Design Considerations for Faceted Search: Literature Review and Case Study

Steve Berry
University of Texas School of Information
4702 Fieldstone Drive
Austin, TX 78735 USA
+1 512 914 9994
sberry@unideal.net

ABSTRACT
This paper reviews the current available literature concerning faceted search to provide a research overview of practices and guidelines for developing effective faceted search interfaces. These findings are then used as the basis for a case study addressing the design of a faceted search interface for classic cars. We find these concepts useful when selectively applied duting the construction of a practical faceted search application for a specific dataset and specific user needs.

Keywords
Information Retrieval, Faceted Search, Metadata categories

1. INTRODUCTION
Faceted search is a term that describes information retrieval interfaces which expose query filtering refinements alongside traditional search results. These interfaces combine the traditional 'browse' and 'keyword' search paradigms into a single, merged user interface. The resulting interactive interface affords users the ability to easily constrain and expand results of search queries by navigating refinements menus of 'facets'. These facets are derived based on either explicit or dynamic underlying ontologies in the source data being searched.

For the purposes of this review we've broken down the underlying components of faceted search into four primary building blocks as follows: Initially we review methods for creating a faceting ontology followed by an analysis of data types typically represented in facets. We then review some interaction patterns and visual design considerations for assembling the user interface.

Finally, these concepts, best-practices and design recommendations were then applied to an interface for searching a specific document set containing classic automobiles.

2. FACET ONTOLOGIES

2a. Deriving fixed facet ontologies from existing data
Faceted search is driven by the ability to expose appropriate categories or metadata as a 'faceted' drill down menu. Creating ontologies that populate these facets is highly context-dependent on the target dataset. For this reason it's difficult to lay out a series of best practices for the creation of facet ontologies. Instead, it helps to explore relevant literature for methods which enable successful ontology designs.

Faceted search interfaces have appeared more frequently in commercial and academic settings in the past decade and are generating a growing body of research concerning their creation. Facet search should offer users an overview of available documents and help them avoid empty result sets and also to bring order to what may otherwise be an overwhelming array of results. Additionally, facets should help users discover and navigate difficult inherent terminology in searchable material [5].

A static, explicit set of facets may most easily be derived from the existing structure of the underlying dataset. The processes for arriving at a serviceable facet ontology are in most cases suggestive from the schema of the data combined with a common-sense evaluation of how users might wish to query that data. A dataset concerning retail items, for instance, might focus on manufacturer, price or sizing as important facets. A similar interface applied to textual archive data might instead choose to expose publication source, the date the item was cataloged or the type of document represented. Whatever the application, it's extremely important from the outset to investigate how users intend to search for and apply the data they find [5] and use this as a basis for defining facets categories which best enable those activities.

2b. Intermediary Ontologies
In cases where metatdata assignment within the source dataset is sparse or poorly applied, designers may find that

the existing data needs augmentation or intermediary levels of categorization to enable appropriate faceting. In certain complex domains, such as legal and medical databases, users may benefit from the creation of an explicit intermediary ontology that bridges an end-user's terminology or categorization model with the pre-existing classifications of the source data [12]. Term mismatch is a persistent problem in many information retrieval tasks and faceted search is no exception [5]. Ideally, intermediary ontologies would be constructed using a combination of standardized ontologies common to semantic technologies and through user-centric research such as open card sorting sessions with representative end users.

Throughout this process it is possible that conflicts may be encountered between the intermediary ontology and the goals of the original data scheme so care must be taken to carefully identify and reconcile these issues when mapping new ontologies back to the source data [12]. The ultimate goal of an intermediary ontology should be to present searchers with intuitive categories and organization targeted at their specific vocabularies in order to reduce the amount of time needed to satisfy a target user's information needs.

