Essays on using Formal Concept Analysis in Information Engineering

Proefschrift voorgedragen
tot het behalen van de graad
van Doctor in de Toegepaste
Economische Wetenschappen
door

Jonas POELMANS
Committee

*Supervisor*  Prof. Dr. Guido Dedene  K.U.Leuven
*Supervisor*  Prof. Dr. Stijn Viaene  K.U.Leuven
*Supervisor*  Prof. Dr. Monique Snoeck  K.U.Leuven
Prof. Dr. Henrik Scharfe  Aalborg University
Prof. Dr. Ir. Rik Maes  Universiteit van Amsterdam
Prof. Dr. Aimé Heene  Universiteit Gent

*Daar de proefschriften in de reeks van de Faculteit Economische en Toegepaste Economische Wetenschappen het persoonlijk werk zijn van hun auteurs, zijn alleen deze laatsten daarvoor verantwoordelijk*
First and foremost I wish to express special thanks to my supervisor Guido Dedene. His guidance during my doctoral trajectory and numerous research ideas were of great value and helped me achieve this end result. I also want to express special thanks to Stijn Viaene for carefully reading and giving feedback on every text I wrote. Our numerous discussions helped me in staying enthusiast on doing research and improving my work. I also want to thank Monique Snoeck for her help with the software engineering part of my thesis and for helping me with Microsoft Word every time I got stuck again. I am very grateful to Paul Elzinga from the Amsterdam-Amstelland police for our close cooperation on the domestic violence, human trafficking and terrorism data. Our discussions always resulted in interesting research avenues and we made many interesting discoveries that will hopefully be beneficial to police officers working in the field. I would like to thank the police of Amsterdam-Amstelland for granting us the liberty to conduct and publish this research. In particular, we are most grateful to Deputy Police Chief Reinder Doeleman and Police Chief Hans Schönfeld for their continued support. I also want to thank my mother Gerda for our research work on breast cancer treatment and for her overall support during the course of my PhD. I want to thank the Gasthuiszusters Antwerpen and in particular Herman Van der Mussele for giving me access to their medical data. I would like to thank Edward Peters for our cooperation on the medical data analysis. I also want to thank Marc Van Hulle for his insights and help with the self organizing maps. I also want to thank my doctoral committee members Rik Maes, Aime Heene and Henrik Scharfe for their feedback on my final text. I would also like to thank my colleagues for the stimulating work environment. In particular, I would like to thank Karel for his help with Matlab programming, Raf for the many snooker evenings filled with fruit beers, Tom for always wishing me a good morning in the early afternoon, Geert and Pieter for the enjoyable talks in the evenings after work, David and Filip for their always funny jokes, Willem for the many enjoyable breaks during long working days, Wouter for his valuable advice on data mining, and finally Jochen and Helen for the pleasant talks during lunch breaks. Special thanks go to Vicky Verhaert for helping me prepare the final version
of my thesis. I would like to thank our secretary Nicole Meesters for the numerous times she helped me arranging my conference travels. I also want to thank my sister, father and my friends for supporting me during the course of my PhD. Finally I want to thank the “Fonds Voor Wetenschappelijk Onderzoek – Vlaanderen” for funding this research. This PhD was written as part of my aspirant of FWO research mandate.

Jonas, Leuven, July 2010
Contents

ACKNOWLEDGEMENTS ................................................................. III
CONTENTS .............................................................................................. V
SAMENVATTING .................................................................................. XIII
CHAPTER 1 .......................................................................................... XXI
INTRODUCTION .................................................................................. XXI
CHAPTER 2 .......................................................................................... 23
FORMAL CONCEPT ANALYSIS IN INFORMATION ENGINEERING:
A SURVEY .......................................................................................... 23
  2.1. Introduction .................................................................................. 23
  2.2. Formal Concept Analysis .............................................................. 24
    2.2.1. FCA essentials ........................................................................ 24
    2.2.2. FCA software ......................................................................... 27
    2.2.3. Web portal ............................................................................ 28
  2.3. Dataset ......................................................................................... 28
  2.4. Studying the literature using FCA .................................................. 29
    2.4.1. Knowledge discovery and data mining ..................................... 30
      2.4.1.1. Some FCA mining applications ........................................ 33
      2.4.1.2. Association rule mining ..................................................... 33
      2.4.1.3. Software mining ............................................................... 35
      2.4.1.4. Web mining ..................................................................... 36
      2.4.1.5. Extending FCA for data mining ....................................... 37
      2.4.1.6. FCA mining applications in biology and medicine... 38
      2.4.1.7. FCA in relation to other machine learning techniques 38
      2.4.1.8. Fuzzy FCA in KDD ......................................................... 39
2.4.2.  Information retrieval ................................................. 39
  2.4.2.1.  Knowledge representation and browsing with FCA . 42
  2.4.2.2.  Knowledge browsing systems based on FCA .......... 43
  2.4.2.3.  Query result improvement with FCA ................... 44
  2.4.2.4.  Defining and processing complex queries with FCA 45
  2.4.2.5.  Domain knowledge incorporation in query search results: contextual answers & ranking ................................. 45
  2.4.2.6.  Fuzzy FCA in IR ........................................... 46
2.4.3.  Scalability .............................................................. 47
  2.4.3.1.  Iceberg lattices .............................................. 48
  2.4.3.2.  Reducing the size of concept lattices ................. 48
  2.4.3.3.  Handling complex data ..................................... 49
  2.4.3.4.  FCA algorithm scalability ................................ 49
2.4.4.  Ontologies .............................................................. 50
  2.4.4.1.  Ontology construction ...................................... 51
  2.4.4.2.  Ontology quality management ............................ 52
  2.4.4.3.  Linguistic applications using FCA and ontologies ... 53
  2.4.4.4.  Ontology mapping and merging .......................... 54
  2.4.4.5.  Fuzzy and rough FCA in ontology construction and mapping 55
2.5.  Extensions and related disciplines of FCA theory .......... 55
  2.5.1.  Fuzzy FCA .......................................................... 56
  2.5.1.1.  Applications of fuzzy FCA .............................. 58
  2.5.1.2.  Fuzzy concept lattice reduction ......................... 58
  2.5.1.3.  Parameterized approaches for fuzzy concept lattices 59
  2.5.1.4.  Multi-adjoint and generalized concept lattices ...... 60
  2.5.1.5.  Variations to traditional fuzzy FCA .................... 61
  2.5.1.6.  Fuzzy attribute logic, implications and inference .... 62
  2.5.2.  Rough set theory .................................................. 62
  2.5.2.1.  Combining FCA with rough set theory ............... 64
  2.5.2.2.  Generalizations of rough set theory .................. 66
  2.5.2.3.  Attribute reduction ......................................... 67
  2.5.2.4.  Fuzzy FCA and rough set theory ....................... 67
2.5.3. Monotone concept and AFS algebra ......................... 68
2.5.4. Triadic FCA .............................................................. 69
2.5.5. Temporal FCA ......................................................... 69
2.6. Mathematical research and algorithmic innovations .... 70
2.7. Conclusions ................................................................. 71
CHAPTER 3 ............................................................................. 73
CURBING DOMESTIC VIOLENCE: INSTANTIATING C-K THEORY
WITH FORMAL CONCEPT ANALYSIS AND EMERGENT SELF
ORGANIZING MAPS ............................................................ 73
3.1. Introduction ................................................................. 73
3.2. Intelligence Led Policing ........................................... 76
  3.2.1. Domestic violence ................................................ 77
  3.2.2. Motivation ............................................................. 80
3.3. FCA, ESOM and C-K theory .................................... 81
  3.3.1. Formal Concept Analysis ...................................... 81
  3.3.2. Emergent Self Organizing Map ............................ 83
    3.3.2.1. Emergent SOM ............................................ 84
    3.3.2.2. ESOM parameter settings ............................ 85
  3.3.3. C-K theory ............................................................ 85
3.4. Instantiating C-K theory with FCA and ESOM .......... 87
3.5. Dataset ......................................................................... 89
  3.5.1. Data preprocessing and feature selection .............. 91
  3.5.2. Initial classification performance ....................... 92
3.6. Iterative knowledge discovery with FCA and ESOM ...... 94
  3.6.1. Transforming existing knowledge into concepts .... 96
  3.6.2. Expanding the space of concepts ...................... 103
  3.6.3. Transforming concepts into knowledge .............. 105
  3.6.4. Expanding the space of knowledge .................... 108
3.7. Actionable results ....................................................... 110
3.8. Comparative study of ESOM and multi-dimensional scaling 116
3.9. Conclusions ............................................................... 120
CHAPTER 4 ............................................................................. 123
EXPRESSING OBJECT-ORIENTED CONCEPTUAL DOMAIN MODELS IN FORMAL CONCEPTS ANALYSIS .................................................... 123

4.1. Using Formal Concept Analysis for verification of process – data matrices in conceptual domain models ................................................. 124

4.1.1. Introduction ................................................................................. 124
4.1.2. Research Question ................................................................. 125
4.1.3. FCA essentials ........................................................................ 127
4.1.4. MERODE essentials ............................................................. 130
4.1.5. FCA and the object-event matrices .................................. 133

4.1.5.1. Object-event table with empty row column ............ 134
4.1.5.2. Propagation rule and subconcept-superconcept relation 135
4.1.5.3. FCA and the contract rule ....................................... 137
4.1.5.4. Verification process ................................................ 140

4.1.6. Conclusions ............................................................................ 141

4.2. Extending Formal concept Analysis with a second ordering relationship on concepts: a software engineering case study ................ 141

4.2.1. Introduction ............................................................................. 142
4.2.2. Expressing UML associations with an FCA lattice ............ 142
4.2.3. Expressing inheritance relationships with an FCA lattice 144
4.2.4. Merging the existence dependency and inheritance lattices 146

4.2.4.1. Properties of extended lattice .................................. 148
4.2.5. Applications ........................................................................... 149
4.2.5.1. Accumulation rule ................................................... 149

4.2.6. Conclusions ............................................................................ 153

4.3. An Iterative Requirements Engineering Framework based on Formal Concept Analysis and C-K theory. ........................................... 154

4.3.2. Requirements engineering artifacts .................................. 156

4.3.2.1. Use cases, conceptual domain model and quality ... 156
4.3.2.2. UML class model normalization .................................... 157
4.3.2.3. UML associations and the use case-entity interaction matrix 165

4.3.3. C-K theory ........................................................................... 166

4.3.4. Iterative requirements engineering process using FCA ... 167
4.3.4.1. C-K theory in requirements engineering ................. 168
4.3.4.2. FCA lattice properties and relation with software artifacts 170
4.3.4.3. Case study: Book Trader System ......................... 173
4.3.4.4. Case study: Hotel Administration System ............ 179
4.3.4.5. Case Study: Ordering system ............................. 182
4.3.4.6. Case study: Elevator repair system ...................... 184
4.3.5. Validation experiment ............................................. 188
4.3.5.1. Participants ..................................................... 189
4.3.5.2. Setup of experiment ......................................... 189
4.3.5.3. Results of experiment ....................................... 191
4.3.6.1. FCA in requirements engineering ....................... 193
4.3.7. Conclusions .......................................................... 194

CHAPTER 5 .............................................................................................. 195

FORMAL CONCEPT ANALYSIS OF TEMPORAL DATA .................. 195

5.1. Detecting and profiling human trafficking suspects with Temporal Concept Analysis .............................................. 195
5.1.2. Backgrounder ......................................................... 197
5.1.2.1. Human trafficking ................................................ 197
5.1.2.2. Current situation ............................................... 197
5.1.3. Dataset ................................................................. 198
5.1.4. Human trafficking indicators ..................................... 199
5.1.5. Temporal Concept Analysis ....................................... 200
5.1.5.1. FCA essentials ............................................... 200
5.1.5.2. TCA essentials ................................................ 201
5.1.6. Detecting and profiling human traffickers .................. 203
5.1.6.1. Detecting suspects with FCA ............................. 203
5.1.6.2. Profiling suspects with TCA ............................... 205
5.1.6.3. Network evolution analysis using TCA .................. 207
5.1.7. Conclusions ............................................................ 208

5.2. Terrorist threat assessment with Temporal Concept Analysis 208
5.2.1. Introduction ............................................................ 209
5.2.2. Backgrounder ........................................................ 210
5.2.2.1. Home-grown terrorism ............................................ 210
5.2.2.2. The four phase model of radicalism ....................... 210
5.2.2.3. Current situation ............................................... 212
5.2.3. Dataset ........................................................................ 212
5.2.4. Formal Concept Analysis ........................................... 213
  5.2.4.1. Temporal Concept Analysis ................................ 214
5.2.5. Research method ..................................................... 214
  5.2.5.1. Extracting potential jihadists with FCA ............... 215
  5.2.5.2. Constructing Jihadism phases with FCA ............. 217
  5.2.5.3. Build detailed TCA lattice profiles for subjects .... 217
5.2.6. Conclusions ............................................................... 218

5.3. Combining business process and data discovery techniques for analyzing and improving integrated care pathways. ............................. 218
  5.3.1. Introduction ........................................................... 219
  5.3.2. Business process discovery .................................... 220
    5.3.2.1. Hidden Markov Models .................................. 221
    5.3.2.2. HMM-based process discovery ....................... 221
  5.3.3. Data discovery with Formal Concept Analysis .......... 222
  5.3.4. Dataset ..................................................................... 222
  5.3.5. Analysis and results .............................................. 223
    5.3.5.1. Quality of care analysis .................................. 224
    5.3.5.2. Process variations .......................................... 226
    5.3.5.3. Workforce intelligence .................................... 229
    5.3.5.4. Data entrance quality issues ......................... 231
  5.3.6. Discussion and conclusions ..................................... 231

CHAPTER 6 ................................................................................... 233
CONCLUSION ............................................................................... 233
  6.1. Thesis conclusion ..................................................... 233
  6.2. Future work ............................................................... 235
APPENDIX A ............................................................................... 239
LITERATURE SURVEY THESAURUS ........................................... 239
APPENDIX B ............................................................................... 241
DOMESTIC VIOLENCE THESAURUS ........................................ 241
Formele Concept Analyse (FCA) is een vrij nieuwe wiskundige techniek (Ganter 1999), die initieel gebruikt is voor de structurering van cluster analyses. De essentie van de techniek bestaat uit een reeks algoritmes om een associatie-matrix van “entiteiten” en “attributen” om te zetten naar een roosterstructuur (“lattice”) van concepten, die (groepen van) entiteiten voorstellen met gelijkaardige attributen, en omgekeerd. Belangrijk hierbij is dat een roosterstructuur een niet-hiërarchische, a-cyklische partiële orderrelatie vormt tussen de concepten.

In de eerste onderzoekslijn is een nieuwe originele toepassing van FCA ontwikkeld, waarbij FCA gebruikt wordt als een techniek om binnen grootschalige ongestructureerde gegevensbronnen concepten en patronen te herkennen. Er bestaat een grote verscheidenheid aan data mining technieken, vooral voor gestructureerde gegevens. Voor ongestructureerde gegevens zijn er niet alleen minder technieken beschikbaar, maar zijn er ook weinig methoden om de resultaten van de data mining analyse te interpreteren, en te valideren. In het kader van een leerstoel met de Politie Amsterdam-Amstelland is aangetoond dat FCA hier een bijzondere bijdrage kan leveren. In een proof-of-concept is een analyse doorgevoerd van politieverslagen rond huishoudelijk geweld binnen gezinnen. FCA is toegepast op reële gegevensverzamelingen, waarbij de roosterstructuur met concepten een duidelijke aanwijzing geeft van de relevante indicatoren voor huishelijk geweld. Meer nog, hiaten in de roosterstructuur openen de vraag naar bijkomende informatie-attributen in de politieverslagen, zodat de agenten, die de basisgegevens verzorgen door hun observatie, bijkomende aandachtspunten krijgen voor hun verslagen.

De toepassing van FCA heeft in de eerste plaats geleid tot een verbeterde definitie van het concept van huishelijk geweld. Uit het concept rooster dat voorvloeit uit FCA kunnen immers regels worden afgeleid, met de kansen dat de regel van toepassing is. Dit laat toe om zowel kennisgaten in de definitie te ontdekken, maar ook relevante uitzonderingen. Een belangrijke aandachtsfactor is de combinatorische explosie van het FCA algoritme bij grotere aantallen attributen, waarvoor in eerste instantie gewerkt is met clustering van die attributen, die op basis van de probabiliteiten een sterke vorm van samenhang vertonen. Om een fijnmazige analyse te doen van de
uitzonderingen is FCA vervolgens uitgebreid en gecombineerd met een emergente (bottom-up) techniek – gebaseerd op neurale netwerken – Emergent Self Organising Maps. Uit de studies rond huiselijk geweld kwam duidelijk naar voor hoe beide technieken elkaar versterken, en leiden tot vrij sterke karakterisatietechnieken, waarbij tot 90% of meer van de gevallen van huiselijk geweld automatisch ontdekt kunnen worden. Het onderzoek heeft geleid tot een significante verbetering van de aanpak van huiselijk geweld. De Politie heeft in dit geval gezorgd voor een uitstekende basisverzameling van (geanonimiseerde) attributen (kenmerken) bij politieverslagen.

De tweede onderzoekslijn, die mee ontwikkeld werd in de gevallenstudie van huiselijk geweld, positioneert FCA als een kennisvernieuwingstechniek. In de literatuur zijn niet zoveel technieken bekend voor de concrete ontwikkeling van innovatieve ideeën. De vroege publicaties over artificiële intelligentie wezen reeds op het belang van de symbolisatie en conceptualisatie van de aanwezige kennis, op een dermate wijze dat mogelijke kennisgaten naar voor komen in de conceptualisatie. Een belangrijke systematisering van kennisvernieuwing is vanuit de design science benadering in de voorbije jaren ontwikkeld in de C-K-theory (Concept-Knowledge-Theory). Daarbij werd echter tot nu toe gebruik gemaakt van hiërarchisch geordende conceptbomen voor de conceptualisatie van bestaande kennis. In deze onderzoekslijn is voor de eerste maal aangetoond hoe FCA binnen de C-K-theory kan gebruikt worden als een techniek voor de disjunctie van bestaande kennis in niet-hiërarchische (partieel geordende) conceptroosters. Bovendien zijn de praktijkervaringen rond de gevallenstudie van huiselijk geweld gepositioneerd als leerprocessen binnen de C-K-theory, met een belangrijke bijdrage voor de conjunctie van nieuwe concepten in kennisvernieuwing.

In een derde onderzoekslijn is onderzocht hoe het gebruik van FCA uitgebreid en verfijnd kan worden voor gegevensbronnen met een inherente tijdsdimensie. In de literatuur is een temporele variant van FCA, Temporal Concept Analysis, bekend. Deze techniek is uitgediept in combinatie met FCA voor een gevallenstudie van politie rapporten voor de detectie en opsporing van mogelijke verdachten in de context van mensenhandel. Er is ondermeer vastgesteld hoe de FCA-roosters kunnen gebruikt worden als een voorstelling voor de verschillende mogelijkheden waaronder een verdachte kan evolueren, waarbij eveneens de probabiliteiten van de kenmerken een belangrijke rol spelen. Dit leidt tot een pro-actieve vroege opsporing met een hogere vaststellingsgraad, hetgeen verder is uitgewerkt in een breedschalig raamwerk voor de opsporing van potentiële terreurverdachten, dat vervolgens nu van de nodige software instrumenten wordt voorzien.

Tijdens de uitwerking van de derde onderzoekslijn zijn beperkingen naar voor gekomen in de tijdgerichte voorstelling van concepten, zeker waar het gaat om de structurele voorstelling van de evolutie van concepten in processen (op basis van sequentie, selectie en iteratie). In eerste instantie is daarom in een vierde onderzoekslijn FCA toegepast met events
(gebeurtenissen) in plaats van gegevensattributen. Hierbij worden niet zozeer de conceptroosters gebruikt voor de voorstelling van processen, maar wel Hidden Markov Modellen (HMM). FCA wordt dan wel gebruikt om de relevante (deel)verzamelingen van gebeurtenissen te filteren, vooral in de context van grootschalige gegevensbronnen. FCA helpt ook bij het in kaart brengen van eigenschappen van proces variaties en anomalieën, zoals kwaliteitsproblemen in de zorg verstrekt aan patiënten. Als gevallenstudie is voor deze onderzoekslijn een verzameling van behandelingseengevens gebruikt – conform aan Healthcare Level 7 – binnen de context van zorgprocessen voor borstkanker in een daartoe gespecialiseerd ziekenhuis. 470 behandelingstappen zijn samengebracht tot 65 relevante stappen, die resulteren in een belangrijke vereenvoudiging van het ontdekken van de zorgprocessen vanuit de behandelingseengevens. De eerste resultaten van deze onderzoekslijn hebben tevens geresulteerd in incrementele verbeteringen in de behandelingprocessen. Uit de gevallenstudie bleek tevens een belangrijke synergie tussen de ontdekking van data-aspecten versus proces-aspecten.

De vijfde onderzoekslijn heeft de mogelijkheden van FCA in verband met event-gebaseerde modellen verder gesystematiseerd in de context van domein modellering binnen software engineering. Daarbij is ondermeer aangetoond hoe de Object-Event tabel binnen de MERODE-methodologie voor software engineering, zich automatisch via FCA laat vertalen is een objectrooster dat isomorf is met de Existence Dependency Graph uit de MERODE-methodologie. FCA voegt algoritme toe aan de MERODE consistentie controle. Het gebruik van meer-waarde-tabellen laat toe om alle consistentieregels rond klassendiagrammen in UML formeel wiskundig uit te drukken. Bovendien kunnen de algoritmes van FCA gebruikt worden om inconsistenties en onvolledigheden in de modellen op te sporen. Hiermee is eigenlijk voor de eerste keer aangetoond dat “conceptuele modellen” ook effectief het begrip “concept” in de wiskundige zin van het woord hanteren. Vervolgens werd een driedimensionale variant van FCA verder onderzocht op zijn toepasbaarheid voor het combineren van bestaansafhankelijkheid met generalisatie/specialisatie. Tevens wordt onderzocht hoe FCA probabilistische informatie kan toevoegen aan de cardinaliteiten binnen klassendiagrammen in UML.

Tenslotte is in een zesde onderzoekslijn alle relevante literatuur rond FCA in kaart gebracht door middel van FCA. De concepten op basis van FCA zijn geïnventariseerd in een literatuur overzicht, dat een relevant beeld en inzicht geeft in de ongeveer 700 papers rond FCA die verschenen zijn in de periode 2003 – 2009. Deze benadering toont eveneens aan hoe FCA als een meta-techniek kan aangewend worden.
Het geleverde onderzoekswerk werd gepubliceerd in de volgende artikels:

JOURNAL PAPERS GEPUBLICEERD/AANVAARD:


**CONFERENCE PROCEEDINGS GEPUBLICEERD/AANVAARD:**


JOURNAL PAPERS INGEDIEND:


BOOK CHAPTER:


NEDERLANDSTALIGE PUBLICATIES:


AWARDS:

Nominated for best paper award at 8th Industrial Conference on Data Mining (ICDM), Leipzig, Germany, July 16-18, 2008

Winner of young professionals best paper award at 9th Industrial Conference on Data Mining (ICDM), Leipzig, Germany, July 20-22, 2009

Winner of best paper award at 10th Industrial Conference on Data Mining (ICDM), Berlin, Germany, July 12-14, 2010
CHAPTER 1

Introduction

Formal Concept Analysis was originally introduced as a mathematical theory by Rudolf Wille in 1982. Between the beginning of 2003 and the end of 2009, over 700 papers have been published in which FCA was used by the authors. We performed a semantic text mining analysis of these papers. We downloaded these 702 pdf-files and built a thesaurus containing terms related to FCA research. We used Lucene to index the abstract, title and keywords of these papers with this thesaurus. After clustering the terms, we obtained several lattices summarizing the most notorious FCA-related research topics. While exploring the literature, we found FCA to be an interesting meta-technique for clustering and categorizing papers in different research topics.

Over the years FCA has found its way from mathematics to computer science, resulting in numerous applications in knowledge discovery (20% of papers), information retrieval (15% of papers), ontology engineering (13% of papers) and software engineering (15% of papers). 18% of the papers described extensions of traditional FCA such as fuzzy FCA and rough FCA.

In this thesis we filled in some of the gaps in the existing literature (Poelmans et al. 2010g). During the past 20 years, the amount of unstructured data available for analysis has been ever-increasing. Today, 90% of the information available to police organizations resides in textual form. We investigated the possibilities of FCA as a human-centered instrument for distilling new knowledge from these data. FCA was found to be particularly useful for exploring and refining the underlying concepts of the data. To cope with scalability issues, we combined its use with Emergent Self Organising Maps. This neural network technique helped us gain insight in the overall distribution of the data and the combination with FCA was found to have significant synergistic results. The knowledge extraction process was framed in the C-K design theory. At the basis of the method are multiple successive iterations through the design square consisting of a concept and knowledge space. The knowledge space consists of the information used to steer the
action environment, while this information is put under scrutiny in the concept space.

The first real life case study zoomed in on the problem of domestic violence at the Amsterdam-Amstelland police. We exposed multiple anomalies and inconsistencies in the data and were able to improve the employed definition of domestic violence. An important spin-off of this KDD exercise was the development of a highly accurate and comprehensible rule-based case labelling system. This system is currently used to automatically assign a label to 75% of incoming cases whereas in the past all cases had to be dealt with manually.

Besides analysing data containing relations between objects and attributes, we analysed data containing objects and events. The binary relation of FCA changes its meaning from “object o has attribute a” to “object o participates in event a”. We studied FCA in relation to the software engineering methodology MERODE. We found that the lattice, automatically obtained from the Object-Event Table, is isomorphic to the Existence Dependency Graph (EDG) in MERODE. We also expanded traditional FCA theory with a third dimension. Traditional FCA only imposes one ordering relation on concepts, namely the subconcept-superconcept relation, whereas for the representation of models containing both ED and inheritance relationships, an extra ordering relationship or dimension is needed. Finally, we framed the requirements engineering process based on FCA in C-K theory. All steps in this process, namely elaboration, verification, modification and validation of the model, can be considered as the four operators of the design square. A case study with 20 students confirmed the method’s practical usefulness.

For the analysis of other phenomena such as human trafficking and terrorism threat, a complicating factor is the inherent time dimension in the data. We applied the temporal variant of FCA, namely Temporal Concept Analysis (TCA), to the unstructured text in a large set of police reports. The aim was to distill potential subjects for further investigation. In both case studies, TCA was found to give interesting insights into the evolution of subjects over time. Amongst other things, several (to the police unknown) persons involved in human trafficking or the recruitment of future potential jihadists were distilled from the data. The intuitive visual interface allowed for an effective interaction between the police officer who used to be numbed by the overload of information, and the data. In a subsequent step, we combined data discovery based on FCA with process discovery using Hidden Markov Models (HMM). This combination was used to gain insight into a breast cancer care process by analyzing patient treatment data. Multiple quality of care issues were exposed that will be resolved in the near future.
CHAPTER 2

Formal Concept Analysis in Information Engineering: a Survey

In this chapter, we analyze the literature on Formal Concept Analysis (FCA) and some closely related disciplines using FCA. We collected 702 papers published between 2003-2009 mentioning Formal Concept Analysis in the abstract. We developed a knowledge browsing environment to support our literature analysis process. The pdf-files containing the papers were converted to plain text and indexed by Lucene using a thesaurus containing terms related to FCA research. We use the visualization capabilities of FCA to explore the literature, to discover and conceptually represent the main research topics in the FCA community. We zoom in on and give an extensive overview of the papers published between 2003 and 2009 on using FCA in knowledge discovery and data mining, information retrieval, ontology engineering and scalability. We also give an overview of the literature on FCA extensions such as fuzzy FCA, rough FCA, temporal and triadic concept analysis.

2.1. Introduction

Formal Concept Analysis (FCA) was invented in the early 1980s by Rudolf Wille as a mathematical theory (Wille 1982). FCA is concerned with the formalization of concepts and conceptual thinking and has been applied in many disciplines such as software engineering, knowledge discovery and information retrieved during the last 15 years (Poelmans et al. 2010b). The mathematical foundation of FCA is described by Ganter et al. (1999) and introductory courses were written by Wolff (1994) and Wille (1997).

A textual overview of part of the literature published until the year 2004 on the mathematical and philosophical background of FCA, some of the applications of FCA in the information retrieval and knowledge discovery

---

1 Part of this chapter has been published in Poelmans, J., Elzinga, P., Viaene, S., Dedene, G. (2010) Formal Concept Analysis in Knowledge Discovery: a Survey. LNCS 6208, 139-153, 18th International Conference on Conceptual Structures.
2.1. INTRODUCTION

field and in logic and AI is given by Priss (2006). An overview of available
FCA software is provided by Tilley (2004). Carpineto et al. (2004a) present
an overview of FCA applications in information retrieval. In Tilley et al.
(2007), an overview of 47 FCA-based software engineering papers is given.
The authors categorized these papers according to the 10 categories as
defined in the ISO 12207 software engineering standard and visualized them
in a concept lattice. In Lakhal et al. (2005), a survey on FCA-based
association rule mining techniques is given.

In this chapter, we describe how we used FCA to create a visual overview
of the existing literature on concept analysis published between the years
2003 and 2009. The core contributions of this chapter are as follows. We
visually represent the literature on FCA using concept lattices, in which the
objects are the scientific papers and the attributes are the relevant terms
available in the title, keywords and abstract of the papers. We developed a
toolset with a central FCA component that we use to index the papers with a
thesaurus containing terms related to FCA research and to generate the
lattices. We zoom in on and give an extensive overview of the papers
published between 2003 and 2009 on using FCA in knowledge discovery and
data mining, information retrieval, ontology engineering and scalability. We
also give an overview of the literature on FCA extensions such as fuzzy FCA,
rough FCA, temporal and triadic concept analysis.

The remainder of this chapter is composed as follows. In section 2.2 we
introduce the essentials of FCA theory and the knowledge browsing
environment we developed to support this literature analysis. In section 2.3
we describe the dataset used. In section 2.4 we visualize the FCA literature
on knowledge discovery, information retrieval, ontology engineering and
scalability using FCA lattices and we summarize the papers published in
these fields. In section 2.5, we repeat this exercise for the extensions of FCA
such as fuzzy FCA, rough FCA, AFS algebra, temporal and triadic concept
analysis. In section 2.6, we give a lattice overview of the mathematical
research and algorithmic innovations in FCA theory. Section 2.7 concludes
the chapter.

2.2. Formal Concept Analysis

2.2.1. FCA essentials

Formal Concept Analysis is a recent mathematical technique that can be used
as an unsupervised clustering technique (Ganter et al. 1999, Wille 1982).
Scientific papers containing terms from the same term-clusters are grouped in
concepts. The starting point of the analysis is a database table consisting of
rows $M$ (i.e. objects), columns $F$ (i.e. attributes) and crosses $T \subseteq M \times F$
(i.e. relationships between objects and attributes). The mathematical structure
used to reference such a cross table is called a formal context $(T, M, F)$. An
example of a cross table is displayed in Table 2.1. In the latter, scientific papers (i.e. the objects) are related (i.e. the crosses) to a number of terms (i.e. the attributes); here a paper is related to a term if the title or abstract of the paper contains this term. The dataset in Table 2.1 is an excerpt of the one we used in our research. Given a formal context, FCA then derives all concepts from this context and orders them according to a subconcept-superconcept relation. This results in a line diagram (a.k.a. lattice).

<table>
<thead>
<tr>
<th></th>
<th>browsing</th>
<th>mining</th>
<th>software</th>
<th>web services</th>
<th>FCA</th>
<th>information retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper 1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Paper 2</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Paper 3</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Paper 4</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Paper 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

The notion of concept is central to FCA. The way FCA looks at concepts is in line with the international standard ISO 704, that formulates the following definition: A concept is considered to be a unit of thought constituted of two parts: its extension and its intension (Ganter et al. 1999, Wille 1982). The extension consists of all objects belonging to the concept, while the intension comprises all attributes shared by those objects. Let us illustrate the notion of concept of a formal context using the data in Table 2.1. For a set of objects \( O \subseteq M \), the common features can be identified, written \( \sigma(O) \), via:

\[ A = \sigma(O) = \{ f \in F \mid \forall o \in O : (o, f) \in T \} \]

Take the attributes that describe paper 4 in Table 2.1, for example. By collecting all reports of this context that share these attributes, we get to a set \( O \subseteq M \) consisting of papers 1 and 4. This set \( O \) of objects is closely connected to set \( A \) consisting of the attributes “browsing”, “software” and “FCA.”

\[ O = \tau(A) = \{ i \in M \mid \forall f \in A : (i, f) \in T \} \]

That is, \( O \) is the set of all objects sharing all attributes of \( A \), and \( A \) is the set of all attributes that are valid descriptions for all the objects contained in \( O \). Each such pair \( (O, A) \) is called a formal concept (or concept) of the given context. The set \( A = \sigma(O) \) is called the intent, while \( O = \tau(A) \) is called the extent of the concept \( (O, A) \).

There is a natural hierarchical ordering relation between the concepts of a given context that is called the subconcept-superconcept relation.
A concept \( d = (O_1, A_1) \) is called a subconcept of a concept \( e = (O_2, A_2) \) (or equivalently, \( e \) is called a superconcept of a concept \( d \)) if the extent of \( d \) is a subset of the extent of \( e \) (or equivalently, if the intent of \( d \) is a superset of the intent of \( e \)). For example, the concept with intent “browsing,” “software,” “mining” and “FCA” is a subconcept of a concept with intent “browsing”, “software” and “FCA.” With reference to Table 2.1, the extent of the latter is composed of papers 1 and 4, while the extent of the former is composed of paper 1.

The set of all concepts of a formal context combined with the subconcept-superconcept relation defined for these concepts gives rise to the mathematical structure of a complete lattice, called the concept lattice of the context. The latter is made accessible to human reasoning by using the representation of a (labeled) line diagram. The line diagram in Figure 2.1, for example, is a compact representation of the concept lattice of the formal context abstracted from Table 2.1. The circles or nodes in this line diagram represent the formal concepts. It displays only concepts that describe objects and is therefore a subpart of the concept lattice. The shaded boxes (upward) linked to a node represent the attributes used to name the concept. The non-shaded boxes (downward) linked to the node represent the objects used to name the concept. The information contained in the formal context of Table 2.1 can be distilled from the line diagram in Figure 2.1 by applying the following reading rule: An object “\( g \)” is described by an attribute “\( m \)” if and only if there is an ascending path from the node named by “\( g \)” to the node named by “\( m \).” For example, paper 1 is described by the attributes “browsing”, “software”, “mining” and “FCA.”
Retrieving the extension of a formal concept from a line diagram such as the one in Figure 2.1 implies collecting all objects on all paths leading down from the corresponding node. To retrieve the intension of a formal concept one traces all paths leading up from the corresponding node in order to collect all attributes. The top and bottom concepts in the lattice are special. The top concept contains all objects in its extension. The bottom concept contains all attributes in its intension. A concept is a subconcept of all concepts that can be reached by travelling upward. This concept will inherit all attributes associated with these superconcepts.

2.2.2. FCA software

We developed a knowledge browsing environment to support our literature analysis process. One of the central components of our text analysis environment is the thesaurus containing the collection of terms describing the different research topics. The initial thesaurus was constructed based on expert prior knowledge and was incrementally improved by analyzing the concept gaps and anomalies in the resulting lattices. The thesaurus is a layered thesaurus containing multiple abstraction levels. The first and finest level of granularity contains the search terms of which most are grouped
2.2. **FORMAL CONCEPT ANALYSIS**

2.2.1. **Term Clusters**

Together based on their semantical meaning to form the term clusters at the second level of granularity.

An excerpt of this thesaurus is shown in Appendix A, which shows amongst others the term cluster “Knowledge discovery”. This term cluster contains search terms “data mining”, “KDD”, “data exploration”, etc. which can be used to automatically detect the presence or absence of the “Knowledge discovery” concept in the papers. Each of these search terms were thoroughly analysed for being sufficiently specific. For example, we first had the search term “exploration” for referring to the “Knowledge Discovery” concept, however when we used this term we found that it also referred to concepts such as “attribute exploration” etc. Therefore we only used the specific variant such as “data exploration”, which always refers to the “Knowledge Discovery” concept. We aimed at composing term clusters that are complete, i.e. we searched for all terms typically referring to for example the “information retrieval” concept. Both specificity and completeness of search terms and term clusters was analysed and validated with FCA lattices on our dataset. We only used abstract, title and keywords because the full text of the paper may mention a number of concepts that are irrelevant to the paper. For example, if the author who wrote an article on information retrieval gives an overview of related work mentioning papers on fuzzy FCA, rough FCA, etc., these concepts may be irrelevant however they are detected in the paper. If they are relevant to the entire paper we found they were typically also mentioned in title, abstract or keywords.

The papers that were downloaded from the World Wide Web (WWW) were all formatted in pdf. These pdf-files were converted to ordinary text and the abstract, title and keywords were extracted. The open source tool Lucene was used to index the extracted parts of the papers using the thesaurus. The result was a cross table describing the relationships between the papers and the term clusters or research topics from the thesaurus. This cross table was used as a basis to generate the lattices.

2.2.3. **Web portal**

We plan to host these 700 papers and the lattices to browse them on the internet. The concept lattices are expanded with hyperlinks to allow easy access to the papers. The user will be able to dynamically compose the lattices with his topics of interest.

2.3. **Dataset**

This Systematic Literature Review (SLR) has been carried out by considering a total of 702 papers related to FCA published between 2003 and 2009 in the literature and extracted from the most relevant scientific sources. The sources...
that were used in the search for primary studies contain the work published in those journals, conferences and workshops which are of recognized quality within the research community. These sources are:

- IEEE Computer Society
- ACM Digital Library
- Sciencedirect
- Springerlink
- EBSCOhost
- Google Scholar
- Conference repositories: ICFCA, ICCS and CLS conference

Other important sources such as DBLP or CiteSeer were not explicitly included since they were indexed by some of the mentioned sources (e.g. Google Scholar). In the selected sources we used various search strings including “Formal Concept Analysis”, “FCA”, “concept lattices”, “Temporal Concept Analysis”. To identify the major categories for the literature survey we also took into account the number of citations of the FCA papers at CiteseerX.

Perhaps the major validity issue facing this systematic literature review is whether we have failed to find all the relevant primary studies, although the scope of conferences and journals covered by the review is sufficiently wide for us to have achieved completeness in the field studied. Nevertheless, we are conscious that it is impossible to achieve total completeness in the field studied. Some relevant studies may exist which have not been included, although the width of the review and our knowledge of this subject have led us to the conclusion that, if they do exist, there are probably not many. We also ensured that papers that appeared in multiple sources were only taken into account once, i.e. duplicate papers were removed.

2.4. **Studying the literature using FCA**

The 702 papers are grouped together according to a number of features within the scope of FCA research. We visualized the papers using FCA lattices, which facilitate our exploration and analysis of the literature. The lattice in Figure 2.2 contains 7 categories under which 55% of the 702 FCA papers can be categorized. Knowledge discovery is the most popular research theme covering 20% of the papers and will be analyzed in detail in section 2.4.1. Recently, improving the scalability of FCA to larger and complex datasets emerged as a new research topic covering 5% of the 702 FCA papers. In particular, we note that almost half of the papers dedicated to this topic work on issues in the KDD domain. Scalability will be discussed in detail in section 2.4.3. Another important research topic in the FCA community is information retrieval covering 15% of the papers. 25 of the
papers on information retrieval describe a combination with KDD approach and in 20 IR papers authors make use of ontologies. 15 IR papers deal with the retrieval of software structures such as software components. The FCA paper on information retrieval will be discussed in detail in section 2.4.2. In 13% of the FCA papers, FCA is used in combination with ontologies or for ontology engineering. FCA research on ontology engineering will be discussed in section 2.4.4. Other important topics are using FCA in software engineering (15%) and for classification (7%).

Fig. 2.2. Lattice containing 702 papers on FCA

2.4.1. Knowledge discovery and data mining

Knowledge discovery and data mining (KDD) is an interdisciplinary research area focusing upon methodologies for extracting useful knowledge from data. In the past, the focus was on developing fully automated tools and techniques that extract new knowledge from data. Unfortunately, these techniques allowed almost no interaction between the human actor and the tool and failed at incorporating valuable expert knowledge into the discovery process.
(Keim 2002), which is needed to go beyond uncovering the fool's gold. These techniques assume a clear definition of the concepts available in the underlying data which is often not the case. Visual data exploration (Eidenberger 2004) and visual analytics (Thomas et al. 2005) are especially useful when little is known about the data and exploration goals are vague. Since the user is directly involved in the exploration process, shifting and adjusting the exploration goals is automatically done if necessary.

In Conceptual Knowledge Processing (CKP) the focus lies on developing methods for processing information and knowledge which stimulate conscious reflection, discursive argumentation and human communication (Wille 2006). The word “conceptual” underlines the constitutive role of the thinking, arguing and communicating human being and the term “processing” refers to the process in which something is gained which may be knowledge. An important subfield of CKP is Conceptual Knowledge Discovery (Stumme 2003). FCA is particularly suited for exploratory data analysis because of its human-centeredness (Hereth et al. 2003). The generation of knowledge is promoted by the FCA representation that makes the inherent logical structure of the information transparent. The philosophical and mathematical origins of using FCA for knowledge discovery have been briefly summarized in Priss (2006). The system TOSCANA has been used as a knowledge discovery tool in various research and commercial projects (Stumme et al. 1998). In chapter 3 and 5 of this thesis we will zoom in on our applications of FCA for kDD.
2.4. STUDYING THE LITERATURE USING FCA

Fig. 2.3. Lattice containing 140 papers on using FCA in KDD

About 74% of the FCA papers on KDD are covered by the research topics in Figure 2.3. The main goal of this lattice is to visualize the main research themes. The remaining 26% of the papers were of diverse nature, often involving applications and in section 2.4.1.1 we first discuss some of these applications of FCA in KDD. In section 2.4.1.2 we zoom in on the 35 papers (25%) in the field of association rule mining. 19% of the KDD papers focus on using FCA in the discovery of structures in software and are discussed in section 2.4.1.3. Section 2.4.1.4 describes the 9% of papers on applications of FCA in web mining. Section 2.4.1.5 discusses some of the extensions of FCA theory for knowledge discovery (11% of papers). In section 2.4.1.6 we describe some of the applications of FCA in biology, chemistry and medicine covering 10% of the KDD papers. In section 2.4.1.7 the relation of FCA to some standard machine learning techniques is investigated (about 4% of papers). The applications on using Fuzzy FCA for KDD, covering 9% of the papers, will be discussed in section 2.4.1.8.
2.4.1.1. Some FCA mining applications

FCA has been used in several application domains and in many different ways for knowledge discovery. Amongst others Silva et al. (2006) propose FCA as an alternative step in the KDD process for data representation and analysis. In Colton et al. (2007), FCA is used in combination with mathematical discovery tools to better facilitate mathematical discovery. Missaoui et al. (2009) investigate the possibilities of using FCA for extracting valuable and actionable knowledge from data warehouses. In Hauff et al. (2007) concept lattices are used for disjoint clustering of transactional databases and several heuristics are developed to tune the support parameters used in the algorithm. The algorithm is applied to location learning to estimate the location of an electronic tag (in an RFID for example) given the signal strengths that can be heard. Ducrou et al. (2005a) developed a conceptual information system to determine surfing conditions on the South Coast of New South Wales in Australia. Maille et al. (2005) use FCA to help experts mine a database containing anecdotal reports of aviation incidents without prior knowledge of the researched concepts. Their FCA tool Kontex helped in bringing original ideas to the aviation safety community. Solesvik et al. (2009) use FCA as a quantitative instrument for partner selection in the context of collaborative ship design. Busch et al. (2004) use FCA for the analysis of psychological data. The authors developed a tacit knowledge inventory based on the measurement of responses to IT work-place scenarios, which is part of a questionnaire given to experts and non-experts in three IT organizations. Using FCA they were able to identify important groups of individuals that responded similarly to the peer-identified experts and the organisation was alerted of the important role these individuals potentially play. In Poelmans et al. (2009), FCA is used in combination with Emergent Self Organising Maps for detecting domestic violence in the unstructured text of police reports. Poelmans et al. (2010c) gives an overview of all their applications of FCA in law enforcement case studies. Girault (2008) presents an unsupervised method for named entity annotation based on concept lattice mining, where FCA is used to analyze the relations between named entities and their syntactic dependencies observed in a training corpus.

2.4.1.2. Association rule mining

Association rule mining is the main topic of 25% of the papers on using FCA for KDD. Association rule mining from a transaction database requires the detection of frequently occurring patterns called frequent itemsets (FI). Recent approaches for FI mining use the closed itemset paradigm to limit the mining effort to the subset of frequent closed itemsets (FCIs). The intent of a concept C is called a closed itemset and consists of the maximum set of
attributes that characterizes $C$. Several FCA-based algorithms were
developed for mining frequent closed itemsets including CLOSE, PASCAL,
CHARM, CLOSET and TITANIC (Stumme 2002) which mines frequent
closed itemsets by constructing an iceberg concept lattice. In Qi et al. (2004)
an algorithm Closearcher is proposed based on FCA for mining frequent
patterns.

The minimal generators for a concept $C$ are the minimal subsets of $C$’s
intent which can similarly characterize $C$. Nehmé et al. (2005) proposes a
novel method for computing the minimal generator family. Tekaya et al.
(2005) propose an algorithm called GenAll to build an FCA lattice in which
each concept is decorated by its minimal generators with the aim to derive
generic bases of association rules. Generic bases constitute reduced sets of
association rules and preserve the most relevant rules without loss of
information. The GenAll algorithm further improves on the algorithm
presented by Nourine et al. (1999). In Hamrouni et al. (2005b), the extraction
of reduced size generic bases of association rules is discussed to decrease the
overwhelming number of association rules resulting from ARM. Hamrouni et
al. (2005a) proposes an algorithm called PRINCE which builds a minimal
generator lattice from which the derivation of the generic association rules
becomes straightforward. Dong et al. (2005) introduce the Succinct System
of Minimal Generators (SSMGS) as a minimal representation of the minimal
generators of all concepts, and provide an efficient algorithm for mining
SSMGS. The SSMGS are also used for losslessly reducing the size of the
representation of all minimal generators. Hamrouni et al. (2007) present a
new sparseness measure for formal contexts using the framework of SSMGS.
Their measure is an aggregation of two complementary measures, namely the
succinctness and compactness measures of each equivalence class, induced
by the closure operator. This is important for the performance of frequent
closed itemset mining algorithms which is closely dependent on the type of
handled extraction context, i.e. sparse or dense. Hermann et al. (2008)
investigate the computational complexity of some of the problems related to
generators of closed itemsets. The authors also present an incremental
polynomial time algorithm that can be used for computing all minimal
generators of an implication-closed set. In Yahia et al. (2004), inference
axioms are provided for deriving all association rules from generic bases.

Valtchev et al. (2004), discuss the existing FCA-based data association
rule mining techniques and provide guidelines for the design of novel ones to
be able to apply FCA in a larger set of situations. They also propose two
online methods for computing the minimal generators of a closure system.
Gupta et al. (2005) discuss how classification rules based on association rules
can be generated using concept lattices. Valtchev et al. (2008) show how
FCIs can be mined incrementally yet efficiently whenever a new transaction
is added to a database whose mining results are available. In Quan et al.
(2009), a new cluster-based method is proposed for mining conceptual
association rules. Maddouri (2005) discusses the discovery of association
rules and proposes a new approach to mine interesting itemsets as the optimal concepts covering a binary table. Maddouri et al. (2006) summarize many of the statistical measures introduced for selecting pertinent formal concepts. Meddouri et al. (2009) present a method for building only a part of the lattice including the best concepts, which are used as classification rules.

Wollbold et al. (2008) make use of FCA to construct a knowledge base consisting of a set of rules such that reasoning over temporal dependencies within gene regulatory networks is possible. Zhou et al. (2005) use FCA to mine association rules from web logs, which can be used for online applications such as web recommendation and personalization. Richards et al. (2003b) explore the possibilities of using FCA for mining knowledge and reorganizing this knowledge into an abstraction hierarchy and to discover higher-level concepts in the knowledge. Richards et al. (2003a) discuss the discovery of multi-level knowledge from rule bases which is important to allow queries at and across different levels of abstraction. FCA is used to develop an abstraction hierarchy and the approach is applied to knowledge bases from the domain of chemical pathology. In Zarate et al. (2009), FCA is used to extract and represent knowledge in the form of a non-redundant canonical rule base with minimal implications from a trained Artificial Neural Network (ANN).

2.4.1.3. Software mining

19% of the 140 KDD papers are related to software mining and describe how FCA can be used to gain insight in amongst others software source code. In Cole et al. (2005), FCA is used to conceptually analyse relational structures in software source code and to detect unnecessary dependencies between software parts. In Cellier et al. (2008), FCA is used in combination with association rules for fault localization in software source code. Wermelinger et al. (2009) uses FCA lattices to visualize the relations between the software artifacts and the developers who should fix the bugs in them. In Eisenbarth et al. (2003), a technique is presented for reconstructing the mapping of features that are triggered by the user to the source code of the system. Mens et al. (2005) use FCA to delve a system's source code for relevant concepts of interest: which concerns are addressed in the code, which patterns, coding idioms and conventions were adopted and where and how are they implemented.

