Towards More Realistic Modelling of a User's Evaluation Process

Wilken Schütz, Ralph Schäfer[1]

DFKI GmbH
Stuhlsatzenhausweg 3, 66123 Saarbrücken
{Wilken.Schuetz,Ralph.Schaefer}@dfki.de

Abstract. Modelling the user's evaluation process is an important task in user adaptive systems. In an idealization, Multi-Attribute Utility Theory (MAUT) can be used to represent that process. In this paper we describe the law of diminishing marginal utility and present a possible representation of that law in the MAUT context. In the following, we analyse the influence of the use of different value functions in three illustrative scenarios. Furthermore, we discuss the important role of modelling the user's evaluation process with respect to the dimension *economy*.

Keywords: User modelling, MAUT, evaluation scheme, diminishing marginal utility

1 Introduction

Every *user* of a company's e-commerce system is a potential *customer* of that company. In order not to lose those potential customers and to be competitive to real world stores, *user-adaptivity* will become a crucial feature in modern online shops. It is important to know the user's interests and preferences to be able to provide a product that really satisfies the customer.

Understanding the process of the user's evaluation of products is the necessary basis for modelling the user's interests. The *Multi-Attribute Utility Theory* (MAUT) [WE86] is an approach to represent that process in an idealised way. According to MAUT, a product is evaluated with respect to different dimensions, where every dimension represents a certain utility of the product for the user, e.g. *reliability*. In this paper, the dependency between a product's attributes and the evaluation with respect to a dimension is analysed. This dependency is defined in form of a value function.

We will also have a look at the impact of the value functions in various application scenarios. On one hand, we differentiate between assisting the user when searching a satisfying product in a catalogue and assisting her when configuring a product. On the other hand we distinguish between the problem of interpreting the user's choices and the question of how to suggest adequate values to the user.

The next section gives a short introduction to evaluating products with MAUT. Section 3 defines an example domain to illustrate the ideas of this work and shows the naive approach to realize MAUT in that domain. A refined kind of value functions that represents the law of diminishing marginal utility is presented in Section 4. Additionally, factors influencing the very important dimension *economy* are discussed.

2 Evaluation of a Product with MAUT

MAUT defines an evaluation scheme for products. For example, consumer organisations like the German „Stiftung Warentest" use this approach for evaluating products.

MAUT is based on the idea that the overall evaluation $v(p)$ of a product p can be defined as the weighted sum of the product's evaluation with respect to domain specific dimensions d_i:

$$v(p) = \sum_{i=1}^{n} w_i \cdot v_i(p) .$$

Here, $v_i(p)$ is the evaluation of the product on dimension d_i and w_i is the weight that specifies the impact of dimension d_i on the overall evaluation. Let n be the number of dimensions, then $\sum_{i=1}^{n} w_i = 1$ must hold.

[1] This work has been supported by the EC through the IST-Programme under contract IST-1999-10688 CAWICOMS (see http://www.cawicoms.org for additional information).

Each dimension represents a certain utility of the product for the user. Each dimension is defined in such a way that a high evaluation with respect to a dimension always means a positive characteristic of the product. So for example typical dimensions for an ink jet printer could be *economy*, *printout-quality*, *performance, reliability,* and *environmental friendliness*.

For every domain, a set A of attributes of the product can be defined. In the printer example, those attributes might be the *number of pages per minute* or the *resolution*. For every instance, i.e. for every concrete product, an attribute a has a certain value or level $l(a)$, e.g. the level of *resolution* could be *2400x1200* for a specific printer. Furthermore only a subset $A_i \subseteq A$ is relevant for dimension d_i. The evaluation on dimension d_i is then defined as follows:

$$v_i(p) = \sum_{a \in A_i} w_{ai} \cdot v_{ai}(l(a))$$

Here again, w_{ai} is the weight that specifies the impact of attribute a on dimension d_i and we have $\sum_{a \in A_i} w_{ai} = 1, \forall i = 1,...,n$. Without loss of generality, we assume that an evaluation always ranges between 0 (worst) and 10 (best). So $v_{ai} : A \rightarrow [0,10]$ is a function that specifies the dependency between an attribute a and a dimension d_i. Such a function is called *value function*.[2]

In the printer example, we could think about a value function which defines that printing only 3 pages per minute or less is worst (0 points) and that printing more than 20 pages per minute is best (10 points) with respect to the dimension *performance*. The easiest function that represents this dependency would be linear and can be displayed graphically as shown in Figure 1.

