.
- 2. Satisfaction with Speed:
 - a. If tree height is smaller than 5, then satisfaction is high;
 - b. If tree height is bigger than 8, then satisfaction is low;
 - c. Otherwise, then satisfaction is medium.
- 3. Satisfaction with Accuracy: according to Table 2. For example, if Accuracy in *Breast* Database is bigger than 80%, the satisfaction is high; if it is smaller than 50%, it is low; otherwise it is medium.

The global satisfaction was calculated according to Equation 3:

Satisfaction = Min(PSC, SSC) (3)

Database	High	Low
<i>Breast</i>	> 80 %	< 50%
<i>Contraceptive</i>	> 50%	< 30%
<i>Glass</i>	> 50%	< 30%
<i>Heart</i>	> 70%	< 50%
<i>Wine</i>	> 80%	< 50%

Table 2. Satisfaction criteria concerning Accuracy.

At each user-system interaction, a pattern would be selected from the database used. Some attributes (75%) were randomly sorted and used as a decision problem. Heuristic Selector was used to check if there was at least one analytical model according to available attributes. If there was a model, the decision process continues, and after extracting the four measures, the global satisfaction was calculated and recorded. In cases which there was not a suitable model, a new one had to be created under demand by Intelligent Model Generator, employing the algorithm shown in Figure 4.

4.1 Proof of concept 1

This experiment aims at verifying if the proposed architecture of iDSS could present good levels of adaption to user and flexibility to problem characteristics. Table 3 presents the experimental setup employed.

After 500 interactions using each of the five databases shown in Table 1, some results were extracted and presented in Tables 4 to 8.

Variable	Value
Number of Generations	1000
Mutations to Question Nodes	150
Mutations to Prediction Nodes	150
Maximum Tree Height	10
Available Information	75%
Number of Interactions	500

Table 3. Experimental setup for Proof of Concept 1.

A global analysis of satisfaction levels suggests that this proof of concept was concluded successfully. In all cases, more than 84% of interactions ended with a High or Medium satisfaction. The number of models is smaller than the number of interactions because in some cases, the same model could be employed in more than one decision problem.

Observed Variables		Accuracy Oriented	Similarity Oriented	Speed Oriented
Number of Models		151	151	169
Accuracy		81.5719 %	80.6971 %	80.8097 %
Cognitive Appropriateness		61.8377 %	60.9271 %	62.7958 %
Tree Height		6.0132	6.2119	6.0946
Satisfaction	High	81.8 %	36.8 %	34.2 %
	Medium	18.2 %	61.4 %	65.8 %
	Low	0	1.8%	0

Table 4. Observed results in *Breast* Database.

Observed Variables		Accuracy Oriented	Similarity Oriented	Speed Oriented
Number of Models		27	26	23
Accuracy		43.5822 %	43.7402 %	43.2494 %
Cognitive Appropriateness		53.7037 %	56.7307 %	53.2608 %
Tree Height		4.2222	5.0384	3.6956
Satisfaction	High	1.4 %	43 %	82.2 %
	Medium	92.2 %	50.4 %	15 %
	Low	6.4 %	6.6 %	2.8 %

Table 5. Observed results in *Contraceptive* Database.

Observed Variables		Accuracy Oriented	Similarity Oriented	Speed Oriented
Number of Models		56	51	57
Accuracy		44.1398 %	41.6459 %	40.9439 %
Cognitive Appropriateness		62.4999 %	61.1111 %	62.2807 %
Tree Height		5.7321	6.7450	6.0175
Satisfaction	High	36.8 %	7.6 %	29.2 %
	Medium	63.2 %	92.4 %	69.6 %
	Low	0	0	1.2 %

Table 6. Observed results in *Glass* Database.

Tree height was respected in all cases, but it varied according to complexity of each database. For example, decision trees created in *Contraceptive* database were the smallest while those created in *Wine* database were the biggest.

Accuracy values ranged from 40.9439% in *Glass* database to 80.8097% in *Breast* Database. This wide range of values was obtained because of inherent difficulties contained in each database. Also, the availability of 75% of information may have avoided the algorithm to employ important attributes to reach better accuracies, making the proof of concept harder but more realistic.

