

Chapter 4

User Interface Design

Moritz Stefaner, Sébastien Ferré,
Saverio Perugini, Jonathan Koren, and Yi Zhang

*“Design is not just what it looks like and feels like.
Design is how it works.”*

Steve Jobs, 1955–

As detailed in Chap. 1, system implementations for dynamic taxonomies and faceted search allow a wide range of query possibilities on the data. Only when these are made accessible by appropriate user interfaces, the resulting applications can support a variety of search, browsing and analysis tasks.

User interface design in this area is confronted with specific challenges. This chapter presents an overview of both established and novel principles and solutions. Based on a definition of core principles (see Sect. 4.1) and challenges (see Sect. 4.2), we define a taxonomy of navigation modes observed in existing applications (see Sect. 4.3). On that basis, design patterns for enabling these navigation modes in user interfaces (see Sect. 4.4) as well as extensions and related approaches (see Sect. 4.5) are discussed. The chapter closes with an approach to personalizing faceted search (see Sect. 4.6).

4.1 Principles

Extending traditional models of Information Retrieval, *search for digital resources* has lately been widely recognized as multi-step processes [32, 130, 179, 228]. To follow the terminology introduced in [130], a search usually involves an *initial constraint definition*, followed by an *orientteering and refinement* phase based on first inspections of the result, and finished with a closer examination of individual results in the so-called *endgame*.

In this context, the exploration of dynamic taxonomies [236] with *facet browsers* is often seen as a most promising candidates for “rich exploration of a domain across a variety of sources from a user-determined perspective” [155]. These make different aspects of the underlying data accessible in parallel. Selecting one of the values, and thus filtering the result set, restricts the available metadata values only to those occurring in the results. Consequently, the user is visually guided through an iterative process of query refinement and expansion, never encountering situations with zero results.

Advanced Search

Type a query into the search box or use the fields below to construct your query

Include **ALL** of the following:

Keyword Anywhere

[Add Another](#)

But does not include:

Keyword Anywhere

[Add Another](#)

Limit my search results by:

Branch: Any

Available now: ☐

Language: Any Language

Format: Any Format

Audience: Any

Date Published: Between YYYY and YYYY

Search [Reset](#)

Fig. 4.1 The advanced search interface for the Oakville Public library at <http://opl.bibliocommons.com/search>

Applications for faceted search and dynamic taxonomies offer the following key features to support a wide range of search and browsing tasks:

- **Unrestricted query formulation over multi-dimensional classification**

Facet browsing applications impose no restrictions, in which order, or in which granularity filters are applied on a result set. Filters stem from various, orthogonal dimensions that can be combined by Boolean operators. This allows the formulation of complex queries, such as “All documents created before date A, related to topic B, and of file type C or D”. The equal treatment of multiple dimensions differs from, e.g. typical web site structures or file systems, where a single taxonomy is the pre-dominant organization principle, and other metadata are only supplements for sorting or filtering.

- **Poka yoke: no more empty result sets**

One of the core principles of dynamic taxonomies is to restrict the available filtering options in the given focus to only those, which will lead to a non-empty result set. Hence, the user can never run into a situation with zero results. This is opposed to the process in a typical *advanced search* situation, where first a complex boolean query is constructed, which is then evaluated on demand (see e.g. Fig. 4.1). That, however, can result in empty result sets, often without further indication on which part of the query could be relaxed in order to retrieve some results. The exclusion of potentially frustrating situations by design is often referred to as poka-yoke principle.¹

¹See e.g. <http://en.wikipedia.org/wiki/Poka-yoke>.