The World According to MARIAN

(How the document universe
is represented and
searched in the
MARIAN / Academy
Digital Library Search System)

Presentation to the
Digital Library Research Laboratory
Robert France
27 Sept. 1999

Motivation: the Document Universe

Representing the universe as objects and links

Searching in a linked representation system

Search Examples

Summary

The Document Universe

Domain for

— Digital Libraries,

— Information Systems, and

— Digital Library Information Systems

Characteristics

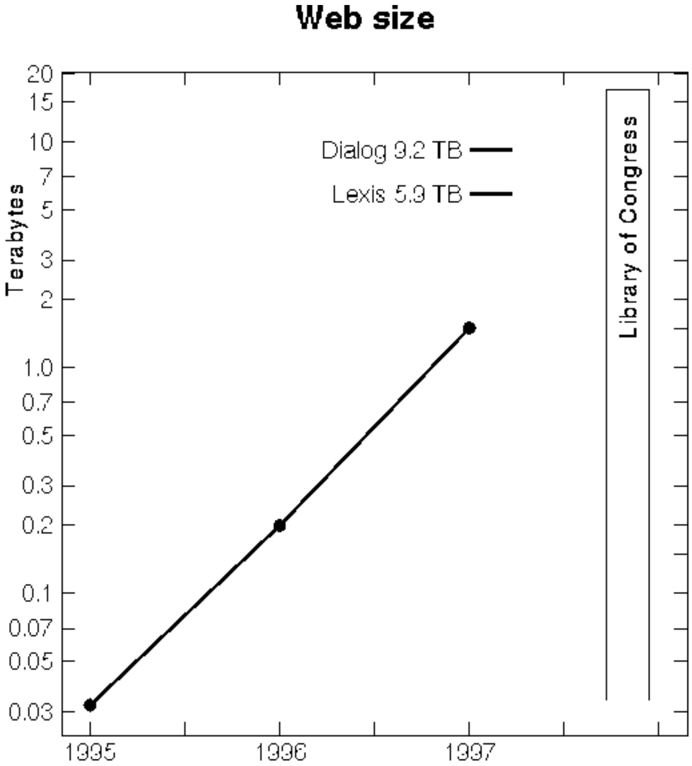
It is very large

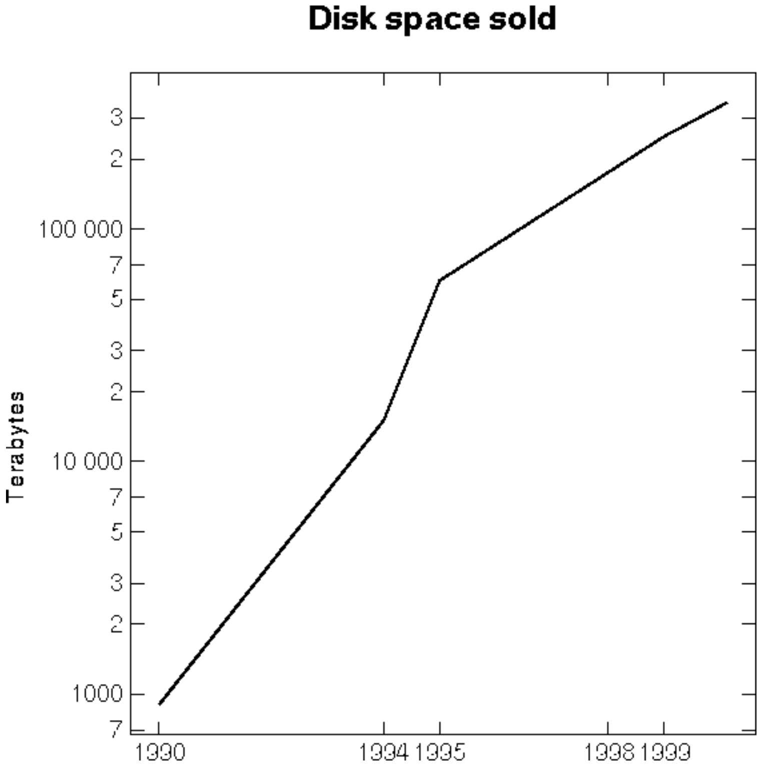
It contains Digital Information Objects

It is not limited to Digital Information Objects

It is rich in relationships

The Document Universe is Very Large





Graphs from Michael Lesk: “How Much Information Is There In the World?” (http://www.lesk.com/mlesk/ksg97/ksg.html)

The Universe Includes Digital Information Objects

These come in many many kinds ...

It is reasonable to treat the kinds as object classes, each with their proper methods.

Among the methods will be
equals()
match()
presentShort()
presentFull()
...

One particular distinguished kind is

Metadata objects

Surrogates for and descriptions of other information objects (digital or non-digital)

The Universe is not Limited to Digital Information Objects

It also includes

— other digital (non-information-bearing) objects

— other (non-digital) information objects

— digital surrogates for real-world objects

These all can also be considered object classes with their own proper methods

The Universe is Rich in Relationships

A large number of these appear “naturally” as binary and directed:

- Bibliographic (e.g., authorship)
- Lexicographic (e.g., synonymy)
- Hypermedial (Link me, baby)
- Knowledge repesentational (e.g., all the various “ISA”s)
- Object-oriented (inheritance)

These can well be represented by links.