Crosscutting concerns, i.e. functionalities that are not assigned to a single modular unit in the implementation, is one of the major problems in software evolution. Aspect Oriented Programming offers mechanisms to factor them out into a modular unit, called an aspect. In Tonella et al. (2004), aspect identification in existing code is supported by means of dynamic code analysis. Execution traces are generated for the use cases that exercise the main functionalities of a given application. The relationship between execution traces and executed computational units is subjected to concept
2.4. STUDYING THE LITERATURE USING FCA

In the resulting lattice, potential aspects are detected. Su et al. (2008) discuss an aspect-mining approach in which execution profiles of legacy systems are analyzed using concept lattices to identify the invoked computational units that traverse system's use case models. They can be abstracted into early-aspects for re-engineering of the legacy system with Aspect Oriented System Design (AOSD). Qu et al. (2007) also discuss the use of FCA for aspect mining to identify crosscutting concerns in a system, thereby improving the system's comprehensibility and enabling migration of existing (object-oriented) programs to aspect-oriented ones. Breu et al. (2006) mined aspects from Eclipse by analyzing where developers added code to the program over time. In Del Grosso et al. (2007), an approach is proposed to identify pieces of functionality to be potentially exported as services from database-oriented applications.

Role Based Access Control (RBAC) is a methodology for providing users in an IT system with specific permissions like read or write. Molloy et al. (2008) use FCA for mining roles from user-permission and user-attribute information to complement the costly top-down approaches for RBAC. Dau et al. (2009) apply FCA in combination with Description Logics to capture the RBAC constraints and for deriving additional constraints.

2.4.1.4. Web mining

Web mining and improving the quality of web search results is investigated in 8% of the KDD papers. Periodic web personalization aims at recommending the most relevant resources to a user during a specific time period by analyzing the periodic access patterns of the user from web usage logs. Cho et al. (2004), use FCA for improving search performance for domain-specific area users by analyzing the keywords and web pages that have been previously used or visited by other users. Beydoun et al. (2007) introduce a system which captures user trails as they search the internet. They construct a semantic web structure from the trails and this semantic web structure is expressed as a conceptual lattice guiding future searches. Beydoun (2008) further investigates the possibilities of FCA for processing students virtual surfing trails to express and exploit the dependencies between visited web-pages to yield subsequent and more effective focused search results. Huang et al. (2008) present a method for web page recommendation where user’s web surfing trails are traced and useful information is extracted. An FCA and ontology knowledge base are automatically generated from this information and while users are searching, the system reasons based on these two knowledge bases and recommends the most relevant web pages to users. He et al. (2007) also propose a method for automatically mining and acquiring web user profiles using FCA. Okubo et al. (2006) show how FCA can be used for the conceptual clustering of web documents and in providing a conceptual meaning for each document cluster. Myat et al. (2005) use FCA for conceptual document clustering to manage the information published on
Du et al. (2009) present a method based on FCA for mining association rules that can be used to match user queries with web pages to avoid returning irrelevant web pages for search engine results. Hsieh et al. (2007) propose a knowledge acquisition system which dynamically constructs the relationships and hierarchy of concepts in a query-based ontology to provide answers for user’s queries. Kim et al. (2007b) discuss a novel approach using FCA to build a contextualized folksonomy and concept hierarchies from tags of blogosphere.

2.4.1.5. Extending FCA for data mining

In the last years, multiple extensions have been introduced into the literature that improve traditional FCA theory’s applicability to knowledge discovery problems. Belohlavek et al. (2009) emphasize the need for taking into account background knowledge in FCA. They present an approach for modeling background knowledge that represents user’s priorities regarding attributes and their relative importance. Only those concepts that are compatible with user’s priorities are considered relevant and are extracted from the data. In Pogel et al. (2008), FCA is used in combination with a tag context to formally incorporate important kinds of background knowledge. The results are Generalized Contingency Structures and Tagged Contingency structures which can be used for data summarization in epidemiology.

In Besson et al. (2006), FCA is extended to cope with faults and to improve formal concepts towards fault tolerance. Pfältz (2007) extends FCA to deal with numerical values. Valverde-Albacete et al. (2006) introduced a generalization of FCA for data mining applications called K-Formal Concept Analysis. This idea was further developed in Valverde-Albacete et al. (2007) where the lattice structure for such generalized contexts was introduced. This research topic was further investigated in Valverde-Albacete et al. (2008). Hashemi et al. (2004) propose a method for efficiently creating a new lattice from an already existing one when the data granularity is changed. Lei et al. (2007) introduces the notion of extended many-valued context to avoid the generation of a large one-valued context in document knowledge discovery. In Deogun et al. (2003) FCA is complemented with Bacchus probability logic, which makes use of statistical and propositional probability inference. The authors introduce a new type of concept called “previously unknown and potentially useful” and formalize KDD as a process to find such concepts. In its classical form FCA considers attributes as a non-ordered set. When attributes of the context are partially ordered to form a taxonomy, conceptual scaling allows the taxonomy to be taken into account by producing a context completed with all attributes deduced from the taxonomy. In Cellier et al. (2008a) an algorithm is proposed to learn concept-based rules in the presence of a taxonomy.
Another FCA research topic is attribute reduction. Shao et al. (2008) show how to remove redundant attributes from real set formal contexts without any loss of knowledge. Wu et al. (2009) discuss the application of viewing data at different levels of granularity to construct a granular data structure which can be used for knowledge reduction in FCA. Wang et al. (2008a) deal with approaches to generalized attribute reduction in a consistent decision formal context. Ganter et al. (2008b) describe how scaled many-valued contexts of FCA may make feature selection easier.

2.4.1.6. FCA mining applications in biology and medicine

10% of the KDD papers describe applications of FCA in biology, chemistry or medicine. In Sato et al. (2007), FCA is used to cluster time-series medical data and to analyze these clusters. Sklenar et al. (2005) used FCA to evaluate epidemiological questionnaire physical activity data to find dependencies between demographic data and degree of physical activity. In Kaytoue et al. (2009), FCA is used for mining and clustering gene expression data. Fu (2006) applies FCA as a tool for analysis and visualization of data in a digital ecosystem. Maddouri (2004) outlines a new incremental learning approach based on FCA that supports incremental concept formation and applies it to the problem of cancer diagnosis. The incremental approach has the advantage of handling the problem of data addition, data deletion, data update, etc.

2.4.1.7. FCA in relation to other machine learning techniques

4% of the KDD papers investigate the relationships between FCA and other machine learning techniques. In Kuznetsov (2004), it is shown how FCA can be used to give natural descriptions of some standard models of machine learning such as version spaces and decision trees. The author also shows how a concept-based learning model from positive and negative examples is described in FCA. Ganter et al. (2003) consider the relation of a learning model described in terms of FCA with a standard model of machine learning called version spaces. The authors also discuss an example from predictive toxicology. In Ganter et al. (2004) labeled graphs from chemistry are reduced to object-attribute representation and FCA is used to generate hypotheses about biological activity of chemical compounds. In Shi et al. (2003), a hybrid conceptual structure based on FCA, conceptual graphs and structured concepts is proposed, which has been developed for knowledge discovery in heterogeneous data sources. Kiu et al. (2008) present a hybrid unsupervised clustering model based on FCA, SOM and K-means clustering for managing ontological knowledge and lexical matching for retrieving knowledge. Their method can be used to facilitate ontological concept visualization and navigation in concept lattice form.
2.4.1.8. Fuzzy FCA in KDD

In fuzzy FCA, each table entry contains a truth degree to which an attribute applies to an object. 9% of the papers use fuzzy FCA for KDD. In Chou et al. (2008), fuzzy FCA is used for tag mining, i.e. to analyze the relationships between semantic tags and Web APIs. In Fenza et al. (2008), fuzzy FCA is used for the discovery of semantic web services. Zhou et al. (2006) use fuzzy FCA to construct a user behavior model from web usage logs to identify the resources that the user is most likely interested in during a given period. Fenza et al. (2009) present a system which uses fuzzy FCA for supporting the user in the discovery of semantic web services. Through a concept-based navigation mechanism, the user discovers conceptual terminology associated to the web resources and may use it to generate an appropriate service request. Yan et al. (2007) use fuzzy set theory to extend the many-valued context from FCA. This fuzzy many-valued context can then be used for document knowledge discovery.

2.4.2. Information retrieval

According to Manning et al. (2008), information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). Information retrieval used to be an activity that only a few people engaged in: librarians, paralegals and similar professional searchers. The world has changed and hundreds of millions of people engage in information retrieval these days when they use a web search engine or search their email. Information retrieval systems can be distinguished by the scale at which they operate, and it is useful to distinguish three prominent scales. In web search, the system has to provide search over billions of documents stored on millions of computers. At the other extreme is personal information retrieval In the last few years, consumer operating systems have integrated information retrieval Email programs usually not only provide search but also text classification. In between is the space of enterprise institutional, and domain-specific search, where retrieval might be provided for collections such as a corporation's internal documents, a database of patents, etc.

The field of information retrieval also covers supporting users in browsing or filtering document collections or further processing a set of retrieved documents. Given a set of documents, clustering is the task of coming up with a good grouping of the documents based on their contents. Given a set of topics, standing information needs, or other categories, classification is the task of deciding which classes, each of a set of documents belong to.
The first attempts to use lattices for information retrieval are summarized in Priss (2000), but none of them resulted in practical implementations. Godin et al. (1989) developed a textual information retrieval system based on document-term lattices but without graphical representations of the lattices. The authors also compared the system’s performance to that of Boolean queries and found that it was similar to and even better than hierarchical classification (Godin et al. 1993). They also worked on software component retrieval (Mili et al. 1997). In Carpineto et al. (2004a), their work on information retrieval was summarized. They argue that FCA can serve three purposes. First, FCA can support query refinement. Because a document-term lattice subdivides a search space into clusters of related documents, lattices can be used to make suggestions for query enlargement in cases where too few documents are retrieved and for query refinement in cases where too many documents are retrieved. Second, lattices can support an integration of querying and navigation (or browsing). An initial query identifies a start node in a document-term lattice. Users can then navigate to related nodes. Further, queries are then used to “prune” a document-term lattice to help users focus their search (Carpineto et al. 1996b). Third, a thesaurus hierarchy can be integrated with a concept lattice, an idea which was independently discussed by different researchers (e.g. Carpineto et al. 1996b, Skorsky 1997, Priss 1997). For many purposes, some extra facilities are needed: process large document collections quickly, allow more flexible matching operations and allow ranked retrieval. In particular, the research on query tuning was of interest to us because the terms we use to annotate documents were applied as query instruments that were sometimes too specific and sometimes too broadly defined. Chapter 3 zooms in on the knowledge browsing capabilities of FCA and the incremental improvement of query search terms.
Fig. 2.4. Lattice containing 103 papers on using FCA in IR

86% of the papers on FCA and information retrieval are covered by the research topics in Figure 2.4. In section 2.4.2.1, the 28% of papers on using FCA for representation of and navigation in document collections are described. The IR systems that were developed based on FCA cover 10% of the papers and are discussed in section 2.4.2.2. Query tuning and query result improvement covers 8% of the papers and are discussed in section 2.4.2.3. Defining and processing complex queries covers 6% of the papers and is described in section 2.4.2.4. Section 2.4.2.5 summarizes the papers on contextual answers (6% of papers) and ranking of query results (6% of papers). The 9% of papers on fuzzy FCA in IR are described in section 2.4.2.6.
2.4.2.1. Knowledge representation and browsing with FCA

Many approaches have been proposed to use FCA for navigation through document collections. In 28% of the IR papers, FCA is used for browsing and navigation through data. In more than half of these papers (18% of total number of papers), a combination of navigation and querying based on the FCA lattices is proposed. Annotation of documents and finding optimal document descriptors plays an important role in effective information retrieval (9% of papers).

Kim et al. (2004) proposed a framework for an FCA-based document, management and retrieval system aimed at small communities in specialized domains. Relevant documents can be annotated with any terms the users or authors prefer. The search mechanism consists of browsing an FCA lattice based on the keywords with which the users annotated the documents. In Kim et al. (2006), the search mechanism was extended by combining the lattice-based browsing structure with conceptual scales. This allows users to obtain more specific search results and reduces the complexity of the visualization of the browsing structure. Stojanovic (2004) presents an approach for efficient navigation through an on-line product catalog. The product database is based on an ontology and is visualized with a lattice, in which users can navigate from a very general product-attribute cluster containing a lot of products to the very specific clusters that seem to contain a few, but for the user highly relevant products. Shi et al. (2008) designed a web-based digital assets navigating system based on FCA. Ducrou et al. (2005b) presented an FCA-based application, D-SIFT, for exploring relational database schema. Ahmad et al. (2003) use FCA to catalogue descriptions associated with image contents for efficient search and retrieval of images from a database. Amato et al. (2008) use FCA lattices to explore a lattice of image clusters and to search for images that are similar to a given image. The exploration proceeds in one of two basic ways: by querying, the user can jump to any cluster of the lattice by specifying the criteria that the sought cluster must satisfy; by navigation: from any cluster, the user can move to a neighbor cluster, thus exploiting the ordering amongst clusters. Tane et al. (2005) introduced the query-based multi context theory, which allows to define a virtual space of FCA-based views on ontological data. Tane et al. (2006) discuss the benefits of the browsing framework for knowledge bases based on supporting the user in defining pertinent views. Eklund et al. (2007) describes Semantology as the theory of semantic structures for representing conceptual knowledge, which is the basis for conceptual knowledge processing.

Efficient service management is the main task for services execution and services composition and there is no good solution until now. Classification and semantic annotation of services are important challenges in service-centric software engineering. It is challenging to effectively manage and search the Web services satisfying the requestor’s requirements. Peng et al.
43 CHAPTER 2: FORMAL CONCEPT ANALYSIS IN INFORMATION ENGINEERING

(2005) present an FCA based method for generating a concept lattice describing the relationships between web services and to accurately retrieving web services from the lattices. Semantic annotation of web pages with domain ontologies facilitates semantic retrieval over the web. Garcia et al. (2006) use amongst others FCA to perform this annotation process. Similarity matching techniques from Case Based Reasoning can be applied to retrieve these annotated pages as cases. Bruno et al. (2005) propose an approach based on FCA and SVMs to automatically classify services to specific domains and identify key concepts inside service textual documentation to build a lattice of relationships between service annotations. Automatic attribute selection is important when using FCA in a free text document retrieval framework. Optimal attributes as document descriptors should produce smaller, clearer and more browsable concept lattices with better clustering features. Cigarran et al. (2005) focus on the automatic selection of noun phrases as documents descriptors to build an FCA based IR framework.

Another domain in which FCA has been applied as an information retrieval technique is software engineering. Poshyvanyk et al. (2007) use a combination of FCA and Latent Semantic Indexing (LSI) for concept location in the source code of Eclipse. LSI is used to map the concepts expressed in queries to relevant parts of the source code. The result is a ranked list of source code elements, organized in an FCA lattice. Muangon et al. (2009) combine FCA with Case Based Reasoning (CBR) for choosing appropriate design patterns for a specific design problem. This approach solves some of the problems of existing design pattern search methods using keyword-search. Design patterns are applied to solve recurring software design problems. Peng et al. (2007) propose a method for the incremental construction of a component retrieval ontology based on FCA. The ontology contains the characterizations of the components stored in the repository.

2.4.2.2. Knowledge browsing systems based on FCA

FCA has been used as the basis for many knowledge browsing systems developed during the past years. Especially its comprehensible visualization capabilities seem to be of interest to the authors of these papers. 10% of the IR papers describe a newly developed FCA-based retrieval system. Kim et al. (2004) presented an FCA-based document navigation system, KANavigator, which uses annotated documents that can be browsed by keyword and displays the direct neighborhood of the documents as its interface. Cigarran et al. (2004) present the JBraindead IR System which combines free-text search with FCA to organize the results of a query. In the ImageSleuth project (Ducrou et al. 2006), FCA was used for conceptual representation, navigation through and clustering of annotated collections of images. In Ducrou (2007), the author built an information space from the Amazon.com online store and used FCA to discover conceptually similar DVDs and
explore the information space of their conceptual neighborhood. The system was called DVDSleuth. Eklund et al. (2008) present AnnotationSleuth to extend a standard search and browsing interface to feature a conceptual neighborhood centered on a formal concept derived from curatorial tags in a museum management system.

Another FCA application is the Mail-Sleuth software (Eklund et al. 2004) which can be used to mine large email archives. The development of this software is based on earlier research on retrieval of information from semi-structured texts (Cole et al. 2001, Cole et al. 2000). Eklund et al. (2005) use FCA for displaying, searching and navigating through help content in a help system. Cole et al. (2003) discuss a document discovery tool based on FCA. The program allows users to navigate through emails using a visual lattice. The paper also discusses how conceptual ontologies can support traditional document retrieval systems and aid knowledge discovery in document collections.

Carpineto et al. (2004b) developed the CREDO system which allows for a lattice-based meta-search of Google results. FOOCA (Koester 2006) is also a web search tool and provides a line diagram representation of standard sequential web search engine retrieval results. Dau et al. (2008) developed SearchSleuth, a program for Web search query refinement. It extends a standard search interface to include a conceptual neighborhood surrounding the formal concept derived from the initial query. This neighborhood consists of more general, specialized and related concepts.

2.4.2.3. Query result improvement with FCA

Search engines are increasingly being used by amongst others web users who have an information need. Since a query provided by a user only approximates a user's need, many techniques have been developed to expand and refine query terms and search results. Query tuning is the process of searching for the query that best approximates the information need of the user. Query refinements can help the user express his original need more clearly. Query expansions help retrieve additional relevant web pages. These topics are investigated in 8 % of the IR papers.

Stojanovic (2003) present an FCA-based method for query refinement that provides a user with the queries that are “nearby” the given query. The authors analyze the ambiguities of that query with respect to the vocabulary used for querying and the information repository. Hai et al. (2006) and Zhang et al. (2008a) also present a new FCA-based approach for query expansion. Hao et al. (2009) propose a new method for expanding query terms based on a variant of FCA. Users can set the query precision in terms of their interests and obtain the additional relevant web pages. Spyrotos et al. (2006) describe an approach for query tuning that integrates navigation and querying into a single process. Typically, navigation and querying are two completely separate processes, and the combination of both results in a more flexible and
Shen et al. (2007) propose an FCA search engine for Chinese information which incorporates a concept lattice based browsing mechanism for the management and retrieval of documents. Langley et al. (2009) explore the use of implicit user feedback in adapting the underlying domain model of an intranet search system. FCA is used as an interactive interface for user exploration of the context of an intranet query. The authors present query refinements based on the lattices.

2.4.2.4. Defining and processing complex queries with FCA

Multiple techniques have been developed to define and process complex queries and to integrate data coming from heterogeneous sources. This topic is discussed in 6% of the IR papers. De Souza et al. (2004) use FCA for processing user queries over a set of overlapping ontologies, which is not always easy, since they have been created by independent groups adopting different configurations for ontology concepts. For example in bioinformatics, it is often difficult to relate the resources with a user query since the query needs to be processed and distributed over several heterogeneous data sources. Messai et al. (2005) present an approach based on FCA to search relevant bioinformatics data sources for a given user query. Nafkha et al. (2005b) investigate the possibilities of using FCA for searching in heterogeneous information sources, while Ceravolo et al. (2007) present a new method for integrating heterogeneous data sources in a single view. The data are retrieved through a common conceptualization, modeled as an ontology. The authors use FCA for the matching and mapping of elements between the common ontology and the heterogeneous data sources. Hitzler et al. (2006) present a new expressive query language which allows to query formal contexts by means of logic programs written over attributes and objects.

2.4.2.5. Domain knowledge incorporation in query search results: contextual answers & ranking

In this section, we discuss the techniques that have been devised to provide contextual answers to user’s queries and to incorporate domain knowledge into the organization of search results. Amongst others Carpineto et al. (2005) state that the main advantage of FCA for IR is the possibility of eliciting context. This topic is discussed in 9% of the IR papers.

One method used to improve the quality of search results is by analyzing the resources that have been accessed by users in the past. This topic has been covered in detail in section 2.4.1.4. Providing contextual answers to user’s queries is another research topic. Pollailon et al. (2007) present a method based on FCA to provide contextual answers to user’s queries from
2.4. STUDYING THE LITERATURE USING FCA

and to help their navigation in heterogeneous data sources. Le Grand et al. (2006) present an information retrieval method based on FCA in conjunction with semantics to provide contextual answers to web queries. An overall lattice is built from tourism web pages. Then, users formulate their query and the best-matching concepts are returned, users may then navigate within the lattice by generalizing or on the contrary by refining their query. Liu et al. (2007) use FCA to optimize a personal news search engine to help users obtain the news content they need rapidly. The proposed technique combines the construction of user background using FCA, the optimization of query keywords based on the user's background and a new layout strategy of search results based on “Concept Tree”.

Another technique which has received quite some attention in the literature is the ranking of query search results. Tang et al. (2007) propose a meta-search engine which invokes multiple search engines and identifies useful documents and ranks the returned results using FCA. Zhang et al. (2008) propose a method based on FCA to build a two-level hierarchy for retrieved search results of a query to facilitate browsing the collection. After formal concepts are extracted using FCA, the concepts most relevant to the query are further extracted. Similarity for measuring document relevance is an important field in IR. Many researchers use FCA lattices for measuring query-document relevance, i.e. concept lattice-based ranking (CLR). Jun et al. (2005) present a new method for CLR based on similarity between the query, a user profile based on FCA and the documents. Messai et al. (2008) partially order the set of attributes with respect to their importance. This hierarchy represents domain knowledge used to improve lattice-based querying and navigation. Hierarchies of attributes are used to define complex queries containing attributes with different levels of importance.

2.4.2.6. Fuzzy FCA in IR

In traditional information retrieval, queries have not taken into account imprecision and retrieve only elements which precisely match to the given boolean query. That is, an element belongs to the result if the query is true for this element; otherwise, no answers are returned to the user. In 9 % of the IR papers, authors make use of fuzzy FCA. Hachani et al. (2009) make use of FCA and fuzzy logic to interactively with the user explain the reasons of the failure of the query (i.e. no answer is returned to the user) and to propose the nearest answers. These answers which are in the neighborhood of the user's original query could be used to serve the user's need. Hoshino et al. (2008) propose a method based on FCA and fuzzy logic which permits the flexible modeling and querying of a database.

In Quan et al. (2004a), fuzzy FCA is used for conceptual clustering in a citation-database document retrieval system. Using fuzzy logic in combination with FCA, a fuzzy concept lattice is constructed on which a fuzzy conceptual clustering technique is performed. Fuzzy queries can then be performed for document retrieval. Chettaoui et al. (2008) use fuzzy FCA
to deal with empty answers for fuzzy queries. Fuzzy querying processing based on galois lattices allows to detect the reasons of empty answers by providing the subqueries that are responsible for the failure. Lattices are also used for answering the user with the nearest answers through fuzzy alternative sub queries.

2.4.3. Scalability

At the international Conference on Formal Concept Analysis in Dresden (ICFCA 2006) an open problem of “handling large contexts” was pointed out. Since then, several studies have focused on the scalability of FCA for efficiently handling large and complex datasets. Many techniques have been devised including nested line diagrams for zooming in and out of the data, conceptual scaling for transforming many-valued contexts into a single-valued context, iceberg lattices and pruning strategies to reduce the size of the concept lattice, etc. Scalability is an important issue for data mining analysts who work with police data. The amount of police reports in a dataset may vary from a few thousand reports to millions of documents. In chapter 3 and 5 we will give a description of our research on such large datasets and how we combined FCA with other instruments that helped in overcoming these hurdles.

![Fig. 2.5. Lattice containing 32 papers on FCA and scalability](image-url)
81% of the papers on FCA and scalability are covered by the research topics in Figure 2.5. 19% of these papers use iceberg lattices and are discussed in section 2.4.3.1. Section 2.4.3.2 describes the 22% papers on reducing the size of concept lattices. Section 2.4.3.3 gives an overview of some techniques for handling complex data. Section 2.4.3.4 discusses parallelization (9% of papers) and the combination with binary decision diagrams (6% of papers) and spatial indexing (16% of papers) for improving the scalability of FCA-based algorithms.

2.4.3.1. Iceberg lattices

When we want to apply FCA to formal contexts which contain hundreds of thousands of formal objects, iceberg lattices with up to a few thousand formal concepts come in handy. An iceberg lattice is a concept lattice of frequent closed itemsets and was initially introduced by Stumme et al. (2002a). It is a concept lattice where concepts which have fewer formal objects than a given threshold minimum are omitted. They have been used from time to time in knowledge discovery applications. For example in Jay et al. (2008a), an experiment is performed in which iceberg lattices are used to discover and represent flows of patients within a healthcare network. In Stumme (2004) it was investigated how iceberg query lattices can be used as condensed representations of frequent datalog queries. The algorithmic innovations for mining frequent closed itemsets are discussed in section 2.4.1.2. The extension of iceberg lattices to three dimensional data is discussed in section 2.5.4.

2.4.3.2. Reducing the size of concept lattices

In this section, we discuss the 22% of scalability papers on reducing the size of concept lattices. Kuznetsov (2007b) introduced stability (Kuznetsov 1990) as a new interest measure for concepts. Stability has been used for pruning concept lattices, e.g. in the field of social networks (Kuznetsov et al. 2007a, Roth et al. 2008a). In Jay et al. (2008b), concept stability and support measures were used to reduce the size of large concept lattices. These lattices were used to discover and represent special groups of individuals called social communities. Roth et al. (2008b) suggest a combined approach of using a pruning strategy based on stability indices of concepts and apply it on its own and in combination with nested line diagrams for representing knowledge communities (a community of embryologists was taken as a case study). Myat et al. (2005) described an approach on using FCA for document clustering and applying a pruning strategy to reduce less useful concepts. Ventos et al. (2005) introduced a new method for reducing the size of a Galois lattice resulting in a so called alpha Galois lattice. They also show that iceberg lattices are a special type of alpha Galois lattice. Torim et al. (2008)
present a method based on the theory of monotone systems, MONOCLE, for presenting information in the lattice in a more compressed form. The result of the method is a sequence of concepts sorted by “goodness” thus enabling the user to select a subset and build a corresponding sub-lattice of desired size. This is achieved by defining a weight function that is monotone. Snasel et al. (2008) study the reduction of concept lattices and implication bases based on matrix reduction and propose 2 kinds of reduction methods for these concept lattices.

2.4.3.3. Handling complex data

In practice, data is generally more complex than the simple attributes and binary relations of formal contexts. A first approach to handle complex data in FCA is conceptual scaling (Ganter 1989), which is a process that takes complex data as an input, and outputs a standard formal context, called the scaled context. Another approach was presented by Ferré (2009) who decomposed contexts into simpler and specialized components named logical context functors. Nested line diagrams is a technique that was initially introduced by Wille (1984). The line diagrams for practical applications typically get very large. Amongst others, TOSCANA uses nested line diagrams where a given context is split into parts, a line diagram is drawn for each part and then these line diagrams are nested into each other. The result is a simplified diagram. Applications of these techniques have been discussed throughout this paper.

2.4.3.4. FCA algorithm scalability

Several algorithms for computing maximal rectangles in binary matrices have been proposed but their major drawback is their computational complexity limiting their application to relatively small datasets. Multiple possible solutions are investigated in 31% of the scalability papers. One way to solve this problem is parallelizing these algorithms. Fu et al. (2004) propose a parallel algorithm based on the NextClosure algorithm to efficiently generate formal concepts for large datasets. Fu et al. (2007) use this new algorithm for analysis of large and high-dimensional biomedical data. Kengue et al. (2005) and Kengue et al. (2007) present a novel parallel divide-and-conquer algorithm to reduce the computational effort of computing concept lattices from large volumes of data. Kraja et al. (2009) introduce a scalable distributed algorithm for computing maximal rectangles in binary object-attribute relational data. They employ the map-reduce framework which is traditionally used for querying and searching in large data collections. Belohlavek et al. (2005) propose a method to reduce the number of extracted formal concepts by means of constraints expressed by attribute-dependency (ADF) formulas. ADF express the relative importance of attributes specified by a user. In Rimsa et al. (2009), an algorithmic innovation was introduced
based on Binary Decision Diagrams (BDDs) to obtain a symbolic representation of a cross table that allows a more efficient extraction of the set of all concepts. In Martin et al. (2006a), it was shown that spatial indexing structures offer faster resolution of FCA queries in Semantic File Systems. These spatial indexing techniques were further improved in Martin et al. (2006b) and Martin et al. (2007).

2.4.4. Ontologies

Ontologies were introduced as a means of formally representing knowledge. Their purpose is to model a shared understanding of the reality as perceived by some individuals in order to support knowledge intensive applications (Gruber 2009). An ontology typically consists of individuals or objects, classes, attributes, relations between individuals and classes or other individuals, function terms, rules, axioms, restrictions and events. The set of objects that can be represented is called the universe of discourse. The axioms are assertions in a logical form that together comprise the overall theory that the ontology describes in its domain of application. Ontologies are typically encoded using ontology languages, such as the Ontology Web Language (OWL). Whereas ontologies often use hierarchical representations for modeling the world, FCA has the benefit of a non-hierarchical partial order representation which has a larger expressive power (Christopher 1965). A key objective of the semantic web is to provide machine interpretable descriptions of web services so that other software agents can use them without having any prior “built-in” knowledge about how to invoke them. Ontologies play a prominent role in the semantic web where they provide semantic information for assisting communication among heterogeneous information repositories. An essential element of our work was desining a thesaurus which could be used to detect the early warning indicators in the unstructured text of police reports. How FCA was used for this purpose is described in detail in chapter 3.
Fig. 2.6. Lattice containing 93 papers on FCA and ontologies

84% of the FCA papers on ontologies are covered by the research topics in Figure 2.6. In section 2.4.4.1 we zoom in on the construction of ontologies using FCA, covering 28% of the 93 papers. 10% of the papers are about improving the quality of ontologies and are discussed in section 2.4.4.2. 6% of the papers describe linguistic applications of FCA and ontologies or the combination with natural language processing and are discussed in section 2.4.4.3. in section 2.4.4.4 we describe the 17% of papers on developing FCA-based similarity measures and using FCA in ontology mapping and merging. 14% of the papers use rough set theory or fuzzy theory in combination with FCA for ontology construction or merging and are summarized in section 2.4.4.5. The applications of FCA and ontologies in KDD and IR have been described in sections 2.4.1 and 2.4.2.

2.4.4.1. Ontology construction

An important topic in the FCA literature is how ontologies can be designed in an efficient manner. This topic is covered by 28% of the papers on FCA and ontologies. The unifying theme across ontology acquisition approaches is the considerable effort associated with developing, validating and connecting ontologies. Cimiano et al. (2004), discuss how FCA can be used to support ontology engineering and how ontologies can be exploited in FCA applications. Wang et al. (2006b) discuss how a domain ontology can be constructed and graphically represented using FCA. The FCA-based ontology is represented in Semantic Web Rule Language (SWRL). Xu et al. (2009) use FCA to build a computer network management information
specification ontology. Bao et al. (2005) present an iterative ontology building process using FCA for the construction of a pressure component design ontology. Chi et al. (2005) describe the construction of ontological knowledge bases for digital archive systems. The authors use FCA for concept extraction, Ontology Web Language (OWL) for presenting knowledge and description logic to define axioms of knowledge. Soon et al. (2004) use FCA for producing task-oriented ontologies. Verbs and nouns are extracted from a document that depicts user actions during a surface water monitoring process and FCA is used for formalization.

Richards (2004) proposes an approach based on FCA for engineering an ontology by retrospectively and automatically discovering the ontology concepts from existing data. Richards (2006) further uses FCA to automatically generate an ontology from a number of typical data sources including propositional rule bases, use cases, historical cases, text and web documents. Rudolph (2004) proposes an incremental method based on FCA which uses empirical data to systematically generate hypothetical axioms about the domain of interests, which are presented to an ontology engineer for decision. The author focused on the axioms that can be expressed as entailment statements in description logic. Hacene et al. (2008) propose an approach for semi-automated construction of ontologies using Relational Concept Analysis (RCA). Text analysis is used to transform a document collection into a set of data Table 2.s, or contexts and inter-context relations. RCA then turns these into a set of concept lattices with inter-related concepts. A core ontology is derived from the lattice in a semi-automated manner by translating relevant elements into ontological concepts and relations. Hwang et al. (2005) uses FCA for the construction of ontologies in the domain of software engineering. There are many conceptual similarities between the design of a class hierarchy in OO software design and an ontology. An OO software designer can design an ontology by organizing classes in a class hierarchy and creating relationships among classes. UML classes can then be generated from the ontology. Lim et al. (2005) present OntoVis, an ontological authoring and visualization tool making use of the clustering of concepts from FCA. The possibilities of using ontologies for supporting the visualization, navigation through and retrieval of information from databases were investigated. Fang et al. (2008) integrate FCA with Protégé to build an ontology-based knowledge sharing platform for patients and physicians.

2.4.4.2. Ontology quality management

Quality management is another important topic in ontology engineering and is covered by 10% of the papers. Rudolph (2008) proposes a notion of logical completeness as a novel quality criterion for ontologies. The author identifies a class of logical propositions which naturally extend domain and range restrictions commonly known from diverse ontology modeling approaches. FCA is used for the efficient interactive specification of all axioms of this
form valid in a domain of interest. Völker et al. (2008b) combine the LExO approach for learning expressive ontology axioms from textual definitions with Relational Exploration, a technique based on attribute exploration from FCA, to interactively clarify underspecified logical dependencies. The exploration guarantees completeness with respect to a certain logical fragment and increases the overall quality of the ontology. Kim et al. (2007a) use FCA for analyzing and resolving structural problems in OWL source code. Sertkaya (2009) describes OntoComp which supports ontology engineers in checking whether an OWL ontology contains all the relevant information about the application domain and in extending the ontology if this is not the case. Using FCA, it acquires complete knowledge about the application domain by asking successive questions to the ontology engineer. Bendaoud et al. (2008) propose an FCA-based system for semi-automatically enriching an initial ontology from a collection of texts in a given domain.

2.4.4.3. Linguistic applications using FCA and ontologies

In this section we discuss the 6% of papers on linguistic applications of FCA and ontologies. Priss et al. (2005) discuss how FCA makes the Semantic Mirrors Method (Tyrik 1998) simpler to understand. The Semantic Mirrors Method is a method for automatic derivation of thesaurus entries from a word-aligned parallel corpus. The method is based on the construction of lattices of linguistic features. The authors also use FCA for conceptual exploration of a medium quality bilingual dictionary. Gamallo et al. (2007) use FCA on an annotated corpus consisting of technical articles in the field of computational linguistics to create bi-dimensional clusters of terms and their lexico-syntactic contexts. Each cluster is a semantic class with a set of terms describing the extension of the class and a set of contexts as the intensional attributes valid for all the terms in the extension. The result is a concept lattice that describes a domain specific ontology of terms. Rudolph et al. (2007) combine natural language processing (NLP) with FCA for ontology creation and refinement. Nazri et al. (2008) investigate the possibilities of combining FCA with Malay NLP tools for automatic ontology building from Malay text. In Xu et al. (2006) FCA is used to build an event ontology from a set of documents and based on the event ontology, the authors explore various relevance measures to derive event relevance. Event-relevance is important in event-based document summarization, which attempts to select and organize the sentences in a summary with respect to the events that the sentences describe.

Chang et al. (2007) introduce a new document classification system based on a domain ontology which is automatically constructed using FCA. In Chang et al. (2008) an automatic document classifier system based on GenOntology and the Naive Bayes classifier is described. FCA is used for establishing the knowledge ontology and the Naive Bayes classifier is then applied to this ontology. In Pan et al. (2009) FCA was used to produce
different levels of ontological concepts from radiology reports. The authors compared the ontology concept lattice of the radiology report’s content before and after the adoption of the “picture archiving and communication system” and observed a delicate change in radiology report terms before and after the adoption of the system.

2.4.4.4. Ontology mapping and merging

Ontology mapping is a key technology to resolve interoperability issues between heterogeneous and distributed ontologies. An ontology in the same domain or overlapping field can be built with different representations including using different representations or different names for the same concept or different structures for the same domain. This topic is discussed in 17% of the papers on FCA and ontologies.

De Souza et al. (2004) use FCA for merging and aligning ontologies that cover overlapping domains. A central issue in the merging process is evaluating the differences between two ontologies, by using similarity measures to identify cross-ontology related concepts. In De Souza et al. (2006), the authors apply the alignment method to ontologies developed for completely different domains. Fan et al. (2007) propose an FCA-based method for ontology mapping, which can perform equal mapping by computing similarity measures between entities of different ontologies and can perform subclass mapping by computing inclusion measures. Formica (2006) discusses the role of FCA in ontology engineering and in particular, the author zooms in on the reuse of independently developed domain ontologies. The author proposes an ontology-based method for assessing similarity between FCA concepts, to support the ontology engineer in ontology merging and mapping, which is a fundamental activity for the Semantic Web. This work is further refined in Formica (2008) where the similarity measure is made independent of the domain expert knowledge. The author uses the information content approach (Resnik 1995) to automatically obtain attribute similarity scores.

Curé et al. (2008) presents an automated approach using FCA for creating a merged ontology which captures the knowledge of the two source ontologies. Wang et al. (2009) use FCA to compute the Concept-Concept similarity, the Concept-Ontology similarity and the Ontology-Ontology similarity for coordinating two Agent Crawlers and deducing the level of understanding between them, to guide them as parts of a search engine. Concept restriction conflict detection and elimination is another aspect of ontology merging. Two equivalent concepts in different source ontologies may have different definitions of value and cardinality restriction. Lu et al. (2008) present a Description Logic (DL)-based conflict detection and elimination approach. Le Grand et al. (2009) use FCA for complex systems analysis and compare different topic maps with each other both in terms of content and structure. Significant concepts and relationships can be identified
and this method can also be used to compare the underlying ontologies or datasets.

Morphisms constitute a general tool for modeling complex relationships between mathematical objects in a disciplined fashion. Krötzsch et al. (2005) discuss the most important morphisms in FCA and propose approaches in ontology research where morphisms help formalize the interplay among distributed knowledge bases.

2.4.4.5. Fuzzy and rough FCA in ontology construction and mapping

In this section, we discuss the 14% of papers on fuzzy and rough FCA in ontology construction and mapping. Quan et al. (2004b) combine fuzzy logic with FCA for the automatic generation of ontologies. These ontologies are used to support the Scholarly Semantic Web, which is used for sharing, reuse and management of scholarly information. Quan et al. (2006a) use fuzzy FCA for automatically generating a fuzzy ontology on uncertainty information and propose the Fuzzy Ontology Generation framework (FOGA). Quan et al. (2006b) use this framework in the context of customer service support, where it plays an essential role in the development of a semantic web help-desk for supporting customer services. Zhou et al. (2007a) also use fuzzy FCA for the creation of a fuzzy ontology. Huang et al. (2008a) combine rough set theory with FCA for semi-automatically constructing domain ontologies. Maio et al. (2009) use fuzzy FCA to generate an ontology-based knowledge network from RSS feeds. The method results in a semantic aggregation of feeds and enables the accessibility of web resources presented by feeds. Zhao et al. (2006) present a novel similarity measure method for ontology mapping based on the combination of rough set theory and FCA, referred to as Rough Formal Concept Analysis (RFCA). Zhao et al. (2008) propose a weighted rough ontology mapping method based on RFCA. The two source ontologies are first transformed into formal contexts and are then merged to obtain a complete concept lattice. Finally, a rough similarity measure is introduced to produce the ontology mapping results. Xu et al. (2008) and Gu et al. (2008) enhance the RFCA ontology mapping method with attribute reduction to improve mapping efficiency and adapt it to a large scale domain ontology.

2.5. Extensions and related disciplines of FCA theory

FCA has been extended in many ways to deal with the requirements of real-world applications. The lattice in Figure 2.7 groups the 702 papers according to some of the extensions of traditional FCA theory. The extensions we choose to zoom in on include fuzzy concept lattice (Belohlavek 1999), triadic concept lattice (Wille 1995), variable threshold concept lattice (Zhang 2007a), rough formal concept lattice (Zhao 2007), etc. 18% of the 702 FCA papers can be categorized under one or more of the research topics in the
2.5. Extensions and Related Disciplines of FCA Theory

The 11% of the papers in which FCA is combined with fuzzy theory are summarized in section 2.5.1. In section 2.5.2, the 7% of papers on combining FCA with rough set theory are discussed. The 1% of papers on FCA and AFS algebra are described in section 2.5.3. Section 2.5.4 summarizes the 1% of papers on triadic concept analysis. Section 2.5.5 describes the 1% of papers on temporal concept analysis.

Fig. 2.7. Lattice describing some of the extensions of traditional FCA theory

2.5.1. Fuzzy FCA

In the basic setting of FCA, attributes are binary, i.e. table entries are 1 or 0 according to whether an object has the attribute or not. If the attributes under consideration are fuzzy (like “tall”, “small”), each table entry contains a truth degree to which an attribute applies to an object. Fuzzy Set Theory was invented by Zadeh (1965). Contrary to classical logic, fuzzy logic uses a scale $L$ of truth degrees, the most favorite choice being $[0,1]$ or some subchain of $[0,1]$. This enables to consider intermediate truth degrees of propositions, e.g. “object $x$ has attribute $y$” has a truth degree 0.8 indicating that the proposition is almost true. In addition to $L$ one has to pick an appropriate collection of logical connectives. This combination of truth
degrees and logical connectives is represented by a complete resituated lattice. In combination with FCA, fuzzy theory can be a powerful instrument for analyzing document knowledge with uncertainty information. In the context of our work it might be interesting to introduce a scale ranging from 0 to 1 that indicates to what degree a case can be seen as domestic violence. In chapter 3 we will see that the notion of domestic violence is not as crisp as many people think but to some extent fuzzy. This fuzzy nature may be further investigated in the future. Belohlavek et al. (2005a) present a survey and comparison of the different approaches to fuzzy concept lattices. Belohlavek et al. (2006a) give an overview of recent developments concerning attribute implications in a fuzzy setting. In databases, these formulas are also called functional dependencies. Fuzzy concept lattices were first introduced by Burusco et al. (1994) and further developed by Pollandt (1997) and Belohlavek (2004).

81% of the papers on fuzzy FCA are covered by the research topics in the lattice of Figure 2.8. Section 2.5.1.1 summarizes some of the applications of fuzzy FCA in linguistics, schema matching, biology, etc. (8% of the papers). Section 2.5.1.2 describes the papers on fuzzy concept lattice reduction (8% of the papers). Section 2.5.1.3 covers the 15% of papers on parameterized approaches to control the number of extracted concepts. In section 2.5.1.4, the papers on multi-adjoint and generalized concept lattices are summarized. Section 2.5.1.5 discusses some of the variations on traditional fuzzy FCA theory. Section 2.5.1.6 covers fuzzy attribute logic, implications and...
inference. The papers on using fuzzy FCA in KDD (16% of papers), IR (12% of papers) and ontology engineering (9% of papers) have been discussed in sections 2.4.1.8, 2.4.2.6 and 2.4.4.5 respectively.

2.5.1.1. Applications of fuzzy FCA

Fuzzy clustering allows objects of a dataset to belong to several clusters simultaneously, with different degrees of membership. The data is thus partitioned into a number of fuzzy partitions. Sassi et al. (2007) propose a new approach for clustering quality evaluation while combining fuzzy logic with FCA. FCA is used for interpreting and distinguishing overlapping clusters and evaluating the quality of clusters. Nobuhara et al. (2006) present a hierarchical representation method for image/video databases based on FCA and fuzzy clustering (FCM). FCM is first performed to a vast amount of objects and the number of objects is reduced into suitable numbers for visualization. The method was confirmed to be helpful to do video clipping and grasp the whole structure of an image database. Geng et al. (2008) present a method to identify topics in email messages. FCA is used as semantic analysis method to group emails containing the same keywords into concepts. The fuzzy membership functions are used to rank the concepts based on the features of the emails. Zhang et al. (2007b) discuss the extraction of fuzzy linguistic summaries from a continuous information system. They use FCA in combination with degree theory to obtain these fuzzy linguistic summaries (Zhang 1996) and give the example of checking quality of sweetened full cream milk powder. Schema matching is the core of data exchange and it takes two schemas S and T as input and returns mapping elements between those two schemas as output. Each mapping element specifies that certain elements of S logically correspond to, i.e. match, certain elements of T, where the semantic of this correspondence is expressed by some expression. Feng et al. (2008) introduce a new schema matching approach based on the naive Bayes, classifier fuzzy concept lattice and a structural similarity measure to calculate the final matching. Bertaux et al. (2009) describe a method to identify ecological traits of species based on the analysis of their biological characteristics. The complex structure of the dataset is formalized as a fuzzy many-valued context and transformed into a binary context through histogram scaling. The core of the method relied on the construction and interpretation of formal concepts. The concepts were interpreted by a hydrobiologist, leading to a set of ecological traits which were inserted in the original context.

2.5.1.2. Fuzzy concept lattice reduction
An issue that may arise while using fuzzy concept lattices is the potentially large number of concepts extracted from the data. Multiple methods have been devised to reduce the number of concepts. Elloumi et al. (2003) use fuzzy FCA to reduce the size of fuzzy data tables to only keep the minimal rows in each table, without losing knowledge. The authors develop a fuzzy extension of a previously proposed algorithm for crisp data reduction. The fuzzy Galois connection based on the Lukasiewicz implication (Elloumi et al. 2001) is used in the definition of the closure operator and a precision level is used, which makes data reduction sensitive to the variation of this precision level. For each precision level, the authors aim to preserve the same knowledge extracted from the initial database. Belohlavek et al. (2005b) investigate crisply generated formal fuzzy concepts which are particular formal fuzzy concepts that can be considered as more important than the others. To cope with the computational complexity of creating concept lattices from a large context, Butka et al. (2008) describe a new approach for the creation of a hierarchy of concepts. The starting set of documents is decomposed into smaller sets of similar documents with the use of a clustering algorithm. Then one concept lattice is built upon every cluster using FCA and these FCA-based models are combined to a simple hierarchy of concept lattices using agglomerative clustering. Zhou et al. (2007b) propose a method to overcome the large number of extracted concepts in an ordinary concept lattice. They use fuzzy FCA for getting a more concise concept representation and a concept hierarchy based classifier is produced. While the accuracy of classification is preserved, the concept hierarchy achieves a good compression rate. Kumar et al. (2009) propose a method based on Fuzzy K-means clustering for reducing the size of concept lattices. They demonstrate the usefulness of their method for information retrieval and information visualization.

Belohlavek (2008a) presents theoretical results regarding decomposition of matrices $I$ with entries from a bounded ordered set $L$. If $I$ is an $n \times m$ matrix, the authors look for a decomposition of $I$ into a product $A \circ B$ of an $n \times k$ matrix $A$ and a $k \times m$ matrix $B$ with entries from $L$ and $k$ as small as possible. A decomposition $I \rightarrow A \circ B$ corresponds to discovery of $k$ factors explaining the original data $I$. Formal concepts are used as factors and can be seen as particular submatrices of $I$. Decompositions using formal concepts as factors are optimal in that they provide us with the least number of factors possible.

2.5.1.3. Parameterized approaches for fuzzy concept lattices

Several parameterized approaches were introduced where the parameters control the number of extracted formal concepts. Hedges are discussed in 8% of the papers and were introduced as parameters for FCA with fuzzy attributes in (Belohlavek et al. 2005c, Belohlavek et al. 2005d, Belohlavek et
For particular choices of hedges one obtains the original approach by Pollandt (1997) and Belohlavek (2002) and the one-sided fuzzy approach (Yahia 2001, Krajci 2003). Thresholds on the other hand are discussed in 5% of the papers and work as follows. Given a collection $A$ of objects, the collection $A'$ of all attributes shared by all objects from $A$ is a fuzzy set and these attributes belong to $A'$ in various degrees. One can pick the threshold $f$, and consider the set of attributes which belong to $A'$ in a degree greater than or equal to $f$. This approach was proposed for $f = 1$ in (Krajci 2003, Yahia et al. 2001). In (Elloumi et al. 2004), this was extended for arbitrary $f$. However, the extent- and intent-forming operators defined in (Elloumi et al. 2004) did not form a Galois connection, which was resolved in (Fan et al. 2007b), where the authors proposed new operators based on the idea of a threshold for general $f$. Belohlavek et al. (2006b) shows that the approach via thresholds can seen as a particular case of the approach via hedges. For data with fuzzy attributes, the fuzzy concept lattice obtained with the operators in (Fan et al. 2007b) is isomorphic to a fuzzy concept lattice with hedges induced from data containing so called shifts of the given fuzzy attributes. The authors also apply the idea of thresholds to attribute implications from data with fuzzy attributes (Belohlavek et al. 2005e). Interesting to note is that shifts of fuzzy attributes play an important role in the efficient computation of a factorization by similarity of a fuzzy concept lattice (Belohlavek 2000, Belohlavek et al. 2004a). Belohlavek et al. (2007a) describe how to obtain the factor lattice from a large fuzzy concept lattice by means of a similarity relation. The factor lattice contains less clusters than the original concept lattice but at the same time represents a reasonable approximation of the original lattice. The user can specify a similarity threshold and a smaller threshold leads to smaller factor lattices, i.e. more comprehensible but less accurate approximations of the original concept lattice. In Belohlavek et al. (2008a), this research is extended to fuzzy concept lattices with hedges. Zhang et al. (2007a) introduce the definition of a variable threshold concept lattice. The authors define three kinds of variable threshold concept lattices: a first between two crisp sets, a second between a crisp and a fuzzy set and a third between a fuzzy and a crisp set. Such a variable threshold lattice contains considerably less concepts than a fuzzy concept lattice.

