Hippalus: Preference-enriched Faceted Exploration

Panagiotis Papadakos and Yannis Tzitzikas
Institute of Computer Science, FORTH-ICS, GREECE, and
Computer Science Department, University of Crete, GREECE
{papadako|tzitzik}@ics.forth.gr

ABSTRACT

In this work we describe and evaluate `Hippalus`, a system that offers exploratory search enriched with *preferences*. `Hippalus` supports the very popular interaction model of Faceted and Dynamic Taxonomies (FDT), enriched with user actions which allow the users to express their *preferences*. The underlying preference framework allows expressing preferences over attributes (facets), whose values can be hierarchically valued and/or multi-valued, and offers automatic conflict resolution. To evaluate the system we conducted a user study with a number of tasks related to a "car selection" scenario. The results of the comparative evaluation, with and without the preference actions, were impressive: with the preference-enriched FDT, all users completed all the tasks successfully in 1/3 of the time, performing 1/3 of the actions compared to the plain FDT. Moreover all users (either plain or expert) preferred the preference enriched interface. The benefits are also evident through various other metrics.

1. INTRODUCTION

Users access large amounts of information resources (documents or data) mainly through search functions, where the user types a few words and the system (web search engine, query engine) returns a linear list of hits. While this is often satisfactory for focalized search, it does not provide enough support for recall-oriented (exploratory) information needs. As several user studies have shown, a high percentage of search tasks are *exploratory* ([1]), the user does not know accurately his information need (e.g. in WSE users provide in average 2.4 words [4]), and such needs cannot be satisfied by a single 'hit'.

A highly prevalent model for exploratory search is the interaction of Faceted and Dynamic Taxonomies (FDT), which allows the user to get an overview of the information space (e.g. search results) and offers him various groupings of the results (based on their attributes, metadata, or other dynamically mined information). These groupings enable the user to restrict his focus gradually and in a simple way (through clicks, i.e. without having to formulate queries), enabling him to locate resources that would be difficult to locate otherwise (especially the low ranked ones). This model is currently used in various domains: e-commerce (e.g. eBay), booking applications (e.g. booking.com), library and bibliographic portals (e.g. ACM Digital Library), museum portals like Europeana, mobile phone browsers, and many others.

The enrichment of search mechanisms with preferences could be proved useful for recall-oriented information needs, because such needs involve decision making. However the current approaches for preference-based access [13], mainly from the area of databases, seem to ignore that users should be acquainted with the information space and the available choices for describing effectively their preferences. On the other hand, the available personalization services over FDT, do not allow the explicit expression of preferences, but try to automatically suggest the most preferred facets and values according to a number of different criteria. In this way the user somehow "loses" the control of the interaction.

In this work, we describe and evaluate `Hippalus`, a preference enriched FDT system, for exploratory browsing. Its functionality is founded on the preference framework described in [15], whose distinctive features is the ability to express preferences over attributes whose values can be hierarchically organized, and/or multi-valued, while scope-based rules resolve automatically the conflicts. We conducted a user study of the `Hippalus` system, over a number of tasks, with and without the preference actions. The gathered results were impressive. Even though the available choices were few (50 cars), with the preference-enriched FDT all users completed all the tasks successfully in 1/3 of the time, performing 1/3 of the actions compared to the plain FDT. Moreover all of the users (either plain or expert) preferred the preference enriched interface.

2. BACKGROUND & RELATED WORK

Faceted and Dynamic Taxonomies

Modern environments should guide users in exploring the information space and in expressing their information needs in a *progressive* manner. Systems supporting FDT offer a simple, efficient and effective way for exploratory tasks [12]. *Dynamic taxonomies* (faceted or not) is an interaction framework based on a multi-dimensional classification of may heterogeneous data objects allowing users to browse and explore the information space in a guided, yet unconstrained way through a simple visual interface. Features of this framework include: (a) display of current results in multiple categorization schemes (called facets - or just attributes), (b) display of facets and values leading to non-empty results only, (c) display of the count information for each value (i.e. the number of results the user will get by selecting that value), and (d) the user can refine his focus gradually, i.e. it is a session-based interaction paradigm in contrast to the stateless query-and-response dialogue of most search systems. Moreover, and as shown in [10, 7], this interaction paradigm can act complementarily to the traditional query-and-response dialogue, by post-processing and post-exploring the results returned by a classical search system.

