A Survey of Hybrid Representation of Concept Lattices in Conceptual Knowledge Processing

Peter Eklund, Jean Villerd

A Survey of Hybrid Representations of Concept Lattices in Conceptual Knowledge Processing

Peter Eklund[1] and Jean Villerd[2]

[1] School of Information Systems and Technology
University of Wollongong, Australia
peklund@uow.edu.au
[2] LORIA – INRIA Nancy - Grand Est Research Centre
Nancy, France
jean.villerd@loria.fr

Abstract. A feature of Formal Concept Analysis is the use of the line diagram of the concept lattice to visualize a conceptual space. The line diagram is a specialized form of Hasse diagram labeled with the object extents and the attribute intents. The line diagram is usually drawn so that its rendering maximizes symmetry and minimizes edge crossings. Further the line diagram is usually layered hierarchically from top to bottom. Variations of the line diagram are frowned upon in the mathematical treatment of Formal Concept Analysis but hybrid presentations of concept lattices have practical value when used in an appropriate application context. This paper surveys previous work on using line diagrams and further explores hybrid visual representations of concept lattices. It identifies connections to other visual information techniques in data mining and information visualisation that can be used to enhance Formal Concept Analysis applications.

1 Introduction

In the last three decades Formal Concept Analysis (FCA) [30] has grown in popularity as a method for data analysis and knowledge representation. One reason is the ability to visualize an information space as a concept lattice or line diagram. Consider the formal context in Fig. 1 and its corresponding line diagram. Fig. 1 presents subtle differences from the standard diagrammatic representation of the line diagram, it includes the use of graduated color on the vertices representing the cardinality of the object extents (the vertex at the top of the lattice being darkest and the bottom node of the lattice being the lightest). This is a minor departure from the conventions of line diagram drawing and this paper explores other more radical ideas. Our purpose with this survey is to look at hybrid visual representations of concept lattices because we believe experimentation in this direction will lead to innovations that enhance the application of Conceptual Knowledge Processing (CKP) [32] for knowledge and data discovery. As well as being useful in various applications, we argue that these departures are sanctioned by Wille's vision of CKP.

	Conceptual Knowledge Processing Method Name	Reference
	Representing a Context by a Cross Table	(M1.1)
	Clarifying a Context	(M1.2)
	Reducing a Context	(M1.3)
→	**Representation of a Concept Hierarchy by a Line Diagram**	**(M1.4)**
→	**Checking a Line Diagram of a Concept Hierarchy**	**(M1.5)**
	Dualizing a Concept Hierarchy	(M1.6)
	Generating Concepts	(M2.1)
→	**Generating All Concepts Within a Line Diagram**	**(M2.2)**
	Determining All Concepts of a Context	(M2.3)
	Determining a Context from an Ordered Collection of Ideas	(M2.4)
	Conceptual Scaling of a Context	(M3.1)
	Conceptual Scaling of a Many-valued Context	(M3.2)
	Nominal Scaling of a Many-valued Context	(M3.3)
	Ordinal Scaling of a Many-valued Context	(M3.4)
	Interordinal Scaling of a Many-valued Context	(M3.5)
	Contraordinal Scaling of a Many-valued Context	(M3.6)
	Concept Classification of Objects	(M4.1)
	Many-valued Classification of Objects	(M4.2)
→	**Partitioning Attributes of a Context (Nested Line Diagrams)**	**(M5.1)**
	Atlas-Decomposition of a Concept Hierarchy	(M5.2)
	Concept Patterns in a Concept Hierarchy	(M5.3)
	Juxtaposition of Contexts with Common Object Collection	(M6.1)
	Aggregation Based on Object Families	(M6.2)
→	**TOSCANA-Aggregation of Concept Hierarchies**	**(M6.3)**
	Identifying a Concept	(M7.1)
	Identifying Concept Patterns	(M7.2)
	Determining the Attribute Implications of a Context	(M8.1)
	Determining Many-valued Attribute Dependencies	(M8.2)
	Attribute Exploration	(M9.1)
	Concept Exploration	(M9.2)
	Discovering Association Rules	(M9.3)
	Retrieval with Contexts and Concept Hierarchies	(M10.1)
→	**Retrieval with a TOSCANA-System**	**(M10.2)**
	Theory Building with Concept Hierarchies	(M11.1)
	Theory Building with TOSCANA	(M11.2)
	Conceptual Graphs Derived from Natural Language	(M12.1)
	Derivation of Judgments from Power Context Families	(M12.2)

Table 1. Methods for Conceptual Knowledge Processing as per Wille [32], in this paper we focus on those methods which touch on line diagrams of a concept lattice, namely those arrowed (→) and in bold, (line diagrams are also significantly used in conceptual scaling, methods M3.1–M.3.6).

2 Representing Line Diagrams

Because this paper concentrates on diagrams from Conceptual Knowledge Processing (CKP) we recap Wille's [32] methods and focus on methods that involve visualizing concept lattices. Wille generalizes CKP from Formal Concept Analysis (FCA), FCA being the mathematization of CKP. Table 1 lists the methods

of CKP showing that the analysis framework has formalized methods, processes and algorithms. Referring to Table 1 we consider only the highlighted methods, those that relate to line diagrams of the concept lattice and elaborate each in turn. The description of the methods is abbreviated from Wille [32] with our commentary.

	small	medium	large	near	far	moon(s)	no_moon
Mercury	×			×			×
Venus	×			×			×
Earth	×			×		×	
Mars	×			×		×	
Jupiter			×		×	×	
Saturn			×		×	×	
Uranus		×			×	×	
Neptune		×			×	×	
Pluto	×				×	×	