2c. Dynamic Category sets

Another proposed method of solving the same "Vocabulary Problem" is through use of Dynamic Category Sets. This approach uses information retrieval matching practices such as spell correction, stop words and thesaurus expansion to dynamically parse and map elements of search phrases to underlying categories which may then be combined into new dynamic categories at search time [13]. These techniques can help mitigate vocabulary mismatches between searchers and information providers, using matching semantics to allow users to create facet intersections and avoid empty result sets. Dynamic Category Sets are effective but may also greatly increase configuration complexity, overload user models and tax performance of the search system.

3. FACET DATA TYPES

The simplest type of facet data may have a flat relationship across a dataset such as 'date created' or 'document type', attributes which exist independently of other facets and are uncomplicated by hierarchical semantic relationships. More complex facets may contain hierarchical dependencies on parent facets such as the dependent relationship between "Make" and "Model" in a database of cars or deeper nesting in the case of "Country", "State" and "City" attributes commonly found in geographical facets. [5].

The types of data exposed by facets may vary depending on the underlying elements of the document model they represent. They usually fall within five commonly used types: Nominal values, such as the label for a category; Ordinal values, a discrete ranking value that may be ordered such as a number or text-based rating; Interval type values, non-zero value representing a continuous range of values; Ratio type values, similar to an ordinal value but allowing zeroes; and the Free-text type, which usually holds text snippets not well suited to the other stated facet types [8].

Beyond these basic facet types, compound or computed facets may be utilized to enhance searching. A "correlated" facet type is based on multiple underlying multi-valued fields which may only exist in particular combinations such as "Size" and "Color" for an item in a retail dataset [3]. Correlated facets may be created to express the intersections of certain non-exclusive facets and avoid both empty result sets and unnecessary multiple refinement actions.

Computed information such as the average rating or average price of a given item can be created at search time and appended to facet values to add additional information scent, guide emphasis or influence order of value presentation. Dynamic multi-dimensional facets of this kind may even be appropriate for replacing some existing OLAP interfaces which essentially present the same computed data views [3].

Existing technologies allow incredible flexibility in the types of data and levels of information density which may be expressed in facets. Prudence and analysis of the information needs of representative users should be the overriding design criteria for implementing complex facets.

4. INTERACTIONS

Facets provide more than just a low-cost query refinement mechanism. They are also a valuable opportunity to reveal aggregate information about the contents of the overall document structure and familiarize users with the provider’s categorization strategy. The net effect of a properly rendered facet menu should act as an additional discovery mechanism for the user [3]. Careful labeling and facet presentation allow searchers the opportunity to recognize terms and labels rather than summoning or remembering them [5]. The standard practice of displaying document counts alongside facet labels furthers awareness of the breadth of the document corpus and informs refinement decisions. User search goals, whether navigational (seeking specific documents) or Informational (searching for more generalized information) tasks [4] must both be accommodated by a competent interface.

Faceted search strikes a fine balance between traditional keyword search and browse/directory patterns commonly used in information retrieval. Clicking on facets is an easier exploration method for searchers than formulating or reformulating search criteria [8]. Broad searches can be issued with low effort and then pared down interactively.

Gaze tracking research has shown that users interacting with a faceted search interface spend roughly one half of their time gazing at the facet area of the UI [9]. This data suggests that care in facet implementation may be as important as search results design or execution of ranking algorithms in those results. In a usability study against architectural data, users were twice as likely to begin by browsing facets than keyword search and 3 times more likely to refine their search with drill-in via facets than search-within the result set using keywords. Furthermore, 85% of those users self-reported a preference for the faceted version of the interface over a simpler keyword based version searching the same data [5].

4a. Searching within facets result sets

Some exploration has been done regarding the integration of keyword search within facet-refined result sets. In this scenario the user is able to type query strings into a text box and have that constraint added to existing query refinements. This approach has been shown to confuse users in some situations. If 'search-within" functionality is deemed necessary, it should be implemented so that the keyword search follows the conventions of any other facet refinements, is made available for easy removal to expand the search [6] and the behavior of this input is explained clearly and early to the user avoid misconceptions [11].