In any case, the user explores or navigates the information space (either the entire information base, or the search results), by setting and changing his *focus*. The notion of focus can be *intensional* or *extensional*. Specifically, any conjunction of values (or any boolean expression of values in general) is a possible *focus*. For example, the initial focus can be the empty, or the top term of a facet. However, the user can also start from an arbitrary set of objects, e.g. the search results returned by a common WSE. In that case we can say that the focus is defined *extensionally*.

FDT and Preferences

Most FDT systems output facets and zoom-points in lexicographical order, or order facets and zoom-points based on the number of indexed documents. Other systems, like eBay, only present a manually chosen subset of facets to the users, and the zoom-points are again ranked based on the number of indexed documents.

Recently, various approaches try to automatically present the most "useful" facets and zoom-points according to various criteria like *set-cover ranking* of indexed objects [2], *interestingness* over a number of criteria [3], or use *collaborative* [8] and *content* filtering [14] to rank facet-values pairs. Minimum-effort driven navigational techniques for enterprise databases, that rapidly drill down to the most prominent tuples are described in [11]. In the same manner, but for zoom-points, [5] propose a system for faceted navigation using a cost model of user navigation. A browsing-oriented approach for facet ranking and grouping of facets and their values according to different intuitions and metrics is provided in [16]. There are various other works, discussed in [9]. But none of them allows users to explicitly express their preferences during the exploration process.

To the best of our knowledge the only model that allows users to define explicitly the desired preference structure in a *gradual* and *flexible* manner, also exploiting attributes with *hierarchically organized values* and possibly *set-valued*, is the one proposed in [15]. In this paper we describe and evaluate the `Hippalus` system, which supports the above framework.

3. THE HIPPALUS SYSTEM

`Hippalus` is a publicly accessible web system[1] demonstrating a preference-enriched FDT-based exploratory process. It offers actions that allows the user to order facets, values, and objects using *best*, *worst*, *prefer to* actions (i.e. relative preferences), *around to* actions (over a specific value), or actions that order them lexicographically, or according to their values or count number. Furthermore, the user is able to compose object related preference actions, using *Priority*, *Pareto*, *Pareto Optimal* (i.e. skyline), and *Combination* (i.e. order according to priority; the rest actions are the least prioritized and use *Pareto* composition) compositions. All the above functionality is offered in an efficient way, by using the algorithms described in [15].

The information base that feeds `Hippalus` is represented in RDF/S (using a schema adequate for representing objects described according to dimensions with hierarchically organized values). For loading and querying such information `Hippalus` uses Jena[2], a Java framework for building Semantic Web applications. `Hippalus` offers a web interface for FDT exploration, enriched with the aforementioned preference actions through HTML 5 *context menus*[3]. The performed actions are internally translated to statements of the preference language described in [15], and are then sent to the server through HTTP requests. The server parses the preference statements and if they are valid, computes the respective preference bucket order. Finally, the resulting according to preference ranked list of facets, terms or objects is sent to the user's browser.

3.1 Interaction and User Interface Design

The most widely adopted approach or policy for FDT visualization (evidenced by the UI design of global systems like booking.com, eBay), is to use a left bar for the facets and the corresponding zoom points. This is also the case for the `Hippalus` system (Figure 1.a[4]). `Hippalus` displays the preference ranked list of objects in the central part of the screen, while the right part is occupied by information that relates to the information thinning process (object restrictions), preference actions history and preference composition. It offers the preference related action through right-click activated pop-up menus (through HTML5 context menus). This policy does not require allocating permanent screen space for these actions. However the user should be aware that these options exist. The design of the preference actions, includes actions that are anchored to one element, and this makes the right-click activated actions straightforward. Moreover, the proposed preference-based framework supports also actions that concern two elements, i.e. relative preferences like $Korean \succ European$. Figure 1(c) shows how such statements can be expressed through a context menu: the action is anchored to *Korean* and the available menu options guide the user through the options that are valid in this specific situation and the specific user focus. Notice the icon in the *European* option in the right most menu. By pressing it, the preference is recorded.

At any time the user can restrict his focus to any hard constraint (i.e. the information thinning process), and his soft constraints (i.e. expressed preferences) will be applied to the current restriction of the object set.

Finally, since the number of objects can be very large, the user can specify a threshold, so that preferences are applied only when the number of objects is reduced under this threshold[5]. Options and parameters regarding the system functionality can be set through a drop-down menu (i.e. simple or full support of preference menus, threshold, etc.)

