

CONTENTS

List of Figures	xiii
List of Definitions	xvii
1 Introduction	1
1.1 Background: the dilemma of software legacies	1
1.2 Database reengineering	3
1.3 Problem definition	4
1.4 The approach	7
1.5 Dissertation outline	10
2 Database Reengineering - A Case Study	11
2.1 A legacy product and document information system	11
2.2 Migration target: a distributed marketing information system	11
2.2.1 Functional requirements	12
2.2.2 Technical requirements	13
2.3 Migration strategy	13
2.4 The reengineering process	14
2.4.1 Legacy schema analysis	15
2.4.2 Conceptual schema migration and redesign	24
2.4.3 Implementation of changes and a middleware for data integration	26
2.5 Summary and concluding remarks	30
3 A Theory to Manage Imperfect Knowledge	33
3.1 Requirements on formalisms to manage DBRE knowledge	33
3.1.1 Quantitative representation of uncertainty	35
3.1.2 Representation and indication of contradicting knowledge	36
3.1.3 Reasoning about incomplete knowledge	36
3.1.4 Representation of ignorance	36
3.1.5 Computational tractability	37
3.2 Evaluation of theories	37
3.2.1 Production systems with confidence factors	38
3.2.2 Probabilistic reasoning	40
3.2.3 Credibilistic reasoning	42
3.2.4 Fuzzy reasoning	44
3.2.5 Possibilistic reasoning	49
3.3 Summary and conclusion	52
4 GFRN as a Basis for Legacy Schema Analysis	55
4.1 Supporting human-centered schema analysis processes	55
4.2 Specification of database reengineering knowledge	57

4.2.1	Informal introduction to GFRNs	58
4.2.2	Integration of automatic analysis operations	63
4.2.3	Formal definition	66
4.3	Knowledge inference with GFRN specifications	76
4.3.1	A fuzzy Petri net model for non-monotonic reasoning	77
4.3.2	The inference process	81
4.4	Implementing the Varlet Analyst	99
4.4.1	Architecture	99
4.4.2	User interface	101
4.5	Evaluation	107
4.6	Related work	109
4.7	Summary	111
5	Conceptual Schema Migration and Data Integration	113
5.1	The migration graph model	115
5.1.1	Graph-based representation of logical and conceptual schema	116
5.1.2	The schema mapping graph model	119
5.2	A graphical formalism to implement schema translators	121
5.2.1	Triple graph grammars	122
5.2.2	Mapping variants to class hierarchies	127
5.2.3	Mapping columns to class attributes	132
5.2.4	Mapping inclusion dependencies to relationships	133
5.2.5	Discussion	136
5.3	Conceptual schema redesign	137
5.3.1	Schema redesign transformations	137
5.3.2	An extensible catalog of schema redesign transformations	138
5.3.3	Complex schema redesign transformations	144
5.4	Incremental change propagation	145
5.4.1	The history graph	146
5.4.2	The propagation mechanism	150
5.5	Implementing the Varlet Migrator	157
5.5.1	Architecture	157
5.5.2	User interface	159
5.6	Data integration	164
5.6.1	Generating descriptions for relational and object-oriented schemas	166
5.6.2	Generating object-relational mapping descriptions	167
5.7	Evaluation	176
5.8	Related work	178
5.8.1	Conceptual schema migration and consistency management	178
5.8.2	Data integration	179
5.9	Summary	180
6	Conclusions and Future Perspectives	181
6.1	Major contributions	181
6.2	Transferability of results	182
6.3	Open problems	183
6.4	Future perspectives	184

A Additional Definitions and Specifications	187
A.1 Interpretation of a logical schema	187
A.2 Specification of the migration graph model	188
B A Catalog of Redesign Transformations	195
References	217
Index	233
Abbreviations	237