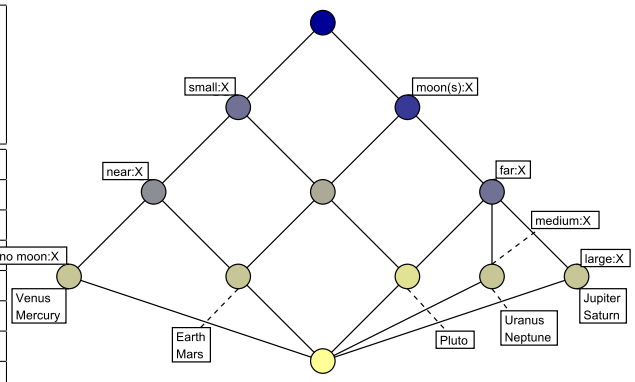


Fig. 1. A cross-table containing information about the planets and the corresponding line diagram of the concept lattice induced by the formal context.

2.1 Representation a Concept Hierarchy by a Line Diagram (M1.4)

Concepts are represented by circles so that upward line segments between them indicate a sub-superconcept relation. Every circle representing a concept generated by an object/attribute has attached from below/above the name of that object/attribute. The labels of object and attribute names allow one to read off the extension and intension of each concept from the line diagram. Starting from any given concept, the extension/intension of a concept consists of all those objects/attributes the names of which are attached to a circle belonging to a downward/upward path of line segments [32].

2.2 Checking a Line Diagram of a Concept Hierarchy (M1.5)

A line diagram represents a concept hierarchy iff the line diagram satisfies: (1) each circle starting a downward line segment must have attached an object name; (2) each circle starting an upward line segment must have attached an attribute name; (3) an object g has an attribute m in the given context iff the names of g and m are attached to the same circle or there is an upward path of line segments from the circle with the name of g to the circle with the name of m; (4) The Basic Theorem on Concept Lattices holds: namely the lattice is complete [32].

The conditions on **(M1.4)** and **(M1.5)** relate to drawing line diagrams and are quite strict as to what constitutes a line diagram of a concept lattice. In [32], Wille describes algorithmic methods for Conceptual Knowledge Processing including an algorithmic description for generating the line diagram in **(M2.2)**.

ConImp [2] was developed in the mid-80s and is the progenitor FCA program but has no capability for lattice rendering, the program is used to manipulate formal contexts and compute concept listings from which a line diagram can be then be drawn by hand. Because ConImp had no automatic mechanism for rendering line diagrams, it encouraged a craft of lattice drawing and certain conventions emerged to inform the process of drawing the line diagram of a concept lattice. Drawing a concept lattice became somewhat of a art with its own conventions and is the main reason that the methods **(M1.4)**, **(M1.5)** and **(M2.2)** from Table 1 are codified in the prescriptive way Wille describes.

2.3 Generating All Concepts Within a Line Diagram: (M2.2)

First, represent the concept having the full object set as its extension by a small circle and attach (from above) the names of all attributes that apply to all these objects. Secondly, choose from the remaining attributes the extension of which are maximal, draw for each of them a circle below, link them to the parent circle by a line segment, and attach (from above) the corresponding attribute names. Then determine all intersections of the extensions of the existing concepts and represent the concepts generated by those intersections by small circles with their respective line segments representing the subconcept-superconcept-relationships. Perform analogously until all attributes are treated. Finally, attach each object name (from below) to that circle from which upward paths of line segments lead exactly to those circles with attached names of attributes applying to the respective object ([33], p.64ff.) [32].

Taken in combination **(M1.4)** **(M1.5)** and **(M2.2)** provide prescriptive detail about the graphical elements to use in the construction of the line diagram, namely small circles and line segments, labels of attributes and objects. It also codifies a top-to-bottom rendering of the conceptual hierarchy. On the other hand, the methods say nothing about layout or the length of line segments. Despite small-scale success with hand-drawing line diagrams. the process of creating and laying-out a line diagram has frustrated full automation, *"... up to now, no universal method is known for drawing well readable line diagrams representing concept hierarchies. For smaller concept hierarchies, the method of Drawing an Additive Line Diagram (see [33], p.75) often leads to well-structured line diagrams. This is the reason that quite a number of computer programs for drawing concept hierarchies use that method (e.g.* Anaconda, Cernato, ConExp, Elba*)"[32].* The result is that drawing line diagrams can only be partially automated and the diagrams improved by direct manipulation by humans.

2.4 Drawing line diagrams

In practice two main approaches for drawing line diagrams can be identified. The first approach is a vector-based method that uses the notion of additive line diagram [33] and aims to optimize the interpretability of the diagram with respect to the original context. Attribute and/or object concepts – the labeled circles described in **(M1.4)** – are drawn first, determining the positions of the remaining

concepts. The second approach considers the line diagram as a directed graph, and adapts existing graph drawing techniques for producing aesthetics diagrams by, among other things, minimizing edge crossings and maximizing symmetry. In this case the order relation between concepts is the only information taken into account, and all concepts are of the same importance.

Additive line diagrams approach Ganter and Wille [33] specify an additive line diagram associated to a concept lattice $\underline{\mathfrak{B}}(G, M, I)$ with the help of two functions:

1. A representation function $\text{rep} : \underline{\mathfrak{B}}(G, M, I) \to \mathfrak{P}(X)$ that assigns to each concept a subset of a representation set X. This function must preserve the partial order of the lattice, i.e. $c_i \leqslant c_j \Leftrightarrow \text{rep}(c_i) \subseteq \text{rep}(c_j)$ for two concepts c_i and c_j. Commonly, the attribute set M is used as the representation set and then we have $\text{rep}(c) = \text{Int}(c)$.
2. A grid projection $\text{vec} : X \to \mathbb{R}^2$ assigning a real vector with a positive y coordinate to each element of X (commonly to each attribute). Then the position of a concept c on the plane follows $\text{pos}(c) = n + \sum_{x \in \text{rep}(c)} \text{vec}(x)$ where the vector n is used to shift the location of the lattice on the display.

