

Agile Browsing of a Document Collection with Dynamic Taxonomies

Sébastien Ferré

IRISA/Université de Rennes 1
Campus de Beaulieu
35042 Rennes cedex, France
ferre@irisa.fr

Abstract

Dynamic taxonomies and faceted search are increasingly used to organize and browse document collections. The main function of dynamic taxonomies is to start with the full collection, and zoom-in to a small enough subset of items for direct inspection. In this paper, we present other navigation modes than zoom-in for less directed and more exploratory browsing of a document collection. The presented navigation modes are zoom-out, shift, pivot, and querying by examples. These modes correspond to query transformations, and make use of boolean operators. Therefore, the current focus is always clearly specified by a query.

1. Introduction

Collections of numerical items are omnipresent: e.g., emails, photos, scientific publications, program source files, job announcements, e-commerce. The browsing of such collections is a key issue for users, and requires three main qualities: precision, ease of use, and confidence. *Precision* means that a subset of items can be retrieved under non-ambiguous criteria; *ease of use* means that users are guided in the expression of those criteria; and *confidence* means that enough feedback is returned to users so that they are confident about the results of their search.

The combination of those qualities requires the combination of querying and navigation [2, 4]. Dynamic Taxonomies (DT) [7] are a general browsing mechanism that fulfills this combination. At any time, the *focus* is set on a subset of items, which can be characterized by a boolean combination of concepts (precision). The focus can be moved to smaller subsets of items by selecting concepts in a *dynamic taxonomy*, which is a taxonomy relevant to the current focus. In this way, users are guided in the refining process that leads to the desired items (ease of use), and feedback is given at any stage about all alternative choices

(confidence). The same results are achieved by Logical Information Systems (LIS) [3], which use *logics* instead of taxonomies.

The goal is to show how dynamic taxonomies can be used to support more agile browsing of an item collection than the common zoom-in navigation mode. The motivation is that the user goal is not always to reach one item, but sometimes to reach and compare various sets of items, or simply to explore the item collection in order to discover interesting (ir)regularities [3]. We show that complex boolean queries can be composed without typing anything, but by simply selecting concepts in dynamic taxonomies. We also show that other navigation modes than zoom-in can be defined: zoom-out, shift, pivot, and querying by examples.

Section 2 recalls the basics of DTs, including how they provide local views on a collection. Section 3 details the different navigation modes, and illustrates them using a collection of photos. Section 4 discusses a few extensions and perspectives that are relevant to the agile browsing of a collection, and concludes this paper.

2. Dynamic Taxonomies as Local Views

A taxonomy is a concept hierarchy, where concepts are ordered from the more general topics to the more specific ones. Taxonomies can be designed for locations (from continents to towns), for dates (from years to days), or for groups of persons (from larger groups to individuals). Generally, the taxonomies used in DTs are multi-dimensional, and so include several such taxonomies. Items can then be classified under several concepts and at different levels of abstraction. When an item belongs to a concept, it also belongs to all ancestor concepts in the taxonomy. A key notion is the *extension* of a concept, which is defined as the set of all items that belong to it. A taxonomy extended with the extension of concepts is called an *infobase*.

The *navigation space* induced by an infobase is a very large graph, whose nodes are navigation places, and edges

