Chapter 1
The Model

Giovanni Maria Sacco

Where is the wisdom we have lost in knowledge?
Where is the knowledge we have lost in information?

T.S. Eliot, Choruses from 'The Rock', 1934

We live in a world where the quantity of available information and its rate of growth are rapidly becoming limiting factors as important as the lack of information has been for thousands of years. The Internet and the World Wide Web are the main enabling technologies for this shift. In the past few years, the global distribution of information through Internet has made an enormous mass of information available to any web-connected location in the world. The physical location of information (large libraries, museums, etc.), one of the largest limiting factors in information availability, is now immaterial. With recent advances in wireless communications, all the information is also available on-the-move.

At the same time, the conversion of existing information (books, images, etc.) to digital format and the creation of new information in the appropriate format, has proved less overwhelming than it appeared in the early 1990's. Social networking and collaborative work and the distributed gathering/conversion of information has caused the quantity of electronically available information to grow at an extremely fast rate.

This situation has resulted in a dramatic information overload. After a decade of using traditional access paradigms, such as queries on structured database systems and information retrieval or search engines, the feeling that "search does not work" and "information is too hard to find" is now reaching a consensus level.

Two different information access modes can be identified: *focalized search* vs. *exploratory search*. In focalized search, the user attempts to quickly locate relevant information items on the basis of their contents. In exploratory search (also called browsing) the user explores relationships among items in a database. For example, consider a student using an electronic encyclopedia to produce a term paper on Michelangelo [236]. He quickly locates the section on Michelangelo (among several thousand other sections). This is focalized search. At that point, he explores relationships between Michelangelo and other painters, sculptors, architects, the Italian Renaissance at large, and the political situation in Italy during that period, etc. This is exploratory search.

Traditionally, research focused on focalized search. Examples include queries on structured databases and information retrieval (IR) techniques [311], recently

G.M. Sacco, Y. Tzitzikas (eds.), *Dynamic Taxonomies and Faceted Search,*
The Information Retrieval Series 25,
DOI 10.1007/978-3-642-02359-0_1,

dubbed search engines. Database queries require structured data and are not easily applicable to many practical situations, in which most information is unstructured or semi-structured. IR techniques try to implement querying for precise results on textual unstructured information bases.[1] Despite their wide usage, the limitations of IR techniques are well known: Blair and Maron [44] report that only 20% of relevant documents are actually retrieved. Such a significant loss of information is due to the extremely wide semantic gap between the user model (concepts) and the model used by commercial retrieval systems (words or strings of characters). Other problems include poor user interaction because the user has to formulate her query with no or very little assistance, and no exploration capabilities since results are presented as a flat list with no systematic organization.[2]

In order to overcome the semantic gap inherent in current IR technology, static taxonomies (such as *Yahoo!*'s) can be used. Such taxonomies are based on a hierarchy of concepts which can be used to select areas of interest and restrict the portion of the infobase to be retrieved. Taxonomies support abstraction and are easily understood by end users. Document classification according to taxonomy entries, usually a manual process although automatic and semiautomatic techniques have been proposed, eliminates the IR semantic gap because documents are now semantically organized. Taxonomies no longer deal with actual data, but rather with metadata, i.e., with a uniform, standardized and controlled description of content which is independent of format and, in the case of textual documents, of the actual terms found in the document.

However, static taxonomies are not scalable for large information bases [247], and the average number of documents retrieved becomes rapidly too large for manual inspection. The rapid decline of *Yahoo!*'s taxonomy as a primary tool to access information provides pragmatic evidence.

Solutions based on semantic networks, general ontologies, and the Semantic Web [40] are more powerful than traditional taxonomies. However, general semantic schemata are intended for programmatic access and are known to be difficult to understand and manipulate by the casual user. Therefore, user interaction must be mediated by specialized agents, and this increases costs, time to market, and decreases the transparency and flexibility of user access. In particular, agent-mediated search is based on the classic knowledge-based system paradigm, which does not take the user into account, but rather establishes a master-slave relationship between the system and the user. Brézillon [48] identifies this as the primary cause of failure for knowledge-based systems and, in particular, the fact that such systems behave as oracles, providing answers they are usually unable to explain to the user.

Regardless of approach, the underlying access paradigm is still focalized search: the user is assumed to know what he wants and the system tries to materialize the

[1] The term information base or infobase, instead of database, is used here to denote a collection of heterogeneous data objects of different format (text, images, video, etc.) which is not necessarily structured.

[2] Recent advances, including tag clouds, additional keyword suggestions, and result clustering address part of these problems and are reviewed in Chap. 3.

result. The techniques differ mainly in the amount of intelligence the system devotes to understanding what the user wants.

1.1 Exploratory Search

We contend that most “search” tasks are exploratory and imprecise in essence, and that using a focalized search paradigm in this context leads perforce to inadequate or frustrating user interactions. There are many different reasons why a user needs to explore an information base and consequently many exploratory patterns occur in practice and often depend on the application domain.

