Chapter 14

CONCEPTUAL ON-LINE ANALYTICAL PROCESSING

Gerd Stumme
Technische Universität Darmstadt, Fachbereich Mathematik
Schloßgartenstr. 7, D–64289 Darmstadt
stumme@mathematik.tu-darmstadt.de

Abstract A *Conceptual Information System* consists of a database together with conceptual hierarchies. The visualization of arbitrary combinations of conceptual hierarchies by *nested line diagrams* allows an on-line interaction with a database to analyze data conceptually. The chapter describes the conception of Conceptual Information Systems and discusses the use of their visualization techniques for On-Line Analytical Processing (OLAP).

1. INTRODUCTION

The conception of Conceptual Information Systems relies on the insight that concepts are basic units of human thinking, and is founded on the mathematical theory of Formal Concept Analysis. *Formal Concept Analysis* [16, 2] reflects an elementary understanding of the concept of 'concept' which arose in philosophy since antiquity [15] and has been established in the German standards DIN 2330 and DIN 2331. Since its foundation in the early 80ies, Formal Concept Analysis has been used in many applications in data analysis, information retrieval, and knowledge discovery.

A *Conceptual Information System* consists of a (relational) database together with conceptual hierarchies. These hierarchies, called *conceptual scales*, are used to support navigation through the data. An important factor for the success of Conceptual Information Systems is the visualization of conceptual scales by *line diagrams*. By combining conceptual scales in *nested line diagrams*, a large variety of perspectives can be generated interactively, in which relationships and dependencies can be investigated. The management system TOSCANA allows an on-line interaction with a database to analyze and explore data conceptually.

Actually, there are over 30 implemented Conceptual Information Systems, including a flight movement information system at Frankfurt Airport [4], a library retrieval system [8], and a decision support environment at a warehouse company [3].

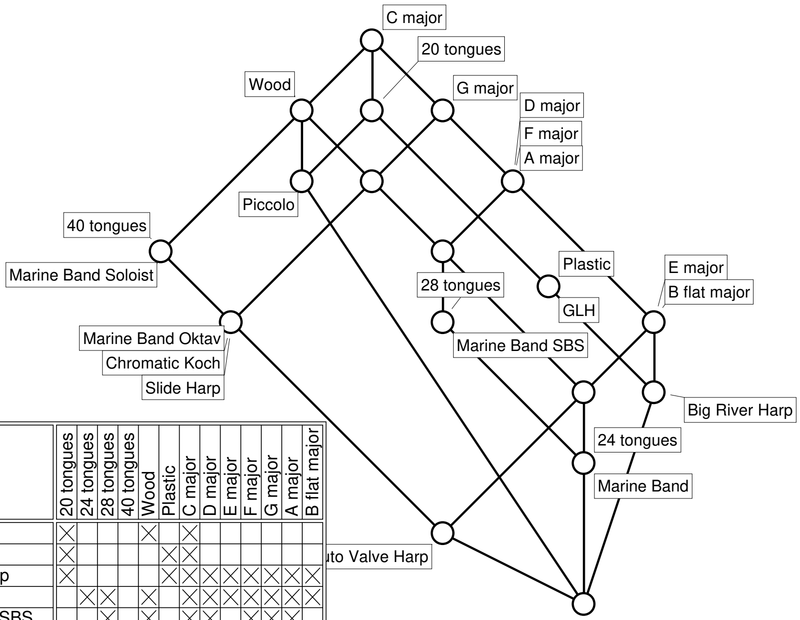
On-Line Analytical Processing (OLAP) relies on the metaphor of a (high-dimensional) cube containing the data. For dimensions which are not structured hierarchically, the cube metaphor provides a good intuitive understanding of multi-dimensional data. But an essential feature of OLAP dimensions is that they are ordered hierarchally: days roll up into months, months into quarters and years, products into product groups and product lines. Often they are trees (*simple hierarchies*), but they may be any arbitrary partially ordered set (*multiple hierarchy*). In this setting, the cube metaphor is not the most natural approach, because it imitates the mathematical construction of a direct product of linear vector spaces which forces the hierarchies to be represented in a flat linear form. Instead of listing the hierarchies on (one-dimensional) axes, we suggest to visualize them by *line diagrams* which were created in order to represent hierarchical structures. By using *nested line diagrams*, arbitrary dimensions can be combined for ad hoc analysis.

In the next section, we introduce the basics of Formal Concept Analysis and Conceptual Information Systems. In Sections 3 and 4, Conceptual On-Line Analytical Processing is discussed. Section 5 reports from an implementation in a warehouse company and concludes with a discussion on future research.