The downside of not offering a search-within option is that facilitating any additional free text searching beyond the initial query would result in additional work from the searcher as they must reformulate the search to reproduce earlier refinements. These issues suggest that the trade-off between interface complexity and reduced functionality should likely pivot on the size and nature of the document set, with larger, more textual document sets warranting keyword search within facet-refined datasets.

4b. Exposing facets through autocomplete

While most faceted interface designs confine facets to a prescribed area alongside search results, some successful designs have integrated the facet hierarchy into an autocomplete pattern. A user may type keywords in a textbox and relevant facets will be displayed in a nearby UI panel, updated with each keystroke [1]. This arrangement allows users to begin interacting with a faceted browse version of the category facets even before being exposed to actual document results. Rather than refreshing the entire result set to modify the facet values, they are updated instantly as the user enters terms. In some cases autocomplete may be used as secondary controls at the individual facet level [7], allowing users to interactively narrow down a large amount of possible values within a facet.

These designs potentially save time in the retrieval process for navigation-driven users. For information-driven users it front-loads the information-scent advantage of the facet ontology, adding additional opportunities for education about the structure of the collection and presenting additional opportunities for serendipitous discovery.

4c. Display order

Typical datasets present the searcher with multiple possible facets and values. Some consideration must be given to the most effective order and presentation of those options. Ideal ordering of facets and their values is, once again, highly dependent on the dataset in question and the information needs of the user searching it. For simple facet display and ordering, probable commonly accessed facets should be made evident through user studies and can be evaluated effectively and later adjusted through log analysis. Deciding what the most effective display order is for values within facets may be determined through a fixed set of rules for display or by dynamically inferring facet importance through ranking functions at search time [2]. In most cases people prefer known, easily understood ordering schemes like alphabetical, ordinal or popularity sorting. In cases where there are a large number of facets that may not all be shown, users prefer to see the most salient/frequent options be displayed with the ability to expand the selection using a 'more' action [5].

Recommender approaches may also be used to promote facet and value suggestions. A content-based recommender system may be implemented in which selection of a given facet may be used as the basis for promoting that facet and related facets' importance in the UI. Conversely, less-used facets may be demoted through lack of use or lack of relevance to previously selected facets. In the case of a new user with no sizable previous selection history we can turn to a collaborative model to promote facets based on the actions of other users [8]. A hybrid method applying subtle boosting from a combination of both content and collaborative recommender algorithms could take the whole notion a step further, mitigating the inherent deficiencies of both approaches [10].

5. DESIGN PRINCIPLES

The design of faceted search interfaces should assess the needs and capabilities of users and modify presentation in ways that support those findings. In addition to application-specific decisions, there are some rules of thumb that can be used to guide designers as they create new interfaces.

While some facet menus may be displayed above the search results, designers must be careful that users are able to still see those results without scrolling down the page to make sure that the major portions of the interface are all available 'above the fold' during initial interactions [6].

Another important consideration is how users control query expansion after selecting refinements. Allowing for easy removal of refinements permits users to make low-cost exploratory selections and easily back away from them, aiding in confident wayfinding within a complex query [5]. These controls should allow users to remove individual refinements without the need to retrace a linear history using the 'back' function of their browser. Such controls are most often implemented successfully through a breadcrumb display of past refinements that may be deleted individually, regardless of the order in which those refinements were accumulated [6].

Ideal strategies for displaying facets vary depending on the number of available facets and the volume of refinement values within them. Some or all facets may initially be displayed in a 'collapsed' state to conceal values unless the user clicks to reveal them. A good rule of thumb for the appropriate number of 'expanded' facets presented for any given search or level of drill-down is four [6], with additional 'collapsed' facets appearing beneath those. Facets which are no longer relevant should remain present for the sake of interface consistency, but appear grayed out much like disabled controls users might encounter in other familiar GUIs [6].

