Explicit Trade-off and Prospective Analysis in Electronic Catalogs

David Portabella Clotet[1] and **Martin Rajman**[2]

Abstract. We consider example-critiquing systems that help people search for their most preferred item in a large electronic catalog. We analyze how such systems can help users in the framework of four existing example-critiquing approaches (RABBIT, FindMe, Incremental Critiquing, ATA and AptDecision). In a second part we consider the use of several types of explicit passive analysis to guide the users in their search, specially in either underconstrained or overconstrained situations. We suggest that such a user-centric search system together with the right explicit passive analysis makes the users feel more confident in their decision and reduces session time and cognitive effort. Finally we present the result of a pilot study.

1 INTRODUCTION

A Multi-Attribute Decision Problem (MADP) is the problem of finding the best outcome or solution based on the user's preferences. However users may not have an accurate idea of their preferences while searching for a product, especially when the product is beyond the user's domain knowledge. They may begin the search with some vague set of preferences and refine them as they learn more about the different possibilities [6].

An example-critiquing interaction is an iterative process where the user and the system collaborate to find the best solution. Based on the current user's preference model, the system shows a set of candidate solutions and the user gives some feedback to the system so that it can update the preference model. The loop continues until the user is convinced with one of the candidate solutions. The type of feedback varies from system to system: it can be a direct manipulation of the preference model, such as adding a constraint such as "a maximum price of 400 Euros", or a more vague feedback such as "a cheaper apartment (than the proposed candidate)".

The rest of this paper is organized as follows: first we study four example-critiquing systems; then we justify the importance of Explicit Passive Analysis (EPA) and propose three types of EPA with informal examples; finally we present the results of a pilot study.

2 EXAMPLE-CRITIQUING SYSTEMS

One of the first systems to implement an example-critiquing approach was the RABBIT system [7], where the preference model is a query explicitly given by the user. This system is sometimes referred as of a *query-building interface*. The candidates shown to the user are simply the list of all items satisfying the query. The query can then be explicitly reformulated by using options such as "prohibit" or "specialize" over an attribute. When the solution space is overconstrained (i.e., there is no solution satisfying the query), the system simply does not show any candidate solution. In contrast, in a *preference-based approach*, the user would have the possibility to define a weight for each constraint and this information would be used to provide a list of ranked partial matches. The system is user-centric in the sense that it is the user who guides the interaction, the system just providing the requested information to the user.

In FindMe [1] there is no direct mapping between the feedback provided by the user and the preference model constructed by the system. The system limits the type of possible user feedbacks to a set of options such as "this apartment is OK, but make it bigger" based on one of the candidate solutions and thus, the approach is system-centric. The advantage of this approach is that the cognitive effort to provide feedback is lower as it doest not need to be very precise. As the interaction goes on, the system builds an implicit preference model using the feedback. In an overconstrained situation, the system performs a pre-defined domain-dependent ordered list of operations to relax the preference model. A problem with this approach is that the user may eventually not understand the justification for the proposed candidates, as the preference-model built by the system is completely hidden to the user who thus may feel frustrated.

The incremental-critiquing approach proposed by McCarthy at [5] shows just one candidate solution and a set of compound critiques that describe the feature relationships that exist between the remaining candidates. The user has the option of directly updating the query by adding or changing a constraint over an attribute, or by accepting one of the proposed compound critiques. The system selects the three best compound critiques with lower support values, as these would eliminate many items from consideration if chosen. This approach has the advantage that, if one of the compound critiques turns out to be interesting to the user, selecting it would probably require less cognitive effort than manually inspecting all the matching items, and extracting and applying the individual critiques one by one. The system builds the implicit user model incrementally, but there is no reason why compound critiques could not also be used in a query building tool as RABBIT.

One important feature proposed in the ATA system [4] and further elaborated in [2] is that the system not only presents the k best ranked items but also a set of alternative suggestions which are significantly different therefore and have the potential to stimulate the expression of hidden preferences, thus avoiding locally optimal solutions. For instance, if the user has not expressed a preference for non-stop flights, presenting an alternative solution that best satisfies the expressed preferences plus the latter one could potentially induce the user to realize that this new preference is important to him.