Those that are naturally m-ary (m>2) can be represented as propositions using links in a well-defined way.

Some links are weighted; some absolute.

Therefore, Digital Libraries need Searchers that:

Can search very large collections efficiently

Can search various kinds of Digital Information Objects

Can search other classes of objects

Can follow relationships

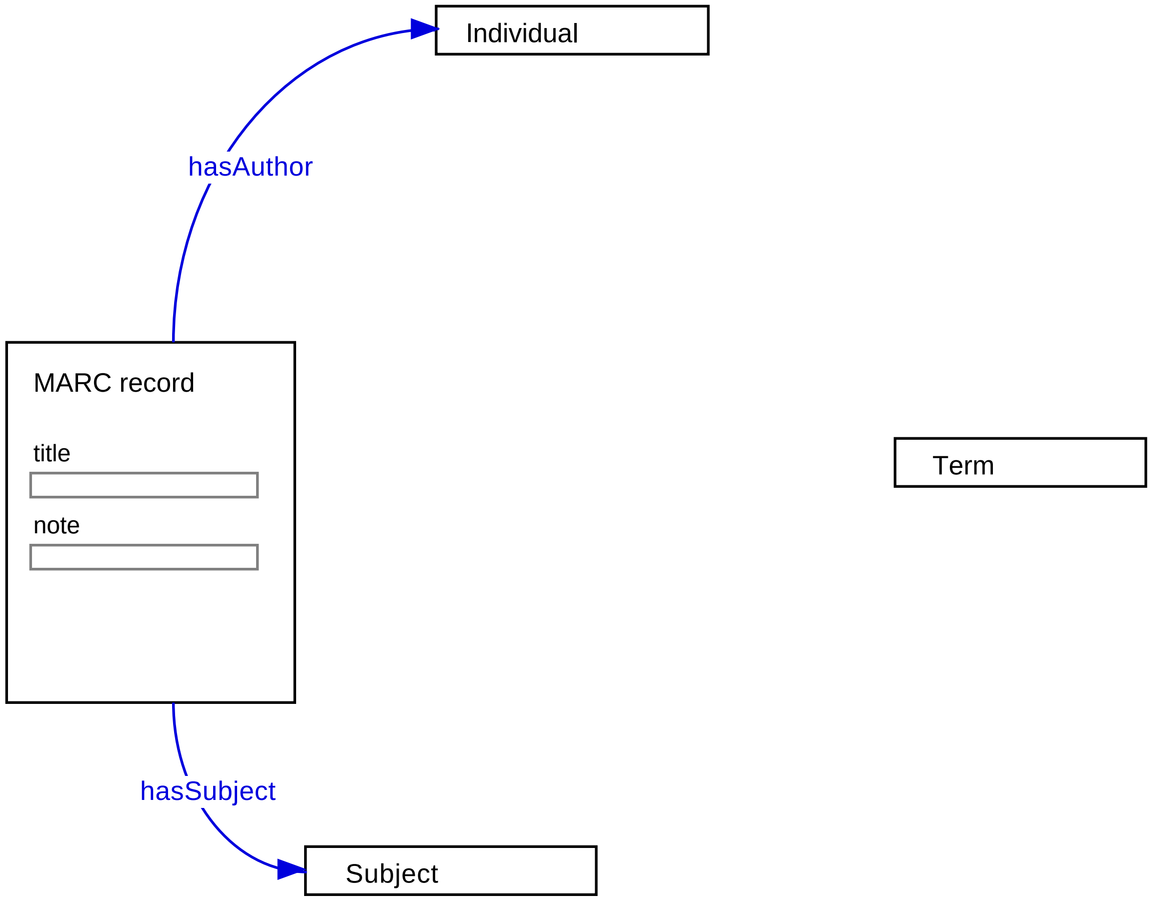
In addition, DL searchers must be:

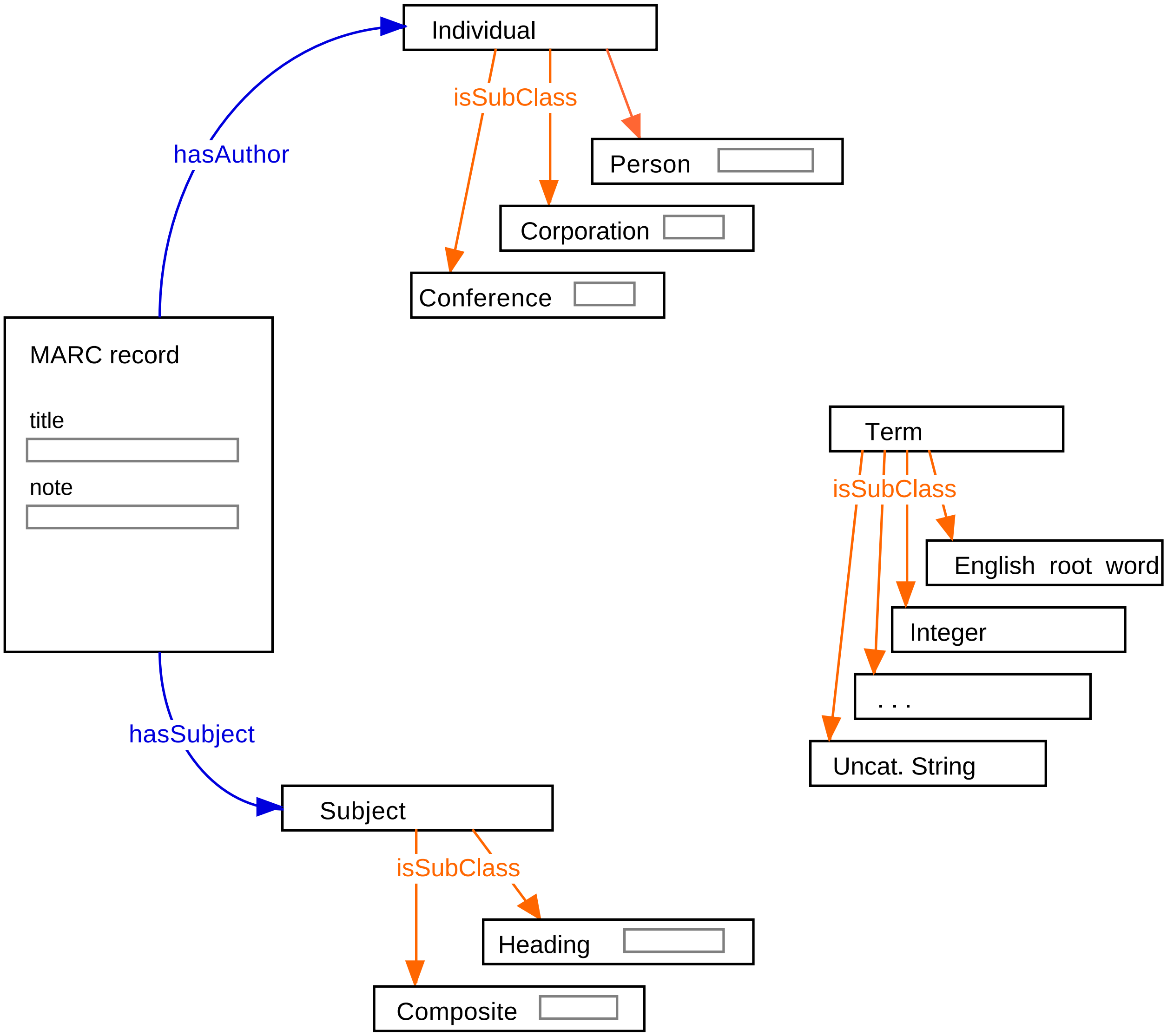
Able to search for information in context

