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 traditionally used for solving 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 proved to be effective due to their generality and flexibility. They are helpful tools for automatically choosing an applicable alternative among several available at each stage in the decision-taking process. An illustrative example as well as an overall comparative evaluation is shown in the business management field.

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.

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

Because of its features, decision processes are well suited as applications using the adaptive techniques. The use of methods based on the adaptive technology is an alternative to be considered in the resolution of complex problems, and those methods can be more efficient than traditional ones [1].

Adaptive devices are built of sets of rules that can be dynamically changed [2], which means that their internal procedures can be self-modified in order to face their input stimuli situations.

This paper shows how the adaptive decision tables' mechanism may be used in the decision-making processes, whose established criteria change at each decision-taking cycle. An application example of the adaptive decision table in the business area is shown in detail, in order to compare it to the traditional decision-making methods, and to make clear the use of adaptive technology.

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 too complex if the sequence of decisions taken affects the subsequent options [4].

In the decision process, the decision-taker usually must analyze the goals to be achieved by his actions, the situation within the problem, the available resources, and the consequence of the decision taken [5].

Hence, any problem whose solution is based on a decision-taking process can be planned using the experience and results of other similar processes. A database of adopted strategies in each case aids the decision process in many of its perspectives, and also improves information quality. In case there is no information in the database, there must have means to enter new data and, therefore, modify the database to improve the model.

Approaches about the decision-taking process can be found in several publications. Among them is the classic rational selection, where Ackoff and Sasieni (1968, apud [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 management science, the use of decision process models implemented in computers may become a way to control and manage the consequences of a decision [4]. Those systems are based on the feedback concept. Depending on the availability of the data, and the performance of the model, the systems are fed back allowing the decision taker to gain control and improve

his performance.
The computational tools currently available for use by organizations are designed to supply quantitative and qualitative information that assists in the decision-taking process.
As a rule, during the development of conventional systems, a previous analysis of the information in the actual context is performed. The system is developed based on the results of this analysis, allowing the information to be programmed and the situation to be simulated. If newer information emerges within the context, the system can no longer fit its answers, since its 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 also it is 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.

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 or 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 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 can be fully described by some finite set of rules.
Application of adaptive technology is based on a formal model known as Adaptive Automata (AA) [6, 2], which is a Structured Pushdown Automata that, through the performance of predefined adaptive functions, change its behavior in response to its input stimuli.
Many projects have been developed using adaptive technology, which shows the versatility and applicability of these techniques in wide-range application.
The use of adaptive technology for solving computational problems is very interesting, since it presents compatible results with the most commonly used techniques with a cost-effective relationship even more interesting. We can 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 the Adaptive Technology Lab web site (www.pcs.usp.br/~lta).

4 Applying Adaptive Technology on Decision-Support Systems

After analyzing the operation, clarity and easiness of the learning process of each adaptive device available nowadays, we have concluded that the adaptive rule-driven device is the best choice for implementing decision-support systems.
In [2], 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. The second layer 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 (the so-called Turing compatibility) at a minimum cost.
To operate such adaptive device, one should initially use the non-adaptive underlying decision table to determine the rule(s) that matches 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, changing, therefore, the systems behavior.

material=	$M\alpha$	$M\alpha$	..	$M\eta$
building=	B_a	B_a	..	B_b
supplier=	S_1	S_2		S_3
price=	BP	GP		GP
proximity=	NB	NB		FB
purchase:=		✓		

Fig 1 – non-adaptive decision table

The Adaptive Decision Table (ADT) was selected as the core of this decision-support system for reasons such as:

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

Further details about formal definition and implementation of ADT can be found in [2, 7].
As can be seen from the differences between the non-

adaptive decision table (fig 1) and the adaptive one (fig 2), the additional cost to 'upgrade' such device is really low. The underlying device is the same.

Attaching an adaptive layer to this device is a straightforward procedure, since it remains as a table, with increased size, though. The meaning of each extra column and row can be found in ref. 2.

As a table, just a few modifications need to be made for the runtime engine. However, ADT's computational power is much higher than its non-adaptive counterpart is. The last one can be used only for Finite State Automata simulation.

