Scaling Conceptual Navigation in Expressivity, Usability, and Efficiency

Sébastien Ferré
Team LIS, Irisa, Université de Rennes 1

ICFCA 2013, Dresden
Motivation

- Information access with more and more information to be accessed
- people to access this information
- questions to ask about this information

This requires information systems that combine
- expressivity: How many questions?
- usability: How many people?
- efficiency: How much information?

To which extent can we combine them?
What level of trade-off is required?
Overview

1. Navigation vs Querying
2. Conceptual Navigation
3. Expressivity, Usability, and Efficiency
4. Abstract Conceptual Navigation
5. Concrete Conceptual Navigations
   - Camelis
   - Sewelis
   - Cubes of Concepts [with Pierre Allard]
   - Possible World Explorer (PEW) [with Sebastian Rudolph]
6. Conclusion and Perspectives
Paradigm 1: Navigation

- **Definition:** moving from place to place by following navigation links
- **Examples:** file systems, hypertext
- **Expressivity:**
  - determined by the navigation structure, generally designed by hand
  - very low in general: one object at-a-time (e.g., folder, page, entity)
- **Usability:** good (point-and-click) but one can get lost in navigation
- **Efficiency:** very high (e.g., scales to the Web!)

*Expressivity is sacrificed for efficiency and usability.*
Paradigm 2: Querying

- **Definition**: formulating a query, and waiting for results
- **Examples**: search engines, databases, natural language interfaces
- **Expressivity**: a large spectrum
  - from **medium** (e.g., keyword search) to **very high** (e.g., SPARQL)
  - from **formal** (e.g., SPARQL) to **natural** (e.g., Wolfram Alpha)
- **Usability**: depends on the query language
  - good for natural languages, very bad for formal languages
  - generally inversely proportional to expressivity
  - in all cases: writer’s block, lack of feedback, empty results
- **Efficiency**: medium (e.g., SPARQL) to high (e.g., keyword search)

*Limited usability, and trade-off with expressivity.*
Paradigm 3: Conceptual Navigation

Combines navigation (paradigm 1) and querying (paradigm 2)

- introduced in FCA [Godin, Missaoui, April 1993]
- the navigation structure is the concept lattice, which is automatically generated from a formal context $K = (O, A, I)$
- queries are sets of attributes, which point to concepts: $q \mapsto (q', q'')$, for $q \subseteq A$
Navigation in the Concept Lattice

(S. Ferré, LIS, Irisa)
Navigation in the Concept Lattice

(S. Ferré, LIS, Irisa)
Navigation in the Concept Lattice

(S. Ferré, LIS, Irisa)
Querying in the Concept Lattice

Conceptual Navigation

(S. Ferré, LIS, Irisa)

ICFCA'13
Evaluation of Conceptual Navigation

- **Expressivity**: low (sets of attributes) but global view
  - slightly increased with scaling: valued attributes, intervals, taxonomies

- **Usability**: ease of use, feedback
  - difficult to read the Hasse diagram of concept lattices for the layman
  - overwhelming information for large contexts
  - mitigated by local views (concept neighborhood)

- **Efficiency**: very low given the exponential complexity of computing concept lattices
  - local views allow for polynomial on-demand computation
  - many approaches to filter or abstract concept lattices, but less concepts means less expressivity
Evaluation of Conceptual Navigation (cont’d)

Expressivity, Usability, and Efficiency

Best at combining expressivity and usability, but low on each axis.
Abstract Conceptual Navigation: principles (1/2)

A theoretical framework

- that encompasses past and future works
  - Logical Concept Analysis (LCA) [ICCS’00]
  - Query-based Faceted Search (QFS) [ISWC’11]
- that relaxes some FCA principles
  - extents are query results
  - no unique most-specific intent
  - concept intent replaced by equivalent queries and an “index”
  - local view: current concept + “neighborhood”
- but that retains the FCA philosophy
  - forth and back between extensional representations (query results) and intensional representations (queries, indexes)
  - navigation structure automatically derived from data
Abstract Conceptual Navigation: principles (2/2)

- **place** = concept = *intension* (query + index) + *extension* (results)
- **link** = neighbour concept = *suggested* query transformation

![Diagram of conceptual navigation](chart.png)

Knowledge Base

query → index

intension

extension

results

concept

links

*
Abstract Conceptual Navigation: definitions (1/2)

Abstract components, and their instanciation for FCA:

- **knowledge base** ($K$): data, facts, rules, domain knowledge, etc.
  - in FCA: formal context, binary relation $K = (O, A, I)$
- **query language** ($Q$): expressible queries
  - in FCA: sets of attributes $Q = 2^A$
- **extensions** ($E$ and $ext \in Q \rightarrow E$): query results
  - in FCA: sets of objects $E = 2^O$
  - in FCA: usual extent $ext(q) = q' = \{ o \in O \mid \forall a \in q : (o, a) \in I\}$

...
Abstract Conceptual Navigation: definitions (2/2)

Abstract components, and their instanciation for FCA:

- **indexes** ($I$ and $index \in \mathbb{Q} \times E \to I$): support feedback and navigation
  
  - in FCA: frequency of attributes over an extension
  - in FCA: $I = A \to \mathbb{N}$
  - in FCA: $index(q, e) = \{ a \mapsto n \mid a \in A, n = \#\{ o \in e \mid (o, a) \in I \}\}$
  - here, the index includes the usual **intent**, and extends it to “partially shared” attributes

- **links** ($links \in \mathbb{Q} \times E \times I \to 2^\mathbb{Q}$): navigation links to “neighbor” queries
  
  - in $links(q, e, i)$, we assume $e = ext(q)$, and $i = index(q, e)$
  - links can be absolute or relative, and can be anchored to elements of $q$, $e$ or $i$
  - in FCA: $links(q, e, i) = \{ \emptyset \} \cup \{ q \cup \{ a \} \mid (a \mapsto n) \in i, 0 < n < \#e \}$
  - here, new search (top) or refinement with a discriminating attribute, avoiding empty results (bottom)
Abstract Conceptual Navigation: definitions (2/2)

Abstract components, and their instanciation for FCA:

- **indexes** \((I\text{ and } index \in Q \times E \rightarrow I)\): support feedback and navigation
  - in FCA: frequency of attributes over an extension
  - in FCA: \(I = A \rightarrow \mathbb{N}\)
  - in FCA: \(index(q, e) = \{a \mapsto n \mid a \in A, n = \#\{o \in e \mid (o, a) \in I\}\}\)
  - here, the index includes the usual intent, and extends it to “partially shared” attributes

- **links** \((\text{links} \in Q \times E \times I \rightarrow 2^Q)\): navigation links to “neighbor” queries
  - in \(\text{links}(q, e, i)\), we assume \(e = \text{ext}(q)\), and \(i = index(q, e)\)
  - links can be absolute or relative, and can be anchored to elements of \(q\), \(e\) or \(i\)
  - in FCA: \(\text{links}(q, e, i) = \{\emptyset\} \cup \{q \cup \{a\} \mid (a \mapsto n) \in i, 0 < n < \#e\}\)
  - here, new search (top) or refinement with a discriminating attribute, avoiding empty results (bottom)
Abstract Conceptual Navigation: evaluation

- **Expressivity:**
  - $K, Q, E, \text{ext}$: theory and efficient algorithms can be reused from database and semantic web research communities
  - difficulty: invent index and links for each query language

- **Usability:**
  - point-and-click user interface, no exposure of FCA notions
  - same interface and interaction paradigm as Faceted Search (FS), which is already in use in many e-commerce sites
  - difficulty: presentation and construction of complex queries

- **Efficiency:**
  - $\text{ext}$: a lot of research exists (e.g., full-text, SPARQL)
  - $\text{index}$: well studied in FS, but still open for more expressive queries
  - $\text{links}$: a priori not an issue (mostly part of index)
  - difficulty: computation of the index for large results
Overview of concrete conceptual navigations:

2000  **CAMELIS**: logical contexts, sets of objects, arbitrary descriptors

2009  **SEWELIS**: RDF graphs, SPARQL-like queries (Semantic Web)

2011  **Cubes of concepts**: dimensions, measures, aggregations (OLAP)  
      [Pierre Allard]

2012  **PEW**: possible world exploration in OWL ontologies  
      [Sebastian Rudolph]
SEWELIS: definitions (1/2)

- **K**: a RDF graph = a set of triples (labeled links) = a set of binary relations = a set of formal contexts
  - similar to context families of RCA [Rouane et al. 2007]
- **Q**: LISQL, a mild-syntax fragment of SPARQL
  - class expressions + co-references + **focus**
  - A person and birth : (year : (1601 or 1649) and place : (?X and part of England)) and father : birth : place : not ?X “people born in 1601 or 1649 somewhere in England, and whose father was not born there”
  - A person and birth : (year : (1601 or 1649) and place : (?X and part of England)) and father : birth : place : not ?X “places in England where a person was born in 1601 or 1649, and where his father was not born”
**SEWELIS: definitions (2/2)**

- $\text{ext}(q) \in E$: extents are still sets of objects (RDF nodes)
  - but membership of an object depends on its relationships to other objects
  - can be defined by translating LISQL into SPARQL

- $\text{index}(q, e) \in I$: similar to FCA for a partially ordered vocabulary
  - vocabulary: classes, properties (and their inverse), entities, co-references
  - partial order: class hierarchy, property hierarchy
  - the extension is included in the index!

