



**HAL**  
open science

## RCAviz: Exploratory search in multi-relational datasets represented using relational concept analysis

Marianne Huchard, Pierre Martin, Emile Muller, Pascal Poncelet, Vincent Raveneau, Arnaud Sallaberry

### ► To cite this version:

Marianne Huchard, Pierre Martin, Emile Muller, Pascal Poncelet, Vincent Raveneau, et al.. RCAviz: Exploratory search in multi-relational datasets represented using relational concept analysis. International Journal of Approximate Reasoning, 2024, 166, pp.109123. 10.1016/j.ijar.2024.109123 . hal-04421960

**HAL Id: hal-04421960**

**<https://hal.science/hal-04421960v1>**

Submitted on 22 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

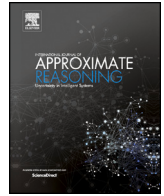
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

## International Journal of Approximate Reasoning

journal homepage: [www.elsevier.com/locate/ijar](http://www.elsevier.com/locate/ijar)

## RCAviz: Exploratory search in multi-relational datasets represented using relational concept analysis

Marianne Huchard<sup>a,\*</sup>, Pierre Martin<sup>b,c,\*</sup>, Emile Muller<sup>a</sup>, Pascal Poncelet<sup>a</sup>, Vincent Raveneau<sup>a</sup>, Arnaud Sallaberry<sup>a,d,\*</sup>

<sup>a</sup> LIRMM, Univ. Montpellier, CNRS, Montpellier, France

<sup>b</sup> CIRAD, UPR AIDA, F-34398 Montpellier, France

<sup>c</sup> AIDA, Univ. Montpellier, CIRAD, Montpellier, France

<sup>d</sup> AMIS, Univ. Paul-Valéry Montpellier 3, Montpellier, France

### ARTICLE INFO

#### Keywords:

Exploratory search  
Formal concept analysis  
Relational concept analysis  
Visual analytics

### ABSTRACT

The conceptual structures built with Formal Concept Analysis (FCA) and its extensions are appropriate constructs for supporting Exploratory Search (ES). FCA indeed classifies a set of objects described by Boolean attributes in a concept lattice which is prone to (intra-lattice) navigation. Relational Concept Analysis (RCA), for its part, classifies several sets of objects connected through multiple binary relationships by using logical operators (quantifiers) which can be approximate. The output is a set of interconnected concept lattices, thus adding inter-lattice navigation opportunities. In this paper, we describe the web platform RCAviz, which aims to support such intra- and inter-lattice navigation. The user can select a subset of objects and attributes as a starting point for navigation. Then RCAviz shows the associated concept and its close intra- and inter-lattice neighbors. The user can access to the objects and attributes introduced and inherited in a concept. They then can navigate, i.e. zoom and pan the current view, and move from one concept to another. Additional views show the previous and the next conceptual structures, as well as an history which allows the user to browse its navigation. A navigation example is shown on a real dataset to illustrate the potential of RCAviz for ES.

### 1. Introduction

Discovering useful knowledge from data requires more than a simple lookup. To this end, Exploratory Search (ES) combines the problem context and a search process to solve complex problems and enhance mental capacities [1]. As each of the four phases of the search process, i.e. problem identification, definition, resolution, and statement, includes uncertainty, the user is faced to an unpredictable non-linear ES path [2]. To reduce uncertainty in some of the phases, various ES approaches were developed in relation to the problem context (e.g. [3]). Eleven characteristics describing an ES approach were identified by [4], i.e. an evolving search process, several one-off pinpoint searches, an evolving information need, multiple targets/goals of search, multiple possible answers, not an expected exact answer, a serendipitous attitude, an open-ended search activity which can occur over time, an anomalous state of knowledge and an ill-structured context of search or goals, multifaceted, and a fluctuating uncertainty. This diversity of characteristics highlights the complexity to tackle ES.

\* Corresponding authors.

E-mail addresses: [marianne.huchard@lirmm.fr](mailto:marianne.huchard@lirmm.fr) (M. Huchard), [pierre.martin@cirad.fr](mailto:pierre.martin@cirad.fr) (P. Martin), [arnaud.sallaberry@lirmm.fr](mailto:arnaud.sallaberry@lirmm.fr) (A. Sallaberry).

<https://doi.org/10.1016/j.ijar.2024.109123>

Received 30 January 2023; Received in revised form 23 November 2023; Accepted 10 January 2024

Available online 17 January 2024

0888-613X/© 2024 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

The literature presents diverse exploration methods. One of them consists in formulating the problem as a query and apply it to dataset. Some improvements of this method consisted in enabling query refinement [5], the combination of queries [6], query suggestion based on search results [7]. To address knowledge graph exploration, three approaches were identified by [8], i.e. node search [9], structure search [10], and query building and suggestions. For the latter approach, the query can be expressed using SPARQL language [11], and then the partitioned results be qualified using different quality metrics [12]. The construction of a query graph and the computation of the semantic similarity of the query graph using approximate aggregate queries were proposed by [13]. By focusing the exploration on the initial interest of the user, these methods hardly consider the evolution of the user needs during the search session. Another method is based on the modeling of the interaction with the user during the exploration, using reinforcement learning [14]. Interest of the latter is personalizing the search and the ability to project future user intents. These methods enable ES to be conducted within the dataset as it is provided in its original form.

Our approach consists in classifying the data using an unsupervised method, i.e. Relational Concept Analysis (RCA) [15], prior conducting ES. The hypothesis is that the relative intrinsic data structure can provide alternative answers to the query when the user is faced with a lack of data. RCA is an extension of Formal Concept Analysis (FCA). FCA aims at classifying a set of objects described by Boolean attributes as a concept lattice [16]. FCA and its various extensions enable the extraction of frequent patterns, attribute implications, mutual exclusions within attribute groups, and object similarities. RCA supports multiple binary relations between several sets of objects. As output, RCA builds conceptual structures (one per object set) interconnected by relational attributes, that embed the inter-object relations provided as input. The use of quantifiers other than existential, e.g. universal or with percentages [17], enlarges the groupings and thus facilitates ES using approximate data groupings. Another extension of FCA, i.e. Graph-FCA devoted to large knowledge graphs, has previously considered ES [18].

The objective of this paper<sup>1</sup> is to present RCAviz, a web application designed to conduct an ES within a multi-relational dataset represented using RCA. Section 2 introduces FCA and RCA. It also describes the need for a user to navigate a set of lattices built using RCA. Section 3 reviews the literature on the FCA/RCA visualization tools, and their ability to conduct ES. RCAviz and its functionalities are presented in Section 4. An example of ES using RCAviz is presented in Section 5. Section 6 concludes on this work.

## 2. Background and motivation

In this section, Subsection 2.1 shortly introduces main notions of FCA and RCA, as well as the construction of a conceptual structure. Then, Subsection 2.2 outlines the navigation requirements of a user in a RCA conceptual structure.

### 2.1. RCA in a nutshell

**FCA.** FCA [16] is a versatile and powerful knowledge engineering method rooted in partial order and lattice theory [21–23]. It has been successfully applied in a variety of applications for many years, as outlined in several surveys [24,25]. FCA is more particularly dedicated to deal with complex structured data, including a significant amount of categorical (symbolic) data. Provided that some effort is made, its procedures and results can be interpreted and explained to data scientists and domain experts. The basic notion is the Formal Context (FCt), which is a 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  $a$ . Table 1 shows a FCt where ingredients (objects) are described by diets and season (attributes) to recommend their consumption. E.g. goatcheese can be included in vegetarian diet and is best in summer.

From the Formal Context, FCA algorithms extract Formal Concepts (FCo). A FCo is a pair  $(Extent, Intent)$ , where  $Extent$  is a maximal set of owner objects and  $Intent$  is a maximal set of their shared attributes. In Table 1, the pair  $(E_{11} = \{\text{shallot, mushroom, tomato, eggplant}\}, I_{11} = \{\text{vege, vegan}\})$  is a concept (denoted Concept\_Ingredient\_11 or CIngredient\_11 for short). Ingredients of  $E_{11}$  own attributes of  $I_{11}$  and reciprocally. Adding any attribute to  $I_{11}$  would reduce the set of owner objects. Adding any object to  $E_{11}$  would reduce the set of shared attributes.

The set of all formal concepts provided with bottom-up extent inclusion or top-down intent inclusion has a lattice structure and is called the Concept Lattice (CL). Diverse simplifications of the lattice can be realized to make it easier to read. In Fig. 1, the Concept Lattice<sup>2</sup> is represented in a simplified manner, exposing only the transitive reduction of the inclusion order. Another applied simplification is shown in Fig. 2. It is based on the following property: when an attribute belongs to the intent of a concept  $C$ , it is (top-down) inherited by all the sub-concepts of  $C$ . Thus it is sufficient to show an attribute in the highest concept having it in its intent. This highest concept is the *introducing* concept of the attribute. The same simplification can be symmetrically applied to the objects, that are bottom-up inherited. In Fig. 2, CIngredient\_11 introduces attribute vegan and inherits attribute vege from CIngredient\_10. CIngredient\_11 does not introduce objects, they are all inherited from sub-concepts CIngredient\_17 (shallot, mushroom) and CIngredient\_15 (tomato, eggplant). This simplified view is adopted in the remainder of the paper, as it is commonly done for the sake of readability.

Specific sub-structures of the Concept Lattice, namely the AOC-poset and the Iceberg lattice are commonly used in applications. The AOC-poset [27] is obtained from the concept lattice by removing the concepts that neither introduce object, nor introduce

<sup>1</sup> This paper is an extended version of [19], that includes new features of RCAviz and a typical usage scenario on real data on plants from the Knomana project [20].

<sup>2</sup> In this paper, figures have been computed thanks to FCA4J (<https://www.lirmm.fr/fca4j/>, accessed: 2023-11, see also [26]) and the obtained DOT files have been improved, e.g. completed or using standardized notations.

**Table 1**  
 Formal Context: objects are ingredients, attributes are diets (vege, vegan), and seasons are the recommended periods to consume these ingredients. The Formal Concept  $C_{Ingredient_{11}} = (E_{11} = \{tomato, shallot, mushroom, eggplant\}, I_{11} = \{vege, vegan\})$  is colored as follows: intent in blue, extent in pink, and relations in purple. (For interpretation of the references to color please refer to the web version of this article.)

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

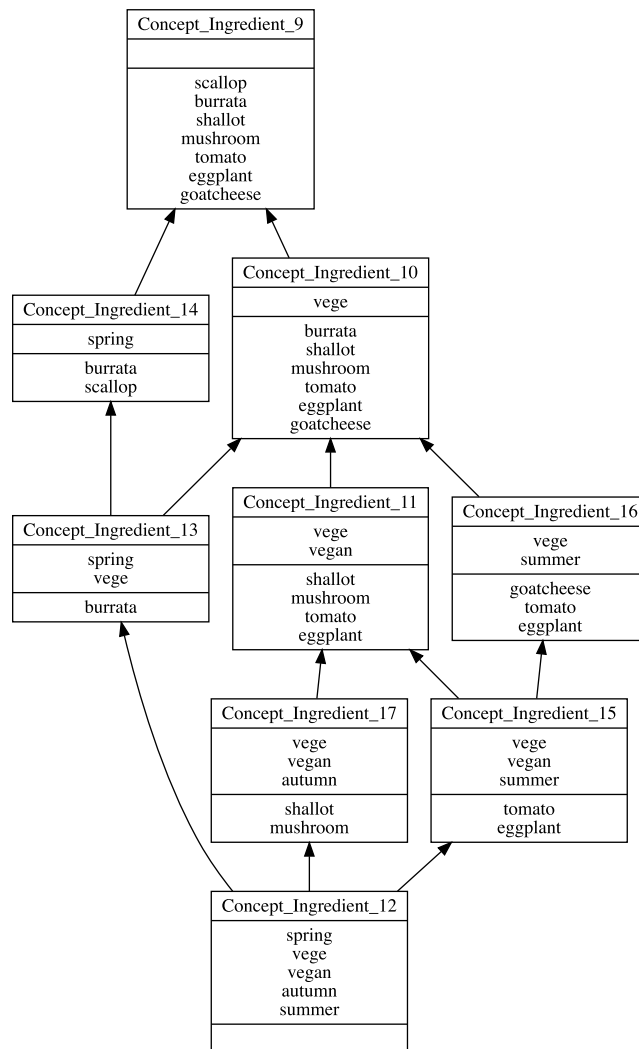


Fig. 1. Concept Lattice extracted from the Formal Context of Table 1 with complete extents and intents.

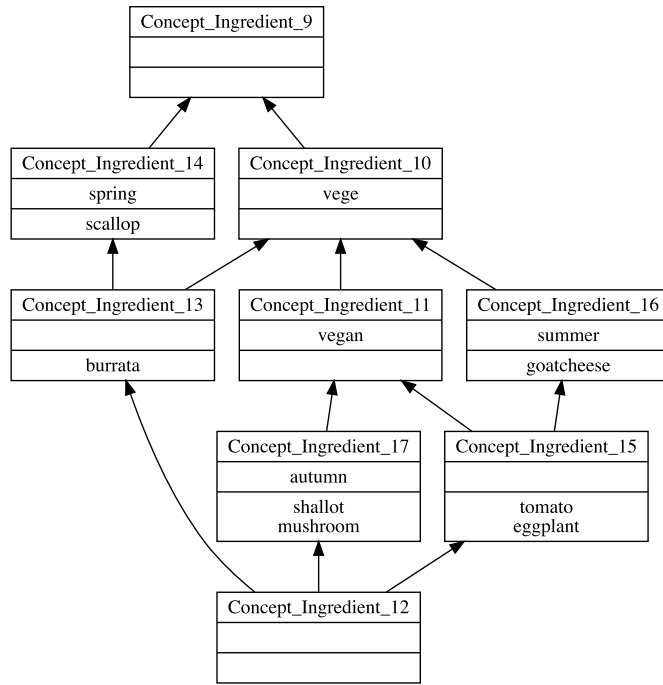


Fig. 2. Concept Lattice extracted from the Formal Context of Table 1 presenting only introduced objects and attributes in concept extents and intents respectively (simplified notation).