Perhaps the most common exploratory pattern is the pragmatic *find the right object* or *object-seeking* pattern which commonly occurs in e-commerce and in other object-selection tasks. We are given an information base consisting of objects characterized by features, and a user who wants to find the object which best suits his needs. For example, consider the purchase of a digital camera. Each camera is characterized by features such as price, weight, resolution, etc. If the user’s primary goal is to minimize cost, he will strive for the best inexpensive camera, with the definition of best inexpensive camera depending on his secondary goal (e.g., high resolution). Thus, for this exploratory approach to be effective, the user needs to:

- quickly find all possibly relevant features. The number of features might be overwhelming and they should be organized in a systematic way to allow user data access at different levels of abstraction: a taxonomic organization is a requirement in most cases;
- freely focus on the most relevant feature according to his individual requirements and discard objects without that feature. In the example above, the user might select a certain price range (e.g., cameras costing less than $200) as a starting point. Cameras outside that range must be discarded;
- explore all the features correlated with the selected one. What are the features (e.g., resolution, zoom, etc.) for cameras under $200? If the user is unable to determine them easily, the next focus cannot be set and the user has to inspect all the inexpensive cameras and find their features by enumeration. On the other hand, if related features are available, she can add to the current focus the next feature in the order of perceived importance and focus on it, thereby discarding other cameras which do not have that feature and consequently further thinning the number of candidate objects.

Advocates of the system-centric approach adopted by agent-mediated search might contend that these requirements are artificial, because the problem can be stated as a classical optimization problem: the “best” camera is the one which minimizes a weighted combination of features. Hence, a precise search rather than user exploration can be used.

From a user-centered perspective, weights cannot be obviously predefined, even for a specific user. Different users have different perceptions, but even the same user can be cost-conscious today, and less so tomorrow after a raise in salary. Thus, weights must be supplied each time by the user according to his perceptions. The user must supply a potentially large number of weights, which is cumbersome, and understand the effects of these weights on the selection mechanism of the agent, which is hopelessly difficult.

On the other hand, the mechanism sketched above only requires that the user identifies the most important feature: ranking instead of weighting, further simplified by the hierarchical abstraction mechanism of a taxonomy. Easy understanding of the consequences of focusing and real-time implementations allow the user to experiment with different strategies and gives the user the feeling to have considered all the possibilities.

The same pattern applies to all object-selection tasks in which different criteria must be reconciled to find a number of *good* candidates. E-commerce is an obvious application, but personnel selection, medical diagnosis, person identification, etc. fall in the same paradigm.

Not all the exploratory search patterns are so pragmatic and require the selection of objects. The exploratory search on Michelangelo discussed above, is a *knowledge-seeking* rather than an object-selection task. In other words, here we use Michelangelo as a focus to get additional relevant knowledge on a specific topic: the goal is to increase our knowledge, rather than pragmatically using this knowledge to select an object.

At the extreme end of this scale, there are *wisdom-seeking* exploratory patterns which are becoming increasingly important, in which the goal of exploration is to understand the inner laws of the information base, and gain insight into the working of the phenomena described by the information base. In these emerging applications, so to say, the journey is much more important than the destination. These exploratory patterns require data-mining capabilities.[3] Some applications are discussed in Sect. 8.5.5.

The best known technique for explorative search is currently hypertext/hypermedia [124]. Hypermedia connects items in the information base through explicit links which the user can traverse at runtime according to his needs. As the vast literature on this subject shows, hypermedia is able to accommodate a large number of user exploratory access patterns. Hypermedia is quite flexible, but it gives no systematic picture of the relationships among documents because there is no systematic organization of links by abstractions, and exploration must consequently be performed one-document-at-a-time, which is quite time consuming and ineffective.

1.2 Dynamic Taxonomies Defined

Traditional access paradigms are not suited to search tasks that are exploratory and imprecise: the user needs to explore the information base, find relationships among

[3] See Sect. 5.1.

concepts, and thin alternatives out in a guided way. New access paradigms supporting exploration are needed. Dynamic taxonomy and faceted search systems focus on user-centered interactive exploratory access, and propose a holistic approach in which modeling, interface, and interaction issues are considered together.

One of the key factors of this model is an explicit quest for simplicity and minimality, as opposed to current research trends which tend to high-complexity solutions. The effort in reducing the model to its minimal components makes it easily understandable and usable by end-users with no need for the mediation of any agent. As importantly, it makes the model easily understandable to researchers as well, and represents a solid foundation on which several extensions and solutions to real-world problems can be built.

Dynamic taxonomies [231, 232, 234, 236] (*DT*, also recently known as *faceted search systems*) are a general knowledge management model based on a multidimensional[4] classification of heterogeneous data *objects*[5] and are used to explore/browse complex information bases in a guided yet unconstrained way through a visual interface. The model is primarily concerned with user-centered access, and object classification is not addressed in the base model.

The conceptual schema or intension of a dynamic taxonomy is a plain taxonomy designed by an expert of the domain: a concept hierarchy going from the most general to the most specific concepts and not requiring any other relationship in addition to *subsumptions*. A concept A is subsumed by a concept B ($A \leq B$) if the set of instances classified under A is intensionally constrained to be equal to or a subset of the set of instances classified under B: $A \subseteq B$. Subsumption models taxonomic IS-A relationships. In this case, $A \leq B$ means either that $A \equiv B$ or that A is a descendant of B in the taxonomy, so that subsumptions define a partial order among concepts. Directed acyclic graph taxonomies modeling multiple inheritance are supported.

