

Ontology Based Interfaces to Access a Library of Virtual Hyperbooks

G. Falquet, C.-L. Mottaz Jiang, J.-C. Ziswiler

Information System Interfaces
CUI – University of Geneva

ECDL'04, September 13-15, 2004, Bath (UK)

Outline

Context

Virtual hyperbook model

Interface model

Hyperbook integration

Implementation

Context

Virtual documents and books

[Crampes] [Garlatti] [De Bra] [Brusilowsky]

- Informational fragments and domain knowledge
- Mechanisms to produce *real* interface documents adapted to user's needs
 - selection
 - assembly
 - presentation
- Restriction: scientific or pedagogical (≠ fiction) monographs

Context

Library of virtual books

- Set of virtual books

New access methods / reading possibilities

- Break the monolithic aspect of the book
- Extend a book with the contents of others
- Generate new books
- Compare, synthesize

G. Falquet, C.-L. Mottaz Jiang, J.-C. Ziswiler ECDDL'04 4

Virtual Hyperbook Model

domain ontology
typed links
information fragments
hyperbook
interface specification
user interface
generated hyperdocuments

G. Falquet, C.-L. Mottaz Jiang, J.-C. Ziswiler ECDDL'04 5

Structure of a hyperbook

Ontology
Algorithm
Graph
Tree
is-a
property
example
example
example
Fragments
Theorem. xxxxxx
yyyyy zzzz yy x
dijgo sds ssssk
sslk

G. Falquet, C.-L. Mottaz Jiang, J.-C. Ziswiler ECDDL'04 6

Reading interface

Interface specification: set of parameterized node schemas
 Interface: set of node instances

Ontology

Fragments

node schema N
select
content
links

node instances
N[r]
N[q]
...
N[p]

selection

assembly

G. Falquet, C.-L. Mottaz Jiang, J.-C. Ziswiler

ECDL'04

7

A node schema

```
// All the fragments linked to concept C
node connected_fragments[C]
{.....
  <f> L.type , F.title , F.content </f>
  .....
}
from Concept C -( Link L )- Fragment F
```

Content specification
hyperbook objets
in xml elements
links

Selection based on path expressions
on the ontology-fragment graph

G. Falquet, C.-L. Mottaz Jiang, J.-C. Ziswiler

ECDL'04

8

Specifying link inference

Ontology

Graph

Tree

is-a

example

example

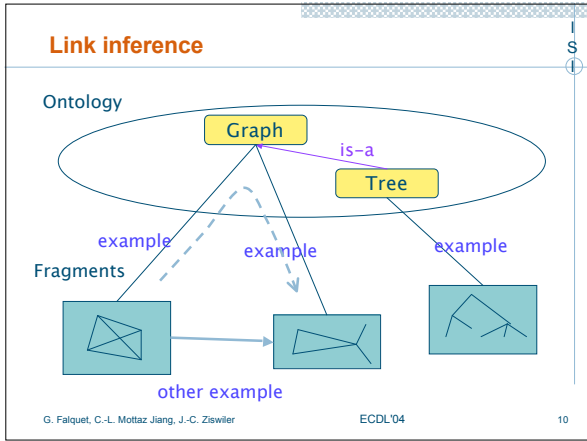
example

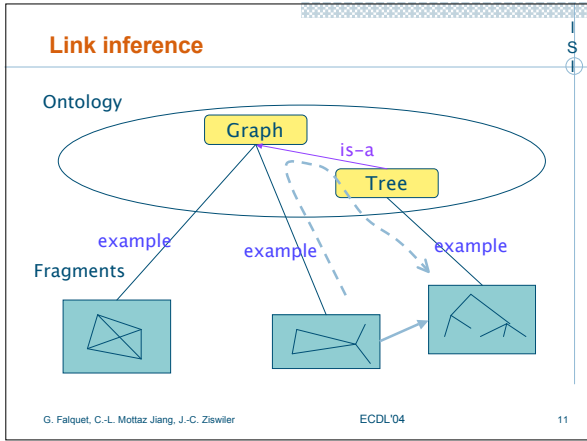
Fragments

G. Falquet, C.-L. Mottaz Jiang, J.-C. Ziswiler

ECDL'04

9





Node schema for link inference

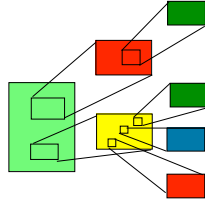
```
// Fragment related to F through a concept
node related_fragments[F]
{.....
  href show_fragment[F2] (F2.title)
  .....}
from
  Fragment F -( L )- Concept C -( L2 )- Fragment F2
```

G. Falquet, C.-L. Mottaz Jiang, J.-C. Ziswiler EC DL'04 12

Building complex nodes

```
node show_fragment[F]
```

```
{  
  F.title, F.content  
  include related_fragments[F]  
  include related_concepts[F]  
}  
from Fragment F
```



nodes as
construction bricks

G. Falquet, C.-L. Mottaz Jiang, J.-C. Ziswiler

ECOL04

13

The screenshot shows a web browser window with a URL bar containing 'cuisin1 ADM...'. The page content includes a main section for 'Exercice d'automate' with a description and a 'Notes voisines' section. A sidebar on the right contains 'Concepts' and 'Notes' sections. The 'Notes' section lists a 'solution' for the exercise.

This screenshot is identical to the previous one, but with green and blue rectangular boxes highlighting the 'Exercice d'automate' content area, the 'Notes voisines' section, and the 'solution' entry in the sidebar's 'Notes' section.