2. CONCEPTUAL INFORMATION SYSTEMS

Conceptual Information Systems are based on the mathematical theory of Formal Concept Analysis. The aim of Formal Concept Analysis [16, 2] is a mathematical formalization of the concept 'concept'. It reflects the philosophical understanding of concepts as units of thought consisting of two parts: the extension containing all objects which belong to the concept and the intension containing the attributes shared by all those objects. This is modeled by *formal concepts* that are derived from a *formal context*.

Definition. A *(formal) context* is a triple $\mathbb{K} := (G, M, I)$ where G and M are sets and I is a relation between G and M. The elements of G and M are called *objects* and *attributes*, respectively, and $(g, m) \in I$ is read *"the object g has the attribute m"*. A *(formal) concept* is a pair (A, B) such that $A \subseteq G$ and $B \subseteq M$ are maximal with $A \times B \subseteq I$. The set A is called the *extent* and the set B the *intent* of the concept. The hierarchical subconcept–superconcept–relation of concepts is formalized by $(A, B) \leq (C, D) :\iff A \subseteq C \ (\iff B \supseteq D)$. The set of all concepts of the context $\mathbb{K}$ together with this order relation is a complete lattice that is called the *concept lattice* of $\mathbb{K}$ and is denoted by $\underline{\mathfrak{B}}(\mathbb{K})$.



	20 tongues	24 tongues	28 tongues	40 tongues	Wood	Plastic	C major	D major	E major	F major	G major	A major	B flat major
Piccolo	×				×		×						
GLH	×					×	×						
Big River Harp	×					×	×	×	×	×	×	×	×
Marine Band		×	×		×		×	×	×	×	×	×	×
Marine Band SBS			×		×		×	×		×	×	×	
Marine Band Soloist				×	×		×						
Marine Band Oktav				×	×		×				×		
Auto Valve Harp				×	×		×	×	×	×	×	×	×
Chromatic Koch				×	×		×				×		
Slide Harp				×	×		×				×		

Figure 14.1 Formal context of harps and its concept lattice

Example. In Figure 14.1, a formal context of the Richter Harps produced by HOHNER INC. is given. A tuple $(g, m) \in M$ is read as 'harp g is available with feature m'. In the line diagram, the circles represent the concepts. A concept is a subconcept of another, if there is an ascending path of straight line segments from the former to the latter. The extent [intent] of each concept contains all objects [attributes] which can be reached from the concept on a descending [ascending] path.

If we are for example interested in a wooden harp tuned in D major, then we take the largest concept that has Wood and D major in its intent. This concept is represented by the circle just above the label 28 tongues. The extent of this concept contains Marine Band SBS, Marine Band, and Auto Valve Harp — so these are exactly the harps available in wood and D major. The intent of this concept contains — beside Wood and D major — the features A, F, G, and C major, so the three harps are available in these tunings also. Such an *implication* corresponds to a functional dependency in database theory.

In many applications, attributes are not one-valued as in the previous example, but allow a range of values. This is modeled by *many-valued contexts*. In order to obtain a concept lattice, many-valued contexts are 'translated' into one-valued contexts by *conceptual scales*.

Definition. A *many-valued context* is a tuple $(G, M, (W_m)_{m \in M}, I)$ where G and M are sets of *objects* and *attributes*, resp., W_m is a set of *values* for each $m \in M$, and $I \subseteq G \times \bigcup_{m \in M}(\{m\} \times W_m)$ such that $(g, m, w_1) \in I$ and $(g, m, w_2) \in I$ imply $w_1 = w_2$. A *conceptual scale* for an attribute $m \in M$ is a context $\mathbb{S}_m := (G_m, M_m, I_m)$ with $W_m \subseteq G_m$. The context (G, M_m, J) with $gJn :\iff \exists w \in W_m \colon (g, m, w) \in I \wedge (w, n) \in I_m$ is called the *realized scale* for the attribute m.

Conceptual Information Systems consist of a many-valued context together with a collection of conceptual scales. The many-valued context is implemented as a relational database. The collection of the scales is called *conceptual scheme*. It is written in the description language CONSCRIPT [13]. Beside the contexts of the conceptual scales, the conceptual scheme also contains the layout of their line diagrams. The layout has to be provided in advance, since experience showed that well readable line diagrams cannot be generated fully automatically in general. For Conceptual Information Systems, the management system TOSCANA [5, 14] has been developed. Based on the paradigm of conceptual landscapes of knowledge [19], TOSCANA supports the navigation through the data by using conceptual scales like maps which are designed for different purposes and in different granularities.

Example. Figure 14.2 shows a realized scale of a Conceptual Information System on pipelines [12]. The many-valued context consists of 3961 pipes, fittings, etc., and of 54 many-valued attributes. It shall support the engineer by choosing suitable parts for a projected pipeline system. Since there are almost 4000 objects, the scale does not display their names, but the contingents only. One can for instance see, that $52 + \ldots + 27 = 348$ of the 3961 different parts are flanges (German: Flansche) which are differentiated further according to the German Industrial Standards (DIN). By *zooming into* this concept, one can see the distribution of the 348 flanges according to another conceptual scale, e. g, the inner diameter or the wall thickness.