Additive line diagrams have the advantage of being tunable through the choice of the representation set, and the technique produces a great number of parallel edges, that in turn improves readability.

Force-directed approach A common graph drawing technique used in the second approach is the force-directed method. The basic idea is to use a physical analogy to model the graph drawing problem. A system of forces is applied to the vertices, which move until a configuration that minimizes the potential energy of the system is reached. An advantage of this method is that it often reveals symmetries existing in the graph. A well-known implementation of this method is Eades' *Spring embedder* model [12], in which two types of forces operate. An attraction force, which makes edges act as springs, that follows a logarithmic force law; and a repulsion force which makes vertices act like positive electrical point charges that repel each other to avoid overlapping, following an inverse-square force law.

In particular, following these ideas Freese [19] proposed a force-directed method for drawing lattices. In order to preserve the top-to-bottom rendering, the y coordinates of the vertices/concepts are computed w.r.t. their depth in the lattice, using a layering approach. Then the forces only impact the x coordinates. The attraction force acts between comparable concepts, so that a concept remains close to its predecessors, and the repulsion force acts between incomparable concepts of the same layer in order to avoid overlapping. Cole [4, 6] combined the force-directed and the additive approaches and introduced several metrics that quantify the *goodness* of a diagram layout.

In the layout strategies mentioned above, no semantics is attached to the length of the edges nor to the distances between concepts in general, and nothing

is said about this in **(M1.4) (M1.5)** and **(M2.2)**. However, one can anticipate a layout strategy where the distances between vertices reflect inversely on the similarity between concepts. This can be done using Kamada and Kamai's force directed approach [22] that modifies the *Spring embedder* model by removing the repulsion force and extending the attraction force to all pairs of vertices, i.e. a spring is attached to all pairs of vertices, whenever an edge exists between the two vertices or not. The attraction force follows Hooke's law, so that the force exerted on the vertices is proportional to the difference between the spring's rest length and the actual distance between the vertices. Each iteration refines the layout of the graph, until the system converges to a state of minimal energy. Although the system may fall into a local optimum, the final layout is intended to provide a faithful representation of the distances between vertices. This approach can also be applied in Multidimensional scaling (MDS) where the layout of high-dimensional objects in a low-dimensional projection is the goal.

Several similarity measures between formal concepts have been proposed, including [23, 27, 21] and some have been applied and studied in [8]. These are based on the symmetric difference between sets and differ in the type of sets involved (extents and/or intents), and in their sensitivity (local dissimilarity or general similarity involving all the objects and/or attributes of the context). Note that all these metrics are distances, a nice property for layout. To our knowledge, only Hannan and Pogel [21] used distance for the purposes of layout as is shown in Fig. 2. In their work, the distance between two concepts c_i and c_j is the cardinal of the symmetric difference between respective extents, i.e. $d(c_i, c_j) = |\mathrm{Ext}(c_i) \triangle \mathrm{Ext}(c_j)| = |(\mathrm{Ext}(c_i) \setminus \mathrm{Ext}(c_j)) \cup (\mathrm{Ext}(c_j) \setminus \mathrm{Ext}(c_i))|$ with the aim to improve the layout of additive line diagrams by applying a force-directed post-processing based on this distance. The final layout – an example of which is Fig. 2 – shows a diagram in which, the more concepts share common objects the closer they appear. Thus, considering two concepts drawn near one another, the association rules between their respective intents may be of high confidence, suggesting visual discovery of *almost exact* rules from layout.

The attributes (resp. contexts) considered so far are binary (i.e. one-valued contexts). In order to handle many-valued contexts containing non-binary attributes such as *gender*, *colour*, *grade* or *age*, a discretization process called *conceptual scaling* is used **(M3.2)**. The core idea is to transform the many-valued context into a *derived* one-valued context according to a set of rules that depends both on the nature of the many-valued attributes (nominal, ordinal or numerical) and on some potential background knowledge about relations between their values. The *conceptual scaling* mechanism itself is out of the scope of the present paper. However since it results in the creation of several new binary attributes for each many-valued attribute, the number of concepts extracted from the derived one-valued context may grow exponentially (in the worst case, $2^{\min(|G|,|M|)}$ concepts are extracted from a context with $|G|$ objects and $|M|$ attributes), leading to unreadable line diagrams. The following section deals with nested line diagrams, an alternative technique to represent and study large contexts by partitioning the attributes.