Objects in the extension are abstract, and consequently objects of any type and format can be managed in a uniform way. Abstract objects can be freely classified under n ($n > 1$) concepts at any level of abstraction (i.e. at any level in the conceptual tree). This multidimensional classification is a generalization of the monodimensional classification scheme used in conventional taxonomies and models common real-life situations. First, objects are very often about different concepts: for example, the present book can be classified under "information retrieval", "ontologies", "multimedia databases", etc. Second, objects to be classified usually have different features, perspectives, or facets (e.g., Time, Location, etc.), each of which can be described by an independent taxonomy. Taxonomies with a multidimensional classification will be called *multidimensional taxonomies*.

---

[4]The reader is referred to Gärdenfors [116] for a detailed discussion of multidimensional spaces in cognitive science and concept formation, induction and semantics.

[5]The term *object* is used to denote an abstract information item which is atomic and whose content and medium are transparent to the model. Objects have been called terms, documents, items, resources, and atoms in literature.

1.2.1 Concepts

A concept C is just an abstract label which identifies all the objects classified under C. Concepts are not textual terms. Although concepts are often externally shown as textual labels, these labels need not have any connection with terms contained in the objects they represent; incidentally, such objects are not necessarily textual. In addition, although concept labels are used to convey the concept meaning to users, such a meaning is not exploited by the model, so that for modeling purposes, each concept can be identified by a unique abstract identifier (such as a unique numeric id).

Concepts are defined by their extension rather than by specific properties they exhibit. Two different types of extension for a concept C are defined. The shallow extension of C (denoted by $shallowExtension(C)$) is defined as the set of objects directly classified under C. The deep extension of C (denoted by $deepExtension(C)$) includes all the shallow extensions for the conceptual subtree rooted in C:

$$deepExtension(C) \\ = \left\{d \mid d \in shallowExtension\left(C'\right) \wedge \left(C' = C \vee C' \textit{ is a descendant of } C\right)\right\}$$

or, equivalently

$$deepExtension(C) = \left\{d \mid d \in shallowExtension\left(C'\right) \wedge \left(C' \leq C\right)\right\}$$

By construction, the shallow and the deep extension for a terminal (leaf) concept are the same. An example of each type of extension is given in Fig. 1.1. The intension is above the line, and the extension is below the line. Circles represent concepts, and objects are represented by rectangles. Solid arcs represent subsumptions, and dotted arcs represent classifications.

We claim that the 'natural' semantics of C is the deep extension of C, because a specific level of abstraction subsumes all its specializations. When we refer to animals we include dogs, cats, aardvarks, etc., and all the objects classified under them. Therefore $objects(C) = deepExtension(C)$. For this reason, the *extension of*

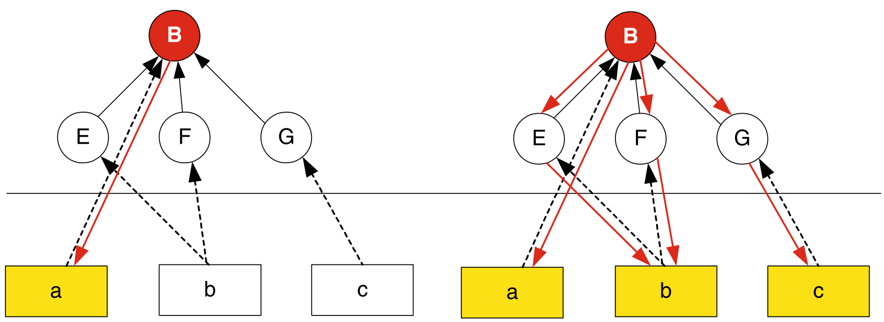

Fig. 1.1 Shallow (at *left*) and deep (at *right*) extensions for concept B

a concept C refers to the deep extension of C in the following, unless otherwise noted.

Subsumptions require that an inclusion constraint is maintained. If C' is a descendant of C in the taxonomy, then $objects(C') \subseteq objects(C)$. If shallow and deep extensions are explicitly stored, this results in a form of *backward inheritance*. An object in the shallow extension of a concept C is also classified in the deep extension of each ancestor of C. The deep extension of the root concept of the taxonomy includes the entire universe U of objects.

The shallow extension is needed only if objects can be classified at any level of the conceptual tree. In the rather common case in which objects are only classified under terminal concepts, the shallow extension is not required, because it is empty for non terminal concepts, and equivalent to the deep extension otherwise.

An immediate consequence of the interpretation of concepts as sets of objects is that logical operations on concepts can be performed by the corresponding set operations on their extensions. The information base can be manipulated and derived concepts created by combining concepts through the normal logical operations (and, or, not).

1.2.2 Relationships Among Concepts

1.2.2.1 The Base Extensional Inference Rule

In conventional taxonomies, concepts are only related by subsumptions. In a dynamic taxonomy, concept relationships other than subsumptions are inferred through the extension only, according to the following base extensional inference rule:

Two concepts A and B are related (denoted by $A \rightleftarrows B$) iff there is at least one object d in the extension which is classified at the same time under A or under one of A's descendants and under B or under one of B's descendants.

