When combined with keyword search, this approach becomes incredibly powerful: so much so that faceted search is now the dominant interaction paradigm for most ecommerce sites and is being applied to an increasingly diverse range of search and discovery applications. However, with this power comes a challenge: given the ease with which information spaces can be explored, what techniques should be employed to communicate the user’s current location and navigation options within that space? And how should these mechanisms be extended to facilitate further exploration of that space?

**Breadcrumbs**

One of the simplest techniques by which navigational state is communicated is through the use of breadcrumbs. Breadcrumbs have a long history in the development of the web (and hypertext systems in general), and have been the subject of many previous research studies.

In their simplest form they are rendered as a trail which indicates the user’s current position in an information hierarchy (or taxonomy), e.g. Home page > Section page > Subsection page.

But the whole point of faceted search is to allow navigation along several independent dimensions, rather than just one. So in this case, we need a mechanism that will scale to accommodate multiple dimensions.

**Inline breadcrumbs**

Perhaps the simplest example of a technique to accommodate multiple simultaneous dimensions is the use of multiple ‘inline breadcrumbs’, as shown on Amazon (1). Looking closely at the faceted navigation menus, we can see that each one contains the current selection (e.g. Format=paperback, language=English etc.) and the means by which the user may remove the current selection for any given facet (e.g. Any Format, Any Language, etc.). Note the subtle use of bolding to indicate the currently selected facet value, the use of indentation to indicate hierarchy, and the use of chevrons to imply going ‘back’ to a previous state.

However, there are two shortcomings...
with this design. First, it doesn’t scale well to results that possess a large number of facets whose combined vertical height extends beyond the current screen height. In such cases, a proportion of the facet menus (and their currently selected values) will be hidden below the fold or otherwise out of view, and therefore not concurrently visible (without vertical scrolling). To a degree, this is mitigated by the presence of the conventional breadcrumb trail displayed immediately above the search results, but the visibility of this is questionable, and the lack of inherent structure makes its informational content relatively difficult to parse or comprehend.

Secondly, and more importantly, is how this approach would scale to accommodate multi-select facets. In other words, how would conjunctive (AND) or disjunctive (OR) facet values be rendered and selected? Despite the evident elegance of the current design, it is hard to see an immediate solution to this issue that would not involve some degree of deviation from the simple inline approach.

**Multi-selectable facets**

An extension to the above approach is to maintain the principle of displaying current facet selections inline (i.e. within the facet menu itself), but allow the facet values to be multi-selectable. An example of this can be seen at eBay (2).

In this example, we can select multiple models (e.g. Golf OR Jetta) and multiple model years (2009 OR 2008 OR 2007) and these choices are displayed inline as selected checkboxes. Interestingly, the mechanism by which multiple values may be selected involves the use of a dialog overlay (3), which presents all the facet values as an extended list of checkboxes. Note how this overlay allows only one facet to be visible at any one time. Like Amazon, eBay also includes a conventional breadcrumb trail above the result, with what appears to be an even less structured treatment (bare facet values delimited by commas). Interestingly, eBay allows the user to close individual facets (presumably to optimise vertical screen space) but only if they have no current value selected. So in the above example, the user can only close the Fuel Type facet and the ones immediately below it. This principle, one suspects, may be specifically to minimise the likelihood of inadvertently hiding selected refinements from view.

Despite these useful modifications, the design still suffers from the same fundamental shortcoming as Amazon: in cases where a given result set possesses a large number of facets, their combined height may extend beyond the current screen height, making the overall navigation state difficult (if not impossible) to comprehend without repeated vertical scrolling.

For relatively simple B2C applications, such as the ecommerce examples above,
perhaps this is a suitable compromise to make. In such cases, the principles of elegance and simplicity are likely to outweigh any requirement for scalability and visibility of navigational state. However, as mentioned earlier, the applications of faceted search are being extended to a growing range of information discovery challenges, including more complex applications in business intelligence and analytics. In such cases, the need for a more scalable approach that more effectively communicates navigational state becomes more significant. In the next section, we’ll look at some of the alternative solutions for addressing this.