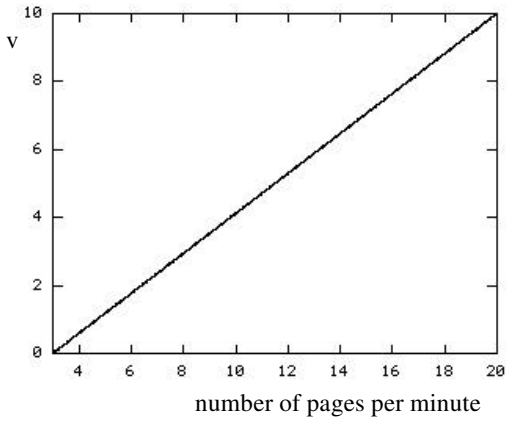

Figure 1: Example of a value function.

The complete evaluation process with MAUT is depicted in form of a value tree as sketched in Figure 2. Note, that it is possible to calculate for each attribute the impact on (or contribution to) the overall evaluation (cf. Section 3.2, 3rd application).

When using MAUT for modelling the user's interests, we assume that the evaluation of attributes with respect to various dimensions is fixed in a certain domain. So the information on the lowest level of the value tree, i.e. that part of the evaluation process that is located below the grey bar in Figure 2 does not dependent on the user's interests. But every weight w_i is user dependent and represents the *user's interest in dimension* d_i.

[2] The use of value functions is also common in the area of Case Based Reasoning. For example, by assuming that the preferences of people can be clustered, a similarity metrics is developped which can be used to assign a preference structure to a user which is derived from a user with similar preferences (cf. [HH98], [HHM01]). The drawback of such an approach is that preferences of an individual which are quite different from the preferences of the cluster members cannot be modelled properly.

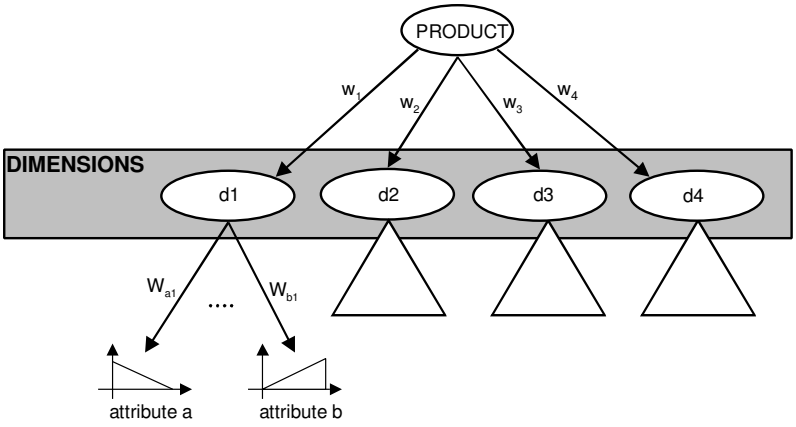

Figure 2: Sketch of a value tree.

3 An Example Scenario

For simplification, we use a domain with quite simple products. Assume that a company is selling pizza via internet. This domain is introduced in the following, to illustrate our approach.

3.1 The Pizza Domain

Every pizza is characterised by the following attributes:

- *Number of toppings:* We do not distinguish between the different toppings here.
- *Diameter of the pizza*: Different sizes from 25 to 30 cm are available.

In our example, we assume that the company has a clear business strategy and calculates the price of a pizza directly from the number of toppings and the diameter of the pizza as follows:

- A „non-topped" pizza with a diameter of 25 cm costs 2,90 €. If the pizza is bigger, it costs additional 0,10 € per cm. The largest pizza has a diameter of 30 cm.
- Every topping costs 0,30 €, but tomatoes and cheese are standard toppings, so a pizza must always have at least those two toppings. The maximum number of toppings is 8.