[1] École Polytechnique Fédérale de Lausanne (EPFL), david.portabella@epfl.ch
[2] EPFL, email: martin.rajman@epfl.ch

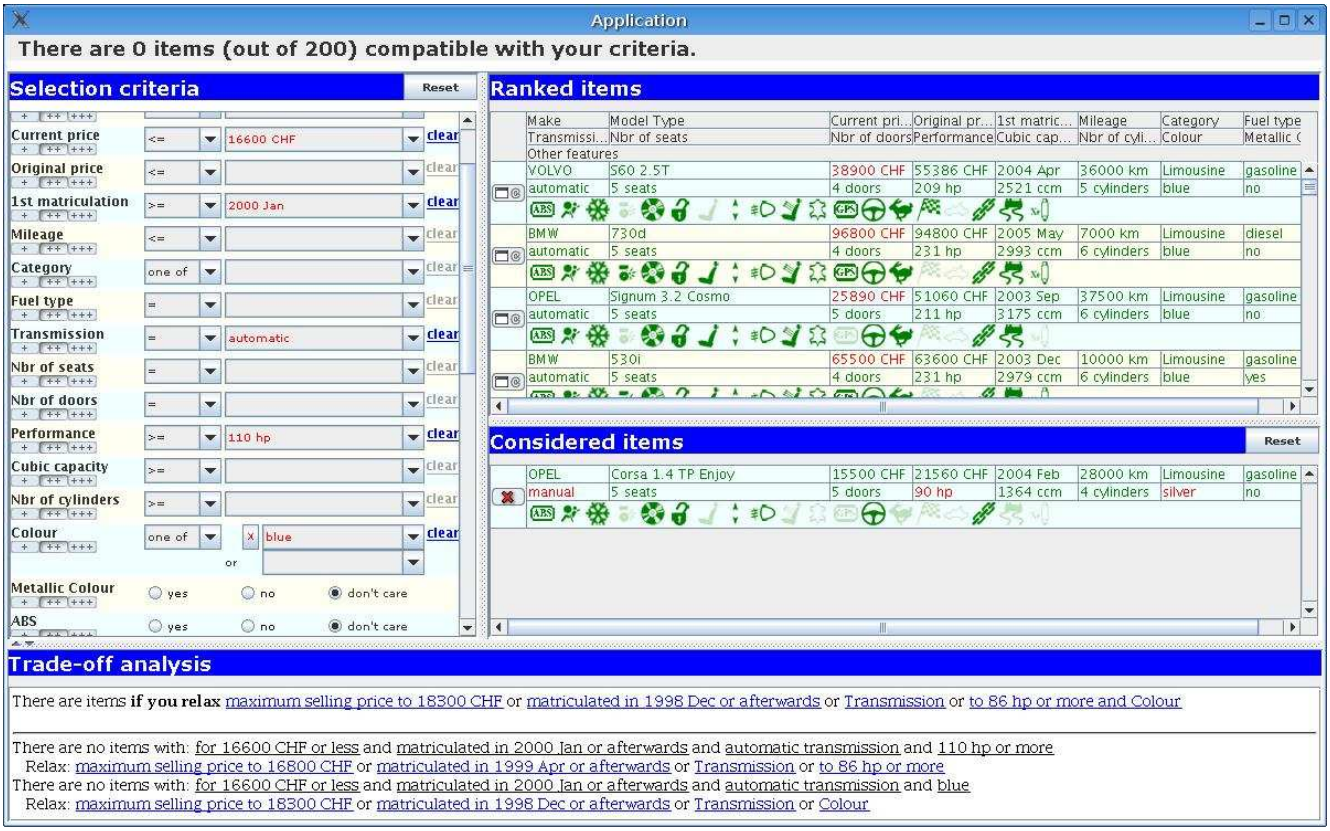

Figure 1. Screenshot of our user interface. At the left is the criteria selection window, where the user can specify search criteria and their weights (preference based approach). At the right is the corresponding ranked list of items (upper part) and the items currently being considered by the user (lower part). At the bottom is the Trade-off Analysis window, which displays the Minimal Conflicting Set Analysis (lower part) and the Possible Relaxations Analysis (upper part).