Breadboxes

The fundamental principle of the above designs is that they display the selected refinements in situ, i.e. inline within the facets themselves. And as we have seen, there are strengths and weaknesses with this approach. An alternative is to display all currently selected facet values in their own dedicated location or container. In the context of faceted search, this container is often referred to as a breadbox. An example of such a device can be seen at Food Network, in the container labelled ‘Your Selections’ (4).

The advantage with this approach is that all of the currently selected refinements are shown grouped together in a single place, which remains visible in the same location no matter how many facets are present. The overall navigational context is therefore immediately visible without vertical scrolling. However, one consequence of this approach is that a different shortcoming now becomes apparent: what facets do these particular values apply to? For example, what facet led to the selection ‘recipe’ or ‘main dish’? Since these values are no longer displayed in context, the user must either recall which facet was associated with each selected value or attempt to infer that from the labels themselves. Again, for relatively simple ecommerce applications, this may be a minor issue: after all, most users could reasonably interpret the values in the breadbox above, even if the precise facets that gave rise to them are now absent. There are, however, other ways to address this issue. The example below from NCSU Libraries illustrates one such approach (5).

In this case, we see that the breadbox contains not just the facet values, but the facet names too, i.e. Format=‘online’, Subject=‘natural language processing’ etc. The word wrapping in this particular example is somewhat unfortunate in that the controls to remove each refinement have wrapped as orphans in two cases, but the principle is clear: each facet value is unambiguously associated with its parent facet. Note also that this design scales easily to support multi-select facets (one of the key shortcomings of the inline breadcrumb design): the multiple selections for the ‘Subject’ facet are clearly visible and comprehensible in the above design.

In both the above examples, the breadbox is laid out vertically. But there are variations on this: in the example below, from University of Toronto Libraries, we see a similar navigational state laid out in a horizontal arrangement (6).

In this example, we see seven separate refinements laid out in sequence, complete with facet labels and removal controls. However, the comparison with the vertical breadbox makes one shortcoming of this approach quite evident: the content of the breadbox is much harder to comprehend, as the structural relationship between the facet/value pairs is now much less apparent.

Note, however, that in all the above examples we’ve assumed that the facets themselves are laid out using the vertical stack pattern. There are, of course, other configurations: notably open parametric, in which the facets are laid out in a consistent horizontal orientation, and displayed in their open state by default. In this configuration, the currently selected facet values are much more immediately visible, to the extent that a separate breadbox may
not always be required.

**Hybrid techniques**
In our discussion above, we have outlined two contrasting approaches to wayfinding in faceted search: breadcrumbs (in particular the inline version) and breadboxes. There are of course other variants, notably Greg Nudelman’s Integrated Faceted Breadcrumb. At first blush, Greg’s design seems quite radical, with the breadcrumb transformed into a flexible container that provides direct support for key navigational functions, such as updating facet values in situ; selecting a ‘See All’ option to remove a refinement; etc. Greg also uses the IFB to argue in support of some key navigational principles such as explicit labelling of the facets (as discussed above), retention of the navigational context wherever possible; and so on.

On reflection however, it could also be argued that many of these actions and principles are precisely those for which the inline breadcrumb approach has already been shown to work well, particularly the eBay incarnation with its support for more complex navigational options such as multi-select AND and OR. So it remains to be seen whether the integrated faceted breadcrumb is the first of a new set of wayfinding techniques in faceted search, or whether it essentially represents a re-allocation of functionality from one location (in this case, the facets) to another (the breadcrumb).

**Conclusion**
One of the great strengths of faceted search is the way it allows users to explore complex information spaces in a seamless and intuitive manner. But with that strength comes a challenge: what mechanisms should be employed to communicate the user’s current location and navigation options within that space? In this article we’ve examined some of the main techniques and reviewed their strengths and weaknesses.

For most simple ecommerce applications, the elegance and simplicity of inline breadcrumbs may be more than adequate.

But for more complex information discovery applications, a more scalable approach may be required which consistently displays the current navigational state in a dedicated container. In cases such as these, the breadbox may be the more scalable approach.

