Adaptive Decision Tables
A Case Study of their Application to Decision-Taking Problems

T. C. Pedrazzi[1], A. H. Tchemra[1,2], R. L. A. Rocha[1]

[1]Department of Computing Engineering, Polytechnic School, University of São Paulo
[2]Department of Mathematics, Universidade Presbiteriana Mackenzie, São Paulo
E-mail: {thiago.pedrazzi, angela.tchemra, luis.rocha}@poli.usp.br

Abstract

Decision tables have been used traditionally in the solution of problems involving decision-taking tasks. In this paper, adaptive devices based on decision tables are used for the solution of decision-taking problems. The resulting adaptive decision tables have shown to be effective for their generality and flexibility, so they are appropriate for helping in the task of automatically choosing among several applicable alternatives at each stage in the process of decision taking. An illustrating example is shown in the business management field, and an overall comparative evaluation is shown.

1 Introduction

The evolution of information technology has contributed to transformations in several areas where its resources are applied. Computers and computational systems are very important support tools in the decision-taking process.

However, the decision processes can vary from the most complex and dynamic to the simplest, depending on the existing variables and the variables that can appear in the in presented problems. Contingent to the complexity, the decision-taking process requires the gathering of most information possible in order to reduce risks. The information can derive from past, present or predicted future facts.

With the use of Adaptive Techniques, decision processes can be adequate applications due to their natural characteristics. The methods developed in Adaptive Technology can be alternatively used in the resolution of complex problems and problems of dynamic nature. These methods can be more efficient than some traditional ones [1].

The adaptive devices are composed of dynamically variable rules [2]. In other words, through them, procedures can be self-modified depending on the detected situation. Therefore, this article will demonstrate how the mechanisms of the adaptive decision tables can be used in the decision-taking process with established criteria and criteria that are modified at every decision-taking cycle. And, to better understand the applied method, an example of application in the business management area will be studied for evaluation purposes. The method will be compared to classic methods whose procedures are, generally, fixed.

2 Decision-taking

A decision is a choice made between two or more available alternatives. Decision-taking is the process of choosing the best alternative to achieve the proposed goals [3].

A decision requires an individual, or a group of individuals, to choose one among several options. The options can vary from two to an unlimited number. The decisions can become complex if the sequence of decisions taken affect the subsequent options [4].

In the decision-taking process, the decision maker generally analyzes the objectives achieved through his actions and the situation involving the problem, taking into consideration the available resources and the decision expected consequences [5].

Hence, all problems whose solutions are based on a decision-taking process can be planned using the experience and results obtained from other cases with similar characteristics. A database that contains the strategies adopted in each case can assist in the various perspectives of the decision process and improve the quality of the information. In case there is no information in the database, there should have means to either enter new data or modify the database to improve the model.

Thus, to take a decision about a particular subject, the individual, or group of individuals, should first find the existing available information and determine whether new information has been added. After this, all information should be compiled for analysis and comparison. With all the results in hands, a solution to the problem should be found.

Approaches about the selection process can be found in several publications. Among them is the classic rational selection, where Ackoff and Sasieni (1968, apud in ref. 4) state that the decision process model should gather all the data that can represent the control variables that will determine the alternative actions, the uncontrollable variables relevant to the problem, and the decision criteria that can lead to the best action. Hence, this model should show the selection outcome.

In ref. 4, in administrative sciences the use of models that represent the decision processes, implemented in computational form, can represent means to control and manage the consequences of decisions. These systems can be based on the feedback concept. Depending on the available data and the process performance, feedback systems are generated to allow the decision maker to obtain more control and better performance.

Nowadays, organizations can acquire decision support systems to execute services, such as, access corporative database and simulate specific problems [6]. Therefore, it is possible to generate environments that provide analysis and evaluation of solutions alternatives in the decision-taking procedures, looking for the best solutions and trying to minimize errors.

The computational tools currently used by organizations are designed to supply quantitative and qualitative information that will assist in the decision-taking process.

As a rule, during the development of conventional systems, if a decision needs to be made, a previous analysis of the present information in the actual context is performed. The system is developed with the results of this analysis, allowing the information to be programmed and the situation to be simulated. If new information is generated for the same situation, the system cannot take care of it for the information is fixed.

With the adaptive techniques, however, the system is not only capable of determining the information needed for the decision-taking situation, but it is also capable of receiving new data which will eventually appear in each cycle of the process. Therefore, the system changes itself to provide better conditions to such decisions. This system is called adaptive device. The main characteristic of the adaptive device is to dynamically modify its own procedures [2], in consequence of the inputs, without external action, such as, of the user.

Complex problems with decision-taking characteristics can be found in the most varied areas. For example, business management, business strategic planning, marketing, production, investments, costs and profit analysis, operational and logistic research problems, military strategies and tactics, air traffic control, medical diagnostics, educational and learning processes, and many other areas with dynamic systems where behavior can change over time.

This work demonstrates how adaptive devices, through its generic characteristics and with wide applicability, can assist in choosing one alternative among many, in face of established criteria to each decision-taking cycle.