LIST OF FIGURES

Figure 1.1.	Reengineering process	3
Figure 1.2.	Conceptual schema as a starting point for subsequent DBRE activities	4
Figure 1.3.	CARE tool classification according to the role of human knowledge	6
Figure 1.4.	Proposed DBRE approach	8
Figure 2.1.	Existing Product and Document Information System (PDIS)	12
Figure 2.2.	Planned Marketing Information System (MIS)	13
Figure 2.3.	Gradual migration strategy	14
Figure 2.4.	The planned reengineering process	15
Figure 2.5.	Constraints resulting from the schema catalog	16
Figure 2.6.	Potential constraints indicated by naming heuristics	16
Figure 2.7.	Detail of PDIS	17
Figure 2.8.	Contradicting indicators for key constraint	18
Figure 2.9.	Potential foreign keys indicated by <i>join</i> patterns	18
Figure 2.10.	Result of the structural completion	19
Figure 2.11.	Assumed hidden common domain relation	20
Figure 2.12.	Labeled variants and additional foreign keys of table <i>PRODREF</i>	20
Figure 2.13.	Variants of table <i>PRODREF</i>	20
Figure 2.14.	Detected optimization and aggregation structures	21
Figure 2.15.	Implication of relational constraints on the cardinality of relationships	22
Figure 2.16.	Summary of analysis results	23
Figure 2.17.	Conceptual schema for PDIS (detail)	25
Figure 2.18.	Extended conceptual schema for MIS (detail)	26
Figure 2.19.	Extended conceptual schema for MIS after iteration (detail)	27
Figure 2.20.	Implemented extensions of the logical schema	28
Figure 2.21.	MIS architecture	28
Figure 2.22.	Design of the middleware layer (detail)	29
Figure 3.1.	Reference architecture of KBS	34
Figure 3.2.	Sample fuzzy sets with continuous and discrete membership functions	45
Figure 3.3.	Sample fuzzy sets for fuzzy predicates <i>AName</i> , <i>LargeExt</i> , and <i>MediumExt</i>	47
Figure 3.4.	Evaluation summary	53
Figure 4.1.	The proposed schema analysis process	56
Figure 4.2.	Simple GFRN	59
Figure 4.3.	Implication with constraint and negation	59
Figure 4.4.	Implication with conjunction	59
Figure 4.5.	Similarity measures for the seven sample attribute names with the string <i>userid</i> .	60
Figure 4.6.	Implication with threshold	60
Figure 4.7.	Premise with universal quantifier	61
Figure 4.8.	Variable aggregation and composition	62
Figure 4.9.	Combination of heuristics in a single GFRN	62
Figure 4.10.	Characteristics for classifying automatic analysis operations	64
Figure 4.11.	GFRN with data- and goal-driven predicates	64
Figure 4.12.	Goal-driven analysis operation <i>validate_IND</i>	65

Figure 4.13. $N(\text{validIND}^2(i,v))$ for the case of no counterexamples	65
Figure 4.14. GFRN to illustrate the formalization	68
Figure 4.15. Translation algorithm $GFRN2NPL^1$	69
Figure 4.16. Translation algorithm $Impl2NPL^1$	70
Figure 4.17. Algorithm $OperateGFRN$	73
Figure 4.18. Excerpt of case study	75
Figure 4.19. Necessity degrees for the facts produced by $\Omega(\text{ANameIsRSName+ID}^1)(B)$	75
Figure 4.20. Necessity degrees for the facts produced by $\omega(\text{validKey}^1)(B, \text{validKey}^1(x))$	76
Figure 4.21. Belief revision phase 1: computation of fuzzy truth tokens	79
Figure 4.22. Belief revision phase 2: Computation of FBMs	79
Figure 4.23. The proposed iterative and interactive inference process	82
Figure 4.24. Representation of an expanded GFRN implication (sample)	83
Figure 4.25. Forward and backward expansion (sample)	84
Figure 4.26. Information sources for inference example	85
Figure 4.27. GFRN to exemplify the inference process	86
Figure 4.28. FPN after the first expansion/evaluation cycle	86
Figure 4.29. FPN after second expansion/evaluation cycle	87
Figure 4.30. FPN after third expansion/evaluation cycle	88
Figure 4.31. Additional implications to specify necessary conditions for R-INDs	88
Figure 4.32. FPN after considering human input	89
Figure 4.33. Final analysis result	89
Figure 4.34. Representation of human assumptions	90
Figure 4.35. Algorithm $GFRNInference$	91
Figure 4.36. Algorithm $CreatePlace$	93
Figure 4.37. Algorithm $ExpandFPN$	94
Figure 4.38. Algorithm $ComputeBindingsForImpl$	96
Figure 4.39. Algorithm $ComplementBindings$	97
Figure 4.40. Example GFRN for termination problem	98
Figure 4.41. Architecture of the <i>Varlet Analyst</i>	100
Figure 4.42. <i>Customization Front-End</i>	102
Figure 4.43. <i>Customization Front-End</i> (2)	103
Figure 4.44. <i>Analysis Front-End</i> (overview)	104
Figure 4.45. <i>Analysis Front-End</i> (detail view)	105
Figure 4.46. Graphical and textual documentation of an analyzed logical schema	107
Figure 5.1. Incremental schema migration and generative data integration	115
Figure 5.2. Migration graph model	117
Figure 5.3. Graph test $DuplicateClassName$	119
Figure 5.4. Sample situation: correspondence among variant and inheritance structures	120
Figure 5.5. Graph production $AddRSToLSchema$	122
Figure 5.6. Mapping rule $MapRSToClass$	123
Figure 5.7. Reverse production $MapRSToClass_{rv}$	124
Figure 5.8. Forward production $MapRSToClass_{fv}$	125
Figure 5.9. Startgraph for schema migration	126
Figure 5.10. Algorithm $MapSchema$	126
Figure 5.11. Mapping rule $MapVariantToConcreteClass$	127
Figure 5.12. Example RS <i>Tenant</i> with two variants and their conceptual representation	128
Figure 5.13. Example application of rules $MapRSToClass$ and $MapVariantToConcreteClass$	128
Figure 5.14. Production $MapVariantsToAbstractClass_{rv}$	129
Figure 5.15. Example application of production $MapVariantsToAbstractClass_{rv}$	130
Figure 5.16. Production $MapVariantsToInheritance_{rv}$	131