### 2.5.1.4. Multi-adjoint and generalized concept lattices

The multi-adjoint framework originated as a generalization of several non-classical logic programming frameworks and is investigated in 5% of the papers. Its semantic structure is the multi-adjoint lattice, in which a lattice is considered together with several conjunctors and implications making up adjoint pairs (Medina et al. 2001). The multi-adjoint approach was further developed in the context of general logical frameworks in Medina et al. (2004) and Julian et al. (2005). Medina et al. (2007) introduce a new...
representation theorem for the multi-adjoint concept lattices which allows the representation theorems of the other paradigms to be proved more directly. Medina et al. (2009a) show that multi-adjoint concept lattices embed different fuzzy extensions including Krajci's generalized concept lattices. Another main result is the representation theorem of this paradigm.

Krajci (2005a) considered the so-called generalized concept lattices, which use different sets of truth values to refer to a subset of objects, to a subset of attributes as well as to a degree to which an object has an attribute. Generalized concept lattices were proposed to form a common platform for (Belohlavek 2004, Pollandt 1997) and the so-called one sided fuzzy concept lattices introduced by (Belohlavek 2005, Yahia 2001, Krajci 2003). Krajci (2005b) showed that generalized concept lattices embed some other approaches like the concept lattice with hedges. Medina et al. (2008) shows that the framework of generalized concept lattices is wide enough to adequately represent the concept lattice of Georgescu. Georgescu et al. (2002) defined the notion of a fuzzy concept lattice associated with fuzzy logic with a non-commutative conjunction. The authors also prove that any concept lattice for non-commutative fuzzy logic can be interpreted in the framework of generalized concept lattices. In the sense that it is isomorphic to a sublattice of the Cartesian product of two generalized concept lattices. Zhang et al. (2004) defined the notion of approximable concept on the Chu space. Chen et al. (2008) introduce two generalizations of this approximable concept lattice: one based on the concept lattice in fuzzy setting (Belohlavek 1999) and the other based on the generalized concept lattice (Krajci 2005).

2.5.1.5. Variations to traditional fuzzy FCA

Belohlavek et al. (2007c) investigate the problem of approximating possibly infinite sets of solutions by finite sets of solutions. These infinite sets of solutions typically appear in constraint-based problems such as “find all collections in a given finite universe satisfying constraint C”. In fuzzy setting, when collections are conceived as fuzzy sets, the set of all such collections may be infinite and computationally intractable when one uses the unit interval [0,1] as the scale of membership degrees. The authors propose to use a finite subset \( K \) of [0,1] which approximates [0,1] to a satisfactory degree and illustrate the idea on FCA. Since the extent to which “object \( o \) has property \( a \)” may be sometimes hard to assess precisely. Djouadi et al. (2009) use a sub-interval from the scale \( L \), rather than a precise value. Such formal contexts lead to interval-valued formal concepts. The authors provide a minimal set of requirements for interval-valued implications in order to fulfill the fuzzy closure properties of the resulting Galois connection. Secondly a, new approach based on a generalization of Gödel’s implication is proposed for building the complete lattice of all interval-valued formal concepts. Scaling is a process of transformation of data tables with general attributes,
2.5. EXTENSIONS AND RELATED DISCIPLINES OF FCA THEORY

The one-sided fuzzy version of FCA is the extension that works with real values in the document-term matrix. Belohlavek (2004b) considers using a fuzzy partial order for a fuzzy concept lattice instead of an ordinary partial order. Belohlavek et al. (2006c) present fuzzy attribute logic (FAL) for reasoning about formulas describing particular attribute dependencies. The formulas are of the form $A \Rightarrow B$ where $A$ and $B$ are collections of attributes. These formulas can be interpreted in 2 ways. First, in database tables with entries containing degrees to which objects have attributes. Second, in database tables where each domain is equipped with a similarity relation assigning a degree of similarity to any pair of domain elements. If the scale contains only two degrees, 0 and 1, two well-known calculi become particular cases of FAL. With the first interpretation, FAL coincides with attribute logic used in FCA, with the second interpretation, the logic coincides with Armstrong system for reasoning about functional dependencies. Belohlavek et al. (2006d) focus on similarity related to attribute implications, i.e. rules $A \Rightarrow B$ describing dependencies “each object which has all attributes from $A$ has also all attributes from $B$.” The authors present several formulas for estimation of similarity of outputs in terms of similarity of inputs. Fan et al. (2006) propose a new form of fuzzy concept lattice and two coherent fuzzy inference methods for this and 3 existing kinds of fuzzy concept lattices. The lower approximate fuzzy inference and the upper approximate fuzzy inference based on the fuzzy concept lattice are proposed, and the combined use of the two methods will make the fuzzy inference more precise.

2.5.2. Rough set theory

Rough Set Theory (RST) was invented by Pawlak (1982) and is a mathematical technique to deal with uncertainty and imperfect knowledge. In rough set theory, the data for analysis are described by an information system $(U, A, F)$, which corresponds to the formal context in FCA, and consists of universe $U$, attributes set $A$ and relation $F$ between $U$ and $A$. Objects characterized by the same properties are indiscernible (similar) in view of the
available information about them. By modeling indiscernibility as an equivalence relation, one can partition a finite universe of objects into pairwise disjoint subsets. The partition provides a granulated view of the universe. An equivalence class is considered as a whole, instead of many individuals. In other words, one can only observe, measure or characterize equivalence classes. The empty set, equivalence classes and unions of equivalence classes form a system of definable subsets under discernibility. All subsets not in the system are consequently approximated through definable sets. Any set of all indiscernible (similar) objects is called an elementary set (neighborhood) and forms a basic granule (atom) of knowledge about the universe. Any union of elementary sets is a crisp (precise) set, otherwise the set is rough (imprecise, vague). Each rough set has boundary-line cases, i.e., objects which cannot be classified with certainty as either members of the set or its complement. Crisp sets have no boundary-line elements at all. Boundary-line cases cannot be properly classified by employing the available knowledge.

Vague concepts (in contrast to precise concepts) cannot be characterized in terms of information about their elements. Any vague concept is replaced by a pair of precise concepts, called the lower and upper approximation of the vague concept. The lower approximation consists of all objects which surely belong to the concept and the upper approximation contains all objects which possibly belong to the concept. The difference between the lower and upper approximation constitutes the boundary region of the vague concept. Rough sets may become interesting for detecting human trafficking suspects. We have a lot of indicators with corresponding term clusters. The terms in these clusters are sometimes too specific and sometimes too general resulting in relevant cases being missed or irrelevant cases being selected. This upper and lower boundary may help us refine our thesaurus and more efficiently extract relevant reports.
Fig. 2.9. Lattice describing the 46 papers on FCA and rough set theory

85% of the papers on rough set theory and FCA are covered by the topics in Figure 2.9. In section 2.5.2.1, we discuss the 28% of papers on combining FCA with RST. Section 2.5.2.2 covers the different types of generalisations of traditional RST that were introduced over the years and the link with FCA (15% of papers). Section 2.5.2.3 covers the 24% of papers on attribute reduction of formal concept lattices using RST. Section 2.5.2.4 describes the 15% of papers on combining fuzzy FCA with RST.

2.5.2.1. Combining FCA with rough set theory

Many efforts have been made to combine FCA and rough set theory (Yao 2004a). This combination is typically referred to as Rough Concept Analysis. Gediga et al. (2002) introduced a lattice constructed based on approximation operators. The extent and intent of a formal concept can be viewed as two systems of definable sets of objects and attributes respectively (Yao 2004a, Yao 2004b). The approximation operators can then be formulated with respect to these systems and introduced into FCA. Deogun et al. (2005)
evaluate three approaches for concept approximation. Concept approximation
is to find the best or closest concepts to approximate a pair of objects and
features. Under the circumstances one cannot find a concept, concept
approximation will give the best solution. Kent (1993) uses an equivalence
relation on the set of objects. With respect to the formal context, a pair of
upper and lower contextual approximations is defined. The two contextual
approximations are then used to define a pair of lower and upper
approximations of concepts. The other approach is based on the system of
definable concepts in the concept lattice (Hu et al. 2001, Saquer et al. 1999).
Saquer et al. (1999) studied approximations of a set of objects, a set of
properties, and a pair of a set of objects and a set of properties, based on the
formal concepts of a concept lattice. For example, given a set of objects, the
authors tried to approximate the set by formal concepts whose extents
approximate the set. An equivalence relation is introduced on the set of
objects from a formal context, which leads to rough set approximations.
Their formulation is flawed since an equivalence class is not necessarily the
extent of a formal concept. The union of extents of a family of formal
concepts may not be the extent of a formal concept. Hu et al. (2001)
suggested an alternative formulation to ensure that approximations are indeed
formal concepts. Instead of an equivalence relation, they defined a partial
order on the set of objects. Unfortunately their definition of lower
approximation had the same shortcoming as Saquer’s.

The notion of approximation operators can be defined based on two
universes linked by a binary relation (Yao 2004a). Based on the common
notion of definability, Yao et al. (2004c) propose a framework for using
rough set approximations in FCA. Comparative examination of RST and
FCA shows that each of them deals with a particular type of definability.
While FCA focuses on sets of objects that can be defined by conjunction of
properties, RST focuses on disjunction of properties. An arbitrary concept is
approximated from below and above by two definable concepts. The author
shows that the problem with existing studies can be solved by a clear
separation of two systems, the FCA lattice and the system of extensions of
formal concepts. The two systems give rise to two different types of
approximation. Yao et al. (2006) further compare and combine both theories
based on definability. There is a close connection between definability and
approximation. A definable set of the universe of objects is definable if and
only if its lower approximation is equal to its upper approximation.

Grabowski et al. (2004) investigate the facilities of the Mizar system
concerning extending and combining theories based on structure and attribute
definitions and as an example, the authors consider the formation of rough
concept analysis out of FCA and RST. Pagliani (2006) introduced a
framework for comparing and combining FCA and rough set systems. Jiang
et al. (2006) introduced rough-valued contexts into FCA and defined how to
obtain formal concepts from these extended contexts. Lai et al. (2009) show
that the expressive power of concept lattices based on rough set theory is
2.5. EXTENSIONS AND RELATED DISCIPLINES OF FCA THEORY

Weaker than that of concept lattices based on FCA. Xu et al. (2007) consider the combination of FCA and RST. The authors present the notion of information concept lattice for which some properties are given. They also present a reduction theory for information concept lattices and compare the information concept lattice with rough set theory and concept lattices. Shao et al. (2007a) combine RST and FCA and propose the notion of information granule in information systems. The authors show all the information granules form a complete lattice and present approaches for attribute reduction and rule acquisition for information granularity lattices.

2.5.2.2. Generalizations of rough set theory

Classical RST is developed based on an equivalence (indiscernibility) relation on a universe of objects. The equivalence relation is a stringent condition that limits the application domain of the rough set model. Generalized formulation has been proposed by using a binary relation on two universes, one is the set of objects and the other the set of properties (Yao 1997, Gediga 2002). A binary relation on two universes is known as a formal context in FCA and serves as a common basis for RST and FCA. Many authors have generalized the notion of approximation operators. The notions of formal concept and lattice can be introduced in RST by constructing different types of formal concepts. Duntsch et al. (2003), following the study of modal logics, defined modal-style operators based on a binary relation and introduced the property-oriented concept lattice. The derivation operator of FCA is a polarity or sufficiency operator used in modal logics, and the rough set approximation operators are the necessity and possibility operators used in modal logics. Yao (2004b) uses the formulation in terms of modal-style operators to gain more insights into RST and FCA and defined the object-oriented formal concept lattice. The derivation operator of FCA is a polarity or sufficiency operator used in modal logics, and the rough set approximation operators are the necessity and possibility operators used in modal logics. Yao (2004b) uses the formulation in terms of modal-style operators to gain more insights into RST and FCA and defined the object-oriented formal concept lattice. The derivation operator of FCA is a polarity or sufficiency operator used in modal logics, and the rough set approximation operators are the necessity and possibility operators used in modal logics. Chen (2009) defines two $L$-rough approximation operators by an arbitrary $L$-relation, some of their properties and relation to Galois connection in FCA are investigated. The generalizations of the property-oriented concept lattice and the object-oriented concept lattice are obtained in $L$-rough sets.

One way to define approximation operators is called the subsystem-based formulation (Yao 2005). The elements of the subsystem are understood as definable or observable sets. Every subset of the universe is approximated from below and above by two sets in the subsystem. There are 2 basic restrictions of the standard Pawlak model. First, a single subsystem of the power set is used. Second, the subsystem is closed under set complement, intersection and union. Many studies on generalized rough set approximations try to remove these restrictions. In Ganter (2008a), the lower and upper approximations were replaced by arbitrary kernel and closure operators respectively and the resulting lattices obtained from generalized approximation operators were described as P-products. Meschke (2009)
further investigates this unifying theory and zooms in on the role of the robust elements and the possible existence of suitable negation operators. Wolski (2005) examines FCA and RST against the background of the theory of finite approximations of continuous topological spaces. The author defines operators of FCA and RST by means of the specialization order on elements of a topological space \( X \) which induces a finite approximation of \( X \). Typically, two subsystems are used; one for lower approximation and one for upper approximation. Xu et al. (2008a) propose a generalized definition of rough set approximations, based on a subsystem of subsets of a universe. The subsystem is not assumed to be closed under set complement, union and intersection. The lower and upper approximation is no longer one set but composed of several sets. As special cases, approximations in FCA and knowledge spaces are examined.

2.5.2.3. Attribute reduction

In this section, we discuss the 24% of papers on attribute reduction of concept lattices using techniques from RST. Zhang et al. (2005) presented a framework for attribute reduction of formal concept lattices and introduced the notion of formal decision context. Wei et al. (2008) present a method for attribute reduction of concept lattices in formal decision contexts. Li (2009a) investigates attribute reduction of object-oriented concept lattices in formal decision contexts. Liu et al. (2007) propose two reduction methods based on rough set theory for concept lattices. The authors apply their multi-step attribute reduction method to the reduction of redundant premises of the multiple rules used in the job shop scheduling problem. Li (2009b) focuses on attribute reduction of formal concepts via covering rough set theory. Wang et al. (2006a) discuss some of the basic relationships between the extent of concepts and the equivalence class in rough set theory. The authors also study the relation between the reduction of formal concept lattices and attribute reduction in rough set theory. Shao et al. (2005) introduce a pair of rough set approximations in FCA, based on both lattice-theoretic and set-theoretic operators. Algorithms for attribute and object reduction in concept lattices are also presented.

A case based reasoning system solves problems using prior experience and is typically oversensitive to noise, resulting in overfitting. Rough set theory is used to assure minimally sufficient cases in its case base. FCA is used to reveal knowledge of attribute dependencies in terms of concept lattices (Tadrat 2007).

2.5.2.4. Fuzzy FCA and rough set theory
Pawlak’s rough set model can be generalized to fuzzy environment to deal with quantitative data. In this section, we summarize the 15% of papers on combining fuzzy FCA with RST. The results are called rough fuzzy sets and fuzzy rough sets. Shao et al. (2007b) study rough set approximations within formal concept analysis in a fuzzy environment and present approaches to attribute reduction and rule acquisition for information granularity lattices. Lai et al. (2009) present a comparative study of concept lattices of fuzzy contexts based on FCA and rough set theory. Every complete fuzzy lattice can be represented as the concept lattice of a fuzzy context based on FCA (Belohlavek 2004). Yao et al. (2009) study the approaches for constructing fuzzy concept lattices based on generalized fuzzy rough approximation operators. The authors then propose three kinds of fuzzy Galois connections and three kinds of lattices can be produced for which the properties are analogous to those of the classical concept lattices. Belohlavek et al. (2008c) present necessary and sufficient conditions on input data for the output concept lattice to form a tree after one removes its last element. These conditions are presented for data with binary attributes and for input data with fuzzy attributes.

2.5.3. Monotone concept and AFS algebra

Deogun et al. (2004) discussed some of the limitations of Wille’s formal concept and proposed the monotone concept. In Wille’s definition of concepts, only one set is allowed as extent (intent). For many applications, it would be useful to allow intents to be defined as a disjunctive expression. The monotone concept is a generalization of Wille’s notion of concept where disjunctions are allowed in the intent and set unions are allowed in the extent. This generalization allows an information retrieval query containing disjunctions to be understood as the intent of a monotone concept whose answer is the extent of that concept. By using rough set theory, Saquer et al. (2003) provided a method to find monotone concepts whose intents are close to the query, and showed how to find monotone concepts whose extents approximate any given set of objects.

AFS (Axiomatic Fuzzy Set) algebra was proposed by Liu (1998) and is a new approach for the semantic interpretation of fuzzy information. In Wang et al. (2008), the AFS formal concept is proposed, which extends the Galois connection of a regular context to the connection between two AFS algebra systems. In an information retrieval system, the logic relationships between queries are usually “and” and “or”. AFS formal concepts can be used to represent queries with complex logic operations. When using an information retrieval system, we often find that not all queries are exactly contained in the database, but some items close to those are enough to satisfy the user’s need. The authors study how to approximate a complex attribute by AFS formal concepts such that the intents of the lower and upper approximating concepts
are close to the complex attribute’s underlying semantics. A complex attribute is compounded by some elementary attributes under logic operations "and" and "or". Zhang et al. (2006) explore the relationships between concept lattices and AFS algebra. The authors analyze concept lattices from the point of AFS algebra. Liu et al. (2006) investigate the relationships between FCA lattices and AFS algebra. The authors show how a concept lattice can be obtained from a given AFS algebra.

2.5.4. Triadic FCA

FCA has been extended to deal with three-dimensional data (Wille 1995). Triadic concepts consist of three sets, namely objects, attributes and conditions under which objects may possess certain attributes. With the increased use of folksonomies as core data structure of social resource sharing systems, the interest in Triadic Concept Analysis increased during the past years. A folksonomy is typically used in social bookmarking systems and consists of assignments of arbitrary keywords - called tags - to resources by users. They are knowledge representation tools for sharing knowledge on the web and unlike ontologies, folksonomies are not formalized and do not force the users to use the same tags. Jaschke et al. (2007) present the algorithm TRIAS for mining all closed itemsets (i.e. frequent tri-concepts) from three-dimensional data structures to allow for mining folksonomies. Kim et al. (2007b) use triadic FCA to contextualize folksonomies with respect to users, communities, goals, etc. Stumme (2005) goes one step back and discusses how traditional line diagrams of standard dyadic concept lattices can be used for exploring triadic data. The author showcases how it can be used for navigating through the Federal Office for Information Security IT baseline Protection Manual. Triadic FCA’s relevance has been investigated by us for software engineering activities. Chapter 4 however showed that it was not useful for this purpose and we had to introduce a second ordering relation on concepts instead of triadic FCA.

2.5.5. Temporal FCA

Temporal Concept Analysis (TCA) is based on FCA and addresses the problem of conceptually representing discrete temporal phenomena (Wolff 2001). Wolff (2005) gives an overview of the present state of TCA theory, some applications in medicine and chemical industry and the computer program SIENA for graphical representation of temporal systems. The pivotal notion of TCA theory is that of a conceptual time system (Wolff et al. 2003) containing the observations of objects at several points of time. Abstraction is made of the duration of an observation and the notion of a point of time, also called time granule is used. The starting point is a set of
which the elements are time granules. This multi-valued context consists of
an event part and a time part. The attributes observed at each of these time
granules are described in the event part of the data table. The pair \((T, C)\)
is called a conceptual time system on the set \(G\) of time granules. \(T\) is called
the time part and \(C\) the event part or space part of \((T, C)\). The combination of \(T\)
and \(C\) is the context of the conceptual time system \((T, C)\). The object
concepts of this context are called situations, the object concepts of \(C\) are
called states and the object concepts of \(T\) are called time states. The sets of
situations, states and time states are called the situation space, the state space
and the time state space of \((T, C)\) respectively. In the visualization of the
data, the “natural temporal ordering” of the observations is expressed using a
time relation \(R\) introduced on the set \(G\) of time granules of a conceptual time
system. Wolff (2001) introduced transitions in conceptual time systems as
pairs of time objects where a transition can be considered as a “step from one
point to another” for an object. Transitions between situations, states, time
states, and phases can then be induced easily. This leads to effective temporal
representations of processes. Each arrow in a TCA lattice represents a
“transition of the object” and corresponds to an element of \(R\). The transitions
form a life track of the object.

Wolff (2002b) relates the notions of states and transitions in automata
theory to the conceptual description of states, situations and transitions in
TCA. The author proves that labeled transition systems with attributes (a
generalization of automata) can be represented as state-transition-systems of
conceptual time systems. Wolff et al. (2004) defined particles and waves
from physics in terms of TCA. Waves and wave packets are “distributed
objects” in the sense that they may appear simultaneously at several places.
Wolff et al. (2005b) investigate how to represent the state of such a
distributed object in TCA theory. Wolff et al. (2005a) investigate the
connection between the theory of computation and TCA. For a given Turing
machine, the uniquely determined sequence of computation steps is
represented for each possible input as a life track of an object in some
conceptually described state space. In Poelmans et al. (2010c) and section
5.1, TCA is used in combination with FCA for detecting and profiling human
trafficking suspects. In section 5.2 we used FCA and TCA for detecting and
profiling terrorism suspects.

2.6. Mathematical research and algorithmic innovations

While we explored the remaining papers, we found that they mainly
contained mathematical research related to FCA theory and algorithmic
innovations for lattice construction and drawing. Some of the most frequently
occurring research topics are displayed in the lattice of Figure 2.10.
In 6% of the papers, the authors investigate the computational complexity of existing algorithms or they compare the performance of a newly developed algorithm to existing ones. The research about combining different types of algebraic structures with FCA covers 5% of the papers. In 5% of the papers, FCA is used in combination with logic or logical reasoning and in most cases, this is Description Logic-based. Lattice construction covers 3% of the papers and lattice drawing innovations 2% of the papers. Neural network techniques and FCA are combined in 3% of the papers. Finally, graph theory and conceptual graphs are combined with FCA in 1% of the papers.

2.7. Conclusions

Since its invention in 1982 as a mathematical technique, FCA became a well-known instrument in computer science. Over 700 papers have been published over the past 7 years on FCA and many of them contained case studies showing the method’s usefulness in real-life practice. This chapter showcased the possibilities of FCA as a meta technique for categorizing the literature on concept analysis. The intuitive visual interface of the concept lattices allowed for an in-depth exploration of the main topics in FCA research. In particular, it’s combination with text mining methods resulted in a powerful synergy of automated text analysis and human control over the discovery process.
One of the most notorious research topics covering 20% of the FCA papers is KDD. FCA has been used effectively in many domains for gaining actionable intelligence from large amounts of information. Information retrieval is another important domain covering 15% of the papers. FCA was found to be an interesting instrument for representation of and navigation in large document collections and multiple IR systems resulted from this research. FCA was also used frequently (13% of papers), amongst others in the context of semantic web, for ontology engineering and merging. Finally, 5% of the papers devoted attention to improve FCA’s applicability to larger data repositories.

In 18% of the papers, traditional concept lattices were extended to deal with uncertain, three-dimensional and temporal data. In particular, combining FCA with fuzzy and rough set theory received considerable attention in the literature. Temporal and Triadic Concept Analysis received only minor attention. In the future, we will host the references and links to the articles on a public interface and hope that this compendium may serve to guide both practitioners and researchers to new and improved avenues for FCA.
In this chapter we propose a human-centered process for knowledge discovery from unstructured text that makes use of Formal Concept Analysis (FCA) (Wille 1982, Ganter 1999) and Emergent Self Organizing Maps (ESOM) (Ultsch et al. 2005a, Ultsch et al. 2005b). Human-centered KDD refers to the constitutive nature of human interpretation for the discovery of knowledge, and stresses the complex, interactive process of KDD as being led by human thought (Brachman et al. 1996). Data mining should be primarily concerned with making it easy, practical and convenient to explore very large databases for organizations and users with vast amounts of data.
but without years of training as data analysts (Fayyad 2002). A significant part of the art of data mining is the user’s intuition with respect to the tools (Pednault 2000, Marchionini 2006).

Visual data exploration (Eidenberger 2004) and visual analytics (Thomas 2005) are especially useful when little is known about the data and exploration goals are vague. Since the user is directly involved in the exploration process, shifting and adjusting the exploration goals is automatically done if necessary. In addition to the direct involvement of the user, the main advantages of visual data exploration over automatic data mining techniques from statistics or machine learning are: visual data exploration can easily deal with highly non-homogeneous and noisy data and visual data exploration usually allows a faster data exploration and often provides better results, especially in cases where automatic algorithms fail. In addition, visual data exploration techniques provide a much higher degree of confidence in the findings of the exploration (Keim 2002).

This chapter extends but also synthesizes our previous work involving FCA and ESOM, two visually appealing data exploration aids, for knowledge discovery from unstructured text. In (Poelmans 2008, Poelmans 2010e), we first discussed the possibilities of using FCA for knowledge discovery in a police environment. A parallel research track consisted of investigating the potential of using ESOM for knowledge discovery. Our first findings using the ESOM are discussed in (Poelmans 2009a, Poelmans 2009c). We also compared ESOM’s performance to that of other SOMs such as the spherical SOM and we found it to be superior (Poelmans 2009b). In (Poelmans 2009d), we briefly presented our idea to use FCA and ESOM together for domestic violence discovery. The ESOM functions as a catalyst for the FCA based discovery process. The proposed methodology recognizes the important role of the domain expert in mining real-world enterprise applications and makes efficient use of specific domain knowledge, including human intelligence and domain-specific constraints.

We chose for a semi-automated approach since the major drawback of all automated and supervised machine learning techniques, including decision trees, is that these algorithms assume that the underlying concepts of the data are clearly defined, which is often not the case. These techniques allow almost no interaction between the human actor and the tool and fail at incorporating valuable expert knowledge into the discovery process (Keim 2002), which is needed to go beyond uncovering the fool’s gold (Smyth 2002). In the paper presented by Hollywood et al. (2009) these problems were clearly addressed in the context of terrorist threat assessment. The central question was whether it is possible to find terrorists with traditional automated data mining techniques and the answer was no.

The knowledge discovery process is conceptualized and interpreted as successive iterations through the C-K theory design square. C-K theory offers a formal framework that interprets existing design theories as special cases of a unified model of reasoning (Hatchuel 1996, Hatchuel 2002). It provides a
clear and precise definition of design that is independent of any domain of professional tradition (Hatchuel 1999). C-K theory defines design reasoning dynamics as a joint expansion of the Concept (C) and Knowledge (K) spaces through a series of continuous transformations within and between the two spaces. The beauty of C-K theory is that it can provide insight into an iterative and expansive knowledge acquisition process (Hatchuel 2003, Hatchuel 2004). One of the core characteristics of C-K theory is this focus on human intelligence as the driving force in expanding the space of knowledge. To our knowledge, this is the first systematic application of C-K theory to the information systems domain. C-K theory is used as a unifying framework to provide a clear structure to the discovery process based on FCA and ESOM. The combined use of FCA and ESOM in the C-K framework not only gives insight into the generic nature of the KDD activity but also makes for significantly improved results. Some of the aspects of this chapter have already been discussed in the literature in a fragmented way (e.g. information retrieval, knowledge browsing, prior knowledge incorporation), but an integrated approach has never been pursued.

To illustrate its effectiveness, we report on a real-life case study on using the process at the Amsterdam-Amstelland police in the Netherlands aimed at distilling concepts for domestic violence from the unstructured text in filed reports. The aim of our research was to conceptualize and improve the definition and understanding of domestic violence with the ultimate goal of improving the detection and handling of domestic violence cases. One important spin-off of this exercise that will be elaborated on in this section was the development of a highly accurate and comprehensible classification procedure for automatically raising a domestic violence flag for incoming police reports. This procedure automatically classifies 91% of incoming cases correctly whereas in the past all cases had to be dealt with manually. We performed this classification exercise to measure the quality of our conceptualization of domestic violence. We have never seen a similar set up in the literature and to the best of our knowledge there is no packaged automated solution to do all the same at once.

Over 90% of the information available to police organizations is stored as plain text. To date, however, analyses have primarily focused on the structured portion of the available data. Only recently the first steps have been taken to apply text mining in criminal analysis (Chen 2004, Ananyan 2002). Domestic violence is one of the top priorities of the Amsterdam-Amstelland police force in the Netherlands (Politie Amsterdam-Amstelland 2009). In the past, intensive audits of the police databases of filed reports established that many of the reports tended to be wrongly labeled as domestic or as non-domestic violence cases. Previous attempts have mainly focused on developing a machine learning classifier that automatically classified incoming cases as domestic or as non-domestic violence. Unfortunately they were unsuccessful because the underlying concept of domestic violence was never challenged. These systems did not provide any insight into the problem, since they are black-boxes and their classification performance was
around 80% only (Elzinga 2006). As a consequence, these systems never made it into operational policing practice. All of these previous attempts had in common that the concept of domestic violence was never challenged. The developers overlooked the complexity of the notion of domestic violence, were unaware that different people have different visions about the nature and scope of it and did not pay attention to niche cases. Moreover, the correctness of the labels assigned to cases by police officers was never verified. We found that different police officers regularly assigned different labels to the same situation. Finally, the developers did not dispose of a high-quality domain-specific thesaurus that contained sufficient discriminant terms for accurately classifying cases. In the past, several automated term extraction and thesaurus building techniques were used (Elzinga 2006). We interviewed several domain experts that were exposed to these efforts. All of them attested to their failure in constructing a useful thesaurus when we asked them for their appraisal of these prior initiatives.

The remainder of this chapter is composed as follows. In section 3.2 we discuss intelligence led policing, domestic violence and the motivation for this research. In section 3.3, we elaborate on the essentials of FCA, ESOM and C-K theory. In section 3.4, we show how we used the synergistic combination of FCA and ESOM to institute the C-K framework. Section 3.5 then discusses the dataset, while section 3.6 showcases the knowledge discovery process and the four C-K operators described in section 3.3. In section 3.7, we summarize the actionable results of the iterative knowledge enrichment. Section 3.8 contains a comparative analysis of ESOM and multi-dimensional scaling. Finally, section 3.9 presents the main conclusions of this chapter.

3.2. Intelligence Led Policing

Policing is a knowledge intensive affair. Over the past fifteen years or so there have been calls for a shift from a more traditional reactive intuition led style of policing to a more proactive intelligence led approach (Collier 2006). Intelligence Led Policing (ILP) promotes this use of factual, evidence based information and analyses to provide management direction and to guide police actions at all levels of a police organization. The goal is specifically to complement intuition led police actions with information coming from analyses on aggregated operational data, such as crime figures and criminal characteristics (Collier 2004). While over 80% of all information available to police organizations resides in textual form, analysis has to date been primarily focused on the structured portion of the available data. Only recently the first steps for applying text mining in criminal analysis have been taken. Though text mining has been identified as a promising area in the formal framework for crime data mining by Chen et al. (2004), this work has hardly found its way into mainstream scientific literature. One of the
notorious exceptions is the paper by Ananyan (2002) in which historical police reports were analyzed to identify hidden patterns.

In 1997, the Ministry of Justice of the Netherlands made its first inquiry into the nature and scope of domestic violence (Van Dijk 1997). It turned out that 45% of the population once fell victim to non-incidental domestic violence. For 27% of the population, the incidents even occurred on a weekly or daily basis. These gloomy statistics brought this topic to the centre of the political agenda. Acting firmly against this phenomenon became one of the pivotal projects of the Balkenende administration when it took office in 2003.

Domestic violence is nowadays one of the top priorities of the police organization of the region Amsterdam-Amstelland in the Netherlands (Politie Amsterdam-Amstelland 2009). Of course, in order to pursue an effective policy against offenders, being able to swiftly recognize cases of domestic violence and label reports accordingly is of the utmost importance. Still, this has proven to be problematic. In the past intensive audits of the police databases related to filed reports established that many reports tended to be wrongly classified as domestic or as non-domestic violence cases.

3.2.1. Domestic violence

According to the U.S. Office on Violence against Women, domestic violence is a “pattern of abusive behavior in any relationship that is used by one partner to gain or maintain power and control over another intimate partner” (Office on Violence against Women 2007). Domestic violence can take the form of physical violence, which includes biting, pushing, maltreating, stabbing or even killing the victim. Physical violence is often accompanied by mental or emotional abuse, which includes insults and verbal threats of physical violence towards the victim, the self or others, including children. Domestic violence occurs all over the world, in various cultures (Watts 2002) and affects people throughout society, irrespective of economic status (Waits 1985).

The XPol database – the database of the Amsterdam-Amstelland police – contains all documents with regard to criminal offences. Documents related to certain types of crime receive corresponding labels. It is of the utmost importance that a correct label is assigned to each of the filed police reports. First, there are some legal consequences. If the police judged an incident to be domestic violence, the public prosecutor can accuse the offender of committing a domestic violence crime. This is taken into account by the judge as an aggravating circumstance, often resulting in a more severe penalty. Second, police officers will be able to better assess new incidents between the perpetrator and the victim, resulting in a more effective way of tackling the problem. Finally, if a domestic violence label was incorrectly assigned to a case, this will result in a waste of the valuable time of the police officers assigned to the case.
Immediately after the reporting of a crime, police officers are given the possibility to judge whether or not it is a domestic violence case. If they believe it is, they can indicate this by assigning the label “domestic violence” to the report. However, not all domestic violence cases are recognised as such by police officers. This may have several reasons, for example, because of a lack of training, a lack of prior experience or new types of domestic violence occurring. As a consequence, many documents are lacking the appropriate label, which put on the agenda the need for a more efficient and effective case triage software program to automatically filter out suspicious cases for in-depth, manual inspection and classification. The in-place case triage system has been configured to filter out these reports for in-depth manual inspection and classification, with the aim of substantially reducing the number of domestic violence cases that are not recognised as such. It retrieves suspicious cases that lack the label of domestic violence and sends them back to the data quality management team. At present, each case retrieved by the in-place case triage system is subjected to an in-depth manual inspection by one of the co-workers of the quality control department. If analysis reveals that a case was wrongly classified as non-domestic violence, it is sent back to the police officer responsible for the case, who is obliged to re-examine and reclassify the police report. It is obvious that this is a very time-consuming and, by consequence, costly procedure. Given that it takes an individual at least five minutes to read and classify a case, it is clear that more accurate triage will result in major savings.

Currently the triage is based on either one or both of the following two criteria being met. The first criterion is whether the perpetrator and the victim live at the same address. The second criterion is whether any or a combination of the following expressions appear in the case documents: "ex-boyfriend", "ex-girlfriend", "ex-husband", "ex-wife", "domestic", "stalk", "lived together", "live together", "son and scared", "child and scared", "child and threat", "son and threat", "child and threat", "daughter and threat" or "daughter and scared".
A summary of the current domestic violence reporting procedure is displayed in Figure 3.1. There are several problems associated with this process. First, recent audits have confirmed that many of the retrieved cases are wrongly selected for in-depth manual inspection. Going back to 2006, the system retrieved 1157 cases, 80% of which actually turned out to be non-domestic violence cases. For example, going back to 2007, the triage system retrieved 1091 of such cases in which the victim made a statement to the police. Second, because of a lack of manpower the data management quality team was not able to analyze each retrieved police report. Third, audits of the police databases revealed that not all domestic violence cases lacking the appropriate label were retrieved by the case triage system. Fourth, no actions have yet been undertaken to address the issue of the filed reports that were wrongly classified as domestic violence.
3.2.2. Motivation

According to R.S. Brachman et al. (1996), much attention and effort has been focused on the development of data mining techniques, but only a minor effort has been devoted to the development of tools that support the analyst in the overall discovery task. They argue for a more human-centered approach. Human-centered KDD refers to the constitutive character of human interpretation for the discovery of knowledge, and stresses the complex, interactive process of KDD as being led by human thought. In most real-world knowledge discovery applications, an indispensable part of the discovery process is that the analyst explores and sifts through the raw data to become familiar with it and to get a feel for what the data may cover. Often an explicit specification of what one is looking for only arises during an interactive process of data exploration, analysis and segmentation. R.S. Brachman et al. (1993) introduce the notion of data archeology for KDD tasks in which a precise specification of the discovery strategy, the crucial questions and the basic goals of the task have to be elaborated during an unpredictable exploration of the data. Data archeology can be considered as a highly human-centered process of asking, exploring, analyzing, interpreting and learning by interacting with the underlying database. Comprehensible support should be provided to the analyst during the KDD process. According to Brachman et al. (1996) this should be embedded into a knowledge discovery support environment. How the process of human-centered KDD can be supported by Formal Concept Analysis (FCA) was for the first time investigated by Stumme et al. (1998).

Smyth et al. (2002) already stated that the algorithm designer and the scientist should be able to bring in prior knowledge so the data mining algorithm does not just rediscover what is already known. Moreover, the scientist should be able to “get inside” and “steer” the direction of the data mining algorithm. FCA fulfils these requirements. Starting from initial knowledge on the problem area, it provides the user with a visual display of the relevant concepts available in the dataset and their relationships. Additionally, the user can visually interact with the concept lattice and thereby steer the knowledge discovery process.

What makes FCA into an especially appealing technique for knowledge discovery in databases is that it meets the important requirement stated by, amongst others, Fayyad et al. (2002) that data mining should be primarily concerned with making it easy, convenient and practical to explore very large databases for organizations and users with vast amounts of data but without years of training as data analysts. FCA offers the user an intuitive visual display of different types of structures available in the dataset and guides the user in the exploration of the dataset. This end-user-friendly interface also makes the data mining more transparent to the user.
When compared to other, more traditional, techniques such as association rules, FCA has a larger explanatory power because of its underlying non-hierarchical structure (Christopher 1965). While traditional association rules are flat, FCA provides an order of significance, which makes its representation richer and more intuitive to use.

3.3. FCA, ESOM and C-K theory

3.3.1. Formal Concept Analysis

FCA arose twenty-five years ago as a mathematical theory (Ganter 1999, Stumme 2002b) and has over the years grown into a powerful tool for data analysis, data visualization and information retrieval (Priss 2005). The usage of FCA for browsing text collections has been suggested before by Cole et al. (2002). However, none of the papers have focused on how FCA can be used in an actionable environment for knowledge enrichment and for discovering different types of knowledge in unstructured text. FCA has been applied in a wide range of domains, including medicine, psychology, social sciences, linguistics, information sciences, machine and civil engineering, etc (Stumme 2000). For instance, FCA has been applied for analyzing data of children with diabetes (Scheich 1993), for developing qualitative theories in music esthetics (Hereth 2000), for database marketing (Hereth 2000), and for an IT security management system (Becker 2000). In (Eklund 2004, Domingo 2005), FCA was used as a visualization technique that allows human actors to quickly gain insight by browsing through information. Full details on the use of FCA in KDD are given in chapter 2.

We previously applied FCA to a relatively small police dataset and were able to establish its practical usefulness (Poelmans 2008). FCA is particularly suited for exploratory data analysis because of its human-centeredness (Hereth 2003, Valtchev 2004). It is a fundamental principle that the generation of knowledge from information is promoted by representations that make the inherent logical structure of the information transparent. FCA builds on the model that concepts are the fundamental units of human thought. Hence, the basic structures of logic and logical structure of information are based on concepts and concept systems (Stumme 1998, Stumme 2002a). Consequently, FCA uses the mathematical abstraction of the concept lattice to describe systems of concepts to support human actors in their information discovery and knowledge creation practice (Wille 2002). Again, the starting point of the analysis is a database table consisting of rows \( M \) (i.e. objects), columns \( F \) (i.e. attributes) and crosses \( T \subseteq M \times F \) (i.e. relationships between objects and attributes). The mathematical structure used to reference such a cross table is called a formal context \((M, F, T)\). An example of a cross table is displayed in Table 3.1. Here, reports of domestic violence (i.e. the objects) are related (i.e. the crosses) to a number of terms (i.e. the attributes): a report is related to a term if the report contains this
3.3. FCA, ESOM and C-K theory

Term. The dataset in Table 3.1 is an excerpt from the one we used in our research. Given a formal context, FCA then derives all concepts from this context and orders them according to a subconcept-superconcept relation, which results in a line diagram (a.k.a. lattice).

<table>
<thead>
<tr>
<th>Table 3.2. Example of a formal context</th>
</tr>
</thead>
<tbody>
<tr>
<td>kicking</td>
</tr>
<tr>
<td>report 1</td>
</tr>
<tr>
<td>report 2</td>
</tr>
<tr>
<td>report 3</td>
</tr>
<tr>
<td>report 4</td>
</tr>
<tr>
<td>report 5</td>
</tr>
</tbody>
</table>

Fig. 3.2. Line diagram corresponding to the context from Table 1

Retrieving the extension of a formal concept from a line diagram such as the one in Figure 3.2 implies collecting all objects on all paths leading down from the corresponding node. In this example, the objects associated with the third concept in row 3 are reports 2 and 3. To retrieve the intension of a
formal concept, one traces all paths leading up from the corresponding node in order to collect all attributes. In this example, the third concept in row 3 is defined by the attributes “stabbing,” “cursing” and “scratching”. The top and bottom concepts in the lattice are special: the top concept contains all objects in its extension, whereas the bottom concept contains all attributes in its intension. A concept is a subconcept of all concepts that can be reached by travelling upward and it will inherit all attributes associated with these superconcepts. Note that the extension of the concept with attributes “kicking” and “dad hits me” is empty. This does not mean that there is no report that contains these attributes. However, it does mean that there is no report containing only these two attributes.

In contrast to most data mining algorithms, the discovery process using FCA is human-centered. It is definitely not a black-box that runs and optimizes without intervention beyond specifying initial model choices and parameters.

3.3.2. Emergent Self Organizing Map

Emergent Self Organizing Maps (ESOM) (Ultsch 2005a, Manyakov et al. 2010) are a special and very recent type of topographic maps (Ritter 1999, Kohonen 1982, Van Hulle 2000). According to (Ultsch 2003), “emergence is the ability of a system to produce a phenomenon on a new, higher level”. In order to achieve emergence, the existence and cooperation of a large number of elementary processes is necessary. An Emergent SOM differs from a traditional SOM in that a very large number of neurons (at least a few thousands) are used (Ultsch 2005b). In the traditional SOM, the number of nodes is too small to show emergence. ESOM is argued to be especially useful for visualizing sparse, high-dimensional datasets, yielding an intuitive overview of their structure (Ultsch 1990). From a practitioner’s point of view, topographic maps are a particularly appealing technique for knowledge discovery in databases (Ultsch 1990, Ultsch 1999) because they perform a non-linear mapping of the high-dimensional data space to a low-dimensional space, usually a two-dimensional one, which facilitates the visualization and exploration of the data (Ultsch 2004). In the past, we applied the ESOM to a police dataset and found its performance to be superior to that of a spherical SOM tool (Poelmans 2009b). We made some interesting discoveries using the ESOM, although the obtained results were limited and not convincing enough to make it into operational policing practice (Poelmans 2009a).

It is claimed by Ultsch and co-workers that the topology preservation of the traditional SOM projection is of little use when the maps are small: the performance of a small SOM is argued to be almost identical to that of a k-means clustering, with k equal to the number of nodes in the map (Ultsch 2005a). Using large numbers of neurons, as in the ESOM, permits one to observe data at a higher level capturing the overall structures, disregarding
the elementary ones and allowing the consideration of structures that otherwise would be invisible.

3.3.2.1. Emergent SOM

An ESOM map is composed of a set of neurons \( I \), arranged in a hexagonal topology map or lattice. A neuron \( n \in I \) is a tuple \((\mathbf{w}_n, p_n)\) in the map, consisting of a weight vector \( \mathbf{w}_n = (w_{n1}, \ldots, w_{nm}) \) with \( \mathbf{w}_n \in \mathbb{R}^m \) and a discrete position \( p_n \in P \), where \( P \) is the map space. The data space \( D \) is a metric subspace of \( \mathbb{R}^m \). The training set \( E = \{x_1, \ldots, x_k\} \) with \( x_1, \ldots, x_k \in \mathbb{R}^m \) consists of input samples presented during the ESOM training. The training algorithm used is the online training algorithm in which the best match for an input vector is searched for, and the corresponding weight vectors, and also those of its neighboring neurons of the map, are updated immediately.

When an input vector \( \mathbf{x} \) is supplied to the training algorithm, the weight \( \mathbf{w}_n \) of a neuron \( n \) is modified as follows:

\[
\Delta \mathbf{w}_n = \eta h(bm, n, r)(\mathbf{x} - \mathbf{w}_n)
\]

with \( \eta \in [0,1] \), \( r \) the neighborhood radius and \( h \) a non-vanishing neighborhood function. The best-matching neuron of an input vector \( \mathbf{x} \in D \)

\[
D \rightarrow I : bm = bm(\mathbf{x})
\]

is the neuron \( n_{bm} \in I \) having the smallest Euclidean distance to \( \mathbf{x} \):

\[
n_{bm} = bm(\mathbf{x}) \iff d(\mathbf{x}, \mathbf{w}_n) \leq d(\mathbf{x}, \mathbf{w}_b) \forall \mathbf{w}_b \in W.
\]

Where \( d(\mathbf{x}, \mathbf{w}_b) \) stands for the Euclidean distance of input vector \( \mathbf{x} \) to weight vector \( \mathbf{w}_b \). The neighborhood of a neuron

\[
N_r = N(n) = \{ n \in M \mid h_n(r) \neq 0 \}
\]

is the set of neurons surrounding neuron \( n \) and determined by the neighborhood set \( h \). The neighborhood defines a subset in the map space of the neurons \( K \), while \( r \) is called the neighborhood range.

The map produced maintains the neighborhood relationships that are present in the input space and can be used to visually detect clusters. It also provides the analyst with an idea of the complexity of the dataset, the distribution of the dataset (e.g. spherical) and the amount of overlap between
the different classes of objects. The lower-dimensional data representation is also an advantage when constructing classifiers. ESOM maps can be created and used for data analysis by means of the publicly available Databionics ESOM Tool (Databionics 2009). With this tool the user can construct ESOMs with either flat or unbounded (i.e. toroidal) topologies.

3.3.2.2. ESOM parameter settings

To simulate the ESOM, we used the Databionics software and its standard parameter settings (Databionics 2010). We did not attempt to optimize them. A SOM with a lattice containing 50 rows and 82 columns of neurons was used (50x82=4100 neurons in total). The weights were initialized randomly by sampling a Gaussian with the same mean and standard deviation as the corresponding features. A Gaussian bell-shaped kernel with initial radius of 24 was used as a neighborhood function. Further, an initial learning rate of 0.5 and a linear cooling strategy for the learning rate were used. The number of training epochs was set to 20. In the map displayed in Figure 3.8, the best matching (nearest-neighbor) nodes are labeled in the two classes for the given test data set (red for domestic violence, green for non-domestic violence). The red squares in all figures represent neurons that mainly contain domestic violence reports, whereas the green squares represent neurons that mainly contain non-domestic violence reports. The U-Matrix (Ultsch et al. 2005) is used as background visualization in the ESOM. The local distance structure is displayed at each neuron as a height value creating a 3D landscape of the high-dimensional data space. The height is calculated as the sum of the distances to all immediate neighbors normalized by the largest occurring height. This value will be large in areas where no or few data points reside (white color) and small in areas of high densities (blue and green color).

3.3.3. C-K theory

The Concept-Knowledge theory (C-K theory) was initially proposed by Hatchuel et al. (1999), Hatchuel et al. (2002) and further developed by Hatchuel et al. (2004). C-K theory is a unified design theory that defines design reasoning dynamics as a joint expansion of the Concept (C) and Knowledge (K) spaces through a series of continuous transformations within and between the two spaces (Hatchuel 2003). C-K theory makes a formal distinction between Concepts and Knowledge: the knowledge space consists of propositions with logical status (i.e. either true or false) for a designer, whereas the concept space consists of propositions without logical status in the knowledge space. According to Hatchuel et al. (2003), concepts have the potential to be transformed into propositions of K but are not themselves elements of K. The transformations within and between the concept and
knowledge spaces are realized by the application of four operators: concept $\rightarrow$ knowledge, knowledge $\rightarrow$ concept, concept $\rightarrow$ concept and knowledge $\rightarrow$ knowledge. These transformations form what Hatchuel calls the design square, which represents the fundamental structure of the design process. The last two operators remain within the concept and knowledge spaces. The first two operators cross the boundary between the Concept and Knowledge domains and reflect a change in the logical status of the propositions under consideration by the designer (from no logical status to true or false, and vice versa).

![Design square](adapted from (Hatchuel 2003))

Design reasoning is modeled as the co-evolution of C and K. Proceeding from K to C, new concepts are formed with existing knowledge. A concept can be expanded by adding, removing or varying some attributes (a “partition” of the concept). Conversely, moving from C to K, designers create new knowledge either to validate a concept or to test a hypothesis, for instance through experimentation or by combining expertise. The iterative interaction between the two spaces is illustrated in Figure 3.3. The beauty of C-K theory is that it offers a better understanding of an expansive process. The combination of existing knowledge creates new concepts (i.e. conceptualization), but the activation and validation of these concepts may also generate new knowledge from which once again new concepts can arise.

However, one of the reasons why it is hard to apply traditional C-K theory in practice is that it lacks an actionable definition of the notions concept, partition and inclusion. In this chapter, we show that these issues can be resolved by implementing the C-K framework with a synergistic combination
of FCA and ESOM for modeling and expanding the space of concepts. One of the limitations of traditional C-K theory is that hierarchical representations are used to model and expand the concept space. These hierarchical representations are limited in their semantic expressiveness, which is one of the reasons why we chose for the non-hierarchical concept representation of FCA. Complementary to FCA, the ESOM functions as a catalyst to make the knowledge discovery process with FCA more efficient. One of the issues we encountered while using FCA was the scalability of the techniques for larger datasets. We choose to solve this problem by using the ESOM maps, which provide a clear picture of the overall distribution of the entire dataset and the available clusters. The combination of the maps and lattices allows for an efficient exploration of the data, leading, amongst other things, to a better selection of police reports for in-depth manual inspection.

