Towards a catalog of design patterns for knowledge discovery with relational concept analysis

Marianne Huchard

Towards a catalog of design patterns for knowledge discovery with relational concept analysis

Marianne Huchard
LIRMM, Montpellier University, CNRS, France

Séminaire
Laboratoire de recherches Transdisciplinaires sur les Écosystèmes Informatiques (LATECE)
Université du Québec à Montréal
16 novembre 2022

Introduction

Formal Concept Analysis

Relational Concept Analysis

Design patterns for RCA

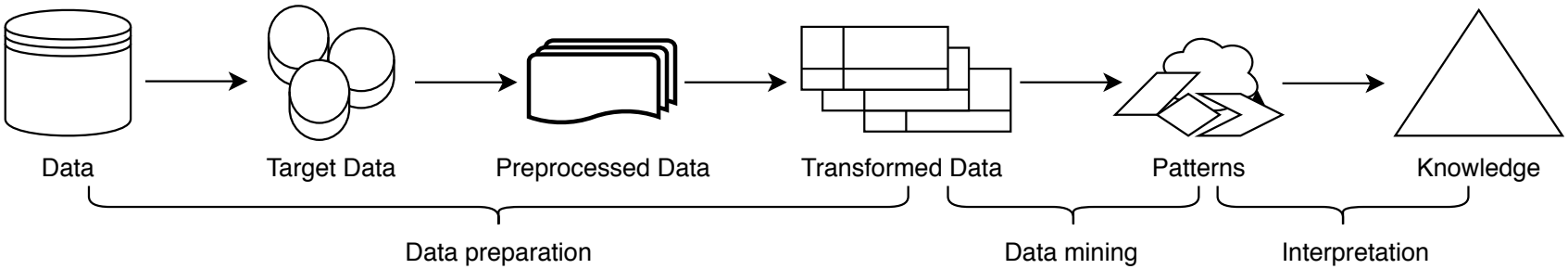
Conclusion

Agenda

Knowledge Discovery (KD)

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. AI Magazine 1996

From Data Mining to Knowledge Discovery in Databases



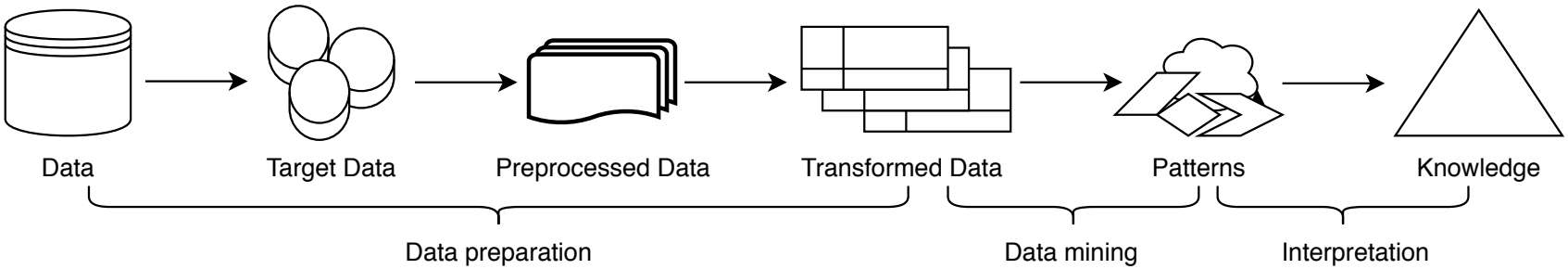
Directions

- Symbolic vs. subsymbolic
- White box vs. black box
- Supervised vs. unsupervised
- Structured vs. unstructured data

Knowledge Discovery (KD)

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. AI Magazine 1996

From Data Mining to Knowledge Discovery in Databases



Directions

- Symbolic vs. subsymbolic
- White box vs. black box
- Supervised vs. unsupervised
- Structured vs. unstructured data

Agenda

Introduction

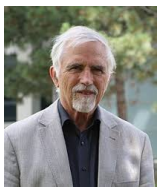
Formal Concept Analysis

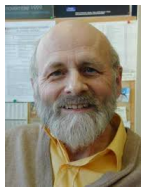
Relational Concept Analysis

Design patterns for RCA

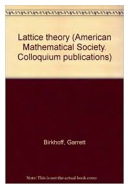
Conclusion

Roots








- Concept lattices
  Rudolf Wille, 1982, Bernhard Ganter & Rudolf Wille, 1999
- Lattice theory, Galois connections, Galois lattices
  Marc Barbut & Bernard Monjardet, 1970; Georges David Birkhoff, 1940; Øystein Ore, 1944
- Galois connection, subgroups/subfields
  Evariste Galois, $\sim$ 1830

Formal Concept Analysis

Formal concepts are "a natural feature of information representation which is as fundamental to hierarchies and object/attribute structures as set theory or relational algebra are for relational databases".

Uta Priss. 40th anniv. vol. of Annual Review of Inf. Sc. and Tech., 2006

Simple but powerful basics

- Formal Context
- Galois connection
- Concept Lattice
- Implications and associations rules

Formal Context = Triple (O, A, R)

O is a finite set of objects, A is a finite set of attributes

$R \subseteq O \times A$ is a binary relation

$(o, a) \in R$ means that object o owns attribute a.

Ingredient	vege	vegan	spring	summer	autumn
goatcheese	×			×	
burrata	×		×		
scallop			×		
tomato	×	×		×	
shallot	×	×			×
mushroom	×	×			×
eggplant	×	×		×	

Formal Concept = Pair (Extent, Intent)

Extent = Maximal set of owner objects

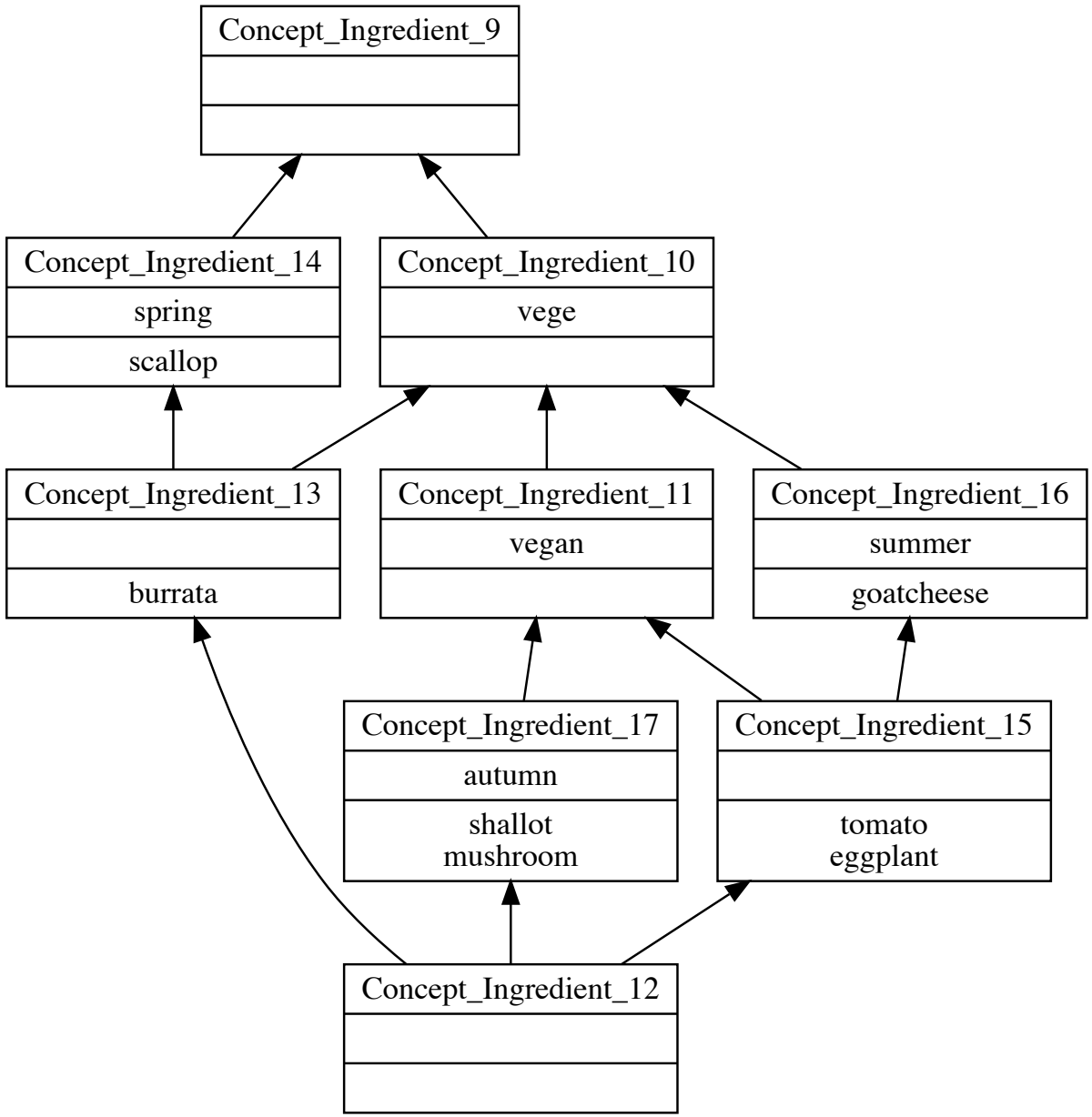
Intent = Maximal set of their shared attributes

Ingredient	vege	vegan	spring	summer	autumn
goatcheese	×			×	
burrata	×		×		
scallop			×		
tomato	×	×		×	
shallot	×	×			×
mushroom	×	×			×
eggplant	×	×		×	

Concept Lattice

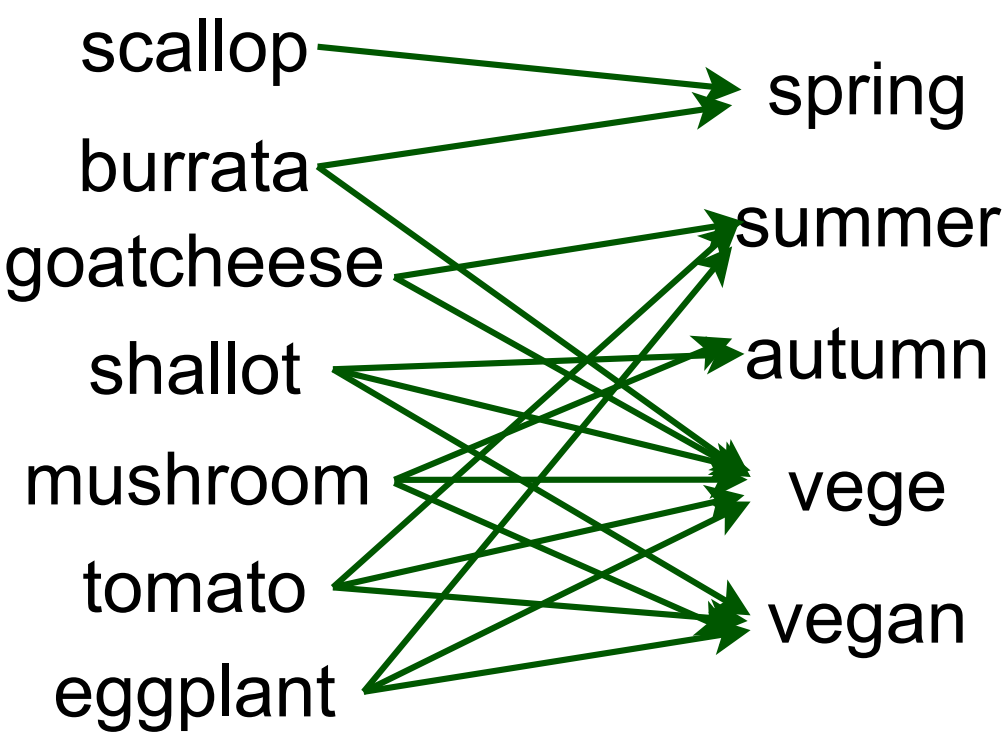
Set of all concepts provided with extent ↑ or intent ↓ inclusion