A pizza can be evaluated regarding the dimensions *economy* and *pleasure*. The cheaper a pizza is the better it scores on *economy*, the more toppings and the bigger the pizza the more it scores on *pleasure*.

Given the explicit pricing rules, it makes sense to use linear value functions, to represent the dependency between dimension *economy* and the attributes of the product. In the following sections we will use that kind of functions in this context. A more detailed discussion about the dimension *economy* is given in Section 4.3.

3.2 Three Applications of MAUT in the E-Commerce Context

We identified two general kinds of e-commerce systems for selling goods via internet: *Catalogue search* and *customisation/configuration*. In the catalogue search context, the user browses a catalogue of products which are offered. This search could be guided in some way, but the crucial point is that the products already exist in a prescribed form. If customisation or configuration is allowed, then the user cannot choose from a certain set of products, but has to specify values of the product's attributes. Those specifications are exploited by the system to generate (or configure) a new product (see also [Stu97]).

For example, in our pizza domain, a company could offer two different kinds of online services for selling their pizzas:

- When using the pizza catalogue, the user can choose between certain kinds of pizzas in certain sizes (see left-hand part of Figure 3). A pizza "Margherita" only has two toppings, "Funghi" has three toppings and so on.
- In the pizza configurator, the user can decide how big the pizza should be and which toppings she wants (see right-hand part of Figure 3).

Figure 3: The online offers of a pizza selling company.

Within this context we can classify three different, general applications of MAUT for user modelling and personalisation:

1.) *Modelling the user's interests* :

The question is how to interpret the user's behaviour and how to adapt the user model accordingly. This problem can be considered in both the catalogue and the configurator case.

A way to use MAUT for modelling the user's interests is presented in [JSS+95]. As a system can only estimate the user's interests with uncertainty, the representation has to be able to take uncertainty into account. In this way, the user's interests are represented in form of a probability distribution. Furthermore a Bayesian Network [Pearl88] was designed that represents the dependencies which are defined in the value tree. By interpreting a decision of the user with such a Bayesian Network, the user's interests are adapted.

2.) *Assisting the user searching a catalogue*:

In this case the system needs to evaluate the objects of the catalogue according to the user's interests and tries to help the user find the best one, e.g. by presenting an ordered list of the items with the highest evaluation.

When using MAUT this task can be solved quite easily by evaluating every item of the catalogue and ordering the items accordingly.

3.) *Assisting the user interacting with a configurator*:

When interacting with a configurator, the user has to specify values for attributes of the product. So, on the basis of an estimated user profile, a user-adaptive system should try to suggest values for any attribute.

This problem can also be solved using the MAUT approach. To do so, all the value functions that are defined for the currently regarded attribute have to be added and every function needs to be weighted according to the user's interest in that dimension. So let's assume now that I_a is defined as a set of indices, with $i \in I_a$ iff there is a value function $v_{ai}(p)$ defined for dimension d_i and attribute a. Then we can calculate the resulting overall value function v_a for attribute a as follows:

$$v_a(p) = \sum_{i \in I_a} w_i w_{ai} v_{ai}(p) .$$

The attribute value that maximizes the value of function v_a is the value that fits the user's interests best.

3.3 Naive Approach with Linear Value Functions

The simplest way for defining a value function is the linear case. If the domain of attribute a ranges from $a_{\min}$ to $a_{\max}$ then we can typically define the following two linear functions:

$$v_{ai}^{+}(x) = \frac{10 \cdot (x - a_{\min})}{a_{\max} - a_{\min}}, \quad v_{ai}^{-}(x) = \frac{10 \cdot (a_{\max} - x)}{a_{\max} - a_{\min}}$$

Please note that x replaces the expression $l(a)$, so x is the value of attribute a. The minimum (maximum) of all possible $l(a)$ is $a_{\min}$ ($a_{\max}$).