For the exploration of relationships between different attributes, it is desirable to visualize more than one conceptual scale at a time. *Nested line diagrams* are used to show the direct product of the scales. We introduce them in the next section where we also discuss their role for On-Line Analytical Processing.

3. CONCEPTUAL ON-LINE ANALYTICAL PROCESSING

On-Line Analytical Processing (OLAP) has become almost synonymous with multi-dimensional data. OLAP addresses many topics, like data preprocessing and efficient data storage for supporting the analysis process (refer e. g. to [7]). Here, we focus on the visualization of the data.

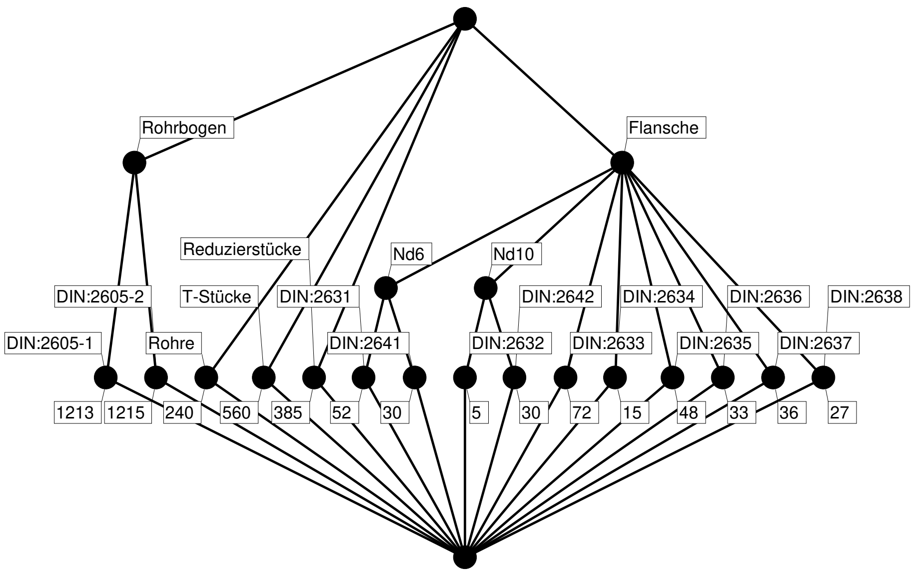


Figure 14.2 Realized scale 'Part Type'

Definition. A *dimension* is a set D, its elements are called its *members*. Let $\mathcal{D} := \{D_1, D_2, \ldots, D_n\}$ be a set of dimensions. Each tuple of $\mathsf{X}\mathcal{D} := D_1 \times D_2 \times \ldots \times D_n$ is called a *member combination*. It addresses a single data point called a *cell*. A *variable* is a partial function $\nu \colon \mathsf{X}\mathcal{D} \to V$ where V is a set. $\nu(d_1, \ldots, d_n)$ is the *value of the cell addressed by the member combination* $(d_1, \ldots, d_n)$. The set $\mathcal{D}$ together with one or more variables is called the *data cube*.

Example. Our example is about sales data of a (fictitious) soft-drink wholesale company. Suppose that we want to examine the sales of beverage in dependence of time, region and type of product. Thus we have three dimensions: REGION, PRODUCT, and TIME. Let's say that they consist of the members $D_{\text{REGION}} :=$ {TOTAL, EUROPE, AMERICA, NORTH AMERICA, SOUTH AMERICA, ASIA}, $D_{\text{PRODUCT}} :=$ {TOTAL, MINERAL WATER, JUICE, ORANGE JUICE, APPLE JUICE, COLA}, $D_{\text{TIME}} :=$ {1996, 1ST QUARTER 1996, 2ND QUARTER 1996, 3RD QUARTER 1996, 4TH QUARTER 1996, 1997, 1ST QUARTER 1997, 2ND QUARTER 1997, 3RD QUARTER 1997, 4TH QUARTER 1997}. In a real application, there will of course be more dimensions, and a much finer granularity, for instance down to city or shop level for REGION, or to day (or even hour) level for TIME. The sales (in million gallons) are represented by a function SALES: $D_{\text{REGION}} \times D_{\text{PRODUCT}} \times D_{\text{TIME}} \to \mathbb{R}^+$. We can imagine the sales as stored in a three-dimensional cube, where the edges are labeled with the members of REGION, PRODUCT, and TIME, resp. Most OLAP tools

