10th Concept Lattices and their Applications
October 18, 2013 – La Rochelle

An efficient Java implementation of the immediate successors calculation

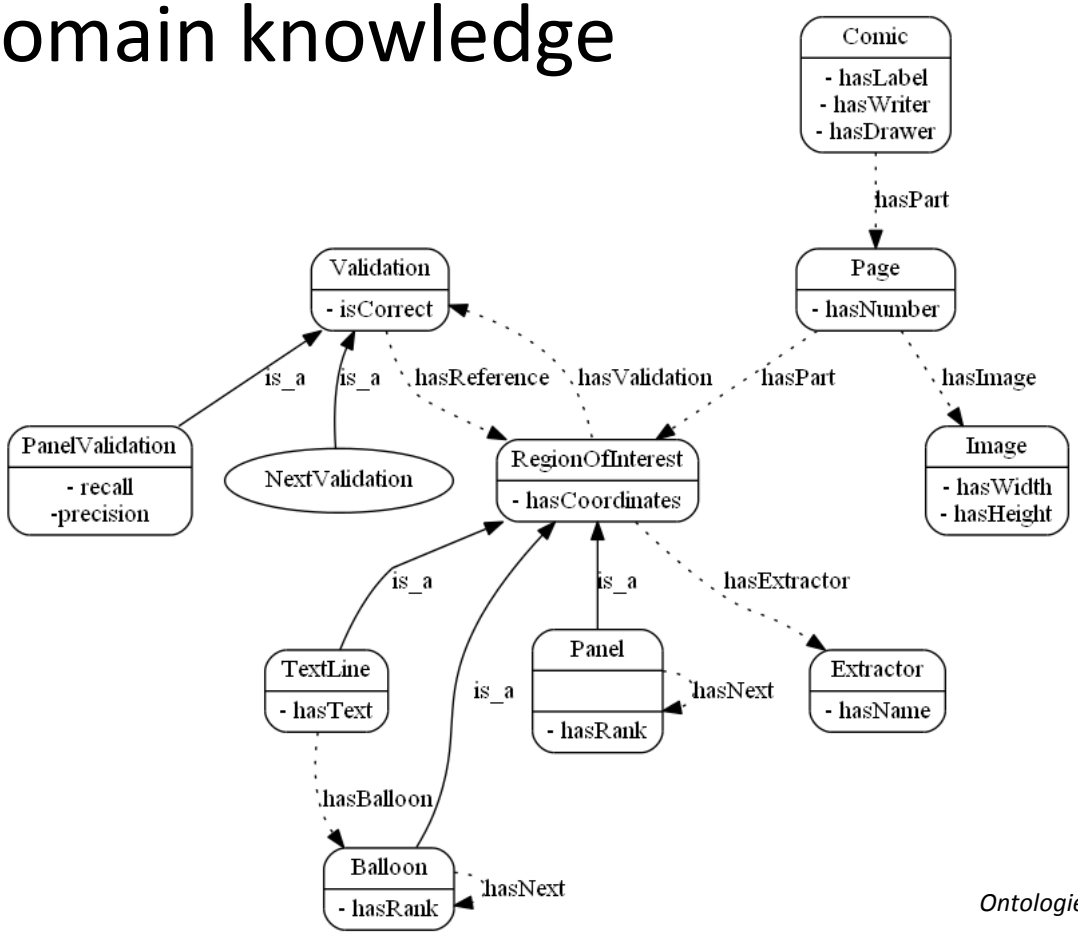
Clément Guérin
Karell Bertet
Arnaud Revel

Context

- eBDtheque project
  - 2 PhD candidates, 4 professors
- Two research axis
  - Extraction of comic books visual items
  - Reasonning on these items
- One goal: how to help comic books to take the right turn on their digital metamorphosis?

Comic books domain model

- A model combining comic books' and image's domain knowledge



Reference

C. Guérin, K. Bertet, A. Revel

Ontologies and spatial relations applied to comic books reading

EKAW's PhD Symposium, Galway, October 2012

What for?

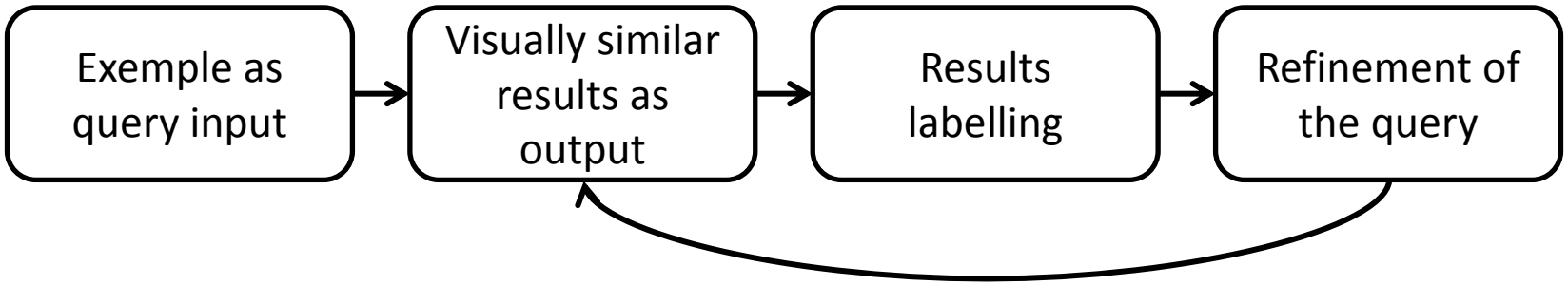
- Tools to ease the conversion from printed to digital comic books
  - Panels and balloons ordering
  - Association of balloons to characters
  - Scene understanding