The left function $v_{ai}^{+}(x)$ represents the case that the evaluation of the minimum value of the attribute domain is worst (value 0), whereas the evaluation of the maximum value $a_{\max}$ is best (value 10), i.e. it represents the principle "*the bigger the better*". The right function $v_{ai}^{-}(x)$ represents the opposite case.

In our pizza domain we have to model the value functions for the two attributes *diameter* and *number of toppings* in relation to the two dimensions *economy* and *pleasure*. In the *economy* case we already stated that linear functions should be used. As the evaluation of both attributes with respect to *economy* gets worse for increasing values of the attribute, $v_{ai}^{-}(x)$ is chosen as defined above.

In this section we will also choose linear value functions[3] to represent the dependency between the two attributes and the dimension *pleasure*. In that case, the linear function needs to represent the fact that the evaluation for $a_{\min}$ is 0 and 10 for $a_{\max}$, so $v_{ai}^{+}(x)$ is chosen.

Now let's have a look at the effect of this decision to the three application scenarios as defined in Section 3.2:

1.) *Modelling the user's interests* :

The user's interest profile is adapted by choosing a probability distribution that fits the user's decision best. So if the system expects a high interest of the user in *pleasure* and low interest in *economy* for example, but the user chooses a small "Funghi", the user model is adapted in such a way that the interest in dimension *economy* increases and decreases for *pleasure*.

For this application linear value functions are often sufficient, because the deviation from an expected value is already represented here and can be interpreted by the system.

2.) *Assisting the user searching a catalogue*:

If all the chosen value functions are linear, the weighted sum of those functions is also linear. So the overall evaluation function $v(p)$ is again linear. This means that there is either no maximum (if the function is parallel to the axis of abscissae) or we get the maximum for an extremum of the input values.

This problem becomes obvious if we have a look at our *Pizza Catalogue* scenario. If the user has the same interest in both dimensions, every possible pizza has the same value. If the user is less interested in *economy*, the big "Palermo" is always evaluated best; if the user is more interested in economy, the small "Margherita" is preferred all the time.

3.) *Assisting the user interacting with a configurator*:

Here we have the same problem again as in the previous case: The value function v_a for a single attribute results from the weighted sum of linear functions. So we get a linear function again. The attribute value whose evaluation results in the maximum of that function v_a is the value that fits the user's interests best. But as the function is linear, the maximum is always located at an extremum of the attribute domain. For this reason, the *Pizza Configurator* always suggests to choose either two toppings and a diameter of 25 cm or eight toppings and 30 cm.

---

[3] We assume that there is a linear correlation between *pleasure* and the diameter of the pizza, although we are aware that the area of the pizza increases more than the diameter.

4 Refining the Value Functions

4.1 The Law of Diminishing Marginal Utility

The "*Law of Diminishing Marginal Utility*" (cf. [Sam80]) states that for any good or service, the marginal utility of that good or service decreases as the quantity of the good increases, ceteris paribus. In other words, total utility increases more and more slowly as the quantity consumed increases.

We illustrate this law in the food domain. For example, if you drink beer, the first sip you take will be the best one. For each additional sip, you will get some extra *pleasure*, but the more sips you take, the smaller the *pleasure* will be which is gained from an additional sip.

4.2 Value Functions Representing the Diminishing Marginal Utility

A value function defines the dependency between an attribute value and the utility for the customer with respect to a certain dimension. So a value function that represents the diminishing marginal utility has to realise the fact that the increase of the function becomes slower.

Logarithmic functions typically represent this relation. So $v_{ai}^{+}(x)$ and $v_{ai}^{-}(x)$ can be defined as follows:

$$v_{ai}^{+}(x) = \frac{10 \cdot \log(x - a_{\min} + 1)}{\log(a_{\max} - a_{\min} + 1)}, \quad v_{ai}^{-}(x) = \frac{10 \cdot \log(a_{\max} - x + 1)}{\log(a_{\max} - a_{\min} + 1)}$$

Another possibility to represent the diminishing marginal utility is using the sinus/cosinus function:

$$v_{ai}^{+}(x) = 10 \cdot \sin\left(\frac{\Pi \cdot (x - a_{\min})}{2 \cdot (a_{\max} - a_{\min})}\right), \quad v_{ai}^{-}(x) = 10 \cdot \cos\left(\frac{\Pi \cdot (x - a_{\min})}{2 \cdot (a_{\max} - a_{\min})}\right)$$

Both the logarithmic and the (co)sinus function, transferred to the pizza example's attribute "*number of toppings*" are shown in Figure 4. The left graph illustrates $v_{ai}^{+}(x)$ represented as logarithmic function, the right one displays the corresponding sinus function.