3.4. Instantiating C-K theory with FCA and ESOM

In this section, we elaborate on the applied process for knowledge discovery based on the visually appealing discovery techniques presented in section 3.3. FCA as a standalone technique suffers from scalability issues when the number of attributes is increased. Exploring high-dimensional data and discovering new concepts with FCA while little is known about the contents is a difficult task. Although the ESOM can provide some insights into the overall distribution of the data and may help in discovering new concepts and knowledge in the data, its capacities for knowledge discovery are limited. The ESOM as a standalone technique does not allow gaining thorough insights into the conceptual structure of the data and the underlying knowledge of police officers. This is important since we want to improve our understanding of the gaps in the current domestic violence definition, the knowledge of police officers concerning the problem, etc. In this chapter, we go beyond the use of either one of these techniques and use them in combination as part of a unifying framework based on C-K theory. The unifying framework gives insight into the generic nature of the KDD activity and is a necessary precondition for successfully embedding the knowledge discovery process based on the synergistic combination of FCA and ESOM in daily policing practice. In this setup, FCA is used as a concept engine, distilling formal concepts from unstructured text. We complement knowledge discovery with the capabilities of ESOM, which functions as a catalyst for the FCA based knowledge extraction. Our approach to knowledge discovery is framed in C-K theory. The K space could be viewed as being composed of actionable information. It contains the existing knowledge used to operate and steer the action environment. The C space, on the other hand, can be considered as the design space. Whereas K is used as the basis for action and decision making, C puts this actionability under scrutiny for potential improvement and learning. At the basis of the knowledge discovery process are many fast iterations through the C-K loop.
During the mining process, two persons, an exploratory data analyst and a domain expert are the driving force behind the exploration and collaborate intensively. There is a continuous process of iterating back and forth between the FCA lattices, the ESOM maps and the police reports. The knowledge discovery process using the combination of FCA and ESOM is summarized in Figure 3.4. It basically consists of iteratively applying the four operators from the design square in Figure 3.3.

Initially, an FCA lattice and an ESOM map are constructed by the exploratory data analyst based on the domain expert’s prior knowledge of the problem area, the police reports contained in the dataset and the terms contained in the thesaurus (i.e. $K \rightarrow C$). The lattice and the ESOM map provide a reduced search space to the domain expert, who then visually inspects and analyzes the lattice and ESOM map (i.e. $C \rightarrow C$). The synergistic combination of FCA and ESOM can be considered as a knowledge browser. Our contention is that it allows for an effective interaction between the human actors and the underlying information. Using FCA, police reports are selected for in-depth manual inspection based on observed anomalies and counter-intuitive facts (i.e. $C \rightarrow K$). Using the ESOM map, police reports are selected based on the analysis of outliers, clusters and areas of the map containing a mixture of domestic and non-
domestic violence cases (i.e. C → K). These police reports are then used to discover new referential terms to improve the thesaurus, to enrich and validate prior domain knowledge, to discover new classification rules or for operational validation (i.e. K → K).

Additionally, based on the classification rules discovered using FCA, we label/relabel cases and use these cases to construct an ESOM risk analysis map. We then project the unlabeled cases onto this map (i.e. K → C). Subsequently, this map is analyzed by the exploratory data analyst and the domain expert, who search the map for outliers, clusters of cases in different areas of the map and areas containing a mixture of domestic and non-domestic violence cases (i.e. C → C). Based on the observations made, representative police reports are again selected for in-depth manual inspection (i.e. C → K). The obtained results, together with the relevant prior knowledge of the domain expert, are then incorporated into the existing visual representation, resulting in a new lattice and ESOM map (i.e. K → C).

3.5. Dataset

Our dataset consists of a selection of 4814 police reports describing a whole range of violent incidents from the year 2007. All domestic violence cases from that period are a subset of this dataset. The selection came about amongst others by filtering out those police reports that did not contain the reporting of a crime by a victim, which is necessary for establishing domestic violence. This happens, for example, when police officers are sent to investigate an incident and afterwards write a report in which they mention their findings, but the victim ends up never making an official statement to the police. The follow-up reports referring to previous cases were also removed. From the 4814 police reports contained in the dataset the following information was extracted: the person who reported the crime, the suspect, the persons involved in the crime, the witnesses, the project code and the statement made by the victim to the police. Of those 4814 reports, 1657 were classified by police officers as domestic violence. These data were used to generate the 4814 html-documents that were used during our research. An example of such a report is displayed in Figure 3.5.

The validation set for our experiment consists of a selection of 4738 cases describing a whole range of violent incidents from the year 2006 where the victim made a statement to the police. Again, the follow-up reports were removed. Of these 4738 cases 1734 were classified as domestic violence by police officers.

<table>
<thead>
<tr>
<th>Title of incident</th>
<th>Violent incident xxx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reporting date</td>
<td>31-03-2008</td>
</tr>
<tr>
<td>Project code</td>
<td>Domestic violence against ex-partner</td>
</tr>
<tr>
<td>Crime location</td>
<td>Amsterdam Wibautstraat yyy</td>
</tr>
</tbody>
</table>
### Reporting of the crime

*Yesterday morning I was taking a bath. Suddenly my daughter ran into the bathroom followed by her ex-boyfriend. She screamed for help. He had a gun in his hand and he was clearly under the influence of beer or drugs. He yelled out that he couldn’t live without her. He threatened to kill me and my daughter if she wouldn’t come back to their house. The neighbors who were alarmed by all the noise came to lend some help. Meanwhile another neighbor phoned the police. I jumped out of the bath and tried to push him on the floor. During this fight I got some serious injuries on my back etc.*

---

<table>
<thead>
<tr>
<th>Suspect (male) Suspect</th>
<th>Zzz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Amsterdam Waterlooplein yyy</td>
</tr>
<tr>
<td>Involved (male) Involved</td>
<td>Neighbours</td>
</tr>
<tr>
<td>(&gt;45yr)</td>
<td>Amsterdam Wibautstraat www</td>
</tr>
<tr>
<td>Victim (female) Victim</td>
<td>Uuu</td>
</tr>
<tr>
<td>(18-45jr)</td>
<td>Amsterdam Waterlooplein vvv</td>
</tr>
</tbody>
</table>

**3.5. Dataset**

The initial phase of the knowledge acquisition process consists of translating the area under investigation into objects, terms and attributes. We considered the police reports from the dataset as objects and the relevant terms contained in these reports as attributes. The terms and term clusters (see section 3.6) are stored in a thesaurus.

We composed an initial thesaurus of which the content was based on expert prior knowledge such as the domestic violence definition. We enriched the thesaurus with terms referring to the different components of the definition such as “hit”, “stab”, “my mother”, “my ex-boyfriend”. Since domestic violence is a phenomenon that according to the literature typically occurs inside the house, we also added terms such as “bathroom”, “living room”. We made an explicit distinction from public locations such as “under the bridge”, “on the street”. The initial thesaurus contained 123 elements.

The reports were indexed using this thesaurus. For each report the thesaurus elements that were encountered were stored in a collection. This collection would be used as input for both the FCA and the ESOM procedure. The thesaurus was refined after each iteration of re-indexing the reports and visualizing and analyzing the data with the FCA lattice and ESOM maps. This process is demonstrated in detail in section 3.6.
3.5.1. Data preprocessing and feature selection

Our initial steps consisted of data preprocessing and applying traditional classification techniques. We have applied feature selection to reduce the input space dimensionality, prior to applying the ESOM tool. We chose to select the 65 most relevant features. Feature selection comprises the identification of the most characterizing features of the observed data. Given the input data $D$ consisting of $N$ samples and $M$ features $X = \{x_i, i = 1 \ldots M\}$, and the target classification variable $c$, the feature selection problem is to find from the $M$-dimensional observation space, $R^M$, a subspace of $m$ features, $R^m$, that optimally characterizes $c$. A heuristic feature selection procedure, known as minimal-redundancy-maximal-relevance (mRMR), as described in (Peng 2005), was considered. In terms of mutual information $I$, the purpose of feature selection is to find a subset $S$ with $m$ features $\{x_i\}$, which jointly have the largest dependency on the target class $c$. This is called the Max-Dependency scheme:

$$\text{Max } D(S, c), D = I(x_1, \ldots, x_m; c) \quad (1)$$

As the Max-Dependency criterion is hard to implement, an alternative is to select features based on maximal relevance criterion (Max-Relevance). Max-Relevance is to search features satisfying (2), which approximates $D(S, c)$ in (1) with the mean value of all mutual information values between individual feature $x_i$ and class $c$:

$$\max D(S, c), D = \frac{1}{|S|} \sum_{x \in S} I(x; c) \quad (2)$$

Features selected according to Max-Relevance could have redundancy, i.e., the dependency among these features could be large. When two features highly depend on each other, the respective class-discriminative power would not change much if one of them was removed. Therefore, the following minimal redundancy (Min-Redundancy) condition can be added to select mutually exclusive features (Ding 2003):

$$\min R(S), R = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j) \quad (3)$$

The criterion combining the above two constraints is called “minimal-redundancy-maximal-relevance (mRMR). The operator $\Phi(D, R)$ is defined to combine $D$ and $R$ and the following is the simplest form to optimize $D$ and $R$ simultaneously:

$$\max \Phi(D, R), \Phi = D - R \quad (4)$$
3.5. DATASET

The outcome of this filter approach is a ranked list of features. To decide on where to cut off this list we use the classifiers discussed in the next section.

3.5.2. Initial classification performance

To obtain the optimal feature set, an SVM, a Neural Network, a kNN (k-nearest-neighbor with k=3) and a Naïve Bayes classifier were used to measure the classification performance for an increasing number of features.

Naïve Bayes is based on the Bayes rule and assumes that feature variables are independent of each other given the target class. Given a sample \( s = \{x_1, \ldots, x_m\} \) for \( m \) features, the posterior probability that \( s \) belongs to class \( c_i \) is

\[
p(c_i | s) \approx \prod_{j=1}^{m} p(x_j | c_i)
\]

where \( p(x_j | c_i) \) is the conditional probability table learned from examples in the training process. Despite the conditional independence assumption, Naïve Bayes has been shown to have good classification performance for many real data sets (Cover 1991). We have used the WEKA package (Weka 2009). We used 10-fold cross-validation.

The Support Vector Machine (SVM) (Vapnik 1995) is a more modern classifier that uses kernels to construct linear classification boundaries in higher dimensional spaces. We make use of the LibSVM package (Hsu 2002). A Radial Basis Function (RBF) was chosen as kernel, the kernel parameter was set to 0.01 and 10-fold cross-validation was used.

Nearest neighbor methods estimate the probability \( p(t|x) \) that an input vector \( x \in \mathbb{R}^n \) belongs to class \( t \in \{0,1\} \) by the proportion of training data instances in the neighborhood of \( x \) that belong to that class. The metric used for evaluating the distance between \( a, b \in \mathbb{R}^n \) is the Euclidean distance:

\[
dist(a, b) = \| a - b \|_2 = \sqrt{(a-b)^T(a-b)}
\]

The version of \( k \)-nearest neighbor that was implemented for this study was chosen because it is especially appropriate for handling discrete data (Webb 1999). The problem with discrete data is that several training data instances may be at the same distance from a test data instance \( x \) as the \( k \)th nearest neighbor, giving rise to a non-unique set of \( k \)-nearest neighbors. The \( k \)-nearest neighbor classification rule then works as follows. Let the number of training data instances at the distance of the \( k \)th nearest neighbor be \( n_k \), with
data instances of class $t = 1$ and $n_k$ data instances of class $t = 0$. Let the total number of training data instances within, but excluding this distance be $N_k$, with $N_k$ data instances of class $t = 1$ and $N_k$ data instances of class $t = 0$ if

$$N_k + \frac{k - N_k}{n_k} \times n_k \geq N_k + \frac{k - N_k}{n_k} \times n_k$$

where $N_k < k \leq N_k + n_k$. Now all training data instances at the distance of the $k$th nearest neighbor are used for classification, although on a proportional basis. The parameter $k$ was set to 2 and 10-fold cross-validation was used.

We also used a feed-forward multiplayer perceptron (MLP) with one hidden layer consisting of 5 neurons and an output layer consisting of one neuron (Matlab Arsenal 2008). The weight decay parameter was set to 0.2 and the number of training cycles to 10. Again we used 10-fold cross-validation.

The classification performance is plotted as a function of the number of features in Figure 3.6. The result of the mrmr algorithm is a ranked list of the best features. The x-axis indicates how many of these best features were used to train the classifiers. The y-axis shows the classification performance for these different feature subsets. We opted to retain the best 44 features which is a compromise for the 4 classifiers. 44 features was one of the points in the curve where the sum of classification performances for the different classifiers was highest. We also tested other maxima such as 15 and 30 but this resulted in a less qualitative graphical image. A toroidal ESOM map was trained on this dataset with a reduced number of features and was compared to that of Figure 3.8. It shows that the density problem (one class label for each density peak) was not solved by lowering the number of features (result not shown).
In this section, we illustrate the abstract description of the knowledge discovery process provided in section 3.4 with a real life case study with the Amsterdam-Amstelland police on domestic violence. We have chosen not to present the sequential build-up of the lattices and ESOM maps, but to make a selection from these lattices and maps, just to help the reader become familiar with the explorative possibilities of the method presented here.

The process displayed in Figure 3.4 contains an iterative learning loop. During the successive iterations through the C-K loop, multiple interesting results emerged from the research. These different types of results will now briefly be described. The analysis process is showcased in detail in the next subsections. The FCA lattices and ESOM maps are mainly used as an instrument to efficiently select representative reports for in-depth manual inspection, to discover new classification rules, to enrich, test and refine expert prior knowledge, to browse and annotate the collection of police reports, etc.

An important aspect of the process consists in searching these reports for new attributes that can be used to discriminate between the domestic and
non-domestic violence reports or that may lead to an enrichment of existing domain knowledge. New referential terms were not selected using a term extractor, but they were obtained by carefully reading some representative reports and then selecting relevant terms as attributes. We built in the necessary validation mechanisms to ensure the completeness of the thesaurus:

1. Word stemming. Each word is reduced to word-stem form.
2. Stop wording. A stop list is used to delete from the texts the words that are insufficiently specific to represent content. The stop list contains many common function words, such as “the”, “or”, etc.
3. Synonym lists. Synonym lists are used to add semantically similar words.
4. Spelling checking. Spelling checking is used to validate the correctness of the term added to the thesaurus and the correctness of the words in the police reports.

During the research the thesaurus was under constant evolution: when new terms and concepts were discovered, the terms were added to the thesaurus. This approach ensured that the thesaurus remained at all times a reflection of the knowledge already gained. Because of the large number of police reports in the dataset, it was not possible to visually analyze concept lattices containing more than 14 attributes. Therefore, terms with a similar semantic meaning or referring to the same domain concept were clustered by the domain experts. When these term clusters were used to create an FCA lattice, they were considered as attributes.

Appendix B shows an excerpt of the thesaurus obtained after multiple data analysis iterations. The thesaurus contains term clusters such as “geweld”, “familie”, etc., search terms such as “slaan”, “stampen”, etc. and some of these search terms were composed by taking a Cartesian product of the contents of some term clusters. For example, “geweld door mijn familie” was composed by taking a Cartesian product of the contents of “geweld” + “door mijn” + “familie”.

After multiple knowledge enrichment operations the performance of the system was validated by the data quality management team who manually read all cases and indicated us where our method went wrong and how we could improve. An excerpt of such a document is displayed in appendix C. This new knowledge was then used as input for one of the further iterations. The system is currently being maintained in this way by the data quality management team on a continuous basis.

During the exploration, we also verified the correctness of the labels assigned by police officers to the selected cases and we searched the reports for new interesting concepts, inconsistencies, etc. This led amongst others to the discovery of faulty case labelings and situations that were often not recognized by police officers as domestic or as non-domestic violence. This
information was used by the data quality management team to significantly improve the quality of the data in the police databases and to improve the way police officers handle domestic violence cases. The information was also useful for the domestic violence program manager to improve the training of police officers. We also found some regularly occurring confusing situations that could not be uniquely classified as domestic or non-domestic violence based on the domestic violence definition. These situations were presented to the program manager and were used to enrich, improve and refine the concept and definition of domestic violence.

During the discovery and conceptualization of the nature of domestic violence from the data at hand, we were able to define a set of accurate and comprehensible classification rules to automatically classify incoming cases as domestic or as non-domestic violence. In the past developing an accurate classifier using decision trees, SVMs, Neural Networks, etc. turned out to be impossible. We found that this was largely due to the incorrect labels assigned by police officers to cases, to the vagueness of the domestic violence definition and to the lack of a high-quality thesaurus. We managed to resolve many of these problems during the exploration with FCA and ESOM, resulting in a set of highly accurate and comprehensible classification rules. All these different aspects of the process, which have only been briefly introduced so far, are discussed more extensively in the next sections.

3.6.1. Transforming existing knowledge into concepts

The process of design reasoning starts by making the transition from the knowledge space to the concept space. The process of transforming propositions of K into concepts of C is called disjunction. The corresponding operator in the design square from Figure 3.3 is the knowledge → concept operator. This operator expands the space of C with elements from K. We used two techniques to perform this knowledge to concept transformation. First, we constructed an FCA lattice based on expert prior knowledge, the police reports in the dataset and the term clusters in the thesaurus. Second, we designed an ESOM map based on the terms in the thesaurus and the police reports in the dataset. Both methods are further discussed in this section.

The definition of domestic violence employed by the police organization of the Netherlands is as follows:

“Domestic violence can be characterized as serious acts of violence committed by someone in the domestic sphere of the victim. Violence includes all forms of physical assault. The domestic sphere includes all partners, ex-partners, family members, relatives and family friends of the victim. The notion of family friend includes persons that have a friendly relationship with the victim and (regularly) meet with the
The lattice in Figure 3.7 was fundamentally influenced by this domestic violence definition. Prior to the analysis with FCA, certain terms were clustered in term clusters based on this definition and added to the thesaurus. We clustered the terms contained in the thesaurus into term clusters associated with one of the two components of the definition (i.e. prior knowledge incorporation).

We first attempted to verify whether a report could be classified as domestic violence by checking it for the occurrence of one or more terms related to each of the two components of the domestic violence definition. In other words, a case would be labelled as domestic violence if the following two conditions were fulfilled. First, a criminal offence had occurred. To verify whether a criminal offence had occurred, the report was searched for terms such as “hit”, “stab” and “kick”. These terms were grouped into the term cluster “acts of violence”. Second, a person in the domestic circle of the victim was involved in the crime. Therefore, the report was searched for terms such as “my dad”, “my ex-boyfriend” and “my uncle”. These terms were grouped into the term cluster “persons of domestic sphere”. It should be noted that a report is always written from the point of view of the victim and not from the point of view of the officer. A victim always adds “my”, “your”, “her” and “his” when referring to the persons involved in the crime. Therefore, the report is searched for terms such as “my dad”, “my mom” and “my son”. These terms are grouped into the term cluster “family members”. The report is also searched for terms such as “my ex-boyfriend”, “my ex-husband”, and “my ex-wife”. These terms are grouped into the term cluster “ex-partners”. Furthermore, the report is searched for terms such as “my nephew”, “her uncle”, “my aunt”, “my step-father” and “his step-daughter”. These terms are grouped under the term cluster “relatives.” Then the report is searched for terms such as “family friend” and “co-occupant”. These terms are grouped into the term cluster “family friends”. Reports that were assigned the label “domestic violence” have been classified as such by police officers. The remaining reports were categorized as non-domestic violence. This results in the lattice displayed in Figure 3.7.
Indexing the 4814 reports from 2007 with the initial thesaurus from section 3.5 resulted in a cross table with all reports as objects and all terms as attributes. This cross table is used for training a toroidal ESOM. The ESOM is represented in Figure 3.8: the green squares refer to neurons that dominantly contain non-domestic violence cases, while the red squares refer to neurons that dominantly contain domestic violence cases.
Using the reference definition of domestic violence employed by the police was but one way to identify term clusters to structure the lattices. Term clusters also emerged from in-depth scanning of certain reports highlighted during a knowledge iteration cycle. This is how, for example, the term cluster “relational problems” was created. We discovered terms such as “relational problems”, “I had a relationship with”, which refer to a broken relationship. A distinction was made between a broken relationship and an ongoing relationship. Terms such as “I have a relationship with” and “live together” were brought together in the cluster “in a relationship”.

According to the literature, domestic violence is a phenomenon that mainly occurs inside the house (Vincent 2000, Black 1999, Beke 2003). Therefore, an attribute called “private locations” was introduced. This term cluster contained terms such as “bathroom”, “living room” and “bedroom”. An attribute called “public locations” was also introduced. The redefined lattice structure, taking into account the analyses of the previous iterations, is displayed in Figure 3.9. In order to keep the lattice comprehensible, the terms
belonging to the clusters “family members”, “relatives”, “partners”, “ex-partners” and “family friends” have been lumped into a cluster “persons”.

**Fig. 3.9. First refined lattice based on the police reports from 2007**

In the analysis of some of the reports selected using ESOM during an earlier iteration, we also found that many cases did not have a formally labelled
suspect. This attribute is also incorporated in the lattice in Figure 3.10. We also found a lot of cases with a description of the suspect. Whether or not perpetrator and victim lived at the same address at the time of the incident was also included as attribute.

![Second refined lattice based on the police reports from 2007](image)

Fig. 3.10. Second refined lattice based on the police reports from 2007

While further exploring the domestic violence reports during successive knowledge creation iterations, it became apparent that in many cases the victim made statements such as “I want to institute legal proceedings against my husband” and “I want to institute legal proceedings against my brother.”
These sentences were brought together into the cluster “legal proceedings against domestic sphere”. Another type of phrasing that was regularly used by victims of domestic violence was, for example, “the crime was committed by my dad” or “the crime was committed by my ex-boyfriend”. These sentences were brought together into the cluster “committed by domestic sphere”. Yet another type of wording that was also frequently used by a victim was phrases such as “I was maltreated by my husband” and “I was threatened by my ex-partner”. These sentences in turn were brought together into the cluster “threatened by domestic sphere”. Finally, neighbourhood quarrels (non-domestic violence) often made reference to phrases such as “I want to institute legal proceedings against my neighbor” and “committed by the man next door”, so these sentences were combined into the cluster “neighbors”. These attributes were included in the lattice of Fig. 3.11.

**Fig. 3.11.** Third refined lattice based on the police reports from 2007

We also use FCA for the validation of some aspects of operational policing practice. For some specific situations it was verified whether police officers disposed of sufficient knowledge about the problem area to recognize these cases as domestic violence. Some very important special domestic violence situations were considered, including incest and honor-related violence. For the first type of situation, reports were searched for terms such as “incest” and “sexual abuse by my father”. For the second type of situation,
reports were searched for terms such as “marriage of convenience” and “marry off”. The resulting lattice after incorporating these special cases is displayed in Figure 3.12.

![Fourth refined lattice](image)

**Fig. 3.12.** Fourth refined lattice based on the police reports from 2007

### 3.6.2. Expanding the space of concepts

The notion of expansion plays a key role in C-K theory. An analyst’s ability to recognize an expansion can depend on his sensitivity to these opportunities, his training or the knowledge at his disposal. In (Hatchuel 2004) it is stated that expansion is a K-relative notion, which means that its significance depends on the knowledge of a designer or any other observer or user. In this chapter, we argue that FCA and ESOM help analysts recognize and exploit these opportunities. Basically, C space expansion is driven by the analyst’s detection and investigation of anomalies, outliers, clusters and concept gaps with these visual exploration tools. Based on these observations, police reports are selected for in-depth manual inspection. This section describes in more detail these two ways of expanding the space of concepts.

We first explain how we used FCA to expand the space of concepts. FCA was used to efficiently explore the data based on the prior knowledge of the
domain expert. Some interesting findings emerged from the interactive exploration of the lattice in Figures 3.7 – 3.12 and warranted further investigation.

Table 3.2. Interesting observations from the lattices in Figures 3.7 – 3.12

<table>
<thead>
<tr>
<th>Non-domestic violence</th>
<th>Domestic violence</th>
</tr>
</thead>
<tbody>
<tr>
<td>No “acts of violence”</td>
<td>128</td>
</tr>
<tr>
<td>No “acts of violence” and “persons of domestic sphere”</td>
<td>63</td>
</tr>
<tr>
<td>“Acts of violence” and no “persons of domestic sphere”</td>
<td>863</td>
</tr>
<tr>
<td>“Relational problems”</td>
<td>58</td>
</tr>
<tr>
<td>“private locations”</td>
<td>1340</td>
</tr>
<tr>
<td>“public locations”</td>
<td>1015</td>
</tr>
<tr>
<td>Acts of violence and same address</td>
<td>37</td>
</tr>
<tr>
<td>Acts of violence and no suspect and description of suspect</td>
<td>695</td>
</tr>
<tr>
<td>Acts of violence and no suspect “legal proceedings against domestic sphere”</td>
<td>1442</td>
</tr>
<tr>
<td>“committed by domestic sphere”</td>
<td>5</td>
</tr>
<tr>
<td>“threatened by domestic sphere”</td>
<td>4</td>
</tr>
<tr>
<td>“neighbors”</td>
<td>67</td>
</tr>
<tr>
<td>“incest”</td>
<td>7</td>
</tr>
<tr>
<td>“honor-related violence”</td>
<td>2</td>
</tr>
</tbody>
</table>

As can be seen from Table 3.2, a total of 60 domestic violence cases did not contain a term from the “acts of violence” term cluster. Of these 60 cases 18 contained a term from the clusters containing terms referring to a person in the domestic sphere of the victim. Interestingly, some 28% (i.e. 863) of the non-domestic violence reports only contain terms from the “acts of violence” cluster, while there are only 72 domestic violence reports in the dataset that share that characteristic. Apparently, some cases that were labeled as domestic violence did not fit the definition of domestic violence that was used to start this discovery exercise in the first place. The reports in question were therefore selected for in-depth investigation.

It should be clear from the lattice in Figure 3.9 that the terms contained in the cluster “relational problems” tend to be associated with domestic violence cases. Apparently, only 58 non-domestic violence reports contained one or more terms from the “relational problems” cluster. We concluded that the presence of at least one of the terms from this cluster in a police report
seemed to be a strong indication for domestic violence. This was enough evidence to warrant manual inspection of these 58 police reports.

We also used FCA to verify the correctness and the practical usefulness of this prior knowledge. Most of the domestic violence cases under scrutiny (1365 cases or 82%) contained one or more terms from the “private locations” term cluster. However, 1340 (42%) of the non-domestic violence cases also contained one or more terms from this same term cluster. In addition, a hypothesis that was formulated prior to the data exploration was that almost no domestic violence case was expected to have taken place on the street. Surprisingly, this hypothesis was proven incorrect by the data. In about one-fourth of the domestic violence cases there had been an incident at a public location. While scrutinising these police reports, we discovered that this was often the case when ex-partners were involved. It became apparent that it was not possible to distinguish domestic from non-domestic violence reports by means of the type of locations mentioned in the reports. Combining the clusters “private locations” and “public locations” with clusters such as “family members” or “ex-persons”, for example, did not yield the expected results in terms of discriminatory power. We noticed that in a large number of the domestic violence cases (416 cases or 28%) the perpetrator and the victim happened to live at the same address at the time the victim made their statement to the police. Most of these cases (379 cases or 91%) were classified as domestic violence.

Visual inspection of the patterns produced by the ESOM map in Figure 3.8 also allowed us to make some interesting observations. For example, colour coding made it easy to detect outlying observations: some red squares are located in the middle of a large group of green squares and vice versa. For further examination we made use of the ESOM tool’s functionality to select neurons and display the cases that had this neuron as their best match. We thought that these neurons were associated with cases that might have been wrongly classified by police officers. Therefore, these cases were also selected for in-depth manual inspection.

3.6.3. Transforming concepts into knowledge

The concept \( \rightarrow \) knowledge operator from Figure 3.4 transforms concepts in \( C \) into logical questions in \( K \). In our case an answer to such a question is found by manually inspecting the selected police reports. We refer to this manual analysis as the validation of concept gaps, giving rise to multiple types of discoveries: confusing situations, new referential terms, faulty case labelings, niche cases and data quality problems.

For example, and with reference to Table 3.2, the 18 cases labeled by police officers as domestic violence that contained a term from the “persons of domestic sphere” but no violence term were selected for manual inspection. Is it possible that there are domestic violence reports in which the victim does mention a person of the domestic sphere, but does not mention an
act of violence? In-depth analysis showed that these 18 reports contained violence related terms that were originally lacking from the initial thesaurus, such as “abduction”, “strangle” and “deprivation of liberty”. Another example is the discovery of 42 cases that did not contain a violence term or a term referring to a person of the domestic sphere. These cases turned out to be wrongly classified as domestic violence. We also analyzed the reports that contained a violence term but no term referring to a person of the domestic sphere. This inspection revealed that more than two thirds of these reports were wrongly classified as domestic violence. In the next section, we will focus on the causes of these labeling errors and the extraction of actionable intelligence from these individual cases that can be used to improve the domestic violence definition and the training of police officers.

Table 3.2 also indicates that there were 58 police reports that were classified as non-domestic violence while containing a term from the “relational problems” cluster. This investigation revealed that a startling 95% of these cases had been wrongly labeled as non-domestic violence. Moreover, about 70% of these cases had as a common feature that a third person made a statement to the police for someone else. Analysis of the remaining 30% of these misclassified cases led to the discovery of an important new concept that was lacking from the domain expert’s initial definition of domestic violence. Many of the reports included expressions such as “I was attacked by the ex-boyfriend of my girlfriend” and “I was harassed by the ex-girlfriend of my boyfriend”. These terms were grouped into the cluster “attack by ex-person against new friend”. This situation is analyzed in detail in the next section together with the resulting actionable intelligence. The term cluster is also used to distil new classification rules in one of the subsequent iterations.

Another interesting finding emerged from our search for novel and potentially interesting classification attributes. The lattice in Figure 3.10 shows that some 34% of the reports (1623 cases) did not mention a suspect. According to the domestic violence definition (which specifies that the perpetrator must belong to the domestic circle of the victim), the offender has to be known in domestic violence cases. Naturally, we had assumed that these reports described non-domestic violence cases. Nevertheless, when looking into these cases, we found that 181 of them turned out to describe domestic violence cases after all. In the next section, we uncover the causes of this phenomenon. Additionally, we found out that some 44% of the reports (711 cases) that lacked a labelled suspect did contain a description of the actual suspect. Of these 711 cases, only 16 reports were classified as domestic violence. After studying these 16 reports, we discovered that the majority of them were wrongly classified as domestic violence.

When studying the remaining 37 non-domestic violence cases more carefully, we found, much to our surprise, that the perpetrator and the victim often lived together in the same institution (e.g. a youth institution, a prison or a retirement home). It turned out that of the 41 cases where the perpetrator
and the victim lived in the same institution only 30 actually had been
classified as cases of domestic violence. The non-domestic violence cases
where the perpetrator and the victim lived at the same address and were not
inhabitants of an institution turned out to be wrongly classified as non-
domestic violence. Therefore, a new attribute called “institution” was
introduced. After browsing the 19 non-domestic violence cases in which the
victim used one or more terms from the “legal proceedings against domestic
sphere” cluster, it turned out that these reports should have been classified as
domestic violence. The same observation was made when the 5 non-domestic
violence reports containing a term from the “committed by domestic sphere”
cluster and the 4 non-domestic violence cases containing a term from the
“threatened by domestic sphere” cluster were analyzed. In-depth
investigation of the 5 domestic violence cases in which a term from the
“neighbours” cluster occurred, showed that these reports should have been
classified as non-domestic violence.

After an in-depth manual inspection of the police reports corresponding to
the ESOM outliers, interesting discoveries were made. For example, we
observed that many of these outlier reports contained several important new
features that were lacking in the domain expert’s understanding of the
problem area. Every time new and important features were discovered in this
way, they were used to enrich the thesaurus. A selection of these features is
displayed in Table 3.3 and 3.4.

Table 3.3. Newly discovered features by studying the domestic violence
outliers in the ESOM map.

<table>
<thead>
<tr>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pepper spray</td>
</tr>
<tr>
<td>Homosexual relationship, lesbian relationship</td>
</tr>
<tr>
<td>Sexual abuse, incest</td>
</tr>
<tr>
<td>Alternative spelling of some words (e.g. ex-boyfriend, exboyfriend, ex boyfriend)</td>
</tr>
<tr>
<td>Weapons lacking in the thesaurus: belt, kitchen knife, baseball bat, etc.</td>
</tr>
<tr>
<td>Terms referring to persons: partner, fiancée, mistress, concubine, man next door, etc.</td>
</tr>
<tr>
<td>Terms referring to relationships: romance, love affair, marriage problems, divorce proceedings, etc.</td>
</tr>
<tr>
<td>Reception centers: woman’s refuge center, home for battered woman, etc.</td>
</tr>
<tr>
<td>Gender of the perpetrator: mostly male</td>
</tr>
<tr>
<td>Gender of the victim: mostly female</td>
</tr>
<tr>
<td>Age of the perpetrator: mostly older than 18 years and younger than 45 years</td>
</tr>
<tr>
<td>Age of the victim: mostly older than 18 years and younger than 45 years</td>
</tr>
<tr>
<td>Terms referring to an extra marital affair: I have an another man, lover, I am unfaithful, etc.</td>
</tr>
</tbody>
</table>
Table 3.4. Newly discovered features by studying the non-domestic violence outliers in the ESOM map.

<table>
<thead>
<tr>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Places of entertainment: club, disco, bar, etc.</td>
</tr>
<tr>
<td>Crime locations: on the street, on a bridge, under a viaduct, on a crossing, etc.</td>
</tr>
<tr>
<td>Public locations: metro station, bus stop, tram stop, etc.</td>
</tr>
<tr>
<td>Reception centers: refugee center, shelter for the homeless, relief center, etc.</td>
</tr>
<tr>
<td>Drugs: drug abuse, drug joint, etc.</td>
</tr>
<tr>
<td>Addresses of youth institutions, prisons, etc.</td>
</tr>
<tr>
<td>Hotel: hotel room, hotel, etc.</td>
</tr>
<tr>
<td>Description of suspect’s origin: Turkish descent, white man, North-African descent, etc.</td>
</tr>
<tr>
<td>Description of suspect’s body: 1.75 meters tall, 119 centimeters tall, muscular appearance, etc.</td>
</tr>
<tr>
<td>Description of suspect’s hair: curly haired, blond hair, redhead, etc.</td>
</tr>
<tr>
<td>Description of suspect’s clothes: black jacket, leather shoes, blue pants, jeans, etc.</td>
</tr>
<tr>
<td>Description of suspect’s face: beard, moustache, facial hair, etc.</td>
</tr>
<tr>
<td>Description of suspect’s accent</td>
</tr>
<tr>
<td>Unknown person is involved in the crime</td>
</tr>
<tr>
<td>Attack by unknown person</td>
</tr>
<tr>
<td>Corporate body</td>
</tr>
<tr>
<td>Neighborhood quarrel</td>
</tr>
</tbody>
</table>

The reports also contained multiple confusing situations. When more detailed information was disclosed to us, these cases were also used to refine the domestic violence definition.

3.6.4. Expanding the space of knowledge

The expansion of the space K constitutes validation or testing of the proposed expansion with the ultimate goal of producing actionable intelligence. K-validation of a concept boils down to a confrontation of the output from the C-K transformation with knowledge sources available to the K space (e.g. cross-checking with other databases, setting up field experiments, soliciting expert advice). These new propositions have logical status. In this section, we show how we obtain actionable intelligence from the observations made during the Concept → Knowledge phase.

Analysis of the misclassified police reports described in the previous section revealed that for some unknown reason police officers regularly seem to misclassify burglary, car theft, bicycle theft and street robbery cases as domestic violence. Therefore, terms such as “street robbery”, burglary” and “car theft” were combined into a new term cluster called “burglary cases”.

In the previous section, we also described how the analysis of the police reports revealed that a situation in which a third person makes a statement for somebody else can be confusing for police officers. For example, one case described a father who made a statement to the police about the sexual abuse of his daughter by her stepfather. This is a clear case of domestic violence, but since it was not the victim who made the statement to the police, the police officer did not recognize it as such. This type of situation is now specifically addressed in police training.

In the previous section, we also described how the analysis of police reports revealed interesting cases in which the ex-boyfriend attacked the new boyfriend. We presented these ambiguous cases to the board members responsible for the domestic violence policy. Police officers and policy makers confirmed that this type of situation was to be seen as domestic violence, mainly because the perpetrator often intends to emotionally hurt the ex-partner. Consequently, the expectation was for the terms contained in this cluster to frequently occur in domestic violence reports. However, this turned out to be incorrect. It became clear from the investigation that in general this type of situation was very confusing to police officers. A quick scan revealed that more than 50% of police officers actually had trouble with such cases. The ensuing investigation and discussions with police officers and policy makers revealed that this situation needed to be addressed during the training of police officers. Several interesting cases like the previous one were identified during the data exploration. All of them resulted in a clearer insight into the nature of domestic violence.

In the previous section, we found that some domestic violence cases did not mention a formally labeled suspect. Analysis revealed that this was a result of police officers’ rather haphazard ways of registering suspects for these cases. Apparently, while some officers immediately registered a suspect at the moment the victim mentioned this person as a suspect, others preferred to first interrogate these suspects before officially labeling them as such. In the latter case, the person would just be added to the list of persons who were said to be involved in or to have witnessed the crime. Because such lists included friends, family members or bystanders, they could potentially be very extensive and diverse, which is why suspects easily got lost in these lists. When we inquired about the proper policy regarding the labeling of suspects, we were told there simply was none. Our analysis made a strong case for the need for such a policy. In the end, the quick-win proposal that could be implemented to solve this issue involved a relatively simple change to the registration software: an additional data entry field would need to be introduced for police officers to register the persons that were mentioned by the victim as offenders.

The same address finding brought about a lively discussion amongst the police officers of the Amsterdam police force. More importantly, it exposed the discord amongst police officers on how to classify such cases. We took
note of all their reflections and presented them to the board members responsible for the domestic violence policy. After intensive debate the classification guidelines, displayed in Table 3.5, were obtained. Careful inquiry into the incest and honor-related violence cases taught us that police officers regularly misclassified incest cases as non-domestic violence. On the other hand, even for insiders it was quite surprising to observe how almost all honor-related violent incidents ended up being correctly classified as domestic violence. The latter was probably attributable to the intensive sensitization campaigns organized to inform police officers of this important societal problem.

The newly obtained knowledge led to a new iteration of the FCA analysis, supported by another run of the ESOM tool. In each iteration, it is possible that one or more new classification rules are discovered. The attribute “corporate body”, for example, was found by first analyzing a cluster of green squares that was located within a group of red squares in an ESOM map. With FCA we found that the presence of a corporate body in a police report almost always excludes domestic violence. Therefore, we introduced a new domestic violence classification rule named “corporate body”. An other example of a classification rule is when a case has no formally labeled suspect and contains a description of a suspect, it can be labeled as non-domestic violence.

### 3.7. Actionable results

Several iterations through the design square resulted in truly valuable upgrades of the K space from the perspective of improving action in the field. This section provides an overview of some of the most important achievements of our work.

First, we were able to refine the definition of domestic violence that would act as a principle guideline for labeling cases. During the exploration, several types of niche cases were identified as valid exceptions to the general definition. No clear labeling guidelines were available, so we formulated advice, grounded in evidence, to redesign the general policy. Eventually, we obtained the classification guidelines displayed in Table 3.5.

<table>
<thead>
<tr>
<th>Perpetrator</th>
<th>Victim</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caretaker Inhabitant</td>
<td>Inhabitant Caretaker</td>
<td>Domestic violence</td>
</tr>
<tr>
<td>Inhabitant younger than 18y</td>
<td>Inhabitant younger than 18y</td>
<td>Domestic violence</td>
</tr>
</tbody>
</table>
In the end the presence or absence of a dependency relationship between the perpetrator and the victim was the decisive factor for classifying a case as either domestic or as non-domestic violence. Nevertheless, we also discovered some regularly occurring situations in which there is a clear dependency relationship between the perpetrator and the victim, but that were typically classified as non-domestic violence by police officers. A selection of these circumstances is listed in Table 3.6. These confusing situations helped to expose the mismatch between the management’s conception of domestic violence and that of police officers. We found that the management employed a much broader definition of domestic violence than most police officers.

**Table 3.6. Circumstances in which the offender abuses the dependency relationship with the victim, but that are not recognized by police officers as domestic violence.**

<table>
<thead>
<tr>
<th>Circumstance</th>
<th>Dependency relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lover boys</td>
<td>The victim is in love with the lover boy, who abuses this dependency relationship to force her into prostitution.</td>
</tr>
<tr>
<td>Extramarital relationship</td>
<td>If the mistress of an adulterer blackmails him, for example by threatening to reveal their affair to his wife, the mistress abuses the dependency relationship that exists between her and the man.</td>
</tr>
<tr>
<td>Violence between a caretaker and an inhabitant of an institution</td>
<td>If the caretaker threatens or harasses the inhabitant (for example, a nurse who maltreats an elderly woman in a retirement home), the latter is often helpless because she depends on the caretaker.</td>
</tr>
<tr>
<td>Violence between colleagues</td>
<td>If two colleagues had a relationship and one keeps stalking the other, this is domestic violence between ex-persons.</td>
</tr>
<tr>
<td>An ex-boyfriend attacks the new boyfriend</td>
<td>This is considered to be domestic violence because the ex-boyfriend often intends to emotionally hurt his ex-girlfriend.</td>
</tr>
<tr>
<td>Third person makes statement to the police</td>
<td>Police officers regularly fail to recognize cases in which a third person makes a statement to the police.</td>
</tr>
</tbody>
</table>
Second, a set of 22 domestic violence and 15 non-domestic violence classification rules were extracted. Using these rules, 75% of cases from the year 2007 could be labeled automatically as either domestic or non-domestic violence. We also applied these rules to two validation sets containing unstructured police reports from the year 2006 and from the year 2008, which yielded similar results, i.e. 72% and 73% respectively. These rules are now fully operational and used to automatically and correctly classify the majority of incoming cases, while in the past all cases had to be dealt with manually. Ten of these domestic violence and five of these non-domestic violence classification rules are displayed in Table 3.7

Table 3.7. Excerpt of discovered classification rules

<table>
<thead>
<tr>
<th>Domestic violence classification rules</th>
<th>Non-domestic violence classification rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Legal proceedings against domestic sphere</td>
<td>1 Unknown perpetrator</td>
</tr>
<tr>
<td>2 Committed by domestic sphere</td>
<td>2 Corporate body</td>
</tr>
<tr>
<td>3 Relational problems and living together</td>
<td>3 Burglary cases</td>
</tr>
<tr>
<td>4 Relational problems and institutions</td>
<td>4 Road rage</td>
</tr>
<tr>
<td>5 Honor related violence</td>
<td>5 Violence at school</td>
</tr>
<tr>
<td>6 Incest</td>
<td></td>
</tr>
<tr>
<td>7 (Court) injunction</td>
<td></td>
</tr>
<tr>
<td>8 Fear of domestic sphere</td>
<td></td>
</tr>
<tr>
<td>9 Attack by ex-person against new friend</td>
<td></td>
</tr>
<tr>
<td>10 Problems with domestic sphere</td>
<td></td>
</tr>
</tbody>
</table>

Third, the set of newly identified classification rules did not just allow the police to classify incoming cases. The rules could also be employed to reclassify cases from the past to result in more correct performance management and reporting over time. Domestic violence cases that were not recognized as such in the past might also be re-opened for investigation. In total, we found 420 filed reports that were wrongly labeled as domestic violence and 912 filed reports that were wrongly labeled as non-domestic violence. Table 3.8 presents an overview of these results.
Table 3.8. Number of filed reports that were incorrectly classified, but corrected by means of the 37 rules

<table>
<thead>
<tr>
<th>Year</th>
<th>Non-domestic corrected to Domestic</th>
<th>Domestic corrected to Non-domestic</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td>307</td>
<td>136</td>
<td>443</td>
</tr>
<tr>
<td>2007</td>
<td>290</td>
<td>115</td>
<td>405</td>
</tr>
<tr>
<td>2008</td>
<td>315</td>
<td>169</td>
<td>484</td>
</tr>
<tr>
<td>Total</td>
<td>912</td>
<td>420</td>
<td>1332</td>
</tr>
</tbody>
</table>

Finally, based on the cases that could be labeled using the classification rules that were discovered, we constructed an ESOM risk analysis map. For each neuron, the number of domestic and non-domestic violence cases contained in the neuron and the 32 surrounding neurons was counted and used to calculate the probability that a police report that has this neuron as its best match described a domestic violence incident. For the visualization, a color scheme consisting of 5 different colors was used. Red indicates a 90-100% probability rate of domestic violence, orange a 70-90% probability rate, yellow a 30-70% probability rate, green a 10-30% probability rate and dark green a 0-10% probability rate. The labels of the cases that could not be categorized using the new classification rules were not used to construct this risk analysis map. However, we projected these remaining cases onto this map afterwards. The map for the dataset of the year 2007 is shown in Figure 3.13. Cases that were labeled by police officers as domestic violence are represented as black dots, while the cases that were labeled as non-domestic violence, are represented as light blue dots.
It was remarkable to observe that some of the remaining cases were located in the red area of the map, but were not classified by police officers as domestic violence. About 6.4% of the remaining cases were located in the red area of the map displayed in Figure 3.13. About 22.1% of the cases located in the red area of the map were classified as non-domestic violence by police officers. In-depth analysis of these police reports revealed that the majority of these cases should have been classified as domestic violence. On the other hand, only a small percentage of the cases located in the dark green and green areas of the map were classified as domestic violence by police officers (4.8% and 12.4% respectively). Further scrutiny revealed that all of these cases actually described non-domestic violence incidents.
Table 3.9. Distribution of remaining cases of 2007 over different map areas

<table>
<thead>
<tr>
<th>Domestic violence probability</th>
<th>Map area color</th>
<th>% of remaining cases located in map area</th>
<th>% classified as domestic violence</th>
<th>% classified as non-domestic violence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-10% dark green</td>
<td>28.1%</td>
<td>4.8%</td>
<td>95.2%</td>
<td></td>
</tr>
<tr>
<td>10-30% green</td>
<td>30.0%</td>
<td>12.4%</td>
<td>88.6%</td>
<td></td>
</tr>
<tr>
<td>30-70% yellow</td>
<td>21.5%</td>
<td>37.7%</td>
<td>62.3%</td>
<td></td>
</tr>
<tr>
<td>70-90% orange</td>
<td>14.0%</td>
<td>64.3%</td>
<td>35.7%</td>
<td></td>
</tr>
<tr>
<td>90-100% red</td>
<td>6.4%</td>
<td>77.9%</td>
<td>22.1%</td>
<td></td>
</tr>
</tbody>
</table>

Based on the map displayed in Figure 3.13, a correct label can be automatically assigned to 64.5% of the remaining cases of the year 2007 (i.e. the cases located in the dark green, green and red areas of the map). The other cases (i.e. the cases located in the yellow area of the map) have to be classified manually. A similar result was obtained for the cases of the year 2006 and 2008. Based on the comprehensible classification rules discovered during the knowledge discovery exercise, we developed a Tomcat-based system to assist analysts in their labeling of cases (Elzinga 2009). The system is currently used as a stand-alone application by the data quality management team (i.e. the back office). The long term goal is to make it available to all police officers in the organization (i.e. the front office) to assist them in their labeling of cases.

The labeling process, as performed by the data quality management team, consists of a number of steps that are, to a large extent, automated by the newly introduced system. First, the user can select a set of police reports for labeling (e.g. all police reports from the month October 2008). Subsequently, the classification rules that were discovered during the exploration of the data are applied to the cases. When a case comes in for labeling, the first step consists in verifying whether one of the domestic violence rules is satisfied. If this is the case, the case is classified as domestic violence. If not, it is verified whether one of the non-domestic violence rules is applicable. If this is the case, the case is classified as non-domestic violence. Otherwise, the case is left unclassified. The remaining cases are projected onto the ESOM risk analysis map based on the cases labeled with the FCA rules. Using the combination of the classification rules and the ESOM risk analysis map, 91.0% of cases can be classified automatically and correctly. This is a major improvement compared to the past situation where each incoming case had to be dealt with manually.
3.8. **Comparative study of ESOM and multi-dimensional scaling**

In this section we compare the usability of ESOM and MDS as text exploration instruments in police investigations. Multi-dimensional scaling (MDS) is a method that uses the similarity or dissimilarity among pairs of objects in the original space to represent the objects in a lower dimensional space for visualization purposes. In our case, we have used the classical metric MDS algorithm (Gower 1966) to visualize, in a two-dimensional space, the distribution of police reports in the sense that two reports will be close to each other when their correlation is high (Borg et al. 2005). Both the MDS and ESOM can be used for detecting closely related data points, but each one has its own focus. Contrary to the ESOM, which starts directly from the document vectors, we first have to construct a dissimilarity matrix prior to the MDS calculation. In our case, it is a (symmetric) 4814 x 4814 matrix containing the Euclidean distances between each pair of normalized document vectors. The MDS algorithm (Kruskal et al. 1978), starts from this calculated distance matrix and uses a function minimisation algorithm to find the best configuration in a lower dimension, i.e. a mapping of the original space on a two-dimensional space, thereby minimising the overall error. The error is defined as the sum of the squared differences between the distances in the original space (as present in the Euclidean distance matrix) and the corresponding ones in the lower dimensional space. We used the cmdscale algorithm from the R package for calculating the MDS map (Gower 1966).