For example, an *unnamed*[6] relationship between terrorism and New York can be inferred, if an object classified under terrorism and New York exists. At the same time, since New York is a descendant of USA, also a relationship between terrorism and USA can be inferred. The extensional inference rule can be seen as a device to infer relationships on the basis of empirical evidence.

Equivalent definitions of the base extensional inference rule are:

$$A \rightleftarrows B \quad \text{iff} \quad \exists o \in U : \ o \in objects(A) \land o \in objects(B)$$
$$A \rightleftarrows B \quad \text{iff} \quad objects(A) \cap objects(B) \neq \emptyset$$

Subsumption relationships in the taxonomy are, by definition, a special case of the extensional inference rule.

[6]Although the name or type of relationships cannot be known through the inference rule only, they can be conveyed through specific design rules for the taxonomy as discussed in Sect. 7.1.1.

1.2.2.2 The Extensional Inference Rule

The base extensional inference rule can be extended to cover the relationship between a given concept C and a concept expressed by an arbitrary subset S of the universe: C is related to S iff there is at least one object o in S which is also in $objects(C)$ or, equivalently, iff $objects(C) \cap S \neq \emptyset$.

Hence, the extensional inference rule can infer relationships not only for base concepts, but also for any logical combination of concepts. Since it is immaterial how S is produced, dynamic taxonomies can infer relationships between a concept and sets of objects produced by other retrieval methods such as database queries, shape retrieval, etc. and, therefore, access through dynamic taxonomies can be easily combined with any other retrieval method. The extensional inference rule reduces to the base rule when the set S is the deep extension of a concept C'.

Given a taxonomy, the set of concepts related to a set S is called the *related set of concepts* ($RS(S)$) and is defined as: $RS(S) = \{C \mid objects(C) \cap S \neq \emptyset\}$. Because of the inclusion constraint in subsumption, if $C \in RS(S)$ and C' is an ancestor of C, then $C' \in RS(S)$. Conversely, if $C \notin RS(S)$ and C' is a descendant of C, then $C' \notin RS(S)$.

In interaction, it is quite useful to know for each C related to S, how many objects are in the intersection $objects(C) \cap S$. We call this quantity *related count*, $rc(C|S)$: $rc(C|S) = |objects(C) \cap S|$. By definition, $rc(C|U) = |objects(C)|$. The related set of concepts can be reformulated in terms of the related count: $RS(S) = \{C \mid rc(C|S) > 0\}$.

1.2.3 Reduced Taxonomies and Exploration

Given a set of objects S, the extensional inference rule can be used to produce a conceptual summary of S according to the original taxonomy by simply pruning from the taxonomy all those concepts C which are not related to S, i.e. all $C \notin RS(S)$. This taxonomy is called a *reduced taxonomy*, $RT(S)$.

The fundamental idea for user-centered exploration is to use the taxonomy to:

(a) set an *interest focus* as a boolean combination of concepts or through an external query, and
(b) summarize concepts related to the interest focus through a reduced taxonomy, from which unrelated concepts are pruned.

This means that the original taxonomy can adapt to and summarize any subset of the universe (hence the term *dynamic taxonomy*), whereas traditional static taxonomies can only summarize the entire universe.

The initial user interest focus F_0 is the universe U, i.e., all the objects in the information base. The user is initially presented with a tree representation of the initial taxonomy for the entire knowledge base. Each concept C has also a related count

$rc(C|F_i)$.[7] The related count is a function of the current focus F_i. Since $F_0 = U$, $rc(C|F_0)$ is equivalent to $|objects(C)|$, i.e., to the cardinality of the deep extension of C.

In the simplest case, the user selects a concept C in the taxonomy and zooms over it. The *zoom operation*[8] changes the current state in two ways. First, the current focus F_i becomes $F_{i-1} \cap objects(C)$. Objects not in the focus are discarded. Second, the tree representation of the taxonomy is modified to summarize the new focus. All and only the concepts related to F_i are retained and the count for each retained concept C' is updated to reflect the number of objects in the focus F_i which are classified under C', i.e., $rc(C'|F_i)$. The reduced taxonomy is derived from the initial taxonomy by pruning all the concepts not related to F_i,[9] and it is a conceptual summary of the set of objects identified by F_i, in the same way as the original taxonomy is a conceptual summary of the universe.

The retrieval process can be seen as an iterative thinning of the information base: the user selects a focus, which restricts the information base by discarding all the objects not in the current focus. Only the concepts used to classify the objects in the focus and their ancestors are retained. These concepts, which summarize the current focus, are those and only those concepts which can be used for further refinements. From a human-computer interaction perspective, the user is effectively guided to reach his goal by a clear and consistent listing of all possible alternatives, and this interaction is often called *guided thinning* or *guided navigation*.

Figures 1.2–1.5 show how the zoom operation works. Figure 1.2 shows a dynamic taxonomy.