[1]http://www.ics.forth.gr/isl/Hippalus

[2]http://jena.apache.org/

[3]Available only to firefox 8 and up.

[4]In the following screens, the underlying information base contains data about 50 cars, as described in Section 4.

[5]The user can reduce the number of objects by selecting facets and zoom-points, restricting his focus.

Regarding the description of the current state, the user is able to view not only the intentional description of his current state, but also the accumulated preferences that he has formulated. Finally, the user is able to store and load his preferences, since exploration is a time depth process.

3.2 Interaction Example

Here we describe a more complete scenario demonstrating how hard and soft constraints can be specified by the user, in an easy and gradual manner. It also aims at making clear the merits of the underlying preference framework (preference inheritance and scope-based conflict resolution). A video showcasing this scenario is available online[6].

The first screen (Fig. 1.a) shows the 50 cars and the left bar shows the attributes, their values (which can be hierarchically organized), accompanied by the number of their occurrences. Figure 1.b shows that one can expand broad values, like `Asian` (from the attribute `Manufacturer`), and that by clicking on the value `Korean` the focus is restricted on three Korean cars. Notice that the left bar has been updated, i.e. only the values that appear in the restricted set are presented (all attributes have count up to 3). With additional clicks the user can further reduce the focus, e.g. from the attribute `Fuel Type` we can see that one of the cars consumes `Diesel` and two cars `Gasoline`. By clicking on `Gasoline` we see these two cars and by mouse over one of them the user gets its "Object Card" showing all attributes of that car. At the right bottom frame the user can see the history of his clicks and can undo any click.

Preferences are activated through right-click menus. Suppose we cancel all clicks and assume that we want to express that we prefer Korean cars to European. This means that we do not want to see only Korean; we just want to get them ranked higher than European. This is shown in Figure 2.a(top) where we see that now the user is getting a linear list of blocks of equally preferred objects, here the first contains Korean cars, the next one European (thanks to inheritance the user does not have to say anything about German, Italian, French, etc).

It is important that preferences can be expressed incrementally and at any point during the interaction. For example suppose that in addition the user prefers prices around 12,000. He can use the action `around 12,090` as shown in Figure 2.a(bottom). We can see that the object order now becomes more refined (the figure shows 14 blocks). Notice that the first block contains one Korean (`Hyundai`) and one `Fiat`. This happens because both of his preference actions have the same priority (and Fiat is closer to 12090). If the user wants to give higher priority to one preference he can use preferences composition tool at the right frame. Figure 2(b) shows the object order obtained after expressing that the preferences over manufacturers have higher priority than the preferences over prices.

At any time the user can click on a value from a facet to restrict the current focus, which is now a preference-based list of cars. For instance, if the user wants to see only cars having two doors, he can click on `2` in the attribute `Doors`. We can see that now he gets only 8 cars, which are ranked according to his preferences so far. The user could cancel this extra restriction from the object restriction history.

In general the user can combine object restriction (or relaxation) actions and preference actions in any order.

[6]http://www.youtube.com/watch?v=Cah-z7KmlXc

4. EVALUATION

The objective was to investigate whether even in a small dataset (50 cars), the addition of preferences to FDT would make the users more effective and satisfied, without making the interaction complex to use or learn. To this end we compared two different UIs: a) `Hippalus` system with exploration and browsing capabilities only, where preference functionality was disabled (UI_1) and b) `Hippalus` system with exploration and browsing capabilities and preference functionality enabled (UI_2). Regarding UI_2, we configured `Hippalus` to provide only preference actions affecting objects (i.e. users were not able to express preferences regarding attributes and their values).

Information Base

We used an information base of 50 cars, where each car is described by 23 attributes, as shown in Fig. 1.a. A number of attributes have hierarchically organized values like `Manufacturer` and `Drive_System`, while the rest like `Doors` `Year`, `Price`, etc. are flat.

Tasks

We created two variations of equal tasks for the plain and two equal variations for the expert users of the evaluation. In our context task equality is defined as tasks that consist of the same kind of preference actions and criteria. For each task, the first subtask was designed around prioritized composition of preference actions, while the second one over *Pareto* composition. Plain users tasks used only 3 criteria, while the expert ones were more difficult and complicated, using a total of 6 different criteria. The tasks that users completed are available in [9, Chapter 6].