In the following we will analyse the impact of using such value functions in our application scenarios:

1.) *Modelling the user's interests* :

The user's interest profile is adapted in the same way as illustrated in Section 3.3. In this application, the system behaviour is qualitatively the same as for linear functions, but it is shifted a little bit in quantity.

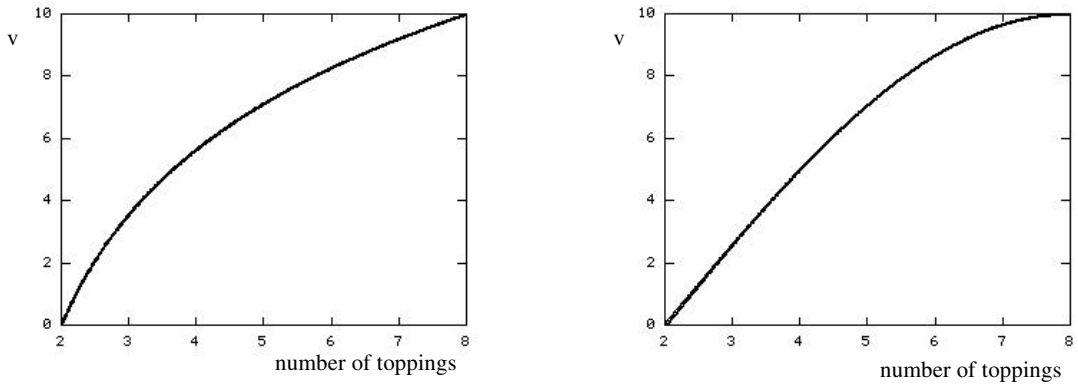

Figure 4: Value functions (left-hand logarithmic function and right hand sinus function) that represent the diminishing marginal utility.

2.) *Assisting the user searching a catalogue*:

Which item of the catalogue has the best evaluation mainly depends on the user's interests. As the overall evaluation function $v(p)$ is not linear but the weighted sum of linear and

logarithmic or sinus functions, the maximum value is somewhere between the extrema, depending on the chosen weights, whereas the high level weights represent the user's interests.

Table 1 shows the results of the *Pizza Catalogue* scenario with the newly introduced value functions. It can be seen that in our illustrative example, the items that are chosen seem to be a more realistic choice than the resulting choice when applying linear value functions.

			Recommendation with logarithmic value functions			**Recommendation with sinus value functions**		
User	estimated interest in		# of toppings	dia-meter	name of the pizza	# of toppings	dia-meter	name of the pizza
	pleasure	*economy*						
u_1	0.3	0.7	2	25	Margherita	2	25	Margherita
u_2	0.4	0.6	3	25	Funghi	3	25	Funghi
u_3	0.5	0.5	4	28	Due	5	28	Inferno
u_4	0.6	0.4	5	28	Inferno	6	28	Capricciosa
u_5	0.7	0.3	7	30	Palermo	6	30	Capricciosa

Table 1: The results of the *Pizza Catalogue* scenario with refined value functions.

3.) *Assisting the user interacting with a configurator*:

The value function v_a for a single attribute now results from the weighted sum of linear and logarithmic or sinus functions. The maximum of such a function depends on the weights that are chosen, because the weights determine how strongly the linear or the non-linear functions influence the resulting value function. The only user-dependent weights are the user's interests with respect to the involved dimensions. So if the interests change, the suggested values change.

