XML DTDs and other Alternatives: Vocabulary Markup Language (Voc-ML) Project & Friends

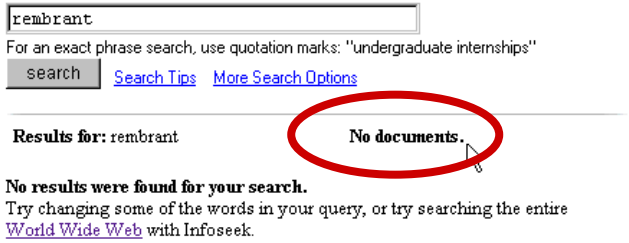
Joseph A. Busch
Director, Solutions Architecture

Outline

- The “real” Semantic Web
- Vocabulary Markup Language (Voc-ML)
  - Namespace registry
  - Schema
  - Services definition
- Voc-ML applications

The problem IS search!

- Data values, NOT just data structures are needed.



Soergel's SemWeb Proposal

- System of integrated access to data on concepts and terminology.
- Bring together variety of sources that exist largely in separate worlds, including dictionaries, thesauri, classification schemes, etc.
- Federated system with multiple collaborators.
- Common interface to all concept & terminology knowledge bases on the Internet.

Dagobert Soergel. "SemWeb: integrated access to distributed ontological resources." (April 1998) Last checked March 29, 2002. http://www.clis.umd.edu/faculty/soergel/soergelsemwebprop.pdf

The Real Semantic Web

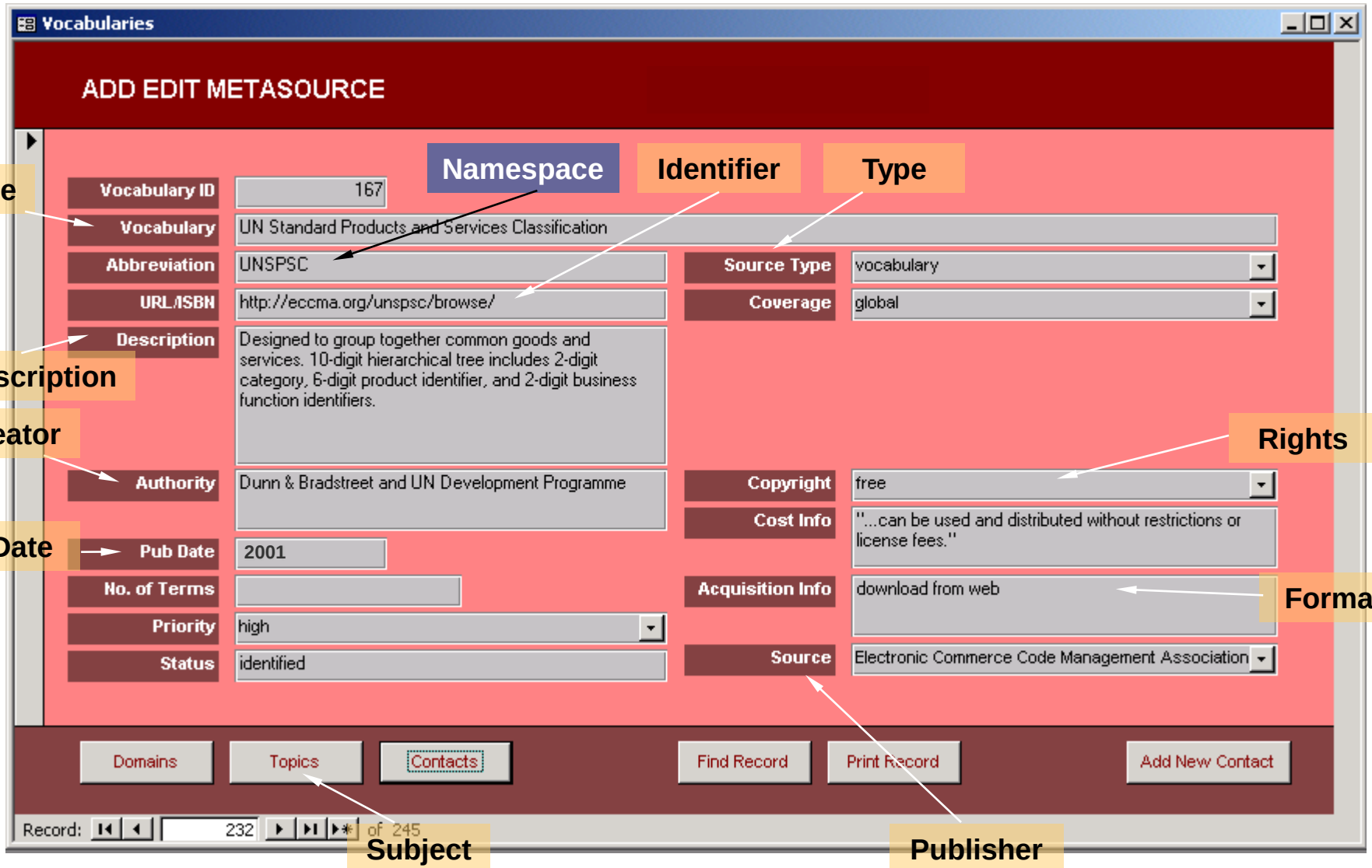
- Namespace for uniquely identifying a semantic scheme & each concept within each scheme.
- Broad template or conceptual schema for holding all types of semantic information & specifying relationships among them.
- Definitions of services for interacting with the System.

