Relational Analytical Tools: VisTool and Formal Concept Analysis

Cliff Joslyn and Susan Mniszewski

Modeling, Algorithms, and Informatics (CCS-3)
joslyn@lanl.gov, smm@lanl.gov

Report Prepared for
"Advanced Knowledge Integration for Assessing Terrorist Threats" Project
October 9, 2002

Contents

List of Figures

List of Tables

1 Introduction

The LANL 2003 Reserve LDRD Project "Advanced Knowledge Integration for Assessing Terrorist Threats" provided a curated database of documents tracking various suspected terrorists, and sometimes including information about their particular expertises, and their involvement in particular terrorist events or groups. This is a report on two of the technical components aimed specifically at this relational nature of the data.

VisTool was developed in prototype form originally for a research project sponsored by the IRS to identify patterns of criminal fraud within databases of electronically filed tax returns. It was developed for the dual purposes of providing a schema-specific visualizing front end for analysts to examine the source database, and to provide a platform within which to implement and explore our research algorithms in user-guided knowledge discovery. In this project, we recovered a prior prototype implementation, and deployed it against the current project database. In Sec. 3 we report on some of VisTool's capabilities, and the results of this deployment.

Formal Concept Analysis (FCA) is a methodology developed to provide an unbiased, visual display of the complex sub-relations present in database tables. It shows great promise for supporting both manual inspsection and query of relational data, and automated hypothesis generation and analysis. In this project, we prepared data appropriate for FCA analysis, and did such analysis using third-party tools. In addition, we have identified and begun exploration of some significant research questions. In Sec. 4 we report both on FCA in general, and the results of this particular deployment against the project database.

We begin in Sec. 2 below with an explication of some of the key ideas in relational databases, illustrating them with a particular simple example, and introducing the structure of the project database as well.

2 Relational Data

In addition to the sheer quantity of information available to today's analyst, knowledge integration is made difficult because of the vast variety of different *kinds* of information available in knowledge bases. In any particular context, an analyst can be dealing with signal data, text documents, images, maps, diagrams, etc., all simultaneously.

In addition to these "raw" sources, the analyst will frequently be interacting with proper **databases**. Here we're referring to data which is relatively highly structured, parsed into separate fields according to its meaning, and arranged into an organized form. This form is almost always *tabular*, usually with multiple tables interacting in a **relational schema** to form what is called a **relational database**.

Almost always these tables are generated by human curation. Typically people over time simply identify and categorize the available information, and enter it into tables. Sometimes they can be assisted by computers, for example by parsing data from structured text sources or some other raw source which has a regular structure.

The processes of working with relational databases can have somewhat contradictory aspects. On one level, their high degree of organization makes them accessible to visual inspection and reasonable queries. But on the other, their potentially vast size, coupled with the potential complexity of their relational schemata, can make finding either explicit and implicit information present in them quite daunting.

Recent years have seen a proliferation of methodologies and tools to help analysts handle relational data, and we have been active in this area this year in both our FCA and Vistool link analytical methods, both in this project and elsewhere.

In this section we lay the groundwork for the discussion of VisTool and FCA by first introducint some of the important basic concepts of relational theory, and then by illustrating these through the description of our project database in relational terms.

2.1 Fields, Records, Relations

We first introduce some basic database concepts.

A **field** is a particular kind of information which is kept track of, corresponding to a particular meaning. Examples could be anything, someone's name, the date of an event, the description of an organization, etc. Fields are **typed** according to their mathematical structure, typical examples being words, numeric integers, floating point numbers, dates, paragraphs, booleans (True/False), etc.

A **key** is a field or combination of fields which are used to "anchor" the other field. These are the fields by which the others are kept track of, or the way you "look data up". For example, in a table of people characteristics, the key field could be the person's name (one field), or a combination of their first and last names (two fields), or their social security number, or some biometric data like

a fingerprint.

The essential characteristic of key fields is that together they are *unique* with respect to the data being kept track of. Sometimes key fields are automatically and arbitrarily created, for example by creating some unique record ID or other unique count to distinguish otherwise unidentified observations.

A **record** is a collection of field values together with some unique key value. It corresponds to a particular fact or observation.

Records are organized into **tables**. Each row is a record, and each column is a field in a record. Thus the cells of the table hold the values of the fields for the corresponding records.

Mathematically, we represent a field as a **dimension**, a set X of possible values, and consider a collection of fields as a collection of such dimensions $\{X_i\}, 1 \leq i \leq N$. A record is then a point $\vec{x} \in \bigtimes_{i=1}^{N} X_i$ in the cross-product of all the dimensions. Some subset of dimensions $\{X_{i'}\} \subseteq \{X_i\}$ (typically a single key dimension $X_{i'} \in \{X_i\}$) is identified as a key, such that those fields are uniquely populated. The **database** $\mathcal{D}$ is then a collection of record $\mathcal{D} := \{\vec{x}_j\}, 1 \leq j \leq M$ arranged in a table, one vector $\vec{x}_j$ in each row. To indicate relative sizes, sometimes we denote $\mathcal{D}_{N,M}$ to indicate a database with N dimensions (columns) and M data points (rows).

In later sections of this report, we will work with the following example, which we'll call the "simple example". Let $N = 4$ and let

$$X_1 = \{1, 2, 3, 4\}, \quad X_2 = \{\text{Andy, Bob, Cliff, Don}\}, \quad X_3 = \{\text{T, F}\}, \quad X_4 = \{1, 2, \ldots, 9\},$$

where:

- X_1 is the number of some observed meeting;
- X_2 is a person reported attending the meeting;
- X_3 is True if the person was hosting the meeting, and false if the person was a guest; and finally
- X_4 is the key field, a unique record identifer.

Then consider the relation show in Table 1. The key field X_4 is shown on the left, and the data fields X_1, X_2, X_3 on the right. Here we have $M = 9$ and, e.g., $\vec{x}_8 = \langle 4, \text{Bob, F}\rangle$, so that at meeting # 4, Bob was observed as a guest, and $j \in X_4$.

Of course, real database are much larger, with N (number of fields) in the hundreds, and M (number of records) in the thousands or more. In the open source database for this project, we used a relatively small number of fields:

X_1**: People:** List of 51 people, shown in Fig. 1.

X_2**: Events:** List of 13 terrorist events, shown in Fig. 2.

X_3**: Groups:** List of 11 terrorist groups, shown in Fig. 3.

X_4**: Expertises:** List of 8 expert skills, shown in Fig. 4.

ID	name
4	Abdullah Ahmed Abdullah
24	Fathur Rohman al-Ghozi
32	Anas al-Liby
41	Omar Abd al-Rahman
44	Khalid al-Shanqiti
45	Marwan Al-Shehhi
52	Ayman Mohammed Rabie al-Zawahiri
57	Ahmed Ibrahim A. Al Haznawi
61	Satam M.A. Al Suqami
64	Ahmed Alghamdi
65	Hamza Alghamdi
66	Saeed Alghamdi
67	Nawaf Alhazmi
68	Salem Alhazmi
72	Khalid Almihdhar
73	Ahmed Alnami
74	Abdulaziz Alomari
75	Wail M. Alshehri
76	Mohand Alshehri
77	Waleed M. Alshehri
84	Muhammad Atef
86	Mohammed Atta
87	Muhsin Musa Matwalli Atwah
89	Faiz Abu Bakar Bafana
103	Osama Bin Laden
110	Ahmed Brahim
125	Ali Saed Bin Ali El-Hoorie
129	Abdelkader Mahmoud Es Sayed
131	Mustafa Mohamed Fadhil
134	Khalid al Fawwaz
135	Ahmed Khalfan Ghailani
142	Ahmed Mohammed Hamed Ali
144	Hani Hanjour
151	Raed Hijazi
161	Ziad Samir Jarrah
174	Peter Kinyanjui
183	Christian Manfred G.
188	Amine Mezbar
193	Fazul Abdullah Mohammed
196	Yunus Moklis
197	Majed Moqed
198	Zacarias Moussaoui
199	Fahid Mohammed Ally Msalam
209	Nizar Ben Mohammed Nawar
213	Mouhamedou Ould Slehi
221	Ahmed Omar Abdel Rahman
232	Abdul Rehman Safani
255	Sheikh Ahmed Salim Swedan
265	Ustadz Nur Mohammed Umog
266	Abdul Rahman Yasin
276	Mohammed Haydar Zammar

Figure 1: People.

Record #	Meeting #	Person	Host?
1	1	Andy	T
2	1	Cliff	T
3	2	Andy	F
4	3	Bob	F
5	3	Cliff	T
6	3	Don	F
7	4	Andy	T
8	4	Bob	F
9	4	Don	F

Table 1: Simple example data table.

```
+----+--------------------------------+
| ID | name                           |
+----+--------------------------------+
|  1 | U.S. Embassy Bombing--Kenya    |
|  2 | Philippine Embassy Bombing     |
|  3 | World Trade Center Bombing     |
|  4 | La Griba Synagogue Explosion   |
|  5 | Pennsylvania Flight            |
|  6 | USS Cole Bombing               |
|  7 | Khobar Towers Bombing          |
|  8 | Luxor Massacre                 |
|  9 | World Trade Center 9/11 Attack |
| 10 | U.S. Embassy Bombing--Tanzania |
| 11 | US Embassy bombing--Paris      |
| 12 | US Embassy Bombing...New Delhi |
| 13 | Pentagon Attack                |
+----+--------------------------------+
```

Figure 2: Events.

ID	name
1	Al-Qaeda
3	Egyptian Islamic Jihad
4	Abu Sayyaf
6	Jemaah Islamiah
7	Shura Council of al-Qaeda
8	Islamic Army of Aden
9	al Jehad
11	Al-Gama'a al-Islamiyya
12	Libyan Islamic Fighting Group
16	Jaamat al-Islamie
26	Moro Islamic Liberation Front

Figure 3: Groups.

ID	name
1	Explosives
2	Terrorist Operations
3	Computers
4	Pilot
5	Financial
6	Agricultural Field
7	Martial Arts
8	Military Strategy
9	Religious Scholar
10	Military Advisor

Figure 4: Expertises.

Meeting #	Meeting name
1	First meeting
2	Second meeting
3	Third meeting
4	Fourth meeting

PersonID	Person Name
a	Andy
b	Bob
c	Cliff
d	Don

Table 2: (Left) Meeting table. (Right) People table.

MeetingID	PersonID
1	a
1	c
2	d
3	b
3	c
3	d
4	a
4	b
4	d

Table 3: The "meetings-people" join table.

2.2 Relational Schemata

Complex databases typically consist not just of single tables, but multiple interacting tables, involving particular groups of fields. Such a **relational decomposition** is most efficient for maintenance and manipulation of complex data.

To continue our simple example, it's more likely that people and the meetings they attended would be listed in separate tables, one for meetings (X_1) and one for people (X_2), as shown in Table 2. Pairs $\langle x_1, x_2 \rangle \in X_1 \times X_2$ are then drawn from the key fields of these two tables, and combined into a single **join table**, as shown in Table 3.

There exists a **relational schema** among the three tables, a portion of which is shown in Fig. 5. For each table, the figure shows the table name, and below the fields. Key fields for each table are shown in italics, and are used to link the different tables together in a mathematically meaningful way.

The single-headed arrow indicates a **many-to-one relation**. Thus, each person record (e.g. Andy) maps to multiple meeting-people records (e.g. records $\langle 1, a \rangle, \langle 4, a \rangle$), as does each meeting record (e.g. Meeting 1 to records $\langle 1, a \rangle, \langle 1, c \rangle$). So we call this table the $X_1 \times X_2$, or Meetings $\times$ People, table.

Note that other join tables are possible, for example $X_1 \times X_3$ = Meetings $\times$ Host. In addition to these two-dimensional, or **binary**, join tables, higher dimensional joins, in this case $X_1 \times X_2 \times X_3$, are also available. We will also consider higher order relations involving not just cross-products $\times$, but also **unions** $\cup$, below in Sec. 4.1.

A portion of the relational schema for the project database is shown in Fig. 6. The original Lotus

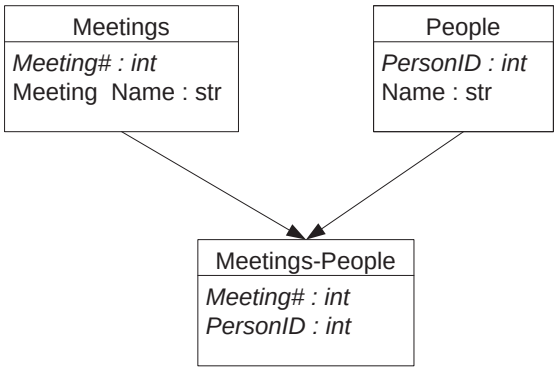


Figure 5: Portion of the relational schema of simple example.

Record	Meeting #	Host?
1	1	T
2	1	T
3	2	F
5	3	T
6	3	F
7	4	T
9	4	F

Table 4: Simple example view.

Notes table as we received it was mapped into the table `p_documents`, and includes many fields, including the ones listed. Note that it is not connected to any of the other tables. We took this original table and decomposed it into the relational schema shown. The base tables `People, Events, Groups`, and `Expertises` are as shown in Figs. 1–4, and some of the relevant join tables considered underneath: People × Events, People × Groups, and People × Expertises. In addition, we constructed some other joins for analytical purposes, as discussed in Sec. 4.4.

2.3 Views

Given a database table, we are usually interested in examining only a portion of it. To identify this portion completely, we therefore have to identify both a subset of columns (fields, dimensions) and a subset of rows (records). Such a dual subset we call a **view** of the database.

A view of the simple example is shown in Table 4. In this case, we restrict the view to the columns X_1, X_3 and to the records $1, 2, 3, 5, 6, 7$ and 9.

3 VisTool

Given this understanding of relational databases, we now describe VisTool, a link analysis package intended to aid in user-guided knowledge discovery of such databases.

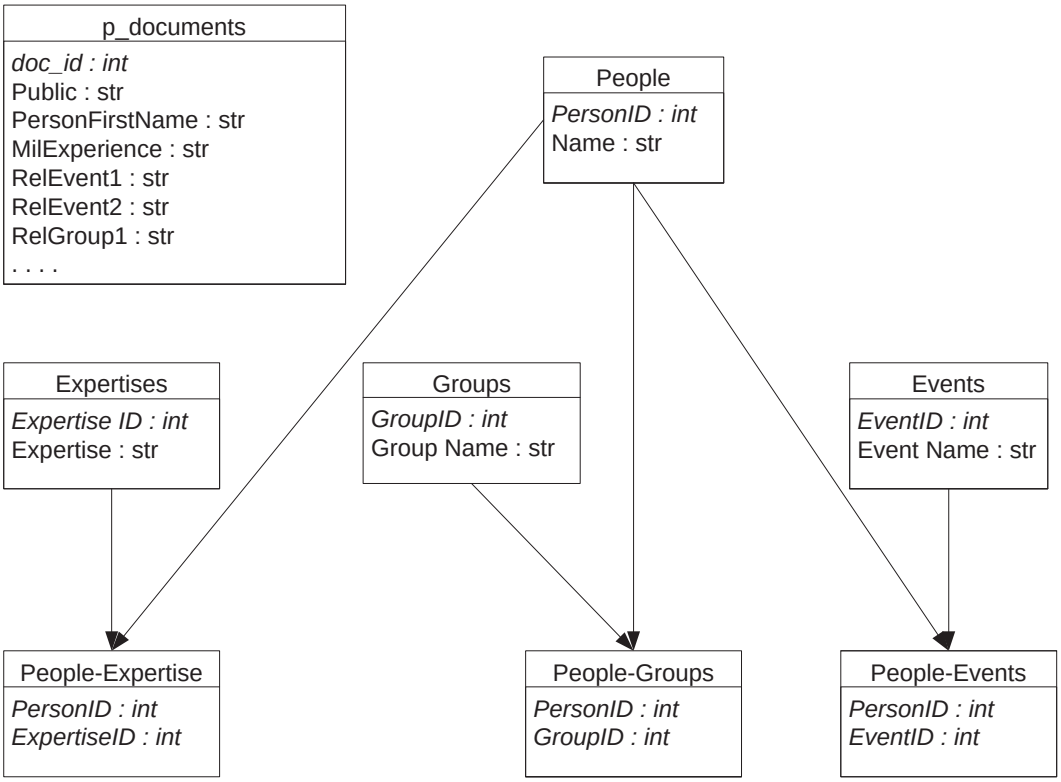


Figure 6: Portion of the relational schema of the project database.

3.1 Link Analysis

As can be seen, when relationally decomposed, even this small portion of our project database is quite complex. Given the $N = 4$ fields we're considering, there are, of course, 4 base tables for each field, and then up to $\binom{4}{2} = 6$ binary join tables, 4 ternary joins, and the complete 4-dimensional join, or a total of 15 possible total tables to consider, excluded tables involving unions (see Sec. 4.1). In addition, for example for the $X_1 \times X_2$ People-Events table, there are potentially $|X_1| \cdot |X_2| = 51 \cdot 13 = 663$ unique records.

So given such a database, our goals include understanding how to generate significant hypotheses concerning such questions as:

1. Which fields are important?
2. For which subsets of the data?
3. Where are the "interesting" areas of structure or activity?

Given the database size and complexity, this is a bit of an overwhelming task. Even though in our case there are only 114 of 663 pairs actually present in People × Events table, it still daunting to consider all these possibilities.

So it is clear that computational tools are necessary to approach a truly large database involving multiple fields (name, aliases, citizenship, address, age, travel dates, education, etc.). On the other hand, the size and complexity of such databases is such that even the largest and most sophisticated computer-based systems are not now, and may never be, able to provide complete, automatic, answers to our questions.

Knowledge of this reality drives our approach to this kind of research and development. In particular, it is predicated on the idea that fully automatic knowledge discovery methods providing complete answers to our questions will *not* be feasible. Instead, we aim at methods which are:

- Appropriate for *moderately* sized databases (10^2 - 10^5 records).
- *Semi-automatic*, and
- *User expert guided*.

The basic idea is to provide an intelligent analyst, a domain expert with background and training in these kinds of mathematical and computer scientific techniques, with a suite of tools which will support him or her to iteratively guide search for areas of local structure.

VisTool supports one such broad methodology, which we call "link analysis" [20, 21]. This term has a definite, but small, presence in the literature [17]. To our knowledge the concept was developed in the mid 1990's within the law enforcement and anti-money laundering communities (see [29], for example), within which it has considerably more recognition.

It is significant to note that link analysis in our sense of discovery specifically in relational databases is usually *not* clearly distinguished from "network analysis" in the sense of the analysis of large, single-link networks. An example, again, is Kleinberg [24], whose approach is decidedly network-theoretical in our sense, despite being called link analytical. Thus establishing this term in a proper way may be difficult, but we believe proper to attempt at this time.

The kinds of questions which link analysis is intended to address concern *collections* of records distributed over *collections* of fields. So, for example, given such a collection of records, how do they implicate one collection of columns or another? Similarly, how do they implicate other connnected collections of records, perhaps being more, fewer, or somehow overlapping?

A central concept to our sense of link analysis is known as **chaining**. It works like this:

- Assume a database $\mathcal{D}_{N,M}$ with N dimensions and M data points.
- Define a "view" on $\mathcal{D}_{N,M}$ as its projection to a particular subset of dimensions $n \subseteq \{1, 2, \ldots, N\}$ and restriction to a particular subset of records $m \subseteq \{1, 2, \ldots, M\}$, denoted $\mathcal{D}_{n,m}$.
- Chaining then consists of moving from one particular view $\mathcal{D}_{n,m}$ to another $\mathcal{D}'_{n',m'}$, where $n \cap n' \neq \emptyset, m \cap m' \neq \emptyset$, or both, so that there are some rows and/or columns which are "held over" from the prior view.