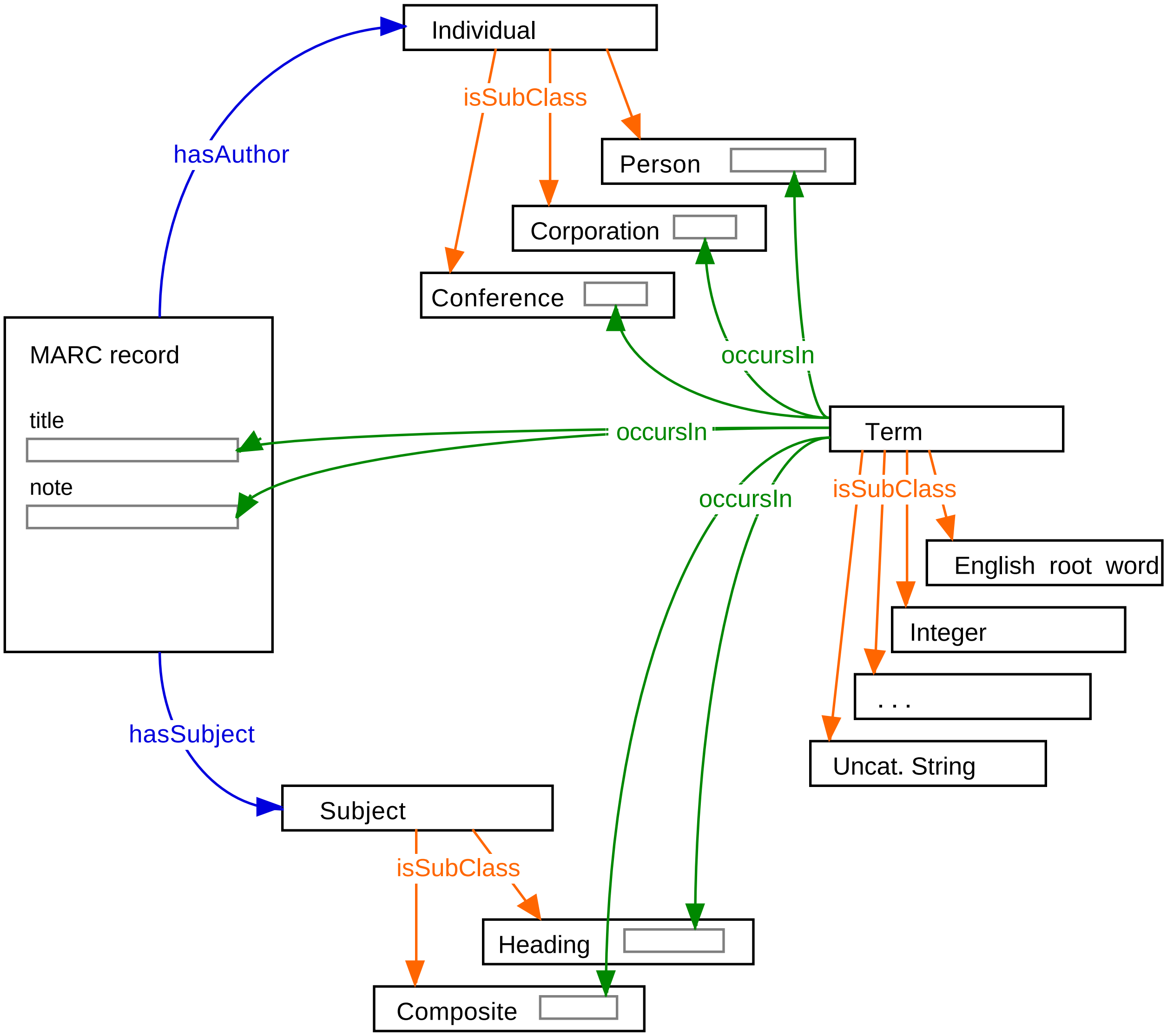
Extensible and reconfigurable

Representation in MARIAN

- Objects
  - Weighted Objects
- Links
  - Weighted Links
- Class Inheritance
- Class Managers


Individual
hasAuthor
MARC record
title
note
Term
hasSubject
Subject


Individual
isSubClass
hasAuthor
Person
Corporation
Conference
MARC record
title
note
Term
isSubClass
English root word
Integer
. . .
Uncat. String
hasSubject
Subject
isSubClass
Heading
Composite


Individual
isSubClass
Person
Corporation
Conference
hasAuthor
MARC record
title
note
occursIn
occursIn
occursIn
Term
isSubClass
English root word
Integer
. . .
Uncat. String
hasSubject
Subject
isSubClass
Heading
Composite

Weighted Object Sets

The operation of a matching function on a class of objects is a set of possible matches, each weighted by how good the match is.

A search can thus be defined as an object (or link) class method that maps object (or link) descriptions into weighted object sets.

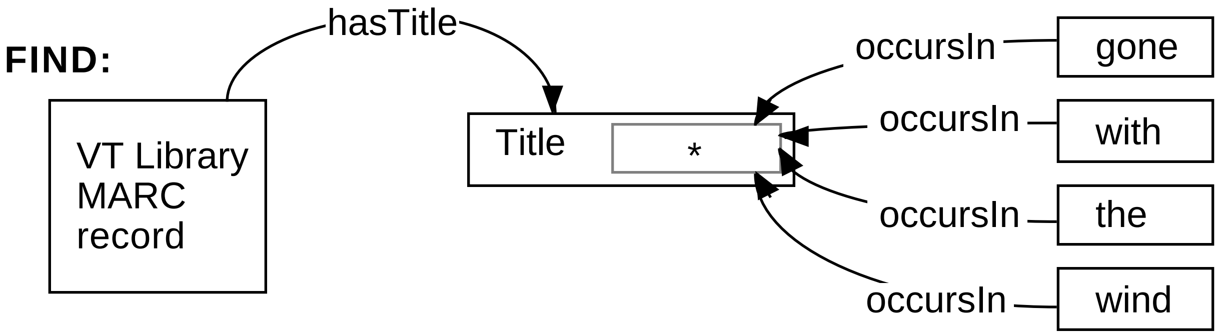
A Searcher is a class manager that defines at least one search method over the objects in the class.

Object class searchers are also a convenient place to define the semantics for combining incoming links.

The user enters:

Title: gone with the wind

MARIAN translates this to:


FIND:

VT Library MARC record

hasTitle

Title

*

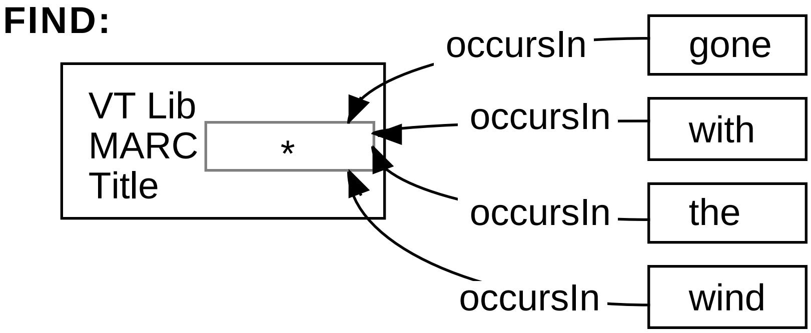
occursIn gone

occursIn with

occursIn the

occursIn wind

Which is actually searched as:


FIND:

VT Lib MARC Title

*

occursIn gone

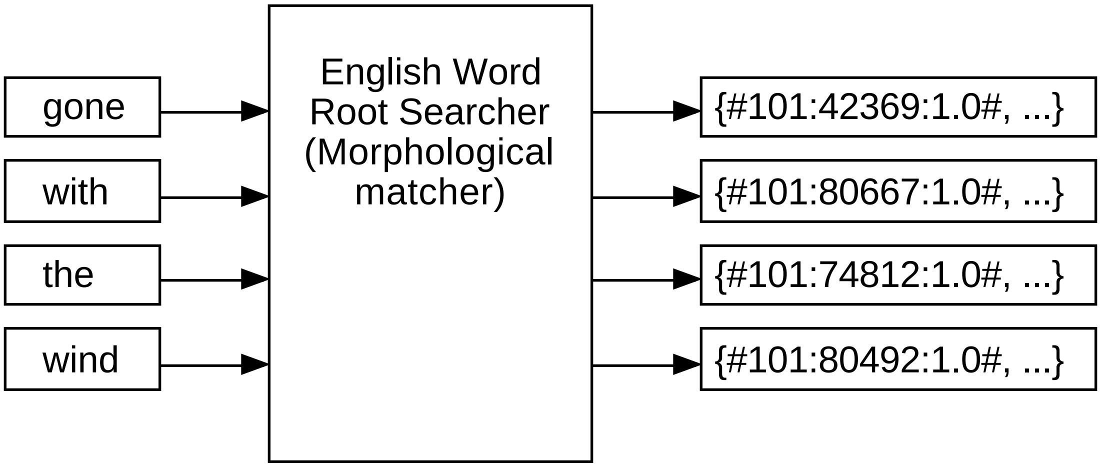
occursIn with

occursIn the

occursIn wind

(After which all the objects found have their ClassIDs reset to VT_MARC_RECORD.)

First, query strings are mapped into WtdObjSets of term IDs.



gone

with

the

wind

English Word Root Searcher (Morphological matcher)

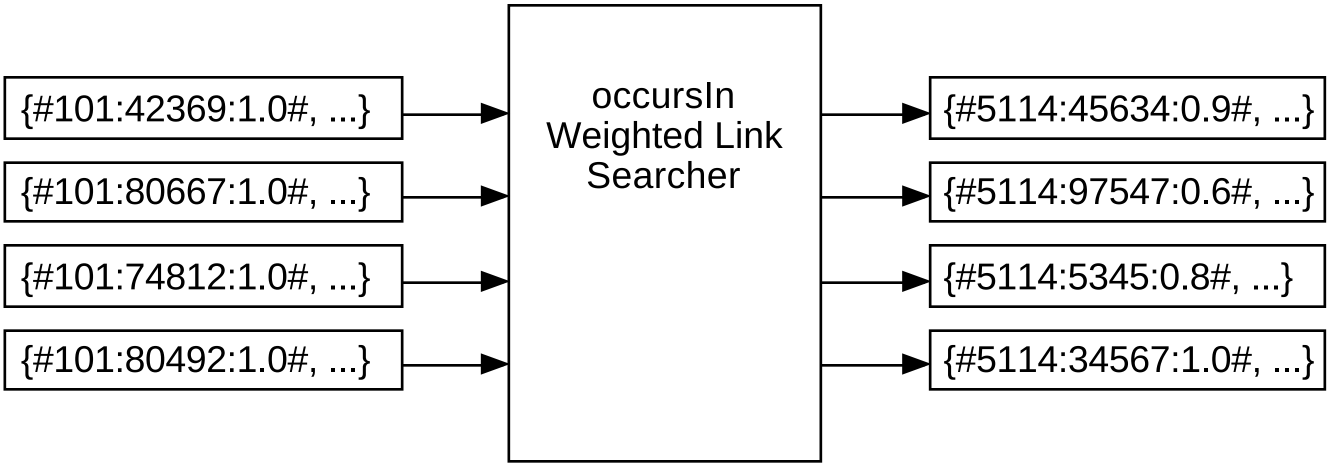
{#101:42369:1.0#, ...}

{#101:80667:1.0#, ...}

{#101:74812:1.0#, ...}

{#101:80492:1.0#, ...}

The weighted link searcher for occursIn links maps each set to a WtdObjSet of titles in which those terms occur.



{#101:42369:1.0#, ...}

{#101:80667:1.0#, ...}

{#101:74812:1.0#, ...}

{#101:80492:1.0#, ...}

occursIn Weighted Link Searcher

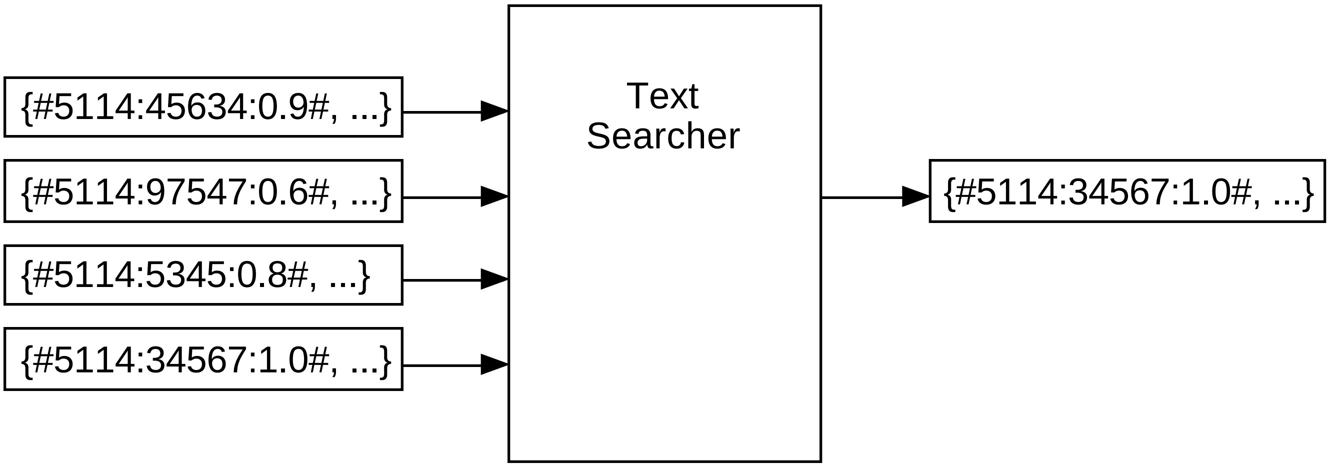
{#5114:45634:0.9#, ...}

{#5114:97547:0.6#, ...}

{#5114:5345:0.8#, ...}

{#5114:34567:1.0#, ...}

The (title) Text Searcher combines the link sets into a single set of titles.