The output of an ESOM calculation is different from that of a metric MDS. The metric MDS algorithm concentrates on the largest dissimilarities whereas ESOM concentrates on the largest similarities. ESOM tries to reproduce the topology of the data in a 2D grid, instead of reproducing distances. Similar documents are represented by neighboring neurons in an ESOM, while a distance in an MDS map can be interpreted as an estimate of the true distance between both (Wehrens et al. 2007). The MDS map trained on the same initial dataset is displayed in Figure 3.14. The red dots indicate police reports labelled as domestic violence, whereas green dots indicate police reports labelled as non-domestic violence.
After multiple successive iterations of refining the thesaurus, training a new map, and analyzing the resulting ESOM, our thesaurus contained more than 800 domain-specific terms, term combinations and term clusters. We found that the classification accuracy of the SVM, Neural network, Naïve Bayes and kNN classifiers improved significantly after adding the newly discovered features to the thesaurus. For example, for the SVM, the best classification accuracy on the initial dataset was around 83%, while the best classification accuracy on the dataset with the refined thesaurus was around 89%. These classification accuracies are again plotted in function of the best selected features in Figure 3.15.
During one of the final iterations and before correcting the wrongly labelled cases, we trained a new toroidal ESOM map and MDS map on the dataset based on the refined thesaurus. The resulting map is displayed in Figure 3.16. The resulting MDS map is displayed in Figure 3.17.
Fig. 3.16. Toroid ESOM map trained on the categorical dataset. See text for an explanation of the arrows and circles.

Fig. 3.17. MDS map trained on the categorical dataset.
Comparing the ESOM map of Figure 3.8 to that of Figure 3.16 reveals that the amount of overlap between the two classes has decreased significantly after the refined thesaurus was introduced. The map in Figure 5 shows 3 different clusters that mainly contain cases labelled as domestic violence. When we inspected the cases contained in the top left cluster (circle marked as 1), we found that this cluster mainly contained the burglary cases that for some unknown reason were wrongly labeled by police officers. During analysis of the cluster located at the left and bottom of the map (circle marked as 2) and some of the outliers (arrows), we discovered a large number of situations that were found to be confusing for police officers. Their opinions differed on how these cases should be labelled. No such clusters were found in the MDS map. Finally, we found that the outliers mostly contained cases that were wrongly labeled as either domestic or non-domestic violence. We conclude that ESOM is better suited in our case for knowledge discovery purposes.

3.9. Conclusions

In this chapter, we proposed an approach to knowledge discovery from unstructured text using FCA and ESOM. The approach was framed within C-K theory (i.e. the design square) to provide a deeper understanding of the nature of the exploration process, a process that is essentially human-centered. With this chapter we argued for the discovery capabilities of FCA and ESOM, acting as information browsers in the hands of human analysts. The tools were shown to help analysts proceed with knowledge expansion by progressively looping through the design square in an effective way. We demonstrated the method using a real-life case study with data from the Amsterdam-Amstelland police. The case focused on the problem of distilling concepts indicating domestic violence from the unstructured text in police reports. The data exploration for this case study resulted in several improvements to the way domestic violence cases are dealt with and reported on in practice. This included the implementation of an effective early case filter to identify cases that truly warrant in-depth manual inspection.

Intensive audits of the police databases revealed that many police reports tended to be wrongly classified as domestic or as non-domestic violence. Our approach was used to discover new features that better distinguish domestic from non-domestic violence cases resulting in higher classification accuracy and an improvement of the domestic violence definition. Additionally, we found some regularly occurring situations that were often wrongly labeled as non-domestic violence by police officers (e.g. lover boys). Eventually, we managed to build an accurate and comprehensible classifier that automatically assigns a correct label to more than 90% of incoming cases. Moreover, a large number of cases incorrectly classified in the past were
detected and corrected thanks to this procedure. We have also performed a comparative study of ESOM and MDS for analyzing large amounts of unstructured text. The police officers who tested both techniques are more satisfied of the user interface of the ESOM tool to be more appealing for exploring this vast amount of police reports than the MDS. Moreover the ESOM was able to recognize two extra data clusters that were of significant importance but not found by MDS.

Potentially, in future work one could investigate how iceberg lattices and alpha lattices could be used to prune the FCA lattices. One could also investigate the potential of conceptual scaling for improving scalability of the lattices which is however very labor-intensive.
CHAPTER 4

Expressing Object-Oriented Conceptual Domain Models in Formal Concepts Analysis

Conceptual modeling is about identifying concepts in the domain of interest and relating them to each other. The pivotal notion of FCA is the formal concept which consists of a set of entities and a set of attributes that are closely related. FCA distills concepts from binary data tables called formal contexts and relates them to each other through a partial ordering. This commonality inspired us to investigate possible synergies between both disciplines.

When applying FCA in requirements engineering we don't use the data perspective where the relationships between entities and their attributes are investigated but the process perspective. In this perspective entities are clustered based on their participation in behavioral elements of the model. These behavioral elements can be events as in MERODE or use cases in general. The aim is to use FCA to improve the quality of a conceptual model starting from a process-data matrix.

In a first step we looked at the MERODE methodology which formulates a number of guidelines for conceptual model quality\(^3\). We first show there is an isomorphism between MERODE artefacts and FCA lattices. If the matrix is correctly filled in according to the MERODE rules then the EDG and FCA lattice are isomorphic. We first discuss the case without inheritance. Subsequently, we consider the case when existence dependency and inheritance are combined, and we extend FCA with a third dimension to model both relationships in one lattice.

In the second step we prove the relevance of FCA as an instrument to improve the quality of conceptual models without using MERODE. Given a matrix that has not been filled in according to the MERODE rules, can FCA give us indications for improving the quality of conceptual models? In the C-K design square, what will be the consequences of replacing MERODE by FCA? The FCA lattices allow the users to do a qualitative analysis of the

\(^3\) Part of this chapter has been published in Poelmans, J., Dedene, G., Snoeck, M., Viaene, S. (2010) Using formal concept analysis for the verification of process-data matrices in conceptual domain models, IASTED International Conference on Software Engineering.
conceptual model which differs from the mechanistic quality evaluation by applying the MERODE rules.

4.1. Using Formal Concept Analysis for verification of process – data matrices in conceptual domain models

One of the first steps in a software engineering process is the elaboration of the conceptual domain model. In this section, we investigate how Formal Concept Analysis can be used to formally underpin the construction of a conceptual domain model (Poelmans et al. 2010a). In particular, we demonstrate that intuitive verification rules for process-data matrices can be formally grounded in FCA theory. As a case study, we show that the well-formedness rules from MERODE are isomorphic to the clustering rules in Formal Concept Analysis, and that the relationships in the class diagram are isomorphic to the subconcept-superconcept relationship in FCA.

4.1.1. Introduction

The complexity of most information systems is caused by the complexity of the reality they have to deal with and statements about the required functionality always have some underlying assumption about the real world. Therefore, it is useful to build a real world model prior to the development of an information system (Wand et al. 2002). High quality conceptual models are critical to the success of system development efforts (Jackson 1983).

Unfortunately, developers often encounter problems while elaborating the business domain model (Paige 2002): inconsistencies arise between static and dynamic schemas, object types are missing in the business domain model, the business domain model contains errors, etc. Quality has been identified as one of the main topics in current conceptual modeling research (Moody 2005). Despite this importance, algorithmic approaches to assure conceptual model quality are virtually nonexistent (Poels 2003). In (Lonsdale Systems 2009), the authors suggested to use CRUD-matrices to analyze consistency in conceptual models. However, this topic was only briefly discussed. Can we mathematically ground this type of analysis? Can we find an algorithmic approach to detect missing object types? Can we benefit from an algorithmic method for enforcing consistency in business models? Can we mathematically analyze completeness of models?

In this section, we explore the possibilities of using a technique known as Formal Concept Analysis (FCA) (Stumme 2002, Ganter 1999) for mathematically underpinning the construction and analysis of conceptual models. FCA arose twenty-five years ago as a mathematical theory (Wille 1982). FCA has been used extensively in software engineering research (Tilley et al. 2005) and in particular in the areas of detailed design and software maintenance. It has been used for reorganizing existing class hierarchies and for refactoring and modifying existing code (Snelling 2000,
CHAPTER 4: EXPRESSING OBJECT ORIENTED CONCEPTUAL MODELS

Snelting 1998). FCA has also been used for identifying class candidates in legacy code (Tonella et al. 1999) and to mine legacy source code (Mens et al. 2005). In requirements analysis, FCA has been used to identify class candidates in use case descriptions (Duwel 1999, Duwel et al. 1998) and to reconcile descriptions written by different stakeholders using controlled vocabulary and grammar (Richards et al. 2002, Richards et al. 2002b). More recently FCA has also been used in combination with ontology. Cimiano investigates how FCA and ontologies may complement each other from an application point of view (Cimiano et al. 2004). Bain applies FCA to identify structure in theories (Bain 2003). Within the area of design, FCA has been applied to classes and methods (Godin et al. 1998). However it has never been applied to the earlier stage of conceptual modeling. In this section 4.1 we apply FCA to the combination of object types and processes to validate the relationships captured by the class diagram and to identify missing object types.

The remainder of this section is composed as follows. In section 4.1.2, we introduce the research question. As FCA is in essence a matrix-technique, we discuss the three most frequently used matrix techniques in conceptual domain modeling: the CRUD-matrix from Information Engineering, the entity-event table from OOSSADM and the object-event table from MERODE. Subsequently we discuss how FCA could be used as formal foundation for well-formedness rules for process-data matrices in conceptual modelling. In section 4.1.3, we introduce the pivotal notions of FCA theory. As MERODE is the only method that defines well-formedness rules for a CRUD-like matrix, in section 4.1.4, we discuss the essentials of MERODE. In section 4.1.5, the close relationship between FCA and these well-formedness rules is investigated. Section 4.1.6 concludes our discussion.

4.1.2. Research Question

In this section, we elaborate on the three most frequently used matrix techniques in object-oriented conceptual domain modeling.

The Create, Read, Update and Delete (CRUD)-matrix was initially introduced in information engineering by Martin (1990). The purpose of this matrix is to illustrate the relationships between objects and the processes in which they participate. A process may either create, read, update or delete an object.

The Object-Oriented Structured Systems Analysis and Design Methodology (OOSSADM) builds on the Jackson Systems Development approach (Jackson 1983). In this approach, entities impose sequence constraints on business events by means of a sequence diagram. As events can appear in the sequence diagrams of multiple entities, Robinson et al. (1994) introduced the notion of Entity-Event matrix to capture the entities that are affected by each business event.
Model driven, Existence dependency Relation, Object oriented DEvelopment (MERODE) is an object-oriented analysis and design methodology (Snoeck et al. 1999, Snoek et al. 1998) and is complementary to UML (OMG 2008), in that it offers a precise and computationally complete methodology. MERODE represents an information system through the definition of business events, their effect on enterprise objects and the related business rules.

Similarly as in OOSSADM, business events are identified as independent concepts, with an object-event table defining which types of objects are affected by which types of events. Each object type has a method for each event type in which it may participate. Such method implements the object’s creation, its state changes (i.e. changes to attribute values) or its deletion as the consequence of an event of the corresponding type.

The events in OOSSADM and in MERODE can be considered as elementary processes that have an effect (create, modify or update) on at least one, but possibly more object types. Hence, in its most simple form, these three tables indicate which objects participate in which elementary processes\(^4\). A convenient way for representing such table is by a cross table, which is a rectangular table of which the rows are labeled by the objects, the columns labeled by events and a cross in an entry indicates that the corresponding object participates in the corresponding event.

One major difference between the three approaches is the way a table is filled. In all three approaches, the table is initially filled on the basis of classical analysis: interviews with key users and logical reasoning. Subsequently, the table should be verified against some quality criteria. Coverage analysis will typically result in criteria that will attempt to identify missing rows, columns or crosses by means of general and intuitive “rules of thumb” such as: for each object type, there should be at least one process that creates objects of that type; each process should at least read some data, etc. Clustering analysis will yield rules of thumb for grouping entities to subdomains and processes to subsystems. Of the three modeling approaches, MERODE is the only approach that defines formal criteria for the completeness and well-formedness of the table.

The CRUD-matrix, Entity-Event and Object-Event table have a clear relationship with a single-valued formal context from FCA (see section 4.1.3); object or entity types in a conceptual model can be mapped to “objects” in FCA and events can be mapped to “attributes” in FCA. Formal Concept Analysis is a recent mathematical technique that can be used as an unsupervised clustering technique. The starting point of the analysis is a table consisting of rows (i.e. objects), columns \(F\) (i.e. attributes) and crosses (i.e. relationships between objects and attributes).

---

\(^4\) In the remainder of this section, we will call these elementary processes "events". It should however be noted that these "events" do not only symbolise the initiating trigger, but also the processing that is activated as response to the event.
The goal of the research is to investigate the possibilities of using FCA as grounding theory assisting in the development of conceptual domain models. The core contributions of this section are as follows. First, we show that conceptual modeling can be considered as an application of FCA. FCA provides a sound mathematical foundation for assisting modelers in the elaboration of conceptual domain models. Moreover, it provides a formal underpinning for the central notion of concept. Second, we show that the well-formedness rules from MERODE - obtained by reasoning on the semantics of existence dependency, on common sense reasoning and on process algebra considerations – are isomorphic to the FCA lattice construction algorithm. Whereas in MERODE, the consistency requirements were statically modeled as meta frame for conceptual models, FCA provides an algorithm that automatically verifies whether the association matrix and existence dependency graph are correct and consistent. Starting from an Object-Event table that is well-formed according to the MERODE-rules, we apply the clustering principles of FCA and demonstrate that FCA comes up with a ordering of concepts that is isomorphic to the ordering imposed by the existence dependency relationship used in MERODE. Hence, we can postulate that FCA offers a theoretical foundation for the consistency rules of MERODE. Reversely, a principal result is that it is possible in conceptual object-oriented analysis to obtain a concept lattice by using the MERODE rules. In this way we demonstrate that FCA is a valid instrument for the formal underpinning of matrix verification techniques (Snoeck et al. 1998).

4.1.3. FCA essentials

Formal Concept Analysis is a recent mathematical technique that can be used as an unsupervised clustering technique. Objects participating in the same set of events are grouped in concepts. The starting point of the analysis is a table consisting of rows \( M \) (i.e. objects), columns \( F \) (i.e. attributes) and crosses \( T \subseteq M \times F \) (i.e. relationships between objects and attributes). The mathematical structure used to reference such a cross table is called a formal context.

Table 4.1. Example of a formal context

<table>
<thead>
<tr>
<th></th>
<th>ENTER</th>
<th>LEAVE</th>
<th>ACQUIRE</th>
<th>CLASSIFY</th>
<th>BORROW</th>
<th>RENEW</th>
<th>RETURN</th>
<th>TITLE</th>
<th>LOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Book</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Loan</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
An example of a cross table is displayed in Table 4.1. In the latter, objects are related (i.e. the crosses) to a number of events (i.e. the attributes); here an object is related to an event if the object participates in the event. Given a formal context, FCA then derives all concepts from this context and orders them according to a subconcept-superconcept relation. This results in a line diagram (a.k.a. lattice).

The notion of concept is central to FCA. The way FCA looks at concepts is in line with the international standard ISO 704, that formulates the following definition: A concept is considered to be a unit of thought constituted of two parts: its extension and its intension (Ganter 1999, Wille 1982). The extension consists of all objects belonging to the concept, while the intension comprises all attributes shared by those objects. Typically, one would think here about informational attributes but-in line with an object oriented approach- one can just as well consider behavioral attributes such as reaction to events or participation in processes. So let us illustrate the notion of concept of a formal context using the data in Table 4.1. For a set of objects \( O \subseteq M \), the events that are common to all objects \( o \) in the set \( O \) can be identified, written \( \sigma(O) \), via:

\[
A = \sigma(O) = \{ f \in F \mid \forall o \in O : (o, f) \in T \}
\]

Take for example the set \( O \subseteq M \) consisting of objects Member, Book and Loan. This set \( O \) of objects is closely connected to a set \( A \) consisting of the attributes “borrow”, “renew”, “return” and “lose”, being the events shared by the objects in \( O \). That is:

\[
\sigma(\{\text{Member, Book, Loan}\}) = \{\text{borrow, renew, return, lose}\}
\]

Reversely, for a set of attributes \( A \), we can define the set of all objects that share all attributes in \( A \):

\[
O = \tau(A) = \{ i \in M \mid \forall f \in A : (i, f) \in T \}
\]

If we take as example the set of events of Loan, namely \{borrow, renew, return, lose\}, we get to the set \( O \subseteq M \) consisting of the objects Member, Book and Loan. That is to say:

\[
\tau(\{\text{borrow, renew, return, lose}\}) = \{\text{Member, Book, Loan}\}
\]

As one can see, there is a natural relationship between \( O \) as the set of all objects sharing all attributes of \( A \), and \( A \) as the set of all attributes that are valid descriptions for all the objects contained in \( O \). Each such pair \((O, A)\) is called a formal concept (or concept) of the given context. The set \( A = \sigma(O) \) is called the intent, while \( O = \tau(A) \) is called the extent of the concept \((O, A)\).
Notice that concepts are always maximal in the sense that the set $O$ contains all objects that share the attributes of $A$ and that $A$ contains all shared attributes of the objects in $O$.

Moreover, there is a natural hierarchical ordering relation between the concepts of a given context that is called the subconcept-superconcept relation:

$$\langle O_i, A_i \rangle \subseteq \langle O_j, A_j \rangle \iff (O_i \subseteq O_j \land A_i \subseteq A_j)$$

A concept $d = \langle O_i, A_i \rangle$ is called a subconcept of a concept $e = \langle O_j, A_j \rangle$ (or equivalently, $e$ is called a superconcept of a concept $d$) if the extent of $d$ is a subset of the extent of $e$ (or equivalently, if the intent of $d$ is a superset of the intent of $e$). For example, the concept with intent “enter,” “leave,” “lose,” “return,” “renew,” and “borrow” is a subconcept of the concept with intent “lose,” “return,” “renew,” and “borrow.” With reference to Table 4.1, the extent of the latter is composed of object types Loan, Member and Book, while the extent of the former is composed of object type Member.

The set of all concepts of a formal context combined with the subconcept-superconcept relation defined for these concepts gives rise to the mathematical structure of a complete lattice, called the concept lattice $β(M,F,T)$ of the context. The latter is made accessible to human reasoning by using the representation of a (labeled) line diagram. The line diagram in Figure 4.1, for example, is a compact representation of the concept lattice of the formal context abstracted from Table 4.1. The circles or nodes in this line diagram represent the formal concepts. It displays only concepts that describe objects and is therefore a subpart of the concept lattice. The shaded boxes (upward) linked to a node represent the attributes used to name the concept. The non-shaded boxes (downward) linked to the node represent the objects used to name the concept. The information contained in the formal context of Table 4.1 can be distilled from the line diagram in Figure 4.1 by applying the following reading rule: An object $g$ is described by an attribute $m$ if and only if there is an ascending path from the node named by $g$ to the node named by $m$. For example, Member is described by the attributes “enter,” “leave,” “lose,” “return,” “renew,” and “borrow.”

Retrieving the extension of a formal concept from a line diagram such as the one in Figure 4.1 implies collecting all objects on all paths leading down from the corresponding node. In this example, the extension associated with the upper node is {Loan, Book, Member}. To retrieve the intension of a formal concept one traces all paths leading up from the corresponding node in order to collect all attributes. In this example, the second concept in row two is defined by the attributes “sell,” “classify,” “acquire,” “lose,” “renew,” “return,” and “borrow.” The top and bottom concepts in the lattice are

---

5 The terms subconcept and superconcept are used here in an FCA-context and should not be confused with the notions of subclass and superclass as used in the OO-paradigm.
4.1. Using FCA for verification of process-data matrices

special. The top concept contains all objects in its extension. The bottom concept contains all attributes in its intension. A concept is a subconcept of all concepts that can be reached by travelling upward. This concept will inherit all attributes associated with these superconcepts. In our example, the first node on the second row with extension \{Member\} is a subconcept of the top node with extension \{Loan, Member, Book\}.

![Fig. 4.1. Line diagram corresponding to the context from Table 4.1](image)

In FCA, the concept generated by an object type $P$ is defined as $\gamma(P) = (\tau(\sigma(P)), \sigma(P))$ and the concept generated by an event type $b$ as $\lambda(b) = (\tau(b), \sigma(\tau(b)))$. In the line diagram, the nodes are labelled by the object types which generate the corresponding concept $C$. These are called the own object types of the concept $C$. An object class $A$ is called an owner class for a use case if this object class $A$ is involved in this use case and there is no class which is existence dependent of $A$ and which is also involved in this use case.

4.1.4. MERODE essentials

The MERODE methodology entails the notion of existence dependency, which superimposes a lattice structure (not to be confused with inheritance hierarchies) on objects. The concept of existence dependency (ED) is based on the notion of the “life” of an object. The life of an object is the span between the point in time of its creation and the point in time of its end. Existence dependency is defined at two levels: at the level of object types or classes and at the level of object occurrences. The existence dependency relation is a partial ordering on objects and object types which is defined as follows.
Definition 1 (Existence Dependency):

Let \( P \) and \( Q \) be object types. \( P \) is existence dependent on \( Q \) (notation: \( P \leftarrow Q \)) if and only if the life of each occurrence \( p \) of type \( P \) is embedded in the life of one single and always the same occurrence \( q \) of type \( Q \). \( p \) is called the dependent object, \( Q \) the master object type. The result is that the life of the existence dependent object cannot start before the life of its master. Similarly, the life of an existence dependent object ends at the latest at the same time that the life of its master ends.

The notion of existence dependency is similar to the notion of weak entity as introduced by Chen and the notion of master entity from OOOSSADM. In the ER-notation (Chen 1977) we can use the notion of a weak entity to denote an existence dependent object type since the existence of a weak entity depends on the existence of the other entities it is related to by means of a weak relationship (Chen 1977). Existence dependency is equivalent to the notion of a weak relationship that is in addition mandatory for the weak entity type.

MERODE requires all objects in the conceptual model to be related through existence dependency relationships only. The class diagram can therefore be represented as an existence dependency graph.

Definition 2 (Existence Dependency Graph):

Let \( M \) be the set of object types in the conceptual schema. The existence dependency graph (EDG) is a relation \( \leftarrow \) which is a bag over \( M \times M \) such that \( \leftarrow \) satisfies the following restrictions:

1) An object type is never existence dependent on itself:
   \[ \forall P \in M: (P,P) \notin \leftarrow \]

2) Existence dependency is acyclic. This means that:
   \[ \forall n \in \mathbb{N}, n \geq 2, \forall P_1, P_2, \ldots, P_n \in M: (P_1,P_2), (P_2,P_3), \ldots, (P_{n-1},P_n) \in \leftarrow \implies (P_n,P_1) \notin \leftarrow \]

\( \leftarrow \) is the non-reflexive transitive closure of \( \leftarrow \):

\( \leftarrow \subseteq M \times M \) such that

1) \[ \forall P, Q \in M: (P,Q) \in \leftarrow \implies (P,Q) \in \leftarrow \]
2) \[ \forall P, Q, R \in M: (P,Q) \in \leftarrow \text{ and } (Q,R) \in \leftarrow \]

---

\(^6\) Bags can contain the same element more than once (as opposed to sets).
In practice, MERODE also demands that the EDG is fully connected.

**Definition 3 (Object Event Table):**

The object-event table is a table with one row for each object type and one column for each event type. Each cell contains either a blank or a 'X', which stands for "participates in event". Let $A$ be the universe of relevant event types. Then $T \subseteq M \times A \times \{', 'X'\}$ such that $\forall P \in M, \forall a \in A :$

$\forall P \in M : x(P) = \{ a \in A | (P, a, 'X') \in T \}$

$A = \cup \{ x(P) | P \in M \}$

The OET is drawn as a matrix containing one row for each object type and one column for each event type. An 'X' on a row-column point of intersection indicates that this particular event type is an element of $x(P)$ (the alphabet of $P$), where $P$ is the object type corresponding to the row.

In MERODE, a multiple of well-formedness rules for object-oriented conceptual models are defined. These rules were elaborated based on reasoning on model quality (completeness and consistency), on object life cycles and the formalization by means of process algebra (Snoeck et al. 1998, Dedene et al. 1995). We now discuss four of these rules that are relevant for this section. It should be noted that we take the MERODE-rules as such and do not aim at motivating these rules in this section. For a motivation, the interested reader is referred to (Snoeck et al. 1999, Snoeck et al. 1998).

**Rule 1:**

The relevant life of a domain object type has a certain duration that can be delimited by two events: one event when the object enters the domain of interest and one event when the object leaves the domain of interest. In other words, each object type should participate in at least two event types: one for its creation and one for its ending. For the object-event table, this means that each row should contain at least two crosses.

**Rule 2:**

Each identified event type must be relevant for at least one object type. For the object-event table, this means that on each column there is at least one row with a cross.
Rule 3 (propagation rule):

A master object type is always involved in all event types in which one of its
dependent object types participates. For example, a state change of a loan,
e.g. because of the return of the book, automatically implies a state change of
the related book and member: the book is back on shelf and the member has
one copy less in loan. Therefore, if P is existence dependent of Q, the
alphabet of P must be a subset of the alphabet of Q. This is called the
propagation rule: \( P \leftrightharpoons Q \Rightarrow x(P) \subseteq x(Q) \).

Rule 4 (contract rule):

The contract rule says that when two object types share two or more event
types, the common event types must be in the alphabet of one or more
common existence dependent object types:

\[
\forall P, Q \in M : \#(x(P) \cap x(Q)) \geq 2 \text{ and } \neg(x(P) \subseteq x(Q) \text{ or } x(Q) \subseteq x(P)) \Rightarrow \exists R_1, \ldots, R_n \in M : \forall i \in \{1, \ldots, n\} : R_i \leftrightharpoons P, Q \text{ and } x(R_1) \cup \ldots \cup x(R_n) = x(P) \cap x(Q)
\]

Consequence: \( x(P) \subseteq x(Q) \Rightarrow P \leftrightharpoons Q \)

Notice that in MERODE the contract rule is only applicable in case of two or
more common event types and that at least one of these must create and
another one must end the existence dependent object types. If there is only
one common event type MERODE does not require the definition of an extra
object type, because an object type requires at least two event types. The
argument for two events is only based on the fact that a life cycle requires a
start and an end.

4.1.5. FCA and the object-event matrices

As explained before, the well-formedness rules from MERODE were
obtained by reasoning on the semantics of existence dependency, on common
sense reasoning and on process algebra considerations. We reformulate each
of the MERODE-rules in FCA terms. The existence of rules 1 and 2 can
easily be motivated in FCA as well: objects without attributes and attributes
(events) that belong to no objects can be considered as incompletenesses in a
conceptual model. For rule 3 it appears that the principle of propagation of
events along existence dependency paths yields the result that the FCA and
EDG lattice are to a large extent isomorphic. In other words, by imposing the
well-formedness rules from MERODE, we obtain a concept lattice in
conceptual object-oriented analysis. Reversely, the subconcept-superconcept relationship of FCA turns out to be isomorphic to the existence dependency relationship. Finally, although rule 4 (contract rule) can be formulated in FCA terms, FCA offers no substantiation for the existence of this rule. This could be a reason to revise that rule in MERODE.

4.1.5.1. Object-event table with empty row column

Consider the object-event table displayed in Table 4.2. The lattice corresponding to Table 4.2 is displayed in Figure 4.2. This table has an empty row, which means there is an object type \( R \) that is involved in no event type at all. In FCA, this implies that the object type \( R \) is part of the extent of the top concept of the corresponding lattice. This top concept has an empty intent. In other words, there is an object type that has no attributes: it participates in no events.

This table also has an empty column, which means there is an event type \( e \) that involves no object type at all. In FCA, this implies that the event type \( e \) is part of the intent of the bottom concept of the lattice. This bottom concept has an empty extent; it is an attribute that belongs to no object. Stated differently: there is an event in which no object participates.

Table 4.2. Object-event table with empty row and column

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P )</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( Q )</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( R )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4.2. Lattice corresponding to Table 2
MERODE considers these two cases as modeling anomalies. Rule 1 demands that each object type participates in at least two event types, one for the creation of objects and one for the deletion of objects. So, empty rows are not allowed. This MERODE rule can be reformulated in FCA terms as follows.

**Rule 1 in FCA terms:**

Let $P$ be an object type, $L$ a concept and $m, n$ events:

\[
\text{Given: } (M, F, T) \\
\forall P \in M : \\
\exists L \in \beta(M, F, T) : P \in \text{ext}(L) \land \exists m, n : m, n \in \text{int}(L) \land m \neq n
\]

MERODE also demands that each event type is relevant for at least one object type, so empty columns are not allowed. This MERODE rule can be reformulated in FCA terms as follows.

**Rule 2 in FCA terms:**

\[
\text{Given: } (M, F, T) \\
\forall a \in F : \exists L \in \beta(M, F, T) : \text{ext}(L) \neq \emptyset \land a \in \text{int}(L)
\]

### 4.1.5.2. Propagation rule and subconcept-superconcept relation

On an implementation level each participation by an object type in a use case represents a method in this class for this use case. In each use case in which the owner class participates, also the classes on which the owner class is existence dependent participate. This is called propagation of use case participations. The motivation for this propagation is that nothing can happen to a dependent object without it happening at the same time to the master object. For example, if something happens to a loan, e.g., it is renewed, automatically the copy and the member are also involved in this renewal.

Object Constraint Language (OCL) is a formal language used to describe expressions in UML models. These expressions typically specify invariant conditions that must hold for the system being modeled or queries over objects described in a model. Starting from a specific object, we can navigate by an association on the class diagram to refer to other objects and properties. The value of this expression is the set of objects on the other side of the association. If the multiplicity of the association-end has a maximum of one ("0..1" or "1") then the value of this expression is an object. To specify navigation to association classes, OCL uses a dot and the name of the association class.
Navigation from a master to a dependent object type may result in zero, one or multiple objects (arrow notation). For implementation level analysis the FCA lattice gives insight into the accessibility of operations. An object can always uniquely navigate to the objects below in the lattice and access their operations. Loan can always access the operations of member, but not vice versa. This can be explained as follows. When an existence dependent object is involved in a use case, its master objects are automatically involved in this use case as well. For example, a state change of a loan, e.g., because of the return of the copy, automatically implies a state change of the related copy and member; the copy is back on shelf and the member has one copy less in loan. By including the alphabet of the dependent object type in the alphabet of its master, all possible places for information gathering and constraint definition are identified.

In this section we demonstrate that because of the propagation rule, every object in the EDG generates a tuple \( \epsilon(P) = (P^*, x(P)) \), where \( P^* \) is the set of all masters of \( P \) and \( x(P) \) is the alphabet of \( P \). Then \( \epsilon(P) \) matches with the concept in the FCA line diagram with label \( P \) and moreover, if \( P \) is existence dependent on \( Q \), then there is an upward path in the FCA lattice from the node with label \( Q \) to the node with label \( P \).

**Definition 3:**

Let \( \epsilon(P) \) be defined as follows:

\[
\epsilon(P) = (P^*, x(P)), \text{ with } P^* = \{ X \mid (P,X) \in \leftarrow \} \cup \{ P \}
\]

**Theorem 3:**

\( \epsilon(P) \) is a concept in FCA

Proof:

\( \epsilon(P) = (P^*, x(P)) \)

To be proven:

\( P^* = \pi(x(P)) = \{ Q \in M \mid \forall e \in x(P) \mid e \in x(Q) \} \)

\( Q \in P^* \)

\( \iff (P,Q) \in \leftarrow \)

\( \iff x(P) \subseteq x(Q) \) (because of propagation rule)

\( \iff \forall e \in x(P) \mid e \in x(Q) \)

\( \iff Q \in \pi(x(P)) \)

QED

**Theorem 4:**
if $P \leftarrow Q$ then $\mathcal{E}(P)$ is a superconcept of $\mathcal{E}(Q)$

Proof

\[
\mathcal{E}(P) = (\{P\} \cup \{X \mid (P,X) \in \mathcal{E}\}, \mathcal{x}(P))
\]

\[
\mathcal{E}(Q) = (\{Q\} \cup \{Y \mid (Q,Y) \in \mathcal{E}\}, \mathcal{x}(Q))
\]

(1):

$P \leftarrow Q$

$\Rightarrow (Q \leftarrow Y \Rightarrow P \leftarrow Y)$

$\Rightarrow \{X \mid (P,X) \in \mathcal{E}\} \supseteq \{Y \mid (Q,Y) \in \mathcal{E}\}$

$\Rightarrow (\{P\} \cup \{X \mid (P,X) \in \mathcal{E}\}) \supseteq (\{Q\} \cup \{Y \mid (Q,Y) \in \mathcal{E}\})$

(2):

$\mathcal{x}(P) \subseteq \mathcal{x}(Q)$ (because of propagation rule)

(1) + (2) $\Rightarrow \mathcal{E}(P)$ is a superconcept of $\mathcal{E}(Q)$

QED.

Theorem 5:

if $\gamma(P)$ is a superconcept of $\gamma(Q)$ in FCA then $P \leftarrow Q$ in MERODE

Proof

$\gamma(P) = (\tau(\sigma(P)), \sigma(P))$

$\gamma(Q) = (\tau(\sigma(Q)), \sigma(Q))$

$\Rightarrow \text{int}(\gamma(P)) \subseteq \text{int}(\gamma(Q))$

$\Rightarrow \mathcal{x}(P) \subseteq \mathcal{x}(Q)$

$\Rightarrow P \leftarrow Q$

QED

Consider for example the object-event table in Table 4.1. The set of events in which Loan participates is a subset of the events in which Member participates. In MERODE, Loan is said to be existence dependent of Member. In FCA, the concept generated by Loan is a superconcept of the concept generated by Member. As a consequence, there is an upward leading path from the node with label Member to the node with label Loan in the FCA lattice (see Figure 4.1).

4.1.5.3. FCA and the contract rule

In this section we look at the FCA equivalent of the MERODE contract rule. If the object-event lattice contains a concept with two or more own event types in its intent and zero own object types in its extent, then we have a node in the line diagram with attribute labels, but without object type labels attached to the node. In MERODE, this indicates a situation where the object
types one level lower in the lattice share at least two events that do not appear in the alphabet of one or more common existence dependent object types. This situation is in contradiction with the contract rule (rule 4). In section 4.3 we will explain how the FCA-equivalent of the contract rule may help in identifying potentially missing object types. We will showcase the benefits of FCA on toy examples, however the full potential of FCA’s partial ordering and visualization of data concepts and its advantages over the OET will be extensively discussed with real life examples in section 4.3.

**Theorem 6:**
If a lattice $\beta(O, A, I)$ contains a concept $C$ such that $\text{own}(\text{ext}(C)) = \emptyset$ and $\text{own}(\text{int}(C)) \geq 2$, then an object type is missing in the object-event table according to the MERODE contract rule.

**Proof:**
1. $(C \in \beta(O, A, I)) \land (\text{ext}(C) = \emptyset) \land \text{int}(C) \geq 2$
2. $\Rightarrow \exists a \in A : \{P \mid Y \in \beta(O, A, I) \land P \in \text{ext}(Y) \land a \in \text{int}(Y)\} = \emptyset \Rightarrow$ violation of Rule 2

$$\Rightarrow \exists P, Q \in O : \exists L, R \in \beta(O, A, I) \land L \neq R \land L \subseteq C \land R \subseteq C \land P \in \text{ext}(R) \land Q \in \text{ext}(L)$$

$$\Rightarrow \text{int}(C) \subseteq x(P) \land \text{int}(C) \subseteq x(Q) \land \text{ext}(C) = \emptyset$$

$$\Rightarrow x(P) \cap x(Q) > 1 \land \forall R \in O : \forall i \in \{1, ..., n\} : R_i \not< P, Q$$

$$\Rightarrow \text{violation of contract rule}$$

QED

Table 4.3. Expanded formal context

<table>
<thead>
<tr>
<th></th>
<th>ENTER</th>
<th>LEAVE</th>
<th>ACQUIRE</th>
<th>CLASSIFY</th>
<th>BORROW</th>
<th>RENEW</th>
<th>RETURN</th>
<th>SEL</th>
<th>LOSE</th>
<th>RESERVE</th>
<th>CANCEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Book</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loan</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
We now illustrate Theorem 6. Suppose that besides borrowing books, it is also possible to reserve books that are not on shelf. If a member changes her mind and decides not to fetch the copy, she can cancel the reservation. The events "reserve" and "cancel" are added to the object-event table from Table 4.1. The resulting object-event table is displayed in Table 4.3.

The shaded area of Table 4.3 shows the common event types of Book and Member. Some of the events are also in the alphabet of the dependent object type Loan but "reserve" and "cancel" do not appear in the alphabet of a common existence dependent object type. Figure 4.3 displays the corresponding lattice when the object type Reservation is missing.

![Fig. 4.3. Potential missing object type](image-url)

The concept with own attributes "reserve" and "cancel" does not have any own object types in its extent. To fulfill the requirements imposed by the MERODE contract rule, an object type that participates in the own event types contained in the intent of the concept should be added to the business
domain model. This object type must be existence dependent of the object
types contained in the extent of the concepts one level lower in the lattice.

In this case, according to the MERODE contract rule, the two event types
should either be included in the alphabet of Loan or they should be included
in the alphabet of a new object type Reservation, dependent of both Member
and Copy. According to MERODE-practices, the latter solution is to be
preferred, because a loan can occur without a reservation and a reservation
can occur without being followed by a loan. Figure 4.4 displays the correct
object-event lattice and Table 4.4 displays the correct formal context.

However, FCA does not provide any formal grounding for the contract
rule and in particular for the requirement that the contract rule should only be
applicable in case of two or more common event types. One could for
example already consider to create an additional object type even if there is a
node with only one event type and no own object type. Also, a node with a
potential missing object type as in Figure 4.3, has already a non-empty
extension. So, there is no immediate reason to add an object in that point of
the lattice. As a result, FCA does not offer a formal foundation for this rule in
MERODE. In fact, in MERODE, this rule was created for deadlock
verification purposes (Dedene et al. 1995), rather than to actually identify
missing object types. The fact that FCA does not immediately support this
rule, could motivate a revision of this rule.

4.1.5.4. Verification process
The results of this section can be used for formal verification purposes as part of a larger software engineering process. The verification process using FCA can be seen as an iterative learning loop. In each iteration, an existing process-data matrix is used to automatically derive an FCA lattice. As we have shown in the previous sections, this lattice can be considered as a conceptual domain model that obeys the well-formedness rules as imposed by MERODE. The lattice structure can then be used for formal verification purposes, i.e. to detect anomalies, missing concepts, missing object types, etc. These changes can be implemented in the existing process-data matrix and a new iteration through the learning loop is started until a correct model is obtained. This iterative requirements engineering process will studied in detail in section 4.3.

4.1.6. Conclusions

In this section, we proposed a novel application of FCA, namely as a formal foundation for the verification of matrices used in conceptual domain modeling. The well-formedness rules for the object-event table in MERODE were developed by reasoning on the semantics of existence dependency, on common sense reasoning and on process algebra considerations. We showed that by imposing the well-formedness rules from MERODE, we obtain a domain model that has all the properties of a concept lattice. This substantiates the well-foundedness of these rules. Reversely, if FCA is used to cluster an object-event matrix and the matrix has been filled in accordance to the rules of MERODE, the concepts identified by FCA can easily be mapped to object types in an enterprise model. In addition, the subconcept-superconcept relationship between FCA-concepts can be mapped to the existence dependency relationship in the enterprise model. In this way, we demonstrated the applicability of FCA as a theory to aid in the development of sound conceptual modeling methods.

The theory of FCA can also be applied to the is-a relationship between concepts. In section 4.2 we will investigate the application of FCA to generalisationspecialisation lattices and how this can be combined with a CRUD-like matrix. Yet we already dare to postulate as a general conclusion that FCA is a valid instrument for formalizing the construction of a conceptual domain model. In section 4.3, the work presented in this section will be empirically validated by applying it to real life case studies.

4.2. Extending Formal concept Analysis with a second ordering relationship on concepts: a software engineering case study

Formal Concept Analysis (FCA) has been used extensively in software engineering research. Traditional FCA theory has however a significant shortcoming when used for underpinning the construction of a conceptual domain model. Conceptual models typically contain two types of relations
between classes, i.e. associations between classes and inheritance relationships, whereas FCA only imposes one ordering on concepts, namely the subconcept-superconcept relation. We eliminate this lack of expressive power of traditional FCA by introducing an extra dimension in FCA. Whereas triadic FCA imposes conditions on which objects participate in which attributes, we introduce a second ordering relation and we show how this expanded version of FCA can be used to model associations between classes and inheritance relationships in one mathematically sound model.

4.2.1. Introduction

Traditional FCA theory has an important shortcoming when we want to use it to formally underpin the construction of conceptual models. These models typically contain two types of relationships between classes, i.e. associations between classes and the inheritance relationships between them. Traditional FCA only has one type of ordering on concepts, namely the subconcept-superconcept relation. In this section, we extend traditional FCA theory with an extra dimension. The aim is to allow the concepts to be ordered, fulfilling the requirements of two types of orderings. We also give some examples and applications in software requirements engineering and modeling. In particular, we show how the traditional subconcept-superconcept relationship can model Existence Dependency relationships (Poelmans et al. 2010a) and how this new dimension we introduced, allows for the integration of inheritance hierarchies.

Triadic FCA (Lehmann et al. 1995), we found to be not useful since this extended theory does not work with two ordering relationships, but rather imposes conditions whether or not a certain object has an attribute.

4.2.2. Expressing UML associations with an FCA lattice

In (Poelmans et al. 2010a) we demonstrated that starting from an entity-use case interaction matrix which indicates the use cases in which the entities participate, an FCA lattice can be automatically derived. This FCA lattice is not necessarily a correct model of the system but may indicate some concerns in the present design. The table corresponding to the extended library example of section 4.1 is displayed in Table 4.5.

---

7 UML identifies a number of supplementary constructs, such as aggregation and association as a class. These constructs can however be considered as special cases of the basic notion of association. They are (as opposed to inheritance) not fundamentally different from the concept of association.
By letting the specialization object types (inheritance) out of the lattice construction, we obtain an FCA lattice containing the associations that should be in the class model according to the matrix. In contrast to UML, FCA imposes a partial order relation on all objects. In section 4.1 we showed that the set of lines which interconnect the FCA concepts form a lattice that is isomorphic to the MERODE EDG. Fig. 4.5 contains a lattice for the library example derived from the entity-use case interaction matrix.

![Existence dependency lattice](image)

**Fig. 4.5. Existence dependency lattice**

In the next section we will add inheritance relationships to these class diagrams.
4.2.3. Expressing inheritance relationships with an FCA lattice

Generalization / specialisation is an abstraction principle that allows to define classes as a refinement of other classes. The more general class is also called supertype or parent class and the refined class is then called a subtype or child class. In this section we will only consider single inheritance. The principle of inheritance says that each object type inherits the operations of its eventual parent object type. Such a rule can be naturally expressed using FCA. In our case, the FCA lattice is build based on object-use case participations. Using the principle of substitutability of Barbara Liskov, we can reformulate the rules of inheritance for Use Case participation. The principle of substitutability states that whenever a generalization object occurs it can be replaced by a specialized object, because subtypes are by definition of inheritance type-conformant to their supertype(s). Suppose for example that a Use Case “register Item” has been defined, and that Item has the subtypes CD-rom and Book, then we assume that “register Item” also applies to CD-roms and books. If however a borrow Use Case has been defined for books, we cannot assume that it also applies to items in general. So, entity-use case interactions of the generalized entity are inherited by the specialized entity, but not reversely. Table 4.6 shows a formal context containing inheritance relationships. If the inherited use case is applicable to the specialized object type (possibly after overriding the corresponding method), the I/ must be followed by C, M or E to indicate the nature of the involvement. If it is not, the use case is specialized for each subtype, which is indicated by S/ followed by C, M or E.

Table 4.6. Formal context containing inheritance relationships

<table>
<thead>
<tr>
<th></th>
<th>CR_ITEM</th>
<th>CR_CDROM</th>
<th>CR_ISSUE</th>
<th>CR_BORROWABLE</th>
<th>E_ITEM</th>
<th>E_BORROWABLE</th>
<th>E_CDROM</th>
<th>E_ISSUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>C</td>
<td>M</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Borrowable Item</td>
<td>S/C</td>
<td>C</td>
<td>I/M</td>
<td>S/E</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDRom</td>
<td>S/C</td>
<td>C</td>
<td>I/M</td>
<td>S/E</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>S/C</td>
<td>C</td>
<td>I/M</td>
<td>S/E</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The FCA lattice obtained from the subpart of the entity-use case interaction matrix, denoting the inheritance relationship is displayed in Fig. 4.6.
Object types lower in the lattice are specializations of object types higher in the lattice. The inheritance relationships between object and event types can be expressed using an FCA lattice. In (Snoeck et al. 1996), the generalization-specialization hierarchy is defined as a tree (no multiple inheritance) and specialization types inherit the alphabet of their subtype. This single inheritance hierarchy can be defined as a special type of lattice $K(O, A, I)$ with the extra constraint that each concept $S$ has at most one superconcept $G$ (with the exception of the infimum of the lattice, which must have an empty extent):

$$\forall G, G', S \in K(O, A, I) : G <_i S \text{ and } G' <_i S \Rightarrow G = G'$$

$G$, $G'$ and $S$ are concepts. $G$ is called a super concept of $S$ provided $\text{ext}(G) \subseteq \text{ext}(S)$ and $\text{int}(S) \subseteq \text{int}(G)$. This is denoted as $G <_i S$. Then, $(K(O, A, I), <_i)$ is a partially ordered set that is denoted by $K(O, A, I)$ and is called the concept lattice of the context.

To obtain the set of parent types of an object type $P$, we take the extent of the concept that owns $P$ (which is called the concept $L$):

$$\gamma'_*(P) = \text{ext}(L)$$

To obtain the set of descendant types of a type $P$, we take the extent of all subconcepts of the concept that owns $P$ (namely $L$):

$$\gamma_*(P) = \{\text{ext}(C \in K(O, A, I)) \mid C <_i L\}$$

To obtain the topmost parent type of a type $P$, we take the extent of the supremum concept of the lattice:

$$\gamma_{\text{max}}(P) = \text{ext}(\text{sup}(K(O, A, I)))$$

---

**Fig. 4.6. Inheritance lattice**

---

[Diagram of an inheritance lattice with nodes for CD ROM, Borrowable Item, Issue, Item, and the relationships of Generalization and Specialization.]
4.2.4. Merging the existence dependency and inheritance lattices

An extended lattice is defined as a combination of a main lattice, inheritance lattices, and sublattices. In this extended lattice, the concepts are ordered according to two ordering relations. The vertical dimension is used for the existence dependency relationships between the concepts. The existence dependence object type will always be above its master in the lattice. The horizontal dimension is used to model the inheritance relationships between the concepts. The specialized object will always be to the right of its generalization in the merged lattice. Both orderings are based on the subconcept-superconcept relationship from traditional FCA but have a different semantic meaning.

![Diagram showing the merging of existence dependency lattice and inheritance lattice to form an extended lattice.]

Fig. 4.7. How the existing dependency lattice and the inheritance lattice are combined to form one extended lattice.

There will always be at least one object type owned by a concept of an inheritance lattice always belongs to the extent of a concept from the main lattice. When merging the existence dependency and inheritance lattices, we need to distinguish between the concepts from the ED and inheritance
lattices. We make use of the notion of specialized concept. Note that we do not show the infimum of the inheritance lattices in the extended lattice. This concept is not useful in this section because we only consider single inheritance, and can thus be omitted from the visualization. Figure 4.7 visually represents the process of merging existence dependency and inheritance lattices. Figure 4.8 explains the structural properties of the result of this process.

Fig. 4.8. Properties of the extended lattice

The relationships amongst the concepts of these lattices after propagation of use case participations through the dashed arrows are as follows:
1. int (B) ⊆ int (C) and ext (C) ⊆ ext (B)
2. int (C) ⊆ int (D) and ext (D) ⊆ ext (C)
3. int (B) ⊆ int (D) and ext (D) ⊆ ext (B)
4. int (A) ⊆ int (B) and ext (B) ⊆ ext (A)
5. int (A) ⊆ int (C) and ext (C) ⊆ ext (A)
6. int (A) ⊆ int (D) and ext (D) ⊆ ext (A)

The propagation of use case participations from A to B and B to D is required because of ED, as explained in section 4.1. The propagation of A to D is the consequence of the transitive ED from A to D via B. This explains properties 3, 4 and 6. The propagation from B to C is required because of inheritance as explained in section 4.2.3. This explains property 1.
4.1. USING FCA FOR VERIFICATION OF PROCESS-DATA MATRICES

The combination of the existence dependency between A and B and the inheritance relationship between B and C, allows to naturally infer a relationship between A and C: C inherits from B the fact that it has a relationship with C. Moreover as A's attributes are a subset of B's attributes, the attributes of A are a subset of those of C, thereby implying a similar superset-subset relation relationship as between B and C. This explains property 5.

Property 2 needs some more explanation. C inherits from B the fact that it is ED from D. However, D has more attributes than B, so the additional attributes of D are not naturally inherited by C. In the other direction, we know that \( \text{int}(B) \subseteq \text{int}(C) \), however, C has more attributes than B. We will now discuss why we postulate that \( \text{int}(C) \subseteq \text{int}(D) \). Although the liskov principle of substitution states that in a relationship between D and B, a B-object can be replaced by a C-object, this can only be done insofar the C-object is capable of adhering to the same attributes as B. The extentsions of C are not visible, nor known by D. The reason why the non-trivial property 2 was introduced is to include the benefits of the accumulation rule as introduced by MERODE in the FCA software model (thoroughly discussed in section 5.1).