Conceptually, first an intelligent analyst considers certain aspects (n) of a certain group of records (m), for example including the place of birth of a group of people who all went to the same school. She then chains to consider another aspect, say the zip codes ($n' \cap n = \emptyset$) of those of that group who went to Harvard ($m' \subseteq m$).

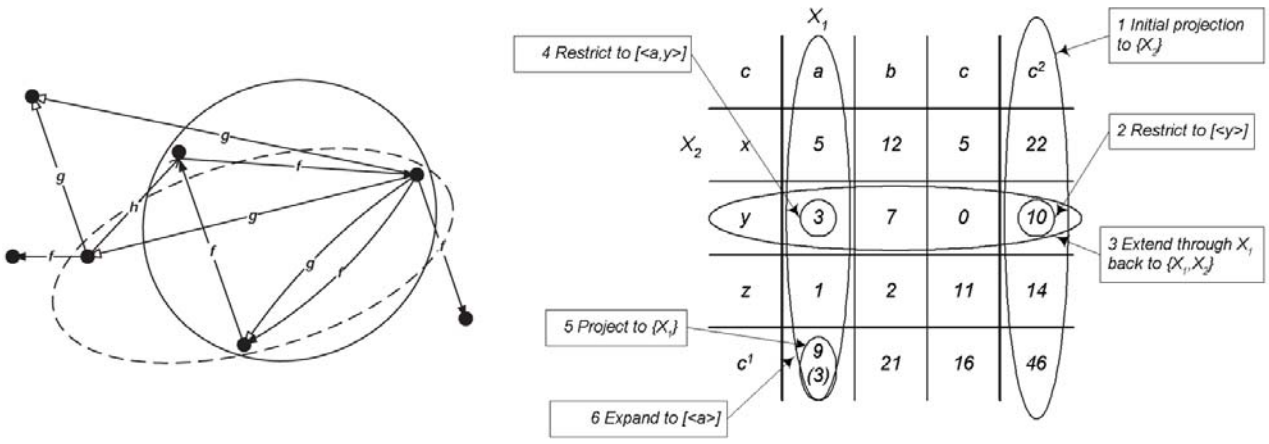


Figure 7: (Left) Chaining in a typed-link network. (Right) In a two-dimensional contingency table [23].

Fig. 7 illustrates this process in two different contexts. The right-hand side shows a database table illustrated as a "contingency table", indicating the number of observations of each record type. The cells indicate the number of records with a certain vector value, and the marginal counts are included on the edges of the matrix. The chaining process begins with an initial view $\mathcal{D}_{\{2\},M}$, all records projected onto the second dimension. The second step restricts this to $\mathcal{D}'_{\{2\},m}$, where $m \subseteq M$ now indicates those ten records $\{\vec{x}\} = \{\langle x_1, x_2 \rangle\}$ such that $x_2 = y$. In the third step, $\mathcal{D}''_{N,m}$ indicates the same set of records, but now extended back both dimensions $N = \{1, 2\} \supseteq \{2\}$. Similar other steps are indicated.

The left-hand side of Fig. 7 shows a database using a quite different representation, namely a **typed-link meta-network** [21]. Here nodes are data records, links are fields held in common,

and the type of the link indicates the type of the field. The concept of chaining is quite similar. The solid boundary indicates a collection of records $m = \{w, y, z\}$ viewed through the single field $n = \{f\}$, yielding $\mathcal{D}_{\{f\},\{w,y,z\}}$. The dashed boundary indicates the transition to a new view on a different field type and somewhat different records $\mathcal{D}'_{\{g\},\{x,w,z\}}$, so that $n \cap n' = \emptyset$, but $m \cap m' \neq \emptyset$.

3.2 Basic VisTool Operation

VisTool is an information theoretical data discovery and link analysis tool developed at the Los Alamos National Laboratory in 1996–1998 as part of a project to detect patterns of criminal fraud in IRS tax databases. VisTool combines multiple functions in one tool:

Schema Specific Interface: VisTool is customized by a knowledgable administrator to reflect a particular relational schema, for example as shown in Fig. 6. From that point forward, users are supported in that they themselves do not need to specify the relational connections among tables.

View Query Manager: Vistool supports the ability to construct and store queries supporting multi-dimensional views.

Data Viewer: Tabular and graphical data display, including statistical properties and histograms.

Data Exploration through Extension and Projection (DEEP): Support for a particular link analysis methodology called Data Exploration through Extension and Projection (DEEP) [23], which we only describe further here very briefly in Sec. 3.3.

For this project, we ported the VisTool legacy code to the project data server, and instantiated two modules:

p_documents Examiner: Allows browsing of the source Lotus Notes `p_documents` table.

peopleChar1 Examiner: Allows browsing of the relationally decomposed schema show in Fig. 6.

Fig. 8 shows the welcome screen for VisTool, including the links to both examiners. Fig. 9 shows the screen allowing selection of the columns for a particular view, and Fig. 10 shows the selection screen for the rows.

In the examples, we show selection of the view with columns $X_1 \times X_2 \times X_3 =$ People $\times$ Events $\times$ Groups, and for rows for all records with $10 \leq$ PeopleID ≤ 15. The result of executing the query is shown in Fig. 11. The bottom of the figure shows the tabular form of the query. The top shows a scatter plot of the two columns Events and Groups. Other views are also available.

3.3 Advanced VisTool Operation

The section above is a very brief description of some basic VisTool operations, effectively its value as a front-end to a relational schema. As such, it's redundant with a number of off-the-shelf utilities.

But VisTool is also intended as a platform for our research efforts, including various link analytical methods. In this section we briefly describe some of these capabilities.

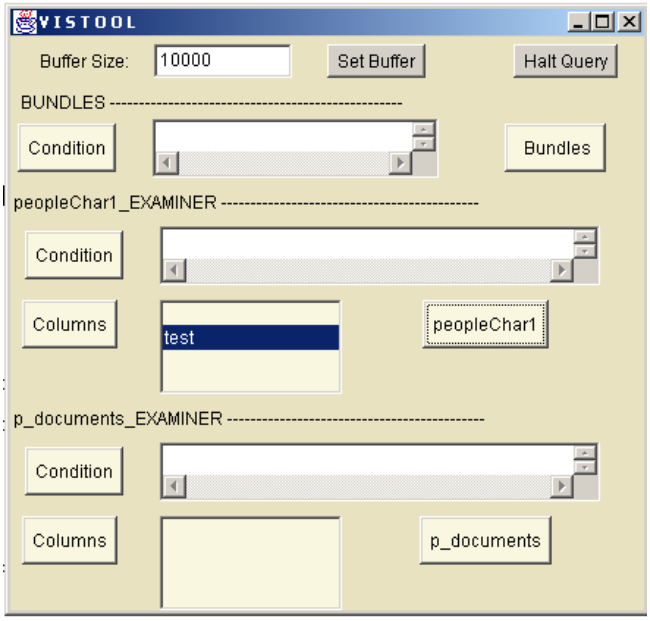


Figure 8: VisTool welcome screen.

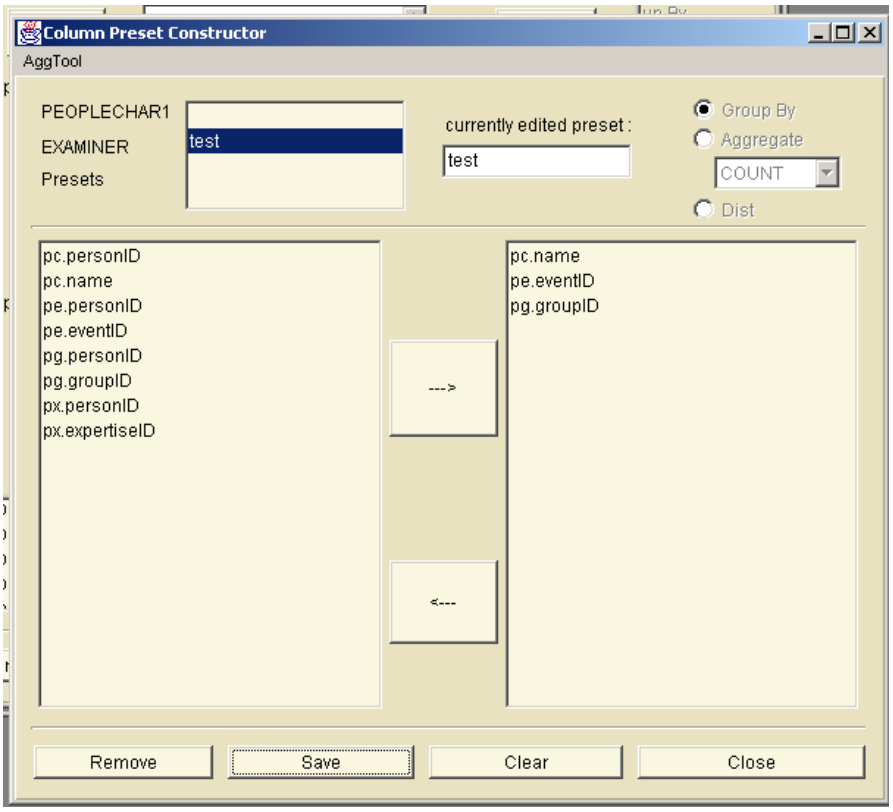


Figure 9: Column selector.

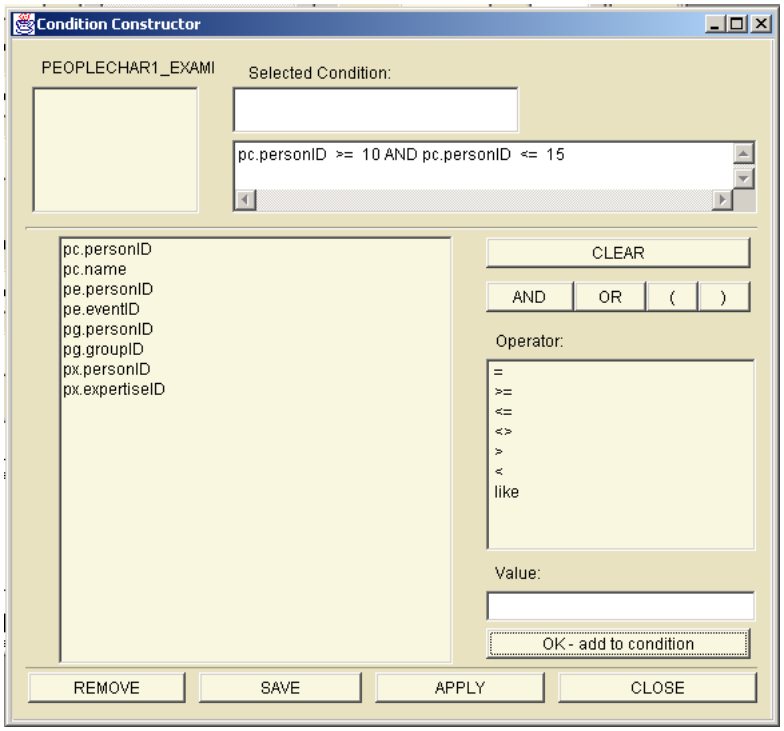


Figure 10: Rows selector.

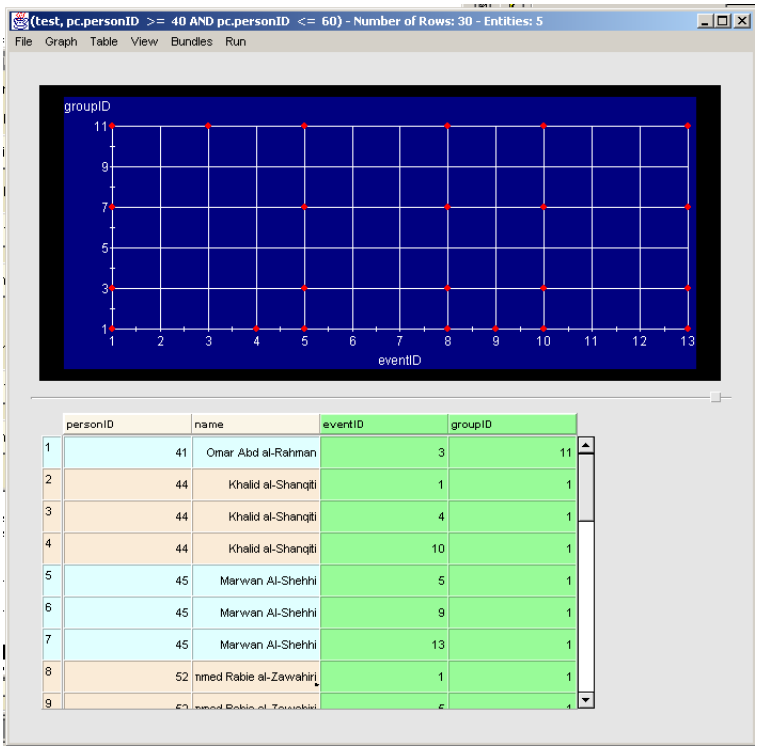


Figure 11: Query result.

Overlap: The DEEP methodology [23] is based on guiding the user through chaining operations through statistical measures on distributions of various available views. A brief example of a component of this capability is shown in Fig. 12, showing the Overlap window available after selecting the Events column from the query table from Fig. 11. The counts of the various events are shown, together with different statistics on the distribution of these counts. In this way, the user is able to judge the relative value of selecting this particular field for projecting the view through. More details of this methodology are available in a technical report [23].

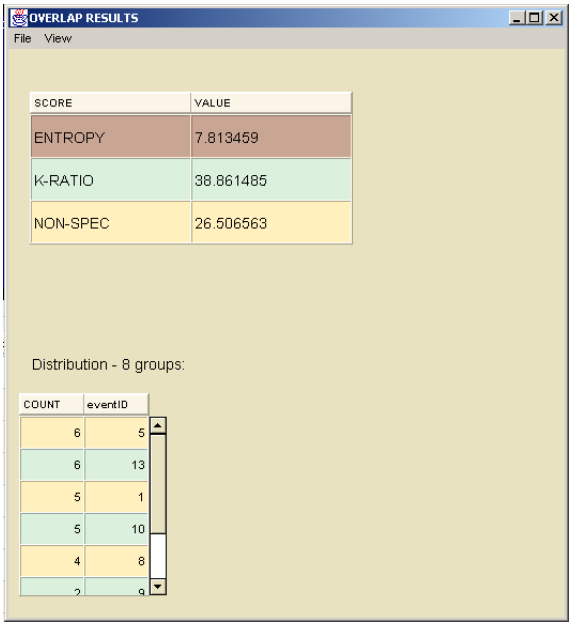


Figure 12: Distribution of Events for query.

Subsetting: Given a particular data view, the user can select various columns and rows, thereby creating a sub-view, and then "subset" those into a new view window.

Bundles: Once subsetted, these sub-views can be saved permanently to disk as "bundles", for later retrieval and examination.

Aggregate Queries: Aggregation operations, such as summation and averaging, are available across different queries, and are used interactively with the DEEP methodology.

3.4 Future Directions

In this project, our goal was simply to deploy VisTool against the project database, thus demonstrating its applicability. We have been successful in this goals, and in so doing, we have identified a number of directions for future development:

Integrated Front End: As a generalized query generation and management platform, VisTool can be generally useful for various data domains, and to support various scientific methodologies. In particular, its support for the generation and identification of various database

views suggests that it would be especially useful as an integrated front end, both to directly support analysts, and to feed other back-end scientific platforms, for example for FCA or Rocha's network analysis. To be more specific, VisTool can be extended to include buttons to take a particular database view, and then run either an FCA or network analysis, probably showing the results through another software platform.

Classified Database: VisTool can be instantiated with respect to other databases, for example within a classified environment.

Further Prototype Development: VisTool development was interrupted in 1998 when its funded effort in the IRS project was terminated. There is a substantial need for continued development of the current prototype implementation.

4 Formal Concept Analysis (FCA)

Formal Concept Analysis (FCA) [16, 18] is a method for computer representation and analysis of complex relational data. FCA works by taking a boolean, binary relation, and calculating the hierarchical relations present among all the distinct subrelations. Such a representation supports both automatic inference and user-guided discovery and exploration of hypotheses.

FCA depends on the well-established subfield of combinatorics called Galois theory [11], and is also closely related to methods in "association rule extraction" [2, 12, 15] and other knowledge representation techniques such as "conceptual graphs" [37, 46].

FCA is becoming established in a number of areas of information science [7, 25]. Concept lattices have been combined with natural language processing for information retrieval [31, 32]. They have been also been used in biology [27, 28], chemistry [3], environmental science [5] for structuring phenotypes/genotypes in behavior genetics [13], and incineration plant process control [42].

Extensions to basic FCA include "fuzzy concept lattices" [4] and "iceberg concept lattices" [41], which have been used for for database marketing, configuration space analysis, transformation of software class hierarchies, ontology learning, and database tuning.

A somewhat detailed mathematical description of FCA is provided in Appendix A. In the next section we provide a qualitative, relatively informal description of the method, including treatment of our simple example, and then an analyst's perspective on how to use FCA. We conclude with a detailed analysis of the project database, followed by future directions.

4.1 Relatively Informal Introduction to FCA

FCA works by taking a boolean, binary relation; then calculating the connections, called **concepts**, among distinct groups of rows and columns; and then calculating the hierarchical relations between these concepts. We describe these steps in detail now, using our simple example.

Identify Binary Relation: First, the user needs to identify a binary (that is, two-dimensional) relation. In our simple example, we'll use $X_1 \times X_2 =$ Meetings $\times$ People, as shown in Table 3.

X_1/X_2	a	b	c	d
1	√		√	
2	√			
3		√	√	√
4	√	√		√

Table 5: Scaled version of Meetings × People.

$X_1/X_2 \cup X_3$	a	b	c	d	Host True?	Host False?
1	√		√		√	
2	√					√
3		√	√	√	√	√
4	√	√		√	√	√

Table 6: Scaled version of Meetings × (People ∪ Host).

Scale to Recover Boolean Relation: FCA requires that each cell in the table be occupied by a Boolean (0/1 or True/False) variable, so that $x \in \{0, 1\}$. In this case, the rows are labeled by meeting #, but the cells contain a variable $x \in \{a, b, c, d\} = X_2$. To recover a boolean relation, we perform a **scaling** relation, effectively treating each value in X_2 as a distinct column. This results in Table 5, where √ indicates 1 or True.

The scaled table is called a **context**, where the rows are called **objects**, and the columns **attributes**. The interpretation is that the objects (here, meetings), either have or don't have particular properties (here, whether a particular person attended).

Use Unions to Accommodate Multiple Fields: We've said that FCA requires a context, which is a binary, boolean relation. Scaling converts non-boolean to boolean relations, and relations of more than dimension two can be accommodating by considering **unions**. Table 6 shows an example for $X_1 \times (X_2 \cup X_3)$. Effectively, the scaled attributes for two variables (in this case, People and Hosts) are "laid beside" each other, and each is considered as just more attributes for the objects (people).

Identify Concepts: A **concept** is a statement that a particular group of objects involve a particular group of attributes, and *vice versa*. There are various ways to calculate concepts, but the essence of it is this: pair a particular collection of columns (resp. rows), with exactly all the rows (columns) associated with all of those columns (rows). Call the collection of rows the **extent**, and the collection of columns the **intent**, and the pair the **concept**.

For convenience, denote e.g. $134 := \{1, 3, 4\}$, $ac := \{a, c\}$, etc., and

$$134/ac := \langle \{1, 3, 4\}, \{a, c\} \rangle \, .$$

Also, denote $0 \subseteq X_1 = \emptyset$. Then, for a given set of rows $A \subseteq X_1$ (resp. columns $B \subseteq X_2$), let $A' \subseteq X_2$ (resp. $B' \subseteq X_1$) be those columns (resp. rows) associated with *all* the columns $x \in A$ (resp. rows $y \in B$). So given A, calculate first $A' \subseteq X_2$, and then calculate $A'' \subseteq X_1$. Then record the pair $C := A''/A'$ as a concept. Do this for all $A \subseteq X_1$. Dually, this can be done for all $B \subseteq X_2$, now recording pairs $C := B'/B''$ as concepts, the result will be the same.