The amount of necessary visible textual information and highly interactive UI elements present in a faceted search interface increases the need for clean, uncluttered design. Clear visual separation of facet menus from search results and refinements is advised [6].

6. CASE STUDY: HEMMINGS CLASSIC CAR SEARCH

Hemmings.com is an automotive website that allows users to search for classified ads for classic parts and vehicles. The core application interface allows users to search and browse through these approximately 30,000 documents and contact sellers if they find an item they wish to purchase. We set out to apply the research in our literature review to the construction of an effective faceted search interface for this dataset that takes into account the needs of users and makes best use of the documents available. This analysis of this design is patterned after the findings of our literature review.

Our initial user research consisted of a 500 person survey of current site users which included request-for-comment questions that were coded and translated into problems and feature requests.

We then conducted a small usability test that asked users to perform specific search tasks on Hemmings and three similar search sites, recording transcripts and generating insights from comments and frustrations.

6a. Ontology creation strategy

The operational database providing the source documents for Hemmings' search has been the basis for the site's existing car search for nearly a decade. Consequently, the document structure is well-developed in terms of metadata and category assignment in order to facilitate the existing traditional parametric search interface.

Each document contains some free text data, usually having between 1 and 7 paragraphs of item description in each document. The remainder of the document fields are normalized descriptive data that can serve as the foundation for our facet construction.

Our usability tests were combined with log analysis and some common-sense domain knowledge to establish the basis for our static facet ontology. Users most often search by make, followed by model of vehicle. Popular refinements include price range, year range, geo distance from searcher and type of ad (vehicles for sale, parts, services, etc.). Usability tests and survey finding showed that users are also interested in filtering by type of seller (private or dealer) as well as the ability to exclude ads which have no photos or prices associated with them.

6b. Augmenting data for facet presentation

Vehicle condition, paint color and transmission type were search criteria mentioned repeatedly in user studies but not present in the dataset. Before construction of the new search interface we added these fields to the classified ad creation process in order to augment the dataset to support these as facets. Unfortunately, not all ad creation processes, such as feed imports from other providers, contained this information in normalized forms.

One frustration expressed by users in our research was the over-granularity of Hemmings' model assignments for vehicles. Some model designations are extremely specific within a related group of models and past interfaces forced users to refine their search to a fine level of specificity when they were interested in a broader view of a model 'grouping'. A relevant example is that of Porsche 911 series of cars. The original taxonomy was designed by automotive editors to precisely reflect variations in the Porsche 911 and contains at least 19 different submodels (911, 911C4, 911SC, 911GT2, etc.) A user wishing to browse all 911 variants would be forced to use a less specific browse refinement such as "All Porsche cars" or else be were forced to navigate to each model, possibly adding 19 additional navigation actions to their search. In this case the goals of faceted search conflicted with the intended accuracy goals of Hemmings' editors.

6c. Intermediary ontology for model designations

In order to bridge the gap between the users' interface needs and the underlying dataset we established an intermediary ontology category called "Model Grouping" that forms a hierarchy between make and model relationships. In the case of the Porsche example we created a "911 series" parent group to use in facet menus and parametric search. This process involved a form of card sorting by the editors within spreadsheets. Editors analyzed each make for emergent model groups and assigned parent values to granular model designations [Figure 1].

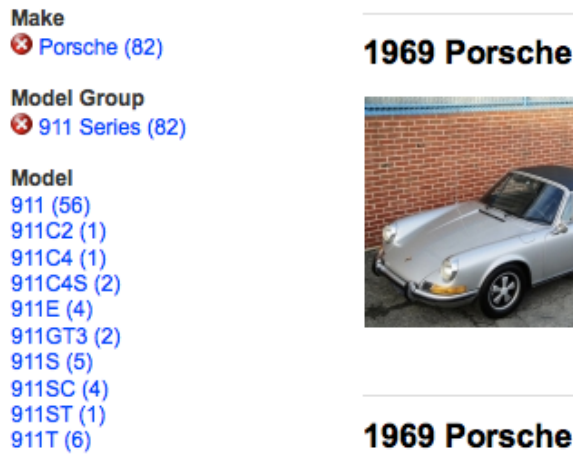

Figure 1. Intermediate ontology for over-granular model designations in source data

This step proved to be extremely difficult due to the nature and variety of car model designations used in the last 120 years by automakers. Simple model grouping categorization proved elusive and in some cases inappropriate or arbitrary. This ontology is still under refinement but appears to be an effective solution to the model-granularity problem.