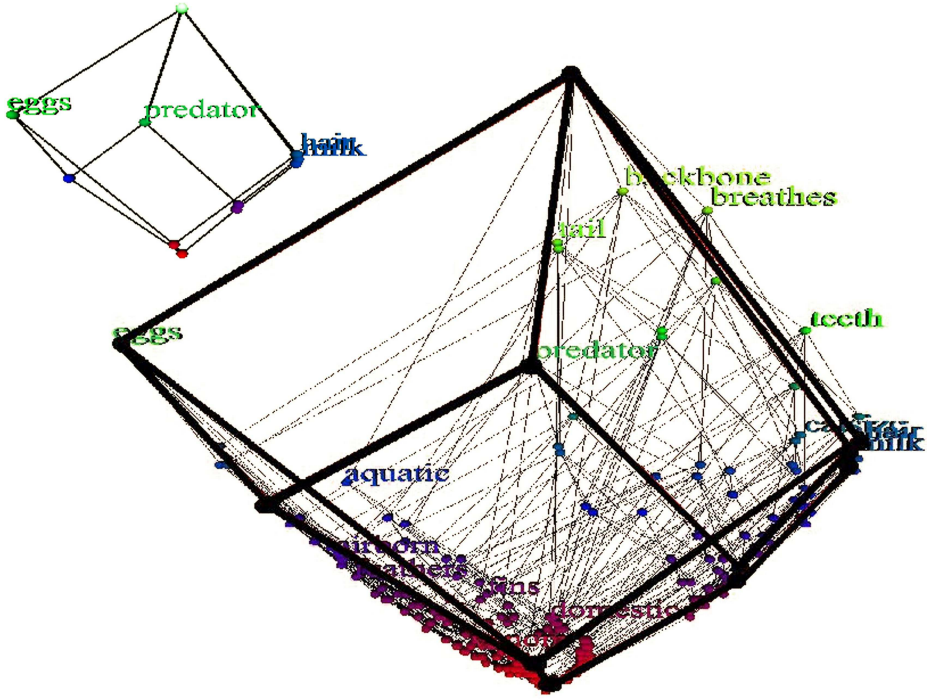


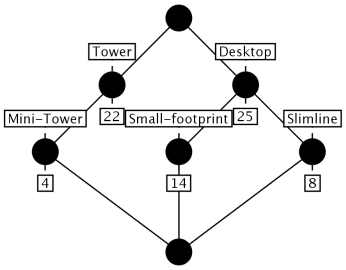
Fig. 2. The concept lattice of the UCI zoological dataset of 101 animals and 15 binary attributes. This lattice has 238 concepts. The concepts "milk" and "hair" are drawn very near each other (the attribute labels overlap – extreme right), while the concept "eggs" is repulsed from the pair – left. The rules "milk"→"hair" and "hair"→"milk", have confidence 95.1% and 90.6% resp. Visual clustering of "milk" and "hair" suggest they are minor variations of one another (re-printed with permission).

2.5 Partition Attributes of a Context (Nested line diagrams (M5.1))

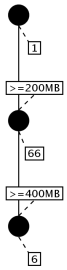
A concept hierarchy can be visualized as a nested line diagram constructed as follows (cf. [33], p.75ff.): First, line diagrams of the concept lattices of the subcontexts are prepared and ordered in a sequence of subcontexts. Then, the line diagram being second in the sequence is copied into each circle of the line diagram; next, the line diagram being third in the sequence is copied into each circle of the line diagram second in the sequence and so on [32].

In TOSCANAJ [1] histograms were used to represent object extents and clustergrams used to display attributes. A key question arising from **(M5.1)** is whether is makes sense to embed a visualization of anything other than a line diagram within a nested line diagram.

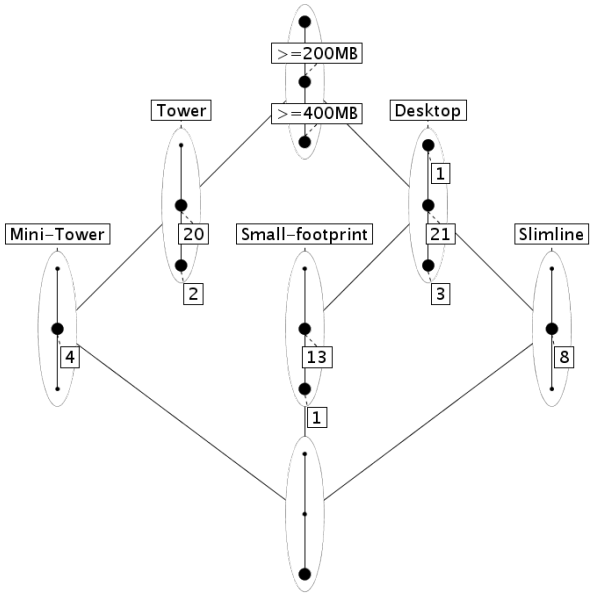
Nested line diagrams are particularly suited to contexts derived from many-valued contexts. Fig. 3 shows the building of a nested line diagram from a many-valued context containing computers as objects and two many-valued attributes *chassis type* and *HDD capacity* (computed using TOSCANAJ). Each of these attributes has been scaled into a set of binary attributes that constitute two



(a) *chassis type* line diagram



(b) *HDD capacity* line diagram



(c) resulting nested line diagram

Fig. 3. A nested line diagram (c) built the from two subcontexts *chassis types* (a) and *HDD storage* (b) Numbers represent extent cardinality.

subcontexts. Following the above instructions, the line diagrams of the two subcontexts are drawn (see Fig. 3(a) and 3(b)), and then the second is copied into each circle of the first (Fig. 3(c)). One can easily observe the distribution of HDD capacities along with the type of chassis. However, two limitations arise: (i) combining more than two many-valued attributes may result in complex nested line diagrams, useful for deep analysis but not suitable for a first-glance navigation; (ii) the scaling pre-processing inevitably results in a loss of precision concerning numerical attributes. Hence, values for the numerical attribute *HDD capacity* have been gathered into two binary attributes, namely $\geqslant$ 200MB and $\geqslant$ 400MB using an ordinal scale **(M3.4)**. When the granularity is modified in this way, the entire nested line diagram must be redrawn. Hybrid solutions to address both these limitations will presented in Section 2.7 below.

2.6 Toscana-Aggregation of Concept Hierarchies (M6.3)

The idea of a TOSCANA-aggregation is to view a related system of concept hierarchies metaphorically as a conceptual landscape of knowledge that can be explored by a purpose-oriented combination and inspection of suitable selections of the given concept hierarchies [32].