Concept #	B	B'	A	A'	Concept #
8	$\emptyset$	1234	0	$abcd$	1
6	a	124	1	ac	2
	b	34	2	a	
5	c	13	3	bcd	3
	d	34	4	abd	4
	ab	4	12	a	
2	ac	1	13	c	5
	ad	4	14	a	
	bc	3	23	$\emptyset$	
7	bd	34	24	a	
	cd	3	34	bd	7
	abc	0	123	$\emptyset$	
4	abd	4	124	a	6
	acd	0	134	$\emptyset$	
3	bcd	3	234	$\emptyset$	
1	$abcd$	0	1234	$\emptyset$	8

Table 7: A, A' and B, B'.

Concept #	Concept
1	$abcd/0$
2	$ac/1$
3	$bcd/3$
4	$abd/4$
5	$c/13$
6	$a/124$
7	$bd/34$
8	$\emptyset/1234$

Table 8: Concepts in the simple example.

Now consider our simple example. Table 7 shows all A/A' and B/B' pairs in our simple example. Now try to identify some concepts. Consider $A = 34$. Then $A' = bd$. Similarly, consider $B = bd$. Then $A' = 34$. Thus 34 is an extent, bd an intent, and $C_1 = 34/bd$ is a concept. This could have also been verified by constructing $A''/A' = (bd)'/bd = 34/bd$.

All the concepts, whether A/A' or B/B' pairs, are also listed with the extents and intents on the outside of Table 7. Note that for each concept, there's the same pair $A/A' = B'/B$ pair. Note also that not all possible A/A' or B/B' pairs are concepts. Through this process, we can determine that there are eight concepts in the simple example, as shown in Table 8.

Show Lattice Relations Among Concepts: The concepts aren't distinct from each other, but rather are related to each other hierarchically, which we show in a lattice diagram. In particular, for two concepts $C_1 = A_1/B_1, C_2 = A_2/B_2$, we say that one is "below" the other, or $C_1 \preceq C_2$, when $A_1 \subseteq A_2$. It follows that $B_1 \supseteq B_2$. Then, simply arrange all the concepts according to this ordering.

The results are shown on the left side of Fig. 13. Each of the eight concepts is a node in the lattice. Each node is labeled on the left with its concept number in italics, and on the right on the upper side with the attributes (intent), and lower side with the objects (extent).

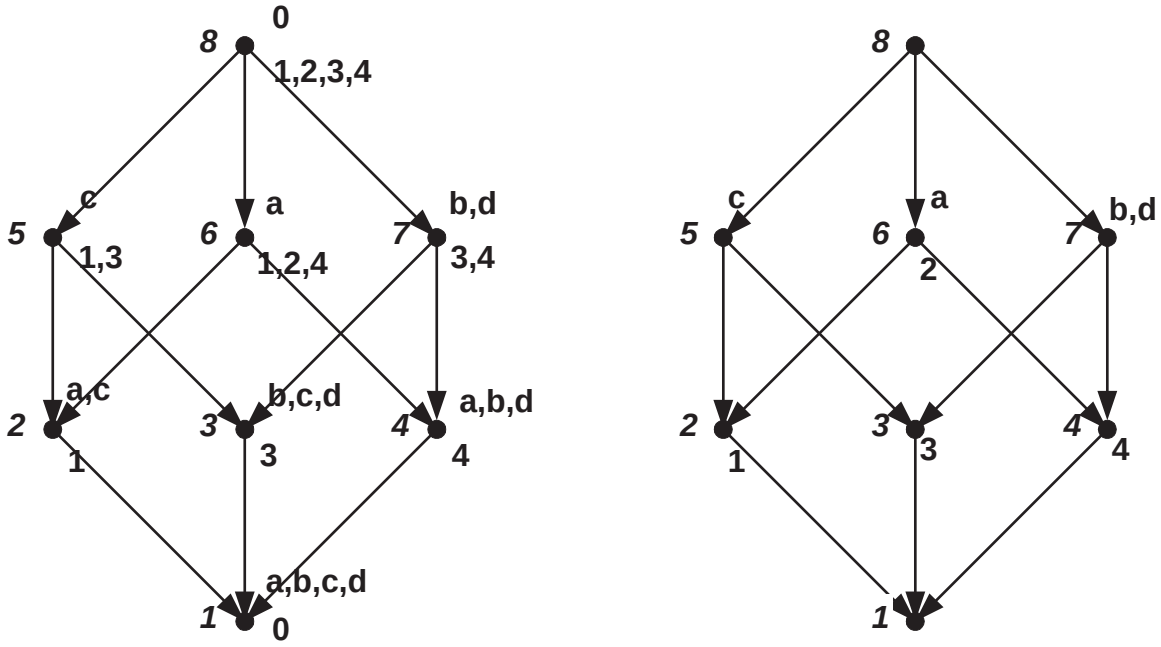


Figure 13: (Left) Concept lattice for simple example. (Right) Reduced concept lattice.

Notice the dual nature of the lattice: the upper nodes have many objects and few attributes, while the lower nodes have many attributes and few objects. Notice also that any object present in any node is also present in all nodes above it, and conversely any attribute present in any node is also present in all nodes below it.

We can exploit these regularities to produce the **reduced lattice** shown on the right side of Fig. 13. Here, each object and each attribute is shown only once, in the "first" place it occurs. These lattices are isomorphic, if it's recognized that in the reduced lattice any object is "really" present in all the nodes above it, and conversely any attribute is really present in all the nodes below it. From here on, we will only consider reduced lattices.

Interpret Lattice: Given a reduced lattice as in the right side of Fig. 13, some possible interprtation are as follows.

- For a given single object, look above. The attributes above are all of those associated with the object, while the other objects above are those implied by the given object. In the example, meeting 4 is associated with people a, b and d. Also, meeting 4 implies meeting 2, in the sense that all the people in meeting 4 are also in meeting 2.
- Similarly, for a given attribute, look below. The objects below are all of those associated with the attribute, while the other attributes below are those implied by the given attribute. In the example, person a is involved in meetings 1,2 and 4.

- Then, consider groups of objects or attributes. Begin with pairs. For pairs of attributes, consider their common objects below, and for objects, their common attributes above. In the example, consider people a and c, which together involve only meeting 1. Note that person a also involves meeting 2, but person c does not. This can be extended to groups of more than two as well.
- Consider "chaining" among nodes. In the example, start with person a. It involves meetings 1,2 and 4, but in different ways. With meeting 2, it's by itself, whereas with meeting 4, it connects to people b and d. Thus meeting 4 is the connection between person a on the one hand, and people b and d on the other.
- Consider co-occurring objects and attributes. In the example, people b and d occur together in concept 7. Thus they are equivalent, in involving only meetings 3 and 4.
- Consider the positions of objects and attributes within the lattice. We remarked above that in general, objects appear low on the lattice, and attributes appear high. Low objects are centrally connected within the table, as are high attributes. Thus, note exceptions, objects which are high and attributes which are low. In the example, consider meeting 2, which appears uncharacteristically alone on the "second row" of the lattice. Note that in the table, meeting 2 involves only person a, and is the only meeting which involves only one person. Moreover, it's included in the concepts involving meetings 1 and 4, through the involvement of person a.

Work Iteratively With the Table: Note that all of the relations discussed above are available from the original data table. Indeed, the lattice is simply an alternative representation of the information in the table. Intuition can be assisted by checking one's conclusions about the lattice against the table, and *vice versa*.

Also, the lattice provides an unbiased representation of the table, in that the ordering of the rows and columns, which is otherwise arbitrary, has no bearing on the structure or layout of the lattice.

Consider Implications: There are some hard and fast, deductive facts available from the lattice which can be automatically generated. These are called "implications" or "association rules". In general, identification of such rules from relational data is a large sub-field within the overal knowledge discovery community. As such, FCA is an important methodology for identifying such rules.

While any rule generation method can suffer from a proliferation of relatively useless rules, in our example we can nonetheless glean a number of interesting facts, such as:

$b \leftrightarrow d$**:** b and d are equivalent.

$1 \rightarrow 2, 4 \rightarrow 2$**:** Meeting 2 is implied by either meetings 1 or 4, since each requires person a, which is sufficient for meeting 2.

Note the following:

- The procedures outlined above to take a scaled relation and generate the reduced lattice, and to generate implications, are best supported through computational tools (see Sec. 4.3). This simple example has been done deliberately "by hand" to provide a simple illustration of the method "in action".

- These instructions are quite qualitative, gleaned from our experience actually working with these lattices. Each investigator will develop their own intuitions (see, for example, Sec. 4.2). We are actively working with the community and within the scientific literature to both develop and advance our understanding of such methods.
- Moreover, we are actively researching statistical measures on lattice-valued spaces [22, 33, 34] which will be especially useful for suggesting hypotheses to investigators in a semi-automatic method. An example would be of the form that $\Pr(b = d) > \Pr(a = c)$, because $|34| > |1|$.

4.2 Using FCA: An Analyst's Approach

In this section we our expereince of how we as "analysts" applied FCA to the project database. In particular, we show a number of contexts available from the overall project database, their concept lattices, and their generated rules. We then provide our own interpretation of the results.

As a tool to support analysts in computer-assisted knowledge discovery, it is important to note that while concept lattices are useful for finding relationships in data and understanding structure, such as hierarchies and networks or the lack there of, this information is available from the interpretation of a lattice display and an implication list. Thus, it is both dependent on the quality of the original context provided to it, and it is still up to the analyst to determine whether the suggested structure is relevant.

Since the resulting concept lattice is only as good as the context data it was generated from, it can also be used for identifying missing or incorrect information. This technique is typically used with qualitative data, such as names, places, groups, etc. Binned quantitative data can also be used.

One could certainly throw an entire database at an FCA tool, resulting in a very large lattice that could only be viewed in pieces or would require sophisticated automated analysis to identify interesting structure. On the other hand, an analyst may be more focused on part of the data or on answering some questions involving some portion of the data, focusing on one question at a time results in a manageable lattice that can be analyzed on a workstation.

An analyst may have a very broad question in mind or only have a fuzzy idea of what they are looking for. The exploration process should entail identifying a starting question and further refining or redirecting based on the resulting concept lattices.

A suggested approach for an analyst is:

- Define a question (e.g., "Is there a relationship between events?").
- Create an object/attribute/relation model (context) based on data available.
- Assemble appropriate data for input.
- Generate concept lattice.
- Analyze lattice and implications (visually and with whatever automatic methods are available).
- Change question or add/subtract data until useful results are obtained.

A starting question must be defined by the analyst. Some details on the other steps follow.

Defining the Model, Assembling Input, Generating a Lattice: One must determine what should be the objects and what should be the attributes and what is the relation based on the question posed.

For example, my question might be "Are a set of terrorist events related?" If data is available on people and the events they were involved in, the objects can be the people and the attributes can be the events. In this case the relation is relating events through people, or Events $\times$ People.

The data is assembled into the input format required by the FCA tool. This can be through database access from a tool directly or may require using SQL to extract the data and transforming it to the tool's required input format using Perl scripts.

A lattice is generated running any of the tools available (see Sec. 4.3), such as ConImp/Diagram or Concepts/Graphplace. A displayed lattice and/or a list of implications are the results.

Interpreting the Output: The generated lattice consists of a set of connected nodes each representing a separate concept. The attribute designation(s) appear above the concept node and the object designation(s) appear below. The object desigination(s) are shown with the last concept node they are involved in. An object is involved in the concept node it appears on and all the concept nodes in the connected sub-graph above. The attribute designation(s) are shown with the first concept node they are involved in. An attribute is involved in the concept node it appears on and all the concept nodes in the connected sub-graph below.

Viewing the lattice, simple relationships and structures may be obvious. Relationships can be identified automatically by generating implications or rules. Each implication is generated in terms of the attributes and is associated with one concept node. Related implications can imply stronger coupling between a set of attributes. Additional observations, such as noting that single objects are tying attributes together, require manual interpretation of the lattice display or additional mining tools.

Tools Used: In working through these examples, it became clear that some other supporting tools were necessary. The generated implications require intelligent processing for grouping and matching to text, as well as a visual way of showing an implication and where it is in the lattice at the same time. Supporting graph theoretic primitives would also be useful in locating interesting structure in the lattice and relationships identified through single entities.

4.3 Supporting Tools

This project was involved in exploring the usefulness of Formal Concept Analysis (FCA) for Homeland Security, not in developing tools at this time. A number of existing and evolving FCA tools were investigated. No one tool embodies all the features that we have found useful. The tools are summarized in Table 9. Comments about the tools as Data Preparation/Lattice Exploration pairs follow. The table and comments are based on a previous survey [14], Web page descriptions, and hands-on experience.

It should be noted that there is no one FCA tool available today that provides all the capabilities that would be useful. Early work in FCA's was centered and is still centered in Germany, which explains the German user interfaces on a number of these old tools. These existing tools focus on lattice generation, and visualization of lattices and sub-lattices. Very little automatic processing is performed on a lattice, other than generation of implications. Measures for calculating the distance

Tool	Platform/ Language	User Interface	Database Access	Data Preparation	Lattice Generation	Lattice Display	Lattice Print	Other Capabilities
Anaconda	Windows, C++	German	√	√	√	√	√	
Toscana	Windows, C++	German	√			√	√	Nested lattices, lattice exploration
Cernato	Windows	German	√	√	√	√	√	
Conimp	Windows, Pascal	English			√			Implications, attribute exploration
Diagram	Windows	English, German				√	√	
Concepts	Linux, C	English			√			Source available
Graphplace	Linux, C	English					√	Source available
Concept Explorer	Java	English		√	√	√	√	Implications, association rules, attribute exploration
ToscanaJ	Java	English	√			√	√	

Table 9: Available FCA software tools.

between two concept nodes would be useful as well as graph theoretic measures for identifying structure or lack of structure within a lattice.

4.3.1 Anaconda/Toscana

Anaconda/Toscana [1, 44] have been used extensively in the past. They provide database access and extensive visual processing. A demo version was obtained that would only operate on included data and had a German user interface. Unfortunately, it was not very useful due to these constraints.

4.3.2 Cernato

Anaconda and Toscana evolved into a tool that is now sold by Navicon called Cernato [6]. It provides database access (though not in the demo version), Excel-style data entry/display, and extensive visualization and filtering. The user interface is available only in German, which can be a problem. The lattice display ability allows rearrangement of the nodes, but moving nodes in one part of the lattice may cause nodes in another part of the lattice to move. This is a problem if you want complete control over the layout. This is a good beginning for an FCA tool other than the user interface and graphics problem. This is a product of Navicon and is very inexpensive.

4.3.3 ConImp/Diagram

ConImp [10] is one of the first FCA programs that was widely used. It is available for Windows and Linux platforms. ConImp takes a straight forward text input `CXT` file for defining the objects, attributes, and relations between them. It includes an editor for further context manipulation such as reduction or transposition. It can generate the lattice but does not display it. It does generate implications such as the Duquenne-Guigues base described in the documentation as implications with an independent premise. The generation of implications shows an example of automated processing on concept lattices. These are useful in understanding the relationships available in the lattice. The attribute exploration feature allows the modification or addition of rules or im-

plications. The user interface is an old-fashioned text style, which took a little getting used to. A document provided with ConImp describing FCA and how to use the various features of this tool and examples made it easy to learn.

The program used to display ConImp contexts from `CXT` files is Diagram. It is an old-fashioned DOS program, but not only does it work, it is also one of the very few tools which allows user manipulation of the resulting lattice. As the number of concept nodes becomes larger, the initial layout of the lattice becomes harder to read and requires this manual rearrangement. The lattice displays can only be printed to a local printer. Due to the control you have in rearranging the displayed lattice, Diagram did prove to be useful.

4.3.4 Concepts/Graphplace

Concepts [8] is a C program that takes a text context description CON file containing the objects, attributes, and relations and generates a lattice that is written to a text file in a readable form. This serves as input to the Graphplace [8] program which generates a postscript file for displaying and printing the lattice. Source code is provided for both allowing for the future potential of modifications and enhancements. Graphplace creates a hierarchical lattice layout which was found to be very useful for analyzing the generated lattices. As the layouts become more complex, there are more line crossings. Layout improvements would be desireable, but this pair of tools provides a good first cut of a hierarchical lattice diagram.

4.3.5 Concept Explorer/ToscanaJ

Concept Explorer [9] and ToscanaJ [43] are the newest generation of tools written in Java. They are rewrites and improvements based on Anaconda, Toscana, and ConImp described previously. They are both currently under development. An early version of ToscanaJ is available that takes an XML description of a lattice as input, offering lattice display capabilities. Concept Explorer will provide data preprocessing capabilities as well as implication generation, association rule generation, and attribute exploration. A running version of Concept Explorer was not available during the course of this project, but there is information and screen shots on their Web page. These tools are worth watching for the future.

4.4 Data Analysis

We now discuss our analysis of the data in the project database. For these examples and their analysis we chose to use ConImp/Diagram for its ability to generate implications and ability to rearrange the graph layout. Concepts/Graphplace were also used because of the hierarchical graph layout which lent itself well to analysis based on the hierarchical structure of the lattice. Perl programs were generated to convert data from SQL queries to the required input formats.

4.4.1 Considered Relations

As mentioned above, in its current form, FCA requires the identification of a context, which is:

- A binary relation, involving only two fields; and

- A Boolean relation, where the cells contain only 0/1 values, for example as the result of scaling.

Therefore our first task is to determine the binary relations to be considered. In the following sections, we begin by analyzing the following binary join tables represented explicitly in the schema (Fig. 6):

- People × Events
- People × Groups
- People × Expertise

In addition, we have constructed the following relations, including some unioned relations, special-purpose for this investigation:

- Groups × Events
- People × (Events ∪ Groups)
- Expertise × Groups
- People × (Groups ∪ Expertise)
- People × (Events ∪ Expertise)
- People × (Events ∪ Groups ∪ Expertise)

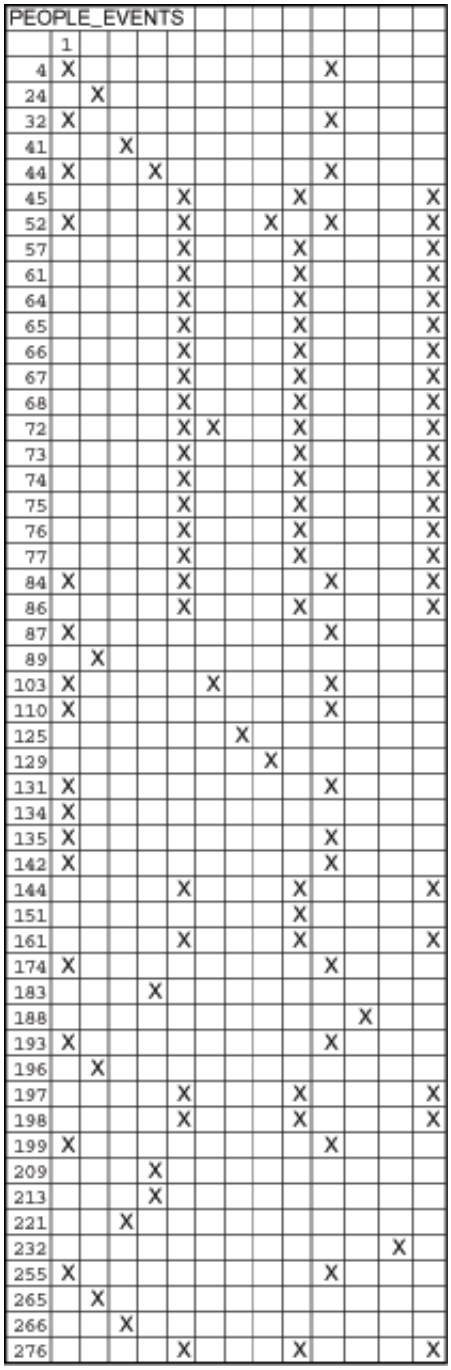
Below, we analyze these relations. In addition to the analytical discussion, in most instances, the tables are provided, and both `CONIMP` and `GraphPlace` diagrams, along with implications.