C_I_11
vege
vegan
shallot
mushroom
tomato
eggplant



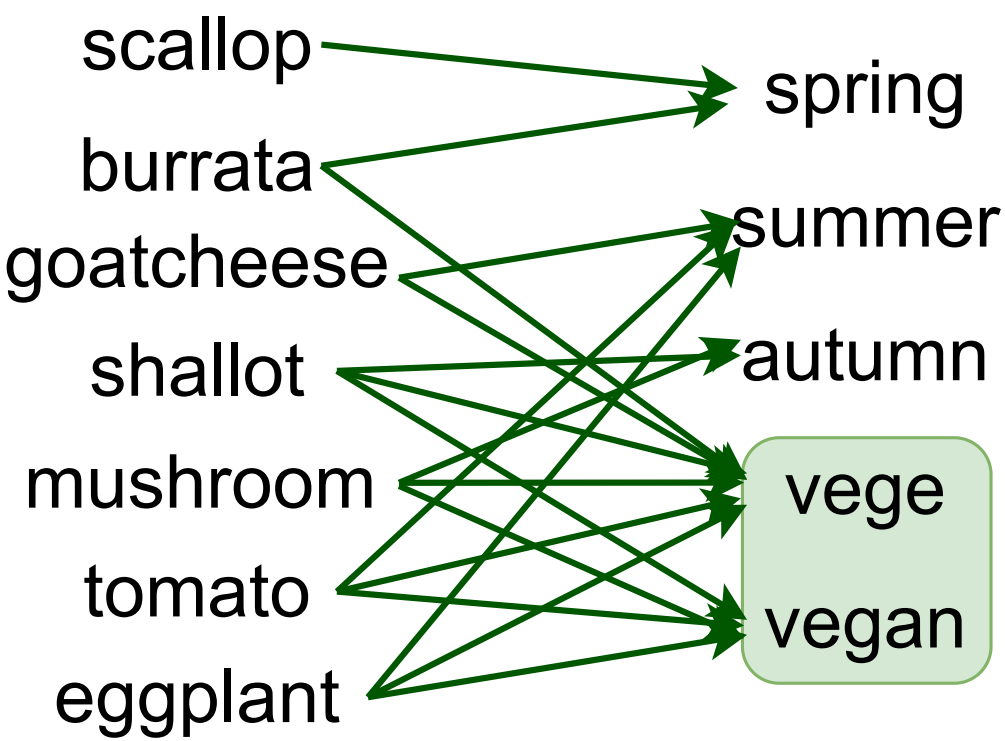
Another view of concept building

Links between objects and attributes



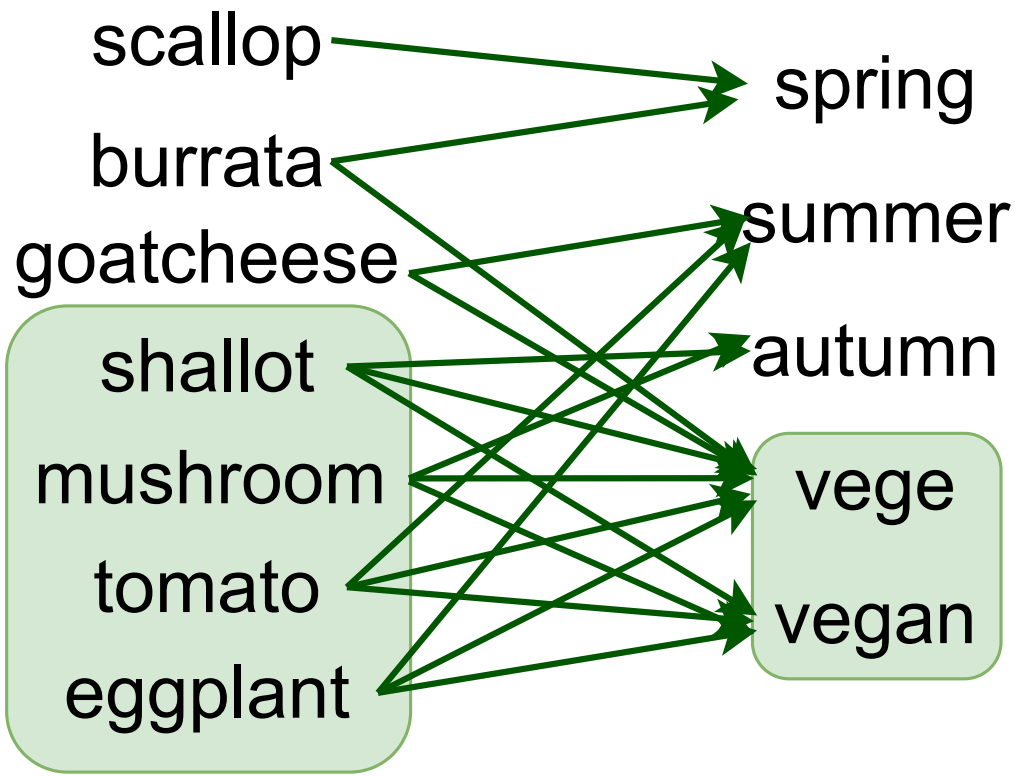
Another view of concept building

Links between objects and attributes vege / vegan



Another view of concept building

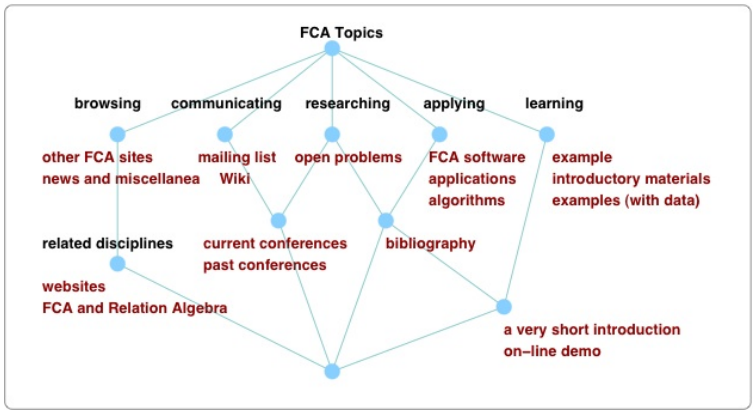
Concept of vegan ingredients



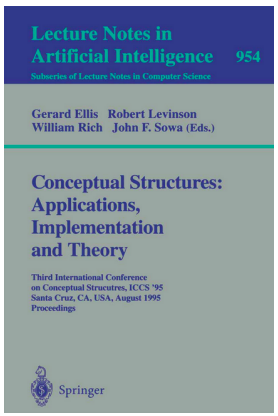
Dissemination 1982-2022

- Conceptualization, philosophical developments
- Data analysis, data mining, clustering
- Knowledge representation (ontology construction)
- Classification, indexation (information retrieval)
- Unsupervised learning
- Supervised learning (adding classes in description)
- Tools
- Applications

Dissemination 1982-2022

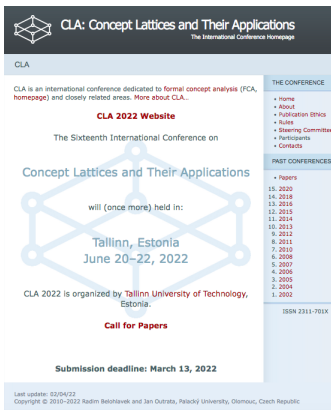


Credits Uta Priss https://upriss.github.io/fca/fca.html

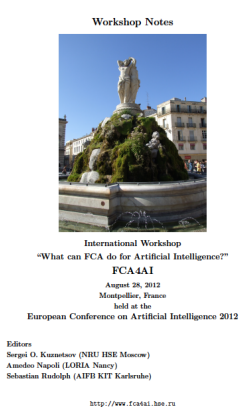


ICCS

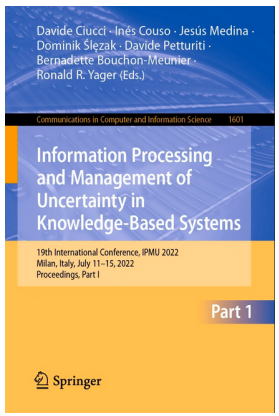
ICFCA



CLA



FCA4AI



IPMU

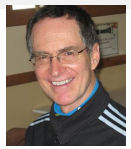
Complex data in FCA

- Multi-valued attributes : integers, double, terms, structures, symbolic objects, etc. (Ganter et Wille, Diday, Polaillon, ...)
- Fuzzy (Belohlavek et al., Cabrera, Cordero, Enciso, Mora, Lòpez-Ròd., Ojeda-Aciego et al., Cornejo, Medina et al., Yahia et al., Dubois Prade ...)
- Value taxonomies (Godin et al., Carpineto et Romano, ...)
- Logical description (Chaudron et al., Ferré et al., ...)
- Graphs (Ganter and Kuznetsov, Liquière, Prediger et Wille, Kötters et al., Graph-FCA Ferré et al....)
- Multi-relational, RCA (Priss, Rouane-Hacène et al., ...); RCA+Fuzzy (Boffa et al.)
- Polyadic (Sacarea, Tronca et al.)
- Sequences (Boukhetta, Demko, Bertet et al., Buzmakov et al.)
- Temporal data (Wolff et al., Nica, Braud, Dolques, Le Ber et al., Boukhetta, Demko, Bertet et al.)
- Pattern Structures (Ganter et al., Kuznetsov et al., Napoli et al., Buzmakov et al.)

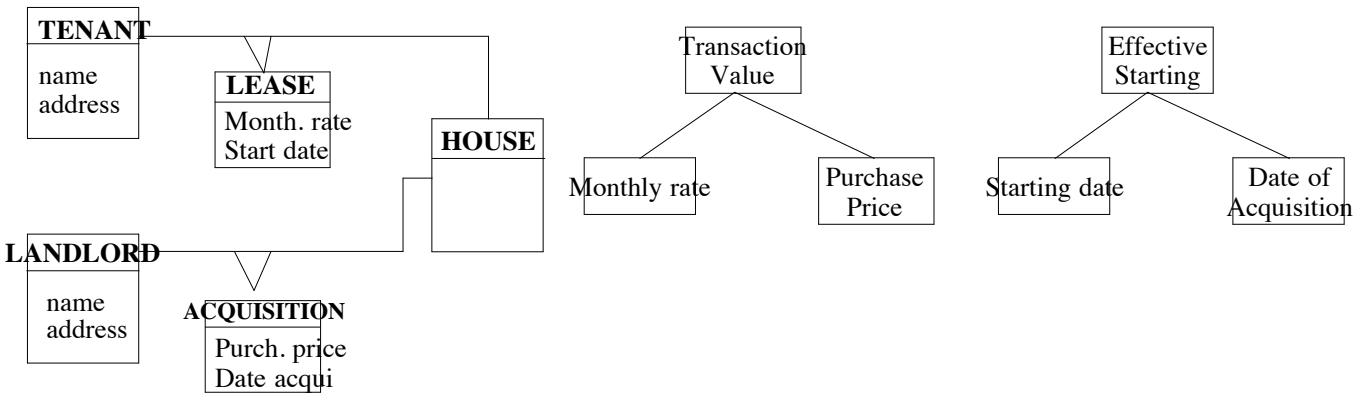
Agenda

Roots: Class model refactorization [Godin&Mili 1993]