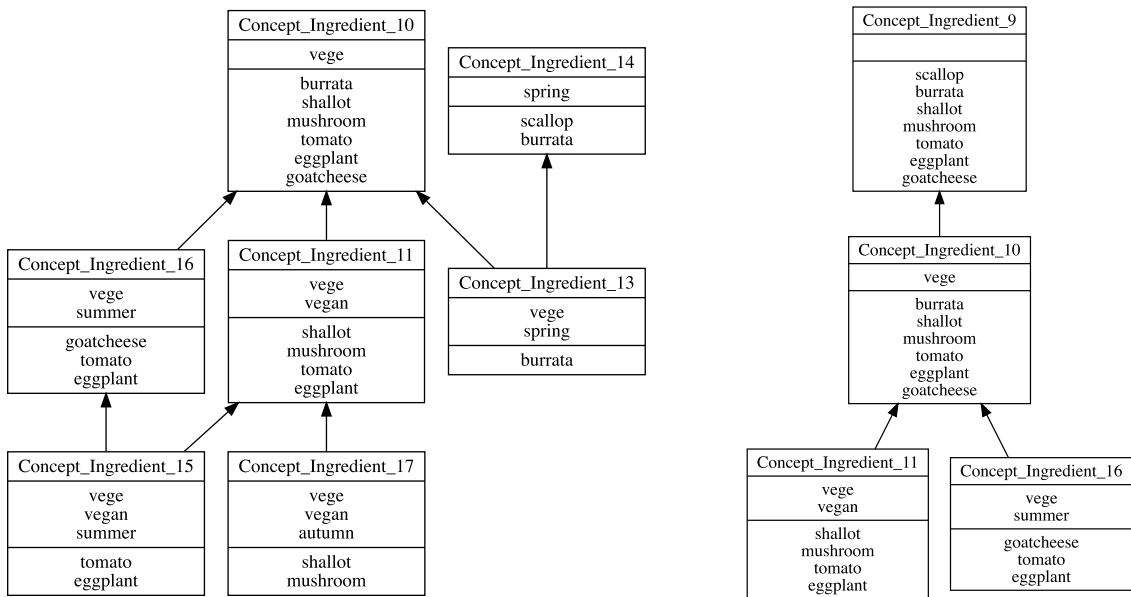


Fig. 3. Sub-orders of the Concept Lattice in the complete notation. In the left-hand side, the AOC-poset contains the introducing concepts. In the right-hand side, the Iceberg lattice contains the frequent concepts with respect to a threshold (here 50%).

attribute (see left-hand side of Fig. 3 for the complete extents and intents, or left-hand side of Fig. 4 in the simplified notation). To obtain the AOC-poset from the Concept Lattice of Fig. 2, we have to remove concepts CIngredient\_9 and CIngredient\_12 which do not introduce neither objects, nor attributes.

In an Iceberg lattice we keep the frequent concepts [28]. Given a threshold  $minsupp \in [0, 1]$ ,  $C$  is a frequent concept if its intent is frequent. FCA4J, the tool used to compute the Iceberg lattice of this paper, keeps the concepts  $C$  such that  $|Extent(C)| \geq [|O| * minsupp]$ . For example, given a fifty percent threshold, only the concepts with at least  $[7 * 0.5] = 3$  objects in their extent are kept. To obtain the Iceberg (50) from the Concept Lattice of Fig. 2, we could remove for example CIngredient\_15 because its extent contains

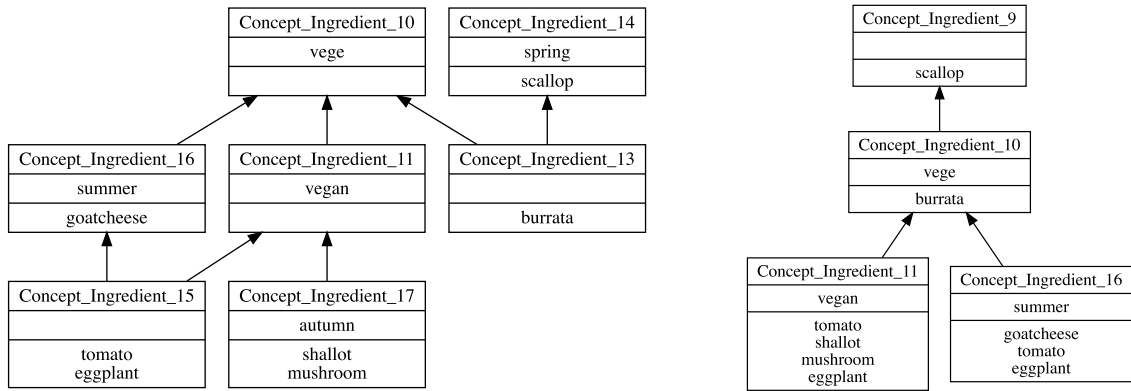


Fig. 4. Sub-orders of the Concept Lattice in the simplified notation. In the left-hand side, the AOC-poset contains the introducing concepts. In the right-hand side, the Iceberg lattice contains the frequent concepts with respect to a threshold (here 50%) [28].

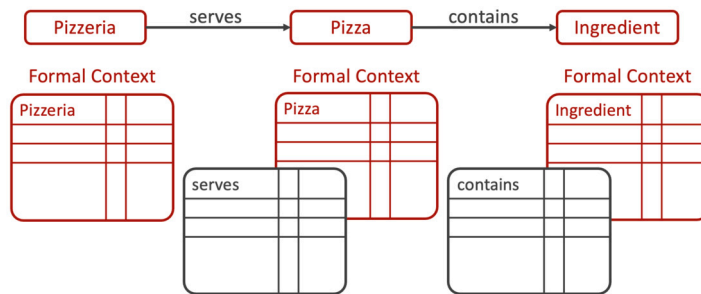


Fig. 5. Top: An entity-relationship (ER) model where pizzerias serve pizzas which contain ingredients. Bottom: Its translation into a Relational Context Family composed of three Formal Contexts (Pizzeria, Pizza, and Ingredient) and two Relational Contexts (serves and contains). Red color is used for ER entities and their corresponding Formal Contexts. Black color is used for ER relations and their corresponding Relational Contexts. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

only two objects out of seven (see right-hand side of Fig. 3 for the complete extents and intents, or right-hand side of Fig. 4 in the simplified notation). In addition, we removed the bottom concept from the figures, since it is added by FCA4J for technical reasons.

Multiple extensions are proposed to deal with complex data in the FCA framework. Without any claim of exhaustiveness, we can mention here multi-valued attributes [16], fuzzy FCA [29], categorical attributes with taxonomies [30], logical descriptions [31], polyadic FCA [32], or pattern structures [33]. RCA [15], Graph-FCA [34], Fuzzy RCA [35] and the approach in [36] have been proposed for dealing more specifically with multi-relational datasets. They produce different complementary results for knowledge engineers [37].

**RCA.** RCA extends the purpose of FCA for taking into account different object categories and links between these objects. The output is the set of concept lattices (one per object category) connected through particular attributes. RCA considers a relational model based on the entity-relationship model, restricted to unidirectional binary relationships. Fig. 5 shows an extension of the example. Its top part represents an entity-relationship model with three entities (Pizzeria, Pizza, and Ingredient). Pizzeria is connected to Pizza via serves, and Pizza is connected to Ingredient via contains. An RCA input follows this model, encoded into a Relational Context Family (RCF). Each entity is represented by a Formal Context (objects described by attributes; to distinguish then latter from complex attributes introduced by RCA, we will call them native attributes). Each relation is represented by a Relational Context (a binary relation between objects of two Formal Contexts). This is illustrated on the bottom of Fig. 5, which outlines the three Formal Contexts (Pizzeria, Pizza, and Ingredient) and the two Relational Contexts (serves and contains) of the RCF for the pizzeria example. Tables 2 and 3 show possible content for these Formal and Relational Contexts.

RCA iteratively builds concepts using native attributes and relations. In the initial step, concepts are built only using the Formal Contexts. For our example, Pizzerias and Pizzas concepts are formed using native attributes that are only identifiers. Ingredients concepts are presented in Fig. 2. In the next steps, the objects are grouped into concepts with respect to their shared native attributes, but also using similar relations they have with concepts built at a previous step. These similar relations are encoded into *relational attributes* that are assigned to objects through extended Formal Contexts. The relational attributes contain an operator inspired by description logics, a relation, and a target concept. For example, in the second step, new concepts can be formed on pizzas, using the contains relation and the ingredients concepts. If we consider CIngredient\_11 which groups the vegan ingredients (its extent is shown in green in Fig. 6), we may be interested by the pizzas that contain only vegan ingredients. This is materialized through a relational attribute  $\exists \forall$  contains(CIngredient\_11), in other words “contains only ingredients that are in CIngredient\_11 (and at least

**Table 2**  
Relational Context Family for the pizzeria example (part 1): Formal Contexts Pizzeria, Pizza, and Ingredient.

Pizzeria	<i>IdHappizzy</i>	<i>IdEataly</i>	<i>IdLafelicita</i>	<i>IdSmallitaly</i>
happizzy	x			
eataly		x		
lafelicita			x	
smallitaly				x

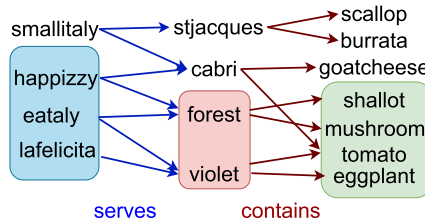
Pizza	<i>IdCabri</i>	<i>IdForest</i>	<i>IdViolet</i>	<i>IdStjacques</i>
cabri	x			
forest		x		
violet			x	
stjacques				x

Ingredient	<i>vege</i>	<i>vegan</i>	<i>spring</i>	<i>summer</i>	<i>autumn</i>
goatcheese	x			x	
burrata	x		x		
scallop			x		
tomato	x	x		x	
shallot	x	x			x
mushroom	x	x			x
eggplant	x	x		x	

**Table 3**  
Relational Context Family for the pizzeria example (part 2): Relational Contexts serves and contains.

serves	<i>cabri</i>	<i>forest</i>	<i>violet</i>	<i>stjacques</i>
happizzy	x	x		
eataly		x	x	
lafelicita			x	
smallitaly	x			x

contains	<i>goatcheese</i>	<i>burrata</i>	<i>scallop</i>	<i>tomato</i>	<i>shallot</i>	<i>mushroom</i>	<i>eggplant</i>
cabri	x			x			
forest					x	x	
violet				x			x
stjacques		x	x				



**Fig. 6.** Objects of Formal Contexts Pizzeria, Pizza, and Ingredient and their links for relations serves and contains. In green, the extent of the concept CIngredient\_11 of vegan ingredients. In pink, the extent of the concept CPizza\_31 of pizzas that contain only vegan ingredients (they share  $\exists V$  contains(CIngredient\_11)). In blue, the extent of the concept CPizzeria\_36 of pizzerias that have at least one pizza in the concept of pizzas that have only vegan ingredients.

one)”. All such relational attributes  $\exists V$  contains(CIngredient\_i) for all ingredient concepts CIngredient\_i are added in the Pizza Formal Context. The concept lattice built on this extended Pizza Formal Context contains an additional concept (CPizza\_31) that factorizes the attribute  $\exists V$  contains(CIngredient\_11) shared by pizzas violet and forest. Its extent is shown in Fig. 6.

At step 3, we may be interested in grouping pizzerias that serve at least one vegan pizza, if we are vegan or going to the pizzeria with a vegan friend. This is materialized through the relational attribute  $\exists$  serves(CPizza\_31). More generally, the Pizzeria Formal Context is extended with relational attributes of the form  $\exists$  serves(CPizza\_i), where CPizza\_i is a concept of the Pizza Formal Context. The Pizzeria Formal Context built at step 3 contains the new concept CPizzeria\_36, that factorizes  $\exists$  serves(CPizza\_31) which is common to happizzy, eataly, and lafelicitata. Its extent is shown in Fig. 6. In other words, it groups pizzerias that serve at least one pizza which contains only vegan ingredients. Fig. 6 highlights the propagation of the concept creation from right to left along the three steps. Fig. 7 shows the Pizzeria, Pizza, and Ingredient concept lattices that contain these concepts. More details about RCA and the theoretical definition are given in [15].

2.2. Outlining navigation needs for RCA

In Section 2.1, we explained how concepts are iteratively built from initial groups of objects that share native attributes, taking advantage of shared relations for propagating this clustering to other objects. Reversely, exploring data organized with RCA benefits from the relational attributes to explore the datasets guided by shared relationships in addition to explore the shared seed attributes, or objects. To illustrate this concretely on a small example, let us use our previous example, and outline how RCA exploration can guide our search for a pizzeria to share a meal with a vegan friend, i.e. a pizzeria serving at least one vegan pizza. The exploration is based on Fig. 7. The built lattices correspond to the expression of a relational query on pizzerias that we can draft as  $\exists$  serves( $\exists V$  contains(vegan)). The initial query consists to look for available vegan ingredients in the pizza datasets. These ingredients are grouped in CIngredient\_11, which is the natural **entry point**. To know the pizzas that contain only vegan ingredients, we **roll up** the relational attribute  $\exists V$  contains(CIngredient\_11). This leads us to Concept CPizza\_31. Its simplified extent is empty, but we can know the grouped pizzas by observing the **full extent**. Similarly, we can roll up the relational attribute  $\exists$  serves(CPizza\_31) and reach CPizzeria\_36. By observing the full extent of CPizzeria\_36, we have our answer. Thanks to FCA, this is not just a flat