Figure 5.17. Example application of production <i>MapVariantsToInheritance_{rv}</i>	132
Figure 5.18. Mapping rule <i>MapColToAttr</i>	133
Figure 5.19. Mapping rule <i>MapRINDToAssoc[1:1]</i>	134
Figure 5.20. Mapping rule <i>MapRINDToAssoc[N:0,1]</i>	135
Figure 5.21. Mapping rule <i>MapRINDToAssoc[0,N:0,1]</i>	135
Figure 5.22. Mapping rule <i>MapIINDToInheritance</i>	136
Figure 5.23. Catalog of conceptual redesign transformations	139
Figure 5.24. Schema transformation <i>SplitClass</i>	140
Figure 5.25. Schema transformation <i>MoveAttribute</i>	141
Figure 5.26. Schema transformation <i>Generalize</i>	142
Figure 5.27. Schema transformation <i>PushUpAttribute</i>	143
Figure 5.28. Complex transformation <i>MoveOverAggregation</i>	145
Figure 5.29. Basic structure of a history graph	146
Figure 5.30. Template of transformation <i>Generalize</i>	147
Figure 5.31. History graph model	149
Figure 5.32. Phase I: forward propagation	151
Figure 5.33. Phase II: backward propagation	152
Figure 5.34. Phase III: reevaluation	152
Figure 5.35. Phase IV: translation	153
Figure 5.36. Transaction <i>PropagateChange</i>	154
Figure 5.37. Graph test <i>Generalize_getParams</i>	155
Figure 5.38. Production <i>Generalize_withParams</i>	156
Figure 5.39. Architecture of the <i>Varlet Migrator</i>	157
Figure 5.40. Using the Progres environment to extend module <i>Redesign Transformation</i>	158
Figure 5.41. Logical schema after first analysis step and its initial conceptual translation	160
Figure 5.42. Redesigned conceptual schema (<i>Migration Front-End</i>)	161
Figure 5.43. Completed logical schema (top) and updated logical schema (bottom)	162
Figure 5.44. Implementation of conceptual extensions (<i>Analysis Front-End</i>)	164
Figure 5.45. <i>ObjectDRIVER</i> overview	165
Figure 5.46. Integration of the <i>ObjectDRIVER</i> middleware generator as a back-end for <i>Varlet</i>	166
Figure 5.47. Relational schema description for <i>ObjectDRIVER</i>	167
Figure 5.48. Object schema description for <i>ObjectDRIVER</i>	168
Figure 5.49. Mapping description for classes and subclasses	169
Figure 5.50. Test <i>getClassInstantiationConstraint</i>	169
Figure 5.51. Mapping description for base table attributes	170
Figure 5.52. Test <i>getAttrMappedToCollInBaseTable</i>	170
Figure 5.53. Mapping description for remote attributes	171
Figure 5.54. Test <i>getAttrMappedToCollInRemoteTable</i>	171
Figure 5.55. Mapping description for base table relationships	172
Figure 5.56. Test <i>getRelMappedToBaseTable</i>	172
Figure 5.57. Mapping description for remote relationships	173
Figure 5.58. Test <i>getRelMappedToRemoteTable</i>	173
Figure 5.59. Mapping description for IND-based inheritance relationships	174
Figure 5.60. Test <i>getInheritMappedTo_IND</i>	174
Figure 5.61. Mapping Description for <i>ObjectDRIVER</i>	175
Figure 5.62. MIS application code (example)	176
Figure 6.1. Self-adapting analysis process	185
Figure B.1. Transformation <i>Aggregate</i>	196
Figure B.2. Transformation <i>AssociationToClass</i>	197
Figure B.3. Transformation <i>ChangeAssocCardinality</i>	198