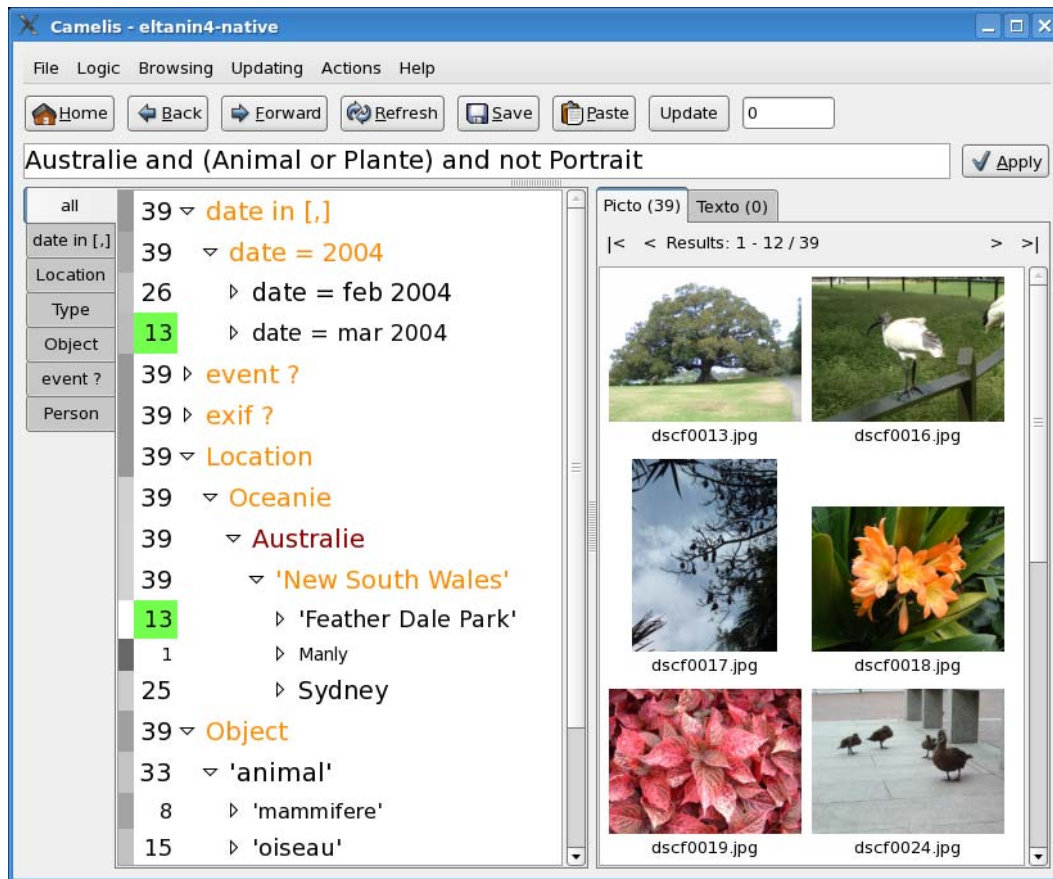


Figure 1. The graphical interface of CAMELIS.

are navigation links. A navigation place is any non-empty set of items that is the extension of some query. A *query* is a boolean combination of concepts, and the extension of a query is computed by set operations (union, intersection, complement), starting from concept extensions, according to boolean operations (resp., *or*, *and*, *not*). Hence, the richer the taxonomy is, the larger the navigation space is. Of course, the navigation space can have a size exponential in the number of items, but it is never computed entirely.

The principle of DTs is to show only a *local view*, focusing on one navigation place at a time. A local view is made of 3 parts: (1) the query, (2) the extension of this query, and (3) the *dynamic taxonomy* over this extension. The dynamic taxonomy plays the role of an index or summary for the current extension, and is defined as the pruning of the taxonomy to retain only concepts having common items with the current extension. Furthermore, the dynamic taxonomy is the support of the various navigation links that lead to related navigation places. The most common navigation link is the zoom-in that leads to smaller extensions. Other kinds of navigation links are introduced in Section 3, and consti-

tute the contribution of this paper. Every navigation link is defined by a query modification, and the current query therefore summarizes the navigation path since the beginning of the session. Even when the user does not need to edit the query, we think it is important to show it because it plays the role of a *breadcrumb* in the navigation process: after many navigation steps, the user may feel lost, but the query always tells her exactly where she is in the navigation space, and which choices she has already made.

Figure 1 is a screenshot of CAMELIS, our generic implementation of DTs, which we use to showcase the navigation modes in various applications. The structure of the local view is reflected by the graphical interface of CAMELIS. At any point in the interaction, the *current query* and its *extent* are displayed respectively as a text field at the top and as an item list at the right (here, photo thumbnails). The query is editable, and the extent can be scrolled page by page. At the left is displayed the dynamic taxonomy, where numbers, colors and scales are used to represent indicators about concepts, in the spirit of *elastic lists* [8]. The numbers are the (local) count of concepts in the current extension, and

the font scale represents the logarithmic ratio between this count and the size of the current extension. The grey-level at the left of each concept count represents the logarithmic ratio between the local count and the global count of the concept: light-grey means that most items of this concept are in the current extension, and dark-grey means that few of them are. The background color of some concept counts show when several concepts have the same items in the current extension. For instance, in Figure 1, we see that the 13 photos taken in March 2004 are the same as the 13 photos taken in “Feather Dale Park”, which means I visited this park in March 2004. Finally, if a concept is shared by all items in the current extent, it is red-colored if it occurs in the query, and orange-colored otherwise. This makes it possible to distinguish between logical and extensional consequences of the query. For instance in Figure 1, all selected photos are from Australia because the query says so (red), but all selected photos are from New South Wales because this is the only part of Australia I have visited (orange). Moreover, the distinction between red, orange, and black colors plays a crucial role in the navigation modes, namely between zoom-in and zoom-out.