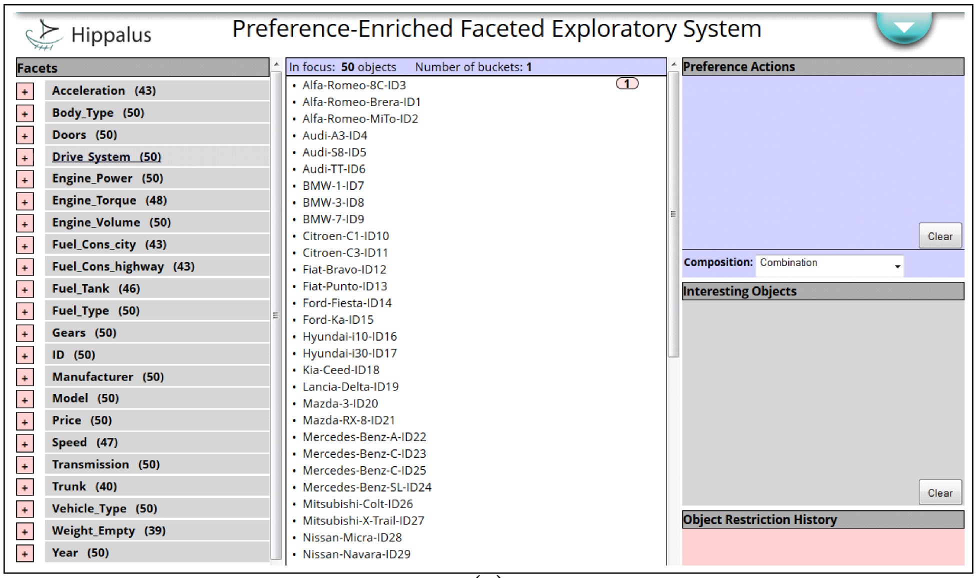
Participants

26 males and females of varying age (i.e. between 23-43 years) and expertise (i.e. tertiary education - PhD level) participated in this study. We formed two groups. The first group, named *plain* users, consisted of 20 regular users, while the second one, *expert* users, consisted of 6 people with a prior experience in using multi-dimensional browsing and access systems that support preferences. Before starting the evaluation, users were given a simple tutorial of 15 minutes. In more details, initially users were given a description of the information base (domain, attributes). In the next five minutes they were described the interactive process of information thinning and finally the rest of the tutorial demonstrated the preference actions by showing specific examples. Finally, users were allowed to get acquainted with the UI and complete a number of simple tasks.

