Enabling Task Centered Knowledge Support through Semantic Markup

Rob Jasper
Mike Uschold
Boeing Phantom Works
Conclusions

✅ Information retrieval should explicitly recognize the overarching task a user is performing
  - User’s are seldom just searching for an artifact (e.g., document)
  - Current IR system interaction fail to recognize which task a user is performing
  - There is important information in the user’s broader goals

✅ Navigation, integration, transformation, and presentation are just as important as retrieval
Answers to your questions

✔ Yes
✔ No
✔ It’s only a prototype
✔ Ask Stefan
✔ I don’t know
✔ I’ll answer that offline
Web Interaction Perspective

✓ Simple Taxonomy of Web Interaction
  - Without any purpose (pure surfing)
  - With purpose (some notion of task)
Focus of Task Centered Knowledge Support

<table>
<thead>
<tr>
<th>Static &amp; Dynamic HTML</th>
<th>Not cost effective</th>
<th>Requires Omniscience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static &amp; Dynamic HTML</td>
<td>Task Centered Knowledge Support</td>
<td>Requires Omniscience</td>
</tr>
</tbody>
</table>
Example Problems (Specific)

Q: Who was the 19th president of the U.S.
A: Rutherford Birchard Hayes
http://www.ipl.org/ref/POTUS/rbhayes.html
Example Problems (General)

Q: Is Rutherford Birchard Hayes related to Pat Hayes?
So, what’s the difference?

Why is the first question specific and second question General?

- Content providers anticipated that people would want to know about Rutherford Birchard Hayes
- Content providers didn’t anticipate that people would want to know how AI researchers are related to U.S. Presidents
Q: Can I replace the interior cabin carpet (part number B-732181289A123) in my Boeing 747 standard commercial grade 100% nylon indoor/outdoor carpet.

Can I anticipate questions like this?
Service Engineering Overview

Operator (Airline)

Service Engineer

Field Service Rep

BOECOM

Operato (Airline)
Service Engineering Tasks

Specific Tasks  Supportable Tasks  General Tasks
Supportable Task — Justification

- High degree of problem commonality
  - Airlines in similar businesses
  - Physical properties of planes
- Ability to leverage work done by others
Task Centered Knowledge Support (Prototype)

Problem Statement

Problem Characterization

Links to Resources

Generic Resources
One Approach

Rely on human experts to locate sources to answers to common questions:

Downside:
1) Labor intensive
2) Not enough context to perform well
3) No integration of content
Prototype Architecture

- XSL
- Template Engine
- SiLRI
- XML Templates
- RDF/FLogic Repository
- Boeing Thesaurus

BOECOM
Sample from Flogic KB

F1:FISR

[Title => "Engine starting difficulty;
URL => "http://...";
Model => "747";
Related_Concepts =>> {starting, jet engine}].

FORALL C, U, L, M
fisr_directly_related_to(C, U, L, M) <-
(EXISTS F F:FISR[Related_Concepts =>> C]
 and fisr_html_ref(F, U, L, M)).

FORALL C, U, L, M
fisr_related_to(C, U, L, M) <-
fisr_directly_related_to(C, U, L, M).

FORALL C, U, L, M
fisr_related_to(C, U, L, M) <-
(EXISTS C0 fisr_directly_related_to(C0, U, L, M)
 and near(1,C0,C)
 and (EXISTS F fisr_html_ref(F, U, L, M))).

ignition:Concept
[name =>> "ignition";
rt =>> {starting,
ignition_time,
ignition_systems,
ignition_temperature}].

engine_starters:Concept
[name =>> "Engine starters";
rt =>> {starting,
jet_engines}].

FORALL X,Y
near(1,X,Y) <- X[related_to =>> Y] or
X[broader_term =>> Y] or
X[narrower_term =>> Y].
XML Template Language

<?xml version="1.0"?>
<Resources>
  <SUBST>
    <HTML-TEMPLATE>
      <FISRS>
        <SUBST QUERY="FORALL URL,Label,Model <- f isr_related_to(@Cnpt, URL, Label, Model).">"
        <HTML-TEMPLATE>
          <FISR>
            <URL><![CDATA[/@URL]]></URL>
            <Model>@Model</Model>
            <Title>@Label</Title>
          </FISR>
        </HTML-TEMPLATE>
      </FISRS>
      • Return all FISRS.....
      • Use Query bindings and substitute for @variables
      • Take transitive closure of the substitutions.
    </SUBST>
  </HTML-TEMPLATE>
</SUBST>
</Resources>
Task Characterization
Goals

- Automatically characterize task based on incoming msg (e.g., NLP, statistical)
- Further refine task breakdown (e.g., KADS)
- Finer grained mapping tasks to resources
- Pull content (not just links)
- Integrate content across resources based on tasks performed
Proposed Architecture

XML

XML

XML

XML

HTML

XML

Relational

Documents

Task Characterization

Integration Engine

Query Services

SOAP / XML RPC

Metadata Repository
Conclusions

✓ Information retrieval should explicitly recognize the overarching task a user is performing
  – User’s are seldom just searching for an artifact (e.g., document)
  – Current IR systems fail to recognize which task a user is performing
  – There is important information in the user’s broader goals

✓ Navigation, integration, transformation, and presentation are just as important as retrieval