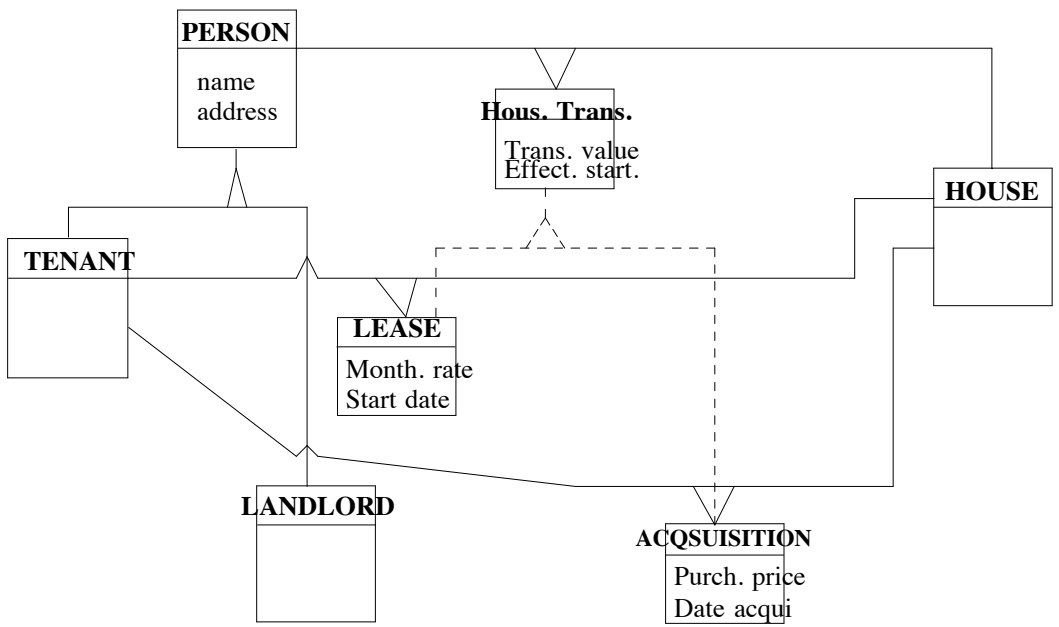
" A formal method that organizes a set of class interfaces into a lattice structure " - Associations in Sect. 3.2.3

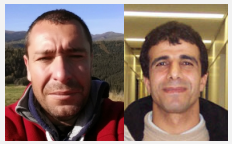



From:

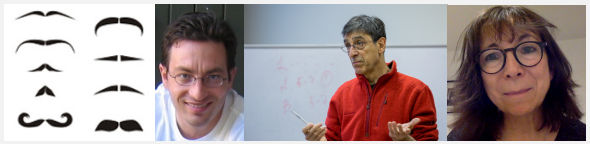


To:





Roots and follow up



- Systematization of the idea of Godin&Mili OOPSLA'93
- Projects
  - MACAO (LIRMM, Orange Labs) 2001-2003/2006-2009
  - PICS CNRS "CAdOE" (LORIA, LIRMM, UQAM) 2011-2013
- Premières thèses de doctorat
  - Cyril Roume, Univ. Montpellier (2004)
  - Mohamed Amine Rouane-Hacène, UQAM (2007)
- Foundational papers
  - M. Huchard, C. Roume, P. Valtchev. When concepts point at other concepts: the case of UML diagram reconstruction. FCAKDD 2002@ECAI
  - M. Dao, M. Huchard, M. Rouane-Hacène, C. Roume, P. Valtchev. Improving generalization level in UML models iterative cross generalization in practice. In Conceptual Structures at Work. ICCS 2004.
  - M. Huchard, M. A. Rouane-Hacène, C. Roume, P. Valtchev: Relational concept discovery in structured datasets. AMAI 2007
  - M. A. Rouane-Hacene, M. Huchard, A. Napoli, P. Valtchev: Relational concept analysis: mining concept lattices from multi-relational data. AMAI 2013

Principles

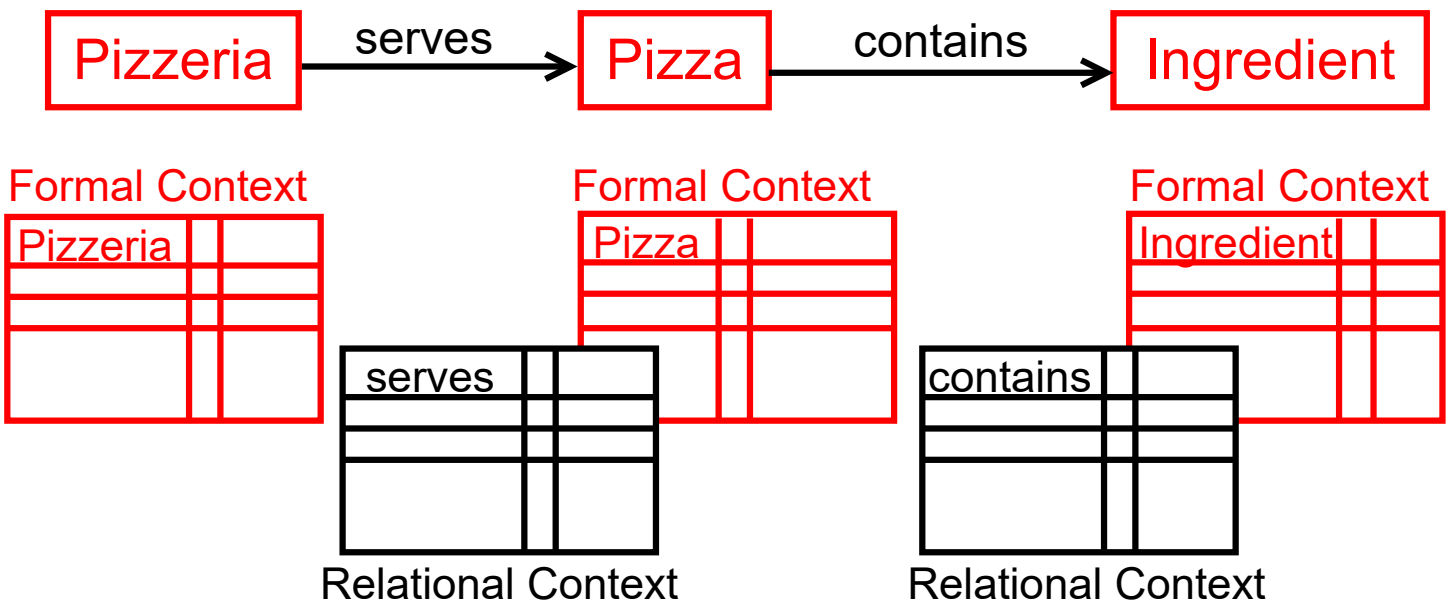
Relational Concept Analysis

- Extends the purpose of FCA for taking into account object categories and links between objects
- Main principles:
  - a relational model based on the entity-relationship model (restricted to unidirectional binary relationships)
  - *relational attributes* integrate relations in formal contexts between objects
  - various operators (*quantifiers*) inspired by description logics
  - iterative and tunable process
- RCA provides a *set of interconnected lattices*
- Produced structures can be written in DLs

Principles: Input data

Relational Context Family (RCF)

- Set of Formal contexts (object-attribute)
  - Represent classes/instances
- Set of Relational contexts (object-object)
  - Represent Unidirectional binary associations/links



Relational Context Family Pizzerias

Pizzeria	*IdHappizzy*	*IdEataly*	*IdLafelicita*	*IdSmallitaly*
happizzy	×			
eataly		×		
lafelicita			×	
smallitaly			×	

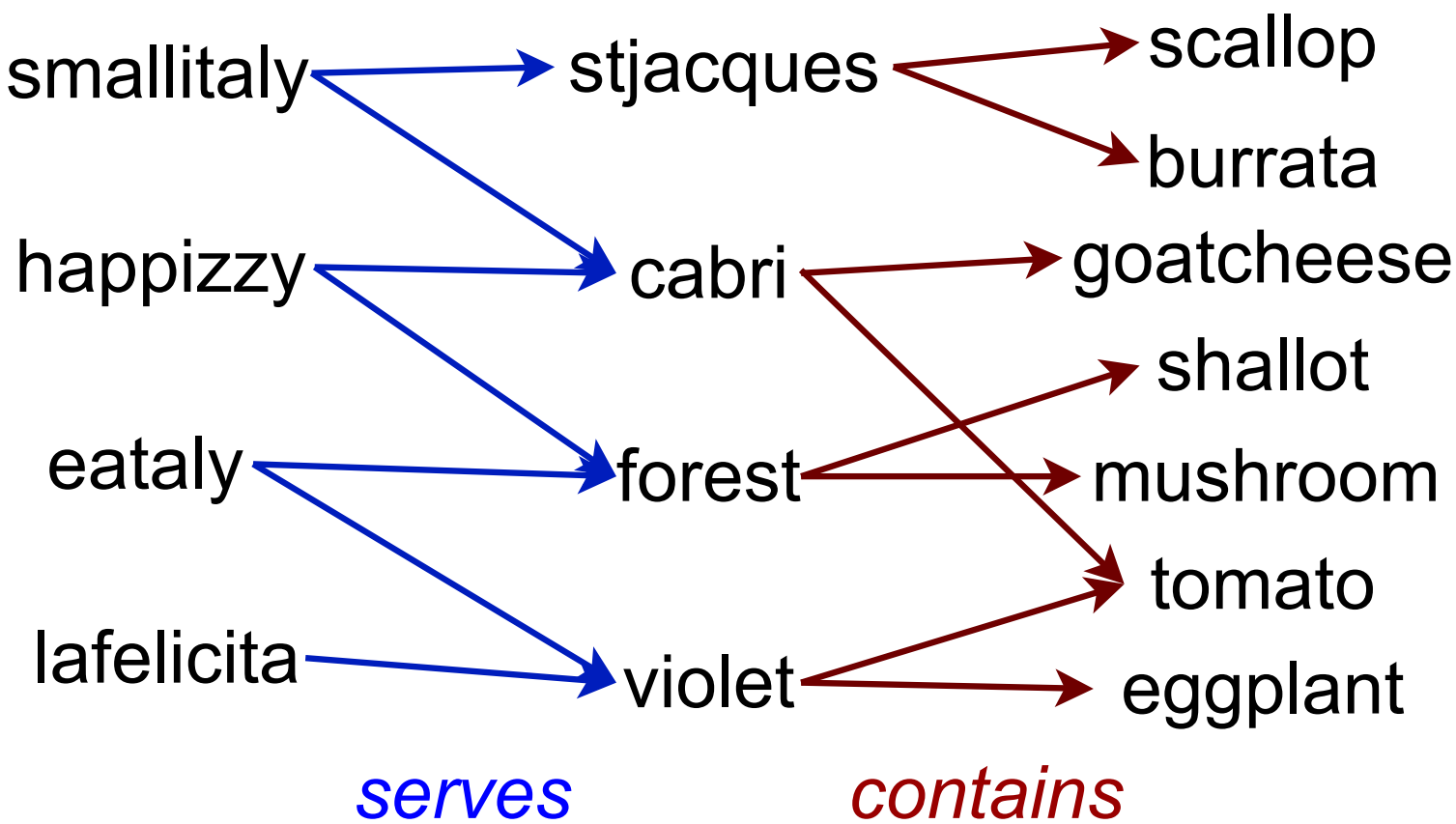
Pizza	*IdCabri*	*IdForest*	*IdViolet*	*IdStjacques*
cabri	×			
forest		×		
violet			×	
stjacques			×	

Ingredient	*vege*	*vegan*	*spring*	*summer*	*autumn*
goatcheese	×			×	
burrata	×		×		
scallop			×		
tomato	×	×		×	
shallot	×	×			×
mushroom	×	×			×
eggplant	×	×		×	