Namespace: NKOS Registry

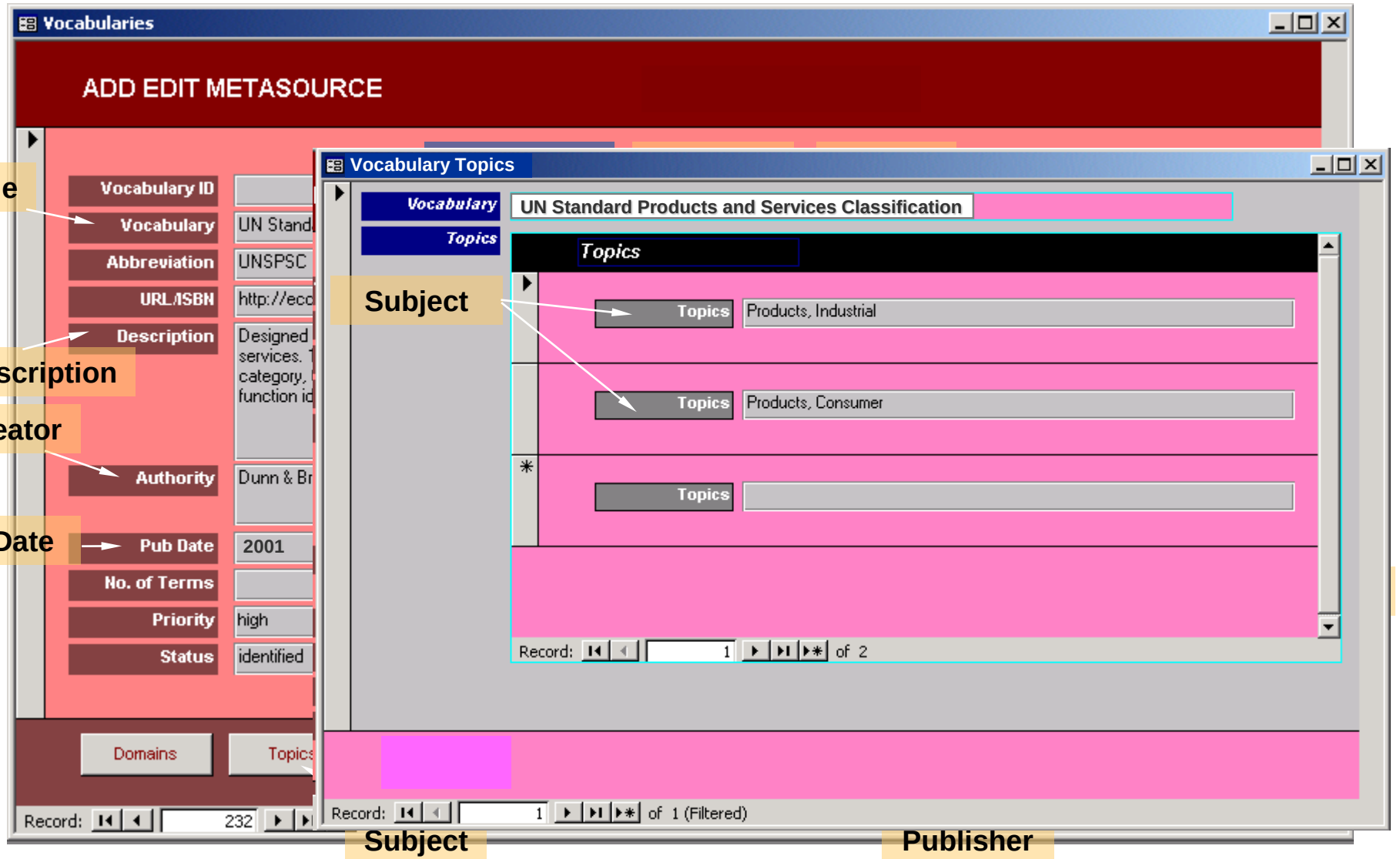
Asset metadata—*The Who, Where and When*	Title, Alternate, Creator, Publisher, Date, Type, Format, Identifier, Language
Subject metadata—*The What and Why*	Subject, Description , Application
Relational metadata—*The Linkages*	Relation
Use metadata—*The How*	Rights, Entity Types, Relationships, Info Given