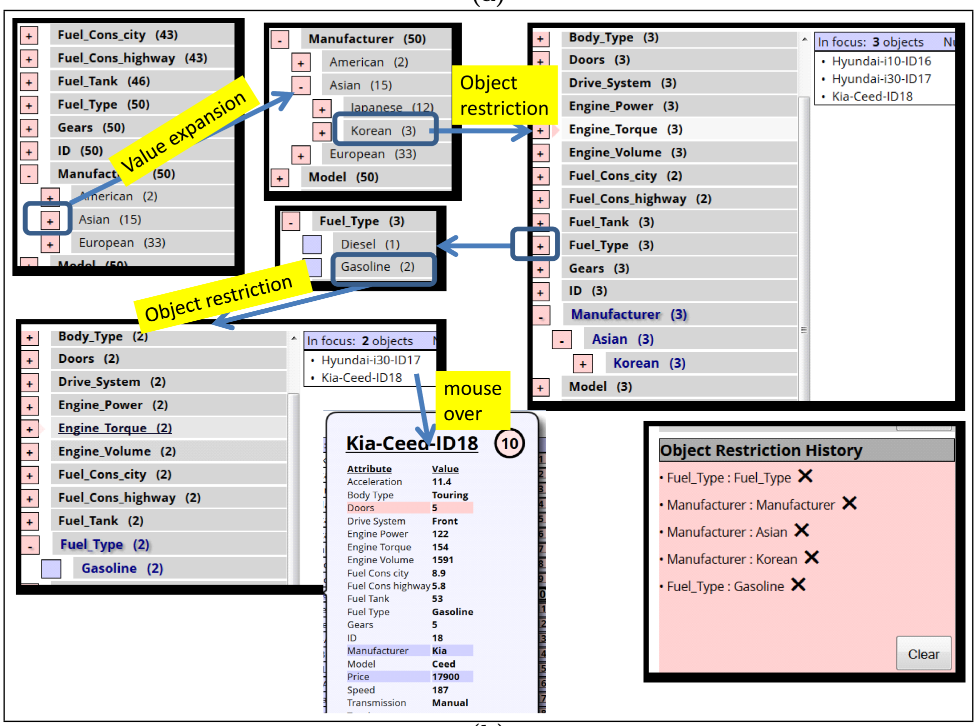
Evaluation

The users were asked to evaluate UI_1 and UI_2 using the previously described tasks. Regarding UI_1, users completed the tasks by using the available information thinning functionality to restrict their focus and by inspecting the available cars. For UI_2, on top of the information thinning functionality they could also submit preference actions. For both UIs, the users provided the set of cars which they believed fulfilled the needs of each task, by drag-&-dropping cars into the "Interesting Objects" frame in the right middle part of the system (Fig. 1.a).

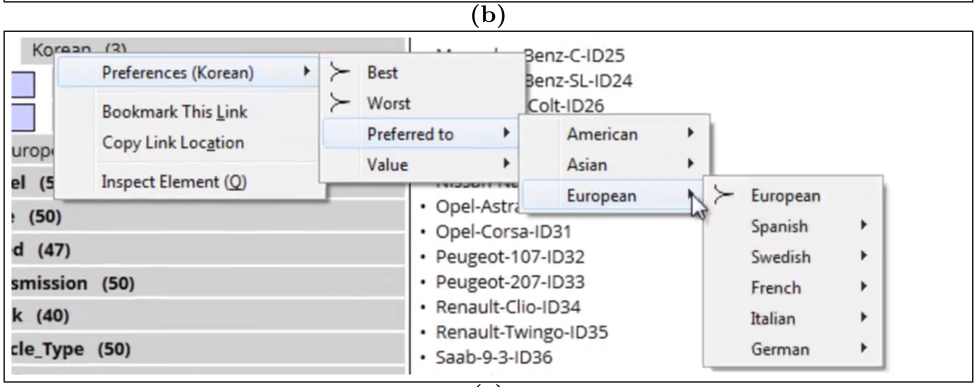
In order to control for order effects and to increase the chance that results can be attributed to the experimental



(a)



(b)



(c)

Figure 1: (a) Initial screen, (b) Value expansion - object restriction, (c) Relative preference $Korean \succ European$