3. Navigation Modes

We start with the common zoom-in mode, but extend it to handle disjunction and negation in queries. We then proceed with other modes: zoom-out, shift, pivot, and querying by examples. All modes are illustrated by examples on Lisa’s photo collection spanning several years. This collection contains 5,820 photos, which are described by date, location, event, type, visible persons and objects, and exif descriptors (e.g., time, flash, orientation). Exif descriptors are automatically extracted from photos, while other descriptors are given manually.

3.1 Zoom-in

Suppose Lisa wants to find some photos from a trip in Australia at the conference ICFCA’04. She first expands the concept `Location`, and finds she has photos from Europe (4859), Africa (162), and Australia (148). After selecting the concept `Australia`¹:

- the query becomes `Australia`,
- the concept `Australia` becomes red because it is now part of the query, and it is automatically expanded to show sub-locations of Australia,
- the concepts `Europe` and `Africa` are no longer visible, because they are no longer relevant, i.e., count = 0, and

¹There are French words in the screenshot as it is a personal photo collection, but english translations are used in the text for consistency.

- the extension displays 10 photos (out of 148).

The sub-location ‘New South Wales’ becomes orange, which means this is the only region of Australia where Lisa has been. More precisely, she finds that she has been mainly in Sydney (105), and in the Blue Mountains (18).

Now she expands the concept `Type` and sees there are different types of photos: buildings (29), animals (34), and plants (6). She becomes interested in Australian organisms, so she selects both `Animal` and `Plant`, which leads her to the refined query `Australia and (Animal or Plant)`, whose extension contains 40 photos. One of these photos is a portrait, which Lisa does not want, so she selects the negation of `Portrait` with the help of a contextual menu. This leads her to the new query `Australia and (Animal or Plant) and not Portrait` (39 photos). By expanding more concepts, she discovers that these photos were taken in February and March 2004 in New South Wales, and that 5 photos of 3 different species of marsupials are present: kangaroo, koala, and wallaby.

These 3 navigation steps lead to local views with increasingly more precise queries, and hence increasingly smaller extensions. This is called the *zoom-in* navigation mode, and it combines the current query and a refinement with a conjunction. A refinement can be a single concept, a disjunction of concepts, or the negation of such a refinement. When a refinement uses only concepts that are relevant and not colored, the new local view is always strictly smaller but not empty. This is an advantage compared to purely querying systems, where it is common to get empty results.

Figure 1 shows the interface obtained after the previous navigation operations. At this stage, Lisa can either browse the 39 photos in `CAMELIS`, or launch a diaporama in an external application.

3.2 Zoom-out

During navigation, the user may want to remove or generalize concepts in the query so as to reach larger extensions: this is the *zoom-out* navigation mode. For instance, Lisa realises she desires more photos of animals and plants. The back button can be used to retract the previous refinement. Hence if she wants to remove the first refinement `Australia`, she needs to move 3 steps backwards, and then re-select the last 2 refinements. She could also edit the query by hand, but users usually prefer to navigate rather than to edit queries [4].

Orange and red concepts are shared by all extension items and, therefore, cannot be used for zoom-in. This makes them available for zoom-out. When a red concept is selected, it is removed from the query. For instance, if Lisa selects `Australia`, the new query is `(Animal or Plant) and not Portrait` (282 photos from many

locations). When an orange concept is selected, it replaces more specific concepts in the query. For instance, if she selects `Pacific`, the new query is `(Animal or Plant)` and not `Portrait` and `Pacific`. In the latter case, the concept `Pacific` becomes red because it is in the new query, and the concept `Australia` becomes black, or orange in the case this generalization adds no additional photo.

Compared to the existing approach, i.e., a list of removable concepts, our approach has 3 advantages: (1) it is integrated into the dynamic taxonomy, (2) it allows the replacement of a concept by a more general one, and (3) it applies to refinements with disjunction and negation.