Groh [20] was the first to experiment with visual abstraction, using an abbreviated form of the line diagram for displaying flight routes in a network as reported in [15]. Further, the conceptual landscape idea [31] was explored extensively by Ducrou [9] and the results were a number of hybrid TOSCANA-systems. Wille's landscape metaphor can therefore be said to have significantly encouraged experimentation with visualization and line diagram abstraction.

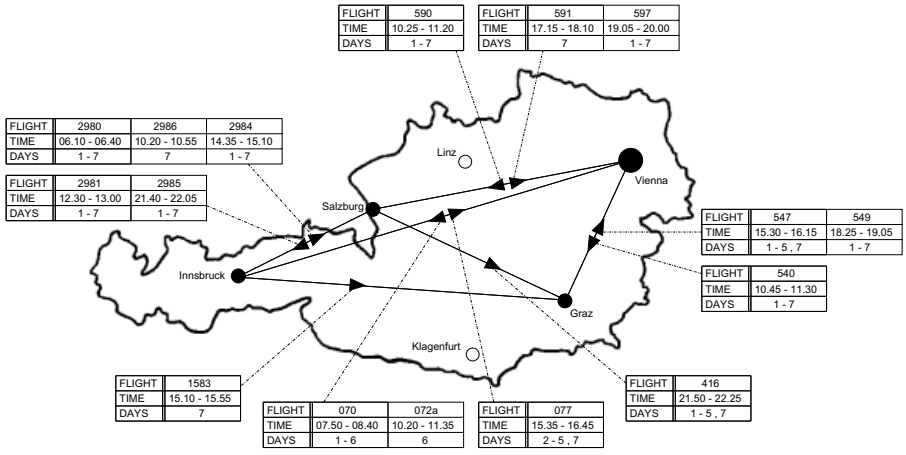


Fig. 4. A diagram showing a representation of a relational power context family modeling a network of airline flights in Austria. The network is depicted as a directed graph and the labels on line segments as relations. The digraph is placed within a backdrop map of Austria with the position of the vertices corresponding to the geo-coordinates of Austrian cities.

2.7 Retrieval with a Toscana-System (M10.2)

The term TOSCANA-system appears to refer to a particular software framework for Formal Concept Analysis, however it is intended by Stumme et al. [29] to mean a general class of Formal Concept Analysis systems that formalize a dataset into a context and to provide different points of view on the data by assisting the user in building and applying different conceptual scales. With this idea in mind we now look how various FCA-applications have dealt with visualization and retrieval. Conceptual knowledge retrieval is often a process where humans search for something with only a vague information need. Therefore humans learn step by step how to specify what they are searching for. Line diagrams of the activated scales are shown to the user who learns by inspecting them on how

to act further [32]. We show how the navigation facilities of line diagram can be enhanced through their combination with non-FCA visualization techniques.

Line diagrams as a support for navigation The powerful classification ability of FCA has found many applications in information retrieval. Some of them have been listed by [26]. Carpineto and Romano [3] argue that, in addition to their classification behaviors for information retrieval tasks, concept lattices can also support an integration of querying and browsing by allowing users to navigate into search results. Nowadays, several Formal Concept Analysis-based applications like CREDO [3], MAILSLEUTH [5, 7] or SEARCHSLEUTH [8] have been developed. Upstream research has studied the understandability of a lattice representation by novice users [13, 11]. A program called IMAGESLEUTH [10, 14] further proposes an interactive Formal Concept Analysis-based image retrieval system in which subjacent lattices are hidden. Users do not interact with an explicit representation of a lattice. They navigate from one concept to another by adding or removing terms suggested by the system. The same design approach is followed in SEARCHSLEUTH [8] and also in [18]. This ensures progressive navigation within the lattice (Wille's method **10.2**).

An hybrid approach to enhance readability During the usability review of MAILSLEUTH in 2003, [13] encountered users who felt that the drawing conventions for a lattice diagram were no different from a graph in mathematics. What makes this a lattice diagram and not a graph? In Hasse diagrams, line segments (represent the cover relation) are unlabelled. It is well understood in mathematics that a partially ordered set is transitive, reflexive and antisymmetric and so to simplify the drawing of an ordered set (via its cover relation) the reflexive and transitive edges are removed, and the directional arrows of the relation are dropped. It is understood by convention in Mathematics that the

Fig. 5. ImageSleuth: the interface presents only the extent of the current concept as thumbnails and generalizations/specializations by removal/addition of attributes to reach the upper and lower neighbors (shown to the top/bottom of the thumbnails). Pre-defined scales (perspectives) are displayed on the left.

Hasse diagram is hierarchical with the edges pointing upward. In other words, if $x < y$ in the partially ordered set then x appears at a clearly lower point than y in the diagram **(M1.5)**. To insinuate structure to novices, the idea of a line diagram with shaded layers was introduced in MailSleuth. The principle is simple, with dark at the top and light at the bottom; the shades progressively lighter as one moves from one level to the next (shown in Fig. 6). The top and bottom elements of the lattice have also been replaced with icons indicating All Mail and No Mail (when the bottom element is the empty set of objects).