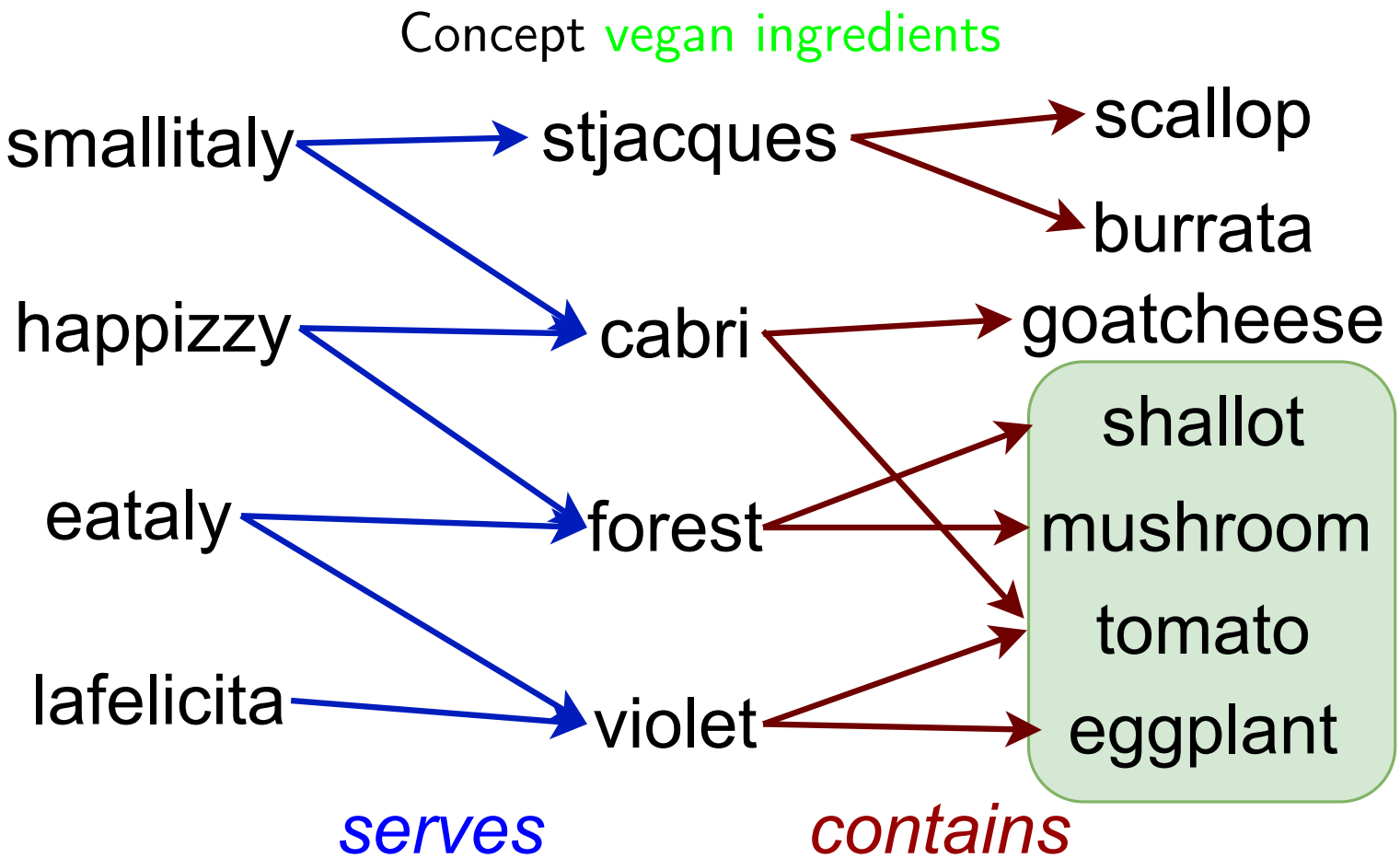
contains	*goatcheese*	*burrata*	*scallop*	*tomato*	*shallot*	*mushroom*	*eggplant*
cabri	x			x			
forest					x	x	
violet				x			x
stjacques		x	x				

serves	*cabri*	*forest*	*violet*	*stjacques*
happizzy	x	x		
eataly		x	x	
lafelicita			x	
smallitaly	x			x