{#5114:45634:0.9#, ...}

{#5114:97547:0.6#, ...}

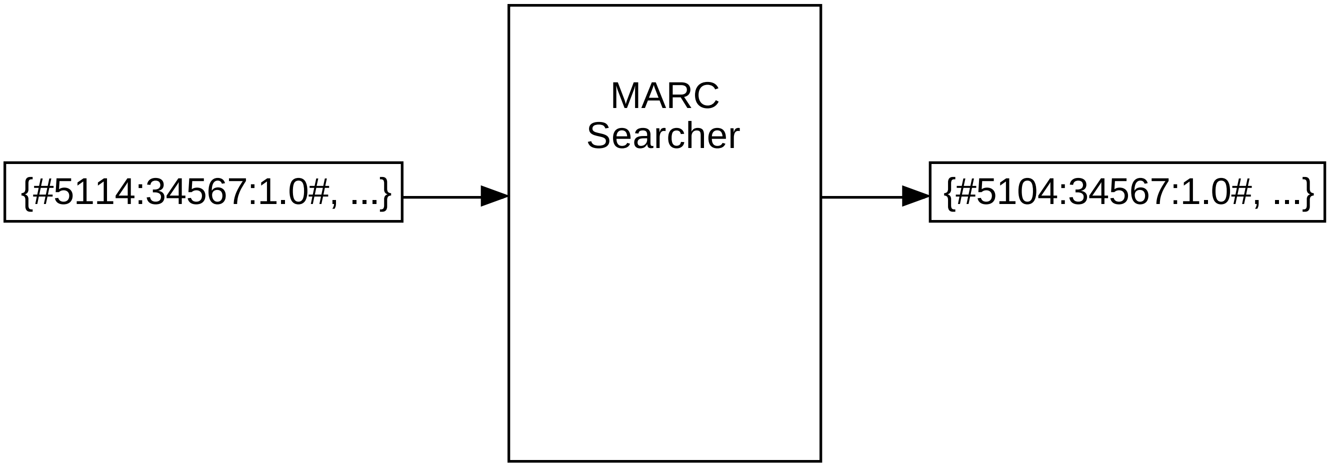
{#5114:5345:0.8#, ...}

{#5114:34567:1.0#, ...}

Text Searcher

{#5114:34567:1.0#, ...}

Finally, the MARC Searcher just rewrites the ClassIDs of the Title objects, producing a WtdObjSet of VT Library MARC objects.



{#5114:34567:1.0#, ...}

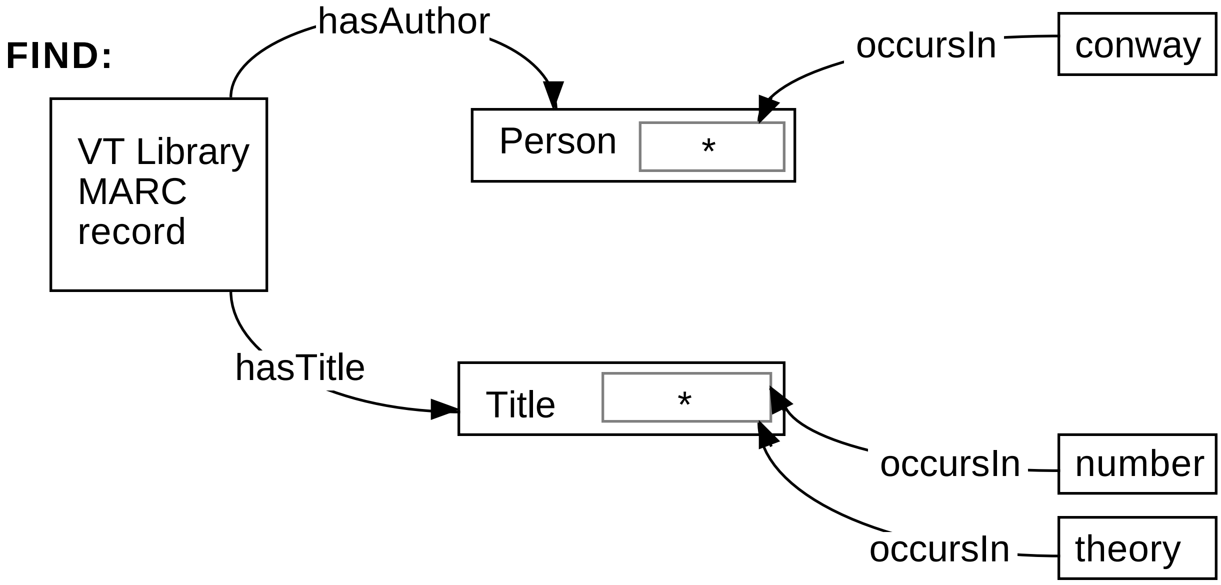
MARC Searcher

{#5104:34567:1.0#, ...}

The user enters:

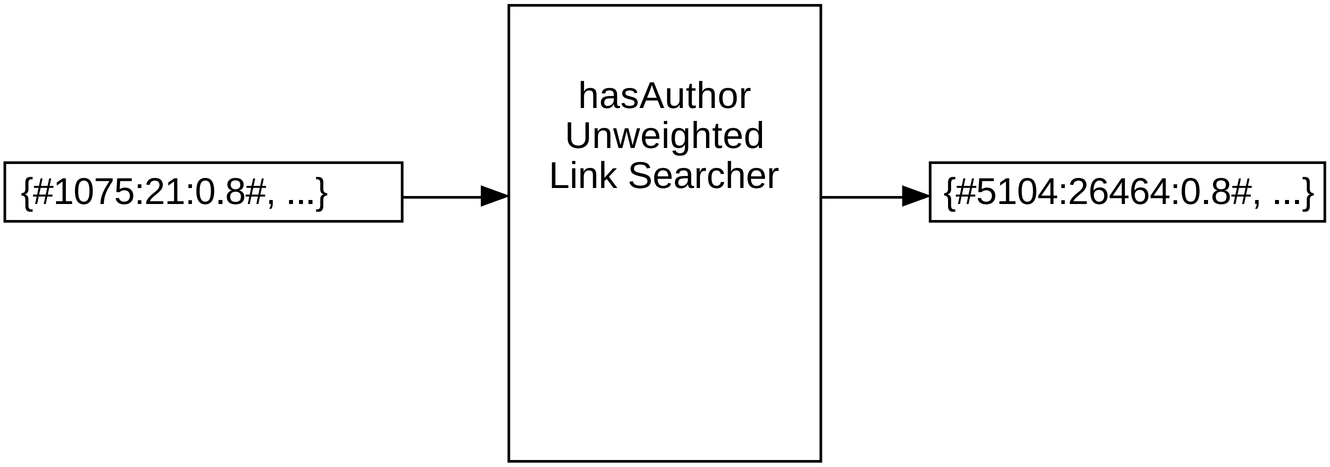
Personal Author: conway
Title: number theory

MARIAN translates this to:

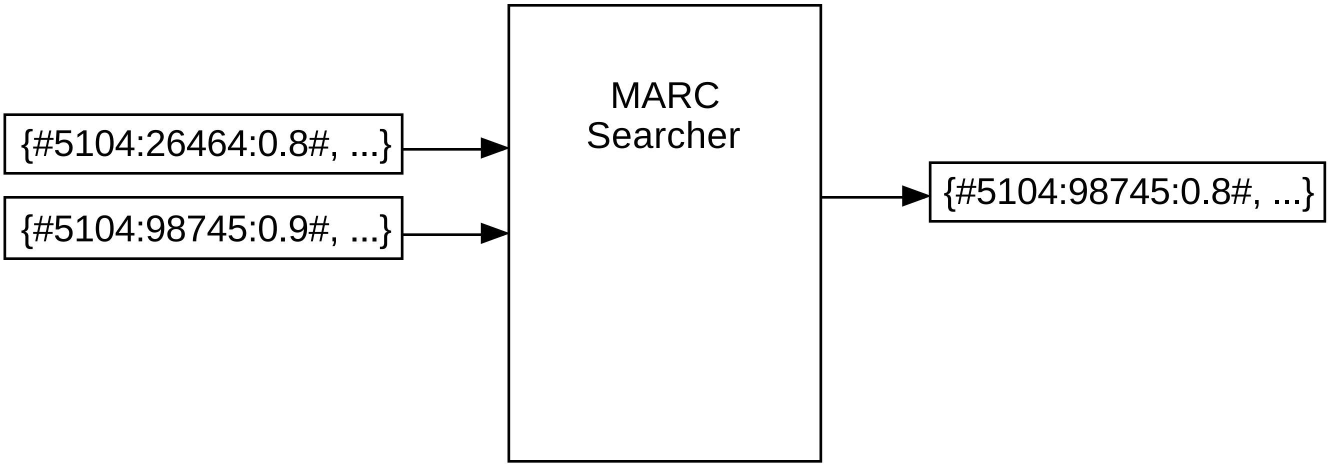


The search on Titles proceeds as before, and the search on Persons proceeds as for Title.

Traversing the (unweighted) hasAuthor link requires an additional call to the correct link searcher.