3.3 Shift

Zoom-in and zoom-out can be combined in 2 forms of *shift* navigation modes. Suppose the current query is now `Australia` and not `Portrait` and `Plant` (6 photos). At this point, Lisa sees that 1 photo has also the type `Landscape`, which interests her. She selects this concept (zoom-in) and, since the result has only 1 photo, she generalizes it by removing the concept `Plant` from the query (zoom-out). Therefore, she has executed a shift from Australian plants (6 photos) to Australian landscapes (80 photos), replacing in the query the concept `Plant` by the concept `Landscape`. From there, she performs a new shift from the concept `Landscape` to the concept `Building`, resulting in 28 photos of Australian buildings. These navigation steps are suggested and supported by photos sharing two concepts, i.e., by extensional relations [7]. This illustrates the relevance of assigning several types to photos, which is common in this photo infobase. The same would apply to persons visible on photos, as a photo can contain several people.

However, the same does not apply to locations, as a photo cannot be taken in 2 incomparable locations (e.g., in Australia and in Europe). Nonetheless, it is still possible to shift between locations, through the taxonomy of locations. Suppose Lisa wants to find building photos from other locations. She first generalizes `Australia` by `Location` in the query (zoom-out), and then browses suggested locations before selecting `Spain` (zoom-in). Thus, she has performed a shift from Australian buildings to Spanish buildings, and find 48 photos (which appear to be mainly churches taken in the north-west of Spain in 2003).

The former form of shift is a zoom-in/zoom-out combination, and can be qualified as *extensional* because it relies on extensional relations in the infobase. The latter form of shift is a zoom-out/zoom-in combination, and can be qualified as *conceptual* because it relies on conceptual relations in the taxonomy.

3.4 Pivot

The user may not remember a concept she wants to use to refine the query, but she can find it through another query. For instance, suppose Lisa wants to retrieve the photos of the building of some town. She does not remember which town it is, but she remembers that the ICFCA conference took place there in 2004. Therefore, she can first reach the query `event contains "ICFCA" and date = 2004` by zoom-in navigation. The resulting extension shows photos of ICFCA'04, and the dynamic taxonomy shows relevant information about these photos, such as precise dates, locations, and so on. By browsing the dynamic taxonomy, she discovers that Sydney, in Australia, is the location of ICFCA'04. Then, she can make the query become `Sydney`, and refine it to the desired query `Sydney and Building` by zoom-in. The concept `Sydney` plays the role of a *pivot* between the 2 queries.

Pivot navigation relies on the ability of DTs to answer queries not only by a set of items (the extension), but also by a set of concepts (the dynamic taxonomy). In previous navigation modes, these concepts were added or removed from the query, whereas here they are used as new queries. Therefore, pivot navigation is a way to restart a search from the results of a first search. This kind of navigation has already been applied in collaborative websites [6, 10].

There is an interesting analogy with natural language. Indeed, the query above can be rephrased as “photos of buildings in the town, *where* the ICFCA conference took place in 2004”. The idea of pivot is reflected by the fact that Sydney occurs in the main sentence as “town”, and in the relative sentence as the relative pronoun “where”. The relative pronoun indicates which facet to browse for a pivot: e.g., “where” indicates a location, “when” indicates a date, and “who” indicates a person. Iterated pivot navigation then corresponds to nested relative sentences, such as “photos of buildings in the town, *where* the ICFCA conference took place in the year, *when* I also visited Hinterzarten”. The first pivot is the year 2004, and the second pivot is the town Sydney.

3.5 Querying by Examples

A query can be determined by the selection of a subset of items, thus supporting querying by examples. The idea is to construct the query as the conjunction of all most specific concepts which are shared by the selected items. For instance, suppose Lisa starts with the query `Australia` and not `Portrait`. While browsing photos in the result, she sees interesting photos of buildings (e.g., 2 photos of the Opera, and 1 photo of the Harbour Bridge), and she would like to find more. By selecting them she moves to a new query that is the conjunction of the concepts shared

by those 3 photos. As usual with this form of navigation, the resulting query is very specific and she receives no additional photos. The concepts that make up the query are red-colored in the dynamic taxonomy, and she finds some of them very specific, e.g., a precise date (22 Feb 2004), a precise location (Sydney). At this step, she can use zoom-out to generalize the query. By removing in the query the concepts related to date and event, the query becomes *Sydney* and *Building*, and she finds 29 photos. From there, she can further zoom-out, zoom-in to find photos of modern buildings, or shift to find buildings in different countries. *Interactive query relaxation* [5] is similar to this process, except that only one facet is retained in the generalized query. For instance, starting with the same photos, Lisa could reach the query *Sydney* or *Building*, but not *Sydney* and *Building*.