- Information retrieval
  - Word spotting
  - Linguistic study
  - Semantic Content Base Image Retrieval

Reference
C. Guérin, K. Bertet, A. Revel
Ontologies and spatial relations applied to comic books reading
EKAW's PhD Symposium, Galway, October 2012

Semantic Content Based Image Retrieval

- From classical Content Based Image Retrieval (CBIR)



- Use the model description of the panels to retrieve *semantically and/or visually similar* panels

Reference

C. Guérin, K. Bertet, A. Revel

An approach to Semantic Content Based Image Retrieval using Logical Concept Analysis. Application to comicbooks.

FCA4AI Workshop of ECAI, Montpellier, August 2012

What annotations?

- Panels attributes
  - shape, size, position, rank, shot type, etc.

- What is “inside” a panel
  - speech balloons, characters, words, objects, visual features…

- What is “outside” a panel
  - the author of the comic, its style, its year of publication, etc.

Reference

C. Guérin, K. Bertet, A. Revel

An approach to Semantic Content Based Image Retrieval using Logical Concept Analysis. Application to comicbooks.

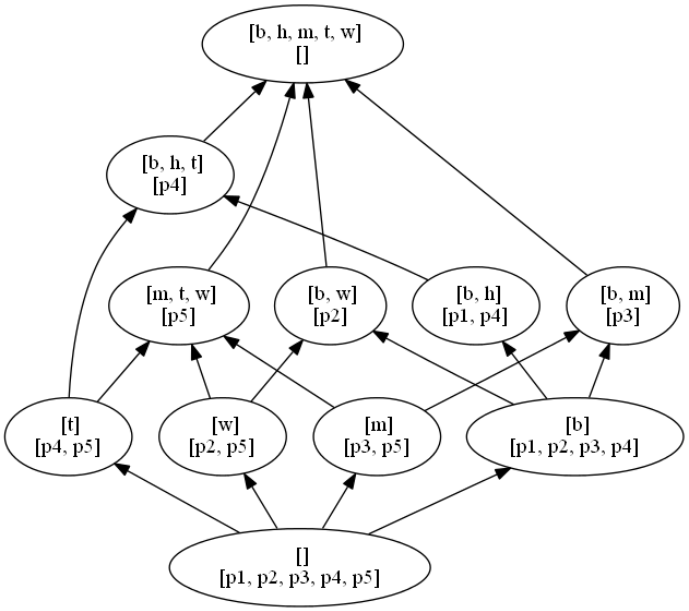
FCA4AI Workshop of ECAI, Montpellier, August 2012

Example

	p1	p2	p3	p4	p5
contains:balloon	1	1	1	1	0
shape:wide	0	1	0	0	1
shape:high	1	0	0	1	0
size:medium	0	0	1	0	1
contains:tree	0	0	0	1	1

Browsing context

- Formal Concept Analysis
  - Concept lattices gather panels described by the same set of attributes in concepts
  - Set of panels (resp. attributes) ordered in a browsable hierarchical structure



- Need to get the successors and predecessors of a concept very quickly

Reference
C. Demko, K. Bertet
Generation algorithm of a concept lattice with limited object access
CLA, Nancy, October 2011

Lattice Java API

- Java language
  - Widely used programming language
  - Lattice Java API

- Implemented immediate successors algorithm from Bordat's (or Lindig's) theorem

Data: A context $K = (O, I, (\alpha, \beta))$; A closure B of the closure lattice of K
Result: The immediate successors of B in the closure lattice
begin
  Init the set system $\mathcal{F}_B$ with $\emptyset$;
  foreach $x \in I \backslash B$ **do**
    Add $\varphi(x + B)$ to $\mathcal{F}_B$
  end
  $Succ$=minimal inclusion subsets of $\mathcal{F}_B$;
  return $Succ$
end

- Not fast enough

Limited Object Access algorithm

- Improved efficiency by reducing the subset of observations to its cardinality

- The count function *c* associates to any subset *X* of attributes the cardinality of the subset *ϐ(X)*

$$\varphi(B + X) \subseteq \varphi(B + x) \Longleftrightarrow c(B + X + x) = c(B + X)$$