Other data values that make sense for the operational needs of the database were found to be confusing or redundant. Values for the 'transmission' attribute were somewhat normalized but consisted of values such as 'five speed manual' or '4 speed automatic'. We solved these issues with additional intermediary ontologies, either explicit or dynamic, that mapped these into values more appropriate for the querying needs of users. In the case of 'transmission-type' a text filter maps these into simple "manual" or "automatic" values at index time for use as a facet. The additional detail of the original value is preserved and still made available in the documents so they can still be found through free text queries.

6d. Use of dynamic category sets

Hemmings users may be unfamiliar with the labels and category names used in the dataset, possibly resulting in a vocabulary mismatch problem that may exclude valid results. Following Daniel Turkelang's work on Dynamic Category Sets we created a synonyms file for thesaurus expansion on certain terms to dynamically map search keywords terms to our standardized facet values. Parsing web search logs we found common misspellings and shorthand labels that would likely result in confusing unsuccessful searches without some level of intervention.

Implementing this secondary level of intermediary ontology ensures that a user who types "Chevy" into a search box is shown facets and results mapped to "Chevrolet" at search time. Other entries in the thesaurus map "Automatic" to auto and "Stick" to "Manual". It's unclear to what extent this automatic term expansion may confuse users so entries are limited to only the most common and obvious cases.

6e. Selection of data types for facets

The most strongly enforced hierarchical relationship in the Hemmings ad data pivots on the make, model-group, and model values. While most all other data has no enforced hierarchical relationships there are certainly instances where attributes are only appropriate for display in the context of the state of 'drilldown' in the interface, due either to lack of normality in the data or the relevance of broadly refined phases of the search. The contextual relevance of a given facet is addressed later in this paper.

Applying Koren and Zhang's basic facet types [8] as a guide for facet construction we began designing the facet values themselves. The primary facets like make, model-grouping, model, seller type and ad type were implemented as nominal types, concise and consistent labels borrowed directly from the schema and reflecting the underlying data values. Nominal types were also used for boolean selections like "has images" and "has price". In some cases we decided the boolean facets would be best represented by persistent checkboxes in the facet menu which later posed implementation issues for our query logic.

Ordinal and ratio facet data types were not used for any user-facing facing facets. While data values for 'popularity' and 'price / average price for similar item' are available in the index, there were no appropriate ways to display this information in the facet menu. These values are instead reserved for result sorting and possibly for ordering facet values when a logical ordering system is not obvious.

We chose to present pre-computed interval facets such as 'decade of model year' or 'price range' as convenient shortcuts for range refinements that might save users time in forming explicit range queries.

The nature and amount of available textual data in the documents does not lend itself to the use of free-text fields that may expose the occurrence of common phrases within ad descriptions. "Body style", "interior color" and "exterior color", however, are not normalized values in the dataset. Users may enter free text descriptions of these attributes and the values take the form of free text not mapped to a pre-established taxonomy as in the case of normalized make and model information. The result is a list of values that may be semantically similar for searchers but not grouped as such by the index. "Black Sapphire metallic" and "Anthracite" exterior color values may be equivalent for a user searching for "black" cars but the underlying data does not support nominal faceting on these values without augmenting the data or modifying the interfaces that supply these values. In a later section we will discuss the implications and use of the free text facets.