We assume that the user zooms on concept C. First, the interest set (i.e. the extension of the interest focus) is computed. In this case, the interest set is the deep extension of C, i.e., $deepExtension(C) = \{b, c, d\}$. This set is computed by following downwards in Fig. 1.3 all the arcs incident to C or one of its descendants $\{H, I\}$. All the objects not in the interest set can be removed from the extension, which is therefore reduced.

Then the reduced taxonomy for this interest set is computed. First, all the concepts related to the interest focus ($RS(C)$) are computed. According to the extensional inference rule, these are all the concepts under which at least an object in the interest set is classified. In Fig. 1.4, we compute this set by following all the arcs originating from the interest set. The set of concepts related to the focus is therefore $RS(C) = \{F, G, H, I, B, C, A\}$.[10]

---

[7] Related counts are generally considered very important in guiding user interactions. However, some systems do not show them, for performance reasons.

[8] Also called zoom-in afterwards.

[9] For simplicity, we assume here that pruned concepts are not shown to the user. However, the very fact that a concept was pruned might be an important information for the user. In this case, pruned concepts are shown to the user in an appropriate format which indicates that they have a related count equal to zero and are not selectable for additional zoom operations.

[10] As remarked before, if a concept is related to a set S, also all of its ancestors are related to S.

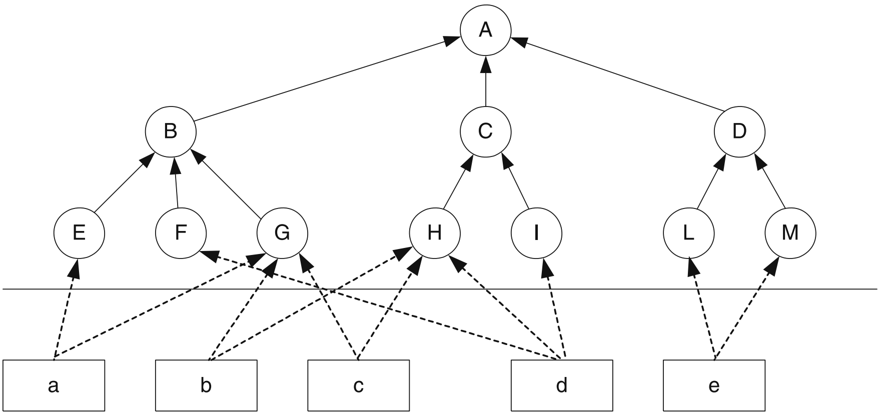

Fig. 1.2 A dynamic taxonomy: the intension is above the line, the extension below. *Solid arcs* denote subsumptions, and *dotted arcs* represent classification

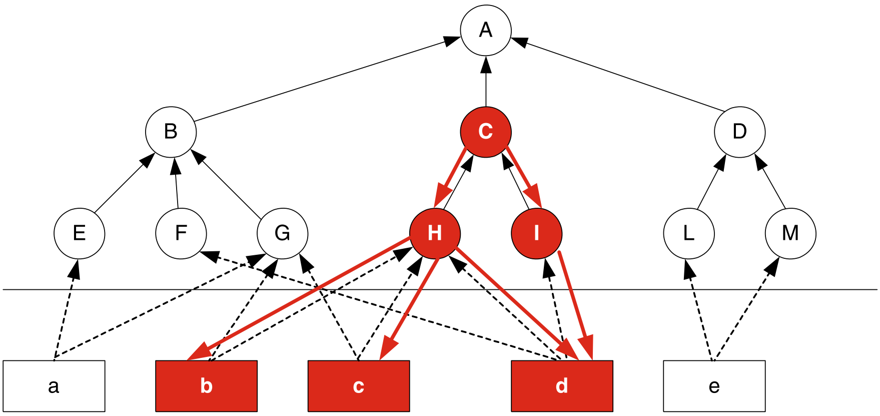

Fig. 1.3 Computing the deep extension for C

Finally, in Fig. 1.5, all the concepts not in $RS(C)$ are removed from the intension, thus producing a reduced taxonomy which fully describes all and only the objects in the current focus C.

A subsequent zoom operation can only be performed on concepts in the reduced taxonomy, i.e., on concepts in $RS(F_1)$. This guarantees that no empty results are ever found, by construction, and that, at any stage, unrelated (and therefore irrelevant) concepts are discarded. The selection of a concept C_2 from the reduced taxonomy determines a new focus $F_2 = F_1 \cap C_2$. For example, the selection of G in the context of F_1 determines $F_2 = \{b, c\}$ and $RS(F_2) = \{G, H, B, C, A\}$.

It is possible to extend zoom operations defined on a single concept to zooms defined on expressions of concepts or on the result of external query methods.