- $\text{links}(q, e, i)$: focus-centered query transformations
  - from index: insertion/removal of index elements
  - from query: focus changes, insertion of Boolean operators
SEWELIS: illustration on genealogy

(S. Ferré, LIS, Irisa)
**SEWELIS: evaluation**

- **Expressivity:** LISQL covers a large part of SPARQL 1.0
  - all Boolean combinations, cyclic graph patterns
  - completeness theorem: every LISQL query is reachable
  - the focus is a key element
  - but no multi-dimensional queries and analytics, weak on concrete domains (e.g., string matching, inequalities)

- **Usability:** similar to faceted search
  - safeness theorem: no empty results
  - user studies performed on CS students
    - complex queries OK, except scope of negation and co-references
    - focus is confusing at the beginning but most users catch it
    - users quickly improve in correctness and response time

- **Efficiency:** same complexity as faceted search for same queries
  - more costly for more complex queries
  - OK up to a few million triples (a few minutes loading time)
Cubes of concepts [with Pierre Allard]

An extension of CAMELIS that supports OLAP-like analytics
- queries (i.e., selections of objects) are extended with
  - dimensions (cube axes): selected attributes
  - measures (cube cells): selected attributes + aggregations
- extensions are cubes of (FCA) concepts
- indexes remain the same
- additional links for changing dimensions and measures

Recently integrated to SEWELIS (2013)
Cubes of concepts: illustration on bibliography

In which years (since 2000) were journal and conference papers published per author and per type?
**P EW [with Sebastian Rudolph]**

An adaptation of **SEWELIS** for **OWL** ontologies

- **K**: a set of OWL axioms, possibly knowledge only (no facts)
  - example: the Pizza ontology

- **Q**: situations
  - OWL class expressions, only atomic negation (subset of LISQL)
  - a Pizza and topping : Tomato

- **ext(q)**: the satisfiability of the query (a Boolean!)

- **index(q, e)**: similar to SEWELIS but satisfiability instead of non-zero frequency

- **links(q, e, i)**: subset of SEWELIS

**Application**: ontology understanding and completion

- exploration of possible worlds (satisfiable class expressions)
- identification of unexpected worlds
- exclusion of unexpected worlds by asserting new axioms
- e.g., we found that a Veggie pizza could contain meat or fish!
Conclusion

- **expressivity**: gets close to SPARQL
- **usability**: similar to faceted search but needs more user studies
- **efficiency**: same complexity as the computation of query results
Conclusion and Perspectives

Perspectives w.r.t. Expressivity

- **Full SPARQL 1.1**: expressions, property paths, named graphs
- **Data mining**: for the discovery of patterns/rules [with Peggy Cellier]
  - queries would be **constraints** on searched patterns/rules (results)
  - constraints could be about support, confidence, rule conclusion, gap size in motifs
  - i.e., navigation in **Inductive Databases**
- **Scientific workflows**: guided composition of tasks [PhD Mouhamadou Ba, 2012-2015]
  - fragment of a programming language (sequences, conditionals, iterations)
  - queries are **programs**, and results are static **evaluations** (e.g., types)
Perspectives w.r.t. Usability

- **Natural language**: for a better *readability* of complex queries
  - only generation: conceptual navigation avoids the hard problem of NL parsing
  - controlled natural languages for syntax (e.g., ACE, SQUALL)
  - lexicons on top of ontologies

- **Visualization**: for a better feedback and understanding-at-a-glance
  - in both results and the index
  - graphics, timelines, maps, excerpts, summaries, etc.
Perspectives w.r.t. Efficiency

- **Approximate results and indexes**: for huge datasets and result sets [MSc Joris Guyonvach]
  - computing indexes on partial results
  - reasoning with ontologies (much smaller than data)
  - objective: conceptual navigation on top of DBpedia (2G triples)

- **Knowledge base distribution**:
  - requires federated search (search over the Web)
  - allows for the parallelization of query evaluation
The End

Thank you for your attention!

Thanks to the former and present members of team LIS (Logical Information Systems) for their collaboration on many aspects of this work. Visit http://www.irisa.fr/LIS/.