3 EXPLICIT PASSIVE ANALYSIS

Two types of approaches emerge from the four above presented cases: (1) the system-centric approach, which tries to build an implicit preference model or somehow limit the type of user feedback and the (2) user-centric approach, in which the user can directly update his query or preference model and the system just helps by providing some kind of passive analysis.

However, in all the reviewed systems except in the case of incremental-critiquing, the passive analysis when available is provided in the form of alternative solutions. We suggest that providing the users with some explicit explanation that can potentially lead to these alternative solutions will make them feel more confident in their decision process and speed up the interaction. For instance, in the example given for the ATA interface, the system could inform the user that, given the current preference model, he may potentially be interested in adding a preference about non-stop flights.

Indeed, we believe that a good way to increase user trust is to make users feel that they have reached a good characterization of the domain (at least in their region of interest). Thus, it is not enough that the system presents the best choice(s) to the user, but the user also needs to be convinced that the proposed choice(s) actually is the best. A natural way to achieve this is to compare a candidate solution with the related alternatives. In other words, we suggest that users use the example-critiquing iterative cycle not just to refine their preference model but also to browse and characterize their region of interest in the domain, i.e. they use the system as a query-building tool.

This browsing phase is even more important when preferences are not additive independent (see [3] for a detailed definition). In this case, the iterative process is also used to avoid expressing a complex preference model. Imagine for example that the user is interested in an apartment in the city center for less than 800 Euros or otherwise an apartment in the suburbs for a maximum of 600 Euros and close to a bus stop. Instead of directly entering this more complex preference model, he would probably first search for one of the cases by entering in an iterative process to refine the preference model and then switch to the other case and repeat the process, to finally select the best option between the two. In this situation we cannot say that the user was constantly refining his preference model, but rather that he studied two different cases separately.

Along these lines, we propose to use a query-building tool to let users browse and characterize their region of interest and to help them to do so by providing three types of explicit passive analysis about their current query, as described hereafter.

A basic form of such type of analysis is to let the user compare, while being in an underconstrained situation, the number of items that would match the query for different values of an attribute. We call this type of approach *Prospective Analysis*, as it provides information to the user about the impact of the different choices before he makes up his decision. For example, if the user has stated that he wishes cars for less than 6000 Euros, and now he is browsing the possible values for matriculation time, the system can indicate that there are 10 cars matriculated after 2005, 20 cars matriculated after

2004 and so on.

As users progressively add constraints, they will eventually end up in an overconstrained situation. Rather than then providing partially matching items, we propose to explicitly provide the user with the *Minimal Conflicting Set Analysis*. For instance, in a catalog of second-hand cars, given the current constraints expressed by the user, the system could indicate that "there are no Audi cars & Cheaper than 6000 Euros & Matriculated later than year 2000" nor "cars Cheaper than 6000 Euros & with Blue color & with Electric windows". Rather than just providing partial matching solutions, this information can potentially help the user to better understand the solution space.

A third type of explicit passive analysis is the *Possible Relaxations Analysis*, which provides concise relaxation suggestions to overcome the minimal conflicting set of constraints. While the information about the minimal conflicting set may be useful to characterize the solution space and thus get more confidence, finding which constraints to relax to overcome the situation is not evident for the user and it may involve trying several relaxation combinations. Explicitly providing a list of possible relaxations may lead to a speed up of the interaction. Notice that just proposing to remove some constraints as it is done in other approaches may not be very satisfactory. Indeed, in the case of a constraint over an ordered attribute such as price, providing the user with information about the minimal modification of that constraint can help the user to better characterize the domain and speed up the interaction. For instance, in the previous example, the system would also inform that there are cars "if the user relaxes the maximum price to 6500 Euros" or "if he relaxes the maximum price to 6200 Euros and without Electric windows". We call *Explicit Trade-off Analysis* the combination of the Minimal Conflicting Set Analysis with the Possible Relaxations Analysis.

