

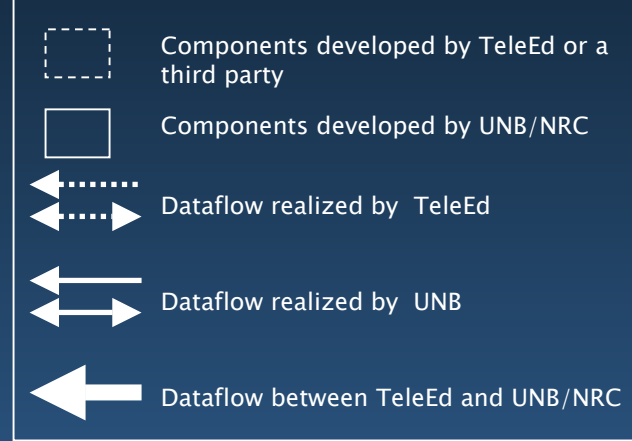
# LOMGen: A Learning Object Metadata Generator Applied to Computer Science Terminology

A. Singh, H. Boley, V.C. Bhavsar  
National Research Council and  
University of New Brunswick

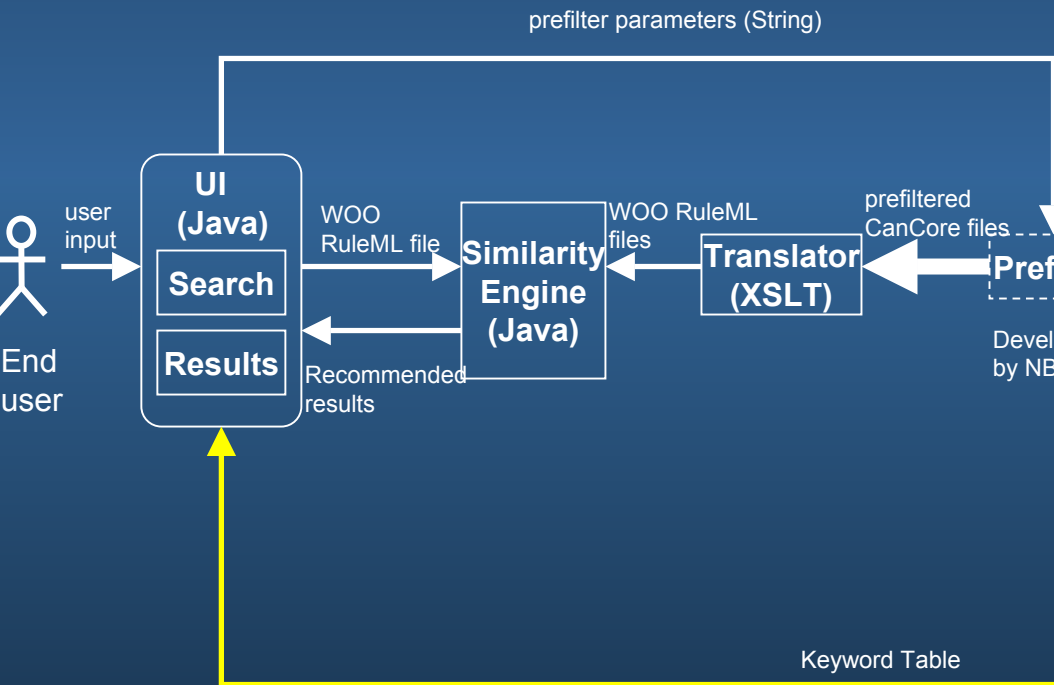
Learning Objects Summit

Fredericton, NB, Canada, March 29-30, 2004

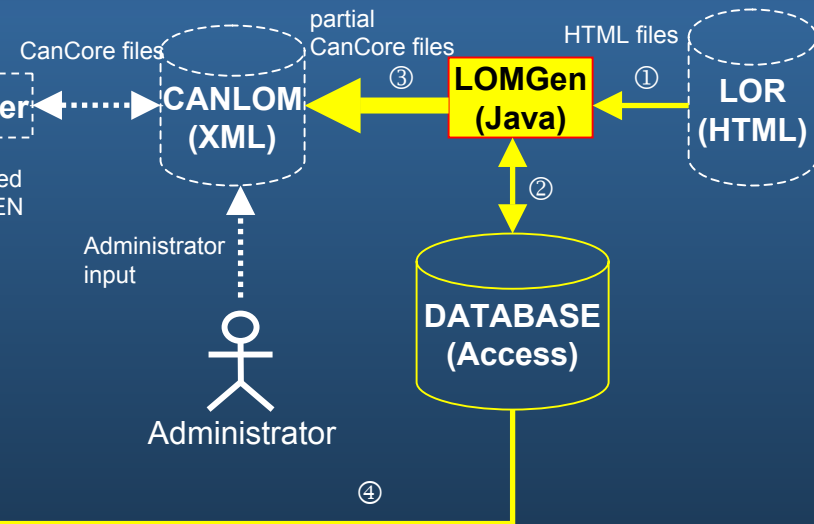
# LOMGen in Context



## Retrieval Components



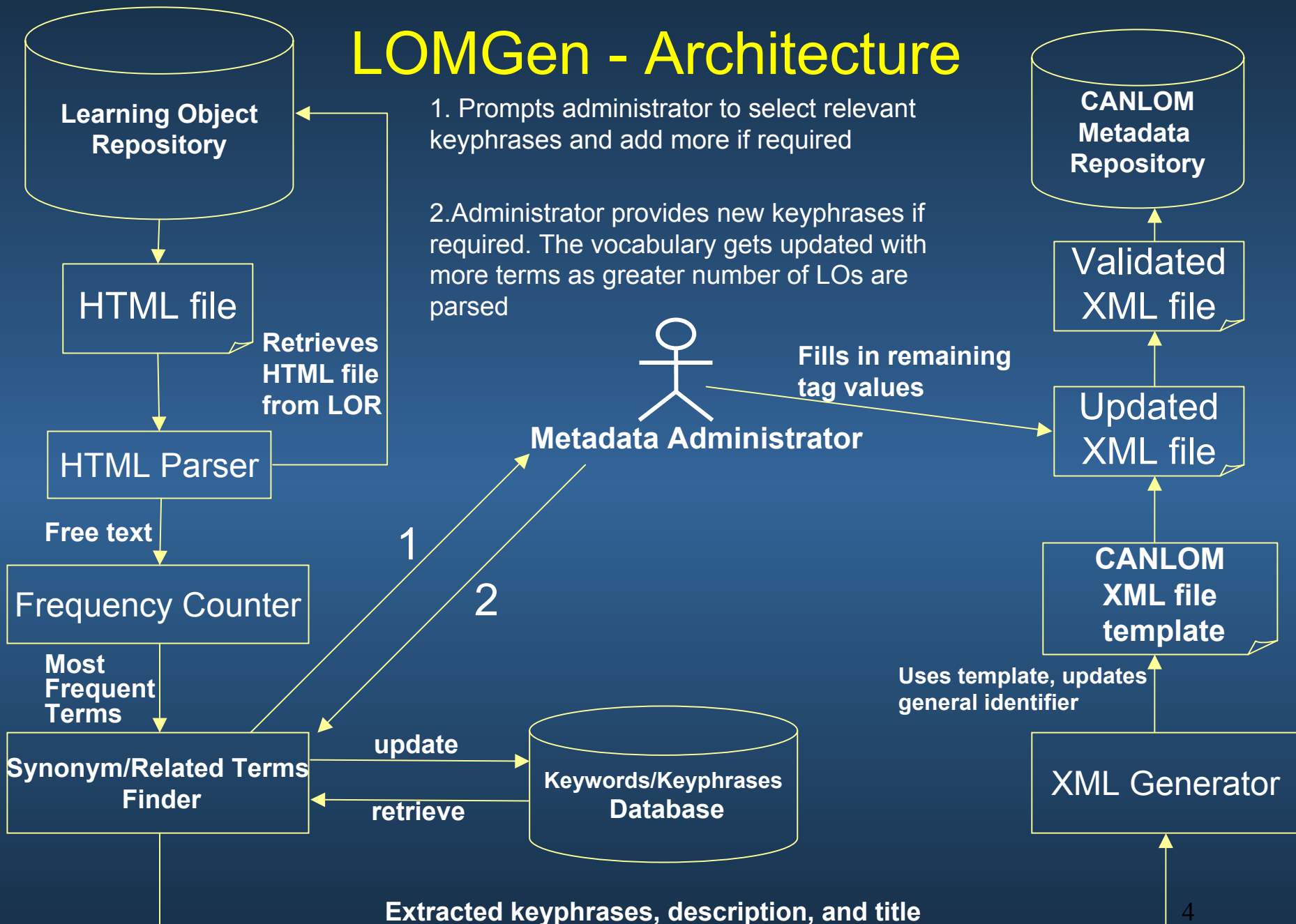
## Indexing Components



# LOMGen - Motivation

- Manually entering metadata is a time consuming process
- Long-term need to automate this process
- Semi-automatic extraction of keywords and keyphrases
- Find related terms and phrases, which may not be present in the LO
- LOMGen achieves these goals with assistance from the administrator

# LOMGen - Architecture



# LOMGen - Components

- HTML File Reader
  - Reads files from the local disk or a URL
- HTML File Parser
  - Parses the HTML files based on the tags and extracts text data
- Frequency Counter
  - Finds the most frequent terms in the text
- Synonym and Related Term Finder
  - Uses a dictionary derived from FOLDOC to generate a set of synonyms and related terms
- LOM Generator
  - Generates LOM for the LO
- Graphical User Interface
  - Allows the metadata administrator to select, and add terms they feel are most important

# HTML File Reader and Parser

- The HTML File Reader retrieves the HTML files (here, LOs) over the Internet or from the local host
- The Parser extracts the `title`, `description` and `keywords` from the meta tags in `head` of the HTML source
- The Parser then removes formatting information from the `body` of the HTML file, passing plain text to the Frequency Counter

# Frequency Counter

- Employs strategies similar to those used by crawlers
- Stop words are eliminated
- Numbers and special characters are ignored
- Stemmer – An iterated Lovins stemmer is used to stem the words to their root. All words with the same root contribute to frequency of the original words