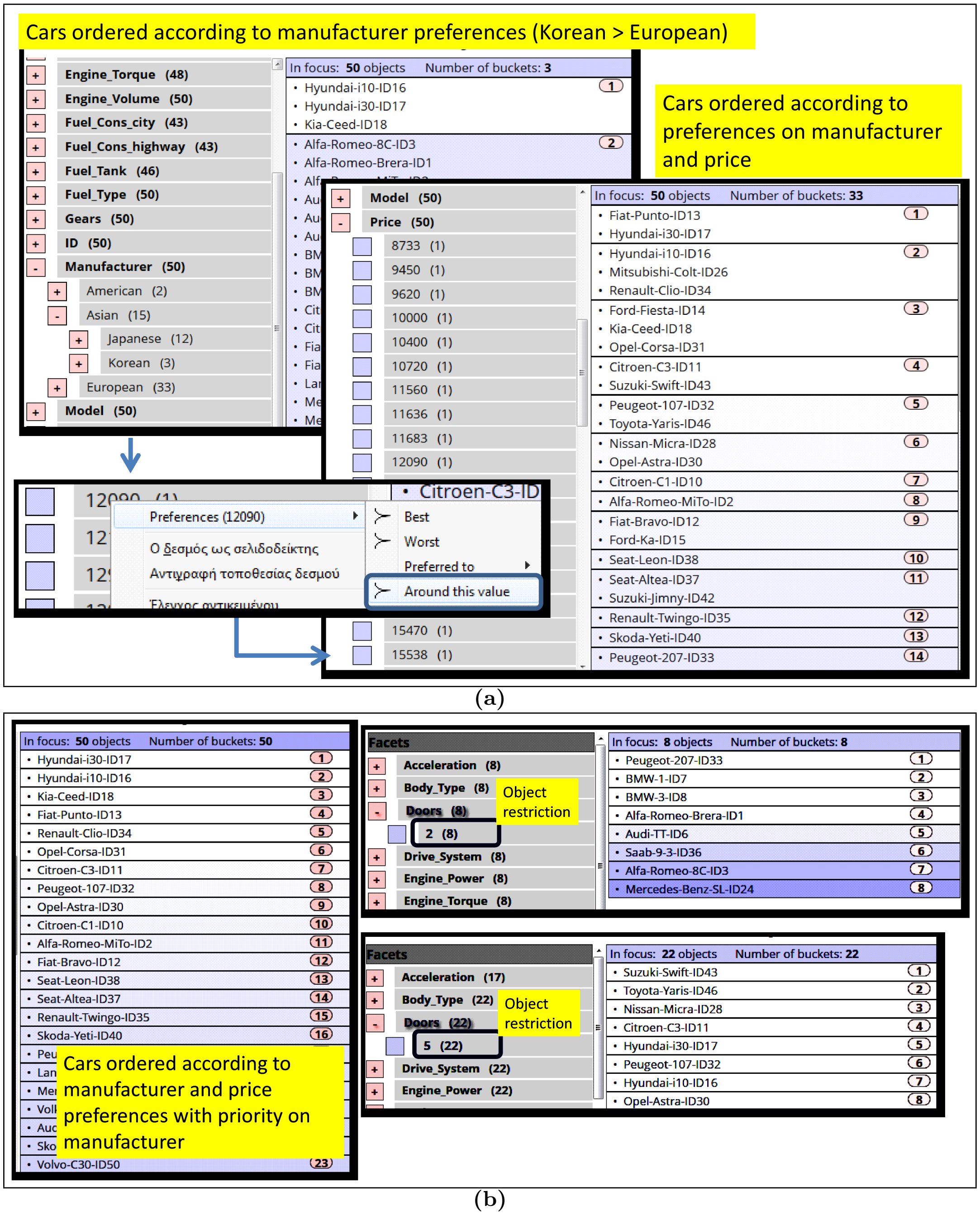


Figure 2: (a) Expressing preferences, (b) Object restrictions after preference expression

treatments and conditions, we used rotation and counterbalancing [6]. Specifically, we used a Graeco-Latin Square Design, rotating both the order of tasks and the order in which subjects experience the interfaces. For each task, an expert user provided the ordering of the collection according to preference by using the Hippalus system. The order was a bucket order (i.e. two cars can be equally preferred).

The users provided scores for the two UIs regarding *Ease of use*, *Usefulness*, *Preference* and *Satisfaction*, using a psychometric Likert scale from 1 to 5. We also calculated *Recall* (i.e task completeness), *Precision*, and *Average Precision* of the answer set, along with *Efficiency* (time to complete a task) and *Number of Actions* per each task using the logged data. Finally, users were asked explicitly if they prefer U_2 over U_1. In case the answer was 'Yes', they were asked how much more useful they found U_2 over U_1 (*very much*, *much*, *enough*, or *little*).

Results

Here we synopsize the main results. All plain and expert users preferred the preferences UI over the plain one. Specifically, 75% of the 20 plain users found the UI_2 to be *very*

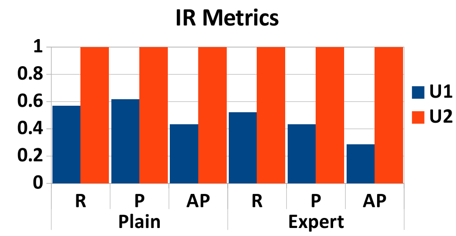


Figure 3: Average values in last step of each task for Recall (R), Precision (P) and Average Precision (AP)

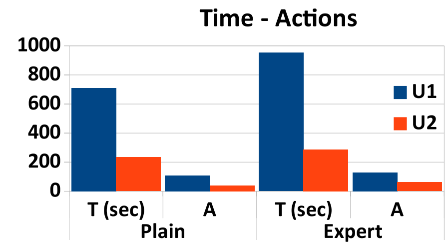


Figure 4: Average timings (T) and actions (A) per task

much , 20% *much* and only 5% *enough* more useful than UI_1. The respective results for the 6 expert users are 50% *very much* and the other 50% *much* more useful. The preference-enabled UI, allowed users to *complete all the tasks successfully*, in average less than a *third of the time* and with a *third of user interactions* compared to the plain FDT UI (Fig. 3 and 4). Furthermore, *none* of the users was able to complete both of the tasks successfully with the plain UI. As a result we *verify* the conclusions of the theoretical user effort analysis in [15], since the preference-based UI helps users to find the desired results in less time and with fewer actions and less decisions. More details and a theoretical analysis of the user effort, decision cost and the gathered results are available in [9, Chapter 6].