Observed Variables		Accuracy Oriented	Similarity Oriented	Speed Oriented
Number of Models		197	187	200
Accuracy		72.6554 %	73.5321 %	72.9747 %
Cognitive Appropriateness		65.4258 %	65.4783 %	65.6111 %
Tree Height		6.6446	6.7967	6.5600
Satisfaction	High	78.0 %	24.6 %	14.4 %
	Medium	22.0 %	75.4 %	85.4 %
	Low	0	0	0.2 %

Table 7. Observed results in *Heart* Database.

Cognitive appropriateness values were also influenced by this restriction of available attributes, because not all attributes considered important by the decision maker, could be used in all decisions.

Based on results and levels of high and medium satisfaction in all three cognitive profiles, it is reasonable to expect that the architecture proposed would be viable to use in real problems.

Observed Variables		Accuracy Oriented	Similarity Oriented	Speed Oriented
Number of Models		196	196	194
Accuracy		77.5942 %	75.2507 %	77.6777 %
Cognitive Appropriateness		64.6683 %	64.7959 %	64.6907 %
Tree Height		7.5000	7.9081	7.5979 %
Satisfaction	High	50.4 %	31.8 %	1.4 %
	Medium	44.2 %	68.2 %	83.0 %
	Low	5.4 %	0	15.6 %

Table 8. Observed results in *Wine* Database.

4.2 Proof of concept 2

In order to verify iDSS ability to evolve and further adapt to user cognitive profile, this second proof of concept was implemented. Its execution was similar to Proof of Concept 1, but after the first 100 interactions, the iDSS was re-trained and re-evaluated. Its experimental setup is shown in Table 9. After 10 re-trainings, final results of satisfaction were registered and plotted in graphs contained in Figure 7.

Figure 7 (a) and (e) show expressive improvements in high satisfaction level, respectively presenting improvements of approximately 30% and 15% respectively. Figure 7 (c) and (d)

presented modest improvements of approximately 5%. Figure 7 (b) presented a decrease in performance of approximately 5%.

Variable	Value
Number of Generations	1000
Mutations to Question Nodes	150
Mutations to Prediction Nodes	150
Maximum Tree Height	10
Available Information	75%
Number of Interactions	100
Number of Re-trainings	10