		Total	Europe	America	North	South	Asia
MineralWater	1997	837	442	268	174	94	127
	1Q7	191	99	63	41	22	29
	2Q7	201	102	66	43	23	33
	3Q7	274	141	82	51	31	51
	4Q7	171	100	57	39	18	14
Cola	1997	1523	432	673	375	298	418
	1Q7	364	99	160	89	71	105
	2Q7	378	103	171	91	80	104
	3Q7	405	120	189	103	86	96
	4Q7	376	110	153	92	61	113
Juice	1997	816	360	257	170	87	199
	1Q7	189	81	62	41	21	46
	2Q7	200	85	63	42	21	52
	3Q7	223	99	68	44	24	56
	4Q7	204	95	64	43	21	45
Total	1997	3176	1234	1198	719	479	744
	1Q7	744	279	285	171	114	180
	2Q7	779	290	300	176	124	189
	3Q7	902	360	339	198	141	203
	4Q7	751	305	274	174	100	172

Figure 14.3 Visualization of the data cube in a spreadsheet (nested diagram)

display the data in a spreadsheet as in Figure 14.3. For instance, we see that $\text{SALES}(\text{COLA}, \text{NORTH AMERICA}, \text{1ST QUARTER } 1997) = 89$.

Definition. A *hierarchy* on a dimension D is a partially ordered set $H := (D, \leq)$. It is called *simple hierarchy*, if it is a tree. Otherwise it is called *multiple hierarchy (within the dimension D)*.

Typically, aggregation follows the hierarchy from bottom to top. The type of aggregation depends on the type of variable. For most variables (e. g., BUDGET or SALES) the values will be summed up. But other ways of aggregation are in use as well. For instance, for share prices or inventory numbers, usually the average is computed.

Example. The hierarchies of the three dimensions PRODUCT, REGION, and TIME are shown in Figure 14.4. They are all simple hierarchies (trees). The sales are aggregated by summation in all dimensions. ORANGE JUICE and APPLE JUICE roll up to JUICE, and JUICE, MINERAL WATER and COLA roll up to TOTAL.

In OLAP terminology, diagrams as in Figure 14.3 are called *nested diagrams*. In this section, we examine how *nested line diagrams* of Conceptual Information Systems can be used as an alternative method of data visualization.

Figure 14.5 shows how the data cube is composed as direct product of the dimensions, where the members of each dimension are ordered in a linear way. Many tools indicate the hierarchies on the dimensions additionally like a PC file manager displays the folder/subfolder hierarchy. But the basically linear arrangement is essential for the cube metaphor. Since the hierarchies model

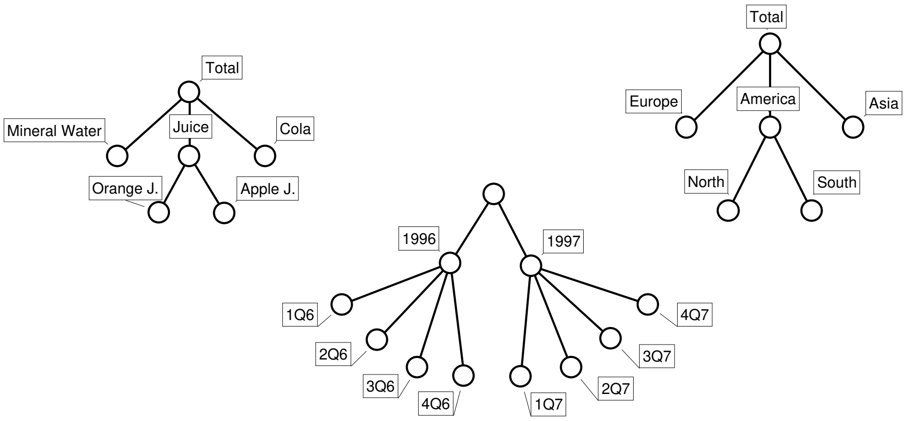


Figure 14.4 The hierarchies

the basic understanding of the conceptual view of the analyst on the data, they should play a prominent role in the visualization. Indeed, they are often used for displaying one single hierarchy, as in Figure 14.4. But if two or more hierarchies occur simultaneously, then this visualization technique is dropped.

In Conceptual Information Systems, nested line diagrams are used for displaying line diagrams of large partially ordered sets (especially conceptual scales). Hierarchical dimensions roughly correspond to conceptual scales, so OLAP analysis tools can roughly be seen as special Conceptual Information Systems. Nested line diagrams can be used for drawing direct products of the dimensions. In contrast to nested diagrams, they do not only provide all member combinations, but also reflect the derived order:

Definition. Let $H_i := (D_i, \leq_i)$, $i = 1, \ldots n$, be hierarchies. Then the *derived order* on the direct product $H := (D, \leq)$ with $D := D_1 \times D_2 \times \ldots \times D_n$ is defined by $(d_1, \ldots, d_n) \leq (e_1, \ldots, e_n) :\Longleftrightarrow \forall i \in \{1, \ldots, n\}\colon d_i \leq_i e_i$.