Questions?
Conclusion

- **expressivity**: gets close to SPARQL
- **usability**: similar to faceted search but needs more user studies
- **efficiency**: same complexity as the computation of query results

(S. Ferré, LIS, Irisa)
CAMELIS: definitions (1/2)

- **K**: logical context \((\mathcal{O}, \mathcal{L}, d)\)
  - \(\mathcal{O}\): collection of objects
  - \(\mathcal{L}\): a logic (partial order of object descriptors) – knowledge valued attributes, taxonomies, intervals of numbers/dates, string patterns, and domain specific descriptors (e.g., DNA sequences, linguistic types, Java method signatures)
  - a toolbox of components (logic functors) that can be composed
  - \(d \in \mathcal{O} \rightarrow \mathcal{L}\): description function – facts

- **Q**: Boolean closure of \(\mathcal{L}\) (and, or, not)

- **ext**\((q \in Q) \in E\): set of objects whose description “matches” \(q\)
  - \(\text{ext}(q \in \mathcal{L}) = \{ o \in \mathcal{O} \mid d(o) \sqsubseteq q \}\)
  - \(\text{ext}(q_1 \text{ and } q_2) = \text{ext}(q_1) \cap \text{ext}(q_2)\)
  - \(\text{ext}(q_1 \text{ or } q_2) = \text{ext}(q_1) \cup \text{ext}(q_2)\)
  - \(\text{ext}(\text{not } q_1) = \mathcal{O} \setminus \text{ext}(q_1)\)
CAMELIS: definitions (1/2)

- $K$: logical context $(O, L, d)$
  - $O$: collection of objects
  - $L$: a logic (partial order of object descriptors) – knowledge valued attributes, taxonomies, intervals of numbers/dates, string patterns, and domain specific descriptors (e.g., DNA sequences, linguistic types, Java method signatures)
  - a toolbox of components (logic functors) that can be composed
  - $d \in O \rightarrow L$: description function – facts

- $Q$: Boolean closure of $L$ (and, or, not)

- $\text{ext}(q \in Q) \in E$: set of objects whose description “matches” $q$
  - $\text{ext}(q \in L) = \{ o \in O \mid d(o) \sqsubseteq q \}$
  - $\text{ext}(q_1 \text{ and } q_2) = \text{ext}(q_1) \cap \text{ext}(q_2)$
  - $\text{ext}(q_1 \text{ or } q_2) = \text{ext}(q_1) \cup \text{ext}(q_2)$
  - $\text{ext}(\text{not } q_1) = O \setminus \text{ext}(q_1)$
**CAMELIS: definitions (2/2)***

- **index**\((q, e) \in I\): similar to FCA for a partially ordered *vocabulary of properties*

- **links**\((q, e, i) \subseteq Q\): different kinds of *moves* in concept lattice
  - **downward**: adding an index element, replacing more general query elements
    
    France \(\rightsquigarrow\) France and Building
  
  - **upward**: removing a query element, or replacing it with a more general descriptor
    
    France and Building \(\rightsquigarrow\) Europe and Building
  
  - **sideward**: downward+upward or upward+downward
    
    Europe and Building \(\rightsquigarrow\) Europe and Lanscape
CAMELIS: illustration on photos

Australie and (Animal or Plante) and not Portrait

File Logic Browsing Updating Actions Help

Home Back Forward Refresh Save Paste Update 0

NOT = >= <= Zoom Pivot

39 ⊸ date in [.,]
39 ⊸ date = 2004
26 ⊸ date = feb 2004
13 ⊸ date = mar 2004
39 ⊸ event ?
39 ⊸ exif ?
39 ⊸ Location
39 ⊸ Oceanie
39 ⊸ Australie
13 ⊸ 'Feather Dale Park'
1 ⊸ Manly
25 ⊸ Sydney
39 ⊸ Object
33 ⊸ 'animal'
15 ⊸ 'oiseau'

Results: 1 - 12 / 39

(S. Ferré, LIS, Irisa)
CAMELIS: evaluation

- **expressivity:**
  - extensible and unlimited for object descriptors (custom logics)
  - no relation between objects
  - not all Boolean combinations reachable by navigation
  - no analytics (grouping, aggregations, ...)

- **usability:**
  - very similar to faceted search
  - Boolean operators may be an issue for some users
  - some query elements have still to be entered manually

- **efficiency:** biggest context on a recent computer
  - 100,000 files × 70 descriptors/file
  - 5 min for loading + saving (including metadata extraction)