Computing Intensions of Digital Library Collections

Carlo Meghini[1] and Nicolas Spyratos[2]

[1] Consiglio Nazionale delle Ricerche, Istituto della Scienza e delle Tecnologie della Informazione, Pisa, Italy
meghini@isti.cnr.it
[2] Université Paris-Sud, Laboratoire de Recherche en Informatique, Orsay Cedex, France
spyratos@lri.fr

Abstract. We model a Digital Library as a formal context in which objects are documents and attributes are terms describing documents contents. A formal concept is very close to the notion of a collection: the concept extent is the extension of the collection; the concept intent consists of a set of terms, the collection intension. The collection intension can be viewed as a simple conjunctive query which evaluates precisely to the extension. However, for certain collections no concept may exist, in which case the concept that best approximates the extension must be used. In so doing, we may end up with a too imprecise concept, in case too many documents denoted by the intension are outside the extension. We then look for a more precise intension by exploring 3 different query languages: conjunctive queries with negation; disjunctions of negation-free conjunctive queries; and disjunctions of conjunctive queries with negation. We show that a precise description can always be found in one of these languages for any set of documents. However, when disjunction is introduced, uniqueness of the solution is lost. In order to deal with this problem, we define a preferential criterion on queries, based on the conciseness of their expression. We then show that minimal queries are hard to find in the last 2 of the three languages above.

1 Introduction

In a Digital Library (DL for short), collections [14,16,1] are sets of documents defined to facilitate the tasks of various DL actors, ranging from content providers for whom physical collections are provided, to users, for whom logical collections are provided. The latter kind of collections typically helps the user in carrying out information access. For discovery, the user requires a "place" where to accumulate the discovered documents, similar to the shopping cart of an e-commerce Web site. This concept is commonly known as *static* collection [20,2]. Static collections are also useful in other tasks, such as cooperative work, where they play the role of a shared information space within a community. A classical example of static collection is the *book-mark* (or *favorites*) of a Web browser. Users may also associate a description of their "view" of the DL to a collection, and access

S.O. Kuznetsov and S. Schmidt (Eds.): ICFCA 2007, LNAI 4390, pp. 66–81, 2007.

the collection whenever they need to explore this view. This concept is commonly captured by so-called *dynamic* collections [4,5,3]. Dynamic collections are not the only way users have in order to know at once the changes in the DL that may be of interest to them. Publish/subscribe (pub-sub for short) mechanisms are another way of achieving the same goal, but with a different modality: while in dynamic collection users are *active,* in the sense that they act by accessing collections, in pub-sub users are *passive,* in the sense that the system intercepts changes in the DL which may be of interest for users, and notifies them. This distinction is also known as *pull* vs. *push* access mode.

We argue that the notions of static and dynamic collections are two sides of the same coin, and propose a general notion of collection, which generalizes both. According to this notion, collections have an extension and an intension, very much like classes in object models or predicates in predicate logics. We then solve a basic problem, arising upon collection creation: the determination of the intension of a collection based on a given extension.

The paper is organized as follows: Sections 2 to 5 introduce our model of a DL, illustrating the most relevant concepts. Section 6 states in precise terms the problem we address. Sections 7 to 10 present different solutions to the problem, by examining different description languages for expressing collection intensions.

2 Terms

The basic ingredient of descriptions are *terms.* A term denotes a set of documents. As such, it may be a keyword describing the content of documents (such as *nuclear waste disposal* or *database*), or their type (*image*); or may be thought of as an attribute value (for instance, *creator= "CM"*). For generality, we do not impose any syntax on terms and treat them just as symbols making up a finite, non-empty set T, which is a proper subset of a countable domain $\mathcal{T}$, $\mathsf{T} \subset \mathcal{T}$, always containing the special term *true*, standing for truth.

Terms are arranged in a taxonomy, that is a binary relation $\leq_{\mathsf{T}}$ on T, reflexive and transitive, having *true* as the greatest element, that is

$$\forall \mathsf{t} \in \mathsf{T},\ \mathsf{t} \leq \mathit{true} \text{ and } \mathit{true} \leq \mathsf{t} \text{ implies } \mathsf{t} = \mathit{true}.$$

Based on $\leq_{\mathsf{T}}$, we define $\equiv_{\mathsf{T}}$ as follows: for any two terms $\mathsf{t}_1, \mathsf{t}_2 \in \mathsf{T}$,

$$\mathsf{t}_1 \equiv_{\mathsf{T}} \mathsf{t}_2 \text{ if and only if } \mathsf{t}_1 \leq \mathsf{t}_2 \text{ and } \mathsf{t}_2 \leq \mathsf{t}_1.$$

It is easy to see that $\equiv_{\mathsf{T}}$ is an equivalence relation. Let T_e be the set of equivalence classes induced by $\equiv_{\mathsf{T}}$, *i.e.*

$$\mathsf{T}_e = \{\ [\mathsf{t}] \mid \mathsf{t} \in \mathsf{T}\}.$$

Clearly, $[\mathit{true}] = \{\mathit{true}\}$. Furthermore, let us extend $\leq_{\mathsf{T}}$ to T_e as follows:

$$[\mathsf{t}_1] \leq_{\mathsf{T}} [\mathsf{t}_2] \text{ iff } \mathsf{t}_1 \leq_{\mathsf{T}} \mathsf{t}_2.$$