Example. The nested line diagram of the direct product of the three dimensions REGION, PRODUCT, and TIME (see Figure 14.5) is displayed in Figure 14.6. The derived order can be read by following ascending paths. Hereby the lines of the outer two levels have to be replaced by sheaves of 4 and $5 \times 4 = 20$ parallel lines, resp., linking corresponding elements. For instance, (SOUTH AMERICA, 3RD QUARTER, MINERAL WATER) $\leq$ (AMERICA, TOTAL, MINERAL WATER), since the cell addressed by the latter member combination can be reached by an ascending path from the former one. For finding out how much Cola was sold in North America in the first quarter of 1997, we have a look in the lower left ellipse (labeled with NORTH) in Figure 14.6. In the leftmost ellipse (1. Q.), we find the entry 89 in the right box (COLA).

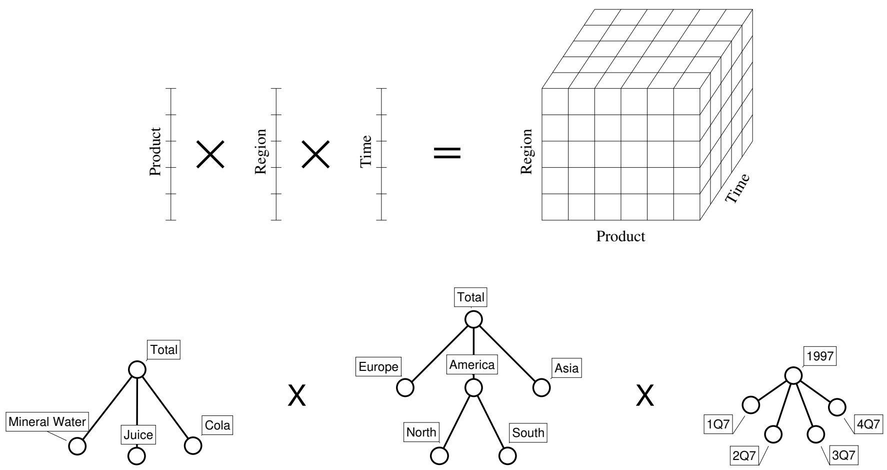


Figure 14.5 The direct product of (linear) dimensions vs. the direct product of partially ordered sets. The result of the latter is shown in Figure 14.6.

Clearly this representation needs more space than the one in Figure 14.3. Its advantage is the clear structuring along the most important — from the analyst's actual point of view — dimension (which is chosen as outermost hierarchy). Figure 14.6 shows that displaying a partially ordered set with 120 elements is close to the boundaries of the system. The spreadsheet can display even larger data volumes, and still look neat at a first glance. But, as in typography, the most important aim is not to provide a neat representation, but one that supports easy reading [11]. TOSCANA is designed for a more general approach, allowing more complicate scales than the dimensions typically used in OLAP applications, which often are just trees. In particular, TOSCANA supports arbitrary partially ordered sets, although (concept) lattices are the standard. In standard TOSCANA applications, not all elements of the hierarchy are labeled, but there may be more than one label for each element. Considering the special features of OLAP data, some improvements on the lay-out are conceivable. They will be discussed in Section 5.

4. SLICE & DICE

Slicing, pivoting, drill-down, and drill-up are the interactive ways of accessing the data stored in a data cube. This section describes them, and presents the corresponding activities for nested line diagrams. These activities can all be performed by interacting with the diagrams, no manipulation language is needed.