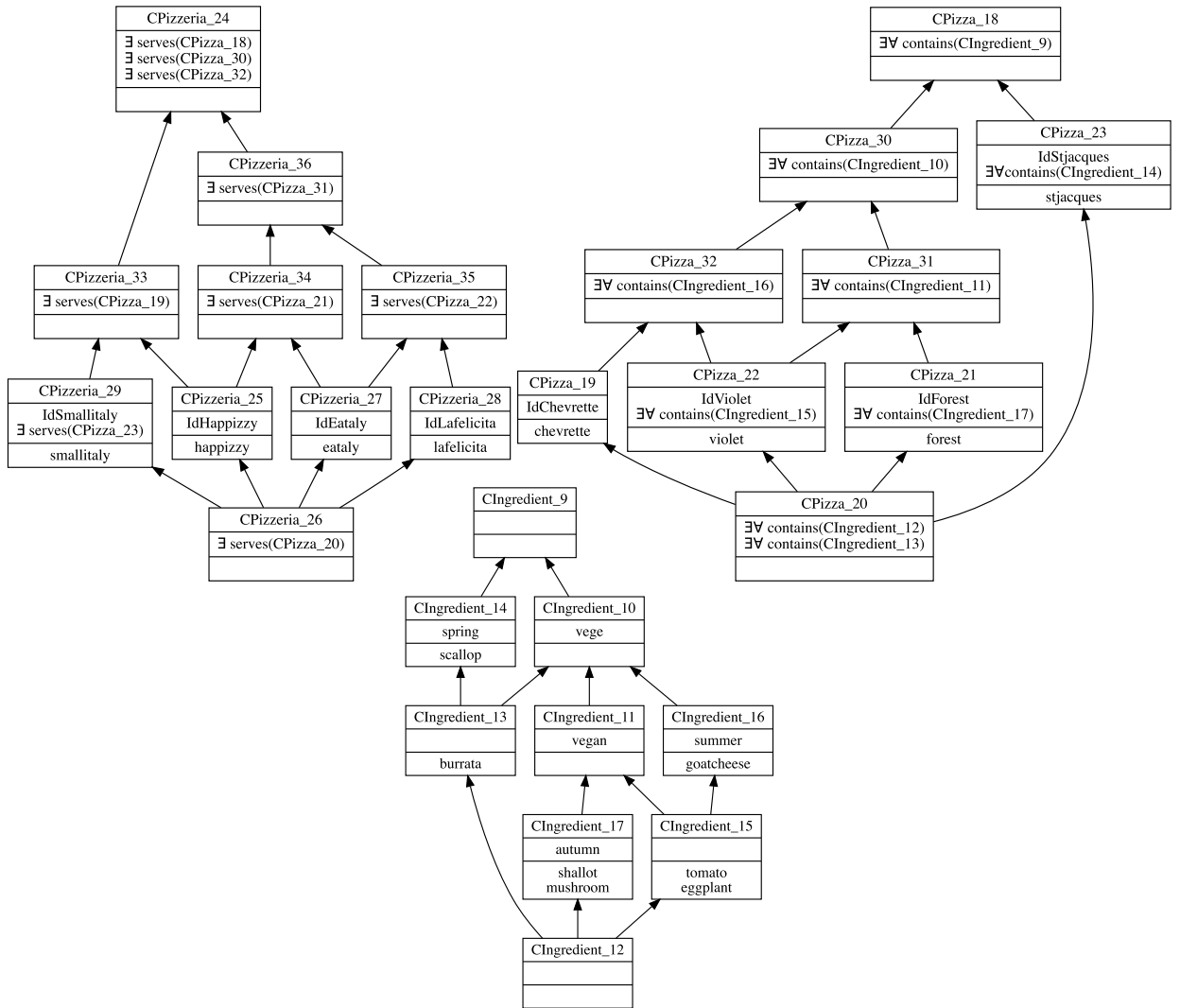


Fig. 7. Pizzeria (left hand side), pizza (right hand side) concept lattices built by RCA at step 3. CPizzeria\_24 introduces the relational attribute  $\exists \text{ serves}(\text{CPizza}_{30})$ , which points to CPizza\_30. CPizza\_30 introduces the relational attribute  $\exists \forall \text{ contains}(\text{CIngredient}_{10})$ , which points to CIngredient\_10 of vege ingredients. CPizzeria\_36 of pizzerias introduces the relational attribute  $\exists \text{ serves}(\text{CPizza}_{31})$ , which points to CPizza\_31. CPizza\_31 introduces the relational attribute  $\exists \forall \text{ contains}(\text{CIngredient}_{11})$ , which points to CIngredient\_11 of vegan ingredients. The simplified notation on extents and intents is used in this figure for the sake of readability. The complete extent of CPizza\_31 is {violet, forest}, as these objects are bottom-up inherited from direct subconcepts CPizza\_22 and CPizza\_21 respectively. The complete extent of CPizzeria\_36 is {happizzy, eataly, lafelicita} as these objects are bottom-up inherited from indirect subconcepts CPizzeria\_25, CPizzeria\_27, and CPizzeria\_28 respectively.

list of pizzerias. These are grouped according to their specific features. E.g. happizzy and eataly more specifically provide forest pizza, reached through  $\exists \text{ serves}(\text{CPizza}_{21})$ , containing autumn ingredients grouped in CIngredient\_17. Similarly, lafelicita and eataly more specifically provide violet pizza (reached through  $\exists \text{ serves}(\text{CPizza}_{22})$ ), containing summer ingredients grouped in CIngredient\_15. By **looking at, and then moving up and down in the neighborhood**, we thus can choose eataly which provides more possibilities. During this exploration, we eventually come back to previously observed concepts. Another situation can occur, where going backward is required: it can arise that the three pizzerias are unfortunately closed or very far away. A solution is to **release the exploration to a previous step** by coming back to CIngredient\_11 and look at the full intent to see what happens when relaxing the vegan constraint. Among the inherited attributes of CIngredient\_11, vege attribute may seem a good alternative for the friend if they accept a slightest change in their diet. This vege attribute is found by going up into CIngredient\_10. Rolling up relational attributes  $\exists \forall \text{ contains}(\text{CIngredient}_{10})$  (in CPizza\_30 extent) and  $\exists \text{ serves}(\text{CPizza}_{30})$  leads to add pizzeria smallitaly to the set of solutions.

From this short story, we can summarize the main needs of a user as follows: (i) initiate exploration by selecting an attribute, (ii) visualizing the neighborhood (iia) and content (iib) of a concept to select the next step and stay in a set of feasible solutions, (iii) moving from one concept to a sub-concept (or a super-concept) in the same conceptual structure, (iv) jumping or roll-back to a



neighboring conceptual structure using a relational attribute, and (v) backtracking easily. In addition, RCA process can be tuned to build concept lattices, AOC-posets, or Iceberg lattices, and it is pertinent to navigate these conceptual structures in the same manner.

### 3. State of the art

In this section, we overview main case studies that have been conducted with RCA (Subsection 3.1) and the ones that motivate or can benefit from the development of an exploration tool. Then we review main existing visualizations for FCA and RCA, and their possible usage in the context of ES (Subsection 3.2).

#### 3.1. An overview of applications of RCA

RCA has been initially designed in the context of software engineering and inspired by [38]. The purpose was to assist a form of normalization of UML class diagrams, in the spirit of database normalization. This normalization aims to provide a maximal factorization of attributes, methods, and association ends. As a consequence, the class diagram does not contain redundancies, abstractions are highlighted, and specialization/generalization links are systematically present [39]. This application of RCA is a typical example where RCAviz would have been useful, as an assistant that delivers step-by-step the possible abstractions among attributes, operations, associations, and classes to the software designer.

Then applications have been developed into different domains. RCA has been used for Information Retrieval (IR), mainly querying, in collections of legal documents connected through cross references [40]. IR is another field where FCA has already been considered with success [41], and which could benefit from an exploratory tool.

In the domain of semantic web and knowledge representation, RCA has been used for the design of ontologies [42,43] or extracting knowledge patterns [44]. For the part of the ontology design which is concerned by extracting new abstractions and presenting them to the ontology designer, RCAviz may be adapted. An approach for extracting coherent families of dependent link key candidates in RDF data has been proposed in [45]. It is an open question to know whether using RCAviz to give an expert the possibility to locally explore the link key candidates organized into concept lattices would be relevant.

Knowledge has been extracted from environmental datasets in [46–48]. In some of these applications, ES is relevant, for the experts to build new hypotheses and experiments.

In software engineering, multiple applications can be found, e.g. UML class model (mentioned previously) and use case model normalization [39,49,50]; Design defects refactoring [51]; Model transformation pattern extraction [52]; Web service or component classification [53]; Variability modeling [54,55]; Feature mining or location [56,57]. In most of these applications, the conceptual structure aims to assist the software engineer to extract interesting clusters (e.g. components), or abstractions (e.g. super-classes), to make organization choices (e.g. refinement links in Feature models), or to assess similarities (e.g. possible web service substitution). It is thus relevant to propose an assistant tool that allows them to locally explore the structures, and incrementally build the software artefacts.

Finally, in industrial information systems, it has been used to combine data coming from different tools and make these tools inter-operate. This has been concretely experimented in the domain of brain activity measurement [58]. RCA also has been used to extract implication rules in aluminum die casting process description and identify anomalies [59]. RCAviz is not adapted to rule visualization and other tools would be useful.

#### 3.2. FCA and RCA visualization

*Visualizing and exploring in FCA tools.* A significant amount of proposals have been done to visualize and explore the conceptual structures.

Some proposals visualize the structure itself. Conexp [60] has been intensively used throughout the FCA literature. The partial order is displayed, with red coloring of edges when some edges are hidden or with ambiguous interpretation. Ancestors and descendants of a concept can be highlighted. Nodes have specific colors to indicate if they introduce objects or attributes, and their shape as well as the labeling can be tuned. Other proposals that show the structure are Latviz [61], Carve (based on a decomposition tree) [62], DAnCe (with concept enumeration guided by the user) [62], and the approach by expandable and collapsible concept trees [63]. Additional views can be provided, like in RV-xplorer [64], where pie-charts show a local view on a concept, its content, and its neighboring, as well as statistics on the next levels to guide the user.

Other approaches do not disclose the overall structure and focus on displaying and interacting with specific objects or attributes representations. CREDO [65] and SearchSleuth [66] propose approaches for web navigation. The navigation is guided by links and terms that can be clicked for specializing or generalizing the query and navigate on the underlying lattice which is itself not displayed. SORTeD [62] is dedicated to the retrieval of documents described and queried by terms. The user interface helps in building and refining conjunctive Boolean queries, by term addition or removal. Latent Semantic Analysis (LSA) is used to rank the results regarding to their similarity to the query terms. In addition, the query terms are ranked regarding their similarity to the result set. SPARKLIS [67] is dedicated to RDF data and uses lists of urls, types, and complex operators for expressing queries. SPARKLIS also provides an interactive history and a visualization of the query in SPARQL. ConceptCloud [68] displays a focus concept with a tag cloud. This tag cloud shows objects and attributes, with size and colors that highlight their importance and their category. Tag selection allows the user to change the focus concept. Other similar approaches use object representations other than terms, such as photos [69] or movie posters [70]. For the design of RCAviz, we choose to show the partial order structure. From existing FCA tools,

**Table 4**

Comparison of RCA tools on their visualization features. Brown color highlights strategies that make these tools very different from RCAviz. Teal color highlights aspects similar to those of RCAviz.

	Output files .dot, .pdf or screen copy	Visualization	Interaction by Navigation in and between structures
Galicia	x	x	
eRCA/Galatea	x	Global/external use of Graphviz	
FCA4J	x	Global/external use of Graphviz	
COGUI/RCA	x	Global/internal use of Graphviz	
RCAexplore	x	Local/in textual concept-browser	In concept-browser and along relational attr. in global visualization

we keep the idea to enable selecting a focus concept with a query formed on top of initial objects or attributes, to allow step-by-step navigation towards sub-concepts and super-concepts, and to ease backtracking to a previous step, with an history. These features have to be extended to the specifics of RCA.

*Visualizing in RCA tools.* Four available tools implement RCA: Galicia,<sup>3</sup> eRCA<sup>4</sup> improved in Galatea,<sup>5</sup> RCAexplore, and a Cogui<sup>6</sup> plugin converted as a Java library named FCA4J. RCAexplore and FCA4J have the largest functional features regarding the number of input/output and accepted quantifiers. In addition, RCAexplore provides the possibility to tune the RCA process by interactively changing at each step: the included contexts, the algorithm and the operators. Regarding the visualization, Galicia provides an advanced tridimensional lattice representation, with a graph layout built thanks to a force algorithm, but no specific operators to navigate within a structure or from one structure to another. RCAexplore provides a visualization tool relevant for navigating small-size structures. This tool shows the whole structure but does not enable zooming. Concepts and relational attributes are clickable to go from one structure to a neighboring structure, but selecting a focus concept and going step-by-step using local views is not proposed. RCAexplore also provides a browser enabling the navigation between concepts textually described. Finally eRCA/Galatea, RCAexplore, and FCA4J produce outputs in the DOT file format, which can be visualized or transformed using Graphviz.<sup>7</sup>

Table 4 synthesizes the main features of the RCA tools with regard to outputs, visualization and interaction. These tools do not propose neither interactive and graphical local views on the different conceptual structures, nor advanced solutions for navigating inside a structure or from one structure to another. In the design of RCAviz, we will propose several visualization panels with local views, and allow the user to know which neighboring structure has been previously accessed or can be accessed through relational attributes. The step-by-step navigation from one structure to another will be done by clicking on the relational attributes. An history will allow easy backtracking or replaying a navigation.

#### 4. Description of RCAviz's features

As shown in Section 2, RCA classifies data in a structure which can be modeled as a set of Directed Acyclic Graphs (DAG) connected via the relational attributes that are a kind of abstraction of the links between individual objects. Each DAG corresponds to a Formal Context: In the pizza example, we have a DAG for pizzerias, a DAG for pizzas, and a DAG for ingredients (see Fig. 7). As explained in Sections 2.2 and 3, we want to explore these DAGs, using the DAG structure and content, and the inter-DAG connections.

Most papers in the literature on the visualization of DAG (see [71] for an introduction) propose algorithms for positioning vertices and drawing edges, and aim to limit the number of edge crossings. This is relevant for small datasets like the pizza dataset. But in practice, DAGs may have hundreds of vertices, thus visualizing the whole structure creates visual cluttering issues. Moreover, in our context, relational attributes form a complementary DAG that we want to leverage to switch from one DAG to another interactively. This is why we opted for a local visualization system with a step-by-step navigation.

##### 4.1. Requirement analysis

In common practice, visualization design iterates on (1) defining needs and suitable data structures, (2) proposing visual encodings and interactive features meeting these needs, (3) providing the results to the users so that they can give feedback and refine their needs (see for instance [72,73]). We made the first design of RCAviz by organizing regular meetings with the three visualization experts in charge of providing the visual platform, an RCA expert, and a dataset expert [19]. For the improved design, we took advantage of the lessons learned through the usage of RCAviz by students in introductory RCA courses and the expertise we gained

<sup>3</sup> <http://www.iro.umontreal.ca/~valtchev/galicia/> (accessed: 2023-11).

<sup>4</sup> <https://code.google.com/archive/p/erca/> (accessed: 2023-11).

<sup>5</sup> <https://github.com/jrfaller/galatea> (accessed: 2023-11).

<sup>6</sup> <http://www.lirmm.fr/cogui/> (accessed: 2023-11).

<sup>7</sup> <https://graphviz.org/> (accessed: 2023-11).



Fig. 8. The *Concept Selector*. The user can select *Objects* and *Attributes* of interest. Then, the list of the available *Concepts* is updated on the right, and the user can select one of them to launch the *Explorer* and the *History*. Concept 11 is in bold face as it introduces the selected attribute *vegan*.

from solving questions on specific datasets using RCA. Here is the final list of the needs identified during these meetings and concrete usages:

**[R1] Selection of a concept.** We need a starting point in a navigation. In RCAviz, this is an initial concept, reached after the selection of attributes or objects of interest for the user.