3 Adaptive Technology

A formal device is said to be adaptive whenever its behavior changes dynamically, in a direct response to its input stimuli, without interference of external agents, even its users. In order to achieve this feature, adaptive devices have to be self-modifiable. In other words, any possible changes in the device's behavior must be known at their full extent at any step of its operation in which the changes have to take place. Therefore, adaptive devices must be able to detect all situations causing possible modifications and to adequately react by imposing corresponding changes to the device's behavior. In this work, devices are considered whose behavior is based on the operation of subjacent non-adaptive devices that be fully described by some finite set of rules.

Application of adaptive technology is based on the formal model known as Adaptive Automata (A.A.) [7], [8], which is a Structured Push down Automata that, through the performance of pre-defined adaptive functions, change its behavior in response to its input stimuli.

Many projects have been developed using adaptive technology, what shows the versatility and usability of these techniques in wide-range application.

Using adaptive technology for solving computational problems has been very interesting, since it presents compatible results to the most-used techniques, with a cost/benefit relation much more interesting. We may list, e.g., Adaptive Statecharts, Adaptive Markov Chains, Adaptive Grammars, and Adaptive Decision Table, among others. Further information about these and other adaptive formalisms can be found at Adaptive Technology Lab web site (www.pcs.usp.br/~lta).

4 Applying Adaptive Technology on Decision-Support Systems

Among many adaptive devices available nowadays for application, after analyzing functioning, clearness and easiness to learning of each, we've concluded that the adaptive rule-driven devices are the best choice for implementing decision-support systems.

In ref. 2 is shown that the adaptive rule-driven devices can be seen as a two-layer system. The first layer is represented by a non-adaptive device, which is the basis for the system and is called underlying device, and the second layer, which is represented by the set of adaptive actions associated to that underlying device. The addition of this second layer empowers a common decision table (or any other underlying device) to Turing's Machine level (so it's called Turing-compatible), by a minimum cost.

For operating such an adaptive device, the non-adaptive underlying decision table is first used for determining the rule(s) matching the current situation of the condition predicates. Then, the selected adaptive rule is performed by executing the indicated adaptive actions associated to that rule. The adaptive rule can change the underlying device rules, so changing the systems behavior.

We've chosen the Adaptive Decision Table (ADT) as the core of this decision-support system for many reasons, such as:

- The non-adaptive underlying device is the ordinary decision table, which is well-known among the information systems solution providers presenting, dough, a higher commercial potential.
- The execution algorithm is quite straightforward, since it's as simple as the underlying device execution,
- Extending the non-adaptive underlying device to the adaptive one is quite easy, as shown in ref. 2, presenting extremely low-cost additions.

Further details in formal definition and implementation of ADT can be found in ref. 2 and ref. 9.

As can be seen from the differences between the non-adaptive decision table (fig 1) and the adaptive one (fig 2), the additional costs for 'upgrading' such device is really low. The underlying device is almost the same, regardless of an additional condition (state) in the adaptive one, which we use for automata simulation.

Attaching an adaptive layer to this device is quit straightforward, since it still remains as a table, only bigger. We can found in ref. 2 the meaning of each extra column and row.

As it's still a table, we need very short modifications in the runtime engine. However, ADT's computational power is infinitely higher than its non-adaptive counterpart, as the last one can be used only for Finite State Automata simulation.

material=	M□	M□	..	M□
building=	O_a	O_a	..	O_b
supplier=	F_0	F_1		F_1
price=	BP	GP		GP
proximity=	NB	NB		FB
accept:=		✓		

fig 1 – non-adaptive decision table

5 Illustrative Example

Let us illustrate the application of an ADT by means of a really simple example, such as purchasing building material using as landscape the civil construction business.

Ideal conditions for purchasing have been modeled such that the price has to be 'good' and the supplier has to be 'near' the building. These criteria (good and near) are pre-definite, for simplifying the example. At the moment of decision taking, both conditions have to be true for the selected supplier. We won't use the possible non-deterministic solution, since the backtracking (or other techniques for simulating parallel processing) increases the running time exponentially.

		Tag →	H	-	-	-	+	S	R	R	R	R	R	R	R	E
Subjacent decision table	Condition	state=					"L"		"I"	"J"	"J"	"J"	"J"	"K"	"L"	
		material=		p1	p1	p1	p1		"□"	"□"	M□	M□	M□			
		building=		p2	p2	p2	p2			"□"	O_a	O_b	O_b			
		supplier=					g1				F_1	F_1	F_2			
		price=		BP	BP	GP	GP				BP	GP	BP			
		proximity=		NB	FB	FB	NB				NB	FB	FB			
	Action	state:=		"K"	"K"	"K"	"K"	"I"	"J"	"K"	"L"	"L"	"L"	"L"		
		get (material)						✓								
		get (building)							✓							
		accept:=													✓	
Adaptive functions	Functions	*F*	B	✓	✓	✓	✓			✓						
	Other names	*p1*	P	"□"	"□"	"□"	"□"			"□"						
		p2	P	"□"	"□"	"□"	"□"			"□"						
		g1	G													

fig. 2 – ADT example before processing

For such this, an adaptive function (F) has been designed, composed by 4 elementary adaptive actions, as can be seen in fig. 2, which change the underlying device this way:

- The first three rules (first, second and third column AFTER the heading column) exclude the rules whose settings aren't the ideallized; therefore they're naturally excluded.
- The fourth rule add a rule whose settings are the ideal supplier.