Table 9. Experimental setup for Proof of Concept 2

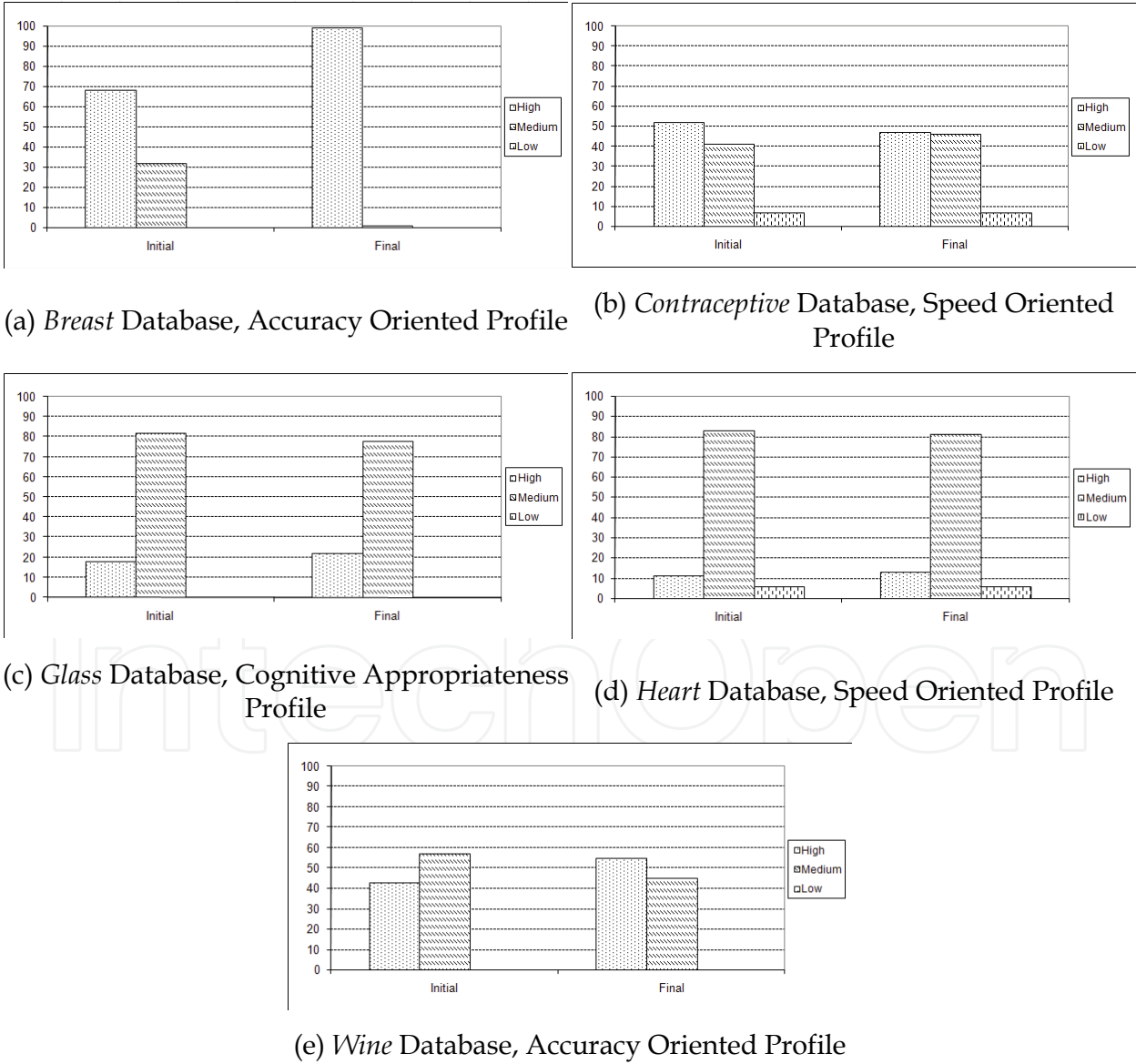


Fig. 7. Some results observed in Proof of Concept 2

These results suggest that the continuous improvement phase is important and useful to further guarantee adaption to user and problem flexibility, especially in real world problems. Satisfaction reduction, observed in Figure 7 (e), is related to training algorithm non-monotonic behavior. In some cases, model database it is already in the best possible configuration for a given problem database. Despite being relevant, it is important to create criteria to acknowledge the insertion of new models, preserving statistics about previous interactions and avoiding unwanted decreases in performance.

5. Conclusion

This chapter has presented an approach to provide DSS with flexibility to problem characteristics and adaption to user cognitive profile. This approach was comprised by: (i) a method that employs user cognitive profile information for decision models creation, (ii) a system architecture and (iii) a training method that enables the iDSS to interact with its user. Two proofs of concept were implemented and their results suggested that the proposed approach is valid and would be viable to tackle in real world decision problems.

The current version of the iDSS employed only Decision Trees to solve classification databases. It is important to highlight that the proposed approach is abstract and thus independent of technique and class of problem. Results shown could be further improved by fine tuning algorithmic parameters. Also, other classification techniques could be employed to further improve system accuracy and capability to deal with different problems. For example, even though not eloquent regarding explanations about classifications performed by Artificial Neural Networks (Haykin, 1994) they could be used to double check if a classification is correct.

As for future works we foresee including three aspects: (i) use of new computational intelligence techniques, (ii) integration of current results with HIDS (Oliveira & Lima Neto 2008) and MO-HIDS (Pacheco et al., 2008), and (iii) study and solution of real world problems.

Classification and Regression problems may be dealt with different Intelligent Computing techniques. According to problem characteristics and user preferences (e.g. need for explanation), the more suitable technique could be selected. Another avenue that this research can take is to extend the range of IC techniques also for usage in Model Database.

The previously mentioned papers of authors have dealt with simulation of decision scenarios, optimization and suggestion of lines of action. However, they used a one-fits-all approach which gave room for almost no flexibility in iDSS, hence an improvement point.

We strongly encourage more real world case studies. A good starting point would be to tackle decision problems in the medical area, which require good levels of accuracy, reliability in the system-user interaction and also, the mandatory need for explanations.

6. References

- Aitkenhead, M. (2008). *A Co-evolving Decision Tree Classification Method*, Expert Systems with Applications, n. 34, 2008, p. 18-25.
- Breiman, L.; Friedman, J.; Olshen, R. & Stone, C. (1984). *Classification and regression trees*, Wadsworth, 1984.
- Chackabarty, I.; Hub, J. & Cuid, D. (2008). *Examining the effects of cognitive style in individuals' technology use decision making*, Decision Support Systems, Vol. 45, 2008, pp. 228-241.