**The BCS Interaction Specialist Group**
website is at: [www.bcs.org/usability](http://www.bcs.org/usability)

---

**CASE STUDY 1**
**Validating faceted search with users**

Faceted search is just one of a smorgasbord of methods used by information architects to help people find content on websites. Another widely-used method is card sorting, where a researcher asks users to categorise the content of a website into groups that make sense to them. This is based on the theory that, if the website uses those same categories and terminology, users should find it easier to locate the content that’s of interest to them.

At least, that’s the theory. In practice, there are a few challenges. For example, I worked with an auction website a while back to help them revise their online help system. There was a large number of help pages (over 850) and these had grown in an ad hoc manner. To ensure the new help system achieved its expected business benefits, the client needed to structure and organise the content before it was integrated into the new interface. However, even the most dedicated user won’t be happy sorting 850 cards of content, so we first had to do something to make the task manageable. We began with a content inventory of the online help system. This was an important first step in describing the relationships between the different pages since it allowed us to answer questions like ‘Which help pages are most commonly accessed?’, ‘What search terms are most common?’ and ‘How many help pages does the typical user view in a session?’ Answers to these questions helped us classify the content into ‘critical’ and ‘secondary’ content. We also weeded out the ‘ROT’: content that was redundant, outdated or trivial. These steps helped us reduce the sheer amount of content to something that was a bit more manageable.

Our next step was to uncover how users thought about the structure of the help system, and it’s here that we used the card sorting technique. We asked 18 participants to sort a selection of help pages into groups that made sense to them. We made sure that our participants were regular users of the website so we could generalise the results to the general user population. Participants worked alone and each person took 60-90 minutes to sort the content. Afterwards, we discussed with our participants the sorting arrangements they had used. We asked them about the overall rationale for their grouping scheme, to help us understand whether their card sort was based on an underlying ‘mental model’ or whether they just made rough groups. We also asked for the best example from each of their groups, as there are often one or two cards that are most representative of the group and other cards that are less representative. And we checked that participants were happy with the overall outcome, to make sure that our results had some face validity.

Based on the research, we established the new navigation structure for the online help system. This was derived from a cluster analysis of the data that showed how often items were grouped together along with a qualitative analysis of the comments and annotations made by participants during sorting.

As a result of our work, the new information architecture reduced the number of support enquiries from users who were unable to find or understand content. Users were now able to solve issues themselves, which indirectly increased the number of listings, sales and registrations. The work also enhanced our client’s reputation as a site that people can use and trust.

Dr David Travis is a user experience consultant at Userfocus. He writes about user experience at [www.userfocus.co.uk/articles](http://www.userfocus.co.uk/articles). You can follow him on Twitter: @userfocus.
Applying card sorting techniques

In his article on improving the information architecture of a large auction site help system, David Travis touched on the application of card sorting to help provide a user-centred solution. Having now been working with and teaching courses on this topic for the past seven years, I am still surprised how little the techniques are known outside of user experience and user-centred design circles.

The problem stems from the temptation to believe that information architectures are meaningful in their own right – regardless of the intended audience. Unfortunately, nothing could be further from the truth. A quick visit to almost any website written by problem-domain experts for non-experts will confirm this. (I hate to name names, but almost any of the more traditional UK government websites still suffer from this problem. For example, search for ‘company car tax’ on the hmrc.gov.uk website and you will receive a motley collection of related articles but none that actually describes the basics. For that you need to know to look for ‘class 1a national insurance.’)

In this kind of setting, card sorting has a wide range of applications: from small-scale qualitative studies using paper cards to research how users think about a problem domain – where terms can be added, deleted or amended; to large-scale quantitative studies using online tools with potentially hundreds of participants and items. (Although, as David mentions, it would be considered inhumane to ask a single participant to sort 850 items – partitioning the domain into smaller projects is the most common solution in this instance.)

Paper and online approaches are the two most common, usually involving sorting ‘items’ representing the leaves of hierarchical trees into ‘groups’ that represent the nodes. However, sorting is not limited to lexical presentations – objects or images can be sorted as well. This can be particularly useful in novel problem domains where no user vocabulary has been established. Naturally, objects need to be sorted in face-to-face sessions (since there are no 3D packages for sorting virtual objects) but images can be sorted in either face-to-face sessions or online.