**[R2] Visualization of a concept: its neighbors [R2a] and its content [R2b].** The user must be able to see the content of a concept, as well as its neighbors, i.e. the lower and higher level concepts in the DAG and the neighboring concepts in the other DAGs. In the initial version, only the objects and attributes introduced by a concept were required. In the improved version, we added the need for visualizing the inherited objects and attributes. It is required to visualize native attributes as well as relational attributes. This is useful to update the initial query with elements that will not bring the user in queries without answers (empty solution set).

**[R3] Step-by-step navigation.** The user must be able to move to neighboring concepts of the displayed concept (whether they belong to the same DAG or not).

**[R4] History.** The user must be able to keep a history of its navigation, i.e. the list of the previously explored concepts. Then, they must also be able to navigate through this history and to save it to be in capacity to resume its exploration later.

RCAviz is designed to meet these requirements, with a visualization comprising: An initial view to select the starting concept, called the *Concept Selector*; Two coordinated views to allow the step-by-step navigation (the *Explorer*) and to visualize and interact with the navigation history (the *History*).

#### 4.2. Concept selector

When RCAviz is launched, the first displayed view is the *Concept Selector*. It is designed to meet the [R1] requirement. The user must first upload a JSON data file, which can be built using FCA4J. This data file contains the DAGs description: for each conceptual structure, the concepts, the introduced attributes, the introduced objects, the links of the Hasse diagram (parents and children of each concept), and optionally the full extent and intent. After uploading the data file, the users either use the proposed context (the first in the JSON file), or they can select another one. Each context corresponds to one DAG of the dataset. The list of objects, attributes, and concepts from the selected context is then displayed, as shown in Fig. 8. In this figure, the objects and the attributes of the “ingredient” context are shown. The user can select one or more objects and one or more attributes. After a selection, the other lists are filtered and updated according to the logical operators used. The list of corresponding concepts on the right shows the concepts that are relevant entry points for the navigation, with regards to the current selection of objects and attributes. E.g. this list can display the concepts containing (*object o1 and object o2*) and (*attribute a1 or attribute a2*). When the object list or the attribute list are long, a search field allows the users to easily find specific elements. They can also select or deselect all the elements of each list. The list of selected items is displayed when hovering over *Selected*. When a unique object (resp. a unique attribute) is selected, the id of the concept, which introduces it, appears in bold face in the right panel. Once the users have decided which concept is relevant for them, they can select it from the concept list on the right and use the bottom right *Confirm* button to launch the *Explorer* and the *History*.

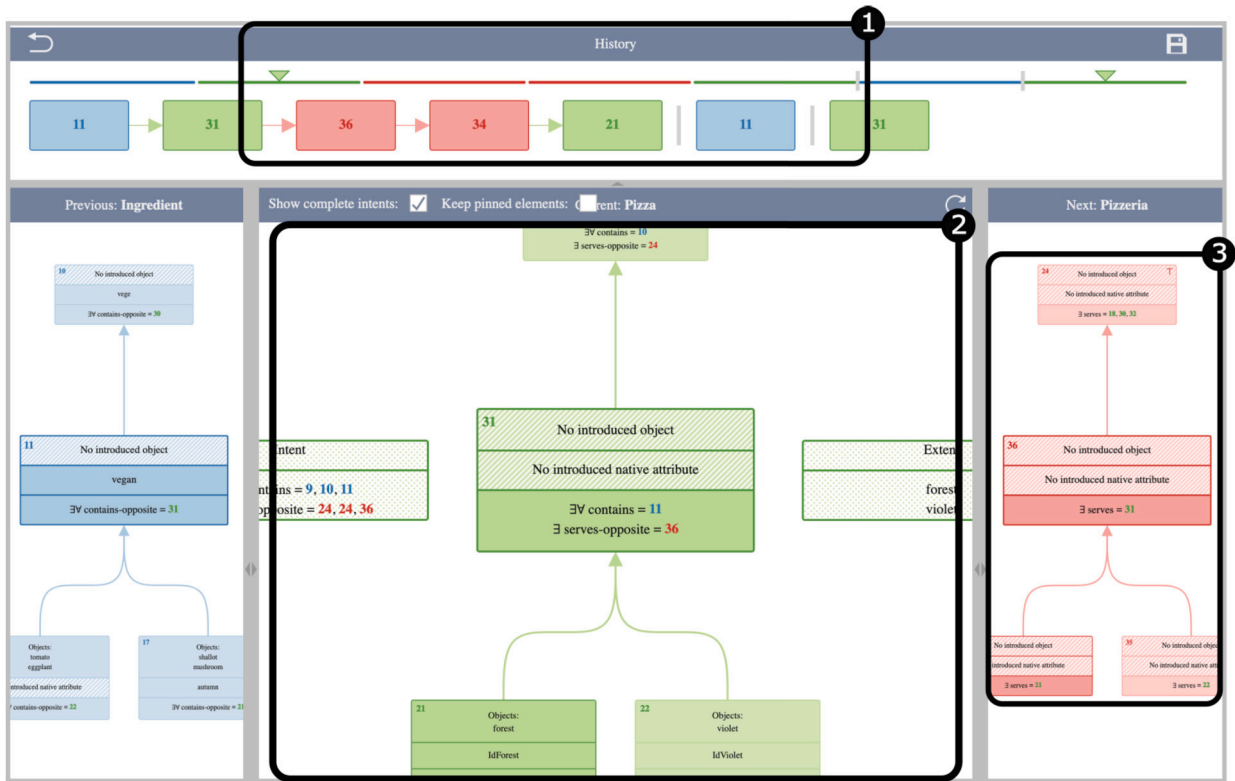


Fig. 9. The RCAviz exploration interface. The *History* (1) is shown in the top panel. It contains a list of the previously explored concepts, that can be clicked to navigate to them. The *Explorer* is shown in the bottom: On its center panel (2), the view shows a concept content and its neighborhood. The user can navigate through the concepts by clicking on them. The left panel shows the previously displayed DAG. When the user hovers over a link pointing to a concept from another DAG, this latter concept is displayed on the right panel (3).

### 4.3. Explorer

Fig. 9 shows the *History* (top panel) and the *Explorer* (bottom panels) on our pizza example, after seven steps of navigation beginning with Concept 11 (vegan ingredients). The *Explorer* is divided into three sub-panels:

1. The middle panel (**current**) displays the currently selected concept at its center, with the concept’s neighborhood (children and parents in the DAG).<sup>8</sup>
2. The left panel (**previous**) shows the previously explored context.
3. The right panel (**next**) is used when the user hovers over a link in the middle panel. In this case, this panel displays a preview of the link target’s context.

By displaying neighborhoods of the currently selected concept, the middle panel answers [R2a]. To answer [R2b], each concept is represented as a colored rectangle composed of three stacked boxes. The concept id appears in the top left corner of the top of the box. The introduced objects are presented in the top box, entitled *Objects*. If there is no introduced object, the box only displays *No introduced object*. By hovering over this first box, if the JSON file contains the information on the full extent (inherited and introduced objects), this extent is shown on the right of the concept in a box entitled *Extent*.

The introduced native (i.e. non relational) attributes appear in the middle of the box. Similarly, if there is no introduced native attribute, the box only displays *No introduced attribute*. The introduced relational attributes appear in the bottom of the box. By hovering over these two attribute boxes, if the JSON file contains the information on the full intent and the checkbox *show complete intent* is ticked, this intent is shown in a single box on the left of the concept box. Otherwise the native attribute intent or the relational attribute intent are shown in separate boxes, when hovering over the relevant section of the concept.

The intent and extent boxes can be pinned/unpinned by clicking on the area of the concept box that displays them when hovered, which allows the user to interact with their content (e.g. to preview or navigate to a relation target). By default, pinned elements are unpinned when navigating to another concept, but the checkbox *keep pinned elements* can be selected to keep them visible.

<sup>8</sup> Note that sub-concepts are parent nodes and super-concepts are children nodes in terms of the displayed partial order where a lower node “specializes” (is lower than) an upper node.

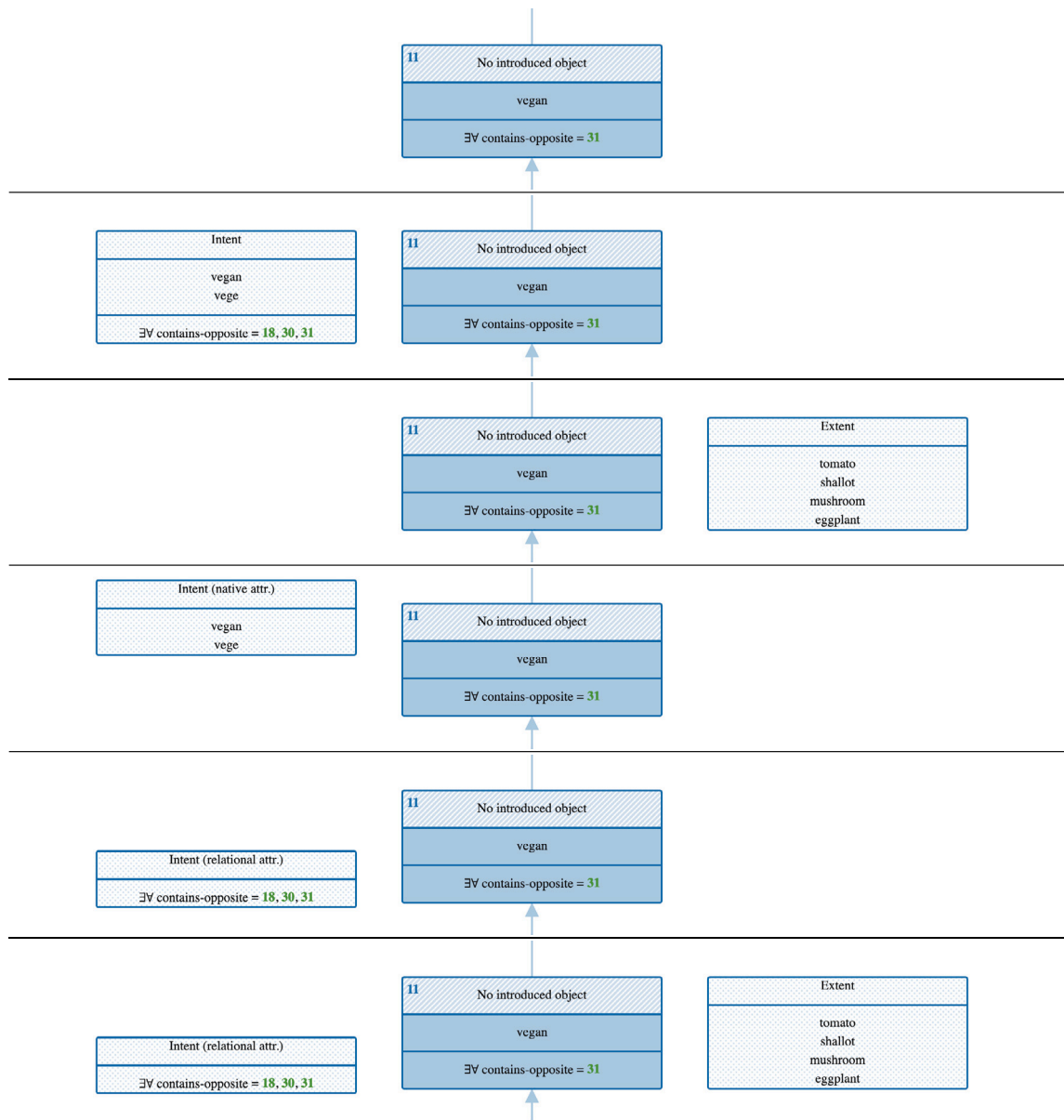


Fig. 10. Possible views of Concept 11 (from top to bottom: i) view with introduced objects and attributes; ii) View with full intent in a unique box; iii) view with full extent; iv) view with full native intent; v) view with full relational intent; vi) example of view with intent and extent.

Variations on how a concept intent and extent can be displayed are shown in Fig. 10 for Concept 11.

Relational attributes are grouped by relation. For instance, in Fig. 9, in the intent of Concept 31 (box on the left of the concept)  $\exists V$  contains = 9, 10, 11 represents relational attributes  $\exists V$  contains(CIngredients\_9),  $\exists V$  contains(CIngredients\_10) and  $\exists V$  contains(CIngredients\_11). When there are many introduced relational attributes with the same relation, only their number is shown in order to save space.

The user can navigate step-by-step by clicking on the concepts [R3]. For example, if they click on the concept 21 (sub-concept of 31), it is positioned in the middle of the view and its parents and children are displayed below and above. T for the top (resp. ⊥ for the bottom), located at the top right of a concept, means that it has no children (resp. no parents).

When the user hovers over a concept id of a relational attribute, whether it is displayed on the concept box or on the intent box, the concept in the corresponding DAG is displayed in the right panel. In Fig. 9, the mouse is positioned on the red link to concept 36, thus the corresponding concept is displayed with its neighborhood in the right panel. Should the user click on the link, the center green DAG would be positioned in the left panel [R3] and the red DAG on the right would be positioned in the central panel. We thus keep track of the previous exploration step [R4]. To facilitate the navigation, an opposite link is provided for each relational attribute. For example, if a concept  $i$  contains a relational attribute serves to a concept  $j$ , then the concept  $j$  contains a relational



attribute serves-opposite to the concept  $i$ . Note that these opposite links do not correspond to a serves-opposite formal context which would be the reverse of serves. They are used to roll up the relational attributes, as we have shown on our example.

A specific color is associated with each DAG. We use a categorical color scale adapted to the representation of nominal variables [74]. In our example, the ingredient DAG is shown in blue, the pizza DAG in green, and the pizzeria DAG in red. The concepts that have already been explored appear with lower brightness than the other concepts [R4]. In Fig. 9, we can see that the green concept 21 has already been explored, but not the 22 or the 30.

Finally, it is common to have a high number of parents or children inducing visualization and navigation issues to see all of them. Therefore, if the number exceeds 5, we display a single “super-concept” indicating the number of these nodes parents or children and proposing to click for more details. When the user clicks on it, a pop-up displays a view similar to the *Concept Selector*. The user can then select attributes, objects, and one or more concepts which, after validation, are displayed in the central panel.

In this release, we have chosen a top to bottom representation of the specialization relation as this is the common representation of partial orders by Hasse diagrams. Alternatively, we may explore a left to right representation, where a concept will be on the left of its sub-concepts, to show more sibling concepts as they will be vertically organized.