4.4.2 People × Events

People × Events addresses the question of "How are events related through people?" The table is shown in Fig. 14, two versions of the lattices in Figs. 15–16, and the generated implications in Fig. 17.

The implications have been translated, grouped, and interpreted. A text description above each group identifies the related events (from the ID's) and some indication of the person or people that are involved in the relation. Grouping of the implications is determined by being an implication from the same concept node or involving overlapping subsets of attributes as in first group. Knowledge about the information helps to determine if grouping implications from different concept nodes makes sense.

You can identify the implications from Fig. 17 in the lattice display in Figs. 15–16. For example, the first group of implications corresponds to the 3 concept nodes circled on the right in Fig. 16. One can see that there are events that are not related at all, such as WTC93(3), Philippines(2), New Delhi(12), Paris(11), and Khobar(7) in Fig. 15. Note that events are typically related through an individual, such as Kenya, Tanzania, and 9/11 related through Ayman Mohammed Rabie al-Zawahiri(52).



PEOPLE_EVENTS													
	1												
4	X									X			
24		X											
32	X									X			
41			X										
44	X			X						X			
45					X				X				X
52	X				X			X		X			X
57					X				X				X
61					X				X				X
64					X				X				X
65					X				X				X
66					X				X				X
67					X				X				X
68					X				X				X
72					X	X			X				X
73					X				X				X
74					X				X				X
75					X				X				X
76					X				X				X
77					X				X				X
84	X				X					X			X
86					X				X				X
87	X									X			
89		X											
103	X					X				X			
110	X									X			
125							X						
129								X					
131	X									X			
134	X												
135	X									X			
142	X									X			
144					X				X				X
151									X				
161					X				X				X
174	X									X			
183				X									
188											X		
193	X									X			
196		X											
197					X				X				X
198					X				X				X
199	X									X			
209				X									
213				X									
221			X										
232												X	
255	X									X			
265		X											
266			X										
276					X				X				X

Figure 14: Person × Events table.

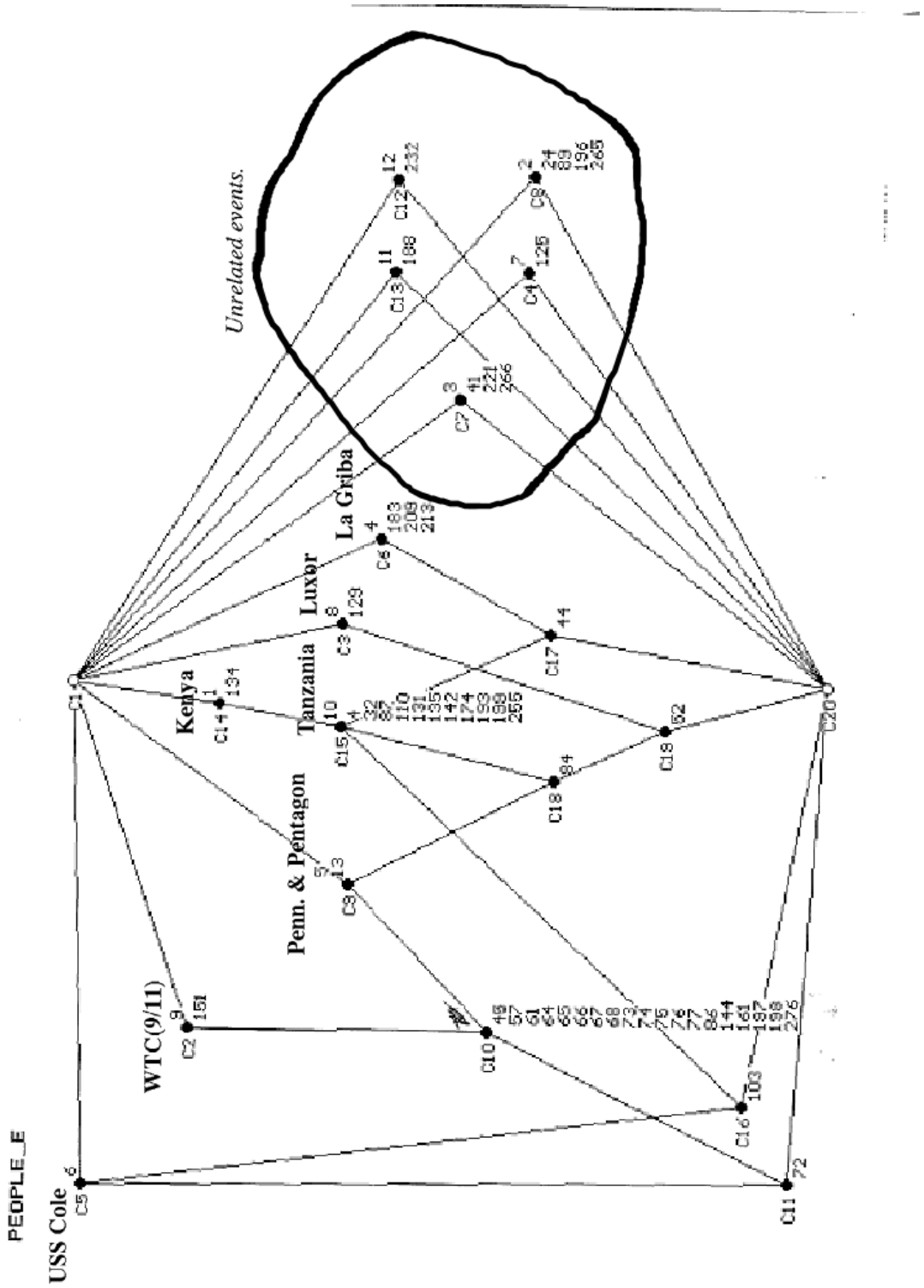


Figure 15: People × Events lattice: `CONIMP`.

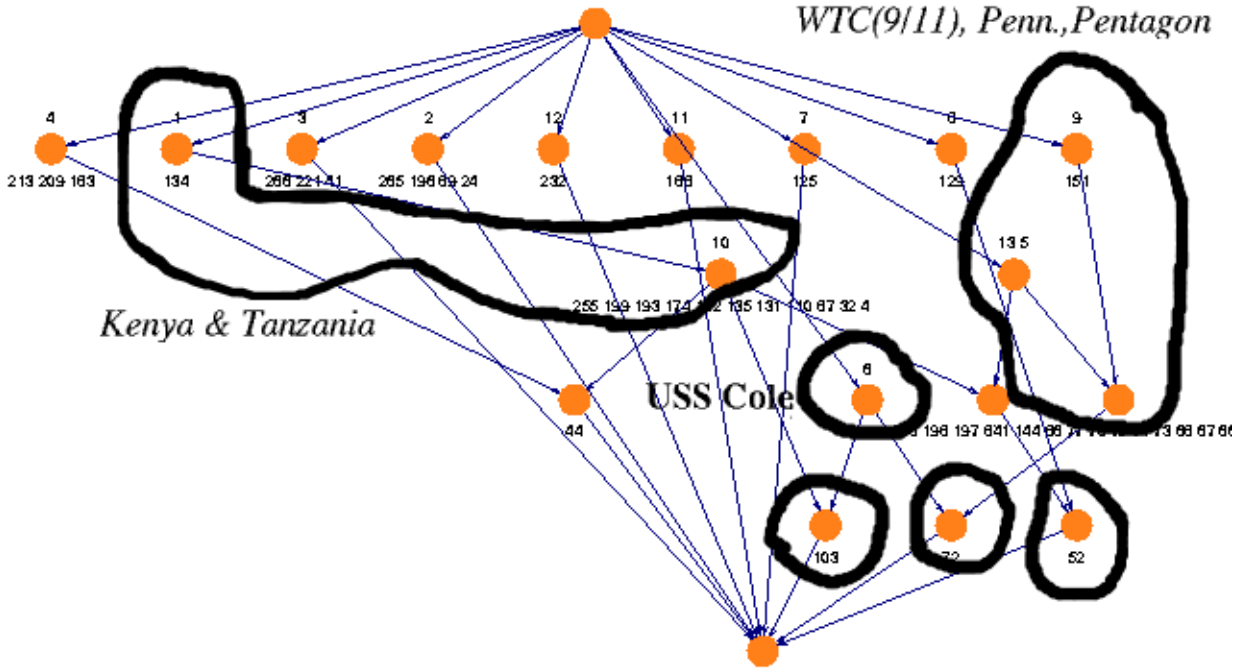


Figure 16: People × Events lattice: `GraphPlace`.

```
Print of a list of implications with independent premise of the context
PEOPLE_EVENTS  in extended format;  (concept no.)
<number of objects non-trivially satisfying the implication> :

9/11, Penn., Pentagon related (9/11) (20 people)
   1 (  9) < 22> :        13   ==>          5
   2 (  9) < 22> :         5   ==>         13
   3 ( 10) < 20> :         5         9   ==>         13
   4 ( 10) < 20> :        13         9   ==>          5

USS Cole related to 9/11 (Khalid Almihdhar)
   5 ( 11) <  1> :        13         6   ==>          5         9
   6 ( 11) <  1> :         6         9   ==>         13         5
   7 ( 11) <  1> :         5         6   ==>         13         9

Kenya and Tanzania related (21 people)
   8 ( 15) < 15> :        10   ==>          1

USS Cole, Kenya, and Tanzania related (Osama Bin Laden)
   9 ( 16) <  1> :         1         6   ==>         10
  10 ( 16) <  1> :        10         6   ==>          1

La Griba, Kenya, and Tanzania related (Khalid al-Shanqiti)
  11 ( 17) <  1> :        10         4   ==>          1
  12 ( 17) <  1> :         1         4   ==>         10

Kenya, Tanzania, and 9/11 related (Muhammad Atef)
  13 ( 18) <  2> :        10         5   ==>          1        13
  14 ( 18) <  2> :        10        13   ==>          1         5
  15 ( 18) <  2> :         1         5   ==>         10        13
  16 ( 18) <  2> :         1        13   ==>         10         5

Luxor, Kenya, Tanzania, and 9/11 related (Ayman Mohammed Rabie al-Zawahiri)
  17 ( 19) <  1> :         1         8   ==>         10        13         5
  18 ( 19) <  1> :        13         8   ==>          1        10         5
  19 ( 19) <  1> :        10         8   ==>          1        13         5
  20 ( 19) <  1> :         5         8   ==>          1        10        13
```

Figure 17: People × Events: Implications

4.4.3 People × Groups

People × Groups addresses the question of "How are groups related through people?" The table is shown in Fig. 18, two versions of the lattices in Figs. 19–20, and the generated implications in Fig. 21.

The implications and lattice suggest an internal hierarchy of subgroups in Al-Qaeda from the first group of implications and the graphical structure of groups under Al-Qaeda. The groups, Abu Sayyaf(4), Moro(26), and Jaamat al(16) are not related in this view (see Fig. 19), where they are in the Groups × Events view (see Sec. 4.4.4). Once again we see single individuals as members of multiple groups tying groups together, such as Mohammed Atta's(86) membership in al Jehad(9), Egyptian Islamic(3), and Al-Qaeda(1) (see Fig. 20).

4.4.4 Groups × Events

Groups × Events addresses the question of "How are events related through groups?" The table is shown in Fig. 22, two versions of the lattices in Figs. 23–24, and the generated implications in Fig. 25.

The implications and lattice show that all events are related through Al-Qaeda. For example the Shura Council was involved in Penn., Pentagon, Kenya, Tanzania, and Luxor. Luxor was not related in the People × Events view. This is an example where looking at the problem a little differently provides different results. The regular structure of the groups in Fig. 24 could be interpreted as suggesting a hierarchy of subgroups within Al-Qaeda.

4.4.5 People × (Events ∪ Groups)

People × (Events ∪ Groups) addresses the question of "How are events and/or groups related through people?" The table is shown in Fig. 26, two versions of the lattices in Figs. 27–28, and the generated implications in Fig. 29.

The implications and lattice show relationships between groups and events defined by people. Al-Qaeda(g1) is shown related to events, WTC(9/11)(e9), Luxor(e8), Khobar(e7), USS Cole(e6), and La Griba(e4) in the first group of implications and in the structure under Al-Qaeda(g1) in the lattice. The Philippines(e2) event in the lattice shows involvement of people from four distinct groups, Jaamat al(g16), Moro(g26), Abu Sayyaf(g4), and Jemaah Islamiah(g6). Common Shura Council(g7) people are shown involved in Kenya(e1), Tanzania(e10), Penn.(e5), and Pentagon(e13). Interestingly, the Shura Council(g7) is not tied to WTC(9/11)(e9). We also see that people from multiple groups were involved in the 9/11 events (e5, e9, e13), Al-Qaeda(g1), Egyptian Islamic(g3), Shura Council(g7), Islamic Army(g8), al Jehad(g9), and Al-Gama'a(g11).

Note also a common node (C31) without *any* people, groups, or events directly related to it. Rather, it's a kind of "virtual" concept, indirectly connecting Shura, Penn., and Pentagon through Ayman Mohammed Rabie al-Zawahiri (p52) and Muhammed Atef (p84).

PEOPLE_GROUPSS											
	1										
4	X				X						
24										X	
32	X				X				X		
41								X			
44	X										
45	X										
52	X	X			X			X			
57	X										
61	X										
64	X										
65	X										
66	X										
67	X										
68	X										
72	X					X					
73	X										
74	X										
75	X										
76	X										
77	X										
84	X				X		X				
86	X	X					X				
87	X										
89	X			X							
103	X										
110	X										
125	X										
129	X										
131	X										
134	X										
135	X										
142	X										
144	X										
151	X										
161	X										
174	X										
183	X										
188	X										
193	X										
196											X
197	X										
198	X										
199	X										
209	X										
213	X										
221	X										
232	X										
255	X										
265			X								
266	X										
276	X										

Figure 18: Person × Groups table.

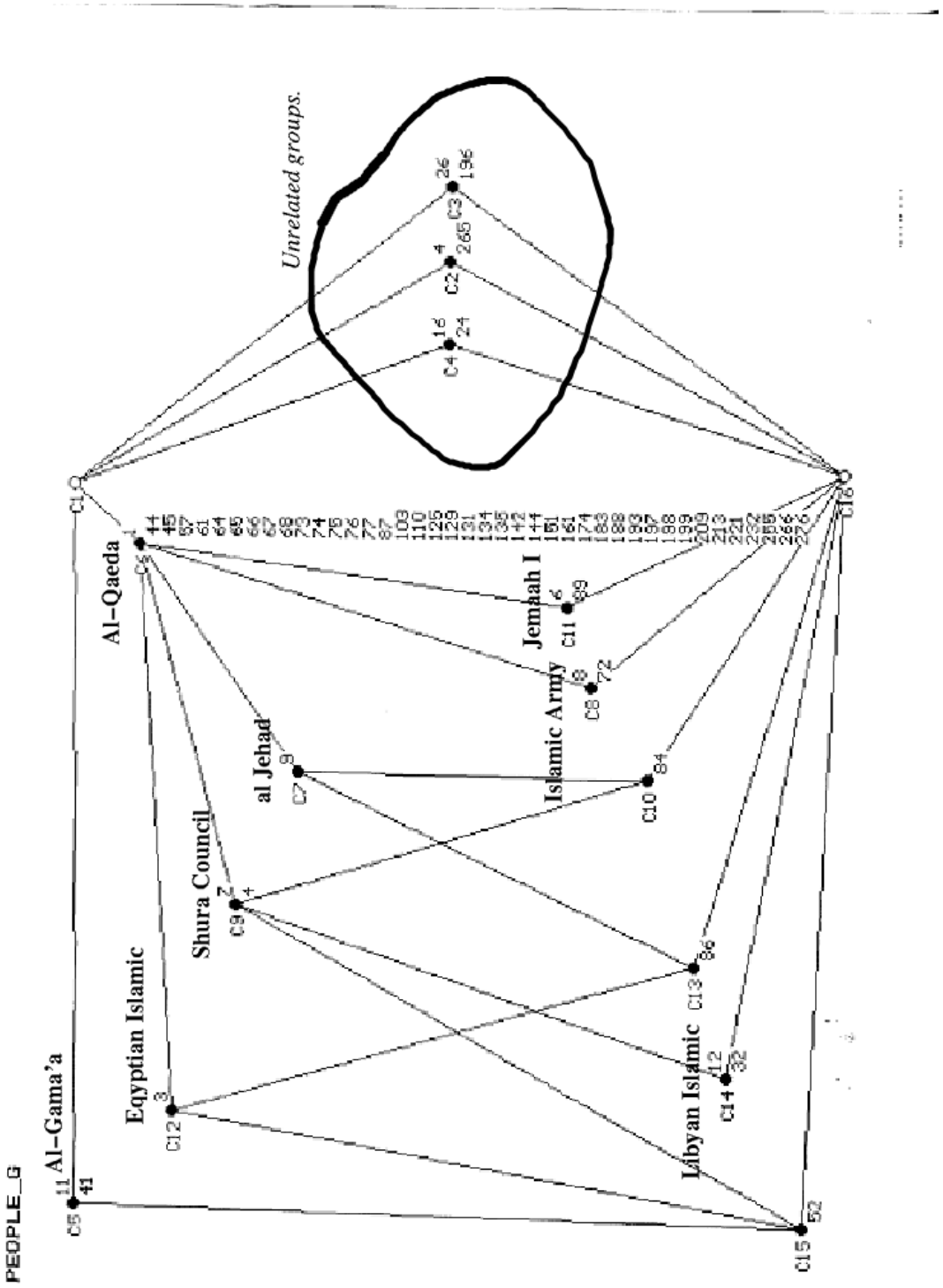


Figure 19: People × Groups lattice: `CONIMP`.