Concept building



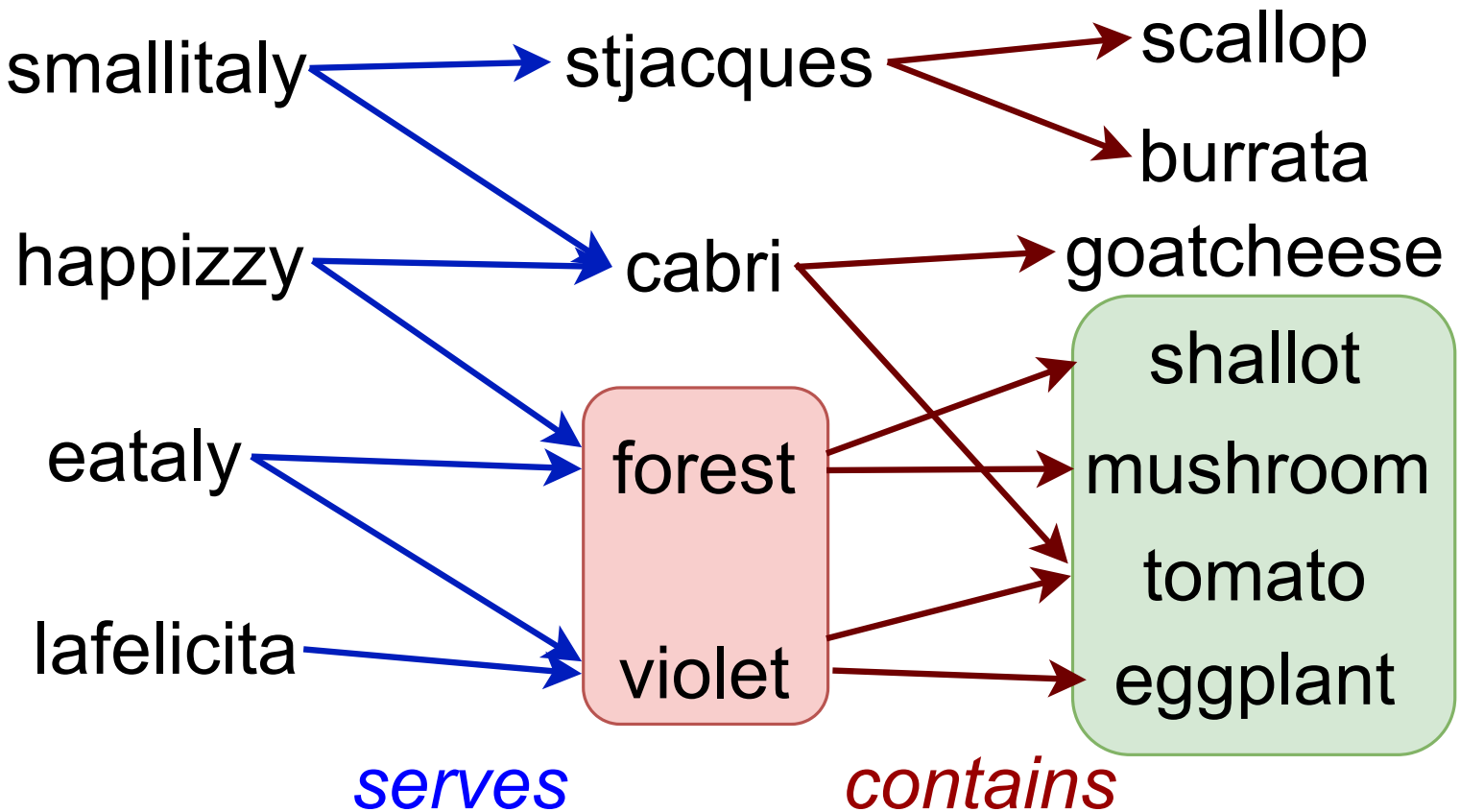
Concept building



Concept building

Concept pizzas that have all their ingredients in concept vegan ingredients

forest and violet share: $\exists\forall contains\ (Vegan)$

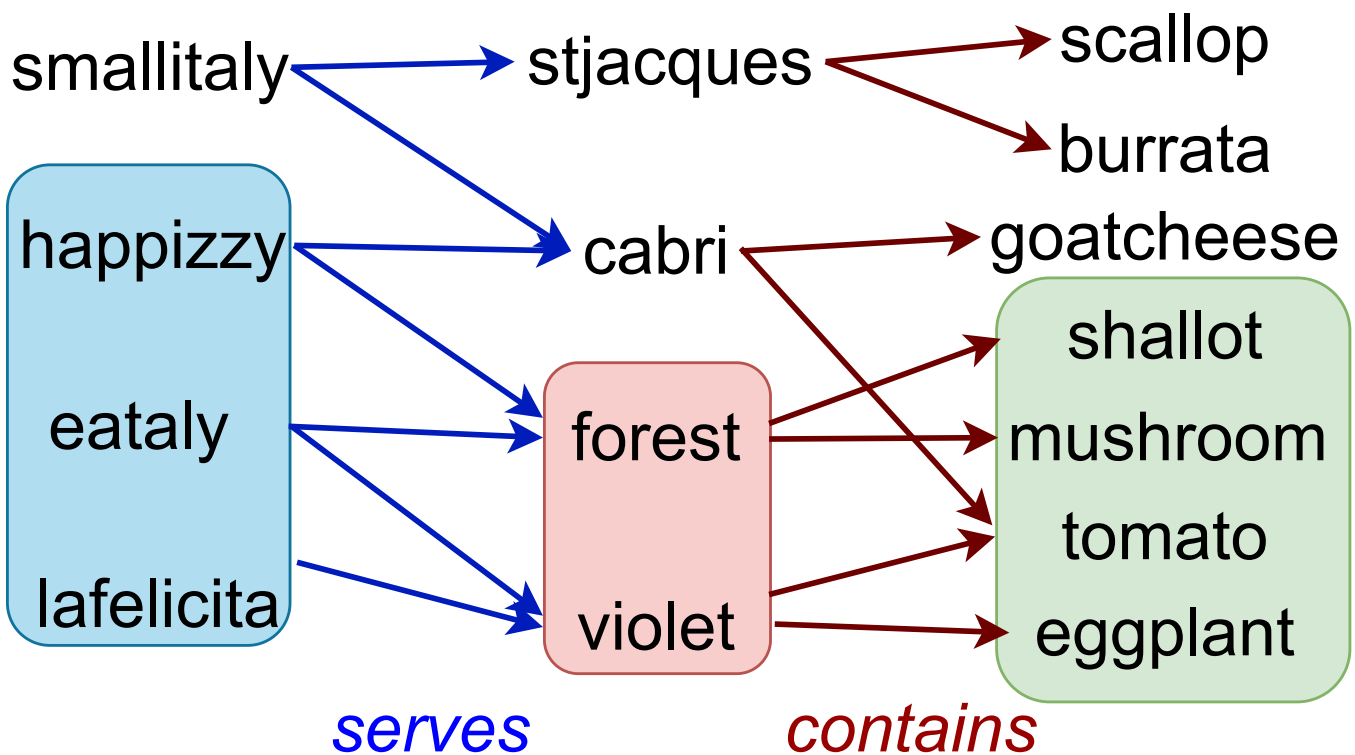


Concept building

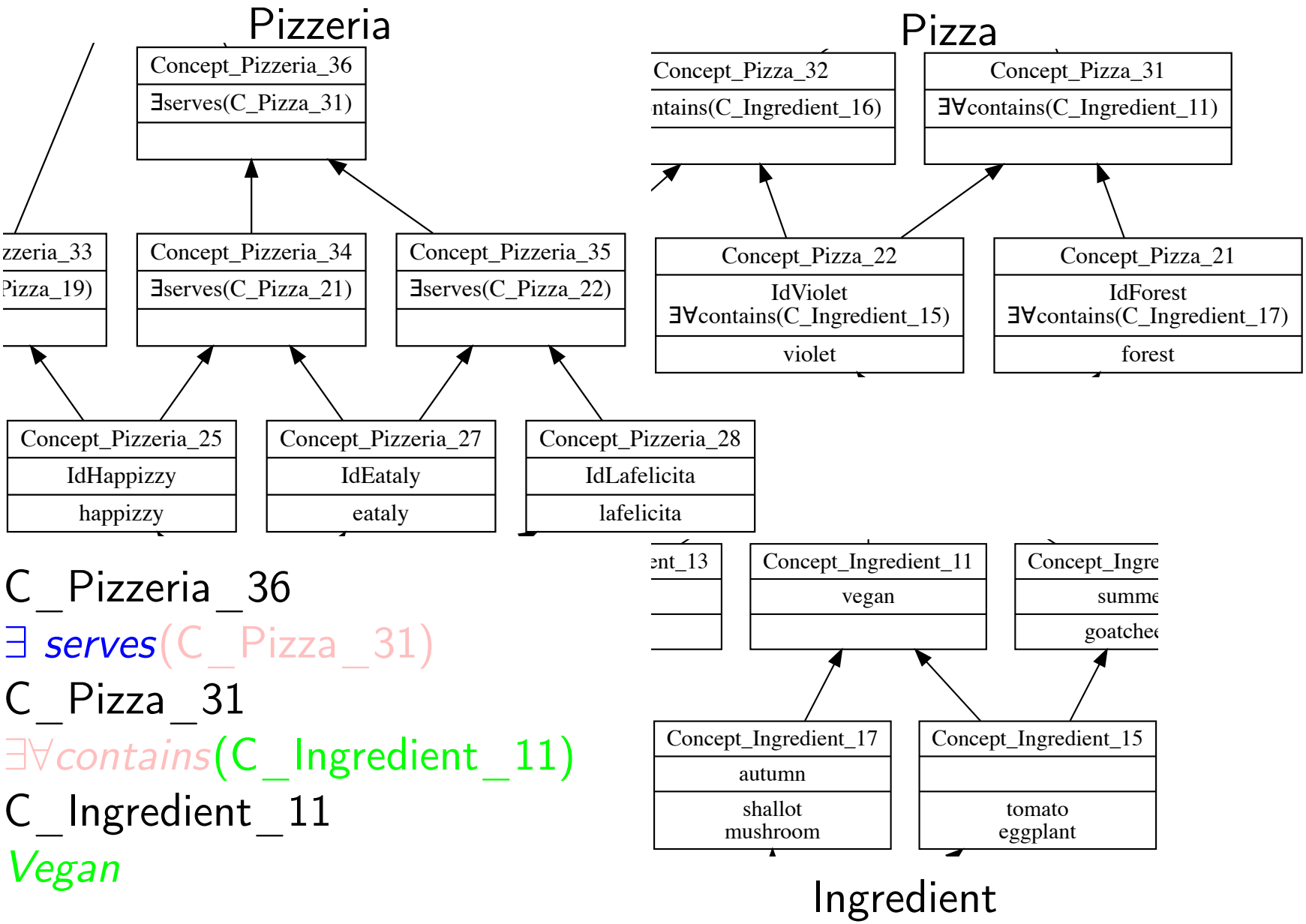
Concept pizzerias that have at least one pizza in concept pizzas that have all their ingredients in concept vegan ingredients

happizzy, eataly and lafelicita share:

$\exists$ *serves* ($\exists\forall$*contains* (*Vegan*))



Interconnected concept lattices



C_Pizzeria_36

∃ *serves*(C_Pizza_31)

C_Pizza_31

∃∀*contains*(C_Ingredient_11)

C_Ingredient_11

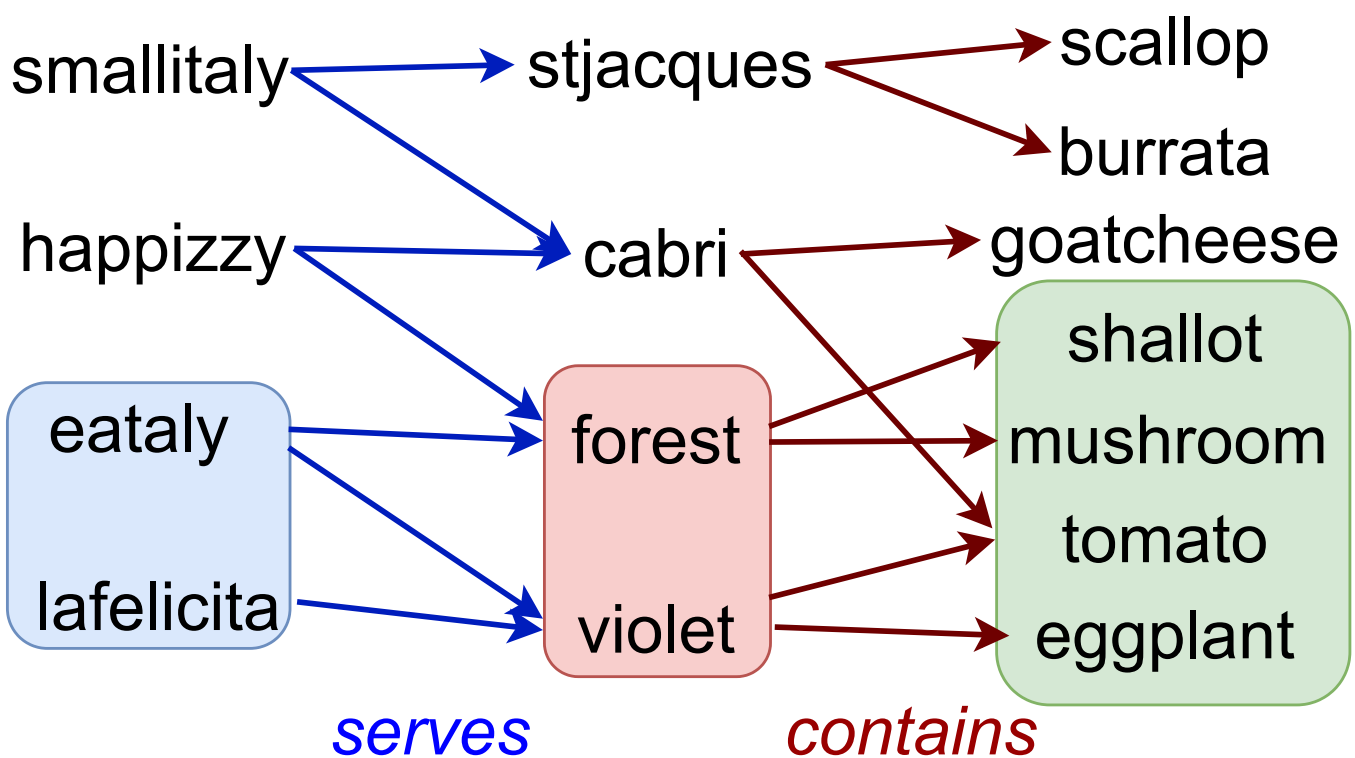
Vegan

Concept building

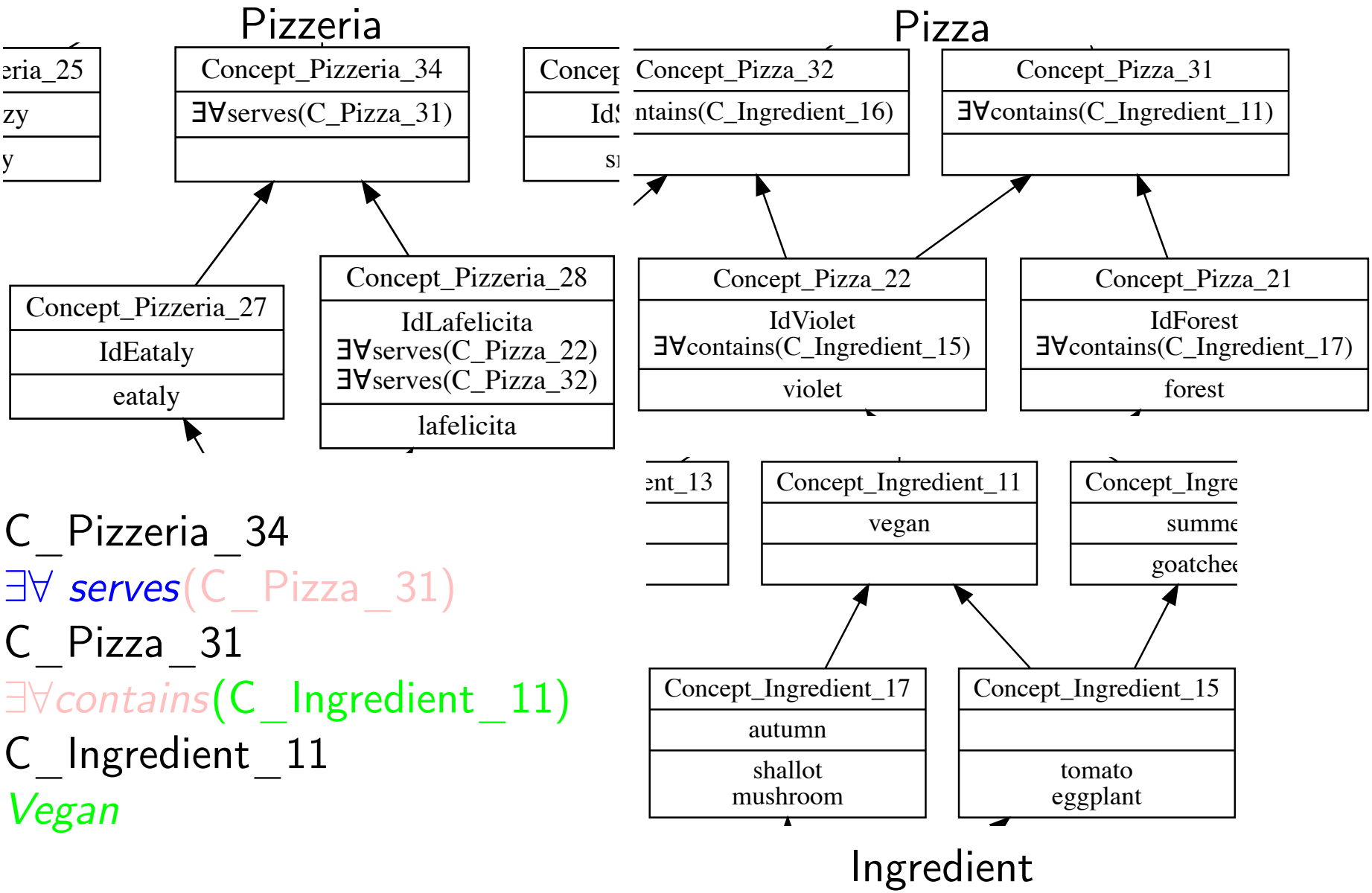
Concept pizzerias that have all their pizzas in concept pizzas that have all their ingredients in concept vegan ingredients

eataly and lafelicita share:

$\exists\forall$ *serves* ($\exists\forall$*contains* (*Vegan*))



Interconnected concept lattices



C_Pizzeria_34

∃∀ *serves*(C_Pizza_31)

C_Pizza_31

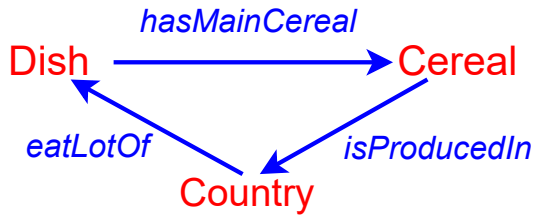
∃∀*contains*(C_Ingredient_11)