#### 4.4. History

The left panel of the *Explorer* shows the previous explored context. However, these functionalities do not fully meet the requirement [R4]. We therefore added the *History*, which is a view entirely dedicated to [R4]. It comprises two parts (see Fig. 9): a colored line at the top and a list of concepts below.

The colored line is made up of segments, each segment representing an explored concept. The concept color refers to its context and DAG. The length of the line adapts to the width of the visualization so that all the explored concepts stay visible. Therefore, the more concepts the user has explored, the smaller the segments representing those concepts are. The user can see if the current concept of the *Explorer* has already been explored with a small triangle placed above each segment representing this concept. For example, in Fig. 9, we see two triangles, one on the last segment representing the concept displayed in the *Explorer* and one on a previous segment: this means that the current concept 31 has already been explored once during the navigation.

The list of the id of the explored concepts is displayed under the colored line. As this list can be long, a horizontal slider is activated when the available space is not enough to display all of them in the view. The concepts are represented by colored rectangles according to the DAG they belong to. An arrow connects two concepts if the user has moved from one to the other by navigating step-by-step. In this *History*, if the user clicks on a previously explored concept, it is displayed in the *Explorer*. A vertical bar is then used in the *History* to show that the sequence was not produced by a step-by-step navigation. In Fig. 9, such a bar appears between concepts 21 and 11, meaning that when concept 21 was in the central panel, the user clicked on the history on the first blue box (concept 11). When the user hovers over a concept, a small red cross appears at the corner. It can be used to remove this concept from the *History*.

Finally, the user can save the state of their navigation as well as the history by clicking on the save button in the top bar. They can then upload the produced file to resume their exploration.

#### 4.5. Implementation

RCAviz is a web platform developed in HTML, CSS and JavaScript. The panels and the visualizations are made using the D3.js<sup>9</sup> [75] library. Saving the history file is made possible with the FileSaver<sup>10</sup> library. The tool<sup>11</sup> and the source code<sup>12</sup> are licensed under MIT and available online. An extensive documentation containing sample datasets is available to allow anyone to use the tool with the samples or with their own datasets.

## 5. A typical usage scenario

In this section, we present a typical usage scenario that illustrates the benefit of RCAviz on a real-world dataset from the domain of agro-ecology. The addressed question and the dataset are presented in Subsection 5.1, and the scenario in Subsection 5.2.

### 5.1. Addressed question and dataset

Data used for this scenario is available on a repository<sup>13</sup> and on the website of RCAviz. This scenario has been elaborated for the Knomana project [20], which aims to identify candidate plants that can be used as alternative solutions to synthetic pesticides for crop and animal protection.

The Knomana dataset gathers descriptions of plants used as biopesticides found in the scientific literature. Each description is characterized using attributes, among which the pesticidal plant, its species, genus and family, the protected organism, the controlled

<sup>9</sup> <https://d3js.org/> (accessed: 2023-11).

<sup>10</sup> <https://github.com/eligrey/FileSaver.js/> (accessed: 2023-11).

<sup>11</sup> <https://rcaviz.lirmm.fr/> (accessed: 2023-11).

<sup>12</sup> <https://gite.lirmm.fr/emuller/rcaviz> (accessed: 2023-11).

<sup>13</sup> <https://seafiler.lirmm.fr/d/4918efce633448b6b030/> (accessed: 2023-11).



Fig. 11. Step 1. Selection of *Fusarium oxysporum* in the Pest context, thanks to the Selector wizard.

aggressor (pest or disease), the method to prepare and apply the biopesticide, and the corresponding reference in the literature. Depending on the expert questions, different parts of the dataset are extracted for exploration purpose, in order to focus on the relevant information.

We revisit an expert question raised and studied in a previous work with an ad hoc algorithm [76]. The expert question was phrased as follows: “I need to control *Fusarium oxysporum* in Burkina Faso. May I find a protection system with these requirements or with close requirements?”. The Knomana dataset excerpt is structured around a model with four Formal Contexts representing respectively the protection systems (named ProtectionSystem), the biopesticide plants (named Plant), the protected organisms (named Protected), and the controlled pests (named Pest). ProtectionSystem is connected to the three other contexts via Relational Contexts called respectively uses (towards Plant), protects (towards Protected), and controls (towards Pest). It comprises 35 protection systems; 14 plants described by species (e.g. *Lantana camara*) and genus (e.g. *Lantana*); two protected organisms (*Arachis hypogaea* and *Sorghum bicolor*) described by species (e.g. *Sorghum bicolor*), genus (e.g. *Sorghum*), and family (e.g. Poaceae). The protection systems are described by 26 Boolean attributes concerning respectively the part of the plant where the biopesticide is applied (leaf, stem, whole plant), the kind of extract (dry, water, essential oil), the form of the protected organism (e.g. seed), the mode of action (e.g. repulsion, toxic), and the country where the observation has been made. Some information are not filled in the dataset. In such case, we used attributes composed with the string “empty<T>”, where T can be, for example “Part” or “Form”. FCA4J has been used to apply RCA and produce the JSON file input of RCAviz. We built the whole concept lattices and used the operator  $\exists$  on the three relational contexts. The result of RCA is a concept lattice for each formal context. These concept lattices respectively contain 73 ProtectionSystem concepts, with 32 relational attributes towards the three other concept lattices; 20 Plant concepts; 5 Protected concepts; 7 Pest concepts.

## 5.2. Usage scenario

To address the expert question about finding a solution for controlling *Fusarium oxysporum* in Burkina Faso, we first select object *Fusarium oxysporum* in the Pest context (see Fig. 11). The Selector view highlights, in bold face, the fact that the Pest concept 46 introduces *Fusarium oxysporum*. In Fig. 11, we select this concept 46 (the concept is marked).

After clicking the confirm button, we reach the Explorer view with Concept 46 as the focus concept (see Fig. 12). This concept contains the relational attribute  $\exists$  controls-opposite = 53, which leads to protection systems that control this pest.

By following the relational attribute  $\exists$  controls-opposite = 53, we reach the protection system concept 53 (see Fig. 13). By opening the full intent, we observe that the protection systems grouped in this concept are all in Benin. The places where the observations are made are important for two reasons: climatic or other conditions may change the effects of a treatment; the biopesticide should be easily found in the place where it is used for several considerations, including economic.

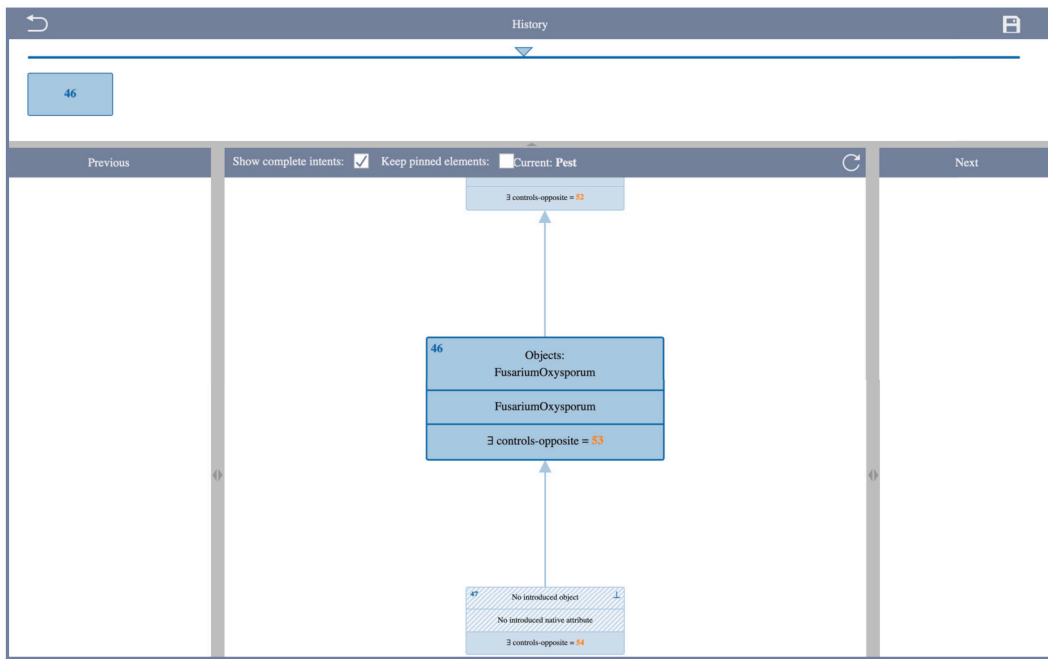


Fig. 12. Step 2. Focus pest concept 46 and its relational attribute  $\exists$  controls-opposite = 53.

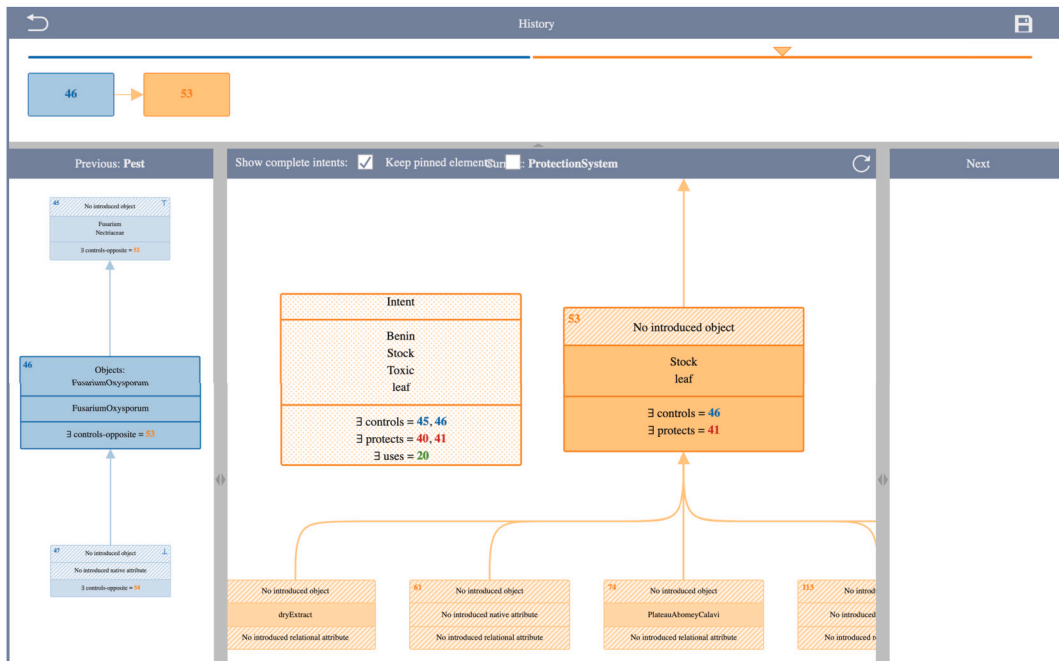


Fig. 13. Step 3. Protection system concept 53 and its full intent opened on its left.

Now we would like to know which plants are used. We explore below the sub-concepts introducing relational attributes with operator and relation  $\exists$  uses. E.g. we can go through the protection system concept 74 (see Fig. 14), and its sub-concept 98 (see Fig. 15). Concept 98 contains  $\exists$  uses = 27. The plant concept 27 introduces *Corymbia citriodora* as can be seen by hovering the relational attribute. Unfortunately, all these protection systems are necessarily in Benin, as recalled in the full intent opened on the left of the concept box. The control of some *Fusarium* species by *Corymbia citriodora* has only been observed in Benin. This is a first interesting track for experts, who could assess the potential and availability of this plant in Burkina Faso.

Experts may also consider looking at plants available in Burkina Faso, for controlling *Fusarium* species. To that end, we return to protection system concept 74 using the history (see Fig. 16) and move to protection system sub-concept 72 (see Fig. 17).



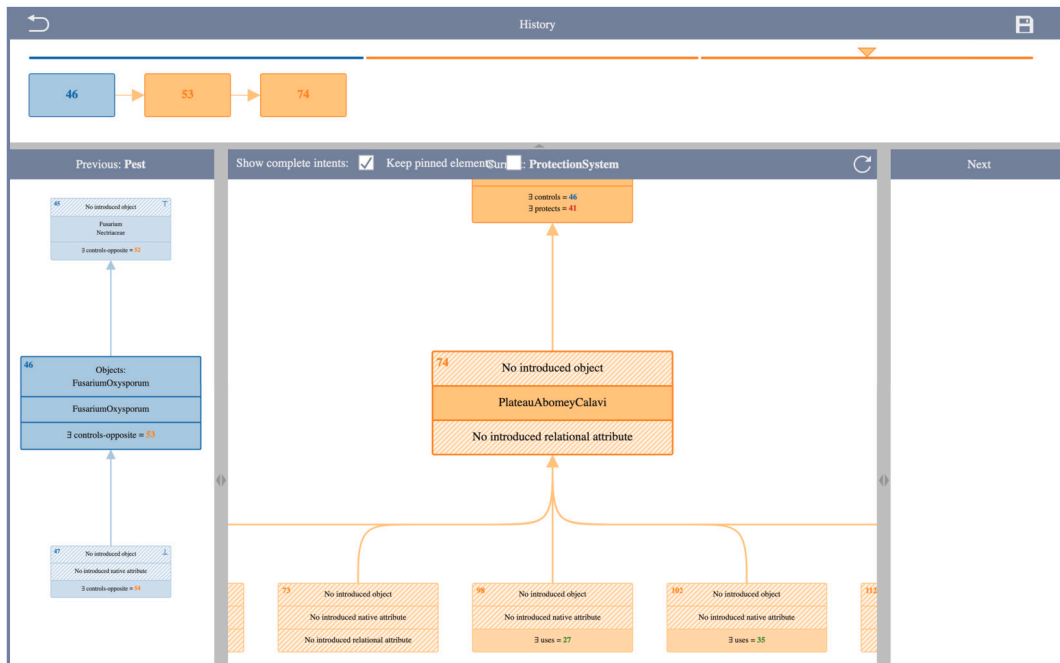


Fig. 14. Step 4. Protection system Concept 74 explored for finding a relational attribute of the form  $\exists$  uses.