We analyzed the applicability of correlated facets in the interface and found that the lack of multi-valued fields did not present many opportunities in our design. We did find use for dynamic category sets to combine makes and models for facets used by autocomplete search discussed in a later section.

The use of pre-computed values to increase information density of facets proposed by Ben-Yitzhak [8] was also explored. The best candidate for this treatment was the model facet which might usefully be combined with an "average price" value to give additional information scent to users browsing a model list. After some experimentation it was decided that simply displaying the mean price alone could be misleading. Sparkline histograms were discussed as an alternative but the end analysis was that the cost of complicating the model facet labels outweighed the value of the possible added information density of graphical facet values.

6f. Removing facet refinements

The research we reviewed was consistently emphatic about allowing easy user interface controls not only to add refinements, but also to remove them.

This was implemented in two ways to make query expansion controls as clear as possible. We created a breadcrumb dialog at the top of the search results that displays all added refinements with obvious 'x' links to permit their removal [Figure 2]. We repeated this pattern in the facet menu, leaving the facet heading visible with the lone available option featuring the same 'x' link [Figure 3].

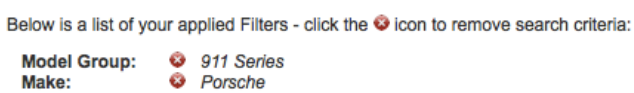

Figure 2. Refinement controls in results

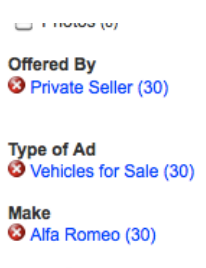

Figure 3. Refinement controls repeated in facets

6g. Facet display logic

The average Hemmings user is older and self-reports average or below-average technical proficiency. For this reason we sought to enable simple, consistent interactions in our facet menu patterns.

Decisions about initial presentation of facets focused on a balance of most-used refinements mined from logs of the existing interface and the predicted utility of new options.

Several facets were tagged as 'context modifiers' because of their hierarchical relationships within the data. Selection of these 'context-modifier' facets would drive small changes in the available facets and URL structure and may modify other items on the page like related content results and page title/header text important to search engine optimization. Selection of a specific 'Make' facet, for instance, would add a new 'Models' category to the facet menu. If a user then selected a certain model then the facets would update again to display relevant model-oriented information details. Information like exterior color and body style would otherwise be an overwhelming series of free-text values at more general levels of drill-down. Some of these same model-specific facets would be useless and inappropriate if the user had selected an 'ad type' of 'Services' where exterior color or transmission-type were simply not applicable. Representing these "contexts" in relation to refinement selection is managed internally by a static matrix representing all possible decision trees of context facet refinement with explicit instructions to generate the correct prescribed facet display list.

The mutable nature of the facet menu created some concern about visual consistency in the menu itself as users traverse available contexts and varying possible facet combinations. Although 'Make' is the most-used facet for users it does not persist across all search contexts and becomes less salient as users descend into further contexts.

The much-requested ability to hide ads containing no photos or pricing information were the only facets we determined should follow what we termed a "persistent checkbox" pattern. These facets differ from the typical "`Label (Document Count)`" facet design by offering the user a textbox to toggle the inclusion of documents without images or prices [Figure 1]. If the user chooses to refine the query to omit documents by selecting the checkbox, the facet persists as a control to easily remove this refinement along with an updated document count.

Figure X. Persistent checkbox facet

The visual consistency of this facet and its relevance across all contexts led us to place it as an 'anchor' for the top of the facet menu.

6h. Facet value ordering

Each facet was analyzed to determine the most natural order for possible values. Alphabetical sorting was used most frequently, particularly on the free-text value types. Where range facets are presented we force the range values into a meaningful descending order. Most facets were very easy to assign sorting strategies to.

Several facets, like “Make” may contain more than 1,000 thousand possible values. Following best-practices prescribed by Marti Hearst [6], we conceal all but the top 30 values for Make ranked by document count and then re-sorted those in alphabetical order. A simple “More” link spawns a scrollable overlay dialogue exposing the full menu of facets [Figure 4].