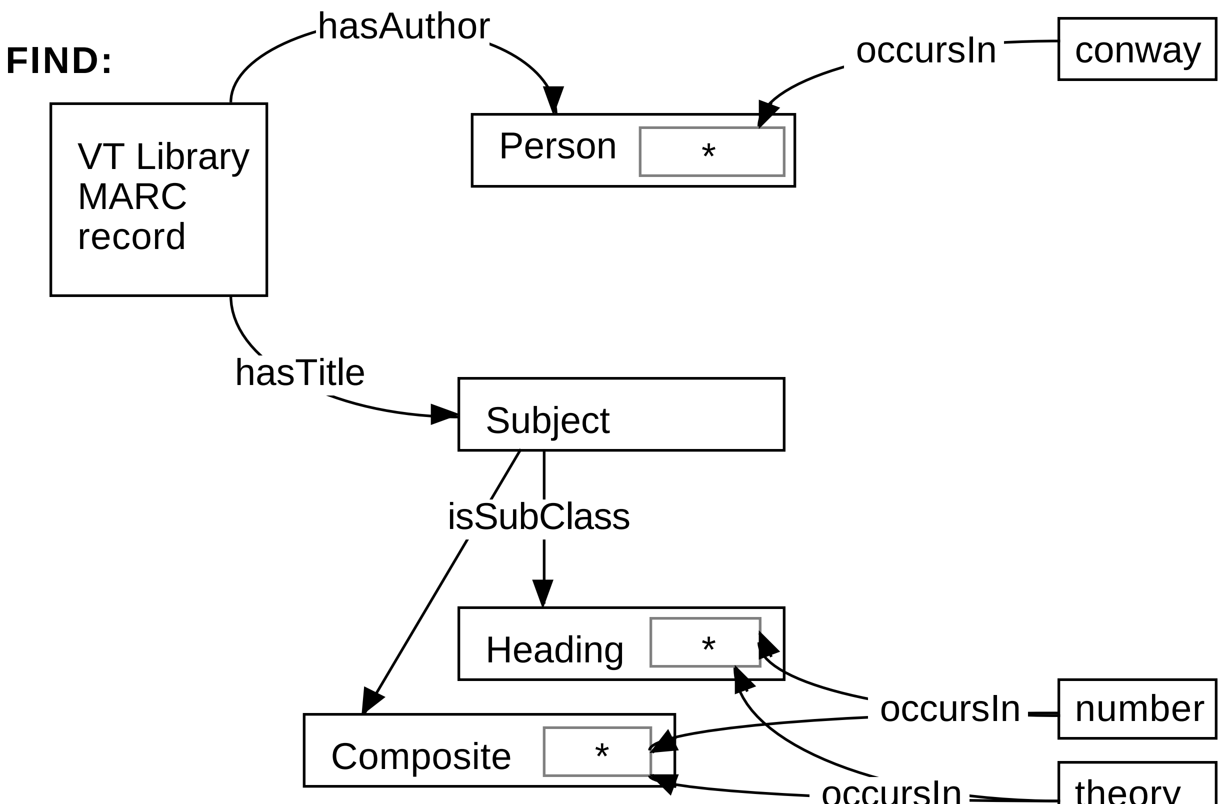
The MARC searcher completes the process by combining Title and hasAuthor sets.



The user enters:

Personal Author: conway
Subject: number theory

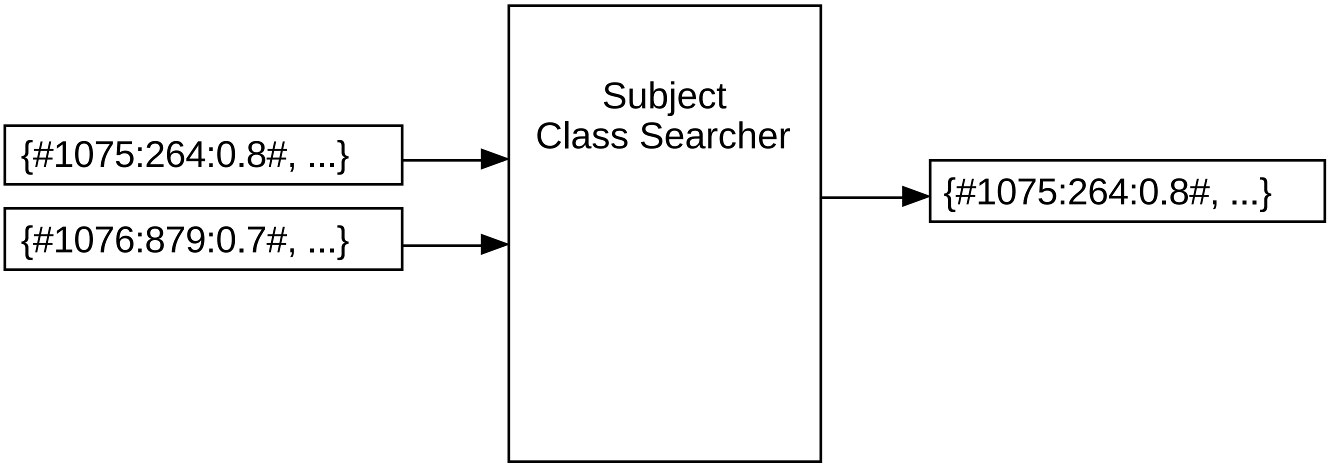
MARIAN translates this to:



The author search proceeds as before.

The Subject searcher has no search methods of its own.

But it does have a method inherited from the Object class searcher for distributing searches among subclasses.



{#1075:264:0.8#, ...}

{#1076:879:0.7#, ...}

Subject
Class Searcher

{#1075:264:0.8#, ...}

This default class method performs a simple merge on the (necessarily disjoint) component sets.

The remainder of the search proceeds as in the last example.

Summary

A system of objects and links is representationally adequate for the digital library domain.

Each class of objects or links can be associated with a searcher that embodies the class semantics.

These searchers all present a common interface defined via creating and manipuating weighted object sets.

The searchers can be built from a few well-chosen operations on weighted object sets.