Let's have a look at our example domain again. As the *Pizza Catalogue* contains many, items, that are evenly distributed on the two attribute domains, the suggestions of the *Pizza Configurator* are the same as in the catalogue case. This is why you can also read the suggestions of the configurator in Table 1. So for example the configurator suggests to have a pizza with a diameter of 25 cm and three toppings if the user is has interest of 60% in *economy*[4].

It is important to note that the value-functions which represent the diminishing marginal utility may not always be the best choice. There could be special situations that need to be described with other psycholgical laws or even with very individual value-functions. But using functions as presented in this paper, is one important step, because the law of diminishing marginal utility is well-analyzed in economics and seems to be an appropriate idealization to describe the user's evaluation process in many situations.

A very special dimension which has not yet taken into account is *economy*. Usually, it is one of the most import dimensions for customers when deciding whether to buy a good or not. Here, especially the price is very relevant. Details will be discussed in the next section.

4.3 Value Functions for the Dimension *Economy*

Economy is a very important dimension. It holds for nearly all products and dimensions that the better they are on a given dimension, the worse they are on *economy*. Usually, people are taking into account the price before buying a product, and very often they make a compromise between the evaluation on *economy* and the other dimensions (e.g. called "good price for value"). This underlines the importance of *economy*.

Most people associate *economy* with price. Price is indeed a very important aspect of *economy*. In the pizza domain, the attributes diameter and number of toppings determine the price and thus the evaluation of *economy*. However, there are other things besides price which influence *economy*, for

[4] It does not suggest a pizza "Funghi" here, since this name it not known to the configurator.

example running costs. If you buy a car, you will take into account the fuel consumption when calculating whether you can afford it. So, *economy* depends not only on price but to the total cost of ownership (TCO).

In order to make price and running costs comparable, you can use the interest rate to calculate a comparable value, e.g. by methods for discounting the interest. With this method a value can be calculated which comprises the price and all running costs during lifetime of the product which we call TCO. Deriving the TCO is not so simple as it relies on assumptions regarding the running costs and there is always the question about what interest rate to use. Here, some simplifying assumptions must be made.

A value function based on the TCO could be defined as follows: The cheapest alternative would yield 10 and the most expensive one 0. This at least will work probably fine if the cheapest alternative is nearly 0. However, recall our Pizza example. What if the cheapest pizza costs 4€ and each topping and each additional centimetre only 1 cent. This would be quite a different situation to the example described above, because usually people would no longer care much about the additional costs, that are negligible. For defining a proper value function, opportunity costs of the money spent have to be taken into account which is not a trivial task.

So, the construction of value functions for *economy* requires some insight into the application domain and assumptions about people's spending patterns and their opportunity costs.

5 Conclusion

In this paper we have shown how to represent the law of diminishing marginal utility when modelling the user's evaluation process. An extensive example and the application of the presented method to three different scenarios showed that the diminishing marginal utility needs to be represented to achieve a natural behaviour of a personalised system. A supplementing discussion showed the importance and complexity of modelling economical utility.

References

[HHM01] Ha, V., Haddawy P., and Miyamoto, J. (2001). *Similarity Measures on Preference Structures, Part II: Utility Functions.* In Proceedings of UAI 01, Seattle, Washington. Available via *http://www.cs.uwm.edu/~vu/papers/uai01.pdf.*

[HH98] Ha, V., and Haddaway, P. (1998). *Toward Case-Based Preference Elicitation: Similarity Measures on Preference Structures.* In Proceedings of UAI98.

[JSS+95] Jameson, A., Schäfer, R., Simons, J., and Weis, T. (1995). *Adaptive provision of evaluation-oriented information: Tasks and techniques*. In C.S. Mellish (ed.), Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pp. 1886 – 1893.

[Pearl88] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA.

[Sam80] Samuelson, P. A. (1980). *Economics (Eleventh Edition)*, McGrawHill.

[Stu97] Stumptner, M. (1997). *An overview of knowledge-based configuration*, in AI Communications, 10(2).

[WE86] Winterfeld, D. von, and Edwards, W. (1986). *Decision Analysis and Behavioral Research*. Cambridge, England: Cambridge UniversityPress.