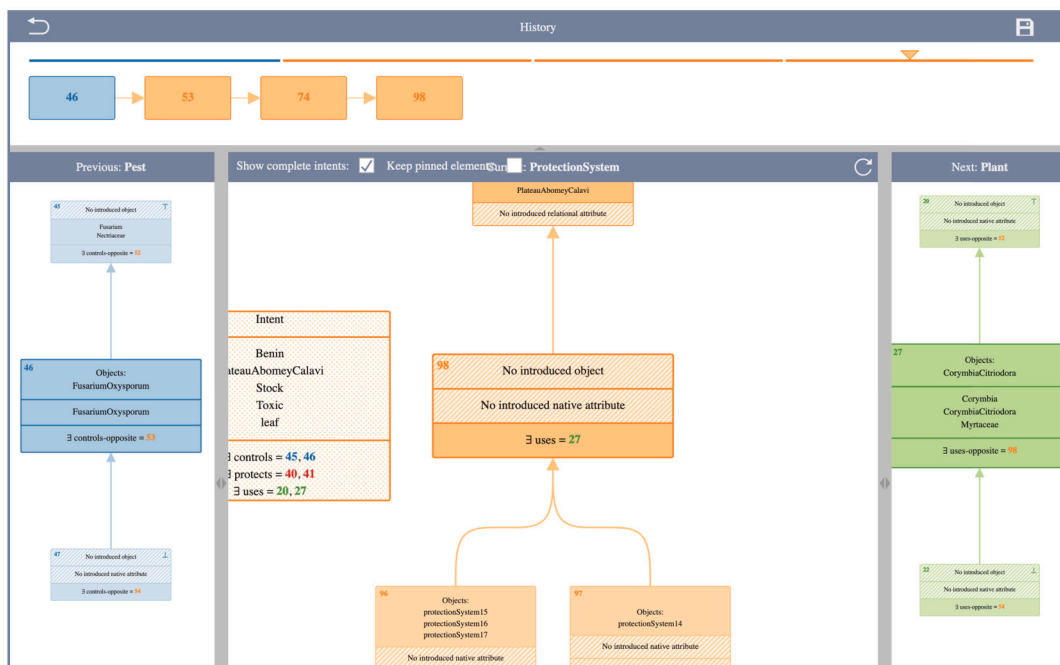


Fig. 15. Step 5. Protection system Concept 98 explored for finding a relational attribute of the form  $\exists$  uses. The full intent contains Benin and  $\exists$  uses = 27, which points to the plant concept which introduces *Corymbia citriodora*. This concept 98 groups protection systems that (among others) control *Fusarium oxysporum* in Benin.

Still in the search for  $\exists$  uses relational attributes, we move from concept 72 to its sub-concept 105 (see Fig. 18). This concept introduces the protection system 22 and inherits  $\exists$  uses = 29 from protection system Concept 106. By hovering on the relational attribute, we see on the right panel that concept 29 corresponds to *Lantana camara*.

We can move to plant concept 29 (see Fig. 19), and roll up to the protection system concept 106 by clicking on the relational attribute  $\exists$  uses-opposite = 106 (see Fig. 20).

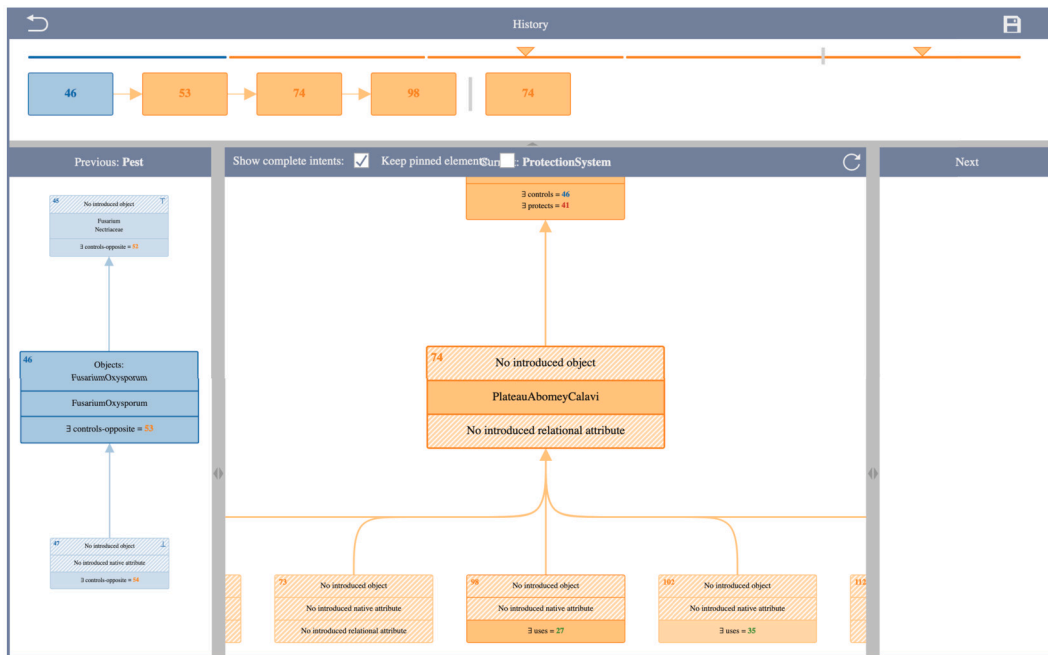


Fig. 16. Step 6. Going back to concept 74 by clicking on it in the history. This is highlighted by the bar between concepts 98 and 74 in the history and the two orange triangles.

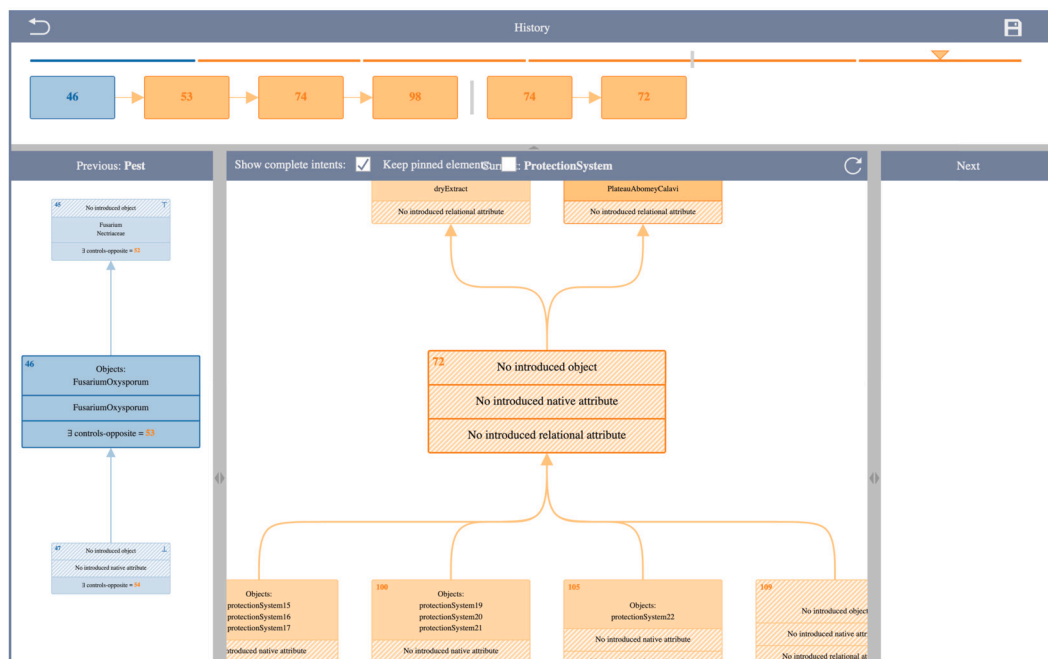


Fig. 17. Step 7. Focus on Concept 72, a step in the exploration.

We can move to Concept 104 which is a cousin of Concept 105 (this latter concept introduces the protection system 22). Concept 104 introduces the protection system 26 (see Fig. 21). It also inherits  $\exists$  controls = 50, meaning that the protection system 26 controls the pest *Fusarium Solani*. This pest is indeed introduced in pest concept 50 as observed by hovering the relational attribute. The control is made with *Lantana camara*, as it is indicated by the relational attribute  $\exists$  uses = 29. The expert can hover the relational attribute, if they forgot what was introduced in Concept 29. Among the inherited attributes of protection system 26, we find Burkina Faso, meaning that *Lantana camara* is used in Burkina Faso to control *Fusarium solani*. Thus it is very likely that *Lantana Camara* can be also used in Burkina Faso, and this time, to control *Fusarium Oxysporum*.

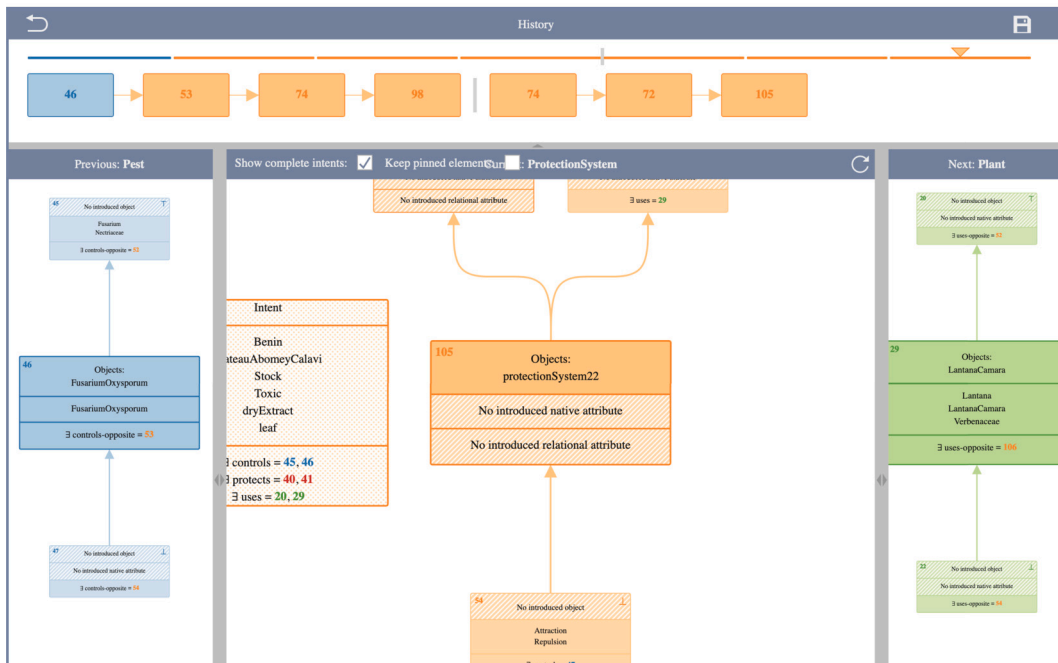


Fig. 18. Step 8. Focus concept 105 which introduces the protection system 22, and inherits  $\exists$  uses = 29. On the right panel, we can see that Concept 29 introduces Lantana camara.

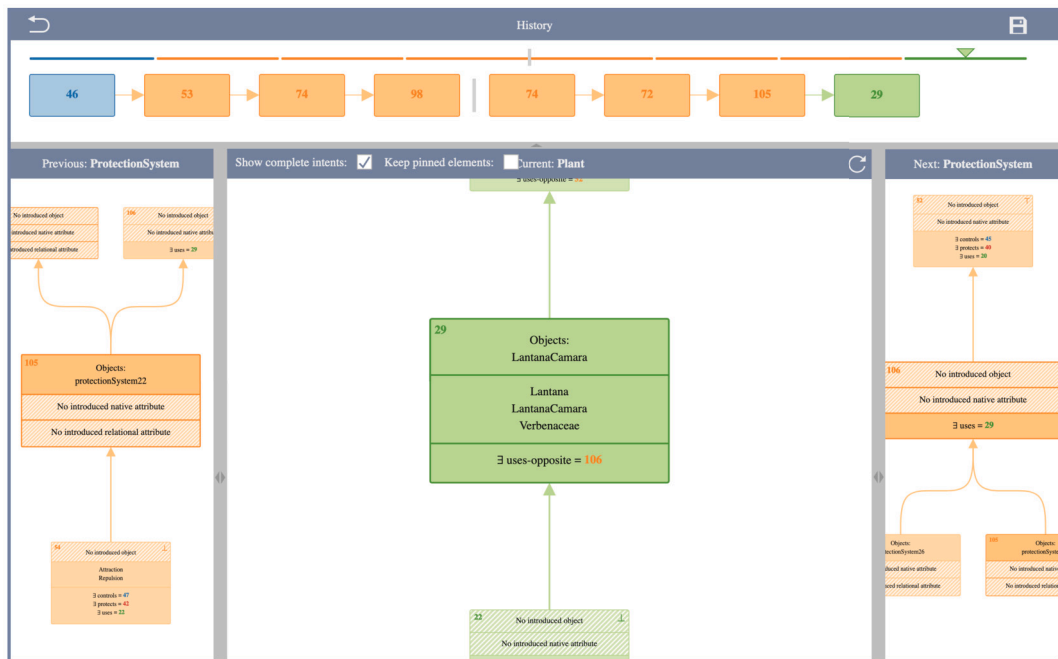


Fig. 19. Step 9. Focus Concept 29, which introduces Lantana camara. It contains the relational attribute  $\exists$  uses-opposite = 106.

To summarize, the exploration scenario revealed: a plant (*Corymbia citriodora*) with a verified usage to control *Fusarium oxysporum*, but not in the right country (Benin); another plant (*Lantana camara*), with a verified usage to control another *Fusarium* species (*Fusarium solani*), in the target country (Burkina Faso). These are two tracks for the experts. They can assess (with new experiments) the availability and the efficiency of *Corymbia citriodora* in Burkina Faso to control *Fusarium oxysporum*. They also can assess the efficiency of *Lantana camara* to control *Fusarium oxysporum*.