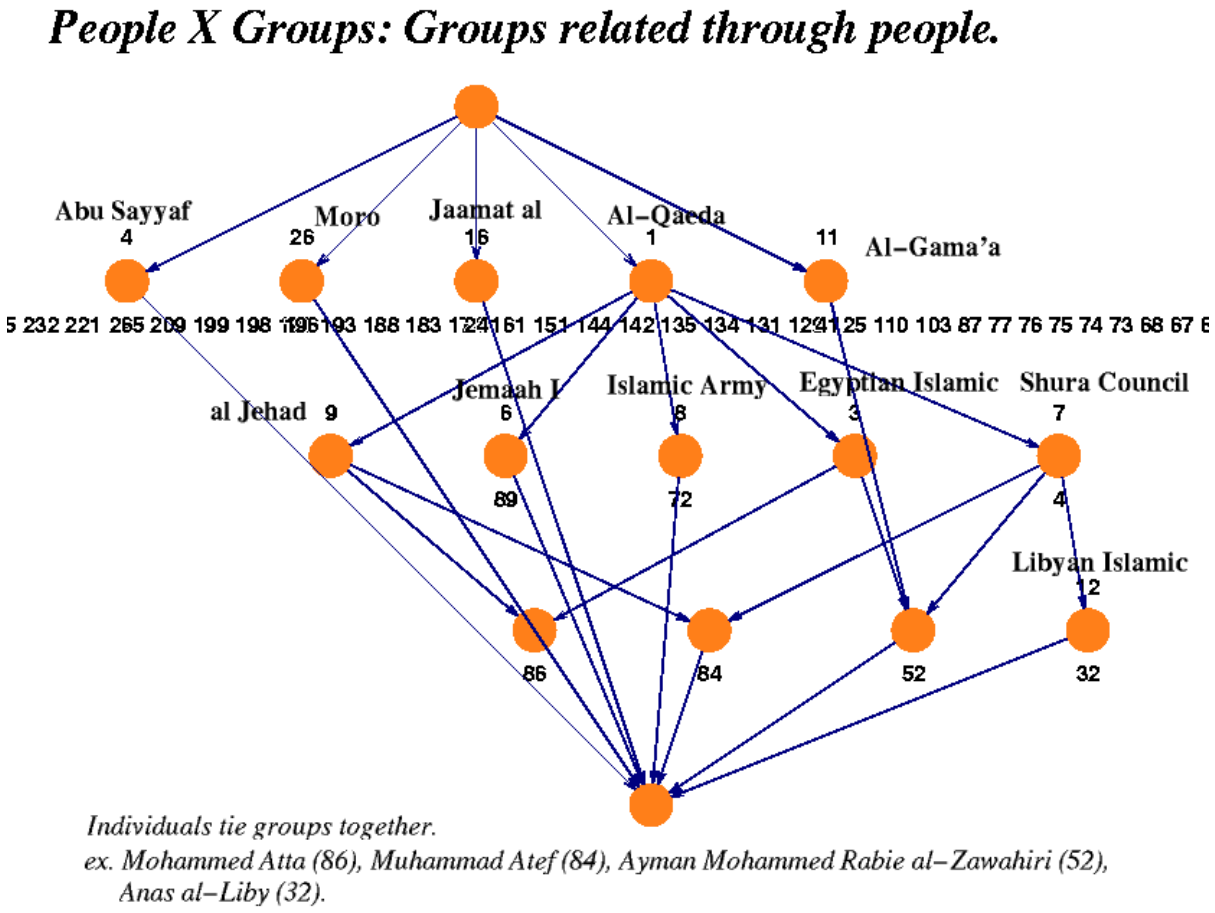


Figure 20: People × Groups lattice: `GraphPlace`.

```
Print of a list of implications with independent premise of the context
PEOPLE_GROUPS  in extended format;  (concept no.)
<number of objects non-trivially satisfying the implication> :

al Jehad, Islamic Army, Shura Council, Jemaah Islamiah, Egyption Islamic
Jihad related to Al-Qaeda (Khalid Almihdhar, Abdullah Ahmed Abdullah,
Muhammad Atef, Faiz Abu Bakar Bafana, Mohammed Atta)
   1 (  7) <  2> :         9   ==>          1
   2 (  8) <  1> :         8   ==>          1
   3 (  9) <  4> :         7   ==>          1
   4 ( 10) <  1> :         7         9   ==>          1
   5 ( 11) <  1> :         6   ==>          1
   6 ( 12) <  2> :         3   ==>          1
   7 ( 13) <  1> :         3         9   ==>          1

Libyan Islamic Fighting Group related to Al-Qaeda and Shura Council
(Anas al-Liby)
   8 ( 14) <  1> :        12   ==>          1         7

Egyption Islamic Jihad, Shura Council, Al-Gama's al-Islamiyya, and Al-Qaeda
related (Ayman Mohammed Rabie al-Zawahiri)
   9 ( 15) <  1> :         3         7   ==>          1        11
  10 ( 15) <  1> :        11         7   ==>          1         3
  11 ( 15) <  1> :         1        11   ==>          3         7
  12 ( 15) <  1> :        11         3   ==>          1         7
```

Figure 21: People × Groups: Implications

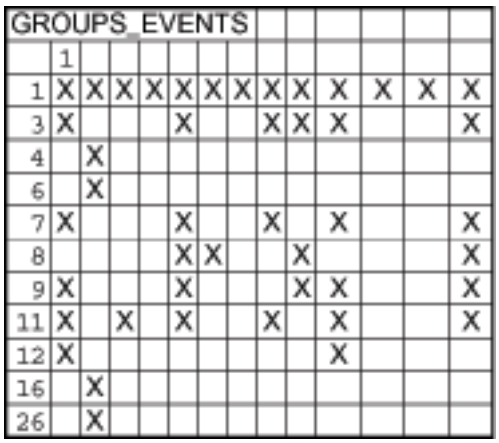


Figure 22: Groups × Events table.

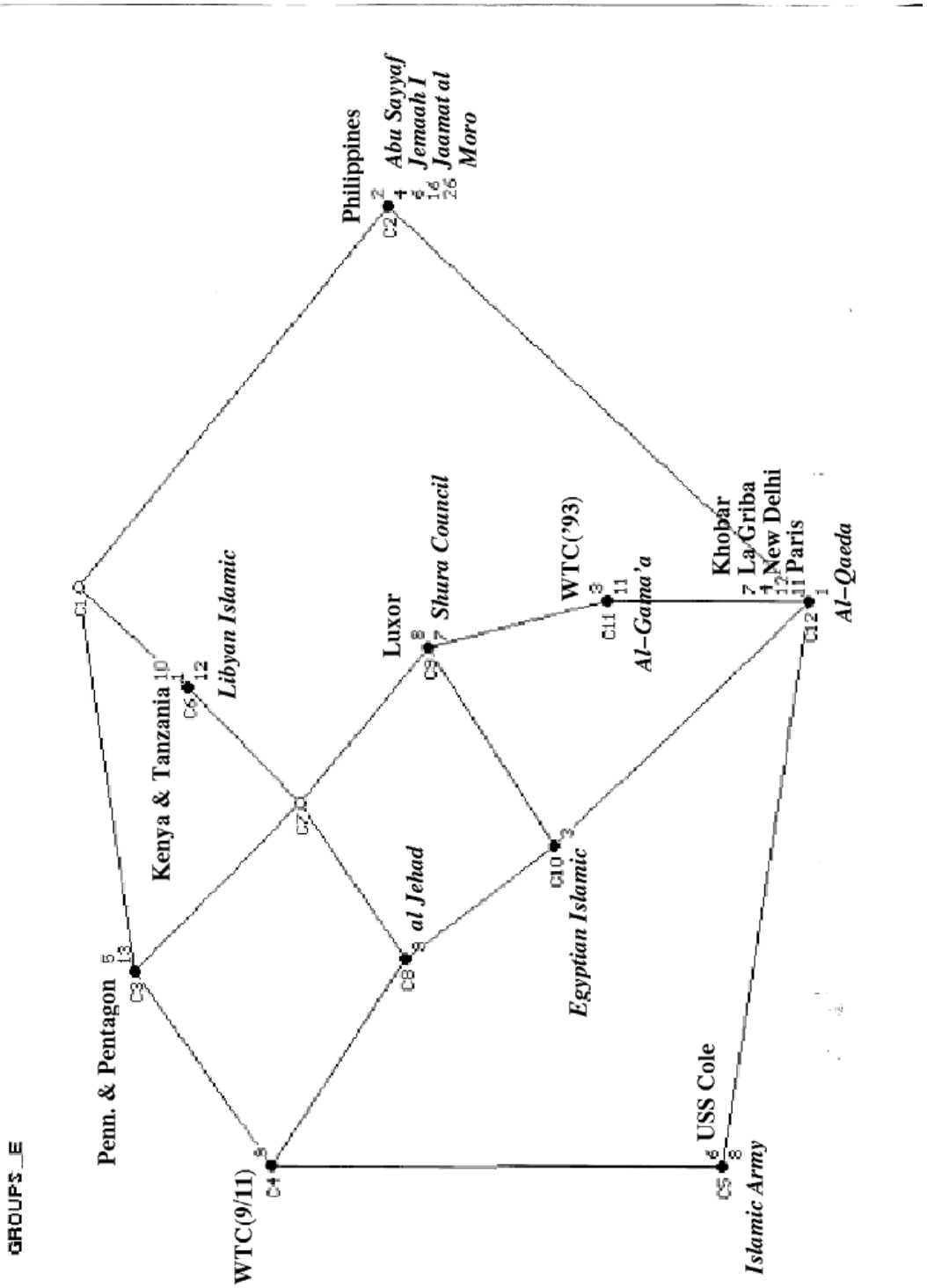


Figure 23: Groups × Events lattice: `CONIMP`.

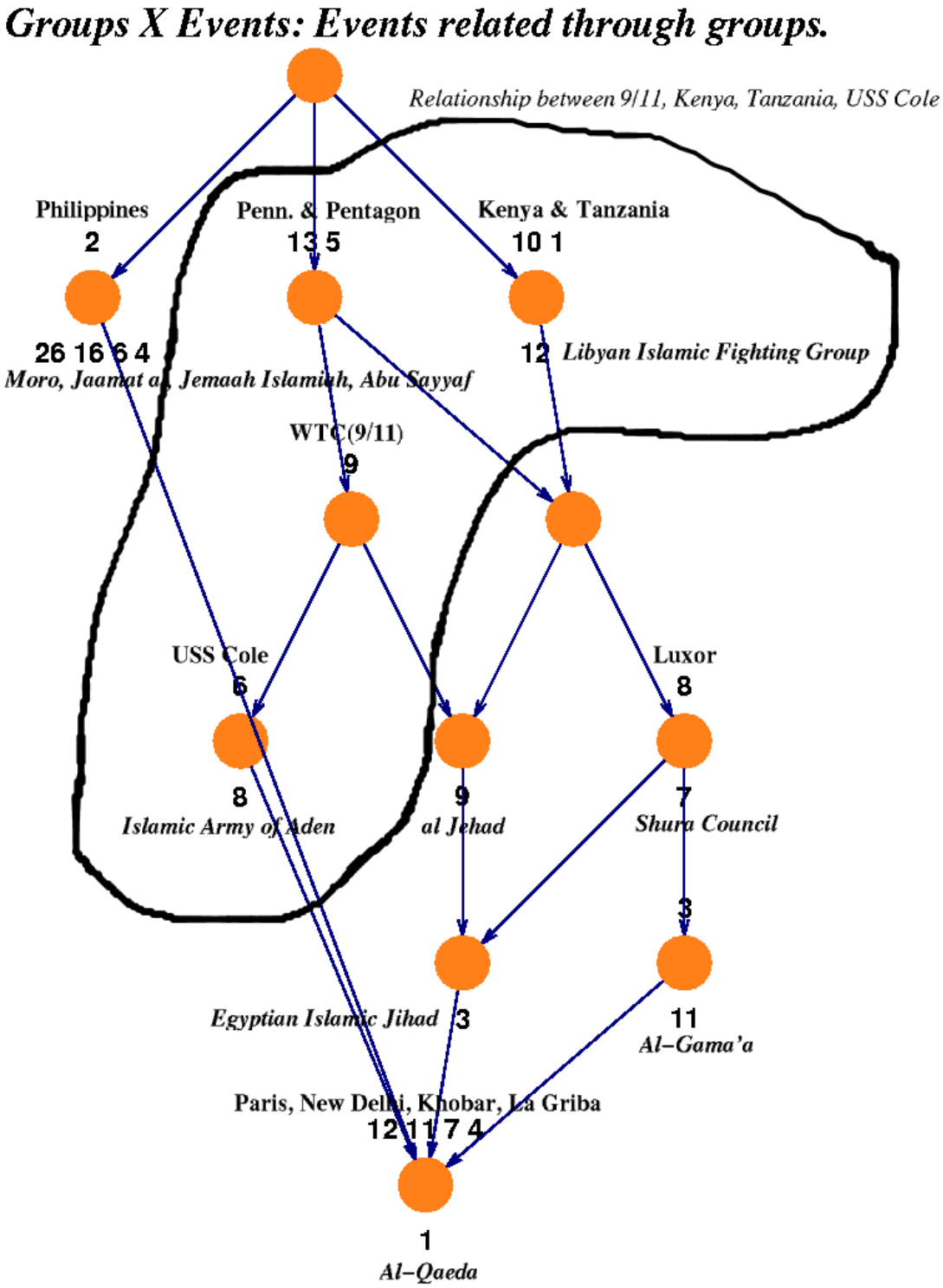


Figure 24: Groups × Events lattice: `GraphPlace`.

```
Print of a list of implications with independent premise of the context
GROUPS_EVENTS  in extended format;  (concept no.)
<number of objects non-trivially satisfying the implication> :

Penn., Pentagon, and 9/11 related (9/11)
   1 (  3) <  6> :        13   ==>          5
   2 (  3) <  6> :         5   ==>         13
   3 (  4) <  4> :         9   ==>         13         5

USS Cole related to 9/11 (Islamic Army of Aden)
   4 (  5) <  2> :         6   ==>         13         5         9

Kenya and Tanzania related (Libyan Islamic Fighting Group)
   5 (  6) <  6> :        10   ==>          1
   6 (  6) <  6> :         1   ==>         10

Kenya, Tanzania, and 9/11 related (al Jehad)
   7 (  7) <  5> :         1         5   ==>         10        13
   8 (  7) <  5> :        10         5   ==>          1        13
   9 (  7) <  5> :         1        13   ==>         10         5
  10 (  7) <  5> :        10        13   ==>          1         5
  11 (  8) <  3> :         1         9   ==>         10        13         5
  12 (  8) <  3> :        10         9   ==>          1        13         5

Luxor, Kenya, Tanzania, and 9/11 related (Egyptian Islamic Jihad,
Shura Council of al-Qaeda)
  13 (  9) <  4> :         8   ==>          1        10        13         5
  14 ( 10) <  2> :         8         9   ==>          1        10        13    5

World Trade Center, Luxor, Kenya, Tanzania, and 9/11 related (Al-Gama'a al-Islamiyya)
  15 ( 11) <  2> :         3   ==>          1        10        13         5   8

USS Cole, World Trade Center, Luxor, Kenya, Tanzania, 9/11, Paris, New Delhi,
Philippine, La Griba, Khobar (Al-Qaeda)
  16 ( 12) <  1> :         6         8   ==>1   10   11  12   13    2    3    4    5
  7    9
  17 ( 12) <  1> :   12   ==>1   10   11   13    2    3    4    5    6    7   8    9
  18 ( 12) <  1> :   11   ==>1   10   12   13   2    3    4    5    6    7   8    9
  19 ( 12) <  1> :    4   ==>1   10   11   12  13    2    3    5    6    7   8    9
  20 ( 12) <  1> :    3    9   ==>1   10   11  12   13    2    4    5    6   7    8
  21 ( 12) <  1> :    1    2   ==>    10   11   12  13    3    4    5    6    7   8    9
  22 ( 12) <  1> :    3    6   ==>1   10   11  12   13    2    4    5    7   8    9
  23 ( 12) <  1> :   10    6   ==>1   11   12  13    2    3    4    5    7   8    9
  24 ( 12) <  1> :    2    9   ==>1   10   11  12   13    3    4    5    6   7    8
  25 ( 12) <  1> :    2    8   ==>1   10   11  12   13    3    4    5    6   7    9
  26 ( 12) <  1> :    1    6   ==>    10   11   12  13    2    3    4    5    7   8    9
  27 ( 12) <  1> :    2    6   ==>1   10   11  12   13    3    4    5    7   8    9
  28 ( 12) <  1> :    7   ==>1   10   11   12  13    2    3    4    5    6   8    9
  29 ( 12) <  1> :    2    5   ==>1   10   11  12   13    3    4    6    7   8    9
  30 ( 12) <  1> :   10    2   ==>1   11   12  13    3    4    5    6    7   8    9
  31 ( 12) <  1> :    2    3   ==>1   10   11  12   13    4    5    6    7   8    9
  32 ( 12) <  1> :   13    2   ==>1   10   11  12    3    4    5    6    7   8    9
```

Figure 25: Groups × Events: Implications

PEOPLE_EVENTS_GROUPS																								
	g1	g3	g4	g6	g7	g8	g9	g11	g12	g16	g26	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	e13
4	X				X							X									X			
24										X			X											
32	X				X				X			X									X			
41								X						X										
44	X											X			X						X			
45	X															X				X				X
52	X	X			X			X				X				X			X		X			X
57	X															X				X				X
61	X															X				X				X
64	X															X				X				X
65	X															X				X				X
66	X															X				X				X
67	X															X				X				X
68	X															X				X				X
72	X					X										X	X			X				X
73	X															X				X				X
74	X															X				X				X
75	X															X				X				X
76	X															X				X				X
77	X															X				X				X
84	X				X		X					X				X					X			X
86	X	X					X									X				X				X
87	X											X									X			
89	X			X									X											
103	X											X					X				X			
110	X											X									X			
125	X																	X						
129	X																		X					
131	X											X									X			
134	X											X												
135	X											X									X			
142	X											X									X			
144	X															X				X				X
151	X																			X				
161	X															X				X				X
174	X											X									X			
183	X														X									
188	X																					X		
193	X											X									X			
196											X		X											
197	X															X				X				X
198	X															X				X				X
199	X											X									X			
209	X														X									
213	X														X									
221	X													X										
232	X																						X	
255	X											X									X			
265			X										X											
266	X													X										
276	X															X				X				X

Figure 26: People × (Events ∪ Groups) table.

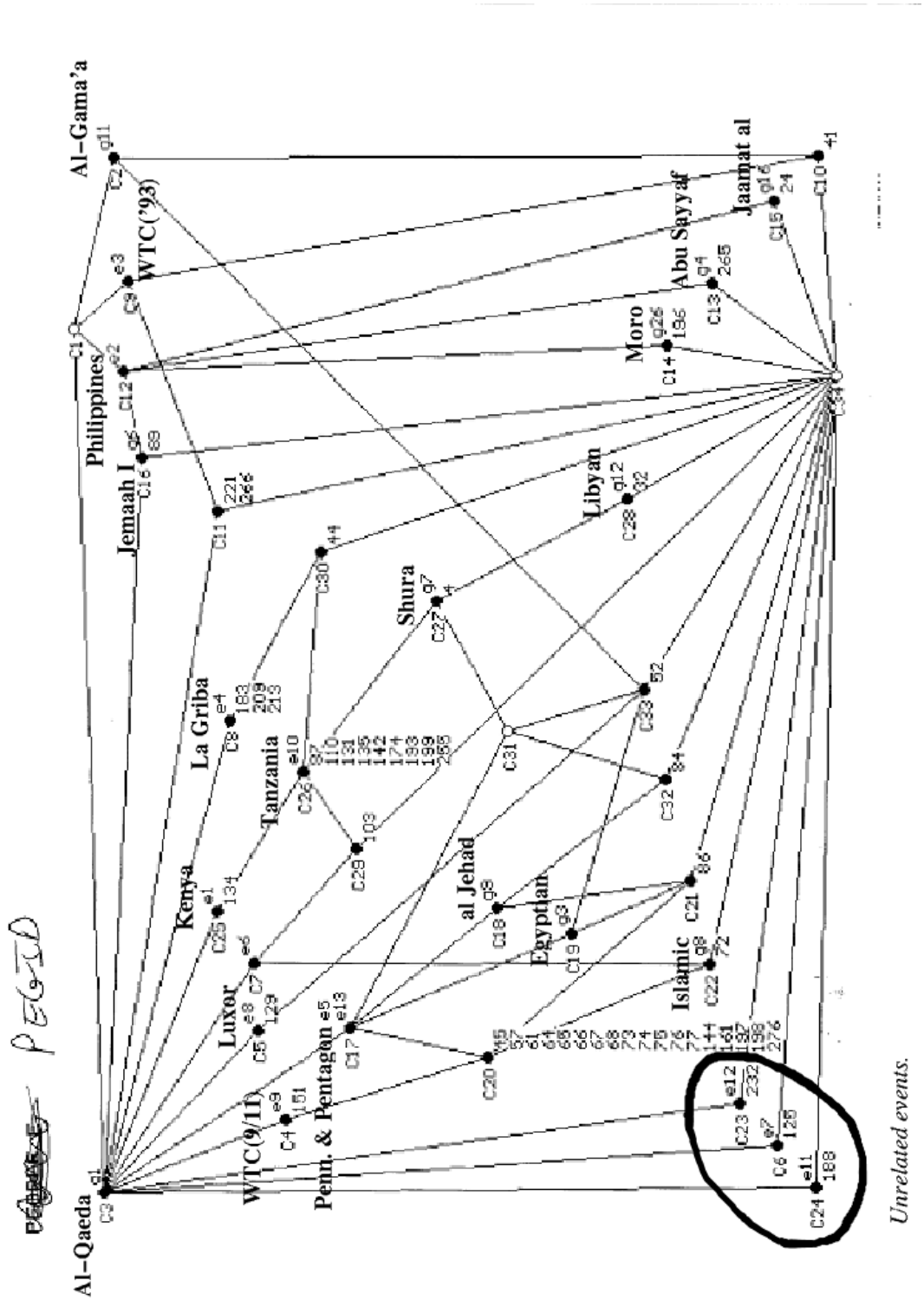


Figure 27: People × (Events ∪ Groups) lattice: `CONIMP`.