http://staff.oclc.org/~vizine/NKOS/Thesaurus_Registry_version3_rev.htm

NKOS registry example



NKOS registry example



Schema: Vocabulary Markup Language (Voc-ML)

- XML schema for the Semantic Web.
- Broad template for structured representation of semantic schemes.
  - Z39.19-1993 and ISO 2788
  - Dublin Core metadata
  - Tags and syntax for uniquely identifying each concept
  - Typed relationships (hierarchical, associative, etc.)
- Host agency: Networked Knowledge Organization Systems

http://nkos.slis.kent.edu/VOCML-1.DOC

Voc-ML schema example

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE MetaSource SYSTEM "Voc-ML.dtd">
<MetaSource>
<SVHeader>
    <dc:Title>UN Standard Product and Services Classification</dc:Title>
    <dc:Creator>Dunn & Bradstreet </dc:Creator>
    <dc:Subject>Products, Industrial</dc:Subject>
    <dc:Subject>Products, Consumer</dc:Subject>
    ........
    <UIDprefix>unspsc </UIDprefix>
</SVHeader>
<SVTerm UID="unspsc::501921">
    <label>Snack foods</label>
    <parent UREF="unspsc::5019">
    <child UREF="unspsc::50192101">
    <child UREF="unspsc::50192102">
    <child UREF="unspsc::50192103">
    <child UREF="unspsc::50192104">
</SVTerm>
.....
```

Dublin Core

Unique ID

Typed Relationships

ADL Thesaurus Protocol: XML Elements

<properties>	Overall properties of the thesaurus	
<term>	A term name and its “preferred” status	(format)
<term-description>	Full term description	(format)
<extended>	Any other thesaurus format	(format)
<list>	List of zero or more terms	<response>
<hierarchy>	A hierarchy of terms	<response>
<error>	Error code and description	<response>
<response>	Response from a thesaurus service	

http://www.alexandria.ucsb.edu/~gjanee/thesaurus/specification.html

ADL Thesaurus Protocol: Services

← get-properties

← query? (*operator, text, fuzzy, format*)

```
<query-operators
  equals="true"
  contains-all-words="true"
  contains-any-words="true"
  matches-regexp="false"/>
```

- text = text
- fuzzy = {true|false}
- format = <term>, <term-description>, <extended>

← get-hierarchies? (starting-term, broader-levels, narrower-levels, format)

Service definition example: ← get-properties

http://nkosregistry.org/unspsc/get-properties