We can see that if this is done, concepts B and C have an identical relationship to the concepts A and D from the main lattice. We can say that from the point of view of A and D, the inheritance constellation consisting of concepts B and C is seen as one “expanded node”. This materialises the Liskov principle of substitutability: from the perspective of A and D, B can be substituted by C at any time. Propagation of use cases is performed through the dashed arrows. The propagation of use case participations from C to D is necessary to fulfill the requirements of the OO principle of substitutability. An object type acquires the use case participations of its existence dependent object types through propagation plus all the use case participations of the object types that are directly or indirectly a specialization of any of the existence dependent object types (discussed in more detail in section 5.1). On an implementation level these propagations can also be interpreted as follows. The dotted arrows indicate the possibility for unique navigation from one object type to another. This lattice structure is in accordance with the principle of substitutability from OO.

4.2.4.1. Properties of extended lattice

**Proposition 1.**

There is at least 1 concept in the inheritance lattice with the same extent and intent as a concept in the ED lattice. Each object type that is owned by concept in the inheritance lattice is also owned by the corresponding concept.
from the ED lattice. In the extended lattice, these 2 concepts are displayed as one concept. They form the hinges between the ED lattice and inheritance lattice:

Given: \((\beta(O, A, I), <)\) and \((K(X, Y, S), <_j)\)

Proposition (required):
\[\forall n \in \mathbb{N} : \exists L \in (\beta(O, A, I), <) : (\exists E \in (K(X, Y, S), <_j)) :\]
\[ext(L) = ext(E) \land int(L) = int(E)\]

In other words, a concept in the inheritance lattice is also a part of the ED lattice. This concept is the link between the two lattices.

**Proposition 2.**

The vertical ordering of concepts according to the ED subconcept / superconcept relation is not violated by introducing inheritance lattices that are attached to the main lattice. For each specialized concept in an inheritance lattice, the following relationships with the main ED are valid:

Given:

\((\beta(O, A, I), <), (K(X, Y, S), <_j)\) and concept \(L\) as defined in proposition 1

Proposition:

\[\forall T \in (\beta(O, A, I), <) :\]
- \((T > L) \iff (\text{int}(T) \subseteq \text{int}(L)) \land (\text{ext}(L) \subseteq \text{ext}(T))\)
  \[\land (\forall W <_j L : \text{int}(T) \subseteq \text{int}(W) \land \text{ext}(W) \subseteq \text{ext}(T))\]
- \((T < L) \iff (\text{int}(L) \subseteq \text{int}(T)) \land (\text{ext}(T) \subseteq \text{ext}(L))\)
  \[\land (\forall W <_j L : \text{int}(W) \subseteq \text{int}(T) \land \text{ext}(T) \subseteq \text{ext}(W))\]

The specialized concepts of a concept \(L\) of a main lattice have the same subconcept-superconcept relations with concepts from the main lattice as \(C\).

### 4.2.5. Applications

#### 4.2.5.1. Accumulation rule

The accumulation rule was introduced to reveal the implicit inheritance relationships in a model, hence allowing verifying the correctness of a software model consisting of 2 ordering relations on concepts. The rule
allows for the detection of inherited existence dependency relationships that are not explicitly modelled. This helps in identifying places in the model where such implicity links are erroneous and hence should be removed by removing existence dependency or inheritance relationships or by refactoring the model. The accumulation principle says that an object type not only acquires the use case participations of its existence dependent object types through propagation but also all the use case participations of the object types that are directly or indirectly a specialization of any of the existence dependent object types. It should be noted that this propagation of use case participations through the accumulation principle results in an implicit participation of the master in the use cases of the specialization of its dependent.

Intuitively, conceptual modelers tend to assume covariance, which allows to define more stringent preconditions for specializations than for generalized object types. Covariance is however almost nowhere supported for design and coding purposes. Contravariance is most used by software designers and states that the properties of a contract or association between the generalized object types cannot be made more stringent at the specialized object type level, only loosened. The specialized object types inherit the properties of the contracts of their supertype and the properties of the associations their supertypes participate in. In Fig. 4.10 the accumulation principle is used to detect the erroneous link between Borrowable Item and Loan, and in Fig. 4.11 between Product and Sale. When we observe the structure from Figure 4.9 in a conceptual domain model expressed as an FCA lattice, the model contains an anomaly.
The model contains two links from the dependent Object type $C$ to its master $D$. Events are propagated from $A$ to $B$ by two paths, namely 1) and 2), while one of them is superfluous. In case such an anomaly is observed, there are two options. Either one deletes the ED link between the generalization object types $A$ and $B$ or one deletes the ED link between the specialization types $C$ and $D$. Option 1 is preferred when there is an object type $E$ that is a specialization of $B$ and that should not inherit the (implicit) ED relationship with $C$ from $B$. This is illustrated using the following two real life examples (taken from (Snoeck 1999)).

Example 1:

Our first example comes from a library environment. In a library, different types of items are available: CD-roms, single issues of journals, bound volumes of journals, and copies of books. CD-roms and single issues of journals must not leave the library. The other types of items can be borrowed (and returned). Only loans of copies can be renewed. A first extended lattice for this example is given in Fig. 4.10. The object-event table is given in Table 4.7 and for readability purposes, we have placed the entities in the columns and the use cases in the rows.

Table 4.7. Formal context of library environment

<table>
<thead>
<tr>
<th></th>
<th>TITLE</th>
<th>ITEM</th>
<th>LOAN</th>
<th>MEMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CDrom</td>
<td>ISSUE</td>
<td>BORROWABLE ITEM</td>
<td></td>
</tr>
<tr>
<td>cr_title</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_title</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>classify</td>
<td>M</td>
<td>M</td>
<td>I/M</td>
<td>I/M</td>
</tr>
<tr>
<td>cr_item</td>
<td>M</td>
<td>C</td>
<td>S/C</td>
<td></td>
</tr>
<tr>
<td>cr_CDrom</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cr_issue</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cr_borrowable_item</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cr_volume</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cr_copy</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_item</td>
<td>M</td>
<td>E</td>
<td>S/E</td>
<td></td>
</tr>
<tr>
<td>e_CDrom</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_issue</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_borrowable_item</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_volume</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_copy</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>borrow</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cr_renewable_loan</td>
<td>M</td>
<td></td>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>

Note: The table is filled with symbols indicating the strength of the relationships, with 'M' for medium, 'S/C' for strong or critical, and 'S/E' for strong or essential.
Notice that in Fig. 4.10 because of the accumulation principle, the renew event was also propagated from Renewable loan to Borrowable item. If inheritance is applied again, Volume will now inherit the renew event from Borrowable item. But this is not what we specified: loans of volumes cannot be renewed. The error stems from the erroneous existence dependency relation between Borrowable item and Loan. A generalization/specialisation hierarchy must always be interpreted as “a generalisation object or a specialisation1 object or a specialisation2 object or...”. The erroneous existence dependency relation therefore means: “a loan and by inheritance a renewable loan are existence dependent on a volume or on a Copy.” But a renewable loan cannot be existence dependent on a Volume. The library schema must be corrected by removing the ED relationship between Borrowable item and Loan and by adding an entity being a specialisation of Loan named “not renewable loan” with an ED relationship to Volume.
Example 2:

In a pharmacy, some products can be sold freely, others only on doctor's order. This is represented in the extended lattice of Fig. 4.11.

![Fig. 4.11. Detecting anomalies using the extended lattice of a pharmacy system.](image)

The original model contained an existence dependency relation between product and sale since the general case is that a sale is always existence dependent on a product. But again, this dotted existence dependency relation must be interpreted as: "a sale, and by inheritance a free sale and a restricted sale are existence dependent on a product, which is a free product OR a restricted product." Since a restricted sale is existence dependent on a restricted product only (by the intermediary of a prescription), the existence dependency relation between product and sale is wrong and should be removed.

4.2.6. Conclusions

In this section we showed how traditional FCA theory can be extended with a second ordering relation on concepts. Two partial orderings, one in the horizontal direction and one in the vertical direction increase the expressiveness of the theory thereby making it usable in situations where it
was previously not. We particularly showcased the possibilities of the extended theory for software engineering activities. Both inheritance and the traditional ordering based on existence dependency can now be modeled and mathematically underpinned with FCA.

4.3. **An Iterative Requirements Engineering Framework based on Formal Concept Analysis and C-K theory.**

In this section, we propose a method for iterative requirements engineering using Formal Concept Analysis. The requirements engineering approach is grounded in the theoretical framework of C-K theory. An essential result of this approach is that we obtain normalized class models. Compared to traditional UML class models, these normalized models are free of ambiguities such as many-to-many, optional-to-optional or reflexive associations which cause amongst others problems at design time. FCA has the benefit of providing a partial ordering of the objects in the conceptual model based on the use cases in which they participate. The four operators of the C-K design square give a clear structure to the requirements engineering process: elaboration, verification, modification and validation. In each of these steps the FCA lattice visualization plays a pivotal role. We empirically show that the strategy works by applying it to a set of real-life case studies and a modeling experiment in which 20 students took part.

4.3.1. **Introduction**

During the conceptual modeling phase, user requirements are represented in a specification of what the system does as if there were a perfect implementation technology available (McMenamin et al. 1984). This is not a model of how an implementation works but of what an implementation must accomplish. Use cases and the conceptual domain model are the most important artifacts resulting from this phase. A use case is a system usage scenario involving one or more actors and the purpose of a use case specification is to describe the flow of events in detail, including how the use case starts, ends, modifies the system and interacts with actors. By analyzing the domain of interest, identifying and modeling relevant entities and relationships we obtain the conceptual domain model. In (Lindland et al. 1994) a framework is presented for evaluating the quality of conceptual models. A distinction is made between syntactic and semantic quality. Several methods have been introduced for detecting and handling syntactic problems such as inconsistencies. 75% of these techniques, such as model checkers, theorem provers, coherence checkers, etc. are formal (Lucas et al. 2009) and unfortunately not very popular in the industrial software development community (Beckert et al. 2006). This unpopularity is usually due to the fact that these approaches are difficult for modelers to use directly
and that the feedback they offer, is usually poor and difficult for non-experts to understand. Semantic validation of requirements and conceptual models is a social rather than a technical process, which is inherently subjective and difficult to formalize (Vliet et al. 2000). While some errors can be detected automatically, most errors can only be detected with the involvement of humans (Moody 1998) since a conceptual model can only be evaluated against people’s (tacit) needs, desires and expectations.

According to Wieringa et al. (2006) one of the problems is that we miss sound methodology that captures the essential elements of requirements engineering. The research on requirements engineering and conceptual modelling quality which has been done so far seems to have had little impact or practice (Moody 2005). Few of the proposals have been widely accepted in practice and many have never been applied outside a research environment. Several authors claimed that researchers need to address the issue of practitioner acceptance (Kaindl et al. 2002, Moody 2003). According to the literature on quality management, the most effective way to improve quality of products is to improve the process by which they are produced (Evans et al. 2004). So far, conceptual modeling quality research has focused almost exclusively on product quality: very few proposals even mention the issue of process quality (Maier et al. 1999, Maier et al. 2001). In requirements engineering, we can distinguish 4 broad categories of activities: elaboration of requirements artifacts, syntactic verification, modification of the model and validation of the semantics of the model with the business users. We propose Formal Concept Analysis (FCA) as a human-centered and easily understandable instrument to support the modeling of a software system (Ganter et al. 1999, Wille 1999). It is a technique for mathematically describing and visualizing concepts and their interrelationships. In particular, the intuitive visual display was found to be of major importance during a number of case studies and a modeling case in which 20 students took part. The lattice helped in stimulating conscious reasoning over syntactic and semantic errors, inconsistencies and different modeling choices that were made. Amongst others, we gained insight in missing objects, missing or faulty assigned operations, wrong dependencies, alternative solutions, etc. FCA allows the user to reason over the semantics, consistency and relations among UML models. A lattice can automatically be derived from an object - use case interaction matrix and easily be transformed into a UML class diagram. This class model construction procedure based on FCA has the additional advantage of resulting in “normalized” class diagrams. These models contain no more many-to-many relations, no more optional-to-optional relations and no more reflexive relations, leading to less ambiguous class diagrams.

The requirements engineering process is framed in the C-K design science theory (Hatchuel et al. 2003) and each of the four categories of activities can be mapped to one of the four operators of the C-K design square. At the core of the method are multiple successive iterations through a learning loop. The actionable information in the K space, i.e. the use cases
and conceptual model, are transformed to an FCA lattice which can be used for formal verification of the model and proposing modifications to the model. The results are fed back to the domain experts, and the semantic validity of the model is analyzed together with the business user. The FCA lattices serve here as a communication instrument.

The remainder of this section is composed as follows. In section 4.3.2 we introduce the essentials of conceptual model quality and UML class model normalization. Section 4.3.3 discusses FCA, C-K theory and the relevance of these techniques in requirements engineering. Section 4.3.4 shows how C-K theory can be used as a framework for iterative requirements engineering and the relevance is showcased with multiple case studies. Section 4.3.5 describes a validation experiment. In section 4.3.6 related work is presented. Finally, section 4.3.7 concludes the discussion.

4.3.2 Requirements engineering artifacts

4.3.2.1 Use cases, conceptual domain model and quality

The quality of the end product depends greatly on the accuracy of the requirements specification and developers are more and more concentrating on how to improve the early stages of development. Both use cases and the conceptual model are important parts of early development of a software system. According to (Belgamo et al. 2004), these requirements models deserve special attention, since it is in the requirements engineering phase where substantial communication difficulties concentrate and many defects may be introduced in the artifacts. Use Cases were introduced in OOSE (Jacobson et al. 1992) and describe the interactions between the system and the external actors. Such an interaction does not have to be atomic and is usually decomposed into steps indicated in the use case specification. An actor is a specific role played by a system user and represents a category of users that share similar behavior when using the system. By users we mean both human beings as well as external systems or devices communicating with the system. An actor is regarded as a class and users as instances of this class. A use case is a system usage scenario involving one or more actors and the purpose of a use case specification is to describe the flow of events in detail, including how the use case starts, ends, modifies the system and interacts with actors.

One of the major activities includes finding out which classes the software will need in order to satisfy the requirements described in the use cases. The behavior in a system should be exactly that which is required to provide the use case to the users of the system. A conceptual model is a collection of concepts linked together to form a model. Another important step is the allocation of the required functionality to an entity or entities in the conceptual model for each use case. For each step described in the use case
specifications, a responsibility should be identified and allocated to an entity. This is a complex but unavoidable task (Insfran et al. 2002). One of the biggest challenges facing software projects is determining when and how to begin the transition from specifying requirements to working on a system design (Reed 2002). Incomplete or incorrect requirements carry the risk of formulating a design based on sketchy requirements. In (Lindland et al. 1994), a framework is presented for evaluating the quality of conceptual models. A distinction is made between syntactic and semantic quality. Semantic quality issues arise when the model lacks something that the domain contains, or it can include something the domain does not have. In other words, the more similar the model and the domain, the better the semantic quality. The two major semantic goals to be achieved are validity and completeness. Validity means that all statements made by the model are correct and relevant to the problem. Completeness means that the model contains all the statements about the domain that are correct and relevant. In the quality management literature the distinction is also often made between product and process quality (Checkland 1991): product quality focuses on the quality of the end product. Product quality criteria are used to conduct inspections of the finished product and to detect and correct defects. Process quality focuses on the quality of the production process. Process quality focuses on defect prevention rather than detection, and aims to reduce reliance on mass inspections as a way of achieving quality (Deming 1986).

4.3.2.2. UML class model normalization

We propose a new best practice for UML class diagrams called normalization. The goal of normalization is to reduce the ambiguity in conceptual models. Currently, there exists a lot of confusion in the literature about best practices in UML modelling and normalization of conceptual models. According to Frisendal et al. (2010), the biggest problem with UML is its complexity. In business concept modelling, intuition is obstructed by unnecessary complexity such as meta-constructs like aggregation, composition, many-to-many associations, inheritance, etc. which are not really necessary for business users to understand. Ambler (2009) defines class normalization as a process of applying simple rules to reduce coupling and increase cohesion within the object designs. A related approach for improving object diagrams is refactoring (Fowler 1999) which however is typically performed on source code instead of models. Falleri et al. (2008) define normalization as removing all redundancies from class models and finding abstractions. They use FCA and Relational Concept Analysis (RCA) to find possible class, association, attribute or method generalizations in models with the aim of improving their abstraction level (Falleri et al. 2008b).

We start our discussion with 4 examples of problems typically associated with traditional UML class diagrams and how developers can benefit from
normalization. Fig. 4.12a contains a reflexive association example. The model aims to represent a flow of activities. However, the UML diagram does not reveal which the start or ending activity is. Although, the UML standard allows to give a name to the start and end of an association, this is often forgotten by software developers resulting in an ambiguous diagram. Therefore, this flow should be modeled as a separate class. A flow is always characterized by two associations, one with the start activity and one with the end activity. These relationships are mandatory for the association-ends with cardinality 1. The normalized model is displayed in Fig. 4.12b.

Fig. 4.12a class diagram with a reflexive association.

![Diagram](image1)

Fig. 4.12b normalized class diagram without reflexive association.

The second example in Fig. 4.13a shows an often encountered many-to-many association. Li et al. (2001) advise to model associative classes as a separate class and decompose the association into two associations between the two classes and the newly added class. This decomposition changes the many-to-
many association into one-to-many associations that are much easier to realize than many-to-many associations. In object models, associations are instantiated as power sets. However, still an improvement is possible in their new model. The authors introduced an association between the 2 original classes which is superfluous since the new object was introduced which has a mandatory association with both entities. This association can be removed.

Also interesting to consider is the literature on Entity-Relationship modeling (Lanzerini et al. 1990) since UML diagrams can be derived from ER schemata. The standard binary association in UML and ER have the attribute unique on each end. Objects at unique ends are counted only once if they are connected to a particular object several times. At non-unique ends every connection is counted even if several of them lead to the same object. If this property is set to unique, the instantiations of the association form a set, if they are non-unique, a bag. The authors of (University of Cape Town 2007) suggest to represent many-to-many relationships as two one-to-many relationships involving a new entity since it is difficult to implement a many-to-many relationship in a database system. This new structure can be implemented within a relational database system. Feinerer et al. (2007) study this unique and non-unique property. Standard ER does not allow non-unique, but UML 2.0 superstructure specification (Object Management Group 2005) does not make statements about instantiations of associations being a set or bag but the tools are standard on unique. Standard in all UML tools, this multiplicity property for many-to-many associations is set to unique (1) which is another argument for the normalization of many-to-many relationships. If isUnique is set to false, links carry an additional identifier apart from their end values. Many-to-many associations as a consequence make the diagrams and query definition unnecessarily complex and analysts cannot model more than 1 relation between the same objects if unique is set on true.

Each instantiation Rent_i of this association is a tuple (Person_i, Car_i). A problem with this representation is that the same person cannot rent the same car more than once since a tuple can only occur once in a set. This problem can be resolved by instantiating this association with an extra class. Again we see that the two association-ends are mandatory for Person and Car. The normalized model is displayed in Fig. 4.13b.

![Fig. 4.13a class diagram with many-to-many association.](image)
Figure 4.14a contains an example of an optional-to-optional association. In particular when it is important to keep a history record for the association and to perform querying afterwards, optional-to-optional associations should be instantiated by an extra class. For example, a Person can be enrolled in 0 or 1 Session. A Session can have 0 or more participants. The information related to an enrolment in a session cannot be kept by the class Person nor Session unless null values are allowed. The presence of null values may result in unpredictable behavior when queries are performed on the model. This association should be instantiated by an extra class and the information should be kept by this separate class Registration. This class Registration will contain all information about instances of this association which an be considered as a best practice. Again, this relationship is mandatory on the association-ends of Person and Session. The normalized model is displayed in Fig. 4.14b.

Figure 4.13b normalized class diagram without many-to-many association.
Figure 4.15a contains an example of an association between 3 partners, indicated by a diamond in UML notation. UML defines an n-ary association as linking n classes, n > 2 and at each end is a multiplicity and uniqueness constraint. According to Genova et al. (2001), understanding n-ary associations is often very difficult for modellers and analysts. The multiplicity values typically specified for n-ary associations provide only partial understanding and are incompletely defined by UML. The authors reveal an ambiguity in the definition of UML minimum multiplicity for n-ary associations and three alternative interpretations are presented, each with its own problems and unexpected consequences. According to the author, many modellers use the ternary symbol in Fig. 4.15a as an abbreviated version of a ternary association with a hidden binary association. The limping links interpretation has ternary links that only link two instances and leave a blank for the third one. This option is however semantically weak and contradicts the UML definition of n-ary association. The other two interpretations are actual pairs which implies that minimum multiplicity must always be 1, which is not consistent with documentation and practice, the potential pairs interpretation seems correct but has a strange effect when value is 1. the authors propose a different notation similar to our normalization proposal. Based on the Merise method (Rochfeld 1986), the ternary association is replaced by a new entity and three binary associations that simulate the ternary association. This entity is called the intersection entity or associative entity (Song 1995). Each instantiation of this association is a triple (Designer_i, Tool_i, Project_i). Again this triple can only occur once in a set. By instantiating this association with a class, a second usage relation between a designer and a tool for a project becomes possible. Again, the association-ends with the weak entity type Usage are mandatory for the classes Designer, Tool and Project. The normalized model is displayed in Fig. 4.15b.
4.2. AN INTERACTIVE REQUIREMENTS ENGINEERING FRAMEWORK

Fig. 4.15a class diagram with association between more than 2 partners and UML diamond notation.

Fig. 4.15b normalized class diagram without diamond notation.

The special cases of aggregation and composition can be interpreted as follows. In Fig. 4.16a we see that the filled diamond denoting composition in UML is a short notation for the diagram in Fig. 4.16b. In Fig. 4.17a, we see the same for the white diamond denoting aggregation. This diagram in Fig. 4.17b can be further normalized to the model underneath it.
Fig. 4.16a class diagram with composition relationship.

Fig. 4.16b normalized class diagram.
4.2. AN INTERATIVE REQUIREMENTS ENGINEERING FRAMEWORK

Fig. 4.17a class diagram with aggregation relationship.

Fig. 4.17b normalized class diagram.

To summarize, each class diagram can be normalized. A normalized class diagram has the following properties. First, it only contains binary associations and no more associations between multiple parties. In other words, the diamond notation from UML is not needed anymore. Second, there are no more reflexive associations, i.e. from a class to itself. Third, each binary association has at least 1 mandatory side, i.e. with cardinality 1. On
the other side, there can be cardinalities 1, 0..1, 0..*, etc. In other words, in a normalized class diagram there are no more many-to-many and optional-to-optional associations since 1 side is always mandatory.

![Class Diagram Without Normalization](image1)

![Normalized Class Diagram](image2)

Fig. 4.18a class diagram without normalization.

Fig. 4.18b normalized class diagram.

Fig. 4.18a and Fig. 4.18b contains an example of a normalization procedure. Normalization has some advantages. First, normalization is a process that converges to a unique solution. Second, normalized class diagrams automatically lead to normalized database schema during implementation. Normalization does not need to happen at database level since it already was performed at class level.

The result of repeating this normalization procedure is a normalized class schema that is partially ordered to form a lattice structure. In this lattice structure, each association-end on the upper side has cardinality “1”.

4.3.2.3. UML associations and the use case-entity interaction matrix

A question that remains unanswered by UML is when we should model an
association. Nor in UML, nor in ER-modeling there exist objective criteria to
determine when an association should be introduced. A possible answer was
introduced in (Snoeck et al. 1996). An association should be introduced when
2 object types have something in common for a certain span in time. In other
words, that they have a part of their life cycle in common.

We only consider business use cases, while IO or system use cases are not
included in our discussion. We also require that the use cases are atomic, i.e.
that they cannot be further decomposed. An atomic use case will change the
system from one state into another and is defined in terms of how this use
case will create or delete an object or form or break an association between
two instances. Such an atomic use case is equivalent to a system operation in
(Larman 1998). The notion of joint action (D’ Souza 1998) can be used to
represent an atomic use case.

For example, a use case like “manage customer” should be decomposed
into “create customer”, “change customer” and “end customer”. We also do
not consider extend and include operations on use cases. In the situation
where two objects share a part of their lifecycle, there is an objective reason
to introduce an association between them. After composing the use case-
entity interaction matrix, we use FCA to come up with a clustering. Entities
that participate in the same use cases are grouped in FCA concepts based on
the use cases in which they participate. In section 4.3.4 we zoom in on FCA’s
relevance in conceptual modeling.

4.3.3. C-K theory

The Concept-Knowledge theory (C-K theory) was initially proposed and
further developed by Hatchuel et al. (1996), Hatchuel et al. (1999) and
Hatchuel et al. (2002). C-K theory is a unified design theory that defines
design reasoning dynamics as a joint expansion of the Concept (C) and
Knowledge (K) spaces through a series of continuous transformations within
and between the two spaces (Hatchuel et al. 2003). C-K theory makes a
formal distinction between Concepts and Knowledge: the knowledge space
consists of propositions with logical status (i.e. either true or false) for a
designer, whereas the concept space consists of propositions without logical
status in the knowledge space. According to Hatchuel et al. (2004), concepts
have the potential to be transformed into propositions of K but are not
themselves elements of K. The transformations within and between the
concept and knowledge spaces are realized by the application of four
operators: concept \( \rightarrow \) knowledge, knowledge \( \rightarrow \) concept, concept \( \rightarrow \) concept
and knowledge \( \rightarrow \) knowledge. These transformations form what Hatchuel
calls the design square, which represents the fundamental structure of the
design process. The last two operators remain within the concept and
knowledge spaces. The first two operators cross the boundary between the
Concept and Knowledge domains and reflect a change in the logical status of
the propositions under consideration by the designer (from no logical status to true or false, and vice versa).

Design reasoning is modeled as the co-evolution of C and K. Proceeding from K to C, new concepts are formed with existing knowledge. A concept can be expanded by adding, removing or varying some attributes (a “partition” of the concept). Conversely, moving from C to K, designers create new knowledge either to validate a concept or to test a hypothesis, for instance through experimentation or by combining expertise. The iterative interaction between the two spaces is illustrated in Fig. 4.21. The beauty of C-K theory is that it offers a better understanding of an expansive process. The combination of existing knowledge creates new concepts (i.e. conceptualisation), but the activation and validation of these concepts may also generate new knowledge from which once again new concepts can arise.

4.3.4. Iterative requirements engineering process using FCA

According to Wieringa et al. (2006), many of the papers published in the requirements engineering field describe techniques for use in requirements engineering practice: for example, how to improve the process of negotiating requirements or how to build use case models, etc. Unfortunately, there are few research papers that investigate the properties of these techniques, or the problems to be solved by these techniques (Wieringa 2005a, Wieringa 2006b). According to the authors, the absence of such research prevents the
transfer of results of requirements engineering research to practice. Companies will hesitate to adopt techniques of which the properties are not well known, not thoroughly investigated or for which it has not been investigated which problems they solve and under which conditions. The methodological framework we use is C-K theory, which gives a clear structure to the process of iterating back and forth between the human actor and the documents describing the system under development.

A problem is a difference between what is perceived and what is desired, that we want to reduce (Wieringa 2003). An action problem is a desire to change the world; a knowledge problem is a desire to increase our knowledge about the world. Action problems can be classified into two kinds. A design problem is a desire to specify a change and an implementation problem is a desire to implement a specified change. To solve a design problem, we must do two things: Analyze the problem and specify a solution. According to (Wieringa 2003), requirements engineering is the problem analysis part of a design problem. It is about a knowledge problem, which the engineer tries to solve by building a theory about the domain of this problem. This knowledge creation process can be seen as a special case of the unified theory on design, called C-K theory. The notion of design as an expansive process addressed in design theories such as C-K theory should not be confused with the software design phase; although a software design process can be seen as a special case of design reasoning in C-K theory. In this section, we discuss how the requirements engineering process can be framed using C-K theory.

4.3.4.1. C-K theory in requirements engineering

Modeling software systems contains both formal and non-formal steps. These non-formal steps should not be unpredictable or irrational, but should follow a systematic way of thinking (Marincic et al. 2008). In some papers on formal methods, the modeling process is presented as if modelers had all the knowledge about the system before they started modeling. In that case it is possible to build a model in a strictly top-down manner. But modelers usually do not know everything up front about the system that they are modeling. One of the essential aspects of the requirements modeling process is iteratively increasing the knowledge available about the system. The source of information can be technical documents or domain experts. Most likely, the modeler does not have a complete knowledge about the system. The need for a structured approach has been described in the many papers on the soundness issues in requirements engineering. In this section, we give a clear structure to this modeling process by using C-K theory. We particularly focus on the iterative refinement steps, which describe how the model grows from an initial, sketchy, general description to its final version. In our approach, the expert is the driving force behind the modeling of software systems.

Requirements engineering is basically a process of iterating back and forth between a concept and a knowledge space. The knowledge space contains the
information available to the domain expert including initial sketchy requirements for the system under development. This knowledge is then conceptually organized and visualized using the FCA lattices. We perform this conceptualization to put the actionable knowledge available in the K space under scrutiny. In the C space, these lattices are used for verification of the model and to detect inconsistencies, anomalies, missing entities, missing use cases, etc. These newly discovered concepts, anomalies and concept gaps are then activated and used to improve the current model. In the K space, these findings are fed back to the domain expert and the lattices are used for validation of the requirements model. They serve as a communication instrument between the software modeler and the business user, for whom the technical jargon of the software engineer is often difficult to understand. Unspoken assumptions and desires on semantics of the model should be made explicit and communicated to all stakeholders (Wieringa 2001).

This process is graphically described in Figure 4.20. During the K \( \rightarrow \) C step, the FCA lattices are constructed that form the core artifacts of our requirements model construction, verification and validation method. These lattices are built from the entity - use case interaction matrix, but can also be based on a use case - operations or entity - operations interaction matrix. The entity - use case interaction lattice partially orders the entities in the
conceptual model based on the use cases in which they participate. This lattice may reveal missing concepts, entities and use cases, but also issues such as use case participations that should be added or removed. Also anomalies in the behavioral side of the model can more easily be detected because of the non-hierarchical partial order relation. The use case - operations interaction lattice gives insight into the operations needed to complete the use cases. Missing operations, use cases that should or should not have certain operations in their execution scenarios, use cases that should have certain operations in common, etc. can be identified. The entity - operations lattice gives additional insight into the behavior of entities in the conceptual model. In the special case of atomic use cases, i.e. the use cases are not further decomposable, every use case corresponds to an operation in the conceptual model and vice versa. In this case only one lattice, namely the entity-use case lattice is needed. These lattices are used during the C -> C step for formal verification of the model, as a human-centered instrument that facilitates the detection of inconsistencies, anomalies, etc. The original model, the discovered anomalies and the proposed modifications are returned to and discussed with the domain expert during the C \(\rightarrow\) K step. The lattices serve as a communication instrument, between the developer and the domain expert. During the K \(\rightarrow\) K step, these lattices are used for semantic validation by the domain expert. This may result in the addition, modification or deletion of use cases, modifications in the conceptual model, etc. These artifacts may be used as input for a new iteration through the C-K loop.

4.3.4.2. FCA lattice properties and relation with software artifacts

A conceptual model is a dual structure of concepts and their instances called objects and behavioral elements of the model (use cases) in which they participate. These concepts or classes are related through associations. In this section we intend to formalize this dual conceptual model structure with FCA. FCA has a well established mathematical foundation whereas conceptual modeling is to some extent still an ambiguous discipline.

After composing the use case-entity interaction matrix, we use FCA to come up with a clustering. Entities that participate in the same use cases are grouped in FCA concepts. These shared use case participations indicate a shared lifecycle of the objects and the lines interconnecting the concepts in an FCA lattice can be used as associations in a UML class diagram between the objects belonging to the interconnected concepts. From this FCA lattice, a UML class diagram can be automatically derived since an FCA lattice based on a correct entity-use case interaction matrix is isomorphic to the correct UML model. Each line interconnecting 2 concepts can be seen as a direct association between the own objects in the extent of the corresponding concepts. A line between 2 concepts in the lattice means that between 2 instantiations exactly 2 own classes in the extent of the 2 concepts, one tuple can be created. This follows from the object-use case interaction matrix.
One of the main contentions of this section is that FCA leads to a normalized class diagram. One of the consequences is that FCA can be used to detect missing classes. An important benefit of FCA over other techniques is its non-hierarchical partial ordering of concepts. This is more expressive than traditional hierarchical tree-like structures, which was already stated in (Christopher 1965). Hierarchical decomposition gives a distorted and simplified view that does not necessarily conform to reality.

Table 4.8 summarizes the properties of an FCA lattice based on a use case-entity interaction matrix and how it can be used to distill the correct model from this interaction matrix and the original UML model. In case there is a discrepancy between one of these columns for the model, solutions can be proposed based on these best practice guidelines. The first column contains an interesting observation made by looking at the FCA lattice. Since this lattice is based on the entity-use case matrix, the corresponding statement in the third column is true. The statements made in column 2 are best practices for the UML model that were introduced to make the use case–entity interaction matrix and UML model consistent with each other. Multiple possibilities for model revision paths can be associated with one lattice observation. For example, take the second row of Table 4.8, if there is no direct line between the concepts with A and B as own objects in their extent in the FCA lattice, then there should be no direct association between A and B in the UML model. This is required because the UML model is isomorphic with the FCA lattice unless of course the use case-entity interaction matrix contains an error. In the use case-entity interaction matrix both A and B should have use cases in which either A or B does not participate. If the domain expert however says that there should be a direct association between A and B which is mandatory for A in the UML model, then the FCA lattice flags an error in the use case-entity interaction matrix and use case participations should be propagated from A to B. This will have the consequence that there will appear a direct line in the FCA lattice between A and B. FCA can be used to flag errors in both the UML model and use case-entity interaction matrix.

Table 4.8. Normalization guidelines

<table>
<thead>
<tr>
<th>Entity-use case FCA lattice</th>
<th>Normalized UML model</th>
<th>Use case-entity matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Entity A belongs to a concept lower in the lattice than B's concept and there is a direct line between both concepts.</td>
<td>There should be a direct association between A and B, which is mandatory (cardinality “1”) for B. B is existence dependent of A.</td>
<td>Entity A should participate in all use cases in which entity B participates.</td>
</tr>
<tr>
<td>2 There is no direct line between concepts with A</td>
<td>There should be no direct association cases in which either A and B have use cases in which either A or B does not participate.</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.8 summarizes the properties of an FCA lattice based on a use case-entity interaction matrix and how it can be used to distill the correct model from this interaction matrix and the original UML model. In case there is a discrepancy between one of these columns for the model, solutions can be proposed based on these best practice guidelines. The first column contains an interesting observation made by looking at the FCA lattice. Since this lattice is based on the entity-use case matrix, the corresponding statement in the third column is true. The statements made in column 2 are best practices for the UML model that were introduced to make the use case–entity interaction matrix and UML model consistent with each other. Multiple possibilities for model revision paths can be associated with one lattice observation. For example, take the second row of Table 4.8, if there is no direct line between the concepts with A and B as own objects in their extent in the FCA lattice, then there should be no direct association between A and B in the UML model. This is required because the UML model is isomorphic with the FCA lattice unless of course the use case-entity interaction matrix contains an error. In the use case-entity interaction matrix both A and B should have use cases in which either A or B does not participate. If the domain expert however says that there should be a direct association between A and B which is mandatory for A in the UML model, then the FCA lattice flags an error in the use case-entity interaction matrix and use case participations should be propagated from A to B. This will have the consequence that there will appear a direct line in the FCA lattice between A and B. FCA can be used to flag errors in both the UML model and use case-entity interaction matrix.
and $B$ as own objects in extent.

3 There is a concept with 2 or more own use cases shared by $A$ and $B$ and no owner entity. Replace direct (many-to-many) association by contract entity with which association is mandatory for $A$ and $B$.

4 There is no upward path from entity $A$’s concept to $B$’s concept. On implementation level this means $A$ can not call the proprietary operations of $B$.

5 There is a node $R$ with two use cases $a$ and $b$ as label and no own entities. Entities $A$ and $B$ lower in the lattice participate in these use cases and there is no upward path between $A$ and $B$ nor are there any entities on the path from $A$ to $R$ or $B$ to $R$.

There is no association between $A$ and $B$ nor a contract entity which interconnects them.

Entity $A$ does not participate in all the use cases in which $B$ participates.

There is no entity $S$ that participates in use cases $a$ and $b$. $A$ and $B$ do not participate in the creation and deletion cases of $S$.

Table 4.8 contains the different types of normalization steps that can be undertaken. A sequence of normalization steps is called a normalization path and each path converges to a unique solution. In each case study we will show the relevance of normalization. In section 4.3.4.5, we see an application of the rule in row 3 of Table 4.8. In section 4.3.4.6 we see a combination of applying rules in row 1 and 2. In section 4.3.4.4 we see an application of the rule in row 5.

The FCA lattice components that facilitate the detection of anomalies during semantic validation of the model are explained in Table 4.9. Table 4.9 summarizes the results of section 4.1.
Table 4.9. Model anomalies and FCA analysis components

<table>
<thead>
<tr>
<th>Model anomalies</th>
<th>FCA analysis component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing or faulty use case participations</td>
<td>Duality of partial ordering based on entities and use cases reveals faulty and incomplete participations</td>
</tr>
<tr>
<td>Missing proprietary use cases such as creation and deletion of an entity</td>
<td>We gain insight into the owned or proprietary use cases of an entity by looking at the concept that owns this entity. The owned use cases by this entity are attached as labels.</td>
</tr>
<tr>
<td>Missing entity</td>
<td>A concept with two or more owned use cases and no own entity.</td>
</tr>
<tr>
<td>Superfluous entity</td>
<td>A concept with more than one own entity indicates in the current model 2 entities have the same behavior and either one of them should be removed or one of them should participate in additional use cases.</td>
</tr>
</tbody>
</table>

The relevance of rule 1 is showcased in section 4.3.4.3 and 4.3.4.4. The relevance of rule 2 is showcased in section 4.3.5.2. The relevance of rule 3 is explained and demonstrated in section 4.3.4.5.

We first showcase our method on some toy examples to make the reader familiar with FCA-based conceptual modeling. Then we provide a real-life case study. We show the detection of some of the typical errors made by modelers using the rules of section 4.3.4.2.

4.3.4.3. Case study: Book Trader System

We now illustrate this process using the Book Trader system introduced in (Liang 2003). The collaboration diagrams and conceptual model displayed in Figures 21a – f contained some errors that remained undetected to the authors. These errors are:

- Entity Line does not participate in any use cases according to the Collaboration diagrams.
- Separate create and delete use cases are missing for entities Book, Order, Line, Customer and Invoice.
- In the collaboration diagrams, Order is involved in the provide-quantity operation which is in contradiction with the class diagram in which Line is having this operation.
- A better modelling option would be to allow new prospective customers to register. Currently, according to this model a Customer
can only be registered as part of the use case Place Order.

- The class Invoice cannot be seen as an object in a business model since it has no further behaviour after its creation, however it should be included in the information architecture.

We found these errors while studying the three lattices and this analysis process will now be discussed in detail. During the $K \rightarrow C$ step, we constructed a cross table indicating which entities participate in which use cases. This table displayed in Table 4.10 is based on the collaboration diagrams and the conceptual model in the original paper and we will call it the entity - use case interaction matrix. Each collaboration diagram represents one use case. When an entity appears in de collaboration diagram, this is registered in the table with a C, M or E. The C indicates that the use case creates the entity, M indicates reads or modifies the entity and E indicates it terminates the entity.

### Table 4.10. Faulty entity-use case matrix

<table>
<thead>
<tr>
<th></th>
<th>PLACE ORDER</th>
<th>DELIVER BOOK</th>
<th>CHECK STOCK</th>
<th>CHECK CREDIT</th>
<th>GENERATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order</td>
<td>C</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Book</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>C</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Invoice</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>

The corresponding lattice is displayed in Fig. 4.22. During the $C \rightarrow C$ step we performed a syntactic verification of the model quality based on the lattice visualization. Using the lattice in Fig. 4.22, we found that the entity Line does not participate in any use cases according to the original collaboration diagrams. Second, when analyzing the attributes of the FCA concepts we found that separate create and delete use cases were missing for most entities defined. UML best practice guidelines advised the definition of both creation and deletion use cases (Rumbaugh et al. 2004). These are some clear modeling anomalies. In the $C \rightarrow K$ step we propose modifications to resolve these anomalies. In the $K \rightarrow K$ step the improved model is communicated to and validated with the domain expert.
Fig. 4.21a. Collaboration diagram (adapted from Liang 2003)

Fig. 4.21b. Collaboration diagram (adapted from Liang 2003)

Fig. 4.21c. Collaboration diagram (adapted from Liang 2003)

Fig. 4.21d. Collaboration diagram (adapted from Liang 2003)
4.2. AN INTERACTIVE REQUIREMENTS ENGINEERING FRAMEWORK
During the K → C step of our second iteration through the design square we constructed a matrix, displayed in Table 4.11 that maps the (non-atomic) use cases against the operations that instantiate them.

Table 4.11. Faulty use case-operations matrix

<table>
<thead>
<tr>
<th></th>
<th>CREATE_NEW_ORDER</th>
<th>PROVIDE_ALL_FEATURES</th>
<th>PROVIDE_TOTAL_PRICE</th>
<th>PROVIDE_QTY</th>
<th>PROVIDE_PRICE</th>
<th>REDUCE_STOCK</th>
<th>REDUCE_CREDIT_BALANCE</th>
<th>PROVIDE_CUSTOMER_NAME</th>
<th>PROVIDE_CUSTOMER_NO</th>
<th>PROVIDE_CREDIT_BALANCE</th>
<th>REDUCE_CREDIT_BALANCE</th>
<th>REDUCE_ORD_NO</th>
<th>REDUCE Quản lý</th>
<th>PROVIDE_CLIENT_NAME</th>
<th>PROVIDE_CLIENT_NO</th>
<th>PROVIDE_CREDIT_BALANCE</th>
<th>REDUCE_CREDIT_BALANCE</th>
<th>REDUCE_ORD_NO</th>
<th>REDUCE Quản lý</th>
<th>PROVIDE_CLIENT_NAME</th>
<th>PROVIDE_CLIENT_NO</th>
<th>PROVIDE_CREDIT_BALANCE</th>
<th>REDUCE_CREDIT_BALANCE</th>
<th>REDUCE_ORD_NO</th>
<th>REDUCE Quản lý</th>
<th>PROVIDE_CLIENT_NAME</th>
<th>PROVIDE_CLIENT_NO</th>
<th>PROVIDE_CREDIT_BALANCE</th>
<th>REDUCE_CREDIT_BALANCE</th>
<th>REDUCE_ORD_NO</th>
<th>REDUCE Quản lý</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place order</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check stock</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deliver Book</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate invoice</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check credit</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
step the operations - use case lattice helped us to identify some of the semantic discrepancies between the original conceptual model and the real world, for example, we see that the record-new-customer operation only occurs in the use case Place_Order. New prospective customers cannot be created without an order. This is an unrealistic constraint that is also incorporated in the original domain model. One could consider to make each mandatory association into an optional association would not better reflect reality. During C \rightarrow K and K \rightarrow K, modifications are proposed and their semantics are discussed.

The third matrix, displayed in Table 4.12, maps the entities against the operations in which they are involved.

**Table 4.12. Faulty entity-operations matrix**

<table>
<thead>
<tr>
<th>Order</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Book</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Invoice</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

The corresponding lattice is displayed in Figure 4.24. When we look at the concept with entity Invoice in its extent, we see that Invoice only participates in a create event. The Invoice entity has no further behavior and should not
be modeled as a separate class. Business modeling analysts will typically not consider this to be an object however it should of course be integrated within the data and information architecture. The creation of an invoice should be modeled as an operation or an event from which a document is generated. Again, creation and delition operations are missing for most entities. Moreover, Line is involved in the provide-quantity operation which is in contradiction with the use case - entity interaction matrix.

Fig. 4.24. Entity – operations interaction matrix

4.3.4.4. Case study: Hotel Administration System

In this section, we showcase how the FCA lattices were used for the semantic validation of a hotel administration system model. Customers can make reservations for a particular room type. Reservations must be confirmed by a letter. If such letter is not received in time, the reservation is cancelled. When a guest checks in for the first time, his details are registered. At the end of the stay, an invoice is sent to the customer who made the reservation. Suppose a business analyst would come up with Table 4.13. Fig. 4.25 shows an excerpt of the initial UML model that was developed for this hotel administration system.
Table 4.13. Incorrect entity-use case matrix for the Hotel Administration

<table>
<thead>
<tr>
<th>Entity</th>
<th>CUSTOMER</th>
<th>TYPE</th>
<th>ROOM</th>
<th>RESERVATION</th>
<th>ROOM</th>
<th>STAY</th>
<th>CONSUMPTION</th>
<th>PAYMENT</th>
<th>REMINDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>cr_customer</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_customer</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cr_room_type</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_room_type</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reserve</td>
<td>M M C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>confirm</td>
<td>M M M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cancel</td>
<td>M M E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>check_in</td>
<td>M M E M C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no_show</td>
<td>M M E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cr_room</td>
<td>M C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_room</td>
<td>M E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_stay</td>
<td>M M E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>consume</td>
<td>M M M C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bill</td>
<td>M M M E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>invoice</td>
<td>M M M C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pay</td>
<td>M M M E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>remind</td>
<td>M M M M M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_invoice</td>
<td>M M M M E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

At first this UML model seems to be alright. During the K → C step the lattice in Fig. 4.25 was obtained from the use case – entity interaction matrix. In this example, the use cases were not further decomposable and every use case corresponds to an operation in the conceptual model. Such a use case can also be called an event. In this case, only one matrix namely the one that maps entities versus use cases is needed. The interplay between use cases and entities and the additional partial ordering relation helped however to reveal some semantic issues in the original UML model during the C → C step.
We used the lattice for semantic validation of the model. First, the person or company who made the reservation for the guest, may be someone else than the person who is staying in the hotel. Second, we see that there is no upward path from the concept with entity Customer to the concept with use cases bill and consume. In other words, the entity Customer does not participate in the use cases bill and consume although guests of the hotel should be able to make consumptions. Second, Customer does not participate in the use case e-stay, whereas the guest of the hotel should be able to check out of the hotel at the end of his stay. Based on these observations we decided to add an entity Guest to the conceptual model during the C → K step. The observation that Customer is not associated with Room is perfectly alright. The Customer makes a Reservation for a Room type, a specific Room is only relevant for the guest of the hotel. Fig. 4.26 contains the correct UML model that was obtained after discussion with and validation by the domain experts during the K → K step and contains the entity-use case interaction lattice corresponding to this correct model.

Fig. 4.25 Incorrect hotel administration model and entity-use case interaction lattice based on original incorrect matrix
4.3.4.5. Case Study: Ordering system

This section discusses the development of the ordering system for computer hardware, office material, etc. of the university KULeuven in Belgium. This ordering process is a standardized process for KULeuven, where an order is placed when a request for ordering a computer is received from an employee. A request is sent to Dell to construct a computer. In a standard setting, payment is only made after the goods were delivered to the KUL. Fig 4.27a contains the initial UML class model for an excerpt of this system, which is not normalized and does not allow for the detection of potential conflicts in use case execution order. After building the use case-entity interaction matrix in Table 4.14 and lattice we obtain the lattice in Fig. 4.27b.
Fig. 4.27a Ordering system model with missing entity

During the C → C step we use FCA for formal verification of the models, detection of missing entities, detection of use cases for which no responsible entity has been assigned, etc. We see that Dell computer and KULeuven jointly participate in 3 main use cases. If this joint participation is not coordinated, both KULeuven and Dell computer can impose sequence constraints on the order of execution for these 3 use cases. This may result in a situation of deadlock. Indeed, for the KULeuven these use cases have a fixed order of execution, namely:

1. Order
2. Deliver
3. Pay

However, for computer manufacturer Dell who also participates in these use cases, the ordering of the use cases is:

1. Order
2. Pay
3. Deliver

We see there is one node in the lattice with no entity label and 3 use case labels attached. We see that the entities KULeuven and Dell participate in these use cases, but an entity responsible for coordinating this joint participation in these use cases is missing. The business processes of KULeuven and Dell have to communicate with each other and the sequence of use case execution should be coordinated. During the C → K step an extra entity named Contract is added as a contract between these two actors to guide and coordinate the collaboration. Figure 4.27b contains the UML model of an excerpt of this system. During the K → K step this modification is explained to the business user, together with the constraints imposed by this “contract” object type on use case execution order.
### Table 4.14. Faulty entity-use case matrix

<table>
<thead>
<tr>
<th></th>
<th>CONSTRUCT</th>
<th>ISSUE_ROLE</th>
<th>PAY</th>
<th>DELIVER</th>
<th>ORDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>KULeuven</td>
<td>C</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Dell</td>
<td>C</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
</tbody>
</table>

![FCA lattice with missing entity and correct ordering system model with contract entity added after normalization](image)

#### 4.3.4.6. Case study: Elevator repair system

In this case study an engineer works for exactly one office. This engineer is responsible for repairing broken elevators. When an elevator fails, an interrupt is sent to an office and an engineering is sent to the elevator. Fig. 4.28, contains the initial UML model and Table 4.15 the entity-use case interaction matrix.

