A Semantics-Based Graph for the Bib-1 Access Points of the Z39.50 Protocol*

Michalis Sfakakis and Sarantos Kapidakis

Archive and Library Sciences Department / Ionian University
Plateia Eleftherias, Paleo Anaktoro GR-49100 Corfu, Greece
{sfakakis, sarantos}@ionio.gr

Abstract. A graph of Access Points can be used as a tool in a number of applications such as, clarification and better understanding of their semantics and inter-relations, query transformations, more precise query formulation, etc. We apply a procedure to create a graph of the Access Points, according to their subset relationship, based on the official semantics of the Bib-1 Access Points of the Z39.50 protocol. In our three-step method, we first construct the relationship graph of the Access Points by testing for subset relationship between any two Access Points, and assigning each Access Point a weight value designating the number of the Access Points, which are subsets to it. In the second step, we apply a topological sorting algorithm on the graph, and finally in the last step, we reject the redundant subset relationships of the Access Points.

1 Introduction

The query mechanism of the Z39.50 [1] protocol, utilizes sets of predefined Access Points combined with specific attributes (i.e. Attribute Sets), in a number of different query language specifications (i.e. query types). One of the elements defined in an Attribute Set is the set of the valid Access Points (i.e. what entities represent the search terms, like Title, Author, etc.) from a specific set of attribute types. The Bib-1 Attribute Set is the most commonly used one, and provides the *Use* attribute type for the specification of an Access Point against which the search term is to be matched.

A semantics-based Access Point graph is necessary to better understand the exact semantics of every Access Point, as well as their inter-relationships, and can be used in a number of cases: When query optimization is involved, the use of the graph could help the transformation process of the query. Also, it could be a helpful tool for automated, or semi automated, procedures when either Access Point replacement is required, due to unsupported Access Points in a query, or Z39.50 queries are used in a heterogeneous information retrieval environment. For the end user, a better understanding of the quality of the search terms that arises from such a graph, could guide him to pose more precise (interactive) queries.

* This work was supported by the DELOS Network of Excellence on Digital Libraries (EU 6. FP IST, G038-507618).

J. Gonzalo et al. (Eds.): ECDL 2006, LNCS 4172, pp. 445 – 448, 2006.

We generate the graph of the Access Points according to the subset relationship among them, based on the MARC elements used on the definition for the semantics of the Bib-1 Access Points [2]. We recall that, the semantics of the Bib-1 Access Points implement the search requirements posed by the user community of the Z39.50 protocol. Also, the existence of the XML syntax (MARCXML) of the MARC 21 specification does not affect the derived graph, due to unchanged semantics.

Highlighting some of our results, shown in fig. 1, the Access Point *Any* is the superset of all the other Access Points. The largest component of the graph with the longest path starts from the *Author-Title-Subject*, which has as subset the *Name*, which has as subset the *Author-name*, which has as subset the *Author-name-personal*, which finally has as subset the *Name-editor* and is also linked to *Name-personal* to which is a subset.

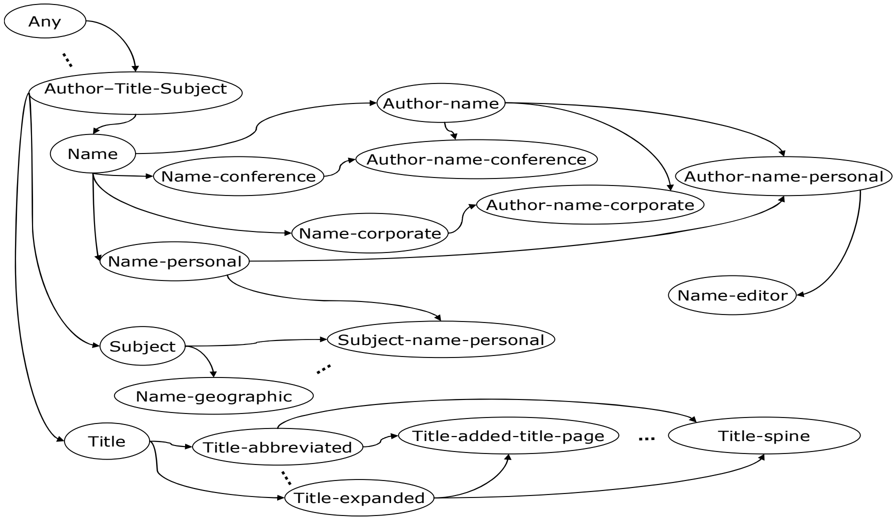

Fig. 1. A representative sample of the semantics-based Bib-1 Access Points graph

2 Method Description

The procedure to create the graph for the interrelations of the Access Points consists of three steps. Initially in the first step, we create the relationship graph of the Access Points by testing for subset relationship between all pairs of Access Points, and assigning to each Access Point a weight value designating the number of the Access Points which are subsets to it. In the second step, we apply a topological sorting algorithm on the graph. Finally, in the third step, we reject the redundant subset relationships by keeping the longest path between every pair of connected Access Points. We consider an explicit relationship as redundant, if we are able to infer its existence from other relationships of the Access Points.

We consider an Access Point to be a subset of another, if the set of the data fields used to create the first is a subset of the set of the data fields used to create the second.

As an example, consider the Access Point *Author-name* which, according to its definition [2], includes the data from the fields with MARC tags included in the set {100, 110, 111, 400, 410, 411, 700, 710, 711, 800, 810, 811}, and also, the Access Point *Author-name-personal* which includes the data from the set of fields {100, 400, 700, 800}. The Access Point *Author-name- personal* is considered being a subset of the *Author-name*.

We represent the relationships between the Access Points with a directed graph G in which the vertices represent Access Points and the edges represent subset relationships. This graph has an edge <*i, j*> if and only if Access Point *i* is a subset of Access Point *j*. The Access Points *Author-name* and *Author-name-personal*, used in the previous example, will be represented by two vertices of the graph and their subset relationship from the edge <*Author-name-personal, Author-name*>. The out-degree of a vertex expresses the number of the subsets for the represented Access Point by the vertex, as specified by the semantics definition of the Bib-1 Access Points.

The following example will better clarify our method. Let's consider that the Bib-1 Attribute Set consists only of the next four Access Points: The *Any*, the *Abstract*, the *Data-acquisition* and the *Note* Access Point. According to the Bib-1 semantics specification, the *Any* Access Point can be thought as the union of all the supported Access Points (i.e. a superset of all the others). The *Abstract* Access Point includes the data from the set of field {520}, the *Data-acquisition* includes the data from the set of field {541-subfield-d}, and finally, the *Note* Access Point includes the data from the set of fields {500, 501, …, 520, …, 535, 536, …, 541, …, 586}. We can see that all the Access Points are subsets to *Any*, and also that, the Access Points *Abstract* and *Data-acquisition* are subsets of the *Note* Access Point. Using these interrelations of the Access Points, we construct the graph G shown in fig. 2, thus completing the first step of our method.

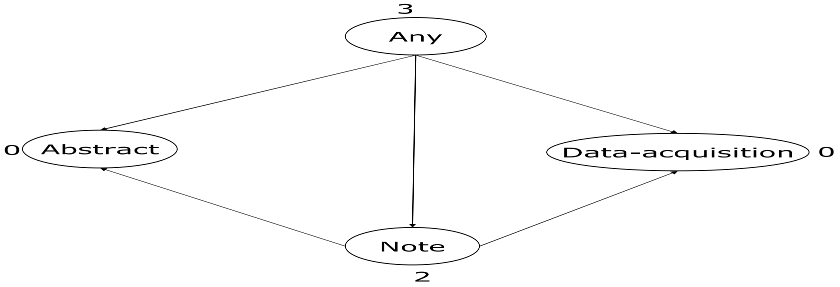

Fig. 2. Step 1: Construction of the G graph. The number near a vertex expresses its out-degree.

After applying the topological sorting algorithm on the graph, step 2, we rearrange the graph as shown in fig. 3. Obviously, this ordering is feasible due to the transitive and irreflexive properties of the proper subset relation.

At the last step we delete the derivative subset relationships. We number the vertices from left to right and for each vertex, we only keep the incoming edge from the highest numbered vertex. The resulting Graph, G0, is the minimal subset of the initial graph G, so that the transitive closure of G0 produces the graph G. The final arrangement of the graph is shown in fig. 4.

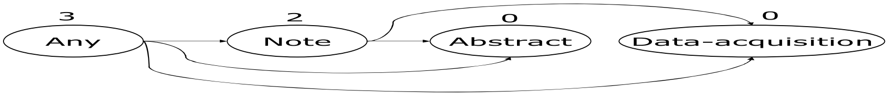

Fig. 3. Step 2: Graph G after the topological sorting

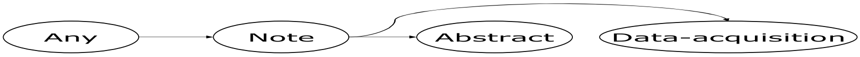

Fig. 4. Step 3: Graph G0, the minimal subset of G

3 Discussion

An important practical use of our results is when Access Point replacement is required due to unsupported Access Points in a Z39.50 query. This case is very common when we query many different Z39.50 servers. The following example illustrates some real world circumstances when a client tries to accomplish a parallel search in many sources, and also how the client could use the Access Point graph. Consider two sources, where the first one supports the Access Point *Author-name* and the second one supports the Access Point *Author-name-personal*. Obviously, all requests to the first server for selecting data using the Access Point *Author-name-personal* will fail. A smart client could substitute the Access Point *Author-name-personal* with the Access Point *Author-name* into the queries, taking into account that the Access Point *Author-name-personal* is a subset of the Access Point *Author-name*. This way, the client could avoid the failure of the query, although, unavoidably, the precision of the resulting query will be less than the precision of the original one. In this example we made the assumption that both sources support the same value combinations for the remaining attribute types (i.e. *Relation*, *Position*, *Completeness, etc.*), in order to simplify its description.

References

1. ANSI/NISO: Z39.50 Information Retrieval: application service definition and protocol specification: approved May 10, 1995.
2. Attribute Set BIB-1 (Z39.50-1995): Semantics. ftp://ftp.loc.gov/pub/z3950/defs/bib1.txt.