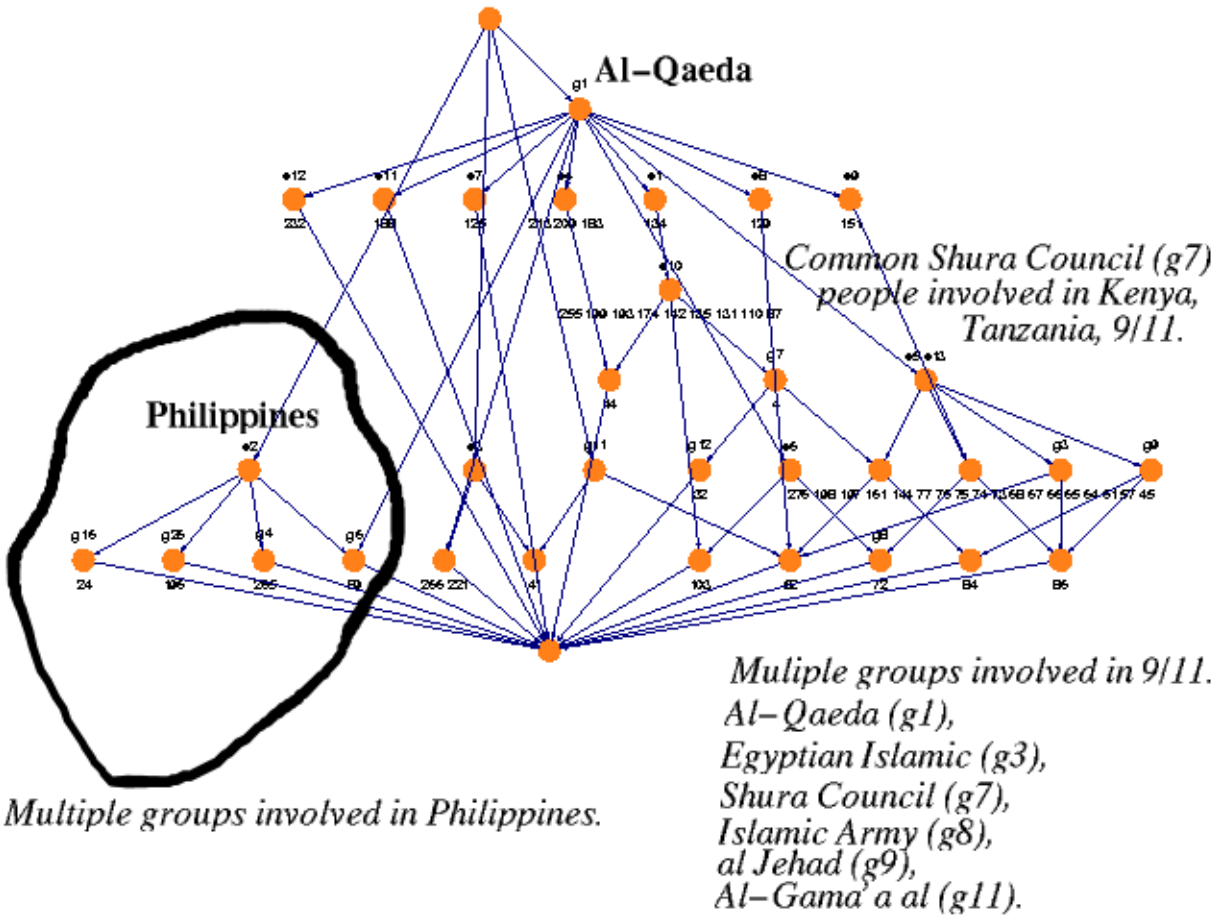


Figure 28: People × (Events ∪ Groups) lattice: `GraphPlace`.

```
9/11, Luxor, Khobar, USS Cole, LaGriba related to Al-Qaeda
(Abdullah Ahmed Abdullah, Abdelkader Mahmoud Es Sayed, Ali Saed Bin Ali El-Hoorie,
Christian Manfred G., Nizar Ben Mohammed Nawa, Mouhamedou Ould Slehi)
   1 (  4) < 21> :        e9   ==>         g1
   2 (  5) <  2> :        e8   ==>         g1
   3 (  6) <  1> :        e7   ==>         g1
   4 (  7) <  2> :        e6   ==>         g1
   5 (  8) <  4> :        e4   ==>         g1

Abu Sayyaf, Moro Islamic Liberation Front, Jaamat al-Islamie
related to Philippine Embassy Bombing (Ustadz Nur Mohammed Umog,
Yunus Moklis, Fathur Rohman al-Ghozi)
   6 ( 13) <  1> :        g4   ==>         e2
   7 ( 14) <  1> :       g26   ==>         e2
   8 ( 15) <  1> :       g16   ==>         e2

Al-Qaeda, Jemaah, and Philippine Embassy Bombing related (Faiz Abu Bakar Bafana)
   9 ( 16) <  1> :        e2        g1   ==>         g6
  10 ( 16) <  1> :        g6   ==>         e2        g1

Al-Qaeda, Egyptian Islamic Jihad, al Jehad related to 9/11
(19 people, Mohammed Atta)
  11 ( 17) < 22> :  e13   ==>    e5   g1
  12 ( 17) < 22> :   e5   ==>   e13   g1
  13 ( 18) <  2> :   g9   ==>   e13   e5   g1
  14 ( 19) <  2> :   g3   ==>   e13   e5   g1
  15 ( 20) < 20> :   e5   e9   ==>   e13   g1
  16 ( 20) < 20> :  e13   e9   ==>    e5   g1
  17 ( 21) <  1> :   g3   g9   ==>   e13   e5   e9  g1
  18 ( 21) <  1> :   e9   g9   ==>   e13   e5   g1  g3
  19 ( 21) <  1> :   e9   g3   ==>   e13   e5   g1  g9

Al-Qaeda and Islamic Army of Aden related to 9/11 and USS Cole
(Khalid Almihdhar)
  20 ( 22) <  1> :   e5   e6   ==>   e13   e9   g1  g8
  21 ( 22) <  1> :   e6   e9   ==>   e13   e5   g1  g8
  22 ( 22) <  1> :  e13   e6   ==>    e5   e9   g1  g8
  23 ( 22) <  1> :   g8   ==>   e13   e5   e6   e9  g1

Paris, New Delhi related to Al-Qaeda (Abdul Rehman Safani, Amine Mezbar)
  24 ( 23) <  1> :  e12   ==>    g1
  25 ( 24) <  1> :  e11   ==>    g1

Kenya and Tanzania related to Al-Qaeda (Khalid al Fawwaz, 9 people)
  26 ( 25) < 16> :   e1   ==>    g1
  27 ( 26) < 15> :  e10   ==>    e1   g1

Kenya and Tanzania related Al-Qaeda, Shura Council, Libyan Islamic Fighting Group
(Abdullah Ahmed Abdullah, Anas al-Liby)
  28 ( 27) <  4> :   g7   ==>    e1  e10   g1
  29 ( 28) <  1> :  g12   ==>    e1  e10   g1   g7

Kenya, Tanzania, and USS Cole related to Al-Qaeda (Osama Bin Laden)
  30 ( 29) <  1> :   e1   e6   ==>   e10   g1
  31 ( 29) <  1> :  e10   e6   ==>    e1   g1

La Griba, Kenya, and Tanzania related to Al-Qaeda (Khalid al-Shanqiti)
  32 ( 30) <  1> :   e1   e4   ==>   e10   g1
  33 ( 30) <  1> :  e10   e4   ==>    e1   g1

Kenya, Tanzania, 9/11 related to Al-Qaeda and Shura Council
 34 ( 31) <  2> :   e1  e13   ==>   e10   e5   g1  g7
  35 ( 31) <  2> :  e10  e13   ==>    e1   e5   g1 g7
  36 ( 31) <  2> :  e13   g7   ==>    e1  e10   e5 g1
  37 ( 31) <  2> :   e5   g7   ==>    e1  e10  e13 g1
  38 ( 31) <  2> :   e1   e5   ==>   e10  e13   g1 g7
  39 ( 31) <  2> :  e10   e5   ==>    e1  e13   g1 g7

Kenya, Tanzania, 9/11 related to Al-Qaeda, Shura Council, and al Jahad
(Muhammad Atef)
  40 ( 32) <  1> :   e1   g9   ==>   e10  e13   e5  g1   g7
  41 ( 32) <  1> :  e10   g9   ==>    e1  e13   e5 g1   g7
  42 ( 32) <  1> :   g7   g9   ==>    e1  e10  e13  e5   g1

Kenya, Tanzania, 9/11, Luxor related to Al-Qaeda, Egyptian Islamic Jihad,
Shura Council, Al-Gama's al-Islamiyya (Ayman Mohammed Rabie al-Zawahiri)
  43 ( 33) <  1> :   e8   g3   ==>    e1  e10  e13  e5   g1  g11   g7
  44 ( 33) <  1> :   g1  g11   ==>    e1  e10  e13  e5   e8   g3   g7
  45 ( 33) <  1> :   e5   e8   ==>    e1  e10  e13  g1  g11   g3   g7
  46 ( 33) <  1> :  e10   g3   ==>    e1  e13   e5  e8   g1  g11   g7
  47 ( 33) <  1> :  e13  g11   ==>    e1  e10   e5  e8   g1   g3   g7
  48 ( 33) <  1> :   e1   e8   ==>   e10  e13   e5  g1  g11   g3   g7
  49 ( 33) <  1> :  e13   e8   ==>    e1  e10   e5  g1  g11   g3   g7
  50 ( 33) <  1> :  e10  g11   ==>    e1  e13   e5  e8   g1   g3   g7
  51 ( 33) <  1> :   e5  g11   ==>    e1  e10  e13  e8   g1   g3   g7
  52 ( 33) <  1> :   e1   g3   ==>   e10  e13   e5  e8   g1  g11   g7
  53 ( 33) <  1> :  e10   e8   ==>    e1  e13   e5  g1  g11   g3   g7
  54 ( 33) <  1> :   e8  g11   ==>    e1  e10  e13  e5   g1   g3   g7
  55 ( 33) <  1> :  g11   g7   ==>    e1  e10  e13  e5   e8   g1   g3
  56 ( 33) <  1> :  g11   g3   ==>    e1  e10  e13  e5   e8   g1   g7
  57 ( 33) <  1> :   e1  g11   ==>   e10  e13   e5  e8   g1   g3   g7
  58 ( 33) <  1> :   g3   g7   ==>    e1  e10  e13  e5   e8   g1  g11
  59 ( 33) <  1> :   e8   g7   ==>    e1  e10  e13  e5   g1  g11   g3
```

Figure 29: People × (Events ∪ Groups): Implications

4.4.6 People × Expertise

First, we would like to understand something about people and their expertise(s). Creating a lattice of People × Expertise (see Fig. 30) shows simple structure. People typically have only one expertise, except for the 3 with leadership (Terrorist Ops(2), Military Strategy(8), Military Advisor(10)) expertise. Most of the people that we have information on are pilots(4).

4.4.7 Expertise × Groups

Is there a relationship between expertise and groups? Creating a lattice of Expertise × Groups (see Fig. 31) hints at some possibliities. All the Pilots(4) are in Islamic Army(8), Egyptian Islamic(3), al Jehad(9), and/or Al-Qaeda(1). The Explosives(1) experts are in Jamaat al(16), Shura Council(7), and/or Al-Qaeda(1). And the Computer(3) experts are in Libyan Islamic(12), Shura Council(7), and/or Al-Qaeda(1). There are only 2 people in the database with Computer expertise and 3 with explosives expertise. What is seen here may be just conincidental or just an artifact of the data.

4.4.8 People's Expertise Within Groups

The Expertise × Groups lattice is further refined by creating a People × (Groups ∪ Expertise) lattice (see Fig. 32). Here we see expertises within groups, such as Explosives(x1), Agriculture(x6), Finance(x5), Computers(x3), Pilots(x4), and Military Strategy(x8) in Al-Qaeda(g1). One can also see an expertise across groups, such as groups Jaamat al(g16), Al-Qaeda(g1), and Shura Council(g7) containing explosives(x1) experts. The number of people and who they are is additionally available in this lattice. Once again, individuals tie together groups and expertise.

4.4.9 People's Expertise Used in Events

Now, "How is expertise distributed through the events?" A lattice of People × (Events ∪ Expertise) is created (see Fig. 33). The Kenya(e1) and Tanzania(e10) events share Computers(x3), Agriculture(x6), Finance(x5), Explosives(x1), and Terrorist Ops(x2) expertise. The 9/11 event (e5,e9,e13) used Piolts(x4), Terrorist Ops(x2), Military Strategy(x8), and Military Advisor(x10) expertise. It appears that there is more expertise information for the Kenya and Tanzania events than for others.

The people (objects) associated with the lowest level of concept nodes can be viewed as being important (perhaps key players) and/or very connected. Osama Bin Laden(103), Muhammad Atef(84), and Ayman Mohammed Rabie al-Zawahiri(52) are shown having Terrorist Ops(x2), Military Strategy(x8), and/or Military Advisor(x10) expertise and are involved in multiple events. Khalid Almihdhar(72) is shown involved in multiple events, USS Cole(e6) and 9/11(e5,e9,e13). The individuals with explosives expertise in the bottom level may be there since the Explosives(x1) expertise is shown to be involved in multiple events, Kenya(e1), Tanzania(e10), and Philippines(e2).

4.4.10 Events, Groups, and Expertise All at Once

Finally, we can look at all the People, Events, Groups, and Expertise information toagether at one time in a People × (Events ∪ Groups ∪ Expertise) lattice (see Fig. 34). This "four-dimensional"

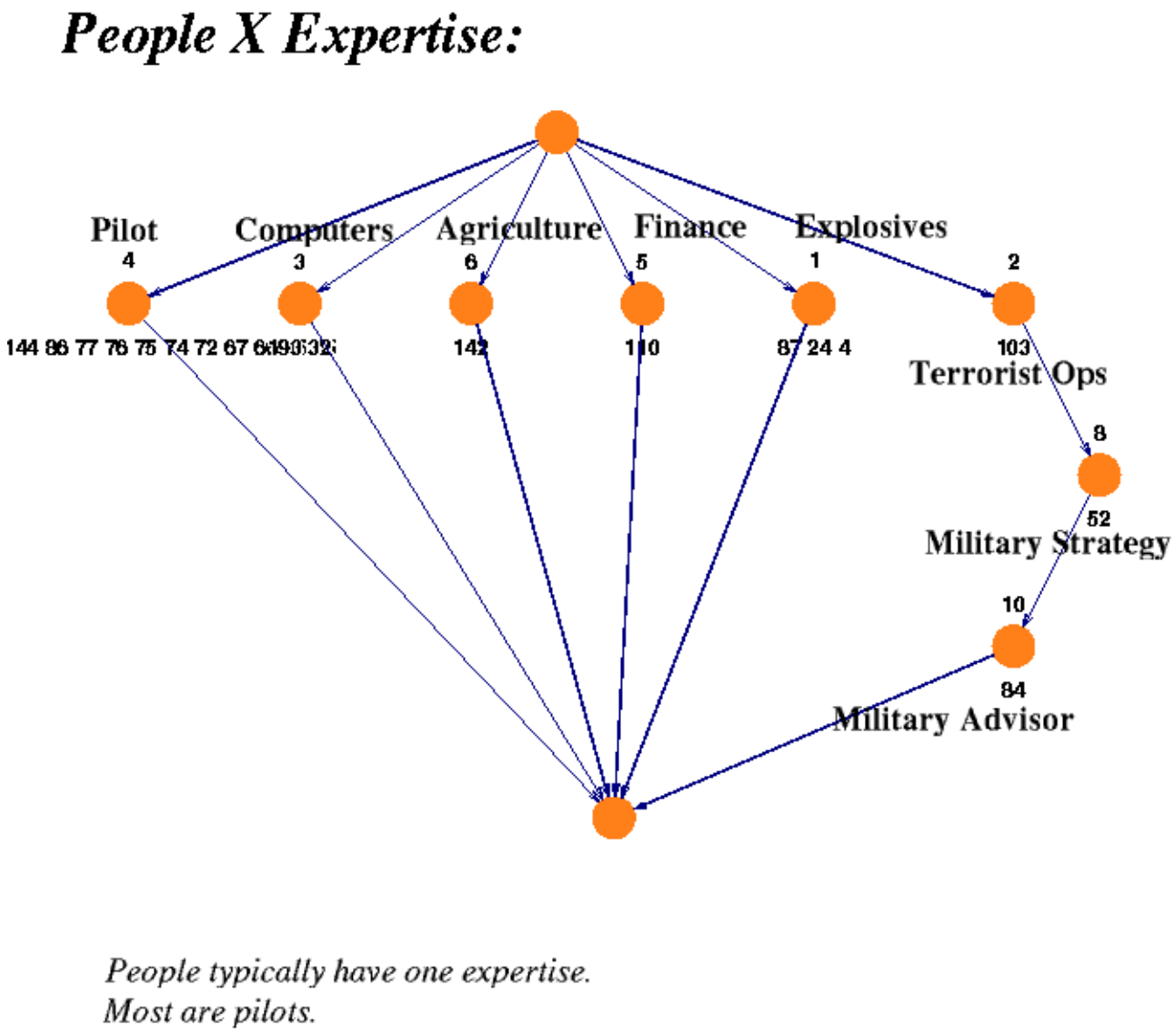


Figure 30: People × Expertise lattice: `GraphPlace`.