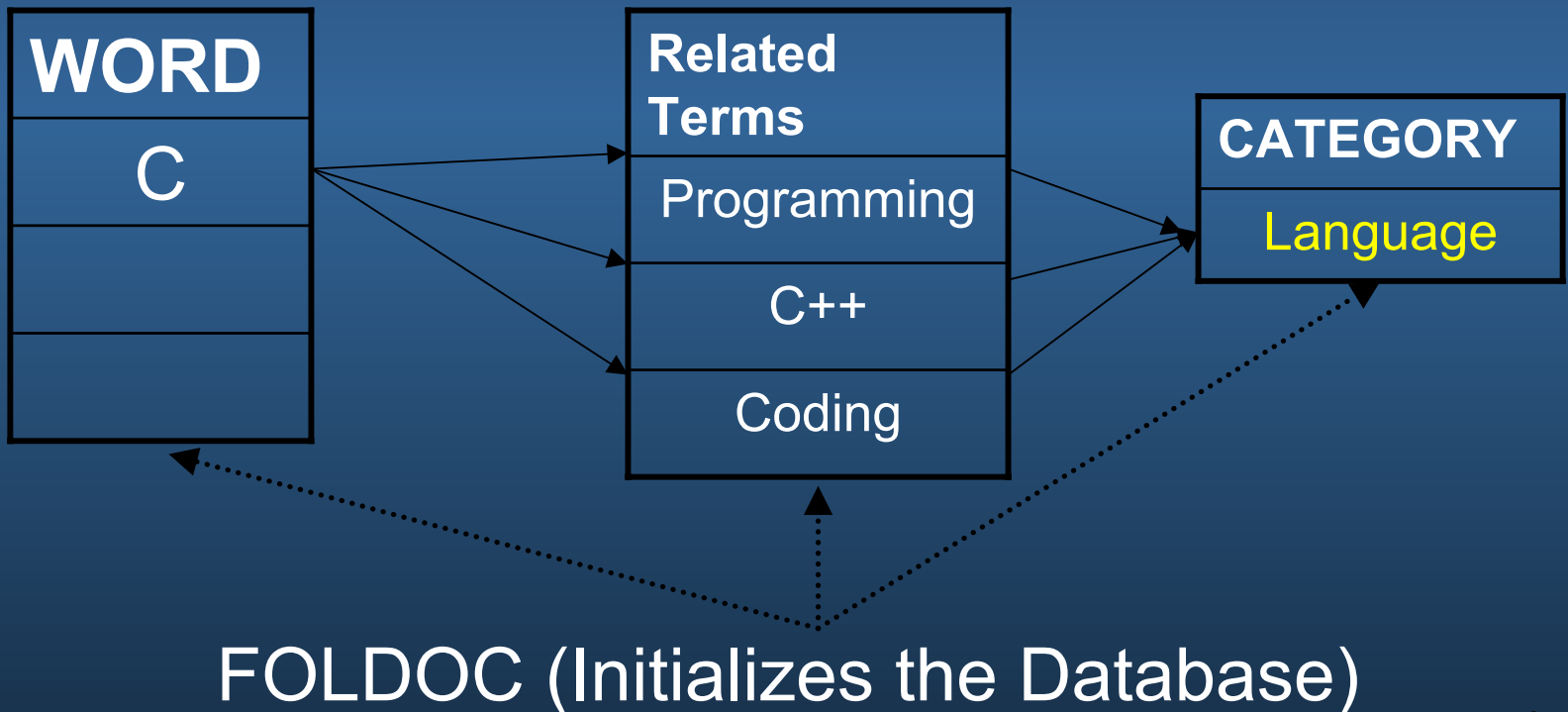
# Related Terms and Domains

- Common words – Terms (from Frequency Counter) and Related terms (as found in Database)
  - Example: RAM, Random Access Memory, SDRAM,... are mapped
- Deep terms – Domain or Concept. These terms map a group of terms to a class or category
  - Example: Memory



# Dictionary Structure in the Database

- Free Online Dictionary of computing is used to initialize the database – FOLDOC does not have a well defined structure
- The database tries to achieve the structure shown below



# Keyword and Keyphrase Identification

## Strategies Employed:

- Frequency
  - Term frequency (one at a time)
  - Loss of syntactic information does not affect LOMGen output much for technical terms
- Keyphrases
  - Keyphrase matching using Database
- These strategies have proven useful in practice

# Administrator GUI

LOMGen Learning Object Metadata Generator

Please select the relevant keyphrases

KEYPHRASE	ADD SYNONYMS (optional)
<input checked="" type="checkbox"/> OWL	Web Ontology Language
<input checked="" type="checkbox"/> semantic web	
<input checked="" type="checkbox"/> RDF	
<input type="checkbox"/> web site	
<input checked="" type="checkbox"/> Internet	
<input checked="" type="checkbox"/> World-Wide Web	
<input checked="" type="checkbox"/> resource description framework	
<input type="checkbox"/> W3C	
<input type="checkbox"/> meta-data	
<input type="checkbox"/> computer science	
<input type="checkbox"/> informatics	
<input type="checkbox"/> web	
<input type="checkbox"/> Donald Knuth	
<input type="checkbox"/> literate programming	

Add more terms to describe the Learning Object (Optional)

Delete  Add

Select a domain term to describe the Learning Object: World-Wide Web

OK

- The keywords and keyphrases extracted from the HTML file are presented to the metadata administrator
- The administrator can select and add any additional terms they find important
- The administrator is also prompted to select a category for the LO

# The Database

- Database stores any new phrases or words selectively added by administrator
  - Serves as a rudimentary learning loop
- Subsequently, on encountering similar LOs, LOMGen provides better choices to the administrator in the GUI
- The newly added words/phrases also help to identify more relevant phrases in the text

# LOM Generation

- LOMGen generates the values for tags in the **General** and **Classification** categories
- A CanLOM compliant XML file is generated and posted to the KnowledgeAgora metadata repository
  - Contains most of the relevant fields
  - Categories like Lifecycle, meta-Metadata have to be filled in manually