### Table 4.15. Initial entity-use case interaction matrix

<table>
<thead>
<tr>
<th></th>
<th>REGISTER_EL</th>
<th>REGISTER_OFICE</th>
<th>REGISTER_ENGINEER</th>
<th>CHANGE_ENGINEER_DETAILS</th>
<th>ARCHIVE_ENGINEER</th>
<th>REPORT_INTERRUPT</th>
<th>CR_REPAIR_ACTION</th>
<th>ARCHIVE_REPAIR</th>
<th>REPORT_INE</th>
<th>ARCHIVE_INE</th>
<th>ARCHIVE_REPA</th>
<th>ARCHIVE_REPAIR_A</th>
<th>ARCHIVE_ACTION</th>
<th>CLOSE_INTERRUPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevator</td>
<td>C</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Office</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
There are however some issues with this model, the model is not normalized and this has the following consequences. During the C → C step we analyzed the requirements artifacts. The UML model shows a direct association between Office and Elevator. Assuming that the unique property has been set (which is reasonable, as unique is “standard”), this means that the same elevator cannot send an interrupt more than once to the same office. The associations between instantiated objects form a set and in a set, a tuple can only occur once. This is an undesirable property of the non-normalized UML model containing the many-to-many relationship. In a many-to-many relationship no entity is foreseen to record more than one association and its properties between the same instantiated objects.

In the FCA lattice based on the use case-entity interaction matrix in Fig. 4.28 we see no direct line between the concepts of Elevator and Office. The lattice indicates there are 2 events not coordinated by an entity and in which both entities participate. Based on the lattice recommendations a new entity should be added during the C → K step with which both Elevator and Engineer have a mandatory association. This entity replaces the original direct association and coordinates the 2 use cases. We call this entity “Interrupt”.

In the FCA lattice there is no direct line between the concept of Engineer and the concept of Elevator. In other words, there is no association in the FCA lattice based on the use case - entity interaction matrix that corresponds to the direct many-to-many association between Engineer and Elevator in the UML model. The consequence of the direct many-to-many association in the UML model is that the same Elevator cannot get a second repair by the same engineer.
Engineer. This is also an undesired property which can be solved by normalization during the C → K step, i.e. in this case by replacing the association by a separate class associated to the entities Elevator and Engineer.

By studying the lattice during the C → C step we see there is no upward path from Office to Engineer. There is also no direct line from the concept with entity Office to the concept with entity Engineer. According to the FCA lattice based on the use case entity interaction matrix there should be no direct association between Office and Engineer or the use case-entity matrix should be modified. In the UML model, Engineer has a mandatory relationship with Office. When an Engineer is created, modified or archived, the corresponding Office should at least be notified (as part of the execution of these use cases) of these changes since Engineer is existence dependent of an office. Other alternatives are also possible, however for this case we chose this configuration, in practice the management will have to decide. If the Office object disappears the Engineer object should be terminated too since the mandatory association disappears. We see however that Office is not involved in all use cases in which Engineer is involved. This problem was solved by propagating the use case participations from Engineer to Office during the C → K step.

Interrupt use cases are not coordinated and Engineer is not involved in these use cases. Also repair use cases are not coordinated by an entity. FCA suggests to introduce Interrupt and RepairAction as 2 entities in the UML model. Then there is still no relation between RepairAction and Interrupt. This makes it impossible to know afterwards which Engineer worked on which Interrupt. There is no direct association between Interrupts and Repairs. As a consequence, Key Performance Indicators cannot be generated from this model and neither data mining nor business intelligence can be applied. The correct associations are missing fundamentally. The solution is to propagate all use case participations from RepairAction to Interrupt. In the FCA lattice, RepairAction will be right above Interrupt. In the UML diagram there is now a direct association which is mandatory for RepairAction. The correct UML model is displayed in Fig. 4.29a, the correct entity-use case interaction matrix in Table 4.16 and the corresponding FCA lattice in Fig. 4.29b.
### Table 4.16. Entity – use case interaction matrix for normalized model

<table>
<thead>
<tr>
<th></th>
<th>Elevator</th>
<th>Office</th>
<th>Engineer</th>
<th>Interrupt</th>
<th>RepairAct ion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevator</td>
<td>C</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Office</td>
<td>C</td>
<td>E</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Engineer</td>
<td>C</td>
<td>M</td>
<td>E</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Interrupt</td>
<td></td>
<td>C</td>
<td>M</td>
<td>M</td>
<td>E</td>
</tr>
<tr>
<td>RepairAct ion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C, E</td>
</tr>
</tbody>
</table>

Fig. 4.29a  Normalized UML model
At first it seems the model contains a circular constraint, namely starting from RepairAction it is possible to navigate to Office by Interrupt, but it is also possible to navigate from Engineer to Office. In the original model both instantiations were considered to be the same, this invariant can be written in OCL as follows (Meyer 1997):

\[
\]

One of the consequences of this OTIS-based example is the following. If we impose this constraint, it is possible that some Offices receive too many requests and other Offices too little. This is an example in which the circular constraint should not be imposed. In the normalized model, engineers of no matter what Office can do a repair action, which solves the former problem. The result is that the notion of Office became a virtual entity in the system and not a physical office anymore.

4.3.5. Validation experiment

We empirically showed the relevance of our research with a modelling experiment in which 20 students took part.
4.3.5.1. Participants

The goal of the experiment was to evaluate the practical feasibility of the C- K design loop supported by FCA for software requirements engineering. The experiment conducted with the collaboration of students, consists of a modelling exercise in which they should distill entities and elementary processes from a textual description of a business process. Then, they should compose a matrix in which is indicated which entities participate in which processes. Their solution is then handed over to the data analyst who uses FCA to detect anomalies, missing object types, etc. and gives suggestions to the students for improving their original model. They then implement these changes. The first experiment is performed in collaboration with students of the course ‘Ontwikkeling van Bedrijfstoepassingen’ in March 2010. The experiment took place as part of an exercise session of the class.

4.3.5.2. Setup of experiment 1

The experiment was built around the Web Shop case.

Web shop case:
In the simple Web shop registered customers can create shopping carts. They can choose from various products, and select them to put them in the shopping cart. Once put in the shopping cart, a product can be confirmed (definitely, and hence archived) or removed from the cart. Of course, at the end of a shopping session, the cart must be paid, and is next delivered to the customer.

Table 4.17. Formal context of web shop case

<table>
<thead>
<tr>
<th></th>
<th>CR_PRODUCT</th>
<th>CH_PRODUCT</th>
<th>END/Product</th>
<th>CR_CUSTOMER</th>
<th>CH_CUSTOMER</th>
<th>END_CART</th>
<th>CR_CART</th>
<th>PAY_CART</th>
<th>DELIVER_CART</th>
<th>ADD_ITEM</th>
<th>REMOVE_ITEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>C M E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>C M E M</td>
<td>M M M M M M M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shopcart</td>
<td>C M E M</td>
<td>M M M M M M M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cartitem</td>
<td>C M E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4.17 contains the use case-entity interaction matrix and Fig. 4.30 the correct normalized class model. The C indicates that the use case creates the entity, M indicates that it modifies the entity and E that it deletes the entity. Fig. 4.31 contains the FCA lattice based on this correct matrix.

Fig. 4.30 Web shop case

Fig. 4.31 FCA lattice based on correct matrix
4.3.5.3. Results of experiment 1

Starting from the use case-entity interaction matrix provided by the students, we derived an FCA lattice. None of the students succeeded in making a fully correct model for the system. The different solutions of students could however be categorized in four broad categories of structurally similar models containing some typical modeling errors. These FCA lattices and in particular the partial ordering of entities made it easy to identify faulty or missing associations and missing or faulty use cases and use case participators. During the semantic analysis of the students’ solutions the lattices made it easy to identify the different modeling options students chose and where they confused different options or mixed them together. This reasoning over models and model choices helped in achieving a uniform software model. We found it more straightforward to distill this information on syntactic and semantic correctness from the FCA lattices than from the UML model and entity-use case interaction matrix. In these traditional artefacts information is more shattered and no partial ordering is available, whereas FCA provides a condensed and complete overview of the model’s syntax and semantics.

The following errors were regularly found using the FCA lattices:

- For each entity the proprietary use cases are made visible as the labels connected to the concept corresponding to the entity. 16/20 students forgot one or more terminate entity use cases, whereas, only 3 students forgot a create entity use case. An excerpt of the lattices is displayed in Fig. 4.32 where the student forgot the EndCUSTOMER and DeliverCART use cases.

![Image of FCA lattice with error example](image)

**Fig. 4.32** Termination use case omission
4.2. An Interactive Requirements Engineering Framework

- By following the lines upward, we can easily see in which use cases an entity participates. By following the lines downward one can see which objects participate in a use case. We found that for use case “RemoveITEM from cart”, “AddITEM to cart”, students tend to forget that also the initiator of this use case, i.e. the Customer, also participates in these use cases together with Product and ShopCart. Only 1/20 students modeled this correctly furthermore only 7/20 students did not forget that the initiator of “CrCART”, the Customer also participates in this use case. Fig. 4.33 contains an example of lattice excerpts showing that Customer does not participate in these 3 use cases.

\[\text{Fig. 4.33 Use case participation error}\]

- Multiple use case participations shared by multiple entities but not coordinated by a contract entity can be found in the lattice as a node with own use cases but no own objects. For 17/20 students the lattice contained such a node where entities may impose different sequence constraints on use case execution order resulting in deadlock. Fig 4.34 contains as example of such a node.
Finally, the FCA models helped in reasoning over modeling choices made by students. From this lattice, the corresponding UML class diagram was distilled and feeded back to the student with the found anomalies.

4.3.6. Related work

4.3.6.1. FCA in requirements engineering

Typically, use case descriptions are written in natural language although sometimes controlled vocabularies are used. Duwel (1999) and Duwel et al. (1998) used FCA to identify class candidates in use case descriptions. The authors considered the use cases as objects of a formal context and the nouns identified within the text were considered as a starting point for a class hierarchy. Tilley et al. present a case study applying Duwel’s approach to an Object-Z specification (Tilley et al. 2003). Richards et al. (2002) and Richards et al. (2002b) apply FCA to use cases in an attempt to reconcile descriptions written by different stakeholders using a controlled vocabulary and grammar. According to the authors, the formal nature of this controlled language facilitates the analysis of use cases to identify misunderstandings, inconsistencies and conflicts. Moreover, similar concepts and differences in terminology were identified using concept lattices. FCA has also been used...
during the design phase of the software engineering process. In this section we focused on early requirements engineering.

### 4.3.7. Conclusions

In this section we showed the relevance of normalized class models in early requirements engineering. FCA was used to derive a concept lattice from the use case-entity interaction matrix. This lattice was used for syntactic verification and semantic validation of the UML model and use case–entity interaction matrix. After a successive number of normalization steps, a normalized UML class model is obtained with desirable properties such as the absence of many-to-many, optional-to-optional and reflexive associations. The iterative process of analyzing and improving the requirements artifacts was framed in C-K theory. The C-K loop consists of four main phases, elaboration, verification, modification and validation. In each phase, the visualization of FCA plays a pivotal role.
CHAPTER 5

Formal Concept Analysis of Temporal Data

In this chapter we investigate the possibilities of expanding FCA with a time dimension. Our first two studies describe the application of the temporal variant of FCA, named Temporal Concept Analysis (TCA), to two police datasets. In the first case study, FCA is used in combination with TCA for detecting and profiling potential human trafficking suspects. The TCA lattices turned out to be especially useful for visually analyzing the evolution of potential suspects and their social network over time. This method has been further developed for pro-actively finding potential terrorism suspects and analyzing their radicalization process over time. The representation of temporal data with TCA however has its limitations. In the third case study, we combined FCA with Hidden Markov Models to distill care processes from breast cancer patient treatment data. This study clearly showed the synergy between process and data discovery and resulted in incremental improvements of patient care processes.

5.1. Detecting and profiling human trafficking suspects with Temporal Concept Analysis

Human trafficking and forced prostitution are a serious problem for the Amsterdam-Amstelland police (the Netherlands)\(^8\). In this section, we present a method based on Temporal Concept Analysis for detecting and profiling human trafficking suspects (Poelmans et al. 2010c). Using traditional Formal Concept Analysis, we first build a lattice with early warning indicators to find persons who are potentially involved in human trafficking. These persons are then subjected to an in-depth investigation and we use Temporal Concept Analysis for constructing visual profiles of these suspects. Finally, the evolution of the social network surrounding these persons is visually analyzed using Temporal Concept Analysis.

\(^8\) This section has been published in Poelmans, J., Elzinga, P., Viaene, S., Dedene, G. (2010) A method based on Temporal Concept Analysis for Detecting and Profiling Human Trafficking Suspects. IASTED International Conference on Artificial Intelligence.
5.1. DETECTING AND PROFILING HUMAN TRAFFICKING SUSPECTS

5.1.1. Introduction

Human trafficking is defined as the recruitment, transportation, harboring and receipt of people for the purpose of slavery, forced labor and servitude (United Nations 2001). Trafficking is considered as a serious problem by the police and government of the Netherlands (Ministerie van Justitie 2009). Unfortunately, research has shown that only a small number of victims of human trafficking makes a statement to the police (Tyldum et al. 2005). Recently, the Amsterdam-Amstelland police installed a human trafficking team to actively search for potential suspects.

Police databases contain a large number of reports describing observations made by officers during motor vehicle inspections, interventions, id controls, etc. These documents are often spread over different database systems, may have different identification keys and only limited information browsing functionality is provided. To analyze a potential suspect, officers currently have to search all these databases and manually inspect all the reports, because some of these observations might contain indications for human trafficking.

In this section, we make use of the techniques known as Formal Concept Analysis (FCA) (Ganter et al. 1999, Wille 1982, Priss 2005, Poelmans et al. 2010b) and Temporal Concept Analysis (TCA) (Wolff 2000a, Wolff 2001, Wolff 2005). TCA was only recently introduced as an extension to FCA and can be used to analyze temporal data. First, we use FCA to identify a reliable set of early warning indicators and to extract potential suspects for in-depth analysis from the large amount of unstructured reports. Second, we showcase how TCA is used to automatically compose a comprehensible and visually appealing “profile” of these suspects that can be used by police officers to quickly decide whether or not this person should be monitored. Finally, we analyze the social network of the suspect and its evolution over time using TCA.

The remainder of this section is composed as follows. In section 5.1.2, we give some background on human trafficking and the current way of working by the police. In section 5.1.3, we elaborate on the dataset used. In section 5.1.4, we discuss the human trafficking indicators. In section 5.1.5, the essentials of TCA theory are introduced. In section 5.1.6, the detection and profiling of human traffickers is explained. Finally, section 5.1.7 concludes the discussion.
CHAPTER 5: FORMAL CONCEPT ANALYSIS OF TEMPORAL DATA

5.1.2. Backgrounder

5.1.2.1. Human trafficking

Human trafficking is the fastest growing criminal industry in the world, with the total annual revenue for trafficking in persons estimated to be between $5 billion and $9 billion (United Nations 2004). The council of Europe states that “people trafficking has reached epidemic proportions over the past decade, with a global annual market of about $42.5 billion” (Equality division 2006). Trafficking victims are typically recruited using coercion, deception, abuse of power or outright abduction. Threats, violence and economic leverage such as debt bondage often make a victim consent to exploitation. Exploitation includes forcing people and children into prostitution, forced labor or slavery. Women are particularly at risk from sex trafficking. Criminals exploit lack of opportunities, promise good jobs or opportunities for study and then force the victims to become prostitutes.

Throughout agents and brokers who arrange the travel and job placements, women are escorted to their destinations and delivered to the employers. Upon reaching their destinations some women learn that they have been deceived about the nature of the work they will do; most have been lied to about the financial arrangements and conditions of their employment; and find themselves in coercive or abusive situations from which escape is both difficult and dangerous (Miko 2000).

Due to the illegal nature of trafficking and differences in research methodology, the exact extent is unknown. Rough estimates suggest that 700,000 to 2 million women and girls are trafficked across international borders every year (O’Neill 1999). The majority of transnational victims are trafficked into commercial sexual exploitation. Since the fall of the Iron Curtain, the impoverished former Eastern bloc countries such as Albania, Moldova, Romania, Bulgaria, Russia, Belarus and Ukraine have been identified as major trafficking source countries for women and children. It is estimated that 2/3 of women trafficked for prostitution worldwide annually come from Eastern Europe, three quarters have never worked as prostitutes before (Highes 2000). For Amsterdam, official statistics say that the majority of victims have either the Dutch nationality, come from Eastern European countries or from West Africa (Dettmeijer-Vermeulen 2008).

5.1.2.2. Current situation

Victims of human trafficking rarely make an official statement to the police. The human trafficking team is installed to proactively search police databases for any signals of human trafficking. Unfortunately, this turns out to be a laborious task. The investigators have to manually read and analyze the police reports, one by one, because only an estimated 10% of the information containing human trafficking indications has been labeled as such by police
officers. As soon as the investigators find sufficient indications against a person, a document based on section 273f of the code of criminal law is composed for the person under scrutiny. Based on this report, a request is sent to the Public Prosecutor to start an in-depth investigation against the potential suspects. After permission is received from the Public Prosecutor, the use of special investigation techniques such as phone taps and observation team is allowed.

5.1.3. Dataset

Our dataset consists of 69788 general police reports from the year 2008. These general reports contain observations made by police officers during motor vehicle inspections, during a police patrol, when a known person was seen at a certain place, etc. An example of a report is displayed in Figure 5.1. We also have 1101 reports at our disposal, for which we know with certainty that they are related to human trafficking. These include arrestment reports, a statement made by a victim to the police, charges against a human trafficking suspect, etc. These reports date back to the years 2005-2008.

<table>
<thead>
<tr>
<th>Title of incident</th>
<th>Observation xxx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reporting date</td>
<td>23-03-2008</td>
</tr>
<tr>
<td>Project code</td>
<td>Prostitution related observation</td>
</tr>
<tr>
<td>Location</td>
<td>Amsterdam Wallengebied yyy</td>
</tr>
<tr>
<td>Suspect (male) (18-45yr)</td>
<td>Zzz</td>
</tr>
<tr>
<td>Involved (female) (18-45yr)</td>
<td>Kkk</td>
</tr>
<tr>
<td>Involved (female) (18-45yr)</td>
<td>Uuu</td>
</tr>
</tbody>
</table>

**Reporting of observation**

On the night of 23 of march 2008 we stopped a car with a Bulgarian license plate for routine motor vehicle inspection. It was a Mercedes GLK with licence plate BL XXX. The car was driving around in circles in a prostitution area. On the backseat of the car we noticed two well dressed young girls. We asked for their identification papers but they didn’t speak English or Dutch. The driver of the car was in possession of their papers and told us that they were on vacation in the Netherlands for two weeks etc.

*Fig. 5.1. Example police report*
5.1.4. Human trafficking indicators

The first step of the research consists of finding a set of indicators on the basis of which we can decide whether or not a person might be involved in human trafficking. We make a distinction between early and late indicators. Many of the strong human trafficking indicators available in some of the reports were only obtained after special investigation techniques such as phone taps or the use of observation teams. One of the main goals of the research consists of being able to automatically find suspects that are potentially related to human trafficking from the large set of observational reports. We thus aim at finding cheap, fast and reliable early warning indicators that give a good indication that a person might be involved in human trafficking. Table 5.1 shows some of the indicators we obtained after analyzing the data at hand. To detect these indicators in the unstructured text of police reports, we use a thesaurus. For each indicator, we have some terms in the thesaurus that indicate whether or not the indicator is present in the police report. To construct a thesaurus, find relevant terms and refine the set of indicators, we use the methodology presented in (Poelmans et al. 2009).

This method is an iterative knowledge discovery from unstructured text approach based on FCA and Emergent Self Organizing Map. During the successive iterations, the data is explored, interesting new concepts are distilled from the texts and conceptually analyzed in a lattice. The focus lies on finding those concepts that help in identifying human trafficking suspects.

Table 5.3. Excerpt of the human trafficking indicators

<table>
<thead>
<tr>
<th>Static indicators</th>
<th>Indicators with time dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>- nationality</td>
<td>- red light district: Wallen</td>
</tr>
<tr>
<td>- violence</td>
<td>- red light district: Ruysdaelkade</td>
</tr>
<tr>
<td>- minors involved</td>
<td>- regularly visiting suspicious club</td>
</tr>
<tr>
<td>- restriction of personal freedom</td>
<td></td>
</tr>
<tr>
<td>- id-problems</td>
<td></td>
</tr>
<tr>
<td>- carrying large amount of money</td>
<td></td>
</tr>
<tr>
<td>- forcing person to work in bad condition</td>
<td></td>
</tr>
<tr>
<td>- prostitute involved</td>
<td></td>
</tr>
<tr>
<td>- dependency relationship</td>
<td></td>
</tr>
<tr>
<td>- injury observed</td>
<td></td>
</tr>
<tr>
<td>- woman not speaking</td>
<td></td>
</tr>
</tbody>
</table>
5.1. Detecting and Profiling Human Trafficking Suspects

- regularly dropping of girls at club
- expensive car
- woman in car
- vacation
- car trade

Indicators coming from social network
- seen with known suspect

We distinguish between three types of indicators. Static indicators typically contain socio-demographical information, such as the nationality of the person involved. We also have indicators with an inherent time dimension, such as for example when a person was signaled several times in a prostitution area and always declared to the police that he was on vacation. Finally, we also have indicators that come from analyzing the social network of the suspect, e.g. if he is seen together with a known criminal.

5.1.5. Temporal Concept Analysis

Temporal Concept Analysis (TCA) is a mathematical theory that was introduced in scientific literature about nine years ago. TCA is based on Formal Concept Analysis (FCA) and addresses the problem of conceptually representing time. TCA is particularly suited for the visual representation of discrete temporal phenomena. In the following sections, we first introduce the essentials of FCA theory. Then, we discuss the extension of FCA with a time dimension, i.e. TCA.

5.1.5.1. FCA essentials

FCA concept lattices are used to describe the conceptual structures inherent in data tables without loss of information by means of line diagrams yielding valuable visualizations of real data (Stumme et al. 1998). In a previous paper, we analyzed the concept of domestic violence using FCA (Poelmans et al. 2009d). The main difference with domestic violence is that there is a time dimension involved in human trafficking. Suspects are often spotted several times by the police and it is important to incorporate this time dimension in the visualization of the data. FCA can be used as an unsupervised clustering technique (Wille 2002, Stumme 2002) and police reports containing terms from the same term clusters are grouped in concepts.

The starting point of the analysis is a database table consisting of rows $M$ (i.e. objects), columns $F$ (i.e. attributes) and crosses $T \subseteq M \times F$ (i.e. relationships between objects and attributes). The mathematical structure
used to represent such a cross table is called a formal context \((M, F, T)\). An example of a cross table is displayed in Table 2. In this table, reports of police observations (i.e. the objects) are related (i.e. the crosses) to a number of terms (i.e. the attributes); here a report is related to a term if the report contains this term. The dataset in Table 2 is an excerpt of the one we used in our research. Given a formal context, FCA then derives all concepts from this context and orders them according to a subconcept-superconcept relation, which results in a line diagram (a.k.a. lattice). Full details on FCA can be found in chapter 2.

Table 5.2. Example of a formal context

<table>
<thead>
<tr>
<th>Prostitution area</th>
<th>Expensive car</th>
<th>Bulgarian</th>
<th>ID problems</th>
<th>Large money amount</th>
<th>Women in car</th>
</tr>
</thead>
<tbody>
<tr>
<td>report 1</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>report 2</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>report 3</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>report 4</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>report 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

The set of all concepts of a formal context combined with the subconcept-superconcept relation defined for these concepts gives rise to the mathematical structure of a complete lattice, called the concept lattice of the context, which is made accessible to human reasoning by using the representation of a (labelled) line diagram. The circles or nodes in this line diagram represent the formal concepts. The shaded boxes (upward) linked to a node represent the attributes used to name the concept. The non-shaded boxes (downward) linked to a node represent the objects used to name the concept. The information contained in the formal context can be distilled from the line diagram in by applying the following reading rule: an object “\(g\)” is described by an attribute “\(m\)” if and only if there is an ascending path from the node named by “\(g\)” to the node named by “\(m\)”.

Retrieving the extension of a formal concept from a line diagram such as the one in Figure 5.2 implies collecting all objects on all paths leading down from the corresponding node. To retrieve the intension of a formal concept, one traces all paths leading up from the corresponding node in order to collect all attributes. The top and bottom concepts in the lattice are special: the top concept contains all objects in its extension, whereas the bottom concept contains all attributes in its intension. A concept is a subconcept of all concepts that can be reached by travelling upward. This concept will inherit all attributes associated with these superconcepts.

5.1.5.2. TCA essentials
The pivotal notion of TCA theory (Wolff 2002, Wolff et al. 2003) is that of a conceptual time system (Wolff 2005). An example of a data table of a conceptual time system is displayed in Table 5.3.

**Table 5.3. Data table of a conceptual time system**

<table>
<thead>
<tr>
<th>Time granule</th>
<th>Time part</th>
<th>Event part</th>
<th>Event part</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Date</td>
<td>Expensive car</td>
<td>Prostitutions area</td>
</tr>
<tr>
<td>0</td>
<td>26/01/2008</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>21/02/2008</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>15/02/2008</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>13/03/2008</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>27/04/2008</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>01/06/2008</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>14/06/2008</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>18/06/2008</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3 contains the observations of one real person at several points of time. To make a single observation, police officers needed some time, varying from a few minutes to a few hours. We abstract from the duration of an observation and use the notion of a point of time, also called time granule. We thus start from a set of which the elements are time granules. In table 5.3 for example, we have 8 time granules. For describing the observations, we use a single valued context with \( G \) as its set of formal objects. This context consists of an event part and a time part. The indicators observed at each of these time granules are described in the event part of the data table. In contrast to (Wolff 2005), where a multi-valued context was used, we only need a single-valued context here. Formally, the conceptual time system we use can be described as follows.

Let \( T := (G, M, I_T) \) and \( C := (G, E, I_C) \) be two single valued contexts respectively on the same object set \( G \). Then the pair \((T, C)\) is called a conceptual time system on the set \( G \) of time granules. \( T \) is called the time part and \( C \) the event part or space part of \((T, C)\). The combination of \( T \) and \( C \) is denoted by \( K_{TC} := TC \). It is the context of the conceptual time system \((T, C)\). The object concepts of \( K_{TC} \) are called situations, the object concepts of \( C \) are called states and the object concepts of \( T \) are called time states. The sets of situations, states and time states are called the situation space, the state space and the time state space of \((T, C)\) respectively. In the visualization of the data, we want to express the “natural temporal ordering” of the observations. In the TCA lattice, a time relation \( R \) is introduced on the set \( G \) of time granules of a conceptual time system. We speak of a conceptual time system with a time relation (CTST).
Let \((T, C)\) be a conceptual time system on \(G\) and \(R \subseteq G \times G\). Then the triple \((T, C, R)\) is called a conceptual time system (on \(G\)) with a time relation. On the set \(G := \{0, 1, 2, 3, 4, 5\}\) of time granules we introduce the relation \(R := \{(0, 1), (1, 2), (2, 3), (3, 4), (4, 5)\}\) shortly described as \(0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5\). We also need the notions of transitions and life tracks. The basic idea of transition is a “step from one point to another”. Each arrow in Figure 5.3 represents a “transition of the suspect” and corresponds to an element of \(R\). The transitions in Figure 5.3 form a life track of the suspect.

5.1.6. Detecting and profiling human traffickers

The method we propose consists of 3 steps. First, we aim at extracting all persons that are potentially involved in human trafficking from the large set of observations. For this extraction process we make use of the visualization capabilities offered by FCA. Second, we construct a detailed profile of these persons using TCA. TCA allows for comprehensible visualization of the indicators found and the evolution over time of these persons. Third, we use TCA to analyze the social network and its evolution over time around these persons.

5.1.6.1. Detecting suspects with FCA

For detecting potential suspects from the large amount of observations, we make use of an FCA lattice. The persons mentioned in the reports are the objects of the lattice. The indicators contained in the observational report of these persons are combined in one feature vector. For the static indicators and indicators coming from the social network analysis, we make use of a single-valued formal context. For the indicators with time dimension we make use of a many-valued formal context. This many-valued context is then scaled to a single valued context. This results in an FCA lattice as displayed in Figure 5.2.

From domain experts, we know that logistics is an important aspect of human trafficking and forced prostitution. Girls have to be transported from the place where they live (often under surveillance of their pimp) to the place where they work. Whereas it is difficult for the police to detect human trafficking and to infiltrate the underground world of forced prostitution, there are a lot of observations performed on the street amongst others during motor vehicle inspections that might give some clues on where to look. The lattice in Figure 5.2 for example contains some of these early warning indicators for forced prostitution and zooms in on some of the logistic aspects of human trafficking.

One target group we selected for in-depth analysis were the 11 persons who were seen in an expensive car with Bulgarian licence plate in a
prostitution area. From domain experts, we know that Bulgarian human traffickers are very active in Amsterdam. We also selected the 10 persons who were involved with prostitutes and had problems with legitimating themselves or the women. We also retrieved the 4 Bulgarian persons who were seen with a large amount of money in the red light district. Finally, we investigated the 26 Bulgarian persons who were involved with prostitutes and committed violence. In the tool we developed, it is possible to click on the persons in which we are interested and automatically a TCA profile (see section 5.1.6.2) is constructed.

Fig. 5.2. FCA lattice used for detecting potential human trafficking suspects
5.1.6.2. Profiling suspects with TCA

For composing a detailed profile of suspicious persons, we make use of TCA. The goal of these profiles is to collect all relevant information around a person and provide a conceptual overview of this person and his evolution over time. The TCA lattice serves as an intuitive knowledge browser making the interaction between the police officer and data more efficient. Based on this lattice, police officers can easily judge whether or not there is sufficient evidence available for the Public Prosecutor to start closely monitoring this person.

In Figure 5.3, a lattice profile of one of the potential suspects selected using the FCA lattice in Figure 5.2 is shown. The arrow in Figure 5.3 shows the life track connecting the successive observations of the person. One can see that the person with Bulgarian nationality was observed 8 times in Amsterdam by the police. During these 8 times, he declared 5 times that he was on vacation in Amsterdam. He was spotted 3 times in a prostitution area, 4 times in an expensive car and 3 times with women in his car. He once had problems with legitimating himself or the girls he was transporting and he once carried a large amount of money with him. These are all clear indications that this person is probably involved in human trafficking. Based on analysis of this lattice, it was decided to start an investigation into this person.

Figure 5.4 contains a TCA lattice profile of another human trafficking suspect discovered using the lattice in Figure 5.2. The lattice in Figure 5.4 shows that the reports related to this person contain some very strong indicators for forced prostitution. The Bulgarian person has been spotted several times during which he was involved with prostitutes and companies that have a dodgy reputation when it comes to forced prostitution. Moreover, there has been one incident in which the person restricted the personal freedom of a woman who was dependent of him. He also used violence against her. Again, these are clear indicators for human trafficking and an investigation was started against this person after analyzing this profile.
5.1. Detecting and Profiling Human Trafficking Suspects

Fig. 5.3. TCA lattice profile of a human trafficking suspect

Fig. 5.4. TCA lattice used for profiling a human trafficking suspect
5.1.6.3. Network evolution analysis using TCA

For each suspect, there are multiple observations available. In each of these observations, one or more other persons may be mentioned. For each of these persons, we collected all data available and incorporated this into the lattice of the suspect. The mathematical foundation we used for this analysis is that of a “conceptual time system with actual objects and a time relation” (CTSOT) (Wolff 2005).

This time line helps us gain insight in the role of the person in the overall network, the evolution over time of his cooperation with other persons, the importance of his position, the role and evolution of the persons surrounding him etc. For example, consider the lattice in Figure 5.5. This is a simplified version of a lattice based on real life data, which contains the life tracks of 3 persons. Consider the life track of “Person 1”. At time granule 1, he was observed by the police in an expensive car and with two women at the back of his car. These women were probably prostitutes according to the officers. They seemed to be scared of the officers; they didn’t say much and didn’t carry their ID with them. Two months later, one of these women “Person 2” was observed in a prostitution area with a Bulgarian male “Person 3” and with bruises on her arm.

![Fig. 5.5. TCA lattice used for gaining insight in social network of human trafficking suspects](image-url)
Based on these observations and the visualization of these relationships, we were able to gain insight in the structure of this network. “Person 1” was probably into the logistics part of the network. “Person 2” was probably a woman who was forced into prostitution and “Person 3” is probably her pimp. Based on these observations, police officers had enough indications to start an official investigation against “Person 1” and “Person 3”, whereas the lattice containing observations of “Person 1” alone did not suffice to start a case against “Person 1”.

5.1.7. Conclusions

In this section, we have investigated the practical usefulness of Temporal Concept Analysis for analyzing police data containing an inherent time dimension. We first used a regular FCA lattice containing early warning indicators to extract potential human trafficking suspects from observational police reports. After finding potential suspects using an FCA lattice, TCA turned out to provide interesting visualization capabilities for composing a detailed profile of these persons and their social network. For each point in time the person was spotted by the police, this TCA lattice contains the human trafficking indications that were observed. These observations are chronologically ordered in the lattice using the TCA notion of life track of a suspect. The visualization of the social network of suspects and its evolution over time using TCA also resulted in new insights during police investigations. In particular, this social network lattice provided insights in the different roles of the observed persons in the global network. Future work includes the embedding into daily policing practice of the techniques presented in this section.

5.2. Terrorist threat assessment with Temporal Concept Analysis

The National Police Service Agency of the Netherlands developed a model to classify (potential) jihadists in four sequential phases of radicalism. The goal of the model is to signal the potential jihadist as early as possible to prevent him or her to enter the next phase. This model has up till now, never been used to actively find new subjects. In this section, we use Formal Concept Analysis to extract and visualize potential jihadists in the different phases of radicalism from a large set of reports describing police observations. We employ Temporal Concept Analysis to visualize how a possible jihadist radicalizes over time. The combination of these instruments allows for easy decision-making on where and when to act.

---

5.2.1. Introduction

In the modern day globalized world, the ease of terrorist network information exchange is characterized by contact moments through the internet and an absence of time and location restrictions. The amount of information available to police forces is continuously increasing and many police forces are not ready for handling data amounts of this size. As a consequence, proactively observing potential threats to our national security becomes increasingly difficult. The National Police Service Agency (KLPD) of the Netherlands started a new Intelligence Led Policing (ILP) project with the aim of collecting terrorist-related information in visually appealing models (Knowledge in Models, KiM, project) to ease the extraction and sharing of actionable knowledge. The KiM project is part of the Program Improvement by Information Security Awareness (VIA). This program is a partnership between the National Coordinator of Counterterrorism (NCTB), the National Forensic Institute (NFI), the General Intelligence and Security Service (AIVD) and the KLPD. Shortly described, the program includes research on and implementation of methods and techniques for supporting police services in their fight against terrorism.

One of the results of this project was the development of a model describing the radicalization process a potential jihadist may pass through before committing attacks. This model consists of 4 phases and its feasibility and practical usefulness have been validated by members of the intelligence services on known suspects. After the validation of this model on known jihadis, the next and probably most important, step consists of finding unknown potential suspects by applying the model to the large amounts of structured and unstructured text available in the police databases.

In this section, we make use of the techniques known as Formal Concept Analysis (FCA) (Ganter et al. 1999, Priss 2005) and Temporal Concept Analysis (TCA) (Wolff 2005). Contextual attribute logic (Ganter et al. 1999b) is used to group and transform the terrorism indicators available into new attributes for generating the concept lattices. After extracting the potential suspects for each phase of the model, a detailed profile based on TCA is constructed giving the history of the suspect and his current level of threat to national security (Elzinga et al. 2010).

The remainder of this section is composed as follows. In section 5.2.2, we give some background on Jihadism in the Netherlands and the four phase model of radicalism used by the KLPD. In section 5.2.3, we elaborate on the dataset used. In section 5.2.4, the essentials of FCA and TCA theories are introduced. In section 5.2.5, the research methodology is explained and the results of the analysis are presented. Finally, section 5.2.6 concludes the discussion.
5.2.2. **Backgrounder**

5.2.2.1. **Home-grown terrorism**

In November 2004 the Dutch society was confronted for the first time with an act of terrorism, namely the brute murder of the Dutch film maker Theo van Gogh. The people suddenly realized that the ideology of violent jihad against the West had also established a foothold in the Netherlands and that the Netherlands as well had become a scene of terrorist violence. It ensued that the murderer, and most other members of the extremist network to which he belonged, were young Muslims born and bred in the Netherlands (AIVD 2006, AIVD 2007).

The latter fact has been seen as a confirmation of the new phase in Islamist terrorism, the phase in which the threat emanates principally from extremist European Muslims who are prepared to commit attacks in their own country, also known as the *European jihad*. The AIVD formulated four general trends in the development of jihadism (AIVD 2006). The first and most important is the evolvement from exogenous foreign terrorist threat to indigenous *home-grown* terrorism. This threat has led to the project VIA, Information Security Awareness coordinated by the National Coordinator of Counterterrorism (NCTb). One of the results of this project is the development of a four phase model of Muslim radicalization by the National Police Service Agency. This model will be discussed in detail in the next section.

5.2.2.2. **The four phase model of radicalism**

The four phase model of radicalism, displayed in Figure 5.6, developed by the National Police Service Agency, is based on the idea a jihadist might pass through several phases before he or she might commit serious acts of terrorism. Several indicators (i.e. words and/or sentences) are associated with each phase which are used to decide based on automated text analysis, to which phase a subject belongs. Due to National Security reasons the indicators can not be published. Interested and authorized intelligence services can contact the National Police Service Agency (www.politie.nl/klpd). An exception is made for the indicator “change of behavior” from type 2-A. Some of the keywords belonging to this indicator are the phrases “not shaking hands with women”, “wearing traditional clothes”, “suddenly let grow a beard” and “Islamistic marriage”.

The model should be interpreted in a bottom up fashion. If 4 or more indicators of type 1 become true or 2 or more of type 2-A, than the subject enters the preliminary phase. But if the number of type 2-A comes below 2 or the number of type 1 comes below 4, then the subject will leave the preliminary phase. If 5 or more indicators of type 1 becomes true and 6 of
type 2-B then the subject will enter the Social alienation phase, etc.

Fig. 5.6 The four phase model of radicalism

In the preliminary phase the subject experiences a crisis of confidence; confidence in the authorities is undermined. At this point it is not so much a matter of an ideological rift, but certainly of distrust. Many young Muslims, especially those who have grown up in the West, turn to Islam in search of their identity and place in Western society. Often their parents still live in accordance with the strict traditional norms and values of Islam, which the young people can no longer relate to. They seek a new place for themselves within Dutch society, where their ethnic, religious and national identity can be a balanced part of the whole.

In the social alienation phase a small minority of these young Muslims cannot handle this situation. On the one hand, the first generation of migrants look down on them for becoming too ‘Dutchified’; on the other hand, they do not really fit in with their Dutch peers because they are viewed in terms of their origins. This is generally where a shift occurs from the desire for a national identity to the desire for a religious identity. Strict Islam, as a guiding principle in their lives, provides certainty and stability because it tells them precisely what to do and what not to do. This increases their susceptibility for the ideology of strict, extremist religion, and makes them feel alienated from the rest of society. This alienation finds expression in increasing rejection of Dutch society.

In the Jihadization phase the subjects are characterized by strong radical
Islamic convictions and the fact that they condone violence. Strong involvement in a radical group may ultimately lead to a willingness to support terrorists in the Netherlands or elsewhere in the world. This may include all sorts of support (e.g. funding). This phase entails further alienation from society and an even greater readiness to make an active contribution to the Jihad. The subjects’ firm belief in the rightness of their radical ideology and of radical Islam may lead to recruitment activities to convince others of radical Islamic beliefs and possibly also of the necessity of the Jihad. Isolation from the rest of the world is part of a gradual process.

The last phase, Jihad/Extremism, is a phase of total isolation. The subjects’ entire lives are governed by their radical Islamic beliefs. This is the last step before carrying out Islamist terrorist acts. In this phase, subjects are prepared to use violence themselves to achieve their objectives. In most cases, the definitive preparation for perpetrating an attack takes the form of physical training, often at a training camp abroad. The final step is actually carrying out violent activities.

5.2.2.3. Current situation

All police forces in the Netherlands (25 police regions and the National Police Service Agency) have a monitoring task of collecting information about potential jihadists. Due to the nature of their activities, potential jihadists will avoid contact with the police and other legal authorities as much as possible. The consequence is that finding new potential jihadists is like finding a needle in the haystack. Attempts were made to search the national police database BlueView containing over 50 million documents. Unfortunately this turned out to be a laborious task.

The four phase model is not used yet as an instrument for finding new potential jihadists from large datasets, but as a checklist. To apply the model on large datasets, the KLPD has started a cooperation with the police Amsterdam-Amstelland who is investigating intelligent text mining applications, like the classification system for domestic violence (Elzinga et al. 2009). This application has been used to evaluate the first version of the four phase model on the police dataset of Amsterdam. The results of this investigation has led to fine tuning the conditions imposed by the model to maximize the recall and to find as many potential jihadists as soon as possible.

5.2.3. Dataset

Our dataset consists of 166577 general police reports from the years 2006 (41990), 2007 (54799) and 2008 (69788) from the region Amsterdam-Amstelland, which holds the communities Amsterdam, Amstelveen,
Uithoorn, OuderAmstel and Diemen. These general reports contain observations made by police officers during motor vehicle inspections, during a police patrol, when a known subject was seen at a certain place, etc. Next to general reports there are incident reports like car accidents, burglary, violence cases, etc. There are two reasons why we have chosen for analyzing the general reports. Since the implementation of an Intelligence Led Policing program at the Amsterdam-Amstelland police, the number of general reports has been growing rapidly over the years. The unstructured text describing the observations made by police officers has a lot of underexploited potential for finding potential extremists or radicalizing subjects. The challenge is to find new potential jihadists within the huge amount of general reports.

An example of a report is displayed in Figure 5.7 where two police officers asked two repeat offenders information about a third subject, called C. The reason of the inquiries is that the officers might have an indication that C might be a recruiter.

<table>
<thead>
<tr>
<th>Ziekteregistratie</th>
<th>2008069601-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Titel oorzaak</td>
<td>Aandachtvraag</td>
</tr>
<tr>
<td>Datum Kennisname</td>
<td>10-02-2008</td>
</tr>
<tr>
<td>Diagnose</td>
<td>O04 Interne abdominale dorp</td>
</tr>
<tr>
<td>Plaats oorzaak</td>
<td>Uithoorn</td>
</tr>
<tr>
<td>Betrokken (maan)</td>
<td>C. geh. land (10)</td>
</tr>
<tr>
<td>Adres</td>
<td>Zevy</td>
</tr>
</tbody>
</table>


Fig. 5.7 Example police report

5.2.4. Formal Concept Analysis

FCA arose twenty-five years ago as a mathematical theory (Ganter et al.
It has over the years grown into a powerful framework for data analysis, data visualization and information retrieval. FCA is particularly suited for exploratory data analysis because of its human-centeredness and can be used as an unsupervised clustering technique (Wille 2002, Stumme 2002). Police reports containing terms from the same term clusters are grouped in concepts.

The starting point of the analysis is a database table consisting of rows $M$ (i.e. objects), columns $F$ (i.e. attributes) and crosses $T \subseteq M \times F$ (i.e. relationships between objects and attributes). The mathematical structure used to represent such a cross table is called a formal context $(T, M, F)$. FCA concept lattices are derived from this table and used to describe the conceptual structures inherent in these data tables without loss of information by means of line diagrams yielding valuable visualizations of real data (Stumme et al. 1998). In chapter 3, we analyzed the concept of domestic violence, where FCA was used to support human actors in their information discovery and knowledge exercise (Poelmans et al. 2009). The main difference with domestic violence are the possible combinations and the frequencies of the indicators which trigger a subject in a phase together with the time dimension describing the subjects radicalization process.

### 5.2.4.1. Temporal Concept Analysis

Temporal Concept Analysis (TCA) is an extension of traditional FCA that was introduced in scientific literature about nine years ago (Wolff 2005, Wolff et al. 2003). TCA addresses the problem of conceptually representing time and is particularly suited for the visual representation of discrete temporal phenomena. The pivotal notion of TCA theory is that of a conceptual time system. In the visualization of the data, we express the “natural temporal ordering” of the observations using a time relation $R$ on the set $G$ of time granules of a conceptual time system. We also use the notions of transitions and life tracks. The basic idea of a transition is a “step from one point to another” and a life track is a sequence of transitions. Full details on TCA can be found in section 5.1.5.2.

### 5.2.5. Research method

The method we propose is summarized in Figure 5.8. First, we extract all subjects who have at least one attribute from the large set of observations with FCA. Second, we construct lattices for each phase of jihadism. Third, we use TCA to profile the selected subjects and their evolution over time.
Fig. 5.8 The process model of extracting and profiling potential jihadists

We used a toolset which was developed for text mining in large sets of documents, extracting entities from these sets and generating cross tables in various formats. It has been used to develop and to apply knowledge models for amongst others detecting domestic violence (Elzinga et al. 2009). The toolset uses a thesaurus where indicators can be defined with specific properties. For the purpose of this investigation, the thesaurus and toolset are extended with the property of range of numbers of different occurrences of an indicator which must true.

5.2.5.1. Extracting potential jihadists with FCA

For detecting potential jihadists from the large amount of observations, we make use of an FCA lattice. The subjects mentioned in the reports are the objects of the lattice. The indicators observed during the observations for these subjects are combined in one feature vector. This results in an FCA lattice as displayed in Figure 5.9.
Out of 166578 documents 18153 subjects are selected into the cross table of FCA. Each of the subjects meets at least one of the 35 indicators of the original model. These indicators are grouped together based on the four phase model of the KLPD.

Table 5.4 Results of extraction of subjects

<table>
<thead>
<tr>
<th>Attribute</th>
<th># of subjects applied to</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31</td>
</tr>
<tr>
<td>2-A</td>
<td>7</td>
</tr>
<tr>
<td>2-B</td>
<td>737</td>
</tr>
<tr>
<td>2-C</td>
<td>749</td>
</tr>
<tr>
<td>2-D</td>
<td>837</td>
</tr>
</tbody>
</table>

Table 5.4 shows the number of subjects who meet the requirements of the attributes of the four phase model. In the next section we will showcase how the combinations of type 1 and type 2 indicators will reveal the subjects in the different phases. One of the advantages of FCA theory is the ability to zoom in and out on the data and to create smaller lattices by amongst others
deselecting the attributes from the main lattice.

5.2.5.2. Constructing Jihadism phases with FCA

The next step consists of constructing a lattice for each phase of jihadism and showing subjects. The FCA lattice serves as an intuitive knowledge browser making the interaction between the police officer and data more efficient. Based on this lattice, police officers can easily extract subjects for in-depth investigation. Figure 5.8 shows the process model of finding potential jihadists. The four lattices will be discussed from left to right.

The first lattice shows the preliminary phase where 38 subjects are detected. An in depth search after these 38 subjects revealed that 19 were highlighted correctly. The other 19 subjects were mostly persons of the domestic sphere of the subject and therefore frequently reported in the same documents with the potential jihadists. Out of the 19 correctly highlighted subjects, 3 were previously unknown by the police Amsterdam-Amstelland, but known by the National Police Agency Service. The second lattice shows 5 subjects for the social alienation phase which were all highlighted correctly. The third lattice shows 11 subjects for the Jihadization phase where 8 subjects are highlighted correctly. The fourth and last lattice shows 2 potential jihadists, who both are highlighted correctly.

5.2.5.3. Build detailed TCA lattice profiles for subjects

To show how the selected subject radicalizes over time a TCA lattice is constructed. C. from the example report is a subject who satisfies the conditions of all phases. Figure 5.10 shows the TCA lattice of C. We clearly see his radicalization process over time in action (black arrow). There were 8 observations of C. that did not trigger sufficient conditions for entering one of the four terrorism threat phases. In 29/9/2006, C. for the first time appeared under the preliminary phase and 13 months later again he was observed and again fulfilled the requirements of the preliminary phase. 5 months later, C. for the first time had all the properties of the social alienation phase and climbed from the fourth to the third phase of alert. Afterwards he was categorized 6 times under the second phase of alert: jihadism. In 17/6/2008 he reached the highest point of alert: Jihad extremism (red oval). Afterwards he was spotted 2 times by the police, once in the Jihadism phase and once outside any phases (2 arrows).
5.2. TERRORIST THREAT ASSESSMENT WITH TCA

5.2.6. Conclusions

In this section, we showed that the combination of techniques known as Formal Concept Analysis and Temporal Concept Analysis provides the user with a powerful method for identifying and profiling potential jihadists. We built a set of attributes based on the original knowledge model of radicalism which is used when searching the police reports. Out of 166,577 police reports we distilled and visualized 38 potential jihadism suspects using FCA. TCA is used to analyze the radicalisation over time of the potential jihadists. Avenues for future research include the embedding of this sandbox discovery model into operational policing practice and applying.

5.3. Combining business process and data discovery techniques for analyzing and improving integrated care pathways.

Hospitals increasingly use process models for structuring their care processes. Activities performed to patients are logged to a database but these data are rarely used for managing and improving the efficiency of care processes and quality of care. In this section, we propose a synergy of
process mining with data discovery techniques. In particular, we analyze a dataset consisting of the activities performed to 148 patients during hospitalization for breast cancer treatment in a hospital in Belgium. We expose multiple quality of care issues that will be resolved in the near future, discover process variations and best practices and we discover issues with the data registration system. For example, 25% of patients receiving breast-conserving therapy did not receive the key intervention “revalidation”. We found this was caused by lowering the length of stay in the hospital over the years without modifying the care process. Whereas the process representations offered by Hidden Markov Models are easier to use than those offered by Formal Concept Analysis, this data discovery technique has proven to be very useful for analyzing process anomalies and exceptions in detail.