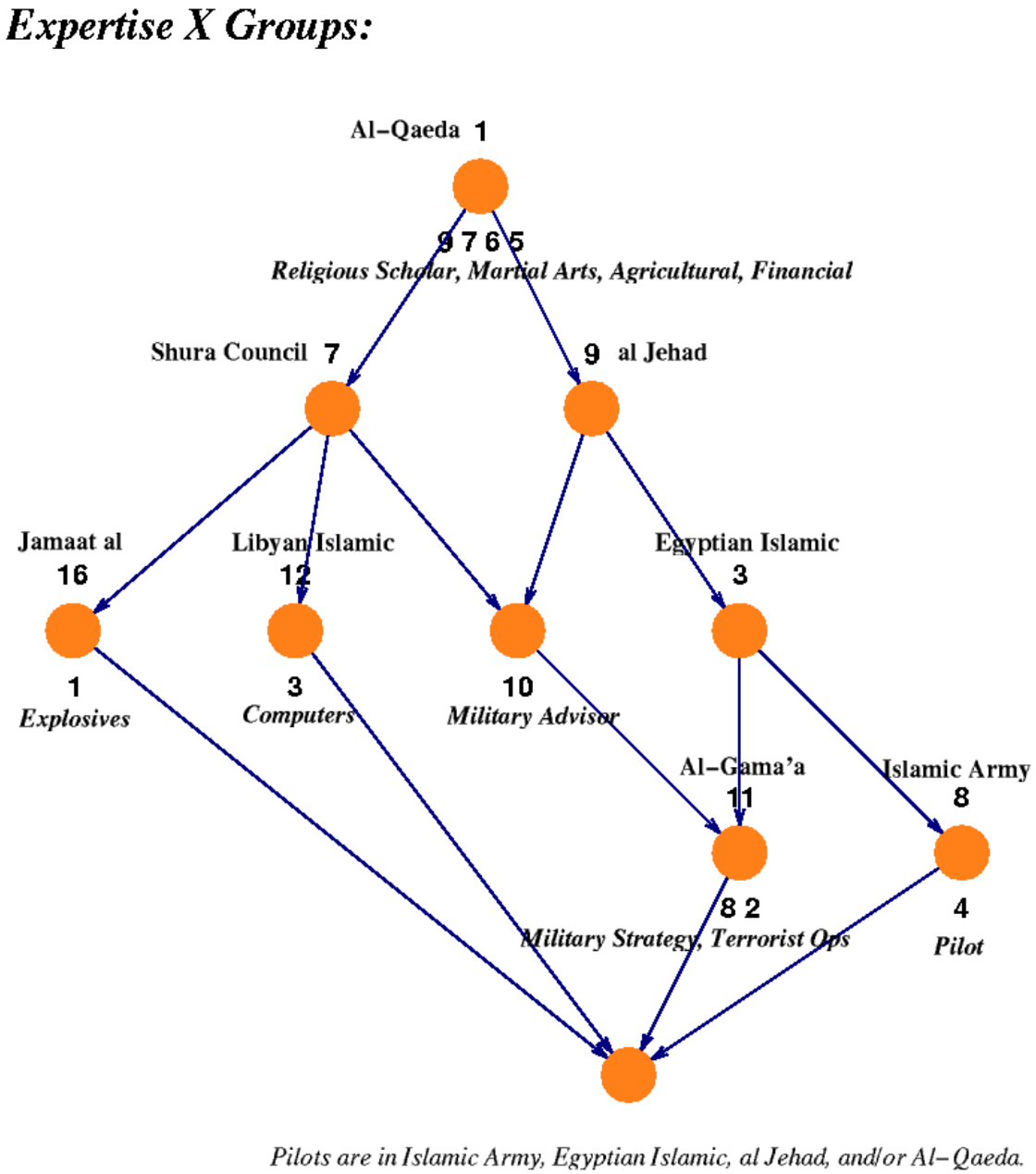


Figure 31: Expertise × Groups lattice: `GraphPlace`.

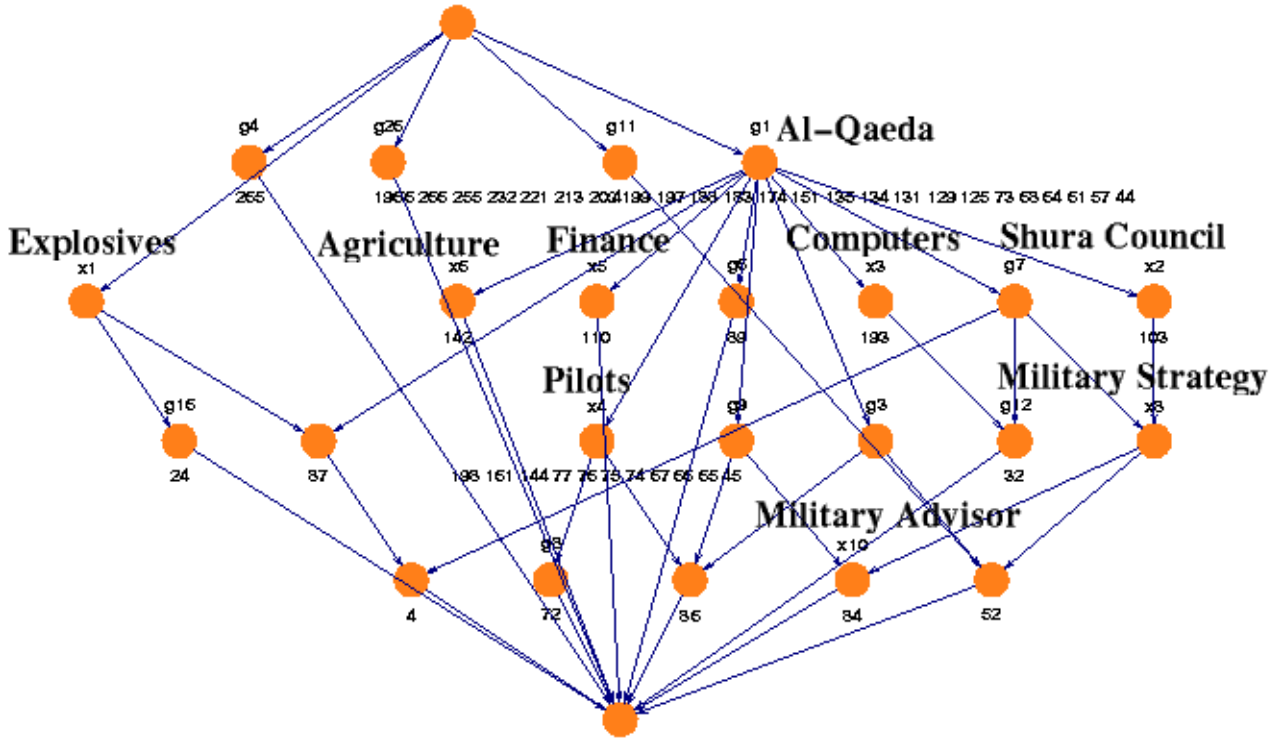


Figure 32: People × (Groups ∪ Expertise) lattice: `GraphPlace`.

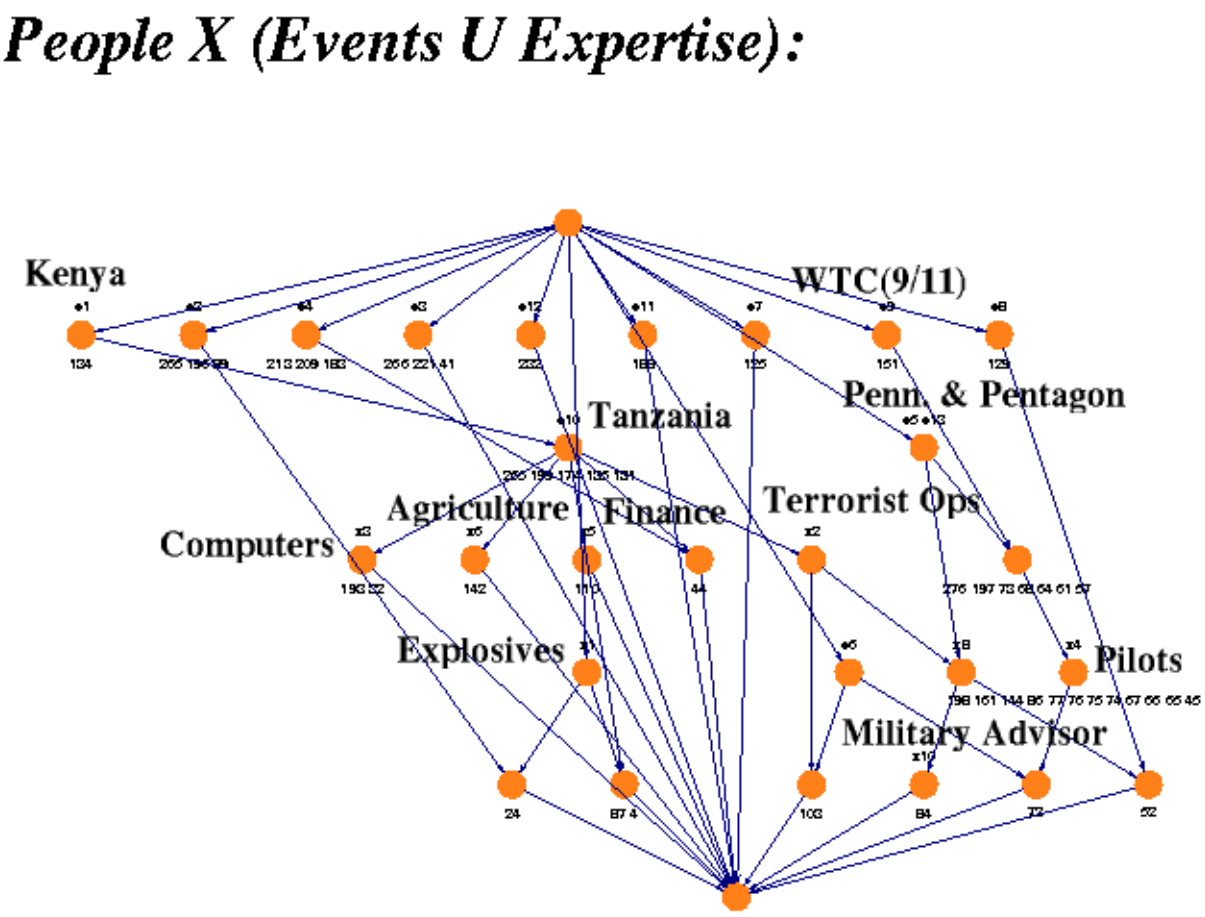


Figure 33: People × (Events ∪ Expertise) lattice: `GraphPlace`.

view is getting too complex to be able to analyze by eye. It includes all of the observations noted in subsets of the lattice. Additionally, one can see multi-attribute relations such as the Kenya(e1), Tanzania(e10), and 9/11 (e5,e9,e13) events related by Shura Council(g7) Military/leadership expertise (x2,x8,x10) by Ayman Mohammed Rabie al-Zawahiri(52) and Muhammad Atef(84).

Lattices can be created with multiple attributes as above. Analyzing these "N-D" views fully will require some automated assistance, such as the implication and graph processing as noted previously.

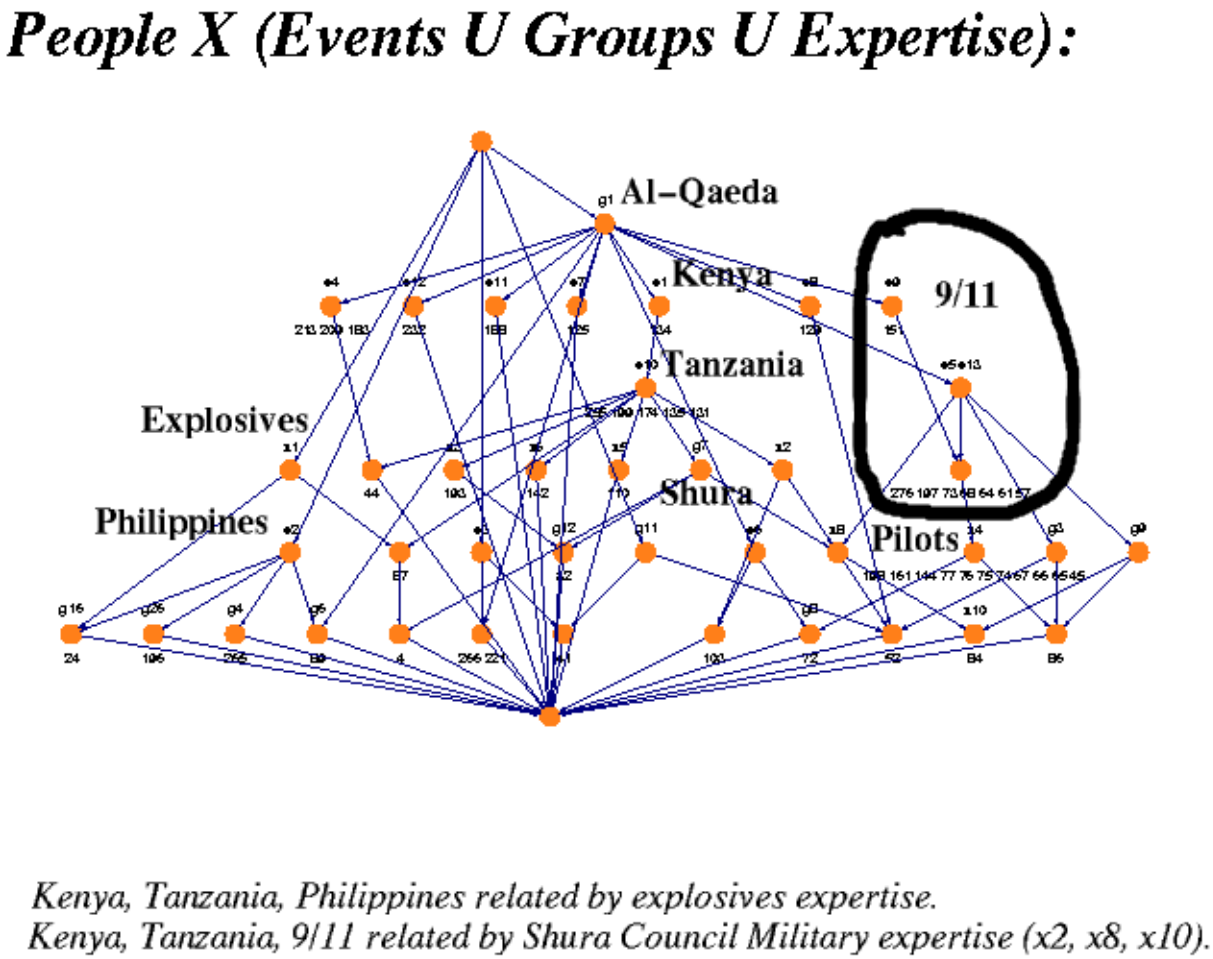


Figure 34: People × (Events × Groups ∪ Expertise) lattice: `GraphPlace`.

4.5 Future Directions

In this project, our goal was to advance our knowledge of FCA and deploy it, using existing tools, against project databases, and, where possible, also identify a research program on the science and technology of FCA itself. We have been successful in all of these goals, and in so doing, have identified a number of directions for future development.

4.5.1 FCA Theory

There are a number of aspects of FCA theory which intrigue us for potential future development:

FCA For Link Analysis: Generally, we see FCA as a central tool and representational mechanism for all forms of relational data. In particular, it should be developed and deployed in the context of the link analytical techniques previously introduced in Sec. 3.1, and incorporated into that mathematical formalism [20, 21]. The general idea is that on the identification of a particular view $\mathcal{D}_{n,m}$, it could be scaled into a context and sent directly through an FCA tool for examination. Chaining in this context amounts to shifting among views by adding or subtracting fields. An example from this report is moving from the People $\times$ Events view (Sec. 4.4.2) to the People $\times$ (Events $\cup$ Groups) view (Sec. 4.4.5). Similar ideas have been advanced by others [42].

However, this overall goal has a number of specific aspects, which are mentioned immediately below.

N-ary Relations: Critical for this effort is a better way to handle non-binary relations, that is, N-ary relations when $N > 2$. Currently, the "unioning" method is available, although it is not entirely satisfactory. In particular, given a view $X_1 \times (X_2 \cup X_3)$, it is not possible to distinguish that X_2 and X_3 in fact are different kinds of variables with different semantics, rather than simply more values of some "new" variable. In particular, what is required is a Galois theory relating at least *triples* of subsets $A \subseteq X_1, B \subseteq X_2$, and $D \subseteq X_3$.

Non-Boolean Relations: The idea of fuzzy concept lattices mentioned before [4] is also potentially powerful, promising new ways to handle non-Boolean relations. The point is that there is a problem with handling non-Boolean relations by scaling, in that scaling does not represent the fact that, for a single record, the various values of the scaled field are mutually exclusive. The problem is even worse when unions are introduced.

Measures in Concept Lattices: Finally is the idea to develop formal measures on Galois lattices. This is related to the idea suggested in Sec. 4.1, that concepts can be examined with respect to their position in the lattice. In particular, we are investigating distance and pseudo-distance measures in lattices in general [22], which, when combined with observations of the cardinalities of extents and intents relative to such a position measure, should be able to yield information about the relative support of different hypotheses. Similar ideas are discussed further in Sec. 4.5.3 below.

4.5.2 FCA As Part of a Suite of Tools

FCA will be useful for Homeland Security applications as part of a suite of software tools. A tool that provides initial data analysis and statistics (such as Viztool) should be used for frontend

preprocessing to focus on a set of data that one would like to analyze with a lattice. Another tool would build the lattice. Another would provide display capability for the lattice and its sub-lattices with a selection of layouts and showing object and/or attribute names and counts as requested. Additional automated analysis tools would provide creation of implications and graph theory primitives.

The implications or rules generated would be supplemented with some intelligent processing for grouping into useful sets and providing an English description, as seen in the FCA examples. Highlighting of an implication in a list should highlight the matching structure in the lattice display.

Graph theory primitives and algorithms would be provided for identifying structure and relations within the lattice [36]. Some of the graph primitives and algorithms that may be useful are center, diameter, radius, in/out degree, connected components, shortest path, minimum spanning tree, and Maxflow. Further work is required to determine which primitives and algorithms would be useful and how they may be interpreted to provide English descriptions.

The integration of techniques from Network Analysis, previously described in this report would provide additional automated analysis for identifying tightly and loosely coupled relations.

4.5.3 Automated Measure of Lattices

Link analysis, in our sense, is a semi-automated process of guided knowledge discovery in databases. We discussed in Sec. 3.3 that statistical measures are avaiable within a tool like VisTool to help guide users to find areas of local structure. And, we discussed in Sec. 4.5.1 about the availability of lattice measures as one possible such statistic. These statistics should be available in an interactive as well as a batch form. As the amount of data being analyzed becomes large this will be important. Identifying interesting areas within a large lattice automatically will allow the user to interactively examine those areas more carefully.

Typically we have seen FCA's built and used in visual tools for human-centered exploration of relationships and hierarchies in existing free software to understand structure. As we have shown, implications or rules can be generated. The question remains, what other kind of automated analysis can be done with an FCA lattice to help find and focus on "something interesting". We would like to be able to handle a large amount of data selected by an analyst and calculate the Galois lattices, useful measures (e.g. distance), comparisons, clustering, and hints on finding areas of local interest.

How does one read or interpret a lattice? Commonly, the concept lattice can be interpreted as a classification hierarchy or ontology, where the higher concepts are more general and the lower concepts are more specific. There can be different semantic interpretations based on the data and the questions to be explored. Location in the lattice may imply the more important ideas, or the key players, or the most involved, isolated instances, or anomalies, or something that has not been seen before. The point is that one needs to understand something about the context of their set of examples and attributes and the questions they want to explore.

A concept lattice can be constructed from all objects and attributes in a database, but would be too large to visualize. It makes more sense to focus on one question at a time, selecting the relevant subset of attributes and appropriate set of objects (examples). The resulting lattice may still be quite large or complex, but with appropriate measures, areas of interest could be identified through automated processing.

Some interesting research in areas relevant to finding something interesting includes:

Implication and Rule Generation: Each node of the concept lattice can also be viewed as 1 or more rules based on their attributes (e.g., a_1 and a_2 implies a_3 for a node that includes $a1, a2$, and $a3$). This can be another way to understand the generalizations in the data. As we have seen previously, ConImp enumerates the implications from a generated lattice. Rulelearner [35] uses these node implications as potential rules for classification. Duquenne *et al.* [13] used Galois lattices and implications to compare data about right- and left-handedness by looking at shared and non-shared implications.

Distance Measures: It does not appear that much work has been done combining distance measures with lattices [30]. One suggestion is the notion of number of hops between nodes in a lattice, where there is one hop between two directly connected nodes. However, elsewhere we have noted some of the deficiences of such simplified measures in lattice-valued spaces [22].

Nguifo and Njiwoua [27] combined lattice-based feature transformations with instance-based learning classification in a tool called IGLUE. It builds a join lattice of the initial context of objects described with binary features. Nodes are selected for relevance using entropy, improved entropy, or Laplace's formula. Laplace's formula was found to perform significantly better than the other to on a number of ML datasets from the UC/Irvine repository. This technique does require positive and negative examples. The number of levels of the lattice and a threshold for the selection function determine the resulting lattice. The attributes that remain are the relevant ones. These attributes or features are transformed into a continuous value d, based on their number of appearances in the lattice. The data is redescribed into ds using only the relevant attributes. These features can now be used in distance calculations between examples using Mahanalobis, Manhattan, or Euclidean distance measures.