5. CONCLUSION

In order to support decision making tasks, exploratory search requires a session-based behaviour that provides informative overviews. FDT is a widely used interaction model and in this paper we have presented an extension of this model with preferences. The enriched interaction is simple for the users, since it is mainly based on clicks over the presented values. In addition, the underlying preference framework is perfectly suited to FDT since it exploits the semantics of hierarchically organized values and automatically resolves any conflicts. This reduces the number of preference actions the users have to express and the dialogue is kept simple and clean (from technicalities). The `Hippalus` system demonstrates the feasibility of this extension. The results of the conducted user study were very satisfying: with the preference-enriched FDT all users completed all the tasks successfully in 1/3 of the time, performing 1/3 of the actions compared to the plain FDT.

Acknowledgments

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operation Program "Education and LifeLong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Herakleitus II. Investing in knowledge society through the European Social Fund.

It was also partially supported by the PlanetData NoE (FP7:ICT-2009.3.4, #257641).

6. REFERENCES

[1] D. Crawford, editor. *Supporting Exploratory Search*, volume 49. ACM, New York, NY, USA, 2006.

[2] W. Dakka, P. Ipeirotis, and K. R. Wood. "Automatic Construction of Multifaceted Browsing Interfaces". In *Procs of CIKM'05*, pages 768–775, Nov. 2005.

[3] D. Dash, J. Rao, N. Megiddo, A. Ailamaki, and G. Lohman. "Dynamic Faceted Search for Discovery-Driven Analysis". In *Procs of CIKM'08*, 2008.

[4] H. Inan. *"Search Analytics: A Guide to Analyzing and Optimizing Website Search Engines"*. Book Surge Publishing, 2006.

[5] A. Kashyap, V. Hristidis, and M. Petropoulos. "FACeTOR: Cost-Driven Exploration of Faceted Query Results". In *Procs of CIKM'10*, pages 719–728. ACM, 2010.

[6] D. Kelly. "Methods for Evaluating Interactive Information Retrieval Systems with Users". *Foundations and Trends in Information Retrieval*, 3(1-2):1–224, 2009.

[7] I. Kitsos, K. Magoutis, and Y. Tzitzikas. Scalable entity-based summarization of web search results using mapreduce. *Distributed and Parallel Databases*, 2013.

[8] J. Koren, Y. Zhang, and X. Liu. "Personalized Interactive Faceted Search". In *WWW'08: Procs of the 17th International Conference on World Wide Web*, pages 477–486, New York, NY, USA, 2008. ACM.

[9] P. Papadakos. *"Interactive Exploration of Multi-Dimensional Information Spaces with Preference Support"*. PhD thesis, University of Crete, November 2013. Available at http://www.ics.forth.gr/_publications/Papadakos_Dissertation.pdf.

[10] P. Papadakos, N. Armenatzoglou, S. Kopidaki, and Y. Tzitzikas. "On Exploiting Static and Dynamically Mined Metadata for Exploratory Web Searching". *Knowledge and Information Systems*, 30(3):493–525, 2012.

[11] S. B. Roy, H. Wang, G. Das, U. Nambiar, and M. Mohania. "Minimum-Effort Driven Dynamic Faceted Search in Structured Databases". In *Procs of CIKM'08*, pages 13–22, 2008.

[12] G. M. Sacco and Y. Tzitzikas, editors. *"Dynamic Taxonomies and Faceted Search: Theory, Practise and Experience"*. Springer, 2009.

[13] K. Stefanidis, G. Koutrika, and E. Pitoura. "A Survey on Representation, Composition and Application of Preferences in Database Systems". *ACM Transactions on Database Systems*, 36:19:1–19:45, August 2011.

[14] M. Tvarožek, M. Barla, G. Frivolt, M. Tomša, and M. Bieliková. "Improving Semantic Search Via Integrated Personalized Faceted and Visual Graph Navigation.". In *SOFSEM*, volume 4910 of *Lecture Notes in Computer Science*, pages 778–789. Springer, 2008.

[15] Y. Tzitzikas and P. Papadakos. Interactive exploration of multi-dimensional and hierarchical information spaces with real-time preference elicitation. *Fundamenta Informaticae*, 122(4):357–399, 2013.

[16] A. Wagner, G. Ladwig, and T. Tran. "Browsing-Oriented Semantic Faceted Search". In *DEXA (1)*, pages 303–319, 2011.