A special case of querying by examples is when selecting only one photo. Since there is only one item in the extension, there is enough concepts to uniquely characterize each photo, and the query contains all the object descriptor concepts, which are more easily read as red concepts in the dynamic taxonomy. Therefore, this is an easy way to access the full description of an item.

4. Discussion and Conclusion

There are many ways to extend and improve the use of dynamic taxonomies for browsing and managing a collection of items. Some extensions are already implemented or being experimented in CAMELIS. First, the graphical interface of CAMELIS not only supports querying and navigation, but also the update of the infobase through the import of items (e.g., a pack of photos), or the addition/removal of items in concept extensions (manual annotations). Second, a *preselection* mechanism helps users to foresee the consequences of choosing some concepts to zoom-in, without changing the current local view, similarly to RB++ [9]. This is realized by modifying the appearance of the dynamic taxonomy and extension (e.g., background lightness), and enables a quick browsing and comparison of different selections. For instance, in the photo collection, Lisa can quickly see when she visited various locations by preselecting each location in turn, and looking at highlighted dates.

The intuitive display of statistical indicators [8, 9], and the definition and navigation of relations between items [1] may improve the agility of navigation. Statistical indicators extend DTs in the direction of data mining. Relations extend DTs towards structured data, and widens the application scope to, e.g., geographical information systems, genealogy, and software development.

In conclusion, dynamic taxonomies support the agile browsing of a collection of documents, given a multi-dimensional classification of them. Documents can be se-

lected according to complex boolean combination of concepts, i.e., queries. Those queries need not be edited by hand, but can be incrementally constructed by successive navigation steps, starting with the most general query. Only relevant navigation steps are suggested to users through dynamic taxonomies. Different navigation modes enable the user to change the query in various directions: zoom-in makes the query more specific, zoom-out makes it more general, shift replaces a query concept by a related concept, pivot replaces the whole query by a related concept. Finally, querying by example allows the definition of a query from a set of items.

References

- [1] O. Bedel, S. Ferré, and O. Ridoux. Handling spatial relations in logical concept analysis to explore geographical data. In R. Medina and S. Obiedkov, editors, *Int. Conf. Formal Concept Analysis*, LNAI 4933, pages 241–257. Springer, 2008.
- [2] Y. Chiaramella. Browsing and querying: two complementary approaches for multimedia information retrieval. In K. T. N. Fuhr, G. Dittrich, editor, *Hypermedia - Information Retrieval - Multimedia*, pages 9–26. Universit tsverlag Konstanz, 1997.
- [3] S. Ferré and O. Ridoux. An introduction to logical information systems. *Information Processing & Management*, 40(3):383–419, 2004.
- [4] R. Godin, R. Missaoui, and A. April. Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. *International Journal of Man-Machine Studies*, 38(5):747–767, 1993.
- [5] J. Heer, M. Agrawala, and W. Willett. Generalized selection via interactive query relaxation. In *Computer-Human Interaction (CHI)*, pages 959–968. ACM, 2008.
- [6] D. R. Millen, J. Feinberg, and B. Kerr. Dogear: Social bookmarking in the enterprise. In R. E. G. et al, editor, *Conf. Human Factors in Computing Systems (CHI)*, pages 111–120. ACM, 2006.
- [7] G. M. Sacco. Dynamic taxonomies: A model for large information bases. *IEEE Transactions Knowledge and Data Engineering*, 12(3):468–479, 2000.
- [8] M. Stefaner and B. Muller. Elastic lists for facet browsers. In *Dynamic Taxonomies and Faceted Search (FIND), DEXA Workshops*, pages 217–221. IEEE Computer Society, 2007.
- [9] J. Zhang and G. Marchionini. Coupling browse and search in highly interactive user interfaces: A study of the relation Browser++. In *Joint Conf. Digital Libraries*, page 384. ACM/IEEE, 2004.
- [10] D. X. Zhou, N. Oostendorp, M. Hess, and P. Resnick. Conversation pivots and double pivots. In *Computer-Human Interaction (CHI)*, pages 959–968. ACM, 2008.