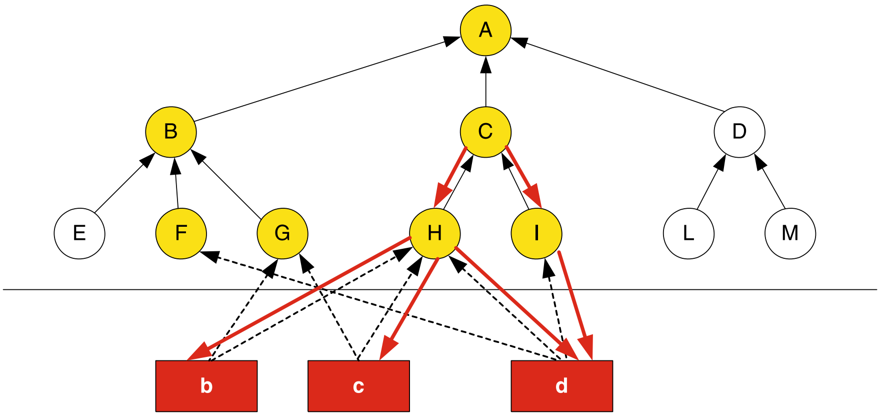

Fig. 1.4 Computing the related set for C ($RS(C)$)

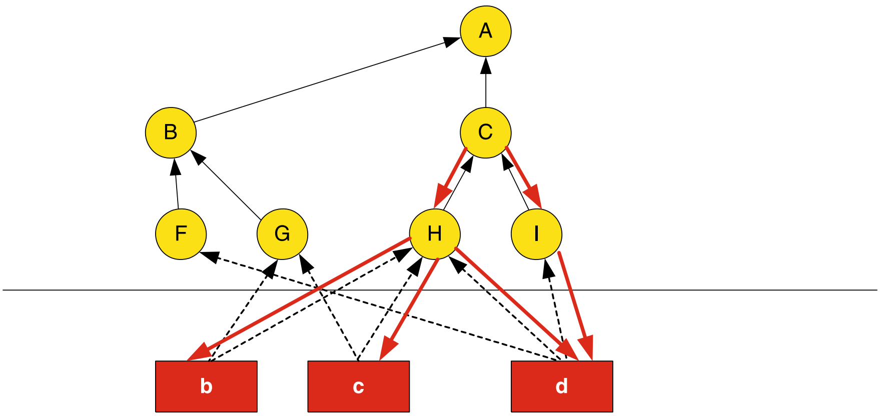

Fig. 1.5 The reduced taxonomy for focus C

1.3 Implications for Schema Design

The derivation of concept relationships through the extensional inference rule has important implications on conceptual modeling. First, a fundamental point is that relationships among concepts need not be anticipated but can be inferred from the actual classification. This simplifies taxonomy creation and maintenance. In traditional approaches, only the relationships among concepts explicitly described in the conceptual schema are available to the user for browsing and retrieval. Therefore, all possible relationships must be anticipated and described: a very difficult if not impossible task. In dynamic taxonomies, no relationships in addition to subsumptions are required, because conceptual relationships are automatically derived from the actual classification. For this reason, dynamic taxonomies easily adapt to new relationships and are able to discover new, unexpected ones.

Second, since dynamic taxonomies synthesize compound concepts, these usually do not need to be represented explicitly. This property removes the main cause of the combinatorial growth of traditional taxonomies. Sacco [236] developed guidelines which produce taxonomies that are compact and easily understood by users. Some are similar to faceted classification [134, 224], at least in its basic form: the taxonomy is organized as a set of independent, 'orthogonal' subtaxonomies (facets or perspectives) to be used to describe data. Additional guidelines [236] address human-computer interaction and recommend a fanout (number of children for a given concept) no larger than ten and a taxonomy depth no higher than four levels. A larger fanout results in longer lists which have to be scanned to find the desired concept; a larger depth makes hierarchy traversal harder. Fanout and depth guidelines suggest taxonomies with a number of terminals ranging from 1,000 to 10,000, which are usually adequate for faceted taxonomies.

As an example of faceted design guidelines, consider a compound concept such as '15th century Italian paintings'. It can be split into its facets: a Location taxonomy (of which Italy is a descendant), a Time taxonomy (of which the fifteenth century is a descendant) and finally an Art taxonomy (of which painting is a descendant). The objects to be classified under the compound concept are classified under Location>Italy, Time>15th century and Art>Painting instead. The extensional inference rule establishes a relationship among these concepts and the compound concept can be recovered by zooming on any permutation of them.

In a conventional classification scheme, such as Dewey indexing [87], in which every object is classified under a single concept, a number of different concepts equal to the Cartesian product of the terminals in the three taxonomies must be defined.[11] Such a combinatorial growth either results in extremely large conceptual taxonomies or in a gross conceptual granularity [236]. In addition, faceted design coupled with dynamic taxonomies makes it simple to focus on a concept, e.g., 15th century, and immediately see all related concepts such as literature, painting, politics, etc., which are recovered through the extensional inference rule. In the compound concept approach, these correlations are unavailable because they are hidden inside the concept label.

An often overlooked consequence of faceted schemata is that the faceted design process breaks relationships into 'primitive' or 'free-standing' concepts which do not depend on other concepts [236]. In our experience, such primitive concepts are relatively stable in time, so that faceted schemata need little maintenance. It is the relationship among these primitive concepts that tends to vary over time, and dynamic taxonomies easily accommodate such variability by computing such relationships dynamically.

In summary, dynamic taxonomies require a simpler schema, dynamically adapt to new relationships, and simplify user discovery of unexpected relationships.