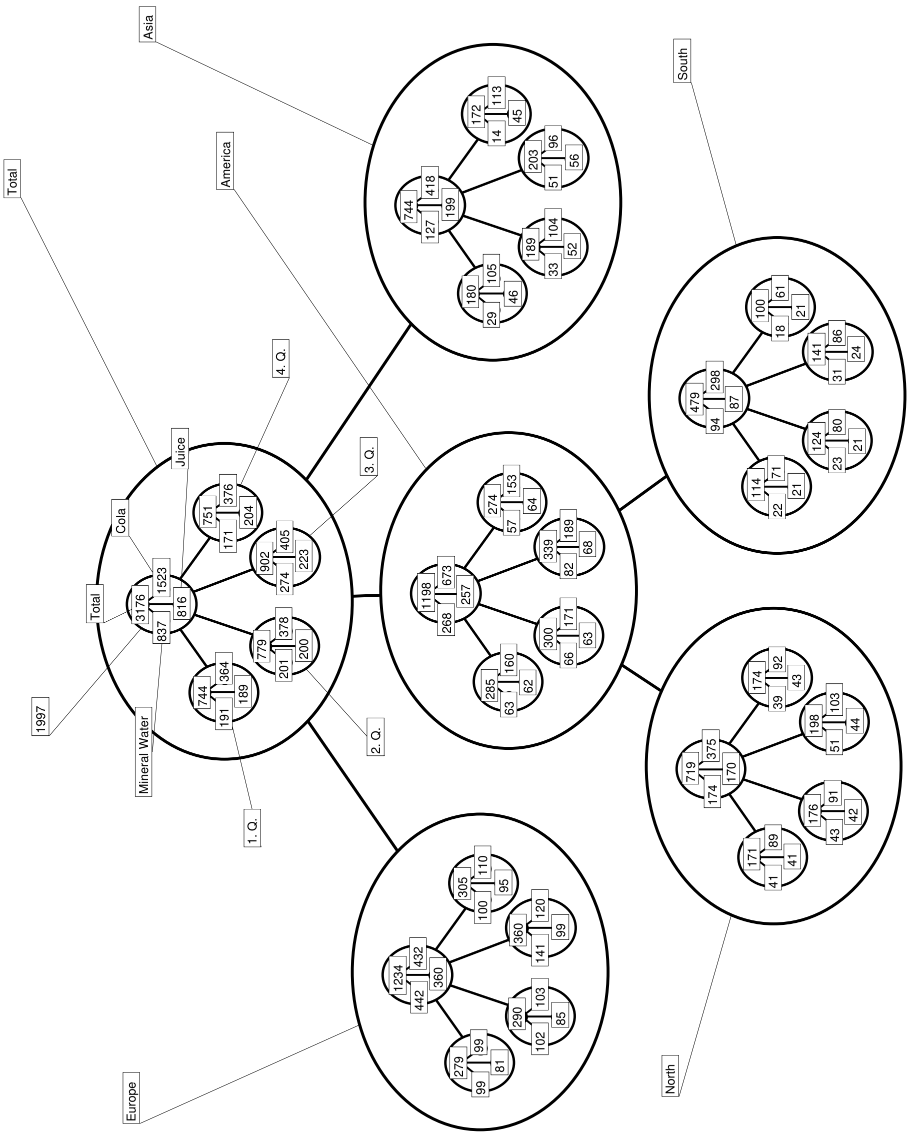


Figure 14.6 Visualization as nested line diagram

Slicing. A *slice* of a data cube is a subset of the data cube, where one or more dimensions are restricted to one single member. For instance, if the analyst is interested in the development of the sales of the different product lines over time, then only the data given in the slice determined by the condition REGION=“TOTAL” is of interest. In Figure 14.3, this turns out to show

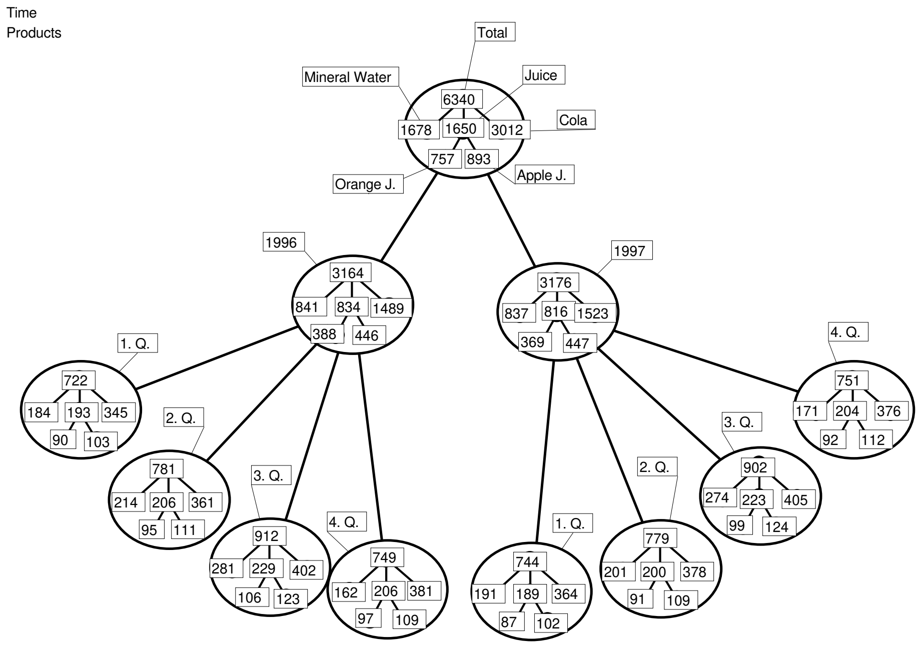


Figure 14.7 Zooming with drill-down

only the left most column (which of course will then be displayed by a two-dimensional spreadsheet). In nested line diagrams, slicing the cube corresponds to *zooming* in one of the ellipses of the outer level. For instance, the condition REGION= "TOTAL" is obtained by zooming into the top element of the outer hierarchy in Figure 14.6. The result is a two level nested line diagram on full screen size. This feature is fully supported by TOSCANA. If the user chooses only some of the hierarchies to be displayed, then this corresponds to zooming in the top elements of all other hierarchies, hence the most general aggregation is applied in the corresponding dimensions.