C_Ingredient_11

Vegan

Aspects of RCA

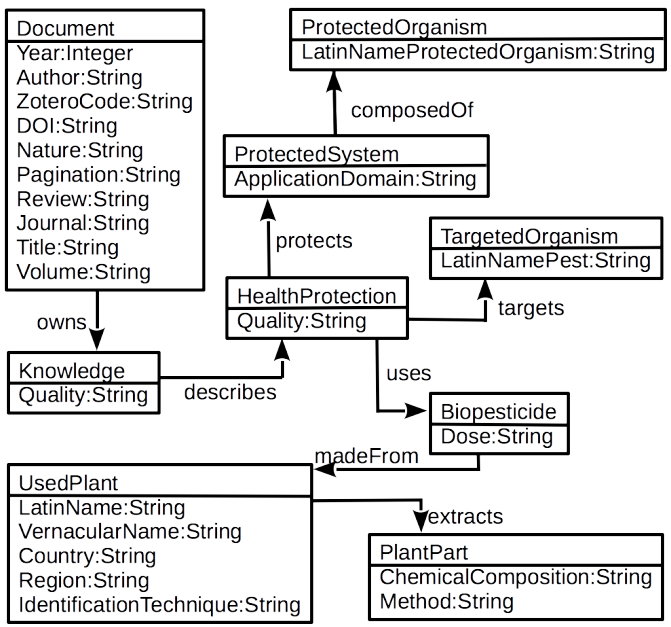
Cyclic-models



A variety of quantifiers

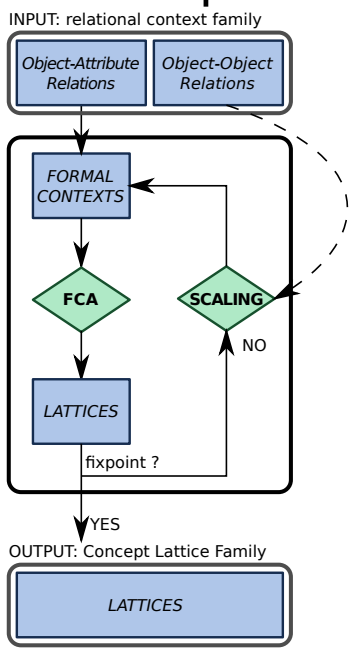
Operator	Attribute form	Condition
Universal (narrow)	$\forall\ r.c$	$r(o) \subseteq Ext(c)$
Universal strict	$\forall\exists\ r.c$	$r(o) \subseteq Ext(c)$ and $r(o) \neq \emptyset$
Universal-percent	$\forall\exists\_\geq n\%\ r.c$	$\lvert r(o) \cap Extent(C)\rvert \geq n\lvert r(o)\rvert/100)$
Covers	$\supseteq\ r.c$	$r(o) \supseteq Ext(c)$
Covers-percent	$\supseteq\ \_\geq n\%\ r.c$	$\lvert r(o) \cap Extent(C)\rvert \geq n\lvert Extent(C)\rvert/100)$
Existential (wide)	$\exists\ r.c$	$r(o) \cap Ext(c) \neq \emptyset$
Universal strict	$\forall\exists\ r.c$	$r(o) \subseteq Ext(c)$ *and* $r(o) \neq \emptyset$
Qualif. card. restriction	$\geq\ n\ r.c$	$r(o) \subseteq Ext(c)$ *and* $\lvert r(o)\rvert \geq\ n$
Card. restriction	$\geq\ n\ r.\top_{\mathcal{L}}$	$\lvert r(o)\rvert \geq\ n$

Complex-models



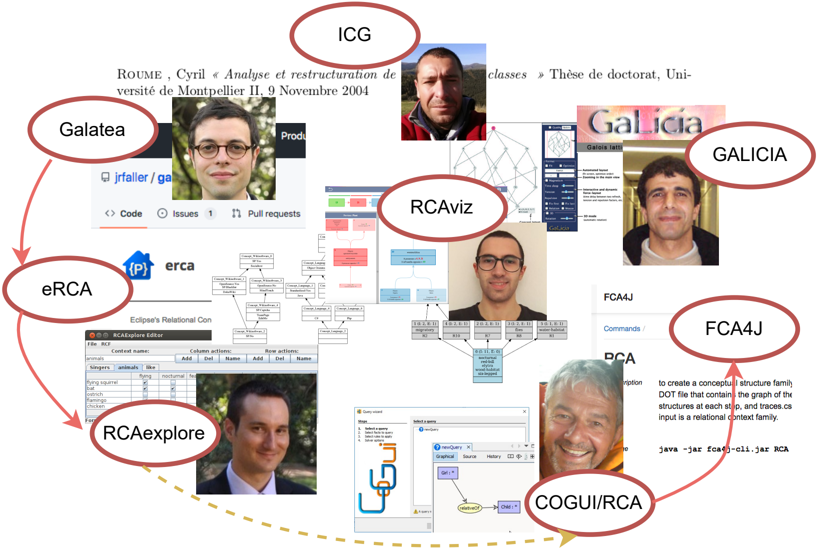
Credits P. Martin

Iterative-process



Credits X. Dolques

RCA tools


ICG
Roume , Cyril « Analyse et restructuration de … classes » Thèse de doctorat, Université de Montpellier II, 9 Novembre 2004
Galatea
jrfaller / ga
Code
Issues 1
Pull requests
eRCA
erca
Eclipse's Relational Con
RCAExplore Editor
File RCF
Context name:
Column actions:
Row actions:
animals
Add
Del
Name
Singers
like
flying
nocturnal
flying squirrel
bat
ostrich
flamingo
chicken
RCAexplore
RCAviz
migratory
flies
water-habitat
nocturnal
red-bill
elytra
wood-habitat
six-legged
Galicia
Galois latti
GALICIA
FCA4J
Commands
RCA
to create a conceptu l structure famil
DOT file that contain the graph of the
structures at each st p, and traces.cs
input is a relational context family.
java -jar fca4 j-cli.jar RCA
Query wizard
Steps
Select a query
newQuery
Graphical
Source
History
Girl : *
relativeOf
Child : *
COGUI/RCA

Application domains

- Environment (Le Ber+, Martin+, Miralles+)
- Health (Wajnberg+)
- Ontologies (Rouane-Hacène+), Link key extraction (Atencia+)
- Software engineering
  - UML class model, use case model normalization (Huchard+, Dolques+)
  - Design defects refactoring (Moha+)
  - Model transformation pattern extraction (Dolques+)
  - Web service, component classification (Azmeh+, Urtado/Vauttier+)
  - Multiple variability modelling (Galasso+, Waffo Kouhoué+), Feature mining or location (Al-Msie'deen, Eyal-Salman, Seriai+, Hlad+),
- Industry (Wajnberg+)
- Legal documents (Mimouni+)

Agenda

Modeling for RCA

- Naive view: A model is provided with data which can be used
- Practice is more complex, e.g.
  - Associations: bidirectional or N-ary
  - Particular relations: E.g. *is-a*, *instance-of*, *contains*
  - The analyst may want to focus on specific parts of the data
  - Data may be provided without a model (e.g. textual data)

$\Rightarrow$ The RCF has to be built by considering a transformation of the initial model, or even created from scratch

The spirit is similar to *scaling* of FCA that transforms non-Boolean attributes into Boolean ones.
Here the needed modeling addresses models, instances (objects) and values.

Rationalizing RCF design

- Recurring situations
- Capitalizing them would accelerate future applications
- Design Pattern approach (C. Alexander, GOF: E. Gamma, J. Vlissides, R. Johnson, R. Helm)
  - Problem
  - Recurring solution
  - Typical example
  - Variants and discussion

Design Pattern Separate/Gather Viewpoints

Problem

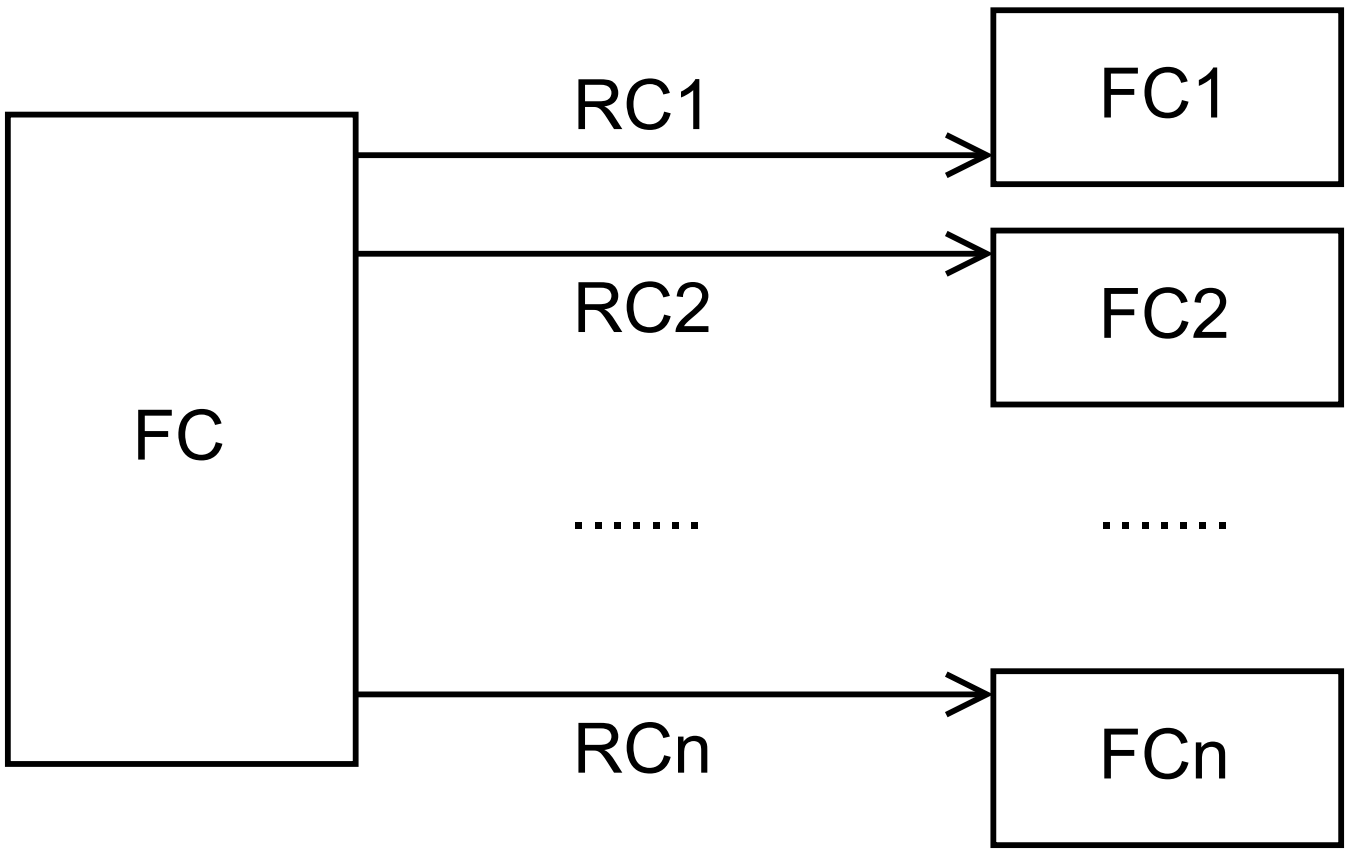
- Objects are described by Boolean attributes that can be put into different categories; or initial multi-valued attributes have been transformed into Boolean attributes
- Each attribute category is a view or a concern
- It is relevant to analyze objects through the perpective of:
  - a single view
  - several views (including all)

Focus on Separate/Gather Viewpoints

Solution

- Formal contexts (FC)
  - One FC for the initial objects
    - FC objects are the initial objects, FC attributes are initial object identifiers, or other description, or none
  - One FC_i for each view $i, 1 \leq i \leq n$, one per attribute category
    - FC_i objects are views on the initial objects, FC_i attributes are initial Boolean attributes of one view
- Relational contexts (RC)
  - For each $i, 1 \leq i \leq n$, one RC_i connecting an object of FC to its corresponding view object in an FC_i