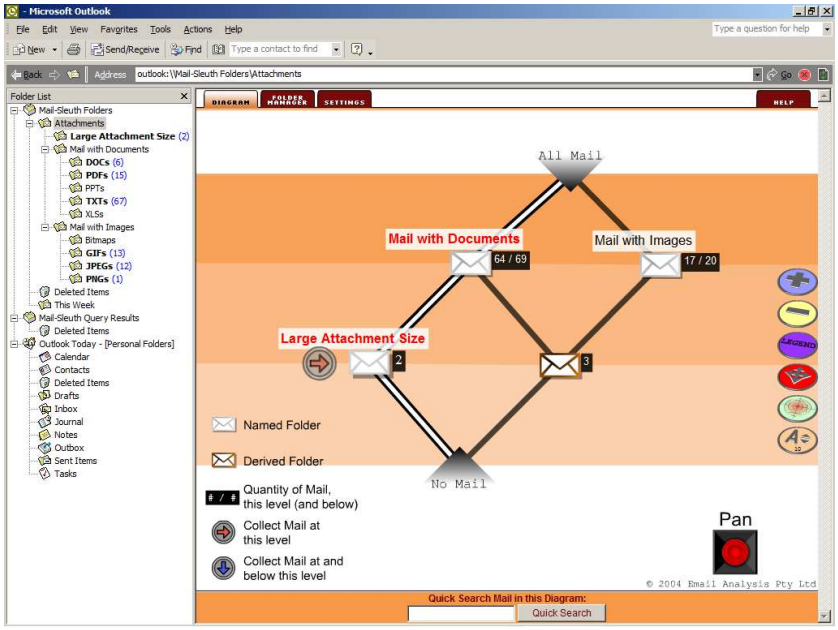


Fig. 6. A screenshot of the MailSleuth program.

Shading does not interfere with Formal Concept Analysis diagrammatic conventions because it is a backdrop to the line diagram. It can also be turned off if the line diagram is to be embedded in a printed document. However, the interaction of the layout algorithm and background layer shading fails (background layers are not aligned) in line diagrams with high dimensionality. On the other hand is possible to use the alignment of the background layers to guide the manual layout process. Once layer shading is used, users are better able to explain (and read) the line diagram from top-to-bottom and bottom-to-top.

Because MailSleuth deals with objects that are emails, it was natural to replace vertices with a literal iconic representation relevant to the domain. In the case where Derived Folders are unrealised, no vertex is drawn at all. Where data is present, an envelope is used, the envelopes animate by "appearing to lift" on rollover with drop shadowing. This helps suggest that vertices in the

line diagram can be moved and therefore manually adjusted by the user. Edge highlighting has been used to emphasise relationships in line diagrams in both TOSCANAJ and in MAILSLEUTH. This idea is used as a method to orient the current vertex in the overall line diagram so that relationships can be identified.

In many TOSCANA-systems the set of objects in the extent is often replaced with a number representing the cardinality of the extent (and/or the contingent), this can extended to allow the edges of the line diagram to be labelled with the ratio of object counts to approximate the idea of support in data mining.

Several strategies can be applied to reduce the number of vertices displayed to avoid visual overloading. Iceberg lattices [28] and alpha lattices [25] are smaller concept lattices retaining significant concepts that respect a threshold criterion based on the extent cardinality. Another lattice reduction strategy is decomposing the overall lattice into smaller sublattices, as done in ImageSleuth [10, 14], with this idea a partition of term space is achieved by a domain expert, attribute filtering or object zooming.

2.8 A hybrid approach to handle numerical attribute

Considering a many-valued context, we have seen that nested line diagrams encounter two major limitations. First, combining more than two many-valued attributes makes the nested line diagram unreadable. Secondly, while no information is lost when building a line diagram from a binary context, the binarization of numerical attributes leads to a loss of precision. An hybrid solution to address these limitations is to partition the context into binary and nominal attributes on the one hand, and ordinal and numerical attributes on the other hand. The binary/nominal line diagram is drawn but its circles are not filled with the ordinal/numerical line diagram. They are filled with a visualization that represents proximities between objects w.r.t. ordinal and numerical attributes. For instance, the circle labeled *Desktop* in Fig. 3(c) would contain 25 points that correspond to the 25 objects in the extent of this concept. The points are drawn in such a way that the distance between two points reflects the distance between the two corresponding objects w.r.t. the numerical attribute *HDD capacity*. These embedded visualizations can be drawn using a force-directed Multidimensional scaling technique [24], which takes as input the Euclidian distances between objects w.r.t. a given set of numerical attributes, and produces 2D embedding of the objects. This solution addresses both the above limitations since (i) the Euclidian distances can be computed w.r.t. any number of attributes; (ii) no scaling pre-processing is required. Note that the previous scaling pre-processing resulted in a predefined clustering of objects, whereas clusters of objects that can appear in embedded visualizations may not be expected, providing new insights into the data. However, this solution is not accurate when the number of objects and binary/nominal attributes grows. Hence, embedded visualization may become overcrowded and too small to be useful. A solution is to follow the *overview + detail* paradigm that splits the screen into two views (see Fig. 7). The overview shows the line diagram built from binary and nominal attributes, while the detailed view shows the embedded visualization of the selected vertex

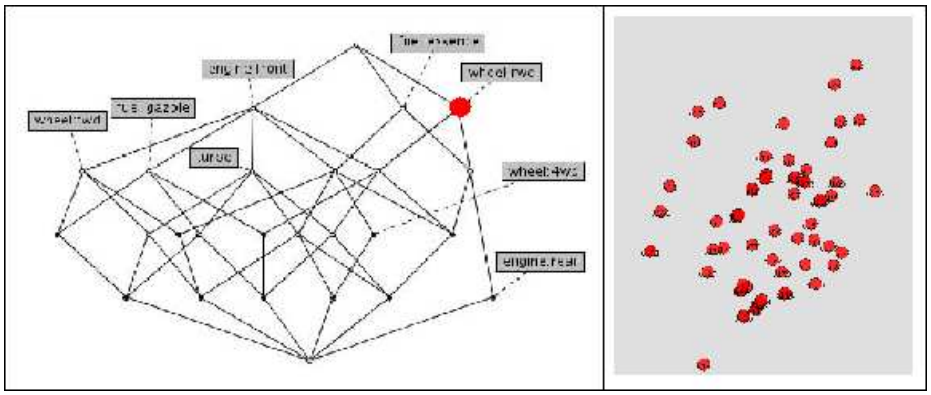


Fig. 7. Hybrid representation associated to a many-valued context. The line diagram left is built from the subcontext containing binary and nominal attributes. The objects contained in the extent of the selected concept are shown on the right by an MDS embedding that reflects their proximities w.r.t. numerical attributes.