- Refer to the reference paper for proof and details

Reference

C. Demko, K. Bertet

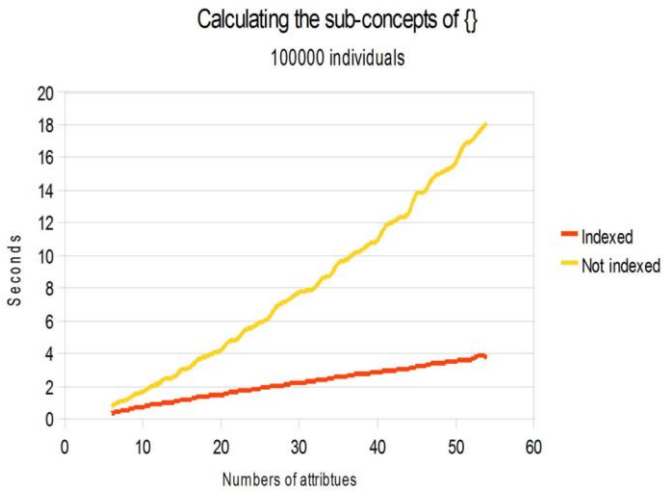
Generation algorithm of a concept lattice with limited object access

CLA, Nancy, October 2011

LOA performances

Data: A context $K = (O; I, (\alpha, \beta))$; A closed set B of the closed set lattice $(\mathbb{C}_I, \subseteq)$ of K
Result: The immediate successors of B in the lattice
begin
  Init the set system $Succ_B$ with $\emptyset$;
  foreach $x \in I \setminus B$ **do**
    add = true;
    foreach $X \in Succ_B$ **do**
      \\ **Case 1: Merge x and X** in the same potential successor
      if $c(B + x) = c(B + X)$ **then**
        if $c(B + X + x) = c(B + x)$ **then**
          replace X by $X + x$ in $Succ_B$;
          add=false; break;
        end
      end
      \\ **Case 2: Eliminate x** as potential successor
      if $c(B + x) < c(B + X)$ **then**
        if $c(B + X + x) = c(B + x)$ **then**
          add=false; break;
        end
      end
      \\ **Case 3: Eliminate X** as potential successor
      if $c(B + x) > c(B + X)$ **then**
        if $c(B + X + x) = c(B + X)$ **then**
          delete X from $Succ_B$
        end
      end
    end
    \\ **Case 4: Insert x** as a new potential successor ;
    if *add* **then** add $\{x\}$ to $Succ_B$;
  end
  return $Succ_B$;
end

- Good performances using SQL indexing mechanism



- How to get it efficient without SQL?

Reference
C. Demko, K. Bertet
Generation algorithm of a concept lattice with limited object access
CLA, Nancy, October 2011

LOA Java implementation

- Naive implementation of LOA using “classical” data containers (*TreeSet*, *HashSet*…)
- The most time consuming task in the immediate successors computation is the calculation of the extent
- The most time consuming task in the extent computation is the intersection between two sets of elements
- Time can be won here

Attributes	Extent
contains:balloon	{p1, p2, p3, p4}
shape:wide	{p2,p5}
shape:high	{p1, p4}
size:medium	{p3,p5}
contains:tree	{p4,p5}

Binary words

- Objects and attributes are reduced to their index
  - They have to be ordered though
  - The index is stored to retrieve
- Extents and intents are expressed as binary words using Java's *BitSet*
- Intersection is performed by a *logical AND*
  - *E(n/w) operations*

	p1	p2	p3	p4	p5
contains:balloon	1	1	1	1	0
shape:high	1	0	0	1	0
contains:tree	0	0	0	1	1
Extent of (b,h,t)	0	0	0	1	0

Experiment

- First dataset:
  - 848 comic books' panels
  - 100 attributes (avg. 7 attributes/panels)

- Second dataset:
  - 848 comic books' panels
  - 3533 attributes (avg. 15 attributes/panels)
  - 3403 of these are shared by less than 3 panels

Results

- Calculation of the *immediate successors* of the *bottom concept* and *immediate predecessors* of the *top concept*

Computation time (in ms)	Immediate successors		Immediate predecessors	
	|O| = 848	|O| = 848	|O| = 100	|O| = 3533
	|I| = 100	|I| = 3533	|I| = 848	|I| = 848
Classical + TreeSet	3.06	11767.52	549.76	994.00
Classical + BitSet	0.77	196.58	62.39	9.77
LOA + TreeSet	0.29	11.26	5.65	1183.75
LOA + BitSet	0.02	0.15	0.24	1.20

- 0.18 second on 500 randomly picked concepts

Conclusion & perspectives

- Conclusions
  - Processing time kept below the second on a reasonable machine
  - Efficient way to browse a context without generating its lattice

- Perspectives
  - Limit the impact of the amount of attributes on the performances
  - Make that library available

Thank you.

Any questions?