And while it may sound tedious, face-to-face applications can be made much easier to prepare and analyse using barcodes – an approach that I developed in frustration many years ago and now used in many fields, ranging from social research to information architecture (see the links on page 12 for free tools and instructions).

Other variables that need to be considered when designing card sorting activities is whether users are allowed to make up their own groups or if they must use only groups you’ve provided.

The former is referred to as an ‘open’ sort and can be particularly useful when trying to better understand a problem domain.

However, it is probably worth giving participants guidance on the number or size of groups you are looking for, particularly if large amorphous collections at one extreme and single-item groups at the other are to be avoided (neither case is particularly helpful, almost entirely independent of the problem domain).

‘Closed’ sorts, by comparison provide relatively little new information but can be used to assess whether your items and groups are well-understood by participants.

In between the closed and open sorts is an approach that I refer to as ‘hybrid’ – participants are given a number of pre-defined groups but are allowed to change these or create their own.

Unfortunately the hybrid approach is not supported by some online tools, although evidence of demand might change this (if an online sorting site does not offer this flexibility send them some feedback to let them know you would find it useful).

William Hudson is a user experience strategist at Syntagm. He teaches courses in user-centred design on a wide variety of topics, ranging from card sorting to Agile User Experience and UCD.

CASE STUDY 2

Applying card sorting techniques

In his article on improving the information architecture of a large auction site help system, David Travis touched on the application of card sorting to help provide a user-centred solution. Having now been working with and teaching courses on this topic for the past seven years, I am still surprised how little the techniques are known outside of user experience and user-centred design circles.

The problem stems from the temptation to believe that information architectures are meaningful in their own right – regardless of the intended audience. Unfortunately, nothing could be further from the truth. A quick visit to almost any website written by problem-domain experts for non-experts will confirm this. (I hate to name names, but almost any of the more traditional UK government websites still suffer from this problem. For example, search for ‘company car tax’ on the hmrc.gov.uk website and you will receive a motley collection of related articles but none that actually describes the basics. For that you need to know to look for ‘class 1a national insurance.’)

In this kind of setting, card sorting has a wide range of applications: from small-scale qualitative studies using paper cards to research how users think about a problem domain – where terms can be added, deleted or amended; to large-scale quantitative studies using online tools with potentially hundreds of participants and items. (Although, as David mentions, it would be considered inhumane to ask a single participant to sort 850 items – partitioning the domain into smaller projects is the most common solution in this instance.)

Paper and online approaches are the two most common, usually involving sorting ‘items’ representing the leaves of hierarchical trees into ‘groups’ that represent the nodes. However, sorting is not limited to lexical presentations – objects or images can be sorted as well. This can be particularly useful in novel problem domains where no user vocabulary has been established. Naturally, objects need to be sorted in face-to-face sessions (since there are no 3D packages for sorting virtual objects) but images can be sorted in either face-to-face sessions or online.

And while it may sound tedious, face-to-face applications can be made much easier to prepare and analyse using barcodes – an approach that I developed in frustration many years ago and now used in many fields, ranging from social research to information architecture (see the links on page 12 for free tools and instructions).

Other variables that need to be considered when designing card sorting activities is whether users are allowed to make up their own groups or if they must use only groups you’ve provided.

The former is referred to as an ‘open’ sort and can be particularly useful when trying to better understand a problem domain.

However, it is probably worth giving participants guidance on the number or size of groups you are looking for, particularly if large amorphous collections at one extreme and single-item groups at the other are to be avoided (neither case is particularly helpful, almost entirely independent of the problem domain).

‘Closed’ sorts, by comparison provide relatively little new information but can be used to assess whether your items and groups are well-understood by participants.

In between the closed and open sorts is an approach that I refer to as ‘hybrid’ – participants are given a number of pre-defined groups but are allowed to change these or create their own.

Unfortunately the hybrid approach is not supported by some online tools, although evidence of demand might change this (if an online sorting site does not offer this flexibility send them some feedback to let them know you would find it useful).

William Hudson is a user experience strategist at Syntagm. He teaches courses in user-centred design on a wide variety of topics, ranging from card sorting to Agile User Experience and UCD.