[11] Usually, many of these concepts are meaningless (see Sect. 6.1).

1.4 Advantages

The advantages of dynamic taxonomies are especially significant in the areas of:

- user interaction, which is simple and natural;
- exploration, where dynamic and unexpected relationships are fully accounted for;
- schema design, where a faceted structure leads to minimal and flexible schemata, and
- search effectiveness, because dynamic taxonomies have an extremely fast convergence to small result sets.[12]

Dynamic taxonomies require a very light theoretical background: the concept of a taxonomic organization and the zoom operation, which seems to be very quickly understood by end-users. The user is effectively guided to reach his goal, because at each stage he has a complete list of all the concepts related to the current focus, which can be used to further refine his exploration. By construction, no empty results can occur because all the unrelated concepts which can cause empty results are automatically pruned and cannot be selected. Though guided, the user is in control of interaction and is free to explore the information base according to his interests: only the concepts which lead to dead ends are pruned, and inferred concept relationships are symmetric.

User control and the adaptive mechanism of dynamic taxonomies, which rapidly becomes clear and transparent, encourage the user to experiment and explore, and gives him the feeling that he has considered all the alternatives in reaching a result. This is confirmed by usability studies [134, 327] conducted on a corpus of art images. Despite slow response times, access through a dynamic taxonomy was shown to produce a faster overall interaction and a significantly better recall (both actual and perceived) than access through text retrieval.

We believe that significant advantages in ease of interaction are a result of using the same representation for querying (setting the focus) and summarizing, so that the user always deals with a single conceptual representation of the infobase. We call this property *self-adapting exploration*.[13]

The user-centered approach of dynamic taxonomies is inherently more acceptable to users than the system-centered approach of agent-mediated knowledge-based systems, so that dynamic taxonomies appear to be the model of choice for several important areas, such as medical diagnosis where knowledge-based systems have not been successful.

Exploration of complex infobases is free. Any combination of concepts (AND, OR, NOT) is supported by the corresponding set operations on their deep extensions, so that arbitrarily complex foci can be defined. Since the reduced taxonomy is dynamically produced, the model adapts to dynamic relationships among concepts (i.e., relationships which can vary in time) and lead to the discovery of unexpected

[12] See Sect. 3.2.

[13] Self-adapting exploration structures will be discussed in detail in Chap. 3.

relationships that even the designer might not be aware of. Most importantly, exploration is not restricted to conceptual manipulation only, but can applied to any traditional search method (e.g. text retrieval, shape retrieval, etc.) because any subset of the universe can be summarized by dynamic taxonomies through a dynamically-created reduced taxonomy.

Since concepts can be dynamically combined at run-time, schemata for dynamic taxonomies are simple and minimal. They are based on subsumptions only and there is no need for compound concepts, which are the major cause of the combinatorial explosion of conventional taxonomies. No assumption is made on object contents, type or format, so that any type of heterogeneous information can be managed in a uniform way. Finally, in dynamic taxonomies, concepts are abstract entities, and no assumption is made on concept labels. Consequently, an appropriate architecture makes the support of multilingual access very easy because it only requires the translation of concept labels into a different language.

The advantages of dynamic taxonomies with respect to the convergence of exploratory patterns are dramatic. The analysis by Sacco[14] [247] shows that 3 zoom operations on terminal concepts are sufficient to reduce a 10,000,000 object information base described by a compact taxonomy with 1,000 concepts to an average of 10 objects. Experimental data on a real newspaper corpus of over 110,000 articles, classified through a taxonomy of 1,100 concepts, reports an average 1,246 objects to be inspected by the user of a static taxonomy vs. an average 27 objects after a single zoom on a dynamic taxonomy.

Finally, the conceptual organization of dynamic taxonomies simplifies the gathering of user interests at a precise conceptual level by simply (and unobtrusively) monitoring the zoom operations issued and the concepts the user focuses on. At the same time, such an organization also allows the user to explicitly specify his own interests in terms of concepts so that precise push strategies can be implemented [231, 232, 242]. Recommendations can also be easily integrated, since reviews, popularity, etc. can be represented by specific facets in the dynamic taxonomy.

1.5 Application Areas

The main industrial application is currently e-commerce. Assisted product selection is a critical step in most large-scale e-commerce systems and the advantages in interaction are so significant as to justify the restructuring of most e-commerce portals, such as *Yahoo!* Shopping.

However, dynamic taxonomies have an extremely wide application range and a growing body of literature indicates that their adoption benefits most web applications. In addition to e-commerce, e-auctions, and e-catalogs, key areas such as e-government, human resources and job placement, news portals, multimedia,

[14] See Sect. 3.2.

medical guideline and diagnostic systems are being investigated and commercial solutions deployed.

Although most current applications fall in the object-seeking category, there is an emerging and growing interest in knowledge-seeking applications, while wisdom-seeking applications are currently at an initial stage. A detailed discussion of existing and emerging application areas is found in Chap. 9.