RCF multi-views model



Focus on Separate Viewpoints: an application

Assist Feature location in Software Products Lines
(Hlad, Lemoine, Huchard, Seriai, GPCE 2021)

- Migrate a similar software family into a software product line
- Feature location: find the software artefacts (E.g. code) corresponding to a feature
- Principle: if a group of artefacts and a group of features are shared by products, this suggests that these features may be coded through these artefacts
- Challenges:
  1. have a sufficient product number to discriminate and associate a single feature to a group of artefacts
  2. some pieces of code do not correspond to a feature but suggest an interaction between features, or missing features in the product description

Focus on Separate Viewpoints: Vehicles

Input

Features and code of similar products

```
[MOVE, SECU,PROGCONT,WATER]
public class Vehicle{
 public void move_(){..}
 public void passControl_(){..}
 public void float_(){..}
} //Oceanis30

[MOVE, SECU,PROGCONT,AIR]
public class Vehicle{
 public void move_(){..}
 public void passControl_(){..}
 public void fly_(){..}
} //A380
```

Output

Annotated SPL code

```
public class Vehicle{
 //#if MOVE & SECU & PROGCONT
 public void move_(){...}
 public void passControl_(){...}
 //#endif
 //#if WATER
  public void float_(){...}
 //#endif
 //#if AIR
  public void fly_(){...}
 //#endif
}
```

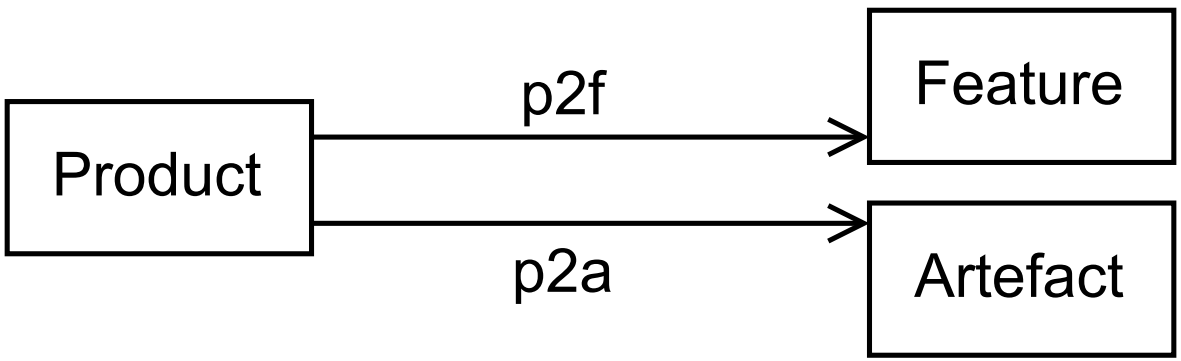
Products derived according selected features

Focus on Separate Viewpoints: an application

Assist Feature location in Software Products Lines
(Hlad, Lemoine, Huchard, Seriai, GPCE 2021)

- Separately group features and artefacts to extract:
  - Feature variability (variability model)
  - Artefact variability (potentially reusable code pieces)
  - Traceability between propositional logic formula on features vs. code artefacts
    - Identification of apparition context of artefacts in terms of positive vs. negative features followed by formula reduction
    - Generation of code annotations `#if FORMULA-ON-FEATURES`
- RCF
  - Objects: similar products
  - View 1: products described by features
  - View 2: products described by software artefacts

RCF product model



Focus on Separate Viewpoints: Vehicles

The two views: by feature and by artefacts

Feature	MOVE	AIR	WATER	GROUND	SECU_	PROG_CONT_	UNAN_INSP_
A380Feat	x	x			x	x	x
Oceanis30Feat	x		x		x	x	
AkoyaFeat	x	x	x		x	x	
SqubaFeat	x		x	x	x	x	
TringaT650Feat	x		x	x	x	x	
TwingoFeat	x			x	x	x	

Artefact	*move*	*ride*	*honk*	*float*	*fly*	*takeoff*	*land*	*ditch*	*dive*	*raisewheel_/*	*reportinsp_/*	*passcontrol*
A380Art	x	x			x	x	x				x	x
Oceanis30Art	x			x								x
AkoyaArt	x			x	x	x	x	x				x
SqubaArt	x	x	x	x					x			x
TringaT650Art	x	x	x	x						x		x
TwingoArt	x	x	x									x

Focus on Separate Viewpoints: Vehicles

Formal Context `Product` ⟶

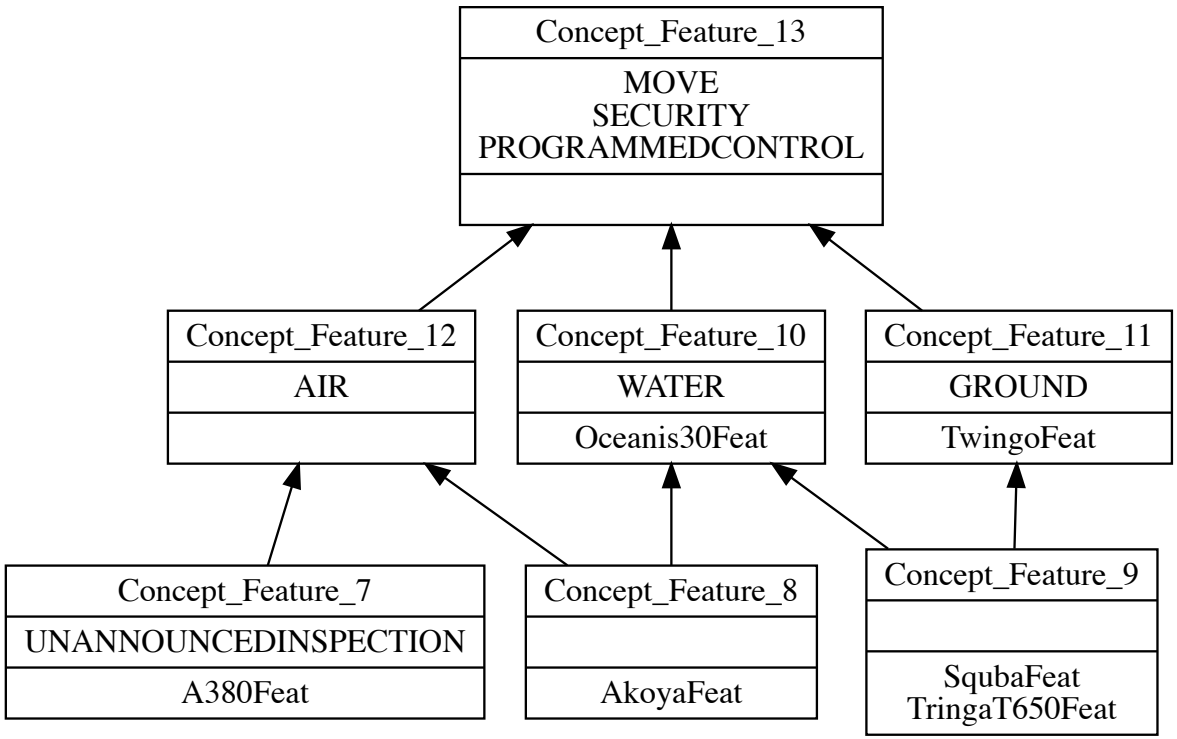
Relational contexts connecting products
to the feature view or to the artefact view ↓

Product
A380
Oceanis30
Akoya
Squba
TringaT650
Twingo

p2f	A380Feat	Oceanis30Feat	AkoyaFeat	SqubaFeat	TringaT650Feat	TwingoFeat
A380	x					
Oceanis30		x				
Akoya			x			
Squba				x		
TringaT650					x	
Twingo						x

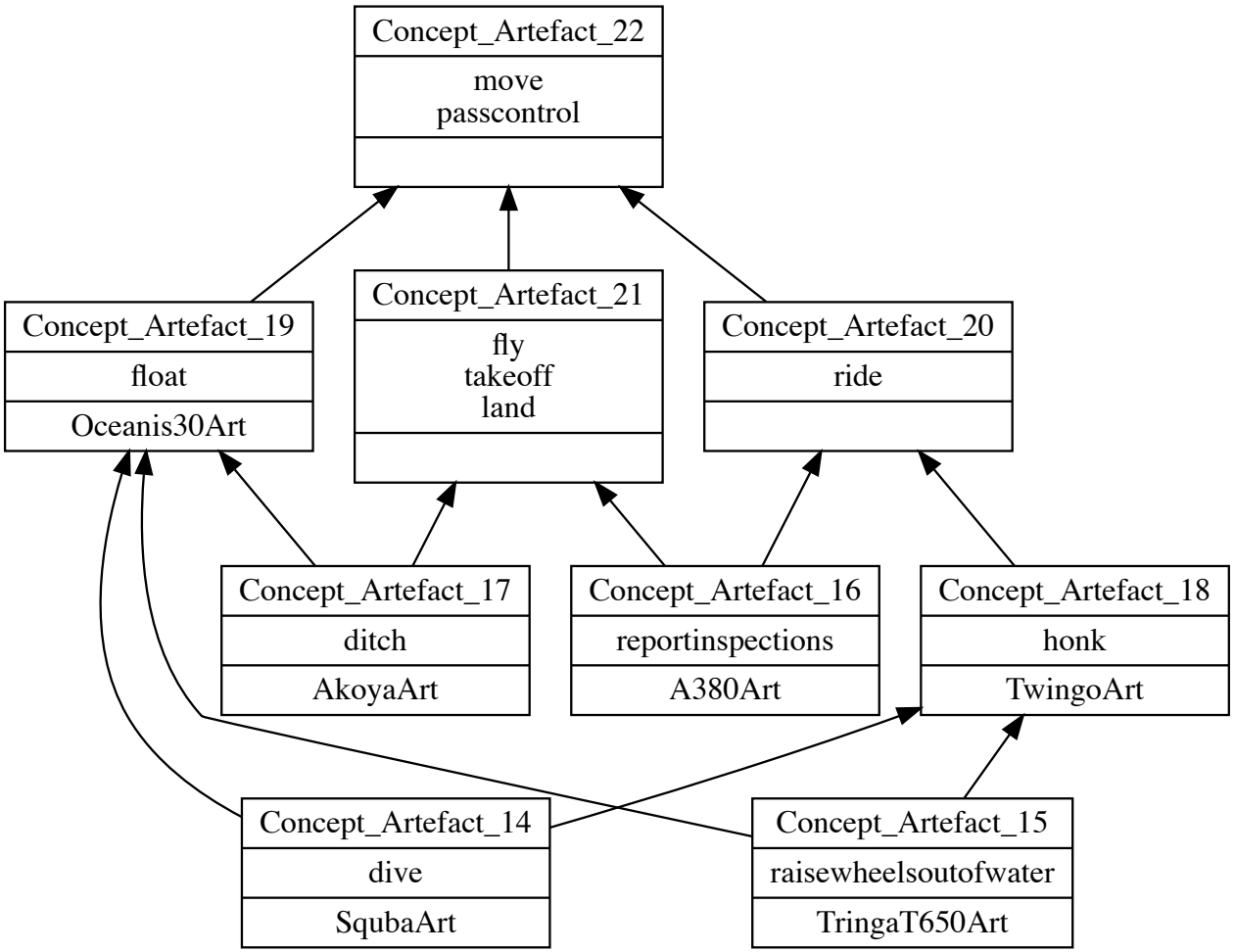
p2a	A380Art	Oceanis30Art	AkoyaArt	SqubaArt	TringaT650Art	TwingoArt
A380	x					
Oceanis30		x				
Akoya			x			
Squba				x		
TringaT650					x	
Twingo						x