Drill-Down. Restricting the number of dimensions means that one can look at them in more detail. Instead of only examining the topmost levels of their hierarchies, one wants to see a finer granularity. This unfolding of the hierarchies is called *drill-down*. Figure 14.7 shows the result of zooming into the top element of the outer hierarchy in Figure 14.6 (i. e., letting REGION= "TOTAL"), and drilling down the PRODUCT hierarchy. Additionally, the TIME dimension has been extended to two years.

At the moment, TOSCANA does not support this way of drill-down very well. The hierarchies have to be prepared in advance and cannot be changed on the fly, forced by the unsolved problem of a satisfying automatic lay-out

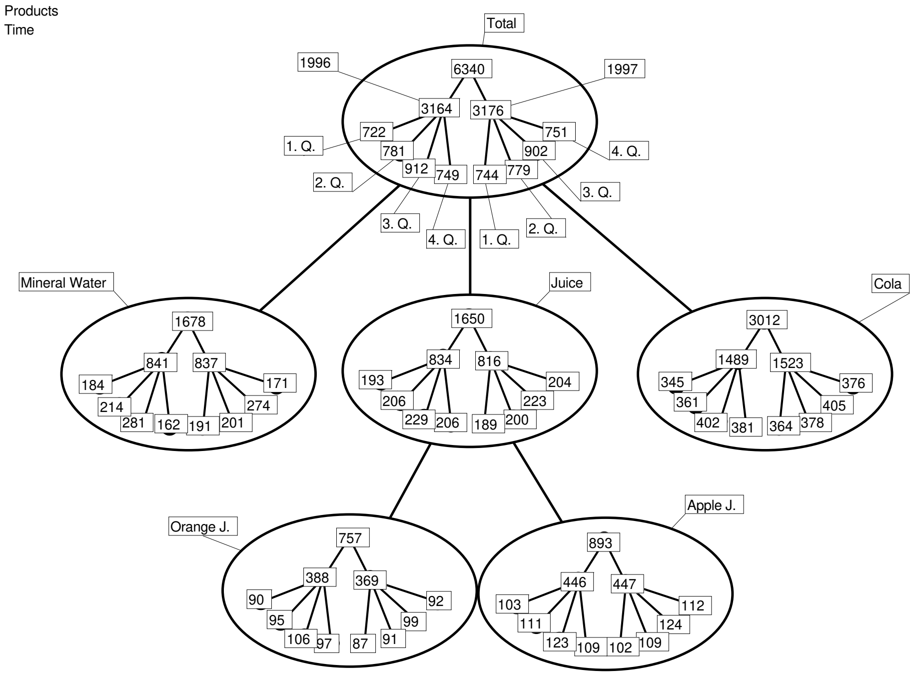


Figure 14.8 Pivoting the hierarchies of Figure 14.7

algorithm for partially ordered sets. In the next section, we discuss how this problem can be encountered. The actual solution is to provide scales with different levels of granularity between which the analyst can choose.

Another way of drill-down is to refer to external information sources, e. g., to databases of the transaction systems or to Internet sites. In TOSCANA, such references can be attached to each data cell. By mouse-click, a report generated by the database or a Web browser will be opened.

Pivoting. Different questions request different views on the data. In a spreadsheet display, this means that the dimensions listed on the vertical and horizontal axis are interchanged. This operation is called *pivoting* or *rotating*. For nested line diagrams, it corresponds to permuting the inner and the outer hierarchies. The result of pivoting the nested line diagram of Figure 14.7 is shown in Figure 14.8. The diagram in Figure 14.7 can be used to examine the question *"How does the composition of sold products change over time?"*, while the pivoted version in Figure 14.8 is more adequate for investigating *"How do the sales evolve in time for each product?"*. Pivoting of hierarchies is a standard feature in TOSCANA.

5. CONCLUSION AND FURTHER DEVELOPMENTS

During the editing process of this volume, a decision support environment based on TOSCANA has been implemented in a Swiss warehouse company, which also makes use of the ideas in this chapter. Being designed for supporting marketing campaigns, the data cells contain groups of customers who bought the same items or share similar demographic attributes. This allows e. g. the targeting of interesting customer groups for mailing campaigns. More details about the system can be found in [3].