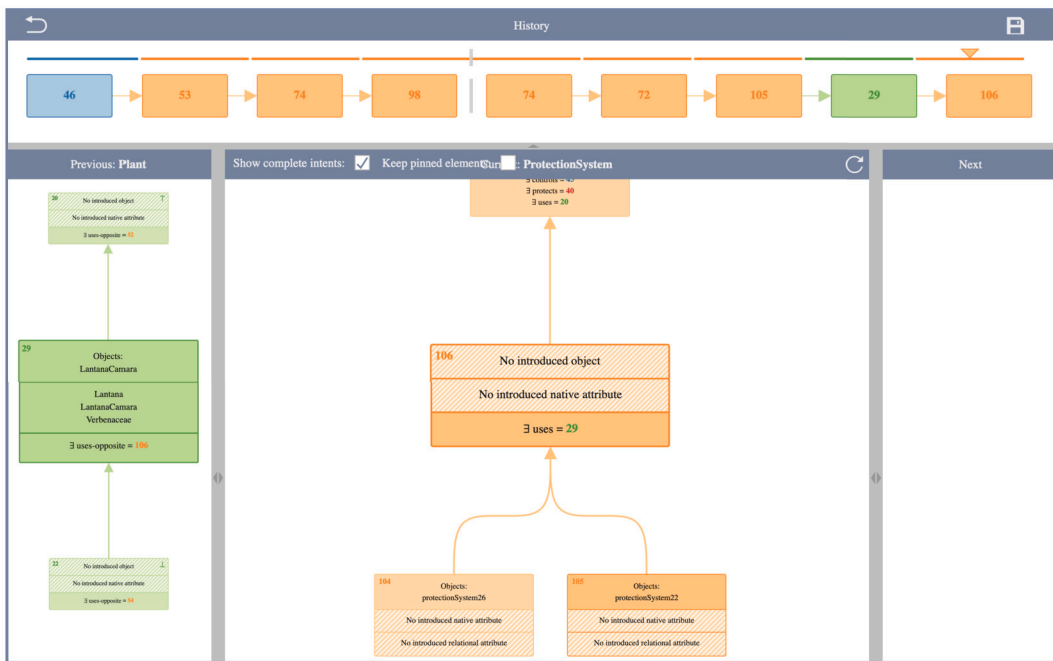


Fig. 20. Step 10. Focus concept 106 which groups protection systems that use Lantana camara.

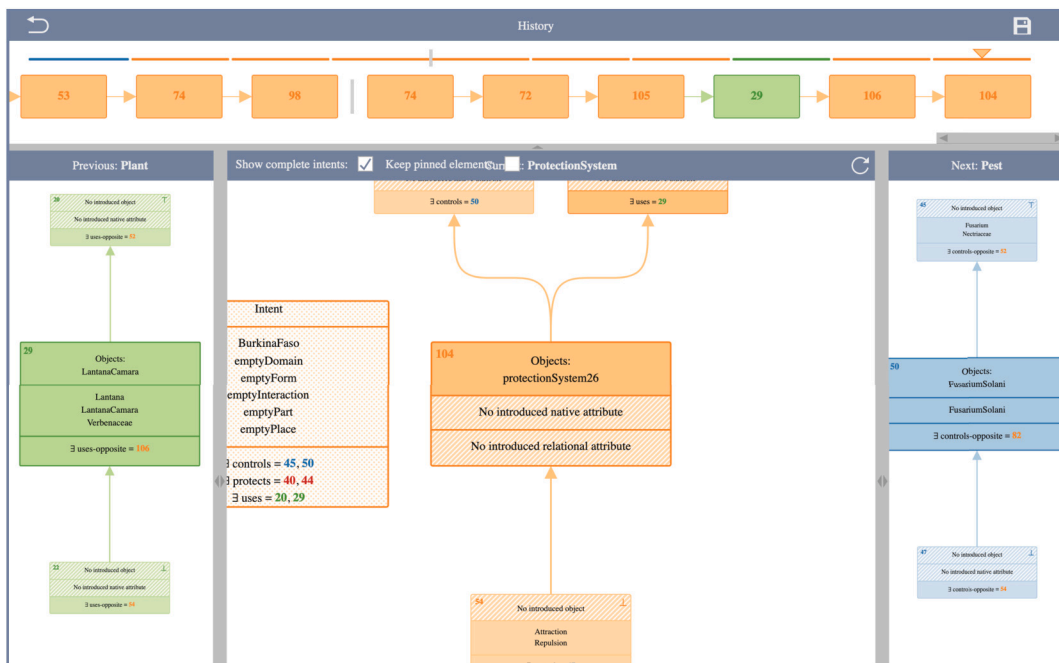


Fig. 21. Step 11. Focus concept 104, which introduces the protection system 26. This system controls Fusarium solani in Burkina Faso, with Lantana camara.

## 6. Conclusion

We presented RCAviz, a web platform dedicated to ES, and more specifically visualize and explore relational conceptual structures computed using RCA. This web platform complements the existing tools, such as Galicia and RCAexplore, which focus on providing RCA algorithms and providing a global visualization. RCAviz is the first RCA tool that considers user interaction with conceptual structures. It proposes advanced solutions for navigating inside a structure or from one structure to another, including an history for an easy back tracking and navigation replaying. RCAviz provides: initial concept selection with queries involving attributes, objects, with formulas including And and Or constructs; local visualization with concept content (introduced objects and attributes,

and optionally inherited ones); local visualization of neighbor concepts; step-by-step navigation among concepts by clicking on the relational attributes; various features to locate the explored part and not be lost, and to navigate forward and backward.

As future work, we plan to integrate additional features. It may seem critical to select a concept in the selector view according to its index number and the selected associated objects and attributes. Thus, we plan to show additional information on the listed concepts, such as their introduced objects and introduced attributes, or their whole extent and intent. Similarly, as selecting a concept in history, only according to its id may be tricky, additional information (e.g. intent/extent) will be presented in a future release.

Other tracks will be considered: specifying the path length to be shown, displaying the complete conceptual structures when they are small, making local computation with specific algorithms [77], visualizing the entity-relationship model, displaying a larger concept neighborhood, or expressing complex queries to initiate the exploration [78]. As implications are relevant companions for the conceptual structure, future development will also consider their visualization.

### CRedit authorship contribution statement

**Marianne Huchard:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Writing – original draft, Writing – review & editing. **Pierre Martin:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Writing – original draft, Writing – review & editing, Data curation, Resources. **Emile Muller:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Software, Writing – review & editing. **Pascal Poncelet:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Writing – original draft, Writing – review & editing. **Vincent Raveneau:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Software, Writing – original draft, Writing – review & editing. **Arnaud Sallaberry:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Writing – original draft, Writing – review & editing.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

Data are available on the RCAviz website. They also will be made available on request to the authors.

### Acknowledgements

The authors warmly thank Alain Gutierrez for his strong support during this work and for enabling FCA4J to generate output files with the JSON data format as required by RCAviz. This work was supported by the French National Research Agency under the Investments for the Future Program, referred as ANR-16-CONV-0004 (#Digitag) and the ANR SmartFCA project, Grant ANR-21-CE23-0023 of the French National Research Agency.

### References

- [1] R.W. White, R.A. Roth, *Exploratory Search: Beyond the Query—Response Paradigm*, Springer International Publishing, Cham, 2009, pp. 9–23, Ch. Defining Exploratory Search.
- [2] T. Jiang, *Library and information sciences: trends and research*, in: Ch. *Exploratory Search: A Critical Analysis of the Theoretical Foundations, System Features, and Research Trends*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 79–103.
- [3] G. Marchionini, *Exploratory search: from finding to understanding*, *Commun. ACM* 49 (4) (2006) 41–46, <https://doi.org/10.1145/1121949.1121979>.
- [4] E. Palagi, F. Gandon, A. Giboin, R. Troncy, *A survey of definitions and models of exploratory search*, in: *Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics, ESIDA '17*, Association for Computing Machinery, New York, NY, USA, 2017, pp. 3–8.
- [5] R.W. White, R.A. Roth, *Exploratory search: beyond the query-response paradigm*, *Synth. Lect. Inf. Concept. Retr. Services* 1 (2009) 98.
- [6] J. Strahl, J. Peltonen, P. Floréen, *Directing and combining multiple queries for exploratory search by visual interactive intent modeling*, in: *Human-Computer Interaction – INTERACT 2021: 18th IFIP TC 13 International Conference, Proceedings, Part III, Bari, Italy, August 30 – September 3, 2021*, Springer-Verlag, Berlin, Heidelberg, 2021, pp. 514–535.
- [7] A. Medlar, J. Li, D. Glowacka, *Query suggestions as summarization in exploratory search*, in: *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval, CHIIR '21*, Association for Computing Machinery, New York, NY, USA, 2021, pp. 119–128.
- [8] M. Lissandrini, D. Mottin, K. Hose, T.B. Pedersen, *Knowledge graph exploration systems: are we lost?*, in: *12th Conference on Innovative Data Systems Research, CIDR 2022*, Online Proceedings, 2022, [www.cidrdb.org](http://www.cidrdb.org).
- [9] S. Metzger, R. Schenkel, M. Sydow, *Qbees: query-by-example entity search in semantic knowledge graphs based on maximal aspects, diversity-awareness and relaxation*, *J. Intell. Inf. Syst.* 49 (3) (2017) 333–366, <https://doi.org/10.1007/s10844-017-0443-x>.
- [10] M. Lissandrini, D. Mottin, T. Palpanas, Y. Velegrakis, *Multi-example search in rich information graphs*, in: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, 2018, pp. 809–820.
- [11] Y. Diao, P. Guzewicz, I. Manolescu, M. Mazuran, *Efficient exploration of interesting aggregates in rdf graphs*, in: *Proceedings of the 2021 International Conference on Management of Data, SIGMOD '21*, Association for Computing Machinery, New York, NY, USA, 2021, pp. 392–404.
- [12] H. Jabeen, D. Graux, G. Sejdin, *Knowledge Graphs and Big Data Processing*, Springer International Publishing, Cham, 2020, pp. 105–121, Ch. Chapter 7 Scalable Knowledge Graph Processing Using SANSa.
- [13] Y. Wang, A. Khan, X. Xu, S. Ye, S. Pan, Y. Zhou, *Approximate and interactive processing of aggregate queries on knowledge graphs: a demonstration*, in: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22*, Association for Computing Machinery, New York, NY, USA, 2022, pp. 5034–5038.