# Snapshots of HTML and generated XML

```
<html>
<head>
  <title>Bayesian Filter - Filtering over 98% of Spam!</title>
  <meta name="description" content="This paper describes how through using a bay
  <meta name="keywords" content="Bayesian filter, bayesian filtering, Bayesian f
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" href=" ../DHTMLmenu/style/wpstyles.css" type="text/css">
</head>
<body>
<table width="95%" border="0" cellspacing="0" cellpadding="20"> <tr> <td rowspan="2" b
<a href="http://www.gfi.com" target="top"> </td><td rowspan="2" valign="top" width="85%" class="BodyText"><b><
Bayesian filtering is the most effective anti-spam technology</span><hr> </b><p>This
white paper describes how Bayesian mathematics can be applied to the spam problem,
resulting in an adaptive, &#145;statistical intelligence&#146; technique that is
much harder to circumvent by spammers. It also explains why the Bayesian approach
is the best way to tackle spam once and for all, as it overcomes the obstacles faced b
as effectively as needed if not combined with a Bayesian filter.</p><hr size="1">
<ul> <li><a href="#howitworks" class="H1">How the Bayesian spam filter works</a></li><
a tailor-made Bayesian word database</a></li><li><a href="#findingspam" class="H1">Fin
spam based on the Bayesian filter</a></li><li><a href="#betterthankeycheck" class="H1">
Bayesian filtering is better than keyword checking in detecting spam</a></li><li><a href
Bayesian method takes the whole message into account</a></li><li><a href="#point2" class="H1">
Bayesian filter is constantly self-adapting</a></li><li><a href="#point3" class="H1">T
Bayesian technique is sensitive to the user</a></li><li><a href="#point4" class="H1">T
Bayesian method is multi-lingual and international</a></li><li><a href="#point5" class="H1">
Bayesian filter is hard to trick as opposed to a keyword filter</a></li></ul></li><li>
filters or updated keyword lists?</a></li><li><a href="#aboutmes" class="H1">About
GFI MailEssentials</a></li><li><a href="#GFI" class="H1">About GFI</a></li></ul><hr si
<p>Spam is an ever-increasing problem. The number of spam mails is increasing
daily - in June 2003, studies showed that over 50% of all email is spam. Added
to this, spammers are becoming more sophisticated and are constantly managing
to outsmart 'static' methods of fighting spam. </p><p>The techniques currently
used by anti-spam software are static, meaning that it is fairly easy to evade
by tweaking the message a little. To do this, spammers simply examine the latest
anti-spam techniques and find ways how to dodge them. </p><p>To effectively combat
spam, an adaptive new technique is needed. This method must be familiar with spammers'
tactics as they change over time. It must also be able to adapt to the particular
```

Learning Object HTML

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!om xmlns="http://itsc.ieee.org/xsd/LOMv1p0" xmlns:xsi="http://www.w3.org/2
  xsi:schemaLocation="http://itsc.ieee.org/xsd/LOMv1p0 http://adlib.athabasca
- <general>
  - <identifier>
    <catalog>URI</catalog>
    <entry>http://www.gfi.com/mes/wpbayesian.htm</entry>
  </identifier>
- <title>
  <string language="en">Bayesian Filter - Filtering over 98% of Spam!</string>
</title>
  <language>en</language>
- <description>
  <string language="en">This paper describes how through using a bayesian fil
  network administrators can achieve a spam detection rate of over 98%. I
  that most events are dependent and that the probability of an event occu
  previous occurrences of that event.</string>
</description>
- <keyword>
  <string language="x-none">Bayesian filter</string>
</keyword>
- <keyword>
  <string language="x-none">electronic mail</string>
</keyword>
- <keyword>
  <string language="x-none">spam</string>
</keyword>
</general>
...
```

Generated XML

# LOMGen - Summary

LOMGen extracts keywords and keyphrases from a HTML document

A dictionary of “surface terms” and “deep terms” stored in a database

LOMGen provides the metadata administrator with a user interface

LOMGen provides the AgentMatcher UI with the list of keyphrases and keywords

Makes similarity computation more accurate

Automates metadata (XML) posting to LOM repository

# Conclusion

- LOMGen was able to get metadata for categories **General** and **Classification**
- Difficult to have a fully automated process for metadata extraction
  - LOs in HTML do not always follow guidelines
  - Some information required for metadata is not available in LOs, hence cannot be extracted
- LOMGen is a step towards automation, and a tool of this kind could be standard in future LO environments
- Demo's next, as time permits