1.6 Faceted Search and Dynamic Taxonomies

Thus far, we have used the term *faceted search* as a synonym of *dynamic taxonomies*. However, faceted classification is just a design guideline, akin to normalization in relational databases, and the dynamic taxonomy model only requires a multidimensional taxonomy. Indeed, there are practical situations in which the violation of the orthogonal organization of facets is beneficial or required. For example, consider the topic taxonomy of a legal database organized for experts, and the same taxonomy organized for laymen. These two subtaxonomies are not facets as they are not orthogonal, but provide two different and useful access paths to the same infobase. So, it can be contended that the term faceted search is misleading since it focuses on an unessential feature. The concepts of extensional inference and of reduced taxonomies, which are a fundamental part of dynamic taxonomies, are not implied by (and do not require) a faceted classification.

In addition, faceted search is presented in literature through examples rather than being formally defined [248], and this has obviously caused a certain amount of confusion. Faceted search can be (and sometimes has been) taken at face value to mean any system based on a faceted classification. This covers very diverse solutions, ranging from the work of Prieto-Diaz [216], based on a faceted classification which can be composed by boolean operators but with no conceptual summarization capabilities, to early attempts by Amazon and Microsoft Knowledge Manager, where faceted subtaxonomies are completely independent and cannot be composed. Most importantly, it is impossible to reason in a rigorous way on features, extensions and challenges without a formal model.

Faceted search systems, though later, are subsumed by the more general dynamic taxonomy model. Although there is a clear evolution towards the more general model, some commonly found restrictions include, among others:

- attribute-value shallow taxonomies. In most systems, a two-level taxonomy is used, describing attributes with their values as immediate children. For example, attributes for a digital camera can be Price, Brand, Weight, etc. Each attribute A will have a number of children, each representing a value of A. This is a significant restriction with respect to the more general multi-taxonomy model supported by dynamic taxonomies, since it does not support any abstraction capabilities. However, this is a frequent approach because it allows a straightforward mapping of database relations to a dynamic taxonomy in which the top-level facets are the attributes.[15]

- objects classified under terminal concepts only. This is really an implication of the previous restriction, because in attribute-value shallow taxonomies attributes are obviously abstract, and it makes no sense to classify an object directly under a specific feature. In general dynamic taxonomies this is not true. For example, a news item may be about an entire nation, a specific region, or a specific town.
- AND only refinement. Whereas dynamic taxonomies support all boolean operations on concepts, most faceted search systems only support refinement by a single selected concept. The lack of OR composition capabilities implies that the user is unable to focus on custom groups obtained by considering different concepts in the taxonomy as equivalent. As an example, consider a user interested in Balkan countries only: either a specific concept grouping such countries has been anticipated and exists in the schema or the user is unable to explore features for Balkan countries, unless OR composition is supported.

1.7 Book Roadmap

Dynamic taxonomies are user-centered and deal holistically with a number of different aspects which include modeling, user interaction, schema design, system implementation and performance. All of these aspects are fundamental in our approach, even though the final yardstick is whether the user can easily and effectively understand and use the model. Dynamic taxonomies are at the crossroads of several independent computer science research areas, and, for this reason, their initial acceptance by the research community was a relatively slow and difficult process because it was hard to classify them in the 'appropriate' conferences and journals, and contributions were usually and incorrectly considered from a single perspective only. On the contrary, the acceptance by the industrial community, once started, was very quick and pervasive.

For the purpose of exposition, the book has been divided into the following five main areas:

- *Modeling issues.* This is the focus of the next two chapters. Chapter 2 describes faceted taxonomy-based information sources. In Chap. 3, dynamic taxonomies and faceted search are compared to other techniques including information retrieval, OLAP, dynamic result clustering, static taxonomies, decision trees, formal concept analysis and description logics. Finally, Chap. 5 extends the model by addressing data mining, structured documents, and extended expressivity through logical information systems, and deals both with modeling and user interaction issues.
- *User Interaction issues.* Chapter 4 provides an extensive analysis of information presentation, interaction modes, user interface design patterns, and personalized faceted search.

[15] See Chap. 7.

- *Taxonomy design.* Chapters 6 and 7 focus on taxonomy design. Chapter 6 introduces the Compound Term Composition Algebra and addresses the integration of different and distributed taxonomy-based sources. Chapter 7 describes guidelines for schema design, and discusses the automatic generation of dynamic taxonomy schemata for structured databases, focusing on relational views and on E–R schemata. Finally, the automatic construction of dynamic taxonomy schemata from text infobases is discussed.
- *Architecture, implementation, and performance issues.* Effective user interaction with dynamic taxonomies requires real-time response for zoom operations, even for very large databases. Chapter 8 discusses architectural alternatives, and analyzes RDBMS-based and special implementations. It also discusses the composition of taxonomies with Logic components.
- *Applications.* Chapter 9 analyzes current and emerging application areas, including e-commerce, multimedia, e-government, human resource management, diagnostic systems, multidimensional file systems, and geographical information systems.

http://www.springer.com/978-3-642-02358-3

Dynamic Taxonomies and Faceted Search
Theory, Practice, and Experience
Sacco, G.M.; Tzitzikas, Y. (Eds.)
2009, XVII, 340 p., Hardcover
ISBN: 978-3-642-02358-3