5.3.1. Introduction

An increasingly competitive health care market forces hospitals to search for ways to improve their processes in order to deliver high quality of care while at the same time reducing costs (Anyanwu et al. 2003). According to (Skinner et al. 2008), the solution to poor quality is not to increase the supply of physicians or specialists or hospital beds, but instead to improve health care systems and incentives to ensure that existing physicians and hospitals provide the best possible quality at the lowest cost. Integrated care pathways are structured multi-disciplinary care plans which detail the essential steps in the care process of a population of patients with a certain clinical problem (Campbell et al. 1998). The aims to achieve with care pathways are improving quality and efficiency of care, to standardize the outcomes of the provided care, to facilitate communication between healthcare professionals and to allow for systematic continuing audit. Care pathways are business process models which describe the expected progress of the patient through the care process and try to model the most standard frequent care pathway, based on expert prior knowledge.

Till date, the continuous monitoring, analysis and improvement of the care pathway's performance was performed in an ad hoc, manual and labor-intensive way. This approach however has some limitations. Modifications to the care process are performed in an ad hoc way and their success can only be measured by the impact of these modifications on the Key Performance Indicators (KPIs). This retrospective impact analysis can only be done after several months, which is an unacceptable long time window in healthcare management. Moreover, this standard model does not capture process

---

5.3. COMBINING BUSINESS PROCESS AND DATA DISCOVERY TECHNIQUES

variations, nor process exceptions and the root causes for inefficiencies are not known. Moreover, in practice there is often a significant gap between what is prescribed or supposed to happen and what actually happens. Process mining is an interesting method for gaining insight into what happens in a healthcare process for a group of patients with the same diagnosis. In (Mans et al. 2009) the applicability of process mining in the healthcare domain was investigated, using Petri-Nets. The idea of process mining (Quaglini 2009) is to extract, monitor and improve real processes by extracting knowledge from event logs.

In this section, we use a unique combination of process discovery techniques and data discovery techniques to gain a deeper understanding of an existing breast cancer care process and the actual activities performed on the working floor to discover process inefficiencies, exceptions and variations immediately and to search for the root causes of inefficiencies. We propose and use a new approach based on Hidden Markov Models to discover a process model from event sequences. Formal concept Analysis (FCA) is used to analyze the characteristics of the clusters of patients that emerged from this process discovery exercise and vice versa to find groups of patients to feed into the process discovery methods (Poelmans et al. 2010d, Poelmans et al. 2010f).

The remainder of this section is composed as follows. In section 5.3.2 we introduce the essentials of business process discovery, Hidden Markov Models and the HMM-based techniques that are proposed for process discovery. In section 5.3.3, we elaborate on FCA as a data discovery technique. In section 5.3.4, we discuss the dataset used. Section 5.3.5 describes the methodology and the results of our discovery exercise. Finally, section 5.3.6 rounds up with conclusions.

5.3.2. Business process discovery

In contrast to process modeling, which is developing a top-down representation of a “to-be” process reality, process discovery is a bottom up approach that tries to gain an understanding of the as-is process realities that are existing at the operational work floor. Discovering irregularities, exceptions and variations by means of analytics is essential in developing process and workforce intelligence. Statistical techniques often consider exceptions as nuisance information and eliminate them as noise. According to (Maruster et al. 2006), statistical techniques are able to capture the general process model rather than the process model containing exceptional paths. For discovering process exceptions, anomalies and variations, the combination of learning techniques, mining and clustering is required to gain sufficient insights in the processes. Most workflow mining methods use Petri-Net like models. In (Maruster et al. 2002), simulated process logs of hospital-wide workflows, containing events like “blood test” or “surgery”
were used to build Petri-Net like models. In (Blum et al. 2008) a statistical approach, using Hidden Markov Models (HMMs) is taken to model the workflow inside the Operation Room. These probabilistic models offer a greater degree of flexibility and are a better option for healthcare, where traditional process mining techniques do not work well (Ferreira et al. 2007).

5.3.2.1. Hidden Markov Models

A Hidden Markov Model (HMM) is a statistical technique that can be used to classify and generate time series. A HMM (Rabiner 1989) can be described as a quintuplet $I = (A, B, T, N, M)$, where $N$ is the number of hidden states and $A$ defines the probabilities of making a transition from one hidden state to another. $M$ is the number of observation symbols, which in our case are the activities that have been performed to the patients. $B$ defines a probability distribution over all observation symbols for each state. $T$ is the initial state distribution accounting for the probability of being in one state at time $t = 0$. For process discovery purposes, HMMs can be used with one observation symbol per state. Since the same symbol may appear in several states, the Markov model is indeed “hidden”. We visualize HMMs by using a graph, where nodes represent the hidden states and the edges represent the transition probabilities. The nodes are labelled according to the observation symbol probability.

5.3.2.2. HMM-based process discovery

There are multiple advantages of using HMMs for process discovery:

- A lot of (open source) algorithms have been published for analyzing and understanding HMMs (e.g. Expectation Maximization, Viterbi algorithm for most probable path for a given pattern of observations, etc.)

- Micro patterns of actor behavior (e.g. medical acts that belong together) can be easily aggregated into one single state in HMMs. Transitions of 100% probability can be aggregated into one single state of activity.

- HMMs can be annotated with a variety of attributes, such as (risk and transition) probabilities, time duration, variances, etc.

- HMMs offer better possibilities to match the models obtained from process discovery with the training/learning datasets. In particular, parallel activities are filtered out in HMMs.
In this section the standard HMM MATLAB toolbox developed by Kevin Murphy was used (Murphy 1998). The patient data were transformed into sequences, and the Expectation Maximization (EM, also known as Baum-Welch) algorithm was used to produce the results for this section. This algorithm combines both forward and backward learning techniques for training an HMM as a process model. The input data were organized according to the Event – Object – Actor standard for process mining input. In this case the input data were obtained from standard clinical patient reporting datasets, compatible with the Healthcare Level 7 record standard.

The only large scale commercial toolset for process discovery (including not only the process analytics, but also the automatic non-invasive gathering of input data) is provided by OpenConnect in its Comprehend product family.

5.3.3. Data discovery with Formal Concept Analysis

Formal Concept Analysis (Ganter 1999) is a data analysis technique that supports the user in analyzing the data and discovering unknown dependencies between data elements. In particular, the visualization capabilities are of interest to the domain expert who wants to explore the information available, but at the same time has not much experience in mathematics or computer science. The details of FCA theory and how we used it for KDD can be found in chapter 3 (Poelmans et al. 2009).

Traditional FCA is mainly using data attributes for concept analysis. In this section the process activities (events) are used as the attributes, whereas the patients are used as the objects in the cross-table that is used as input for FCA. In analogy with (Poelmans et al. 2009) where coherent data attributes were clustered to reduce the computational complexity of FCA, coherent events have been clustered in this study.

5.3.4. Dataset

Our dataset consists of 148 breast cancer patients that were hospitalized during the period from January 2008 till June 2008. They all followed the care trajectory determined by the clinical pathway Primary Operable Breast Cancer (POBC), which structures one of the most complex care processes in the hospital. The treatment of breast cancer consists of 4 phases in which 34 doctors, 52 nurses and 14 paramedics are involved. Fig. 5.11 contains a high-level summary of the breast cancer care process. Before the patient is hospitalized, she ambulatory receives a number of pre-operative investigative tests. During the surgery support phase she is prepared for the surgery she will receive, while being in the hospital. After surgery she remains hospitalized for a couple of days until she can safely go home. The post-operative activities are also performed in an ambulatory fashion. Every
activity or treatment step performed to a patient is logged in a database and in the dataset we included all the activities performed during the surgery support phase to each of these patients.

![Breast cancer care process](image)

**Fig. 5.11. Breast cancer care process**

Each activity has a unique identifier and we have 469 identifiers in total for the clinical path POBC. Using the timestamps assigned to the performed activities, we turned the data for each patient into a sequence of events. These sequences of events were used as input for the process discovery methods. We also clustered activities with a similar semantical meaning to reduce the complexity of the lattices and process models.

### 5.3.5. Analysis and results

One of the most important tasks of the care process manager is to gain insight into what’s happening on the working floor. The goal was to develop an approach that optimally supports this manager’s role. The synergy of process and data discovery techniques in healthcare we propose, has some major advantages over the traditional way of working:

- Significantly reduces the workload for the care process manager who has to monitor over 42 care processes.
- Many unknown data dependencies are revealed that stay hidden for traditional statistical analysis techniques, which typically only look at one or two aspects of the process simultaneously.
- Provides a structured method for finding knowledge gaps, outliers, quality of care issues, process anomalies and inefficiencies.
- Much more information is provided to the process manager, much more quickly. This allows for better analysis and real-time anticipation on potential problems, whereas in the past, this could be
done only after a yearly, very time-consuming and labor-intensive retrospective data analysis.

- The method allows the user to zoom in on different aspects of the provided care.

The process models allow for the extraction and visualization of the most frequent standard care pathway. While analyzing these models, we observed many anomalies and process exceptions that were hard to explain. Therefore, we used FCA to zoom in on and analyze these observations in detail.

5.3.5.1. Quality of care analysis

Our initial process model was built from the full dataset with 148 patients and 469 activity codes. We observed a relatively linear process for the group of patients with a length of stay in the hospital less than 10 days. However, there were 12 patients for which the process model was very complex. They all had in common that their length of stay in the hospital was longer than 9 days. Fig. 5.12 contains screenshots from the output produced by the Comprehend toolset. The upper part displays the obtained process map on the set of patients with a length of stay lower than or equal to 10 days in the hospital and the lower part displays the obtained map for the patients with a length of stay lower than 10 days.

![Comprehend process map for patients with a length of stay smaller than 10 days (upper part) and process map for patients with a length of stay larger or equal than 10 days (lower part)](image)

Fig. 5.12. Comprehend process map for patients with a length of stay smaller than 10 days (upper part) and process map for patients with a length of stay larger or equal than 10 days (lower part)
We built an FCA lattice to explore their characteristics. This lattice in Fig. 5.13 gave us some first interesting insights in the problem. We will try to summarize the most important ones.

- One of our clinical indicators is the pain score which tells us at which days the pain experienced by patients reaches its highest level. We always saw peaks on 1 and day 4 of hospitalization however until now we had no idea why. The lattice gave us an interesting suggestion that this might be due to an overlooked connection between removal of the wound drains and insufficient pain medication. We were able to find that wound drains is probably the most contributing factor to an increased pain score experience by patients and that pain medication should be administered before removing the drains (= improving quality of care).

Fig. 5.13. Lattice containing 12 patients with length of stay larger or equal to 10 days
5.3. COMBINING BUSINESS PROCESS AND DATA DISCOVERY TECHNIQUES

- We were able to find a quality problem in the care provided to these 12 patients. For 1 patient the history record (containing amongst others clinical, psychosocial information) was not consulted prior to the start of treatment. This may result in an inappropriate nursing care thereby potentially neglecting physical and psychosocial patient needs.

Probably one of the main reasons of the increased length of stay we found to be the following: neurological/psychiatric problems, wound infection, subsequent bleeding. This makes the care process more complex and result in more investigative tests. Since these additional morbidities are probably one of the root causes for this increased length of stay, there treatment should be anticipated on and optimalized during the preoperative phase.

5.3.5.2. Process variations

There are five types of breast cancer surgery: mastectomy, breast conserving surgery, lymph node removal and the combination of either mastectomy or breast conserving surgery with lymph node removal. For each of these surgery types, we extracted the corresponding patients in the dataset and constructed a process model and an FCA lattice for in-depth analysis of the characteristics of these groups.

Mastectomy surgery consists of completely removing the breast and during breast conserving surgery only the tumor is removed. The process models showed that the complexity of the care process is much larger for the mastectomy patients. Since mastectomy is a more complex surgery type, we expected that the FCA lattices would also be more complex than for breast conserving surgery. Surprisingly we found out that this was not true. The complexity of the lattice was larger for the breast conserving surgery patients and we found that this was due to the less uniform structure of this care process, in which for many patients some essential care interventions were missing. Fig. 5.14 contains the interventions performed to the 60 patients receiving breast-conserving surgery with lymph node removal. The lattice shows that 3 of these patients did not receive a consultation from the social support service. 15 patients did not have an appointment with a physiotherapist and did not receive revalidation therapy. 1 patient did not receive a pre-operative preparation and 2 patients were missing emotional support before and after surgery.
The originally developed breast-conserving surgery care pathway was written for a certain length of stay for the patients in the hospital. This length of stay was significantly reduced over the past years without modifying the care process model. As a consequence, we found it became impossible to execute the prescribed process model in practice and patients are receiving suboptimal care. The activities performed to the patients should be reorganized and a new care pathway, taking into account this time restriction, should be developed.
Fig. 5.15. Lattice containing 37 patients receiving mastectomy surgery with lymph node removal

Fig. 5.15 shows the lattice for the 37 patients receiving mastectomy surgery with lymph node removal, which has a much less complex structure than the lattice for the breast conserving surgery with lymph node removal. For the mastectomy patients, we found that most patients received all key interventions prescribed in the clinical pathway. Only for two patients there was a quality of care issue, namely 1 patient did not receive emotional support and 1 patient did not receive a breast prosthesis. These shortcomings in the provided care however may have serious consequences for her psychological well-being.
5.3.5.3. Workforce intelligence

We also made a lattice for each type of surgery in which we used as attributes the names of the surgeons and the length of stay of the patients in the hospital. We calculated the average length of stay of the patients and looked at how many patients stayed longer, equal or shorter than this average time of stay. Fig. 5.16 contains the lattice for the 60 patients receiving breast conserving surgery with attributes length of stay and doctor performing the operation.

![Fig. 5.16. lattice for 60 patients receiving breast conserving surgery](image)

We saw for the breast conserving surgery with lymph node removal that 25 patients with a length of stay smaller than 4 days were treated by "surgeon 9", whereas almost all patients treated by the other doctors had a longer length of stay.

We extracted these subsets of patients and constructed a process model for the groups of patients with a length of stay smaller than 4 days, equal to four days and larger than 4 days. This way, we were able to extract some best practices that could be used to improve the care provided to all patients. Fig. 5.17 contains the HMM process model extracted from the datasets with the 10 breast-conserving surgery patients with a length of stay in the hospital of 4 days (the average length of stay). This process model was chosen because of its simplicity in comparison with the other models and since it most closely resembles the standard care process as perceived by the domain experts.

Table 5.5 contains some of the complexity measures for these process variations. For each surgery type and length of stay, the number of patients, the average number of activities and the number of unique activities performed to these patients is given. For visualizing the process maps, we laid a cutoff point at 5%, i.e. all transitions with a lower probability of...
occurrence were removed from the process representation. The table also contains the number of remaining unique activities and the number of connections after filtering. The structural complexity measure after filtering is the sum of these two measures.

**Fig. 5.17.** Process model for 10 breast-conserving surgery patients with length of stay of 4 days
Table 5.5. Complexity measures for the two process variations with the largest number of patients.

<table>
<thead>
<tr>
<th>SURGERY</th>
<th>( \text{\textbackslash LOS} )</th>
<th>LOW</th>
<th>AVG</th>
<th>HIGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast Conserving Therapy with Lymph Node Removal</td>
<td>Length of stay</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(&lt; 4 \text{ days} = 4 \text{ days} &gt; 4 \text{ days})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td># patients</td>
<td>32</td>
<td>10</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Avg. # activities</td>
<td>97</td>
<td>146</td>
<td>184</td>
</tr>
<tr>
<td></td>
<td># unique activities</td>
<td>32</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td># unique act filtered</td>
<td>24</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td># connections filtered</td>
<td>98</td>
<td>80</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Struct. Complex. filtered</td>
<td>122</td>
<td>102</td>
<td>114</td>
</tr>
<tr>
<td>Mastectomy with Lymph Node Removal</td>
<td>Length of stay</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(&lt; 7 \text{ days} = 7 \text{ days} &gt; 7 \text{ days})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td># patients</td>
<td>17</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Avg. # activities</td>
<td>187</td>
<td>206</td>
<td>268</td>
</tr>
<tr>
<td></td>
<td># unique activities</td>
<td>27</td>
<td>21</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td># unique act filtered</td>
<td>19</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td># connections filtered</td>
<td>83</td>
<td>78</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Struct. Complex. filtered</td>
<td>102</td>
<td>98</td>
<td>124</td>
</tr>
</tbody>
</table>

5.3.5.4. Data entrance quality issues

Using the process models, we also found some data entrance quality problems. For some patients, activities were registered after the day of discharge. We found that this was due to an error in the computer program combined with sloppy data entry by the nursing staff. We also found many semantically identical activities that had different activity numbers.

When we analyzed the process models, we found that some of the events typically were not ordered in the sequence that they are performed in real life. In other words, the timing of the events as can be found in the data does not always correspond to the timing at the real-life working floor. We found this is due to an error in the computer system which sometimes imposes a certain sequence of events and does not allow for a correct registration of activities. There is a discrepancy between this built-in top-down developed model and the reality. This discrepancy is probably due to the insufficient insight into the reality of the working floor when the system was developed. The anomalies found during this process mining exercise will be used as input for the development of the new IT systems.

5.3.6. Discussion and conclusions

Neither process nor data discovery techniques alone are sufficient for discovering knowledge gaps in particular domains such as healthcare. In this section, we showed that the combination of both gives significant synergistic
results. Whereas FCA does not provide easy to use process representations, it has proven to be very useful for process analysis, i.e. to analyze anomalies and exceptions in detail.

Initially, we thought FCA would only be useful for post-factum analysis of the results obtained through process discovery, but in this case we also found that FCA can play a significant role in the discovery process itself. In particular, concept lattices were used to improve the detection and understanding of outliers in the data. These exceptions are not noise, but are the activities performed to human beings, so every exception counts and must be understood. Concept lattices were also used to reduce the workspace, to cluster related events together in an objective manner.

Using this combination of techniques, we exposed multiple quality of care issues. We gained a better understanding of the process variations and better understood where we should take action to improve our healthcare processes. The impact of co-morbidities of patients on the overall care process was found to be of importance and offers some opportunities for improving quality and efficiency of care. Further, reducing the length of stay of breast-conserving therapy patients was discovered to be the root cause for a suboptimal care, missing some key interventions, provided to patients. Finally, we found the length of stay for patients receiving breast-conserving surgery was significantly different for different surgeons. This situation may be improved by uniformization of discharge criteria.

Avenues for future research include the use of supervised clustering, mainly to obtain normalized process models, in which many-to-many transitions are eliminated (as argued in Peters et al. 2009). The normalized clusters will give the best views on process variations. Again, a posterior data discovery (FCA) can be used to understand the meaning of the different clusters.
CHAPTER 6

Conclusion

In this chapter we give some concluding remarks on the research performed in this thesis and we describe some interesting avenues for future research.

6.1. Thesis conclusion

In this thesis, we investigated the possibilities of using FCA in information engineering. The main theme was applying FCA for concept discovery and concept visualization in various domains such as scientific literature review, text mining, software engineering, temporal data mining and process mining. Each case study revealed the benefit of FCA as a human-centered instrument for data analysis that made domains previously inaccessible to analysts because of the overload of information, available for human reasoning and knowledge creation.

In chapter 2 we gave an extensive overview of the literature on FCA. This was the first systematic literature review on this already mature research field covering over 700 papers. Using Lucene and a thesaurus containing terms and phrases referring to research topics in the FCA community we indexed these papers. After performing a clustering of terms we built multiple FCA lattices and analysed these in detail. Data mining and knowledge discovery, information retrieval and ontology engineering were some of the most prominent research topics. Also, multiple authors expanded FCA with fuzzy theory or rough set theory and for temporal or triadic data. Mathematical research and algorithmic innovations comprise the final part of the literature review. By using FCA to characterize the literature on concept analysis, we not only gained insight into the main research topics but also discovered multiple gaps in the literature which we tried to fill in this thesis. In chapter 2, FCA was found to be an interesting meta technique for exploring large amounts of text which was further investigated in Chapter 3.

In our study on domestic violence we used FCA for exploring and refining the underlying concepts of police data. Traditional machine learning
and classification techniques build a model on the data without challenging the underlying concepts of the domain. In chapter 3 we proposed FCA as a human-centered KDD instrument, that truly engages the analyst in the knowledge acquisition process. Terms are clustered in term clusters and the concept lattice shows the relationships between these term clusters and the police reports. We combined FCA with Emergent Self Organizing Maps to discover emergent structures in the high-dimensional data space. The KDD process was framed in C-K theory and interpreted as multiple successive iterations through the design square. There was a continuous process of iterating back and forth between analyzing the FCA and ESOM artifacts, selecting reports for in-depth manual inspection, gaining new knowledge and beginning a new knowledge creation cycle. Using FCA and ESOM we analyzed a large set of unstructured text reports from 2007 indicating incidents in the Amsterdam-Amstelland police region. We not only uncovered the true nature of domestic violence but also found multiple anomalies, faulty case labelings, confusing situations for police officers, niche cases, concept gaps, etc. This resulted in a refinement of the domestic violence definition, improvement of police training, reopening and relabeling filed reports and an automated domestic violence detection system. This system is based on 37 classification rules that were discovered during the successive knowledge discovery iterations. Each of these rules consist of a combination of early warning indicators which flag the nature of the case. If a domestic violence incident is detected, a red flag is raised. 75% of the incoming cases can be labeled correctly with this system.

In chapter 4 we applied FCA in software engineering and more in particular in the software requirements analysis phase. First, we showed how the concepts from MERODE such as existence dependency graph, object-event table and the well-formedness rules can be mapped to FCA theory. The result of applying FCA to a well-formed object-event table is a lattice that is isomorphic to the EDG from MERODE. This lattice can be used for detecting anomalies and concept gaps in the conceptual model. Major results are that the well-formedness rules imposed in MERODE as a meta frame are an inherent part of the FCA lattice construction algorithm and that FCA can a used to mathematically underpin the construction of conceptual models. We then proposed FCA as a human-centered instrument to obtain normalized class models. We looked at the requirements engineering process as successive iterations through the C-K design square. The FCA artefacts are used for formal verification, anomaly detection, validation and modification of use cases and the conceptual model. The method's usefulness was shown on multiple case studies. Finally, we extended traditional FCA with a third dimension to deal with inheritance. Traditional FCA imposes only one partial ordering relation on objects whereas we need two ordering relationships in software engineering namely for UML associations or existence dependency relations and inheritance hierarchies in a model. The properties of this merged lattice were investigated and applied to a number of software engineering case studies.
In Chapter 5, we analyzed FCA’s applicability to data with an inherent time dimension. We twice made a combination of FCA and Temporal Concept Analysis. In our first case study we used FCA to distill potential human trafficking suspects from observational police reports and for suspicious persons a detailed profile was constructed with TCA. These profiles aided police officers in deciding which subjects should be monitored or further investigated. In a next step, we analyzed the social network of a suspicious person with TCA and used it to gain insight into the network’s structure. We then repeated this exercise for terrorism subjects and based our text analysis method on the early warning indicators of the four phase model developed by the KLPD. The results were the discovery of several persons who were radicalizing or reached a critical radicalization phase but were not known by the police of Amsterdam-Amstelland. These subjects are currently being monitored by police authorities. For analyzing care processes, TCA’s representation turned out to be too complex. Therefore we chose to use Hidden Markov Models for distilling process models from patient treatment data. FCA was used to expose quality of care issues and identifying characteristics of best practice process variations. These insights are currently being used to improve the quality of the provided care.

6.2. Future work

Terrorist threat assessment.

Many general reports related to terrorist activity have not been labelled as such by police officers. We want to find all relevant reports since each one may contain crucial information. Using an incremental learning algorithm we plan to build a classifier to automatically label reports. Since there are only few reports labelled as terrorism-related we first construct this model on a small partition of the dataset. The assigned labels are manually verified and a new model is built on a larger training set. The same procedure is repeated until a scalable and operationally useable classification model is obtained. We also intend to use TopicView as a means to validate some of the relationships between police reports and indicators. TopicView will amongst others be used to scan general police reports and incoming email messages on terrorist activity and will offer interesting relationships to the analyst for further investigation. The analyst can confirm or decline these associations and build an FCA model on these manually validated data.

Human trafficking.

One of the first steps in future research will be expanding and refining the set
of terms related to indicators. Using a combination of FCA, ESOM and Natural Language Processing techniques we intend to build a thesaurus capturing the essential concepts underlying the domain, we will complement our research and current analyses with traditional Social Network Analysis and use FCA to characterize the found groups of suspects. The human trafficking team will provide us with a labelled dataset of significant size. After a testing phase in which the practical usefulness of our method is validated, we will embed our analysis approach in daily operational policing practice.

**Domestic violence.**

Till date, we only performed analyses on reports containing a statement made by the victim to the police. Recently, the criminal code of the Netherlands changed and now allows for proactive searching of suspects. In the future, our analyses will mainly focus on general reports describing observations made by officers. We will also develop a risk assessment model for estimating the probability that a person will become a repeat offender. This model will be based on early warning indicators, some of them were already discovered during the KDD exercise.

**Clinical pathways.**

In the Sint-Augustinus hospital in Antwerp, 42 clinical pathways are used to structure care processes. These pathways should be monitored, analysed and improved on a continuous basis. We plan to build a toolset based on FCA and Hidden Markov Models to support the care process manager in this task. Furthermore, we will investigate the benefits of using normalised process models in which there are no more many-to-many transitions resulting in less ambiguities. Finally, we aim at distilling patient profiles from recorded data. These profiles will be used to predict amongst others increased length of stay risk, risk of complications during treatment, etc. and adjust care appropriately.

**Financial Crime Analysis.**

Money laundering and financial crime in general are serious problems for the Amsterdam-Amstelland police. Large amounts of transactions, money flows that are only partially visible to law enforcement authorities, etc. made it difficult to detect suspicious behavior. The domain is characterized by vast amounts of data which are rapidly changing on a continuous basis. We will
investigate the possibilities of Emergent Self Organizing Maps, process
discovery and neural network pattern recognition techniques to gain insight
in these data.

**Intelligence Led Policing and Concept Discovery Toolset CORDIET.**

In cooperation with the Amsterdam-Amstelland police and Moscow Higher
School of Economics we will develop and implement a toolset based on FCA
and C-K theory for analyzing police data. The C-K design square will be at
the core of this toolset. Each of the activities belonging to one of the 4 arrows
will be implemented and provided to the user. The tool will consist of a main
window and four tales corresponding to each of the four arrows. Functionality will include text mining support such as indexing police reports
using Lucene with a thesaurus, FCA lattice visualization and making the
lattices browsable, connectors to database systems, etc.

**Service Oriented Architecture.**

We will apply FCA to a dataset of the Ministry of Finance of the
Netherlands. Services will amongst others be mapped against service
components. The result will be a lattice of service and service component
clusters. New services, opportunities for service reuse, etc. may be identified.

**K-Formal Concept Analysis.**

The algebraic expressions of Finite State Machines of objects in a conceptual
ruodel characterize their behavior. These behavioral expressions can be
considered as elements of a semiring. By applying a variant of $K$-FCA we
obtain a partial order lattice of finite state machines. FSMs lower in the
lattice have less degrees of freedom than FSMs higher in the lattice. This
partial order makes consistency analysis an easier task to perform.
APPENDIX A

Literature survey thesaurus

Appendix A of this thesis contains a small excerpt of the thesaurus used during the literature study. The first column of the following table contains the name of the term clusters, the second column contains the associated search terms.

<table>
<thead>
<tr>
<th>Term cluster</th>
<th>Search terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge discovery</td>
<td>CKDD, concept discovery, data exploration, data mining; datamining, DM, exploratory data analysis, exploratory processes, exploring data, exploring information, exploring knowledge, information exploration, KDD, Knowledge discovery, knowledge exploration, knowledge extraction, machine learning, mining</td>
</tr>
<tr>
<td>Association rule mining</td>
<td>ARM, association rule mining, association rules; association rule, CHARM, classification rules, closed itemset; closed itemsets, CLOSET, extraction of rules, frequent itemsets; frequent itemset; frequent closed itemset; frequent closed itemsets, generators of closed itemset; generators of closed itemsets, mingen, PRINCE, rule extraction, TITANIC</td>
</tr>
<tr>
<td>Scalability</td>
<td>alpha lattices, exploration of a large number of objects, handling large formal context, huge database, Iceberg concept lattices, iceberg lattices; iceberg lattice, iceberg sets, large contexts, large data, large database, large databases, pruning strategy; pruning strategies, reduce less useful concepts, reduce the size of large concept lattices, scalability,</td>
</tr>
<tr>
<td><strong>APPENDIX B</strong></td>
<td>240</td>
</tr>
<tr>
<td>----------------</td>
<td>-----</td>
</tr>
</tbody>
</table>

| **scalable** |  |
| **Ontology** | ontological, ontology construction, ontology, ontologies |

| **Information retrieval** | concept retrieval, conceptual retrieval, CREDO;credo, data retrieval, document retrieval, information retrieval, instance retrieval, IR, knowledge retrieval, lookup method, queries, query, restructuring help system, retrieval, SNOMED;snomed, text retrieval, CEM, Conceptual Email Manager, Mail-Sleuth, Mail-Strainer, manage email |

| **Web mining** | digital library, internet, mining the web, query web documents, search results, virtual surfing trials, web data mining;web datamining, web document management, web documents, web information, web mining;webmining, web search, web user profiles, web-based mail browser, weblogs;web logs, web-pages;web pages, world wide web |

| **Fuzzy FCA** | fuzzifications, fuzzy, uncertain information |

| **Software mining** | AOSD, aspect identification, aspect mining;aspect-mining, aspect oriented;aspect-oriented, class hierarchies, crosscutting concerns, design of class hierarchies, modularization, modularity, Object-Oriented concept analysis, reengineering class hierarchies, reversed engineering, software, software evolution, source code, use case |

| **Rough set theory** | RFCA, rough set, rough sets |
APPENDIX B

Domestic violence thesaurus

Appendix B of this thesis contains a small excerpt of the thesaurus used during the domestic violence case study. The first column of the following table contains the name of the term clusters, the second column contains the associated search terms.

<table>
<thead>
<tr>
<th>Personen en betrokkenen</th>
<th>vader (concept); mijn vader; jouw vader; je vader; haar vader; zijn vader; vader van kind moeder (concept); mijn moeder; jouw moeder; je moeder; haar moeder; zijn moeder; moeder van kind mama (concept); mijn mama; jouw mama; je mama; haar mama; zijn mama papa (concept); mijn papa; jouw papa; je papa; haar papa; zijn papa ouders (concept); mijn ouders; jouw ouders; je ouders; haar ouders; zijn ouders; ouders van kind broer (concept); mijn broer; jouw broer; je broer; haar broer; zijn broer broer (concept); mijn broertje; jouw broertje; je broertje; haar broertje; zijn broertje zus (concept); mijn zus; jouw zus; je zus; haar zus; zijn zus zus (concept); mijn zusje; jouw zusje; je zusje; haar zusje; zijn zusje</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geweldsvormen</td>
<td>slaan, vuistslag geven, schoppen, bedreigen; bedreiging, neerslaan; neer slaan, neersteken; neer steken, stompen, knijpen, steken, vluchten, terroriseren; terrorismering; geterroriseerd; geterroriseert, mishandelen; mishandeling; eenvoudige mishandeling; zware mishandeling, stalken; stalking; stalker, krassen; krabben, bijten, duwen, worstelen; worsteling, verkrachten; verkrachting, lastig</td>
</tr>
<tr>
<td>Locaties in huis</td>
<td>thuis, in huis, woning, keuken, badkamer, zolder, woonkamer, gang;hal, slaapkamer, studeerkamer, kinderkamer, toilet;plee;wc, kelder, schuur;schuurtje, berging, garage, logeerkamer, serre, souterrain, tuin</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Mishandeld door persoon uit huiselijke kring</td>
<td>ik mishandeld door (concept);mishandeld door mijn vader;mishandeld door mijn moeder;mishandeld door mijn mama;mishandeld door mijn papa;mishandeld door mijn ouders;mishandeld door mijn broer;mishandeld door mijn zus;mishandeld door mijn zuster;mishandeld door mijn kind;mishandeld door mijn man;mishandeld door mijn echtgenoot;mishandeld door mijn vrouw;mishandeld door mijn echtgenote;mishandeld door mijn stiefvader;mishandeld door mijn stiefmoeder;mishandeld door mijn schoonvader;mishandeld door mijn schoonmoeder;mishandeld door mijn dochter;mishandeld door mijn zoon;mishandeld door mijn schoonzoon;mishandeld door mijn schoonzoen;…</td>
</tr>
<tr>
<td>Bedreiging door persoon buiten huiselijke kring</td>
<td>bedreiging door werknemer;bedreiging door ex-werknemer;bedreiging door ex-werknemer;bedreiging door collega;bedreiging door ex-collega;bedreiging door ex-collega;bedreiging door ex-collega;bedreiging door ex-werknemer;bedreiging door baas;bedreiging door voorman;bedreiging door supervisor;bedreiging door buren;bedreiging door buurman;bedreiging door buurvrouw;bedreiging door buurmeisje;bedreiging door buurjongen;bedreiging door bazin;bedreiging door hospita;bedreiging door concierge;bedreiging door huisbaas;bedreiging door portier;bedreiging door uitsmijter;bedreiging door werkneemster;bedreiging door ex-werkneemster;bedreiging door ex-werkneemster;bedreiging door ex-werkneemster;…</td>
</tr>
</tbody>
</table>
APPENDIX C

Domestic violence review reports

Appendix C of this thesis contains a small screenshot of the review reports we received from the data quality management team during the domestic violence case study. Their feedback was used to improve our system.
List of figures

FIG. 2.1. LINE DIAGRAM CORRESPONDING TO THE CONTEXT FROM TABLE 2.1 ........................................................................................................27
FIG. 2.2. LATTICE CONTAINING 702 PAPERS ON FCA ........................................30
FIG. 2.3. LATTICE CONTAINING 140 PAPERS ON USING FCA IN KDD .................................................................32
FIG. 2.4. LATTICE CONTAINING 103 PAPERS ON USING FCA IN IR .................................................................41
FIG. 2.5. LATTICE CONTAINING 32 PAPERS ON FCA AND SCALABILITY .................................................................47
FIG. 2.6. LATTICE CONTAINING 93 PAPERS ON FCA AND ONTOLOGIES .................................................................51
FIG. 2.7. LATTICE DESCRIBING SOME OF THE EXTENSIONS OF TRADITIONAL FCA THEORY .................................................................56
FIG. 2.8. LATTICE DESCRIBING THE 75 PAPERS ON FCA AND FUZZY THEORY .................................................................57
FIG. 2.9. LATTICE DESCRIBING THE 46 PAPERS ON FCA AND ROUGH SET THEORY .................................................................64
FIG. 2.10. LATTICE DESCRIBING THE FCA PAPERS ON MATHEMATICAL RESEARCH AND ALGORITHMIC INNOVATIONS .................................................................71
FIG. 3.1. CURRENT DOMESTIC VIOLENCE REPORTING PROCEDURE ..................................................................................79
FIG. 3.2. LINE DIAGRAM CORRESPONDING TO THE CONTEXT FROM TABLE 1 ..................................................................................82
FIG. 3.3. DESIGN SQUARE (ADAPTED FROM (HATCHUEL 2003)) .................................................................86
FIG. 3.4. KNOWLEDGE DISCOVERY PROCESS .................................................................................................88
FIG. 3.5. EXAMPLE POLICE REPORT .................................................................................................90
FIG. 3.6. CLASSIFICATION PERFORMANCE FOR DIFFERENT SUBSETS OF THE RANKED LIST OF FEATURES PRODUCED BY THE MRMR ALGORITHM .................................................................................................94
**LIST OF FIGURES**

246

<table>
<thead>
<tr>
<th>FIG.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7</td>
<td>INITIAL LATTICE BASED ON THE POLICE REPORTS FROM 2007</td>
<td>98</td>
</tr>
<tr>
<td>3.8</td>
<td>TOROIDAL ESOM MAP</td>
<td>99</td>
</tr>
<tr>
<td>3.9</td>
<td>FIRST REFINED LATTICE BASED ON THE POLICE REPORTS FROM 2007</td>
<td>100</td>
</tr>
<tr>
<td>3.10</td>
<td>SECOND REFINED LATTICE BASED ON THE POLICE REPORTS FROM 2007</td>
<td>101</td>
</tr>
<tr>
<td>3.11</td>
<td>THIRD REFINED LATTICE BASED ON THE POLICE REPORTS FROM 2007</td>
<td>102</td>
</tr>
<tr>
<td>3.12</td>
<td>FOURTH REFINED LATTICE BASED ON THE POLICE REPORTS FROM 2007</td>
<td>103</td>
</tr>
<tr>
<td>3.13</td>
<td>ESOM RISK ANALYSIS MAP FOR THE YEAR 2007 AND REMAINING CASES MADE VISIBLE</td>
<td>114</td>
</tr>
<tr>
<td>3.14</td>
<td>MDS MAP TRAINED ON THE CATEGORICAL DATASET</td>
<td>117</td>
</tr>
<tr>
<td>3.15</td>
<td>CLASSIFICATION PERFORMANCE</td>
<td>118</td>
</tr>
<tr>
<td>3.16</td>
<td>TOROID ESOM MAP TRAINED ON THE CATEGORICAL DATASET. SEE TEXT FOR AN</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>EXPLANATION OF THE ARROWS AND CIRCLES.</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>LINE DIAGRAM CORRESPONDING TO THE CONTEXT FROM TABLE 4.1</td>
<td>130</td>
</tr>
<tr>
<td>4.2</td>
<td>LATTICE CORRESPONDING TO TABLE 2</td>
<td>134</td>
</tr>
<tr>
<td>4.3</td>
<td>POTENTIAL MISSING OBJECT TYPE</td>
<td>139</td>
</tr>
<tr>
<td>4.4</td>
<td>CORRECT OBJECT-EVENT LATTICE</td>
<td>140</td>
</tr>
<tr>
<td>4.5</td>
<td>EXISTENCE DEPENDENCY LATTICE</td>
<td>143</td>
</tr>
<tr>
<td>4.6</td>
<td>INHERITANCE LATTICE</td>
<td>145</td>
</tr>
<tr>
<td>4.7</td>
<td>HOW THE EXISTING DEPENDENCY LATTICE AND THE INHERITANCE LATTICE ARE</td>
<td>146</td>
</tr>
<tr>
<td></td>
<td>COMBINED TO FORM ONE EXTENDED LATTICE.</td>
<td></td>
</tr>
<tr>
<td>4.8</td>
<td>PROPERTIES OF THE EXTENDED LATTICE</td>
<td>147</td>
</tr>
<tr>
<td>4.9</td>
<td>DETECTING A VIOLATION OF THE ACCUMULATION RULE WITH THE EXTENDED LATTICE</td>
<td>150</td>
</tr>
<tr>
<td>4.10</td>
<td>DETECTING ANOMALIES WITH THE EXTENDED LATTICE OF A LIBRARY SYSTEM</td>
<td>152</td>
</tr>
</tbody>
</table>
FIG. 4.11. DETECTING ANOMALIES USING THE EXTENDED LATTICE OF A PHARMACY SYSTEM ................................................ 153
FIG. 4.12A CLASS DIAGRAM WITH A REFLEXIVE ASSOCIATION. 158
FIG. 4.12B NORMALIZED CLASS DIAGRAM WITHOUT REFLEXIVE ASSOCIATION. ................................................................. 158
FIG. 4.13A CLASS DIAGRAM WITH MANY-TO-MANY ASSOCIATION. ........................................................................................ 159
FIG. 4.13B NORMALIZED CLASS DIAGRAM WITHOUT MANY-TO-MANY ASSOCIATION. ............................................................ 160
FIG. 4.14A CLASS DIAGRAM WITH OPTIONAL-TO-OPTIONAL ASSOCIATION ................................................................................. 160
FIG. 4.14B NORMALIZED CLASS DIAGRAM WITHOUT OPTIONAL-TO-OPTIONAL ASSOCIATION. ......................................................... 161
FIG. 4.15A CLASS DIAGRAM WITH ASSOCIATION BETWEEN MORE THAN 2 PARTNERS AND UML DIAMOND NOTATION ........... 162
FIG. 4.15B NORMALIZED CLASS DIAGRAM WITHOUT DIAMOND NOTATION. .................................................................................. 162
FIG. 4.16A CLASS DIAGRAM WITH COMPOSITION RELATIONSHIP. ................................................................................................. 163
FIG. 4.16B NORMALIZED CLASS DIAGRAM. ............................................. 163
FIG. 4.17A CLASS DIAGRAM WITH AGGREGATION RELATIONSHIP. ................................................................................................. 164
FIG. 4.17B NORMALIZED CLASS DIAGRAM. ............................................. 164
FIG. 4.18A CLASS DIAGRAM WITHOUT NORMALIZATION. .......... 165
FIG. 4.18B NORMALIZED CLASS DIAGRAM. ............................................. 165
FIG. 4.19 DESIGN SQUARE (ADAPTED FROM HATCHUEL ET AL. 2003)................................................................................................. 167
FIG. 4.20. ITERATIVE C-K REQUIREMENTS ENGINEERING PROCESS ................................................................. 169
FIG. 4.21A. COLLABORATION DIAGRAM PLACE ORDER \( ? \) (ADAPTED FROM LIANG 2003) ................................................................. 175
FIG. 4.21B. COLLABORATION DIAGRAM (ADAPTED FROM LIANG 2003)................................................................................................. 175
FIG. 4.21C. COLLABORATION DIAGRAM (ADAPTED FROM LIANG 2003)................................................................................................. 175
FIG. 4.21D. COLLABORATION DIAGRAM (ADAPTED FROM LIANG 2003)................................................................................................. 175
FIG. 4.21E. COLLABORATION DIAGRAM (ADAPTED FROM LIANG 2003) ................................................................. 176
FIG. 4.21F. CLASS DIAGRAM (ADAPTED FROM LIANG 2003) ................. 176
FIG. 4.22. ENTITY – USE CASE INTERACTION MATRIX ...................... 177
FIG. 4.23. USE CASE – OPERATIONS INTERACTION LATTICE .............. 178
FIG. 4.24. ENTITY – OPERATIONS INTERACTION MATRIX .............. 179
FIG. 4.25 INCORRECT HOTEL ADMINISTRATION MODEL AND ENTITY-USE CASE INTERACTION LATTICE BASED ON ORIGINAL INCORRECT MATRIX ............................................................................. 181
FIG. 4.26 CORRECT HOTEL ADMINISTRATION LATTICE AND UML MODEL WHICH IS ISOMORPHIC TO THIS LATTICE ................................................................. 182
FIG. 4.27A ORDERING SYSTEM MODEL WITH MISSING ENTITY 183
FIG. 4.27B FCA LATTICE WITH MISSING ENTITY AND CORRECT ORDERING SYSTEM MODEL WITH CONTRACT ENTITY ADDED AFTER NORMALIZATION ............................................................................. 184
FIG. 4.28 INITIAL UML MODEL AND FCA LATTICE FOR INITIAL MODEL OF ELEVATOR REPAIR SYSTEM ................................................................. 185
FIG. 4.29A NORMALIZED UML MODEL ................................................ 187
FIG. 4.29B FCA LATTICE FOR NORMALIZED MODEL .................... 188
FIG. 4.30 WEB SHOP CASE ............................................................... 190
FIG. 4.31 FCA LATTICE BASED ON CORRECT MATRIX .................. 190
FIG. 4.32 TERMINATION USE CASE OMISSION ............................... 191
FIG. 4.33 USE CASE PARTICIPATION ERROR ................................... 192
FIG. 4.34 CONTRACT ENTITY OMISSION .......................................... 193
FIG. 5.1. EXAMPLE POLICE REPORT .................................................... 198
FIG. 5.2. FCA LATTICE USED FOR DETECTING POTENTIAL HUMAN TRAFFICKING SUSPECTS ............................................................................. 204
FIG. 5.3. TCA LATTICE PROFILE OF A HUMAN TRAFFICKING SUSPECT ............................................................................. 206
FIG. 5.4. TCA LATTICE USED FOR PROFILING A HUMAN TRAFFICKING SUSPECT ............................................................................. 206
FIG. 5.5. TCA LATTICE USED FOR GAINING INSIGHT IN SOCIAL NETWORK OF HUMAN TRAFFICKING SUSPECTS ............................. 207
FIG. 5.6 THE FOUR PHASE MODEL OF RADICALISM ...................... 211
FIG. 5.7 EXAMPLE POLICE REPORT .................................................... 213
FIG. 5.8 THE PROCESS MODEL OF EXTRACTING AND PROFILING POTENTIAL JIHADISTS ................................................................. 215
FIG. 5.9 FCA LATTICE USED FOR EXTRACTING POTENTIAL JIHADISTS ................................................................................... 216
FIG. 5.10 TCA LATTICE FOR SUBJECT C ........................................... 218
FIG. 5.11. BREAST CANCER CARE PROCESS .................................... 223
FIG. 5.12. COMPREHEND PROCESS MAP FOR PATIENTS WITH A LENGTH OF STAY SMALLER THAN 10 DAYS (UPPER PART) AND PROCESS MAP FOR PATIENTS WITH A LENGTH OF STAY LARGER OR EQUAL THAN 10 DAYS (LOWER PART) .............................. 224
FIG. 5.13. LATTICE CONTAINING 12 PATIENTS WITH LENGTH OF STAY LARGER OR EQUAL TO 10 DAYS ................................. 225
FIG. 5.14. LATTICE CONTAINING 60 PATIENTS RECEIVING BREAST-CONSERVING SURGERY WITH LYMPH NODE REMOVAL .......................................................... 227
FIG. 5.15. LATTICE CONTAINING 37 PATIENTS RECEIVING MASTECTOMY SURGERY WITH LYMPH NODE REMOVAL ............. 228
FIG. 5.16. LATTICE FOR 60 PATIENTS RECEIVING BREAST CONSERVING SURGERY .............................................................. 229
FIG. 5.17. PROCESS MODEL FOR 10 BREAST-CONSERVING SURGERY PATIENTS WITH LENGTH OF STAY OF 4 DAYS ........ 230


Contemporary Approaches And Emergent Traditions, North-Holland, Amsterdam, Netherlands, pp. 397-403.


Systems Design and Applications.


146. Ganter, B. (2008a) Lattices of Rough Set Abstractions as P-Products. R. Medina et al. (Eds.): ICFCA, LNAI 4933, 199-216.


transfer: obstacles, incentives and improvement agenda.
Requirements Engineering 7, 113-123.

FCA-Based Methods for Mining Gene Expression Data. S. Ferré et
al. (Eds.): ICFCA, LNAI 5548, 251-266. Springer.

IEEE transactions on visualisation and computer graphics. Vol. 8,
No. 1.

Computation of Closed Itemsets and Implication Rule Bases. I.
Stojmenovic et al. (Eds.): ISPA, LNCS 4742, 359-370. Springer.

Algorithm for Lattice Construction. B. Ganter et al. (Eds.): ICFCA,
LNAI 3403, 249-264. Springer.

and formal concept analysis. Fundamenta Informaticae, 27, 169-181.

aanpak. Directie Preventie, Jeugd en Sanctiebeleid van de
Nederlandse justitie.

OWL Ontology based on the Context Family Model. IEEE.

for Mining Contextualized Folksonomy. SAC, March 11-15, Seoul,
Korea.

and Retrieval for Specialised Domains on the Web. Int. J. of human
computer studies, 60(2): 201-241.

Using Conceptual Scales. A. Hoffmann et al. (Eds.): PKAW, LNAI
4303, 132-143. Springer.

Through Hybrid Unsupervised Clustering Techniques. Y. Zhang et
al. (Eds.): APWeb, LNCs 4976, 499-510. Springer.


comparison with monotonic modelling methods. ICSE 2008 Leipzig, Germany. ACM.


268. Matlab Arsenal (2009) http://www.informedia.cs.cmu.edu/yanrong/MATLABArsenal/MATLABArsenalDoc/MATLABArsenal/NeuralNet.html


on Machine learning and Cybernetics, Shanghai, 26-29 August.


356. Rochfeld, A. (1986) Merise, an information system design and development methodology tutorial. Entity-Relationship Approach:
Ten years of Experience in Information Modelling. Proc. 5th Int. Conf. On Entity-Relationship approach, 489-528.


299-312. Springer.

Through Refinement Enlargement in a Formal Context. J. Dix et al.
(Eds.): FoIKS, LNCS 3861, 278-293. Springer.

Proc. IEEE Int. Conf. on Web Intelligence.

IEEE Int. Conf. on e-Technology, e-Commerce and e-Service.

Analysis. A. Hameurlain et al. (Eds.): DEXA. LNCS, vol. 2453.
Springer.

394. Stumme, G. (2002), Formal Concept Analysis on its Way from
Mathematics to Computer Science. Proc. 10th Intl. Conf. on

discovery and processing. Int. J. Human-Computer Studies 59, 287-
325. Elsevier.

et al. (Eds.): ICCS, LNAI 3127, 109-125. Springer.

397. Stumme, G. (2005) A finite state model for on-line analytical
processing in triadic contexts. B. Ganter et al. (Eds.): ICFCA, LNCS
3403, 315–328. Springer.

Iceberg Concept Lattices with TITANIC. Data and knowledge
engineering 42 (2), 189-222.

Wissenverarbeitung – methoden und anwendungen. Springer
Heidelberg.

discovery in databases using Formal Concept Analysis Methods.
PKDD, 450-458.

Oriented Software Reverse Engineering at Requirements Level.
IEEE Int. Conf. on Computer Science and Software Engineering.


Beijing Platform for Action, 14-15 December.


484. Yao, Y.Y. (2004a) Concept Lattices in Rough Set Theory. IEEE.


Doctoral dissertations from the faculty of business and economics

See: http://www.kuleuven.ac.be/doctoraatsverdediging/archief.htm