```
<response>
    <properties>
        <dc.name>UN Standard Product and Services Classification</dc.name>
        <dc.Creator>Dunn & Bradstreet</dc.Creator>
        <dc.Subject>Products, Industrial</dc.Subject>
        <dc.Subject>Products, Consumer</dc.Subject>
        .....
        <query-operators
          equals="true"
          contains-all-words="true"
          contains-any-words="true"
          matches-regexp="false"/>
        <extended-schema>http://eccma.org/unspsc.dtd</extended-schema>
    </properties>
</response>
```

Service definition example: ← query?

http://nkosregistry.org/unspsc/query?operator=contains-any-words&text=snack+foods&format=term

```
<response>
    <term-description>
        <term>Snack food</term>
        <scope-note>Use this category for food eaten between regular meals.</scope-note>
        <broader>
            <term>Prepared and preserved foods</term>
        </broader>
        <narrower>
            <term>Pretzels</term>
            <term>Corn chips</term>
            <term>Potato chips</term>
            <term>Popcorn</term>
        </narrower>
        <used-for>
            <term preferred="false">Junk foods</term>
        </used-for>
        <related>
            <term>Crackers</term>
        </related>
    </term-description>
<response>
```

Service definition example: ← get-hierarchies?

http://nkosregistry.org/unspsc/get-hierarchies?starting-term=snack%20foods&broader-levels=-2&narrower-levels=1&format=term

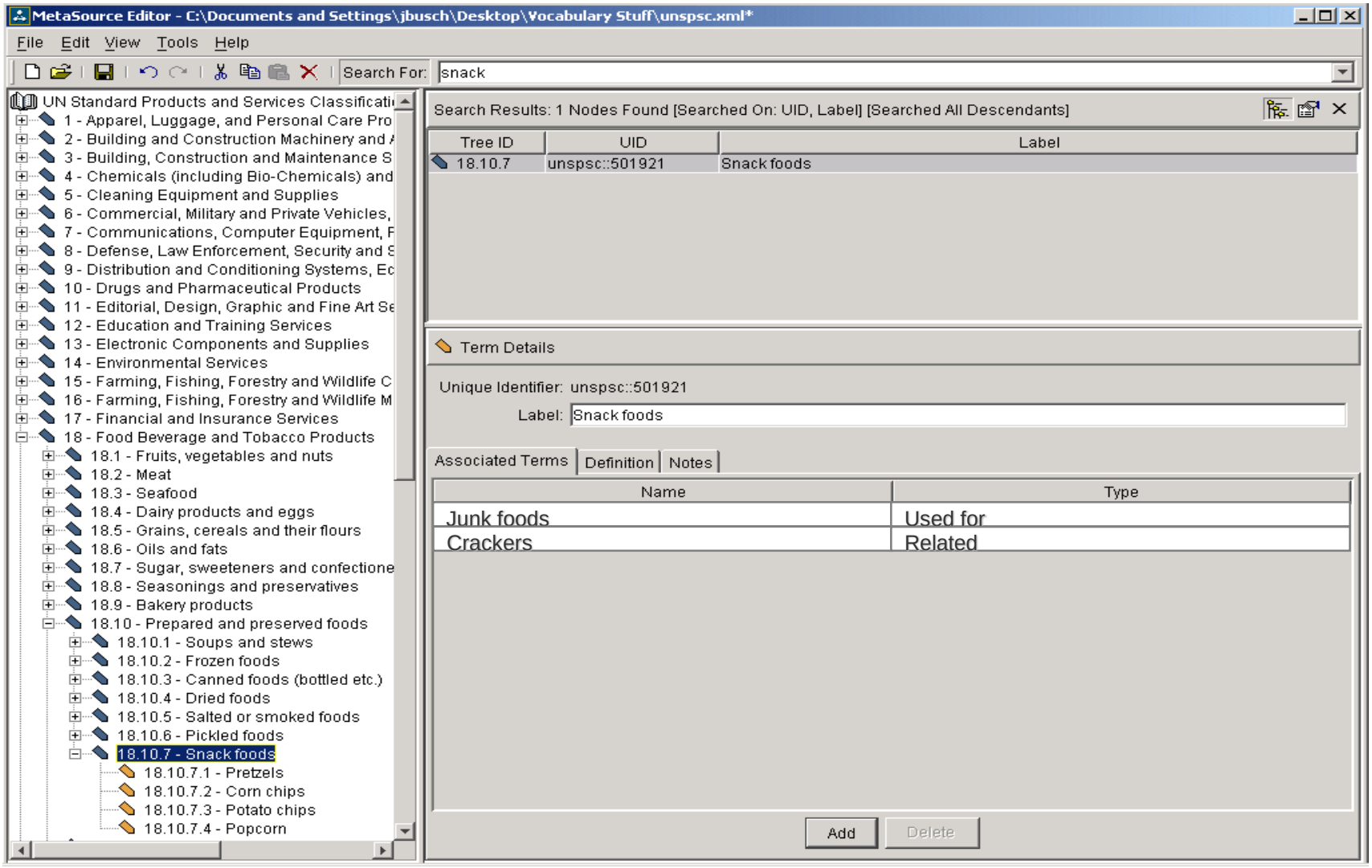
```
<hierarchy direction="broader" maxlevels="-2">
    <node>
    <term>Snack foods</term>
        <node>
        <term>Prepared and preserved foods</term>
            <node>
            <term>Food Beverage and Tobacco Products</term>
            </node>
        </node>
    </node>
</hierarchy>
```