4 PILOT STUDY

We have carried out a pilot study[3] evaluating our interface on two binary control variables: (1) whether or not to show the explicit passive analysis and (2) whether to use a query-building or a preference-based approach. The corresponding four interfaces were implemented using the same software so that their "look and feel" was as similar as possible to get a fair comparison. See Figure 1 for a screenshot.

As a database, we used an extract of 7000 second-hand cars kindly provided by the company Comparis[4], with 35 attributes for each car.

We have tuned our prototype so that it computes the minimal conflicting set for combinations of a maximum of four constraints. Notice that the selection of the appropriate value for this maximum deserves further investigation. Similarly, we have limited the maximum number of attributes in a relaxation suggestion to three.

Although the pilot study involved only four users, we were able to obtain some interesting preliminary results:

- Users using the query-building approach felt more in control about the search system and appreciated it.
- Users using the preference-based approach did hardly make use of the weights and were dissatisfied by the fact that they didn't understand how the items were ranked
- Users using the query-building approach were dissatisfied by the fact that no item was displayed in underconstrained situations.
- More importantly, users using the Explicit Trade-off Analysis (in both the query-building and the preference-based approach) were more confident with their choice, while there was no significant difference with the interaction time.
- An unexpected result was that the use Prospective Analysis together with Explicit Trade-off Analysis performed worse than using Explicit Trade-off Analysis alone. This may be due to the fact that the Prospective Analysis was inciting the users not to enter in an overconstrained situation, therefore making them lose the benefit of the Trade-off analysis.

Although still preliminary, these results are promising. We are therefore currently preparing a more extensive user study with about forty users.

[3] Our ECatalog software can be downloaded at http://icwww.epfl.ch/~portabel/ecatalogs as well as, for the sake of research only, an extract of the Comparis cars database

[4] http://www.comparis.ch/.

5 CONCLUSION

We have compared four existing example-critiquing systems for electronic catalogs both in terms of their user- or system-centric nature and in terms of the type of analysis provided to the user.

Based on this study, we suggest that providing an analysis in an explicit form rather than hiding it in a set of proposed alternative suggestions may greatly help the user to better understand the solution space, and therefore feel more confident about the final decision and speed up the interaction.

We have proposed the use of Prospective Analysis and Trade-off Analysis, the latter one consisting of the Minimal Conflicting Set Analysis along with the Possible Relaxations Analysis. We have built a concrete prototype and carried out a small pilot study to evaluate our approach. While Prospective Analysis lead to unexpectedly bad results, the Trade-off Analysis appeared as very promising. We now plan to carry out a user study with a larger number of subjects to confirm and precise the preliminary results obtained so far.

6 ACKNOWLEDGMENTS

We would like to thank Patrick Jermann for his help in preparing the user study. We also would like to thank Comparis and Johann Burkhard for sharing their database. Thanks also to the anonymous reviewers for their comments.

REFERENCES

[1] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young, 'Knowledge-based navigation of complex information spaces', in *AAAI/IAAI, Vol. 1*, pp. 462–468, (1996).

[2] Boi Faltings, Pearl Pu, Marc Torrens, and Paolo Viappiani, 'Designing example-critiquing interaction', in *International Conference on Intelligent User Interfaces*, pp. 22–29, Island of Madeira (Portugal), (January 2004). ACM.

[3] R.L. Keeney and H. Raiffa, *Decisions with multiple objectives: Preferences and value tradeoffs*, J. Wiley, New York, 1976.

[4] Greg Linden, Steve Hanks, and Neal Lesh, 'Interactive assessment of user preference models: The automated travel assistant', in *Proceedings, User Modeling '97*, (1997).

[5] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth, 'An analysis of critique diversity in case-based recommendation.', in *FLAIRS Conference*, eds., Ingrid Russell and Zdravko Markov, pp. 123–128. AAAI Press, (2005).

[6] J.W. Payne, J.R. Bettman, and J.E. Johnson, *The adaptative decision making*, Cambridge University Press, 1993.

[7] Michael David Williams, 'What makes rabbit run?', *Int. J. Man-Mach. Stud.*, **21**(4), 333–352, (1984).