A non-adaptive device rule has format as a 5-tuple (Mη, Bk, Sn, MPr, PO), with:

•Mη – Material Name / Code to be purchased.

•Bk – Building name to which the material will be used.

•Sn – Supplier's Name / Code.

•MPr – Material Price, which can be GP (good price) or BP (bad price).

•PO – Proximity to the Building, which can be NB (near the building) or FB (far from the building).

In the example, we will simulate an acquiring, where the material code is *M*β for using at the *Ob* building.

Initially, the device is started by applying the starting rule, which is identified by a 'S' on the TAG row (as may be seen in Fig. 2)

Then, the action **get(material)**, which is checked in this rule, will be applied. This action actually acquires the

material Name / Code. The condition **state** is set to “I”.
Now, searching the ADT for a rule that tests the condition state against the value “I” and the condition **material** fulfilled, the first column with an “R” heading will be applied.
At this point, the condition **state** will be set to “J” and the system will acquire the building name.
After that, the next rule eligible for applying is the second whose heading is “R”, which will start an adaptive action (named **F** in the example), passing as arguments **material** and **building** information just read.
The adaptive action will look for rules in the underlying device and delete any that has same material and building information and doesn’t fulfill the ideal conditions (that could be {Mβ, Ob, F1, GP, FB} or {Mβ, Ob, F2, BP, FB} or else {Mβ, Ob, F1, BP, NB}, and include a new supplier, which name will be generated by generator *g1*, with the desired settings {Mβ, Ob, g1, GP, NB}.
The system will then choose the new supplier, just added. If there were already a rule with ideal conditions, it would be chosen before the activation of any adaptive action. Such this is an intrinsic adaptive technology characteristic. The resulting ADT after this process can be seen at fig. 3.

		Tag →	H	-	-	-	+	S	R	R	R	R	R	R	E
Subjacent decision table	Condition	state=					"L"		"I"	"J"	"J"	"J"	"K"	"L"	
		material=		p1	p1	p1	p1		"□"	"□"	M□	M□			
		building=		p2	p2	p2	p2			"□"	O_a	O_b			
		supplier=					g1				F_1	F_x			
		price=		BP	BP	GP	GP				BP	GP			
		proximity=		NB	FB	FB	NB				NB	NB			
	Action	state:=		"K"	"K"	"K"	"K"	"I"	"J"	"K"	"L"	"L"	"L"		
		get (material)						✓							
		get (building)							✓						
		accept:=												✓	
Adaptive functions	Functions	*F*	B	✓	✓	✓	✓			✓					
	Other names	*p1*	P	"□"	"□"	"□"	"□"			"□"					
		p2	P	"□"	"□"	"□"	"□"			"□"					
		g1	G												

fig. 3 – ADT example after processing

6 Conclusion

Among attractive characteristics of this project, we can emphasize two of them: reliability and affordability. Reliability is achieved by its formal developing process, which through mathematical definitions, can ‘predict’ its behavior, preventing that way misleading. Production cost, as have been shown through this paper is quite low, though.
In conclusion, we expect that this project, as many others in this area, helps the reduction of production cost and increasing the spreading of adaptive solution to business problems.

References

[1] Pistori, H. (2003) Tecnologia Adaptativa em Engenharia de Computação: Estado da Arte e Aplicações. Tese de Doutorado, USP, São Paulo.

[2] Neto, J. J. (2001) Adaptive rule-driven devices - general formulation and case study. In: International Conference on Implementation and Application of Automata - CIAA 2001, 6th, Pretoria, South Africa, July 2001. Lectures Notes on Computer Science 2494. Berlin: Springer-Verlag, p.234-250.

[3] Certo, S. C. (2000) Modern management: diversity, quality, ethics and the global environment. New Jersey: Prentice Hall.

[4] Pidd, M. (1998) Modelagem empresarial: ferramentas para tomada de decisão. Trad. Gustavo Severo de Borba et al. Porto Alegre: Artes Médicas.

[5] Laudon, K., Laudon, J. (1999) Sistemas de Informação com Internet. LTC, Rio de Janeiro.

[6] Binder, F. V. (1994) Sistemas de Apoio à Decisão. Érica, São Paulo[1] Pistori, H. (2003) Tecnologia Adaptativa em Engenharia de Computação: Estado da Arte e Aplicações. Tese de Doutorado, USP, São Paulo.

[7] Neto, J. (1994) Adaptive automata for context-dependent languages. ACM SIGPLAN Notices, v.29, n.9, p.115-24, 1994.

[8] Neto, J. J., Pariente, C. A. B. (2002) Adaptive automata – a reduced complexity proposal. Conference on Implementation and Application of Automata – CIAA 2002, pages 161-170, July 2002.

[9] Rocha, R. L. A., Neto, J. J. (2003) Uma proposta de linguagem funcional com características adaptativas. CACIC 2003.