```
<hierarchy direction="narrower" maxlevels="1">
        <node>
        <term>Pretzels</term>
        <term>Corn chips</term>
        <term>Potato chips</term>
        <term>Popcorn</term>
        </node>
</hierarchy>
```

Application: Visual vocabulary editor

Portability.	▪ Voc-ML input/output. ▪ Utilities to convert outline, spreadsheet, directory to Voc-ML
Manage namespace unique ID's (not just a list of labels).	▪ Enforce namespace and ID uniqueness.
Allow polyhierarchy (membership in multiple classes).	▪ Copy & paste term to additional parent.
Allow typed equivalents.	▪ Add/edit equivalents with pre-defined types.
Easy to use.	▪ File manager style hierarchy display. ▪ Drag & drop/cut & paste terms and their children. ▪ Undo. ▪ Right-click functions.

Application: Visual vocabulary editor



Application: Manage product taxonomies

Organize (and reorganize) product classes for diverse purposes	▪ Drag and drop editing; Preserve unique ID within namespace
Allow products to have many aliases	▪ Alternates associated with unique ID
Allow products to exist in more than one class	▪ Polyhierarchy (to allow multiple parents)
Map products across multiple taxonomies	▪ Relationships across different namespaces (e.g. linked parallel hierarchies in different languages