in the overview. The detailed view acts as a kind of magnifying glass on the content (the extent) of the selected concept.

2.9 An hybrid approach to provide insights concerning navigation costs regarding a concept's neighbors

A final idea is to combine tag clouds with Formal Concept Analysis. This is shown in Fig. 8. This shows a tagging and annotation system called the Virtual Museum of the Pacific [17, 16]. Here, any attributes that lead to the upper neighbors are shown above the images, in this case `male, melanesia` and `PNG` and attributes (when added) that lead to lower neighbors in the concept lattice are shown below the images, in this example `fishing, container, spiritual` and `magic`. The size of the attributes is determined by the size the extent of the concepts that they lead to so the position and font size of upper and lower neighbors is conditioned by the extent sizes of the formal concepts they lead to. For example, in this case `fishing` is shown larger than `container, spiritual` and `magic` since it is a smaller specialization while `container` and `magic` are the same size and so have the same number of objects in their extents. The result is two tag clouds, one for the attributes leading to upper neighbors and another for lower neighbors.

3 Conclusion

This paper is a survey of visualization using line diagrams of concept lattices but with specific purpose. We argue that in the strict understanding of Conceptual Knowledge Processing variations of the standard techniques of line diagram drawing are supported. Further, that in various analytical contexts for

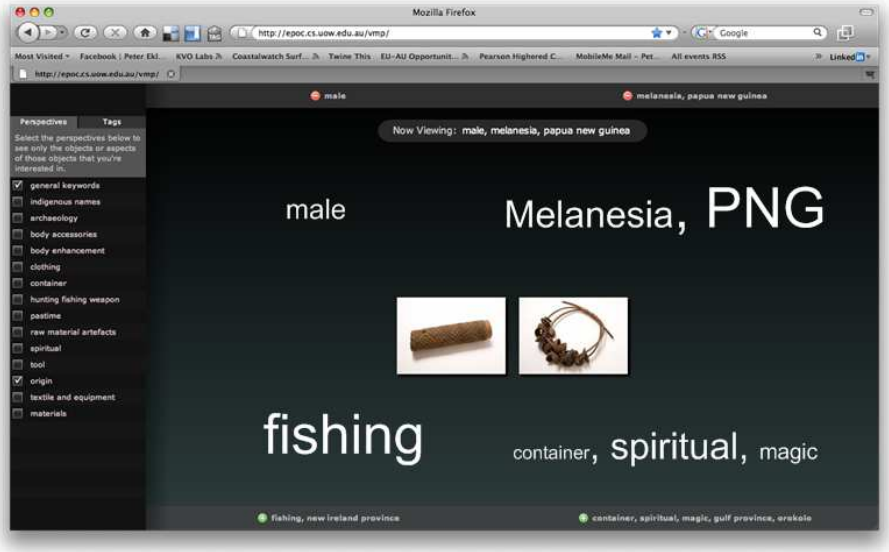


Fig. 8. A screenshot of the Virtual Museum of the Pacific program showing upper and lower neighbors of a formal concept as a tag cloud. The larger the lower neighbour, the larger the extent of the resulting formal concept being navigated to (minimal change). The larger the upper neighbor the greater the generalisation (maximal change).

Knowledge and Data Discovery, these variations of useful. Our examples cover the coloring of vertices to present the cardinality of formal concept extents; the placement of vertices of the line diagram to reflect association rule confidence; the use of icons for vertices, layer shading and abstractions of the line diagram that conform to Wille's landscape metaphors; and the use of iceberg lattices to minimize visual clutter and using similarity measures to condition the length of edges. Further, we have demonstrated the utility of showing only the conceptual neighborhood of the concept lattice and finally mixing this idea with the traditional idea of an attribute tag-cloud.

References

1. P. Becker and J. Hereth Correia. The ToscanaJ suite for implementing Conceptual Information Systems. LNAI3626, pages 250–271. Springer-Verlag, 2005.
2. P. Burmeister. Formal concept analysis with conimp: Introduction to the basic features'. Technical report, TU-Darmstadt, http://www.mathematik.tu-darmstadt.de/∼burmeister, 1996.
3. C. Carpineto and G. Romano. Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. *Journal of Universal Computer Science*, 10(8):985–1013, 2004.
4. R. Cole. Automatic layout of concept lattices using force directed placement and genetic algorithms. In *Proc. of the 23th Australiasian Computer Science Conference*, pages 47–53. IEEE Computer Society, 2000.