Figure B.4. Transformation <i>ChangeAttributeType</i>	198
Figure B.5. Transformation <i>ClassToAssociation</i>	199
Figure B.6. Transformation <i>CreateAssociation</i>	200
Figure B.7. Transformation <i>CreateAttribute</i>	200
Figure B.8. Transformation <i>CreateClass</i>	201
Figure B.9. Transformation <i>CreateInheritance</i>	201
Figure B.10. Transformation <i>CreateKey</i>	202
Figure B.11. Transformation <i>ConvertAbstract</i>	202
Figure B.12. Transformation <i>ConvertConcrete</i>	203
Figure B.13. Transformation <i>DisAggregate</i>	204
Figure B.14. Transformation <i>Generalize</i>	205
Figure B.15. Transformation <i>MergeClass</i>	206
Figure B.16. Transformation <i>MoveAttribute</i>	207
Figure B.17. Transformation <i>PushDownAttribute</i>	208
Figure B.18. Transformation <i>PushDownAssociation</i>	209
Figure B.19. Transformation <i>PushUpAttribute</i>	210
Figure B.20. Transformation <i>PushUpAssociation</i>	211
Figure B.21. Transformation <i>Remove</i>	212
Figure B.22. Transformation <i>RenameClass</i>	212
Figure B.23. Transformation <i>RenameAttribute</i>	212
Figure B.24. Transformation <i>RenameRelationship</i>	213
Figure B.25. Transformation <i>SplitClass</i>	213
Figure B.26. Transformation <i>Specialize</i>	214
Figure B.27. Transformation <i>SwapAssocDirection</i>	215

LIST OF DEFINITIONS

Definition 1.1	Legacy software system	1
Definition 1.2	Software reengineering	2
Definition 1.3	Reverse engineering	2
Definition 1.4	Database reengineering	4
Definition 3.1	Data model	37
Definition 3.2	Database	38
Definition 3.3	Relational database	38
Definition 3.4	(Notation)	38
Definition 3.5	Flattening	38
Definition 3.6	Basic probability assignment, focal proposition	42
Definition 3.7	Combination of evidences	43
Definition 3.8	Belief function	43
Definition 3.9	Plausibility function	44
Definition 3.10	Fuzzy set	45
Definition 3.11	t-norm and t-conorm	46
Definition 3.12	Fuzzy relation	47
Definition 3.13	Fuzzy logical operators	48
Definition 3.14	Fuzzy inference	48
Definition 3.15	Necessity-valued formula	49
Definition 3.16	Classical projection	50
Definition 3.17	α -cut	50
Definition 3.18	Partial contradicting set of formulae	51
Definition 3.19	Best model	51
Definition 3.20	Formal system for NPL ¹	51
Definition 4.1	Signature of an analyzed logical schema	58
Definition 4.2	Signature of a GFRN	66
Definition 4.3	Context sensitive syntax	67
Definition 4.4	Declarative semantics of GFRNs	68
Definition 4.5	Extent of a predicate	71
Definition 4.6	Data-driven analysis operation	71
Definition 4.7	Goal-driven analysis operation	71
Definition 4.8	Application context	72
Definition 4.9	Expansion of formulae over a finite universe	72
Definition 4.10	Occurrence of literals	72
Definition 4.11	Semantics of automatic analysis operations	72
Definition 4.12	Fuzzy Petri net	78
Definition 4.13	Stability	80
Definition 4.14	Predecessor	80
Definition 4.15	Axiom	80
Definition 4.16	Axiom-based marking	81
Definition 4.17	Grounded place	92
Definition 4.18	Derivability	95

Definition 4.19 Derivation sink95
Definition 5.1 Graph115
Definition 5.2 Graph production121
Definition 5.3 Application of a production121
Definition 5.4 1-context of a set of nodes148
Definition 5.5 Context of a transformation application148
Definition 5.6 History graph149
Definition 5.7 Application of transformations to a history graph150
Definition A.1 Interpretation of a logical schema187