Feature view



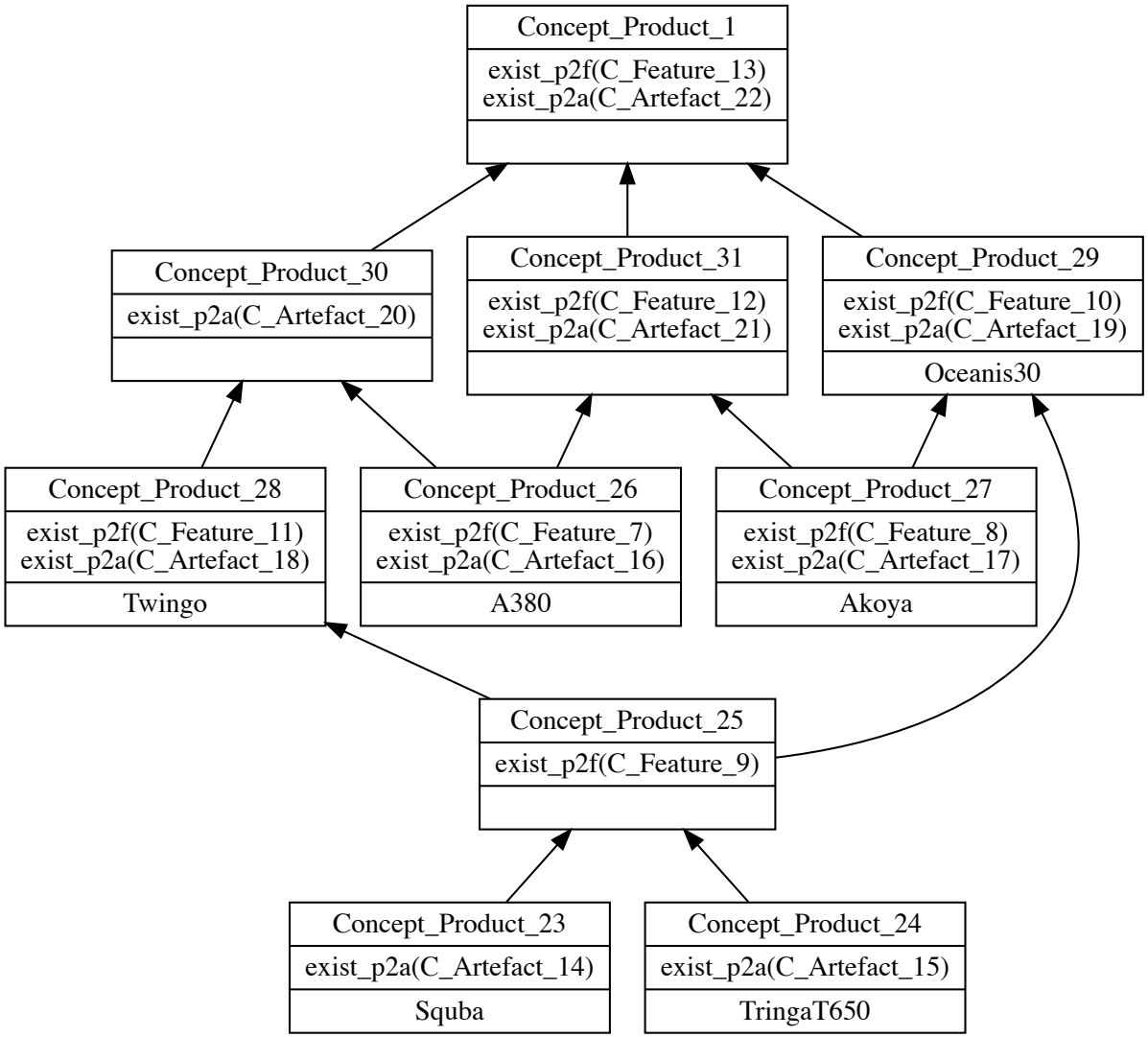
May be used to derive a feature model: *J. Galasso-Carbonnel et al. Modelling equivalence classes of feature models with concept lattices to assist their extraction from product descriptions. J. Syst. Softw. 152: 1-23 (2019)*

Artefact view



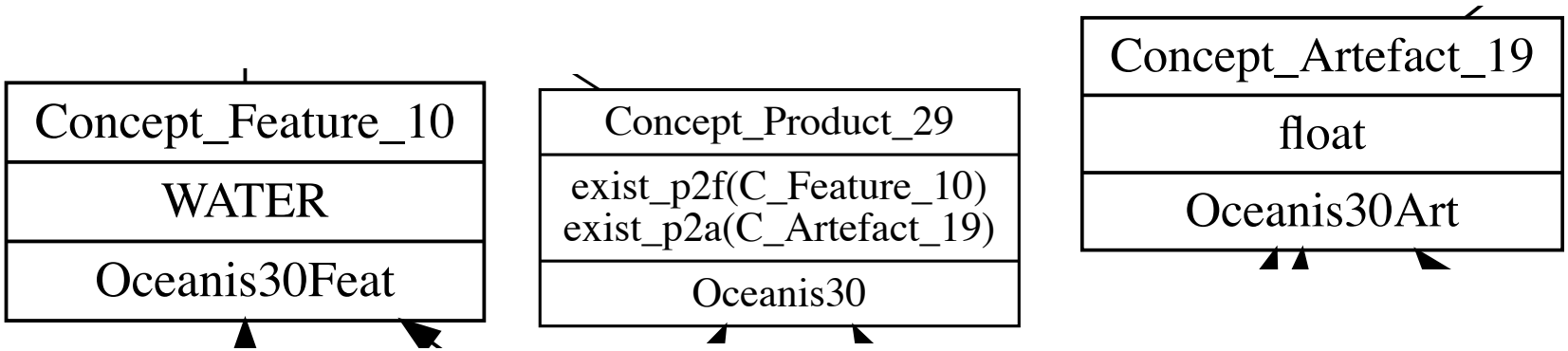
May be used to suggest reusable code pieces: E.g. methods `fly`, `takeoff`, `land` can be grouped in a component `AirVehicle`

Gathered product views



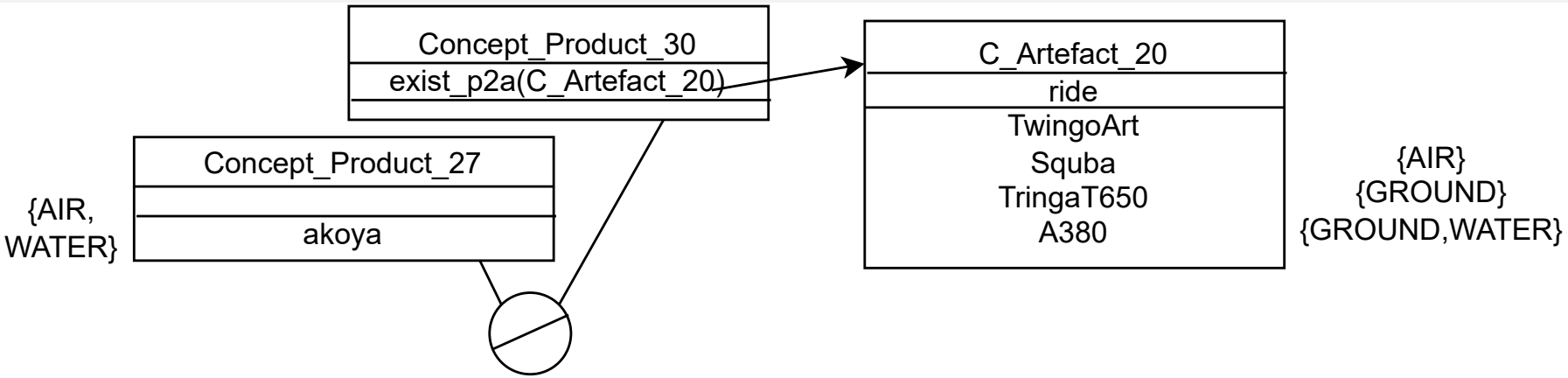
Used to find annotations and create the annotated code of the SPL

Gathered product views: annotation extraction




```
public class Vehicle{
...
 //#if WATER
  public void float_(){...}
 //#endif
...
}
```

Gathered product views: annotation extraction




```
public class Vehicle{
 //#if (AIR || GROUND) && ! ( AIR  &&  WATER )
  public void ride_(){...}
 //#endif
...
}
```

Extent(C_Prod._30)={Twingo, Squba, TringaT650, A380}
Positive part from C_Art._20: (AIR || GROUND)

Extent(C_Prod._30) ∩ Extent(C_Prod._27) = ∅
Extent(C_Prod._27 = {Akoya}
Negative part from C_Prod._27 = ! (AIR && WATER)

Focus on Separate Viewpoints: an application

For more details

- Identification of artefacts at the instruction-level
- Reduction of annotation formulas (e.g. by factorization, using implications from the AOC-poset, etc.)
- Tools: CLEF (J. Galasso), IsiSPL (N. Hlad, B. Lemoine)
- Experimentations
  - FakePrinterSPL, DrawSPL, ArgoUML, NotePad, GameOfLife, Elevator
  - évaluation de la régénération correcte des produits et réduction des formules en nombre ($\sim$-30%) et taille des formules ($\sim$-60% de features)

Focus on Separate Viewpoints: an application

For more details

- Readings
  - N. Hlad, thèse de doctorat, *IsiSPL : un processus automatisé pour faciliter l'ingénierie des lignes de produits logiciels selon une stratégie d'adoption industrielle réactive ou extractive*, Univ. Montpellier 2022, ISIA
  - B. Lemoine, thèse de Master, *Localisation de caractéristiques dans le cadre des lignes de produits logiciels*, Univ. Montpellier 2021, prix IESF-CODIGE
  - N. Hlad et al. *Leveraging relational concept analysis for automated feature location in software product lines.* GPCE 2021: 170-183

Focus on Separate Viewpoints

Another developed application

- Analyzing Visual Accessibility Options in Operating Systems
  - Objects are operating systems (OS); Views are three visual accessibility options categories (contrast, text, zoom)
  - Allows to analyze constraints on these options e.g.: implications, mutual exclusion
  - Objectives: recommendations for a new system version (OS developer), or user assistance in changing versions
  - A. Waffo Kouhoué et al. *Exploring Variability of Visual Accessibility Options in Operating Systems.* Future Internet 13(9): 230 (2021)

Towards a catalog of DPs

Opportunities for other design patterns

- Collapse Specialization — Godin/Mili (classes), Huchard/Miralles+ (Java interfaces, UML), Martin+ (plants)
- N-ary relation reification /split — Martin+ (plants)
- Separate topics — Martin+ (plants), Miralles+ (UML)
- Instances2Model — Dolques+ (Model transformation pattern extraction), B.Seriai+ (RDF)
- Workflow — Azmeh+ (web services workflow classification)
- Query — Azmeh+ (web service replacement in a workflow)
- Temporal — Le Ber+ (hydrological data)
- Correspondence/matching — Dolques+ (Model transformation pattern extraction)

Agenda

Conclusion

- FCA as a symbolic/white-box/structured-data KD method
- RCA, a variant for multi-relational datasets
- Practicing RCA demands a quite complex data preparation
- Recurring situations to be capitalized in Design Patterns

Perspectives

- Pursue the formalization of DPs
- Develop the corresponding interpretation tools
- Analyze effects on extracted knowledge:
  - conceptual structures
  - logical constraints: implication and association rules, mutual exclusion, OR groups, XOR groups
- Develop a methodology integrating DPs usage
- Build bridges with Graph-FCA and Polyadic Concept Analysis

Thank you!

SmartFCA

Supported by the ANR SmartFCA project
Grant ANR-21-CE23-0023 of the French National Research Agency
anr

This reflexion could not have happened without ...