5. R. Cole, P. Eklund, and G. Stumme. CEM — A Program for Visualization and Discovery in Email. In D.A. Zighed, J. Komorowski, and J. Zytkow, editors, *Proc. of the 4th European Conf on Principles and Practice of Knowledge Discovery in Databases (PKDD'00)*, LNAI 1910, pages 367–374. Springer-Verlag, 2000.
6. R.J. Cole, J. Ducrou, and P. Eklund. Automated layout of small lattices using layer diagrams. In *Proc. of the 4th Int. Conference on Formal Concept Analysis*, volume 3874/2006 of *LNAI*, pages 291–305. Springer, 2006.
7. R.J. Cole, P. Eklund, and G. Stumme. CEM - a program for visualization and discovery in email. In *Principles of Data Mining and Knowledge Discovery: 4th European Conference, PKDD 2000, Lyon, France, September 2000. Proceedings*, pages 367 – 374, 2000.
8. F. Dau, J. Ducrou, and P. Eklund. Concept Similarity and Related Categories in SearchSleuth. In *Proceedings of the 16th International Conference on Conceptual Structures (ICCS'08)*, LNCS 5113, pages 255–268. Springer-Verlag, 2008.
9. J. Ducrou. *Design for Conceptual Knowledge Processing: Case Studies in Applied Formal Concept Analysis*. PhD thesis, School of Information Technology and Computer Science, The University of Wollongong, 2007.
10. J. Ducrou and P. Eklund. An intelligent user interface for browsing and search MPEG-7 images using concept lattices. *Int. Journal of Foundations of Computer Science*, 19(2):359–381, 2008.
11. J. Ducrou and P. Eklund. Faceted document navigation using conceptual structures. In P. Hitzler and H. Schärf, editors, *Conceptual Structures in Practice*, pages 251–278. CRC Press, 2009.
12. P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.
13. P. Eklund, J. Ducrou, and P. Brawn. Concept lattices for information visualization: Can novices read line diagrams. In Peter Eklund, editor, *Proceedings of the 2nd International Conference on Formal Concept Analysis - ICFCA'04*. Springer-Verlag, February 2004.
14. P. Eklund, J. Ducrou, and T. Wilson. An intelligent user interface for browsing and search MPEG-7 images using concept lattices. In *Proceedings of the 4th international conference on concept lattices and their applications*, LNCS 4923, pages 1–22. Springer-Verlag, 2008.
15. P. Eklund, B. Groh, G. Stumme, and R. Wille. A contextual-logic extension of toscana. In *Conceptual structures: logical, linguistic and computational issues*, LNAI 1867, pages 453–467. Springer, 2000.
16. P. Eklund, T. Wray, and J. Ducrou. Web services and digital ecosystem support using formal concept analysis. In N. Spyratos, editor, *The International ACM Conference on Management of Emergent Digital EcoSystems (MEDES09)*, page in press. ACM Press, 2009.
17. P. Eklund, T. Wray, P. Goodall, B. Bunt, A. Lawson, L. Christidis, V. Daniels, and M. Van Olffen. Designing the Digital Ecosystem of the Virtual Museum of the Pacific. In *3rd IEEE International Conference on Digital Ecosystems and Technologies*, pages 805–811. IEEE Press, 2009.
18. S. Ferre. Camelis: a logical information system to organize and browse a collection of documents. *Int. J. General Systems*, 38(4), 2009.
19. R. Freese. Automated lattice drawing. In *Concept Lattices*, LNCS 2961, pages 112–127. Springer-Verlag, 2004.
20. B. Groh. *A Contextual-Logic Framework Based on Relational Power Context Families*. PhD thesis, School of Information Technology, Griffith University, 2002.

21. T. Hannan and A. Pogel. Spring-based lattice drawing highlighting conceptual similarity. In *Proceedings of the International Conference on Formal Concept Analysis, ICFCA 2006*, LNCS 3974, pages 264–279, Berlin, 2006. Springer-Verlag.
22. T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.
23. K. Lengnink. *"Ahnlichkeit als Distanz in Begriffsverbänden.* In R Wille G Stumme, editor, *Begriffliche Wissensverarbeitung: Methoden und Anwendungen*, pages 57–71. Springer-Verlag, 2001.
24. A. Morrison and M. Chalmers. Improving Hybrid MDS with Pivot-Based Searching. In *Proceedings of the 2003 IEEE Symposium on Information Visualization (Info Vis' 03)*, pages 85–90, 2003.
25. N. Pernelle, V. Ventos, and H. Soldano. ZooM: Alpha Galois Lattices for Conceptual Clustering. In *Proc. Ofthe Managing Specialization/Generalization Hierarchies (MASPEGHI) Workshop*, 2003.
26. U. Priss. Formal concept analysis in information science. *Annual Review of Information Science and Technology*, 40:521–543, 2006.
27. J. Saquer and J. S. Deogun. Concept aproximations based on rough sets and similarity measures. In *Int. J. Appl. Math. Comput. Sci.*, volume 11, pages 655 – 674, 2001.
28. G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing iceberg concept lattices with Titanic. *Data & Knowledge Engineering*, 42(2):189–222, 2002.
29. Gerd Stumme, Rudolf Wille, and Uta Wille. Conceptual knowledge discovery in databases using formal concept analysis methods. In *Principles of Data Mining and Knowledge Discovery*, pages 450–458, 1998.
30. R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In Ivan Rival, editor, *Ordered Sets*, pages 445–470. Reidel, 1982.
31. R. Wille. Conceptual landscapes of knowledge: A pragmatic paradigm for knowledge processing. In W. Gaul and H. Locarek-Junge, editors, *Classification in the Information Age*, pages 344–356. Springer-Verlag, 1999.
32. R. Wille. Methods of conceptual knowledge processing. In *Prof. of the 5th Int. Conf. Formal Concept Analysis (ICFCA 2006)*, LNCS4390, pages 1–29. Springer-Verlag, 2006.
33. R. Wille and B. Ganter. *Formal Concept Analysis: Mathematical Foundations.* Springer-Verlag, Berlin, 1999.