- [14] T. Ruotsalo, J. Peltonen, M.J.A. Eugster, D. Glowacka, P. Floréen, P. Myllymäki, G. Jacucci, S. Kaski, Interactive intent modeling for exploratory search, *ACM Trans. Inf. Syst.* 36 (4) (oct 2018), <https://doi.org/10.1145/3231593>.
- [15] M. Rouane Hacene, M. Huchard, A. Napoli, P. Valtchev, Relational concept analysis: mining concept lattices from multi-relational data, *Ann. Math. Artif. Intell.* 67 (1) (2013) 81–108.
- [16] B. Ganter, R. Wille, *Formal Concept Analysis - Mathematical Foundations*, Springer, 1999.
- [17] A. Braud, X. Dolques, M. Huchard, F. Le Ber, Generalization effect of quantifiers in a classification based on relational concept analysis, *Knowl.-Based Syst.* 160 (2018) 119–135, <https://doi.org/10.1016/j.knosys.2018.06.011>.
- [18] S. Ferré, Conceptual navigation in large knowledge graphs, in: Springer (Ed.), *Complex Data Analytics with Formal Concept Analysis*, Springer, 2022, pp. 1–30.
- [19] E. Muller, M. Huchard, P. Martin, P. Poncelet, A. Sallaberry, Rcaviz: visualizing and exploring relational conceptual structures, in: P. Cordero, O. Křídlo (Eds.), *Proceedings of the Sixteenth International Conference on Concept Lattices and Their Applications (CLA 2022)*, Tallinn, Estonia, June 20–22, 2022, in: CEUR Workshop Proceedings, vol. 3308, CEUR-WS.org, 2022, pp. 133–146, <http://ceur-ws.org/Vol-3308/Paper11.pdf>.
- [20] P.J. Silvie, P. Martin, M. Huchard, P. Keip, A. Gutierrez, S. Sarter, Prototyping a knowledge-based system to identify botanical extracts for plant health in sub-Saharan Africa, *Plants* 10 (5) (2021), <https://doi.org/10.3390/plants10050896>, <https://www.mdpi.com/2223-7747/10/5/896>.
- [21] G. Birkhoff, *Lattice Theory*, Colloquium Publications, American Mathematical Society, 1940.
- [22] O. Ore, Galois connexions, *Trans. Am. Math. Soc.* 55 (1944) 493–513.
- [23] M. Barbut, B. Monjardet, *Ordre et classification – Algèbre et combinatoire*, Tome 2, Hachette, Paris, 1970.
- [24] T. Tilley, R. Cole, P. Becker, P.W. Eklund, A survey of formal concept analysis support for software engineering activities, in: *Formal Concept Analysis, Foundations and Applications*, 2005, pp. 250–271.
- [25] J. Poelmans, D.I. Ignatov, S.O. Kuznetsov, G. Dedene, Formal concept analysis in knowledge processing: a survey on applications, *Expert Syst. Appl.* 40 (16) (2013) 6538–6560, <https://doi.org/10.1016/j.eswa.2013.05.009>.
- [26] A. Gutierrez, M. Huchard, P. Martin, FCA4J: a Java library for relational concept analysis and formal concept analysis, in: P. Cordero, O. Křídlo (Eds.), *Proceedings of the Sixteenth International Conference on Concept Lattices and Their Applications (CLA 2022)*, Tallinn, Estonia, June 20–22, 2022, in: CEUR Workshop Proceedings, vol. 3308, CEUR-WS.org, 2022, pp. 207–212, <http://ceur-ws.org/Vol-3308/Paper17.pdf>.
- [27] A. Berry, A. Gutierrez, M. Huchard, A. Napoli, A. Sigayret, Hermes: a simple and efficient algorithm for building the aoc-poset of a binary relation, *Ann. Math. Artif. Intell.* 72 (1–2) (2014) 45–71, <https://doi.org/10.1007/s10472-014-9418-6>.
- [28] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal, Computing iceberg concept lattices with Titanic, *Data Knowl. Eng.* 42 (2) (2002) 189–222, [https://doi.org/10.1016/S0169-023X\(02\)00057-5](https://doi.org/10.1016/S0169-023X(02)00057-5).
- [29] R. Belohlávek, V. Vychodil, What is a fuzzy concept lattice?, in: R. Belohlávek, V. Snásel (Eds.), *Proceedings of the CLA 2005 International Workshop on Concept Lattices and Their Applications Olomouc*, Czech Republic, September 7–9, 2005, in: CEUR Workshop Proceedings, vol. 162, CEUR-WS.org, 2005, <http://ceur-ws.org/Vol-162/paper4.pdf>.
- [30] C. Carpineto, G. Romano, *Concept Data Analysis: Theory and Applications*, John Wiley & Sons, Chichester, England, 2004.
- [31] S. Ferré, O. Ridoux, A logical generalization of formal concept analysis, in: B. Ganter, G.W. Mineau (Eds.), *Conceptual Structures: Logical, Linguistic, and Computational Issues*, 8th International Conference on Conceptual Structures, ICCS 2000, Proceedings, Darmstadt, Germany, August 14–18, 2000, in: *Lecture Notes in Computer Science*, vol. 1867, Springer, 2000, pp. 371–384.
- [32] F. Lehmann, R. Wille, A triadic approach to formal concept analysis, in: G. Ellis, R. Levinson, W. Rich, J.F. Sowa (Eds.), *Conceptual Structures: Applications, Implementation and Theory*, Third International Conference on Conceptual Structures, ICCS '95, Proceedings, Santa Cruz, California, USA, August 14–18, 1995, in: *Lecture Notes in Computer Science*, vol. 954, Springer, 1995, pp. 32–43.
- [33] G. Voussadakakis, Polyadic concept analysis, *Order* 19 (3) (2002) 295–304, <https://doi.org/10.1023/A:1021252203599>.
- [34] S. Ferré, P. Cellier, Graph-fca: an extension of formal concept analysis to knowledge graphs, *Discrete Appl. Math.* 273 (2020) 81–102, <https://doi.org/10.1016/j.dam.2019.03.003>.
- [35] S. Boffa, Extracting concepts from fuzzy relational context families, *IEEE Trans. Fuzzy Syst.* 31 (4) (2023) 1202–1213, <https://doi.org/10.1109/TFUZZ.2022.3197826>.
- [36] J. Kötters, P.W. Eklund, Conjunctive query pattern structures: a relational database model for formal concept analysis, *Discrete Appl. Math.* 273 (2020) 144–171, <https://doi.org/10.1016/j.dam.2019.08.019>.
- [37] P. Keip, S. Ferré, A. Gutierrez, M. Huchard, P.J. Silvie, P. Martin, Practical comparison of FCA extensions to model indeterminate value of ternary data, in: F.J. Valverde-Albacete, M. Trnecka (Eds.), *Proceedings of the Fifteenth International Conference on Concept Lattices and Their Applications*, Tallinn, Estonia, June 29–July 1, 2020, in: CEUR Workshop Proceedings, vol. 2668, CEUR-WS.org, 2020, pp. 197–208, <http://ceur-ws.org/Vol-2668/paper15.pdf>.
- [38] R. Godin, H. Mili, Building and maintaining analysis-level class hierarchies using Galois lattices, in: T. Babitsky, J. Salmons (Eds.), *Proceedings of the Eighth Annual Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA 1993*, Washington, DC, USA, September 26–October 1, 1993, ACM, 1993, pp. 394–410.
- [39] M. Huchard, M. Rouane Hacene, C. Roume, P. Valtchev, Relational concept discovery in structured datasets, *Ann. Math. Artif. Intell.* 49 (1–4) (2007) 39–76, <https://doi.org/10.1007/s10472-007-9056-3>.
- [40] N. Mimouni, M. Fernández, A. Nazarenko, D. Bourcier, S. Salotti, A relational approach for information retrieval on XML legal sources, in: E. Francesconi, B. Verheij (Eds.), *International Conference on Artificial Intelligence and Law, ICAIL '13*, Rome, Italy, June 10–14, 2013, ACM, 2013, pp. 212–216.
- [41] V. Codocedo, A. Napoli, Formal concept analysis and information retrieval - a survey, in: J. Baixeries, C. Sacarea, M. Ojeda-Aciego (Eds.), *Formal Concept Analysis - 13th International Conference, ICFA 2015*, Proceedings, Nerja, Spain, June 23–26, 2015, in: *Lecture Notes in Computer Science*, vol. 9113, Springer, 2015, pp. 61–77.
- [42] A.M. Rouane Hacene, A. Napoli, P. Valtchev, Y. Toussaint, R. Bendaoud, Ontology learning from text using relational concept analysis, in: *International MCETECH Conference on e-Technologies - MCETECH 2008*, Montréal, Canada, 2008, <https://hal.inria.fr/inria-00322007>.
- [43] S. Kasri, F. Benchikha, Refactoring ontologies using design patterns and relational concepts analysis to integrate views: the case of tourism, *Int. J. Metadata Semant. Ontol.* 11 (4) (2016) 243–263, <https://doi.org/10.1504/IJMSO.2016.10004260>.
- [44] M. Rouane Hacene, M. Huchard, A. Napoli, P. Valtchev, Using formal concept analysis for discovering knowledge patterns, in: M. Kryszkiewicz, S.A. Obiedkov (Eds.), *Proceedings of the 7th International Conference on Concept Lattices and Their Applications*, Sevilla, Spain, October 19–21, 2010, in: CEUR Workshop Proceedings, vol. 672, CEUR-WS.org, 2010, pp. 223–234, <http://ceur-ws.org/Vol-672/paper20.pdf>.
- [45] M. Atencia, J. David, J. Euzenat, A. Napoli, J. Vizzini, Link key candidate extraction with relational concept analysis, *Discrete Appl. Math.* 273 (2020) 2–20, <https://doi.org/10.1016/j.dam.2019.02.012>.
- [46] C. Nica, A. Braud, F. Le Ber, Rca-seq: an original approach for enhancing the analysis of sequential data based on hierarchies of multilevel closed partially-ordered patterns, *Discrete Appl. Math.* 273 (2020) 232–251, <https://doi.org/10.1016/j.dam.2019.02.037>.
- [47] L. Mahrach, A. Gutierrez, M. Huchard, P. Keip, P. Marnotte, P.J. Silvie, P. Martin, Combining implications and conceptual analysis to learn from a pesticidal plant knowledge base, in: T. Braun, M. Gehrke, T. Hanika, N. Hernandez (Eds.), *Graph-Based Representation and Reasoning - 26th International Conference on Conceptual Structures, ICCS 2021*, Virtual Event, Proceedings, September 20–22, 2021, in: *Lecture Notes in Computer Science*, vol. 12879, Springer, 2021, pp. 57–72.
- [48] A. Braud, X. Dolques, A. Gutierrez, M. Huchard, P. Keip, F. Le Ber, P. Martin, C. Nica, P.J. Silvie, Dealing with large volumes of complex relational data using RCA, in: R. Missaoui, L. Kwuida, T. Abdesslem (Eds.), *Complex Data Analytics with Formal Concept Analysis*, Springer International Publishing, 2022, pp. 105–134.

- [49] X. Dolques, M. Huchard, C. Nebut, P. Reitz, Fixing generalization defects in UML use case diagrams, *Fundam. Inform.* 115 (4) (2012) 327–356, <https://doi.org/10.3233/FI-2012-658>.
- [50] A.O. Guédi, A. Miralles, M. Huchard, C. Nebut, A practical application of relational concept analysis to class model factorization: lessons learned from a thematic information system, in: M. Ojeda-Aciego, J. Outrata (Eds.), *Proceedings of the Tenth International Conference on Concept Lattices and Their Applications*, La Rochelle, France, October 15–18, 2013, in: *CEUR Workshop Proceedings*, vol. 1062, CEUR-WS.org, 2013, pp. 9–20, <http://ceur-ws.org/Vol-1062/paper1.pdf>.
- [51] N. Moha, M. Rouane Hacene, P. Valtchev, Y. Guéhéneuc, Refactorings of design defects using relational concept analysis, in: R. Medina, S.A. Obiedkov (Eds.), *Formal Concept Analysis, 6th International Conference, ICFA 2008, Proceedings*, Montreal, Canada, February 25–28, 2008, in: *Lecture Notes in Computer Science*, vol. 4933, Springer, 2008, pp. 289–304.
- [52] X. Dolques, M. Huchard, C. Nebut, P. Reitz, Learning transformation rules from transformation examples: an approach based on relational concept analysis, in: *Workshops Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference, EDOCW 2010*, Vitória, Brazil, 25–29 October 2010, IEEE Computer Society, 2010, pp. 27–32.
- [53] Z. Azme, M. Driss, F. Hamoui, M. Huchard, N. Moha, C. Tibermacine, Selection of composable web services driven by user requirements, in: *IEEE International Conference on Web Services, ICWS 2011*, Washington, DC, USA, July 4–9, 2011, IEEE Computer Society, 2011, pp. 395–402.
- [54] J. Carbonnel, M. Huchard, C. Nebut, Modelling equivalence classes of feature models with concept lattices to assist their extraction from product descriptions, *J. Syst. Softw.* 152 (2019) 1–23, <https://doi.org/10.1016/j.jss.2019.02.027>.
- [55] A. Waffo Kouhoué, Y. Bonavero, T. Bouétou Bouétou, M. Huchard, Exploring variability of visual accessibility options in operating systems, *Future Internet* 13 (9) (2021) 230, <https://doi.org/10.3390/fi13090230>.
- [56] R. Al-Msie'deen, A. Seriai, M. Huchard, C. Urtado, S. Vauttier, Documenting the mined feature implementations from the object-oriented source code of a collection of software product variants, in: M.Z. Reformat (Ed.), *The 26th International Conference on Software Engineering and Knowledge Engineering*, Hyatt Regency, Vancouver, BC, Canada, July 1–3, 2013, Knowledge Systems Institute Graduate School, 2014, pp. 138–143.
- [57] N. Hlad, B. Lemoine, M. Huchard, A. Seriai, Leveraging relational concept analysis for automated feature location in software product lines, in: E. Tilevich, C.D. Roover (Eds.), *GPCE '21: Concepts and Experiences*, Chicago, IL, USA, October 17–18, 2021, ACM, 2021, pp. 170–183.
- [58] M. Wajnberg, M. Lezoche, A. Blondin-Massé, P. Valchev, H. Panetto, L. Tyvaert, Semantic interoperability of large systems through a formal method: relational concept analysis, in: 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018, *IFAC-PapersOnLine* 51 (11) (2018) 1397–1402, <https://doi.org/10.1016/j.ifacol.2018.08.330>.
- [59] M. Wajnberg, P. Valtchev, M. Lezoche, A.B. Massé, H. Panetto, Concept analysis-based association mining from linked data: a case in industrial decision making, in: *Proc. of the Joint Ontology Works. 2019 Episode V: The Styrian Autumn of Ontology*, in: *CEUR Workshop Proceedings*, vol. 2518, CEUR-WS.org, 2019, <http://ceur-ws.org/Vol-2518/paper-DAOSI3.pdf>.
- [60] S.A. Yevtushenko, Conexp, <http://conexp.sourceforge.net/>, 2018.
- [61] M. Alam, T.N.N. Le, A. Napoli, Latviz: a new practical tool for performing interactive exploration over concept lattices, in: *Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications*, Moscow, Russia, July 18–22, 2016, in: *CEUR Workshop Proceedings*, vol. 1624, CEUR-WS.org, 2016, pp. 9–20.
- [62] T. Pattison, Interactive visualization of formal concept lattices, in: *ED/GVIP@Diagrams*, in: *CEUR Workshop Proceedings*, vol. 1244, CEUR-WS.org, 2014, pp. 78–89.
- [63] S. Andrews, L. Hirsch, A tool for creating and visualising formal concept trees, in: *CSTIW@ICCS*, in: *CEUR Workshop Proceedings*, vol. 1637, 2016, pp. 1–9.
- [64] M. Alam, M. Osmuk, A. Napoli, RV-Explorer: a way to navigate lattice-based views over RDF graphs, in: *Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications*, Clermont-Ferrand, France, October 13–16, 2015, in: *CEUR Workshop Proceedings*, vol. 1466, CEUR-WS.org, 2015, pp. 23–34.
- [65] C. Carpineto, G. Romano, Exploiting the potential of concept lattices for information retrieval with CREDO, *J.UCS* 10 (8) (2004) 985–1013.
- [66] J. Ducrou, P. Eklund, SearchSleuth: the conceptual neighbourhood of a Web query, in: *Proceedings of the 2007 International Conference on Concept Lattices and their Applications (CLA '07)*, in: *CEUR Workshop Proceedings*, ISSN 1613-0073, vol. 331, 2007, pp. 249–259.
- [67] S. Ferré, Sparklis: an expressive query builder for SPARQL endpoints with guidance in natural language, *Semant. Web* 8 (3) (2017) 405–418.
- [68] G.J. Greene, M. Esterhuizen, B. Fischer, Visualizing and exploring software version control repositories using interactive tag clouds over formal concept lattices, *Inf. Softw. Technol.* 87 (2017) 223–241.
- [69] S. Ferré, CAMELIS: organizing and browsing a personal photo collection with a logical information system, in: *Proceedings of the 2007 International Conference on Concept Lattices and their Applications (CLA '07)*, Montpellier, France, in: *CEUR Workshop Proceedings*, ISSN 1613-0073, vol. 331, 2007, pp. 108–119.
- [70] M. Crampes, M. Plantié, Visualizing and interacting with concept hierarchies, in: *Proc. of WIMS14*, ACM, 2014.
- [71] P. Healy, N.S. Nikolov, Hierarchical drawing algorithms, in: R. Tamassia (Ed.), *Handbook on Graph Drawing and Visualization*, Chapman and Hall/CRC, 2013, pp. 409–453.
- [72] T. Munzner, A nested process model for visualization design and validation, *IEEE Trans. Vis. Comput. Graph.* 15 (6) (2009) 921–928, <https://doi.org/10.1109/TVCG.2009.111>.
- [73] M. Sedlmair, M.D. Meyer, T. Munzner, Design study methodology: reflections from the trenches and the stacks, *IEEE Trans. Vis. Comput. Graph.* 18 (12) (2012) 2431–2440, <https://doi.org/10.1109/TVCG.2012.213>.
- [74] T. Munzner, *Visualization Analysis and Design*, A.K. Peters Visualization Series, CRC Press, 2014.
- [75] M. Bostock, V. Ogievetsky, J. Heer, D<sup>3</sup> data-driven documents, *IEEE Trans. Vis. Comput. Graph.* 17 (12) (2011) 2301–2309, <https://doi.org/10.1109/TVCG.2011.185>.
- [76] P. Keip, A. Ouzerdine, M. Huchard, P.J. Silvie, P. Martin, Navigation conceptuelle dans une base de connaissances sur l'usage des plantes en santé animale et végétale, in: P. Bellot, L. Soulier (Eds.), *Conférence en Recherche d'Informations et Applications - CORIA 2019, 16th French Information Retrieval Conference*, Proceedings, ARIA, 2019.
- [77] A. Bazin, J. Carbonnel, M. Huchard, G. Kahn, P. Keip, A. Ouzerdine, On-demand relational concept analysis, in: *ICFA 2019*, in: *LNCS*, vol. 11511, Springer, 2019, pp. 155–172.
- [78] Z. Azme, M. Huchard, A. Napoli, M. Rouane Hacene, P. Valtchev, Querying relational concept lattices, in: A. Napoli, V. Vychodil (Eds.), *Proceedings of the Eighth International Conference on Concept Lattices and Their Applications*, Nancy, France, October 17–20, 2011, in: *CEUR Workshop Proceedings*, vol. 959, CEUR-WS.org, 2011, pp. 377–392, <http://ceur-ws.org/Vol-959/paper26.pdf>.