There remain open research issues. Drill-down requires techniques for extending and pruning hierarchies on the fly. For arbitrary hierarchies (and even for lattices), the development of fully automatic algorithms providing satisfying diagrams is an open challenge. But most OLAP hierarchies are trees, which can be drawn automatically. Beside supporting drill-down, an automatic layout routine can solve the problem of efficiently exploiting the whole screen space.

Data cubes are usually only sparsely populated. Often less than 10% of the cells contain data. For the visualization this implies that not the whole direct product of the hierarchies needs to be displayed. In this case *local scaling* [9] can be used for automatic pruning. Local Scaling and Conceptual On-Line Analytical Processing are features addressed in the current development of TOSCANA 3.0.

Formal Concept Analysis has been applied independently in different domains related to data analysis, knowledge discovery and decision support. For instance, Formal Concept Analysis has recently been applied for mining association rules in large databases [6]. However, for supporting managers in their tasks, data analysis, knowledge discovery, and decision support tools should be integrated in a homogeneous environment. Our current research is focussed on the theoretical foundations of this integration, based on the unified knowledge representation provided by Formal Concept Analysis.

References

[1] E.F. Codd, S.B. Codd, C.T. Salley: *Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate.* www.hyperion.com/whitepapers.cfm

[2] B. Ganter, R. Wille: *Formal Concept Analysis: Mathematical Foundations*. Springer, Heidelberg 1999

[3] J. Hereth, G. Stumme, U. Wille, R. Wille: Conceptual Knowledge Discovery and Data Analysis. *Proc. ICCS 2000* (submitted)

[4] U. Kaufmann: *Begriffliche Analyse von Daten über Flugereignisse – Implementierung eines Erkundungs- und Analysesystems mit TOSCANA*.

Diplomarbeit, TU Darmstadt, 1996

[5] W. Kollewe, M. Skorsky, F. Vogt, R. Wille: TOSCANA – ein Werkzeug zur begrifflichen Analyse und Erkundung von Daten. In: R. Wille, M. Zickwolff (eds.): *Begriffliche Wissensverarbeitung – Grundfragen und Aufgaben*. B. I.–Wissenschaftsverlag, Mannheim 1994

[6] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal: Efficient mining of association rules using closed itemset lattices. *Information Systems* **24**(1), 1999, 25–46

[7] Pilot Software: *An introduction to OLAP: Multidimensional terminology and technology*. White Paper, Pilot Software, 1999, www.pilotsw.com/olap/olap.htm

[8] T. Rock, R. Wille: Ein TOSCANA-System zur Literatursuche. In: G. Stumme and R. Wille (eds.): *Begriffliche Wissensverarbeitung: Methoden und Anwendungen*. Springer, Berlin-Heidelberg 2000, 239–253

[9] G. Stumme: Local scaling in conceptual data systems. LNAI **1115**, Springer, Heidelberg 1996, 308–320

[10] G. Stumme, R. Wille, U. Wille: Conceptual Knowledge Discovery in Databases Using Formal Concept Analysis Methods. In: J. M. Żytkow, M. Quafofou (eds.): *Principles of Data Mining and Knowledge Discovery*. LNAI **1510**, Springer, Heidelberg 1998, 450–458

[11] J. Tschichold: *Erfreuliche Drucksachen durch gute Typographie: eine Fibel für jedermann*. Maro-Verlag, Augsburg, 2nd edition 1992

[12] N. Vogel: *Ein Begriffliches Erkundungssystem für Rohrleitungen*. Diplomarbeit, TH Darmstadt 1995

[13] F. Vogt: *Datenstrukturen und Algorithmen zur Formalen Begriffsanalyse: Eine C++–Klassenbibliothek*. Springer, Heidelberg 1996

[14] F. Vogt, R. Wille: TOSCANA — A graphical tool for analyzing and exploring data. LNCS **894**, Springer, Heidelberg 1995, 226–233

[15] H. Wagner: Begriff. In: H. M. Baumgartner, C. Wild (eds.): *Handbuch philosophischer Grundbegriffe.* Kösel Verlag, München 1973, 191–209

[16] R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets*. Reidel, Dordrecht–Boston 1982, 445–470

[17] R. Wille: Line diagrams of hierarchical concept systems. *Int. Classif.* **11** (1984), 77–86

[18] R. Wille: Lattices in data analysis: how to draw them with a computer In: I. Rival (ed.): *Algorithms and order*. Kluwer, Dordrecht 1989, 33–58

[19] R. Wille: Conceptual landscapes of knowledge: A pragmatic paradigm of knowledge processing. In: *Proc. KRUSE '98*, Vancouver, Canada, August 11–13, 1997, 2–13