- Chiavenato, I. (2004). *Administração nos novos tempos*, 2nd Ed., Elsevier, Rio de Janeiro, 2004.
- Deb, K.; Pratap, A.; Agarwal, S. & Meyarivan, T. (2000). *A fast and elitist multi-objective Genetic algorithm: NSGA-II*, KanGAL technical report, Indian Institute of Technology, Kanpur, India, 2000.
- Djamasbi, S. & Stronga, D. (2008). *The effect of positive mood on intention to use computerized decision aids*, Information & Management, Vol. 45, 2008, pp. 43-51.
- Eberhart, R. & Kennedy, J. (1995). *A new optimizer using particle swarm theory*, Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan. p. 39-43, 1995.
- Haykin, S. (1994). *Neural Networks – A Comprehensive Foundation*. Prentice-Hall International Editions, NJ, USA, 1994.
- Haupt, R. & Haupt, S. (2004). *Practical Genetic Algorithms*, 2nd Ed., Wiley Interscience, 2004.
- Jonhson, J. (2008). *Man, my brain is tired: Linking depletion and cognitive effort in choice*, Journal of Consumer Psychology, Vol. 18, 2008, pp. 14-16.
- Laudon, K. & Laudon, J. (2004). *Sistemas de informação gerenciais: administrando a empresa digital*, 5ª Ed., Pearson Prentice Hall, 2004.
- Moreau, E. (2006). *The impact of intelligent decision support systems on intellectual task success: An empirical investigation*, Decision Support Systems, n. 42, p. 593-607, 2006.
- Newman, D.; Hettich, S.; Blake, C. & Merz, C. (1998). *UCI Repository of machine learning databases*, University of California, Irvine, Dept. of Information and Computer Sciences. USA, 1998.
- Oliveira, F. & Lima Neto, F. (2008). *An Evolutionary Approach to Provide Flexible Decision Dialogues in Intelligent Decision Support Systems*, Proceedings of the 8th International Conference on Hybrid Intelligent Systems (HIS 2008), Barcelona, Spain, 2008.
- Pacheco, D.; Oliveira, F. & Lima Neto, F. (2008). *Including Multi-objective Abilities in the Hybrid Intelligent Suite for Decision Support*, Proceedings of International Joint Conference on Neural Networks-IJCNN, 2008, Hong Kong, China.
- Quinlan, R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA.
- Rud, O. (2001). *Data Mining CookBook. Modeling Data for Marketing, Risk and Customer Relationship Management*, Wiley Computer Publishing, 2001.
- Russel, S. & Norvig, P. (2002). *Artificial Intelligence, A modern Approach*, Prentice Hall, 2002.
- Zitzler, E.; Laumanns, M & Thiele, L. (2008). *SPEA2: Improving the strength pareto evolutionary algorithm*, Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), 2001.



Decision Support Systems Advances in

Edited by Ger Devlin

ISBN 978-953-307-069-8

Hard cover, 342 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010

This book by In-Tech publishing helps the reader understand the power of informed decision making by covering a broad range of DSS (Decision Support Systems) applications in the fields of medical, environmental, transport and business. The expertise of the chapter writers spans an equally extensive spectrum of researchers from around the globe including universities in Canada, Mexico, Brazil and the United States, to institutes and universities in Italy, Germany, Poland, France, United Kingdom, Romania, Turkey and Ireland to as far east as Malaysia and Singapore and as far north as Finland. Decision Support Systems are not a new technology but they have evolved and developed with the ever demanding necessity to analyse a large number of options for decision makers (DM) for specific situations, where there is an increasing level of uncertainty about the problem at hand and where there is a high impact relative to the correct decisions to be made. DSS's offer decision makers a more stable solution to solving the semi-structured and unstructured problem. This is exactly what the reader will see in this book.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Flavio R. S. Oliveira and Fernando B. Lima Neto (2010). Flexible Dialogues in Decision Support Systems, Decision Support Systems Advances in, Ger Devlin (Ed.), ISBN: 978-953-307-069-8, InTech, Available from: <http://www.intechopen.com/books/decision-support-systems-advances-in/flexible-dialogues-in-decision-support-systems>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen