Toward a Knowledge-Based Solution for Information Discovery in Complex and Dynamic Domains

Eloise Currie and Mary Parmelee
SAS Institute, Cary NC
About SAS: The Power to Know®

- **SAS**: The Market Leader in Business Intelligence Software
- Founded: 1976
- World’s Largest Privately Held Software Company
- Worldwide Offices: 269
- Worldwide Employees: 9,238
- Worldwide Revenue 2003: $1.34 billion
- Reinvestment in R&D 2003: 26%

**SAS** Solutions:
- Used at more than 40,000 sites
- Used by 96 of the top 100 of the 2003 Fortune Global 500
The Problem with Information

- From a Global Perspective
  - As the volume of online information grows, information retrieval (IR) has become a major challenge.

- How much is it changing?
  - In 1995, over 90% of corporate documents were in paper form. By 2005, less than 30% will remain in paper form.
  - Within the next 3 years, the world will produce as much data as has been produced since **THE DAWN OF TIME!**
The Problem with Information

From a SAS Perspective

1. Address Customer Pains:
   - SAS customers are satisfied with the quality of our documentation, but they have difficulty locating information.

2. Manage Growth and Complexity
   - How much is it changing?
   - SAS product growth: 1 > 170+
   - SAS user growth: 4 million users worldwide
Toward a Solution: Preliminary Research

**Observation**
- Online information and queries are communicated via natural language, which has two main properties:
  - semantics gives meaning in **context**
  - syntactics give structure and order
- Yet, most IR systems match only syntactics

**Objective**
- Create an IR system that leverages the semantics of natural language.

**Investigation**
- Emerging technologies, initiatives and standards: Semantic Web, Ontologies, RDF
- Consulted IR Experts (UNC-Chapel Hill)
- Tools: Protégé, Jena Toolkit
Toward a Solution: Progress to Date

2001: Proof of concept project

- **Tiny domain** (subject area): two pages of documentation
- **Rudimentary UI**
- **Deliverable**: development methodology and repeatable process

  -Browsable directory tree
  -Resources associated with a node in the tree
Toward a Solution: Progress to Date

2002: Pilot Project

- **Larger domain**: a complete product user’s guide
- **More robust UI**
- **Deliverable**: A functioning mini system

Additional features:
- Added search functionality
- Search results pane
Toward a Solution: Progress to Date

February 2004: Prototype Project

- **Large domain**: Includes several products
- **Intelligence layer**: Enables advanced search and reasoning capability
- **Advanced UI**: Delivers information in context while minimizing the complexity surfaced to the user
- **Deliverable**: Fully Functional Prototype
Knowledge Base System Architecture

Content Development → Intelligence Layer → Contextual Information Delivery

- **Create Resource Set**
  - Authored Content
  - Resource Repository
  - Static Resource Assembly
  - Hardcopy
  - Compact Disc
  - HTML

- **Classify Resources**
  - Knowledge Model (Ontologies)

- **Process Knowledge Model**
  - Knowledge Repository

- **Retrieves Resources**
  - Unstructured Content
  - Process Controller
  - Full Text Engine
  - Inference Engine
  - Queries KR and Retrieves Metadata

- **Deliver**
  - KB Portal
  - KB Portal
  - Happier Client!
System Development Process

We begin with a document collection (a “resource set”).

1. Use SAS® Text Miner to create a hierarchy of resource clusters
2. Use a custom Protégé plugin to generate a Domain ontology that categorizes resources based on content
3. Use a custom Protégé plugin to extract resource information and generate a Resource Manager ontology
4. Merge Domain and Resource ontologies into a Master ontology
5. Use a custom Protégé plugin to map Resource instances to Domain instance slots
6. Use Protégé to develop the merged ontology into a production Master ontology
7. Use a custom Protégé plugin to reverse map Domain Instances to Resource Instance Slots
### Step 1: Use SAS® Text Miner to Create a Hierarchy of Resource Clusters

<table>
<thead>
<tr>
<th>Steps</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing</td>
<td>Create a SAS data set from the document collection.</td>
</tr>
<tr>
<td>Text parsing</td>
<td>Generate quantitative representation of the content</td>
</tr>
<tr>
<td>Transformation</td>
<td>Consolidate quantitative representation</td>
</tr>
<tr>
<td>Document analysis</td>
<td>Cluster documents by concept</td>
</tr>
</tbody>
</table>
Step 1: Hierarchical Output

Resource Set

- "window" 1402 docs
- "data" 326 docs
- "display" 1076 docs
- "property" 159 docs
- "method" 292 docs
- "permission" 329 docs
- "metadata" 1402 docs
- "define" 821 docs
- "table" 599 docs
- "option" 477 docs
- "interface" 529 docs
- "SAS 9.1" 200 docs

"Hybrid" cluster, built up from root clusters

"Root" cluster
Step 2: Use a Custom Protégé Plugin to Generate a Domain Ontology in Protégé

- Saves Text Miner hierarchy as a Protégé ontology
- Preserves URI’s of documents in a Resource ID slot

Specify input and output locations

Specify ontology name
Step 2: Domain Ontology in Protégé

- Each direct instance is a Text Miner document cluster
- Generic instance and class names will be edited

**Classes**

- Classes have generic names generated by Text Miner

**Direct Instances**

- Direct Instances have generic names generated by Protégé
- Significant cluster terms identified by Text Miner
- Resource IDs of those resources assigned to this direct instance of the Domain ontology
Step 3: Use a Custom Protégé Plugin to Generate a Resource Manager Ontology in Protégé

- Extracts resource information
- Preserves URI’s of documents in a Resource ID slot
- Saves resource information in a Protégé ontology

Specify path and name of Resource Manager ontology
Step 3: Resource Manager Ontology

Each Direct Instance is a document in the Resource class

Instances of the Resource class
Step 4: Merge the Domain and Resource Ontologies into a Master Ontology
Step 5: Use a Custom Protégé Plugin to Map Resource Instances to Domain Instance Slots

- Maps by Resource ID
- Populates resource instance slot

Select “Map Resources to Categories”

Specify path of Master Ontology
Step 5: Resource Instances Mapped to Domain Instance Slots

Domain class

Resource Manager instances are mapped to Domain instance slots

Resource Manager Description slot facilitates ontology refinement

Domain instance
Step 6: Refine and Expand the Merged Ontology into a Production Master Ontology

- Add Related Terms to enable fuzzy matching of misspellings, synonymous phrases and alternative word forms

- Refine the Domain Hierarchy
  - Add the SAS Offerings Model
    - SAS products and solutions
  - Develop full ontology from Text Miner hierarchy
  - Add metaclasses to assign URIs at the class level
Step 6: Production Master Ontology

SAS Offerings Model

Domain Ontology Classes with edited, meaningful names

Resource Manager ontology classes

Classes with edited, meaningful names

Tool: an application used primarily to create

InternalLabel
tools

Label
Step 6: Add Metaclasses

Add Metaclasses to assign URIs at the class level
Step 7: Use a Custom Protégé Plugin to Map Domain Instances to Resource Instance Slots

Reverse mapping of Resource to Domain instance (Step 5)

Select “Map Categories to Resources”

Specify path of existing Master Ontology
Step 7: Domain Instances are Mapped to Resource Instance Slots

Resource Manager
Resource Class

Resource Manager instance

Domain instances to which this Resource Manager instance is assigned
Ontologies Define the Intelligence Layer

Knowledge Base Resources optimized for:
Browse          Search          Preferences

Intelligence Layer

Teradata          MySQL          DB2
Databases          Multimedia      Distributed Systems

XML         PDF         Word         HTML
Unstructured Resources
Knowledge Base Prototype

Delivers information in context using

• Browsable categories
• Categorized search results
• Hover text descriptions
• Category bread crumb trails
• Category and full text search
• Fuzzy matching
Browse View: Browsable Directory and Hover Text Contextual Cues

Domain ontology displayed as a category hierarchy

Includes a high-level overview of SAS Marketing Automation

Hover text provides a contextual category description

Resource instances mapped to current category
Search Results View: Search Expansion
Fuzzy Match Synonymous Phrase

enter phrase “grouping reports” and push the search button

system matches to phrase equivalency “portfolios”

breadcrumb trail to the root category provides context

Optional full text search filtered by the current category
# Search Results View: Contextual Cues

Grouped Results of Full Text Search

## Reports > Portfolios

**Running Portfolios**

All the reports stored in a report portfolio can be run together. You might want to do this to run a number of reports several times. For example, you might have a portfolio of marketing reports generated on a weekly basis.

## Word Processing > Save

**Saving Reports**

A report can be saved on the system for later use. Reports are usually saved in portfolios. These are used to group together several reports in a logical way. For example, you might have a portfolio of monthly marketing reports.

## Report > Campaign Report

**Saving Reports**

A report can be saved on the system for later use. Reports are usually saved in portfolios. These are used to group together several reports in a logical way. For example, you might have a portfolio of monthly marketing reports.

**Renaming and Deleting Reports**

Portfolios and reports which you own can both be renamed and deleted. Follow the steps below. Click the Open icon. The Report and Portfolio Management dialog will appear.

**Retrieving and Editing Reports**

Saved reports can be opened for viewing, copying to the clipboard, and printing. You might also want to retrieve saved reports to run them against an updated database.
Our Vision for an Integrated Solution: The Value of XML-Based Modular Content

- **Why XML?**
  - Accepted standard—already used by a critical mass
  - Facilitates platform independent interoperability
  - Provides a content development framework that supports modular writing

- **Why Modular Writing?**
  - Reusability
    - Controls work redundancy
    - Reduces semantic heterogeneity
      - using the same terminology to mean different things
      - using different terminology to mean the same thing
  - Facilitates content classification: “about” one thing
  - Enables advanced information retrieval and delivery techniques
    - dynamic assembly of complex resources that are relevant to a user’s current context
Knowledge Base System Architecture

Content Development → Intelligence Layer → Dynamic Information Delivery

Create Resource Set (Domain)
- Authored Content
- Resource Repository
- Static Resource Assembly
- Hardcopy
- Compact Disc
- HTML

Classify Content Modules
- Knowledge Model (Ontologies)
- Unstructured Content

Instantiate Model in Metadata Language
- Process Controller
- Relay Queries and Retrieves Results
- Full Text Engine
- Inference Engine
- Queries KR and Retrieves Metadata

Retrieves Resources

Deliver
- KB Portal
- Deliver Resources
- Client Queries

Client Queries

Happier Client!
One Vision for an Integrated Solution

Content Development → Intelligence Layer → Dynamic Information Delivery

Create Resource Set (Domain)
- Authoring Modular Content
- XML Content Module Repository
- Static Deliverable Assembly
- Hardcopy
- Compact Disc
- Integrated Help

Classify Content Modules
- Knowledge Model (Ontologies)
- Instantiate Model in Metadata Language
- Knowledge Repository

Retrieve modules
- Unstructured Content
- Document Assembler Assembles and Formats Documents
- Delivers Instructions
- Inference Engine
- Queries KR and Retrieves Metadata

Deliver
- KB Portal
- Delivers Complex Resources
- Client Queries

Satisfied Client!

7th International Protégé Conference: July 7, 2004
Questions/Comments?

Thanks to Contributors: Dee Stribling and Chris Goolsby