Digital library of hyperbooks

Integrate hyperbooks in a library

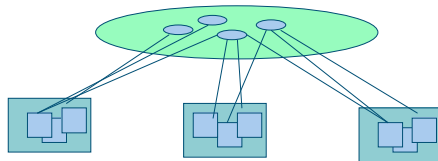
Provide global reading interfaces

- Interconnect hyperbooks
- Combine the contents of several hyperbooks

Semantic coherence must be ensured

Strong integration

A single ontology for all the hyperbooks



Drawbacks

- unification of the book concepts (difficult, conflict resolution, ...)
- a single point of view

Multiple point of views in ontologies

Single point of view

- one concept, several definitions => conflict
- resolution: come to one, consensual, definition

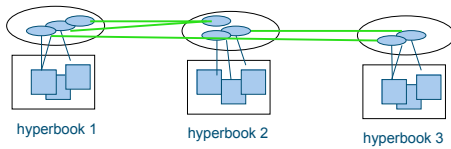
Multiple point of views:

several definitions are accepted
provided they

- belong to different point of views
e.g. "electron" defined by
 - a physicist, a chemist, an electronic engineer
- are not contradictory

Multi-point of view approach for hyperbooks

- Each book represents a point of view
 - concepts may have different definitions
- Ontology alignment instead of integration
 - **similarity** links between concepts



Global reading in the integrated DL

Extending a book

Reading a book through the interface of another one

- apply the interface (node schemas) of book A to book B

Second level authoring

- re-use existing node schemas or create new ones
- new virtual book

Other forms ...

Extending a hyperbook

- Adapt the hyperbook interface
- Apply it to the whole library

in path expressions, replace

... --> Concept c --> ...

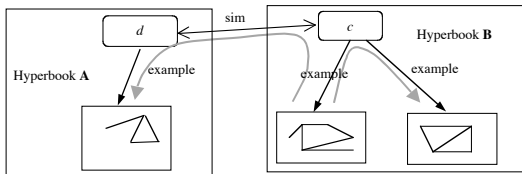
by

... --> Concept c --"sim"--> Concept c' --> ...

or

... --> Concept c --"sim" [level > threshold] --> Concept c' --> ...

Cross book link inference



Finite state automaton

ofs1

A FSA is comprised of

1. an alphabet (set of symbols) $A = \{s_1, s_2, \dots, s_n\}$
2. a set of states $Q = \{q_1, q_2, \dots, q_m\}$ (les ronds)
3. an initial state q_1 de Q
4. a set F of final states, F is a subset of Q
5. a transition function δ from $Q \times A$ to Q

[\[Edit\]](#) [\[Del\]](#) [\[New\]](#)

Example Fragments

[Accept an even number of b's](#)
[Accept numeric expressions](#)

From other books

Example (from concept "Deterministic FSA" in "TCS-1")
[Check divisibility by 5](#)
[Accept \$\(ab+aa\)^+b\$](#)

Example (from concept "State machine" in "APPL")
[Light switch](#)
[Traffic control](#)
[Automatic gate](#)

Related Concepts

[←-reference Regular language](#)
[←-reference Accepting words](#)
[←-\[Equivalent...\]-> Non deterministic FSA](#)
[←-\[Compar...\]-> Turing machine](#)

Linked Fragments (by type)

property (1)
 example (2)
 exercise (4)
 reference (2)

Finite state automaton

ofs1

A FSA is comprised of

1. an alphabet (set of symbols) $A = \{s_1, s_2, \dots, s_n\}$
2. a set of states $Q = \{q_1, q_2, \dots, q_m\}$ (les ronds)
3. an initial state q_1 de Q
4. a set F of final states, F is a subset of Q
5. a transition function δ from $Q \times A$ to Q

[\[Edit\]](#) [\[Del\]](#) [\[New\]](#)

Example Fragments

[Accept an even number of b's](#)
[Accept numeric expressions](#)

From other books

Example (from concept "Deterministic FSA" in "TCS-1")
[Check divisibility by 5](#)
[Accept \$\(ab+aa\)^+b\$](#)

Example (from concept "State machine" in "APPL")
[Light switch](#)
[Traffic control](#)
[Automatic gate](#)

Related Concepts

[←-reference Regular language](#)
[←-reference Accepting words](#)
[←-\[Equivalent...\]-> Non deterministic FSA](#)
[←-\[Compar...\]-> Turing machine](#)

Linked Fragments (by type)

property (1)
 example (2)
 exercise (4)
 reference (2)

Similarity links

SIM links set manually or automatically

for instance

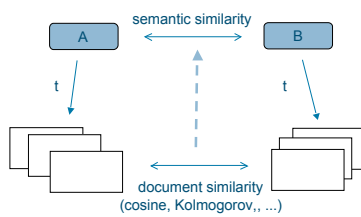
- [Rodríguez and Egenhofer] semantic similarity technique
 - similarity of terms (synonyms)
 - + similarity of attributes
 - + similarity of semantic neighbourhoods

+

typed document similarity

Typed document similarity

Document similarity => concept similarity



Implementation

Hyperbook (frag. + onto + links)

- relational database
- Fragment, Concept, Link ... tables

Interface specification

- Lazy declarative hypertext view system
- node => Lazy node

Similarity

- computed and stored in DB
- inserted by users

Ongoing and future work

Current experiments

- Collaborative writing of several hyperbooks by students
- Hyperbook integration (single interface)

Concept similarity functions

- Incremental computation of similarities

Interface specification

- Towards reader/writer-defined interfaces