Figure 4. “More” option allows for concealment of large numbers of facet values

6i. Evaluating “Search Within” and autocomplete interactions

Some constant UI elements in the design are free text search boxes available at the top of the facet menu and beneath the results pagination. Heeding both Hearst [5] and Sacco’s [11] findings, we made the default behavior of these controls to completely re-issue the search based on submitted keywords. We amended this by adding a checkbox that allows the user to explicitly decide they wish to search within the existing result set, hoping that this UI addition clearly describes the difference between the possible actions of the keyword search [Figure 5].

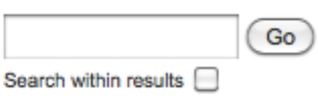

Figure 5. Disambiguating search box behavior in UI

Autocomplete behavior was combined with dynamic category sets to produce a dynamic list of make and make/model combinations to be displayed as the user types in the search box. A facet query is issued to the search index on each key-press event to present the user with probable generated values for their query [Figure 6]. Currently we have this configured to show only the labels for those facets but have built in the ability to also reveal available document counts in order to do A/B comparisons in usability tests.

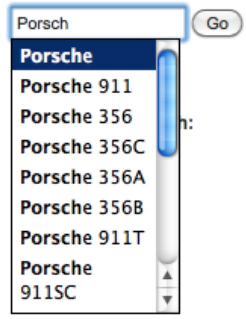

Figure 6. Autocomplete pattern paired with Dynamic Category Sets

Brief experimentation with extending autocomplete to individual facets with large numbers of possible values was abandoned due to resulting interface complexity weighed against a perceived lack of utility within this particular dataset.

6j. Visual design decisions

As with every other consideration, simplicity over complexity became our stated visual design strategy. Our final design follows the common “Facets on left, refinements on top” model our users have likely encountered on the web and competitor sites like ebay, hopefully presenting them with the shortest learning curve for navigating our facets [Figure 7].

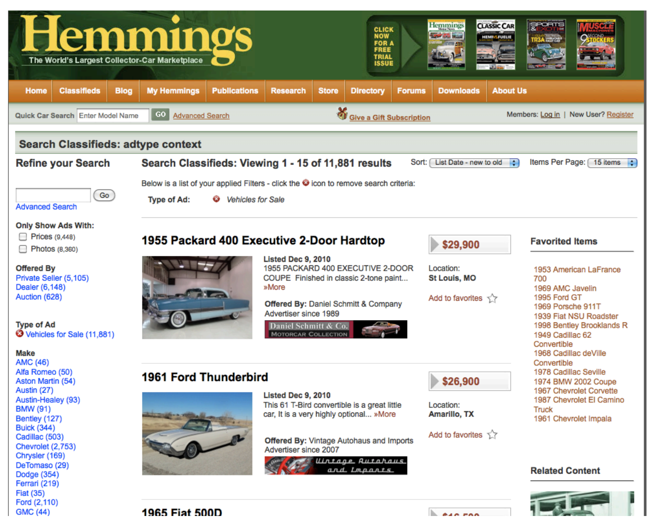

Figure 7. Overall visual design. Simple and uncluttered.

Wherever possible, we tried to avoid borders and horizontal or vertical rules in favor of clean whitespace in an attempt to remove anything that may clutter the already-dense user interface.

As explained in the section about refinements, we decided to persist facet labels where possible, even if they contain only one option. This allowed us to avoid the possibly confusing ‘disabled’ facets proposed by Hearst. We show no more than 7 facets at any given contextual state but eschew using 'collapsed' facets in favor of 'more' controls for hiding large value sets.

CONCLUSIONS - RESEARCH QUESTIONS

Faceted search constitutes an additive level of complexity built on the success of previous search interface designs. This complexity presents many opportunities for improvement of search experience and effectiveness but also may generate interfaces that overload users' comprehension of the data presented. Cautious implementation of concepts found in the literature is advised and adherence to simple best-practices is the best path forward for interface designers.