Identifying Interesting Attributes: In FCA, a conceptual scale is used to represent a conceptual hierarchy describing the semantics for the range of values of one or more attributes. In essence, conceptual scales can be created for relevant combinations of attributes on the same data set, each one as a different lattice. Stumme has suggested [38] that a set of conceptual scales can be ranked by calculating a Chi-Square based measure and arranging the scales in descending order. This could help suggest the set of attributes to focus on for selected data.

Identifying Interesting Concepts: Concepts could be selected based on the number of objects (frequency or relative frequency) they represent or by how many sub-concepts they have. Selecting of concepts based on entropy [27, 28] may also useful. Stumme has also suggested [40] that a refinement of Pearson's Chi-Square calculations for contingency tables on the expected and observed frequencies of concepts between two scales may be useful for focusing on concepts that are dependent on the attributes in both scales.

Compression of lattices involves keeping the most important information, which should be the most interesting concepts based on some criteria. Van der Merwe and Kourie [45] listed criteria for selecting concepts, including those with extent size within a defined range, based on the number of child or parent concepts, based on an estimate of prior probability of the concept, and based on the difference between the expected and observed concept probabilities.

Clustering: A lattice itself can be considered a clustering structure. Clusters can be determined by the more general concepts. Conceptual clustering can be done with Iceberg Concept

Lattices [39] which use the topmost part of a lattice to define the clusters based on all the attributes of a dataset that are present up to some selected percentage threshold.

4.5.4 When the Amount of Data Becomes Large

There will come a time when the amount of data we want to process using FCA will become quite large and visualization on a workstation screen will become difficult to impossible. The data could be processed in pieces, with thresholds (e.g., based on frequency), or as Iceberg lattices [39] which serve as a condensed representation of frequent patterns.

Another approach is large-scale parallel processing for generation of the lattice and automated analysis to support interactive use or as a batch run. There are different algorithms for generating lattices such as Concepts, Lattice, NextConcept, and Titanic [26] that can be parallelized. The task of mining frequent patterns is similar for lattices and association rules. Parallel algorithms are available for association rule mining [19] that can be used as a starting point for parallel lattice processing.

The automated analysis methods such as generation of implications and graph algorithms would also need to be developed for parallel lattice processing. Visualization of these large lattices could be by selecting a sub-lattice based on some criteria or viewing condensed versions. Virtual Reality methods would be useful if one wanted to view and explore an entire lattice at once.

5 Acknowledgements

Many people, some directly working on this project and others just helping out, were instrumental in making this work possible. We first have to thank Jane Riese of CCN-12, without whose efforts well above and beyond the call of duty VisTool would not have been resurrected. Jim Gattiker did a wonderful job wrangling the data into the relational schema, and Eugene Gavrilov must be thanked for the initial heroic effort to get us supported with a MySQL installation. John Hogden was consistently supportive and encouraging, patiently bearing with our rantings about lattices through a number of afternoon sessions. Finally, without our new NIS-8 friends (April, Anna, Susan, and Ann) this all would have been impossible, irrelevant, unsupported, and unverifiable.

A FCA Mathematics

The following is adapted from Ganter and Wille [16, pp. 1-23]. To ease the reader, we've standardized, and, we hope, clairified notation. We've also pointed out some things we think are important.

A.1 Galois Connections and Lattices

Let $\langle V, \leq \rangle, \langle W, \leq \rangle$ be two posets.

Definition 1 (Dually Adjoint Mappings = Galois Connection) The functions $\varphi: V \mapsto W$ and $\psi: W \mapsto V$ are **dually adjoint**, or have a **Galois connection**, if:

- $\forall v_1, v_2 \in V, v_1 \leq v_2 \rightarrow \varphi(v_1) \geq \varphi(v_2)$

- $\forall w_1, w_2 \in W, w_1 \leq w_2 \rightarrow \psi(w_1) \geq \psi(w_2)$
- $\forall v \in V, v \leq \psi(\varphi(v))$
- $\forall w \in W, w \leq \varphi(\psi(w))$

Note that this Galois relation specifices a kind of "inverse" mapping: while $v_1 \leq v_2$, nonetheless it's $\varphi(v_1) \geq \varphi(v_2)$.

Now let $\langle V, \leq \rangle, \langle W, \leq \rangle$ be more specifically two complete lattices.

Definition 2 (Lattice Concepts) In a **lattice**, any two elements v_1, v_2 have a unique glb $v_1 \wedge v_2$ and an lub $v_1 \vee v_2$. In a **complete lattice** there's further a unique global infemum $\mathbf{0} = \bigwedge_{v \in V} v$ and supremum $\mathbf{1} = \bigvee_{v \in V} v$. Let $v^+ := \{v' \geq v\}$ be the **principle filter** (the "cone above v") of v and $v_- := \{v' \leq v\}$ its **principle ideal** (the "cone below").

Theorem 3 φ has a unique dual adjoint ψ iff

$$\forall v, \quad \exists w, \quad \varphi(v_-) \subseteq w^+.$$

Theorem 4 (Dually Adjoint Mappings Between Lattices) If V and W are complete, then φ has a dual adjoint ψ iff

$$\forall A \subseteq V, \qquad \varphi\left(\bigvee_{v \in A} v\right) = \bigwedge_{v \in A} \varphi(v)$$

A.2 Concept Lattices as "Galois Lattices"

Definition 5 (Context, Concept, etc.) Define a **context** as the tuple $\mathcal{K} := \{G, M, I\}$, where $G = \{a\}$ and $M = \{b\}$ are sets and $I \subseteq G \times M$. Denote $A \subseteq G, B \subseteq M$, and

$$A^I := \{b : \forall a \in A, \langle a, b \rangle \in I\}, \qquad B^I := \{a : \forall b \in B, \langle a, b \rangle \in I\}.$$

For short-hand denote $A' := A^I, B' := B^I$.

Note that we got a little confused for a while because, despite cursory appearances to the contrary, A^I is *not* the image of A in I, denoted $I(A)$. That would be $I(A) = \{b : \langle a, b \rangle \in I\}$. Where I^A differs is that it's not the set of all b associated with *any* of the a, but rather those associated with *all* and *each* of the b.

Thus, FCA stands in a converse relation to the "transitive closure" or "ping-ponging" methods of looking at propagation through a binary relation. As an example, consider Fig. 35, which shows the relation of the simple example as a bipartate graph. Consider then an initial set $A = \{3\}$, and then its sequence of images:

$$A = \{3\}, \quad I(A) = \{b, c, d\}, \quad I^{-1}(I(A)) = \{1, 3, 4\},$$

$$I(I^{-1}(I(A))) = \{a, b, c, d\} = G, \quad I^{-1}(I(I^{-1}(I(A)))) = \{1, 2, 3, 4\} = M.$$

This is typical: the images grow to find the connected components of the initial set.

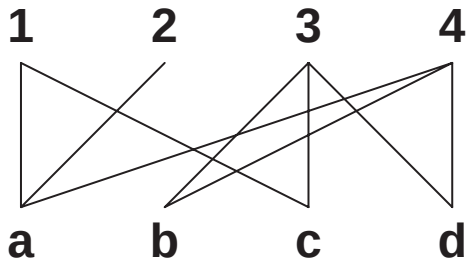


Figure 35: Simple example as a bipartate graph.

Now consider the sequence of images using the FCA operations:

$$A = \{3\}, \quad A' = \{b, c, d\}, \quad A'' = \{3\} = A, \quad A''' = \{b, c, d\} = A'.$$

This is because the FCA operations are a **closure** property. Another way of looking at it is that the FCA image is the *intersection* of images, whereas the relational operation is the *union* of them:

$$I^A = \bigcap_{a \in A} I(a), \qquad I(A) = \bigcup_{a \in A} I(a).$$

So continuing on, $\left\langle 2^G, \subseteq \right\rangle$ and $\left\langle 2^M, \subseteq \right\rangle$ are complete lattices, where 2^X is the power set of X.

Theorem 6 $\varphi_I : 2^G \mapsto 2^M$ and $\psi_I : 2^M \mapsto 2^G$, where $\varphi_I(A) := A', \psi_I(B) := B'$, are dually adjoint by the ordering $\subseteq$.

Definition 7 (Formal Concept) A is an **extent**, B an **intent**, and $C := \langle A, B \rangle \in 2^G \times 2^M$ a **concept** if $A' = B$ and $B' = A$.

Definition 8 (Concept Lattice) Let $\mathcal{C}(\mathcal{K}) := \{C\} = \{\langle A, B \rangle\} \subseteq 2^G \times 2^M$ be the set of all concepts of $\mathcal{K}$. Define an ordering on $\mathcal{C}(\mathcal{K})$ as

$$C_1 \preceq C_2 := A_1 \subseteq A_2.$$

Then $\underline{\mathcal{C}}(\mathcal{K}) := \left\langle 2^G \times 2^M, \preceq \right\rangle$ is a **concept lattice**.

Corollary 9 $C_1 \preceq C_2 \rightarrow B_1 \supseteq B_2$

Where there is no ambiguity, let $\underline{\mathcal{C}} := \underline{\mathcal{C}}(\mathcal{K})$.

A concept lattice is what we've been calling a **Galois lattice**. This makes sense in that $\preceq$ is defined on a cartesian product of 2^G and 2^M, and the orderings $\subseteq$ defined by these two components are dually adjoint, or have a Galois connection, by (6).

Note also that $\underline{\mathcal{C}} = \underline{\mathcal{C}}(\langle G, M, I \rangle) = \left\langle 2^G \times 2^M, \preceq \right\rangle$ is a lattice quite distinct from both its consituents $\left\langle 2^G, \subseteq \right\rangle$ and $\left\langle 2^M, \subseteq \right\rangle$. Not only is $\underline{\mathcal{C}}$ at a much "higher level", but it also reflects the nature of this arbitrary relation I between G and M. Indeed, the whole purpose of this methodology is to represent I, our database or relation of interest. $\left\langle 2^G, \subseteq \right\rangle$ and $\left\langle 2^M, \subseteq \right\rangle$ are simply supporting structures on the domain and codomain of that I.

Corollary 10 $\forall A, \langle A'', A' \rangle \in \underline{\mathcal{C}}$.

Theorem 11 $\underline{\mathcal{C}}$ is a complete lattice with infemum and supremum

$$\mathbf{0}(\underline{\mathcal{C}}) = \bigwedge_{C \in \underline{\mathcal{C}}} C = \left\langle \bigcap_{C \in \underline{\mathcal{C}}} A, \left(\bigcup_{C \in \underline{\mathcal{C}}} B \right)'' \right\rangle,$$

$$\mathbf{1}(\underline{\mathcal{C}}) = \bigvee_{C \in \underline{\mathcal{C}}} C = \left\langle \left(\bigcup_{C \in \underline{\mathcal{C}}} A \right)'', \bigcap_{C \in \underline{\mathcal{C}}} B \right\rangle.$$

References

[1] *Anaconda*, `http://www.mathematik.tu-darmstadt.de/ags/ag1/Software/Anaconda/Welcome_en.html.`

[2] Anantaram, C; Nagaraja, G; and Nori, KV: (1998) "Verification of Accuracy of Rules in a Rule Based System", *Data and Knoelwegde Engineering*, v. **27**, pp. 115-138

[3] Bartel, Hans-Georg: (2000) "Formal Concept Analysis and Chemeometrics", *Communications in Mathematics and Computer Chemistry*, v. **42**, pp. 25-38

[4] Belohlavek, Radim: (2002) "Logical Precision in Concept Lattices", *J. Logic Computation*, v. **12**:1, pp. 137-148

[5] Brüggemann, R; Volgt, K; and Steinberg, CEW: (1997) "Application of Formal Concept Analysis to Evaluate Environmental Databases", *Chemosphere*, v. **35**:3, pp. 479-486

[6] *Cernato*, `http://www.navicon.de`

[7] Cole, R and Eklund, P: (1999) "Analyzing an Email Collection Using Formal Concept Analysis", *Principles of Data Mining and Knowledge Discovery*, v. **1704**, pp. 309-315

[8] *Concepts*, `ftp://ftp.ips.cs.tu-bs.de:pub/local/softech/misc`

[9] *Concept Explorer*, `http://www.mathematik.tu-darmstadt.de/ags/ag1/Software/ConExp/index.html.`

[10] *ConImp*, `http://www.mathematik.tu-darmstadt.de/ags/ag1/Software/DOS-Programme/Welcome_de.html`

[11] Cristofor, Dana; Cristofor, Dan; and Simovici, Dan: (2000) "Galois Connections and Data Mining", *J Universal Computer Science*, v. **6**:1, pp. 60-73

[12] Cristofor, Laurentiu and Simovici, Dan: (2002) *Mining Association Rules in Entity-Relationship Modeled Databases*

[13] Duquenne, V; Chabert, C; and Cherfouh, A et al.: (2001) "Structuration of Phenotypes/Genotypes through Galois Lattices and Implications", in: *ICCS'01 Int. Wshop. on Concept Lattices-Based KSS*

[14] *FCA Surveyt*, `http://www.mathematik.tu-darmstadt.de/~plueschke/fcatools/programs.html.`

[15] Ganter, Bernhard and Kuznetsov, Sergei O: (2000) "Formalizing Hypotheses with Concepts", *Lecture Notes in Artificial Intelligence*, v. **1867**, pp. 342-356, Springer-Verlag

[16] Ganter, Bernhard and Wille, Rudolf: (1999) *Formal Concept Analysis*, Springer-Verlag

[17] Glass, JT; Zaloom, Victor; and Gates, David: (1991) "Computer-Aided Link Analysis (CALA)", *Computers in Industry*, pp. 179-187

[18] Hereth, Joachim; Stumme, Gerd; and Wille, R et al.: (2000) "Conceptual Knowledge Discovery and Data Analysis", *Lecture Notes in Artificial Intelligence*, v. **1867**, pp. 421-437

[19] Joshi, MV; Han, E; and Karypis, G et al.: (1999) "Efficient Parallel Algorithms for Mining Associations", in: *Large-Scale Parallel Data Mining: LNAI*, v. **1759**, ed. Zaki, M. J. and Ho, C, pp. 83-126, Springer

[20] Joslyn, Cliff: (2002) "Network Worlds: From Link Analysis to Virtual Places", in: *Proc. 2002 Conf. on Virtual Worlds and Simulation*, `ftp://ftp.c3.lanl.gov/~joslyn/vwsim02f.pdf`

[21] Joslyn, Cliff: (2002) "Link Anlaysis of Social Meta-Networks", *2002 Conf. on Computational Analysis of Social and Organizational Systems (CASOS 02)*, `ftp://ftp.c3.lanl.gov/~joslyn/casos02f.pdf`

[22] Joslyn, Cliff: (2002) "The Bio-Ontological Challenge: Representations of, and Measures in, Lattice-Valued Spaces", *Workshop on Enabling Concepts for Systems Biological Modeling*, Santa Fe, `ftp://ftp.c3.lanl.gov/~joslyn/enablingf.pdf`

[23] Joslyn, Cliff and Mniszeiski, Susan: (2002) "DEEP: Data Exploration through Extension and Projection", *Knowledge Discovery and Data Mining*, in preparation, `ftp://ftp.c3.lanl.gov/~joslyn/deep.pdf`

[24] Kleinberg, Jon M: (1999) "Hubs, Authorities, and Communities", *ACM Computing Surveys*, v. **31**:S4, pp. U21-U23

[25] Kuznetsov, SO: (2001) "Machine Learning on the Basis of Formal Concept Analysis", *Automation and Remote Control*, v. **62**:10, pp. 1543-1564, `http://sciserver.lanl.gov/pdflinks/01112720022216841--journals--00051179--v62i0010--1543_mlotbofca.`

[26] Lindig, C: (2000) "Fast Concept Analysis", in: *8th International Conference on Conceptual Structures*

[27] Nguifo, EM and Njiwoua, P: (1998) "Using Lattice-based Framework as a Tool for Feature Extraction", in: *Feature Extraction, Construction, and Selection: A Data Mining Perspective*, Kluwer

[28] Nguifo, EM and Njiwoua, P: (2001) "IGLUE: A Lattice-based Constructive Induction System", *Intelligent Data Analysis Journal*, v. **5**:1, pp. 73-81, IOS Press

[29] Office of Technology Assesment: (1995) "Information Information Technologies for Control of Money Laundering", *OTA-ITC-630*, US GPO, Washington DC, `http://www.wws.princeton.edu/~ota/disk1/1995/9529_n.html`

[30] Priss, UE: (1997) "A Graphical Interface for Document Retrieval Based on Formal Concept Analysis",

[31] Priss, UE: (1999) "Efficient Implementation of Semantic Relations in Lexical Databases", *Computational Intelligence*, v. **15**:1, pp. 79-87

[32] Priss, UE: (2000) "Lattice-based Information Retrieval", *Knowledge Organization*

[33] Rada, Roy; Mili, Hafedh; Bicknell, E; and Maria Blettner: (1989) "Development and Application of a Metric on Semantic Nets", *IEEE Trans. on Systems, Man and Cybernetics*, v. **19**:1, pp. 17-30

[34] Resnick, Philip: (1999) "Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems in Ambiguity in Natural Language", *J. Artificial Intelligence Research*, v. **11**, pp. 95-130

[35] Sahami, Mehran: (1995) "Learning Classification Rules Using Lattices", in: *European Conference on Machine Learning*, pp. 343-346

[36] Skiena, Steven: (1990) *Implementing Discrete Mathematics*, Addison-Wesley, Reading MA

[37] Sowa, John F: (2000) *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole, Pacific Grove

[38] Stumme, G.: (1998) "Exploring Conceptual Similarities of Objects for Analyzing Inconsistencies in Relational Databases", In: L. Bing, W. Hsu, W. Ke (Eds.): *Proc. Workshop on Knowledge Discovery and Data Mining, 5th Pacific Rim Intl. Conf. on Artificial Intelligence*, Singapore, Nov. 22-27, pp. 41-50

[39] Stumme, G: (2002) "Efficient Data Mining Based on Formal Concept Analysis", in: *LNCS 2453*, Springer

[40] Stumme, G: (2002) "Formal Concept Analysis on its Way from Mathematics to Computer Science", in: *ICCS 2002, LNAI 2393*, pp. 2-19, Springer

[41] Stumme, G., Taouil, R., Bastide, Y., and Lakhal, L., "Conceptual Clustering with Iceberg Concept Lattices", In: R. Klinkenberg, S. Rüping, A. Fick, N. Henze, C. Herzog, R. Molitor, O. Schröder (Eds.): *Proc. GI-Fachgruppentreffen Maschinelles Lernen '01*, Universität Dortmund 763, (FGML 2001).

[42] Stumme, G., Wille, R., Wille, U.: (1998) "Conceptual Knowledge Discovery in DataBases Using Formal Concept Analysis Methods", In: J. M. Zytkow, M. Quafofou (Eds.): *Principles of Data Mining and Knowledge Discovery. Proc. 2nd European Symposium on PKDD'98*, LNAI 1510, Springer, Heidelberg, pp. 450-458.

[43] *ToscanaJ*, `http://sourceforge.net/projects/toscanaj`

[44] *Toscana*, `http://www.mathematik.tu-darmstadt.de/ags/ag1/Software/Toscana/Welcome_en.html`

[45] Van der Merwe, FJ and Kourie, DG: (2002) "Compressed Pseudo-Lattices", *JETAI Journal on Concept Lattices for KDD*

[46] Wille, Rudolf: (1997) "Conceptual Graphs and Formal Concept Analysis", in: *Lecture Notes in Artificial Intelligence*, v. **1257**, pp. 290-303