Generate and maintain linkages to associated documentation	▪ Associate metadata labels as well as namespace with unique ID

Application: Manage product taxonomies

Product Information → **Categorization & Metadata** → **Deployment & Re-use**

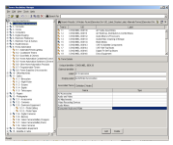
- Windows File Server
- XML/HTML
- RDBMS

Metadata Management

Application Server

- Print
- Intranet Portal
- Enterprise Apps (SAP, Siebel)
- WWW

Polyhierarchy

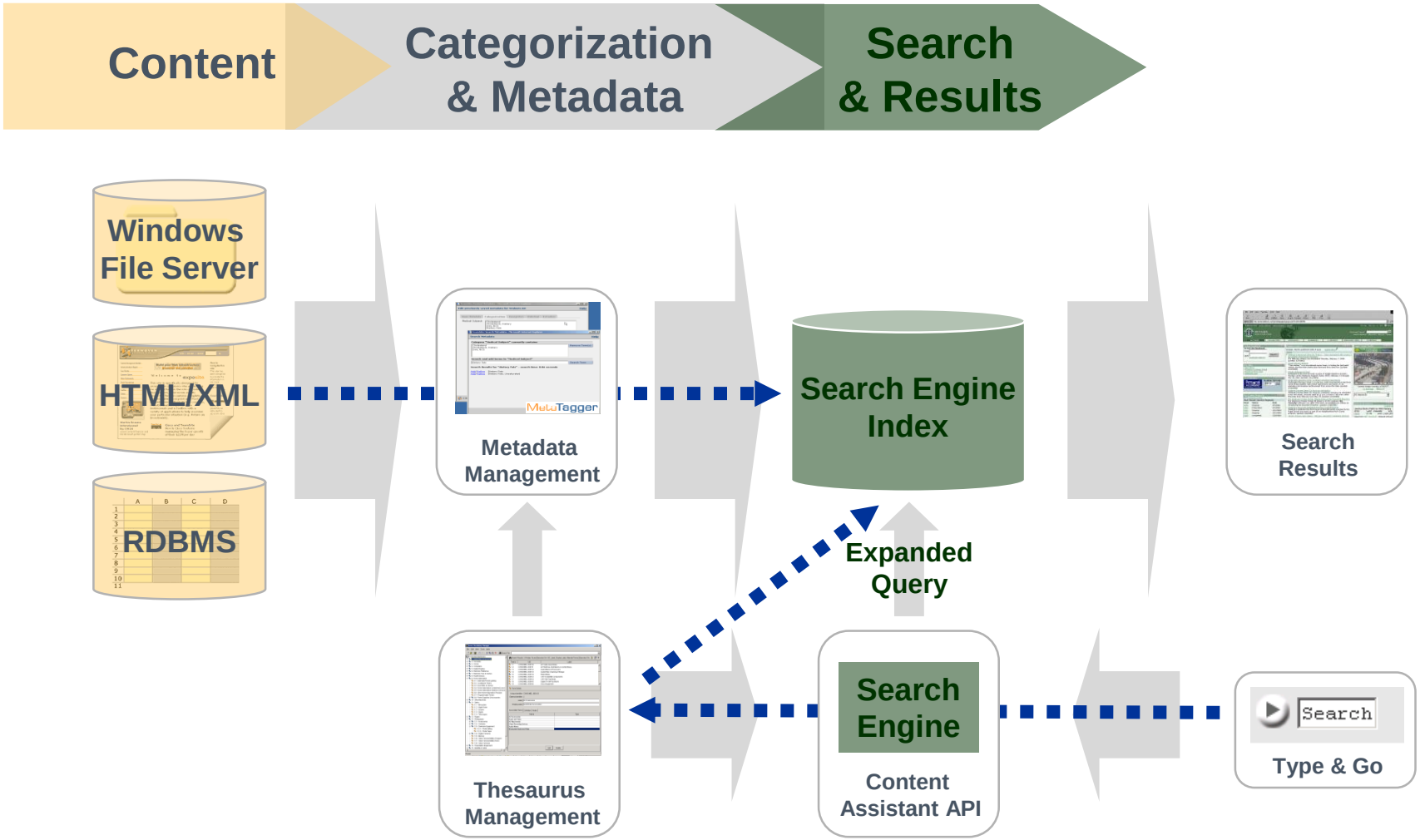


Taxonomy Management

Application: Search query intermediation

Map content to controlled vocabulary (thesaurus, etc.)	▪ Associate metadata labels.
Control exactly what is indexed (metadata not just full text) and when.	▪ Deploy metadata direct to search engine without spidering.
Deploy controlled vocabulary (thesaurus, etc.) to search engine.	▪ Re-direct user queries to thesaurus, and return expanded query based on rules.

Application: Search query intermediation



Contact Information

Joseph A. Busch
Director, Solutions Architecture
Interwoven
803 11th Avenue
Sunnyvale, CA 94089
(408) 220-6974
jbusch@interwoven.com

Visit www.interwoven.com
Enterprise Content Management