		Tag →	H	-	-	-	+	S	R	R	R	R	R	R	R	E
Subjacent decision table	Condition	material=		p1	p1	p1	p1		"μ"	M_α	M_α	M_α	M_β	M_β	M_χ	
		building=		p2	p2	p2	p2		"o"	B_a	B_b	B_b	B_a	B_b	B_a	
		supplier=					g1			S_1	S_1	S_2	S_1	S_2	S_2	
		price=		BP	BP	GP	GP			GP	GP	BP	GP	GP	BP	
		proximity=		NB	FB	FB	NB			NB	FB	FB	NB	NB	NB	
	Act	get(data)						✓								
Adaptive functions	Func	*F*	B													
	Other names	*p1*	P						"μ"							
		p2	P						"o"							
		g1	G													

Fig. 2 – ADT example before processing

5 Illustrative Example

The application of an ADT will be illustrated through a very simple example, such as the purchase of construction material for civil engineering business.

Ideal conditions for the purchase have been established, such as price has to be 'good' and the supplier has to be 'near' the building. These criteria (good and near) are pre-definite to simplify the example. At the moment of decision-taking, both conditions have to be true for the selected supplier. The possible non-deterministic solution will not be used, since the backtracking (or other techniques to simulate parallel processing) increases exponentially the running time.

Since this example must be short in length, some auxiliary functions such as the one that would perform price analysis, proximity criteria and insertion of new materials were not shown.

For inserting new materials, an adaptive function would receive material name / code to be inserted together its respective building name, and create new columns for it in the ADT. Note that the fields **supplier**, **price** and **proximity** would be empty.

A timely-started function (**bid**) would then replace the previous price analysis based on new market information, and also fulfill the just-inserted material's price and supplier field on those that does not have these fields fulfilled.

There may be another function that, based on GPS information and on some distance criteria, would analyze if the supplier are 'near' or 'far from' the building. Again, this function was also omitted for sake of simplicity.

An adaptive function (F) has been designed for this example. This function is composed of 4 elementary adaptive actions, as seen in fig. 2, which change the underlying device in the following manner:

- The first three rules (first, second and third column after the heading column) exclude the rules whose settings are not ideal; therefore, they are naturally excluded.
- The fourth rule add a rule whose settings are the ideal supplier.

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

•Mη – Material Name / Code to be purchased.

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

•Sn – Supplier's Name / Code.

•MPr – Price of the material, 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 a purchasing, where the material code is *Mβ* to be used at the *Bb* 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(data)**, which is checked in this rule, will be applied. This action actually acquires the material Name / Code and the building name

Now, we search the ADT for a rule that tests the conditions **material** and **building** against the values just

read (and, of course, with the ideal conditions for purchasing).
There is no eligible rule for this case, so the adaptive rule will call the adaptive function, passing as arguments **material** and **building** information just read.
The adaptive action will search 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, Bb, ,GP,FB} or {Mβ, Bb, ,BP,FB} or else {Mβ, Bb, ,BP,NB}), and include a new supplier, which name will be generated by generator *g1*, with the desired settings {Mβ, Bb, *g1*,GP, NB}.
The system will then select the new supplier, just added. If there were already a rule with ideal conditions, it would be selected before the activation of any adaptive action. This is an intrinsic adaptive technology characteristic. The resulting ADT after this process can be seen in fig. 3.

		Tag →	H	-	-	-	+	S	R	R	R	R	R	R	E
Subjacent decision table	Condition	material=		p1	p1	p1	p1		"μ"	M_α	M_α	M_β	M_β	M_χ	
		building=		p2	p2	p2	p2		"o"	B_a	B_b	B_a	B_b	B_a	
		supplier=					g1			S_1	S_x	S_1	S_2	S_2	
		price=		BP	BP	GP	GP			GP	GP	GP	GP	BP	
		proximity=		NB	FB	FB	NB			NB	NB	NB	NB	NB	
	Act	get(data)						✓							
Adaptive functions	Func	*F*	B												
	Other names	*p1*	P						"μ"						
		p2	P						"o"						
		g1	G												

Fig. 3 – ADT example after processing

6 Conclusion

This study has shown that some real-world problems may be adequately modeled through adaptive techniques. The study has also shown that where conventional modeling may not be adequate, an adaptive modeling can fit the needs.
Among attractive characteristics of this project, two can be emphasized: reliability and affordability. Reliability is achieved by its formal developing process, which through mathematical definitions can 'predict' its behavior, preventing misleading. The production cost, however, as presented on this paper is very low.
In conclusion, this project is expected, as many others in this area, to help in the reduction of production cost and in the increasing application of adaptive solution to business problems in general.

References

[1] 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.

[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 Informaçăo com Internet. LTC, Rio de Janeiro

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

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