Our case study demonstrates an applied example of the implementation of a faceted search interface based on theory and concepts in the relevant literature.

Next Steps - Evaluation

The example provided in the case study is based on sound principles but will undoubtedly be subject to iteration based on evaluation in the form of more usability studies and analyzing user behavior through web logs.

Further Research Questions generated from the case study

1. What are the best techniques for combining initial parametric search with a faceted result set? Do users expect the parametric selections for the initial query to be later exposed as removable refinements or should initial search criteria held separate from facet refinements, mutable only by reinitiating the initial query?

2. Common faceted interfaces rely on simple textual links to refresh the result set and facet view for each selection. What are the best practices for allowing more complex facet options such as multiple selections enabled by checkbox controls that can be checked or unchecked and persist in the facet menu?

3. Does enabling meaningful URL structures that correspond with important facet selections aid at all in understanding and wayfinding for users?

4. Is there a usability risk or advantage associated with using rich UI elements like slider controls to refine continuous numeric range refinements in place of textboxes or pre-computed range values?

REFERENCES

1. Bast, H. and Weber, I. 2006. When You're Lost for Words: Faceted Search with Autocompletion. ACM SIGIR Workshop on Faceted Search, 2006 (Seattle WA, August 2006), p. 31--35.
2. Basu, R.S. and Wang, H. and Das, G. and Nambiar, U. and Mohania, M. 2008, Minimum-Effort Driven Dynamic Faceted Search in Structured Databases, in Proceeding of the 17th ACM conference on Information and knowledge management (2008) p. 13—22.
3. Ben-Yitzhak O., Golbandi, N. et al. 2008. Beyond basic faceted search. In Proc. of 1st Int'l Conf. on Web Search and Data Mining (WSDM) (2008), p. 33--44.
4. Cutrell E. and Guan Z. 2007. What are you looking for?: an eye-tracking study of information usage in web search. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, p. 407--416.
5. Hearst, M. and Elliott, A. and English, J. and Sinha, R. and Swearingen, K. and Yee, K.P. 2002, Finding the flow in website search. Communications of the ACM (2002), 45(9): p. 42-49.
6. Hearst M. 2006. Design recommendations for hierarchical faceted search interfaces. Paper presented at the ACM SIGIR 2006 Workshop on Faceted Search (August 2006), p. 26--30
7. Hearst M. 2008. UI's for Faceted Navigation: Recent Advances and Remaining Open Problems. Proc. 2008 Workshop on Human-Computer Interaction and Information Retrieval (2008).
8. Koren J., Zhang Y. and Liu X. 2008. Personalized interactive faceted search. In Proceeding of the 17th international conference on World Wide Web (ACM, New York, NY, USA 2008), p. 477--486
9. Kules B., R. Capra, M. Banta, and T. Sierra, What do exploratory searchers look at in a faceted search interface?, in Proceedings of the 9th ACM/IEEE- CS joint conference on Digital libraries, p. 313--322, Austin, TX, USA: ACM Press, 2009.
10. Martínez, L., Pérez, L. G., Barranco, M., & Espinilla, M. 2008. Improving the Effectiveness of Knowledge Based Recommender Systems Using Incomplete Linguistic Preference Relations. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 16(supp), 33.
11. Sacco, G., & Tzitzikas, Y. 2009. Dynamic taxonomies and faceted search: Theory, practice, and experience. Springer Publishing Company, Inc., Berlin DE, 2009.
12. Souminen, O. and and Viljanen, K. and Hyvanen, E. 2007. User-centric faceted search for semantic portals. The Semantic Web: Research and Applications. Springer (2007). p. 256--370.
13. Tunkelang, D. 2006. Dynamic Category Sets: An Approach for Faceted Search. In SIGIR Faceted Search Workshop (2006).