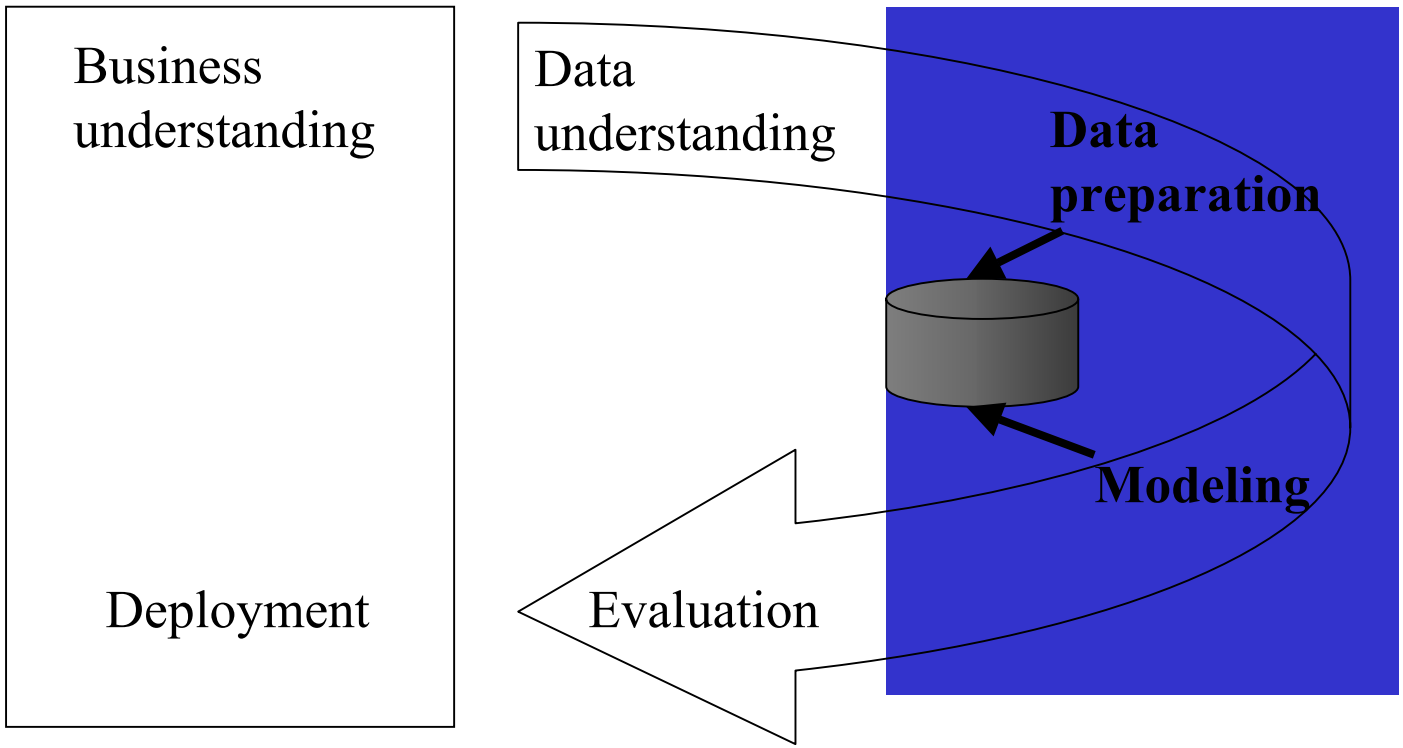
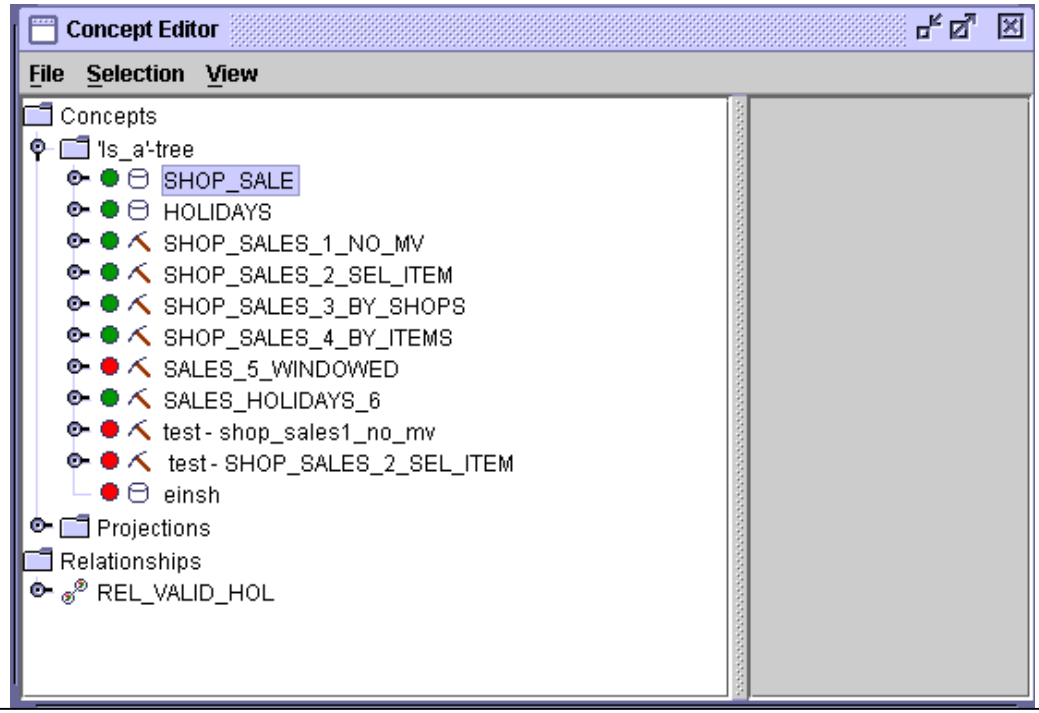
Mining Mart im KDD Prozess

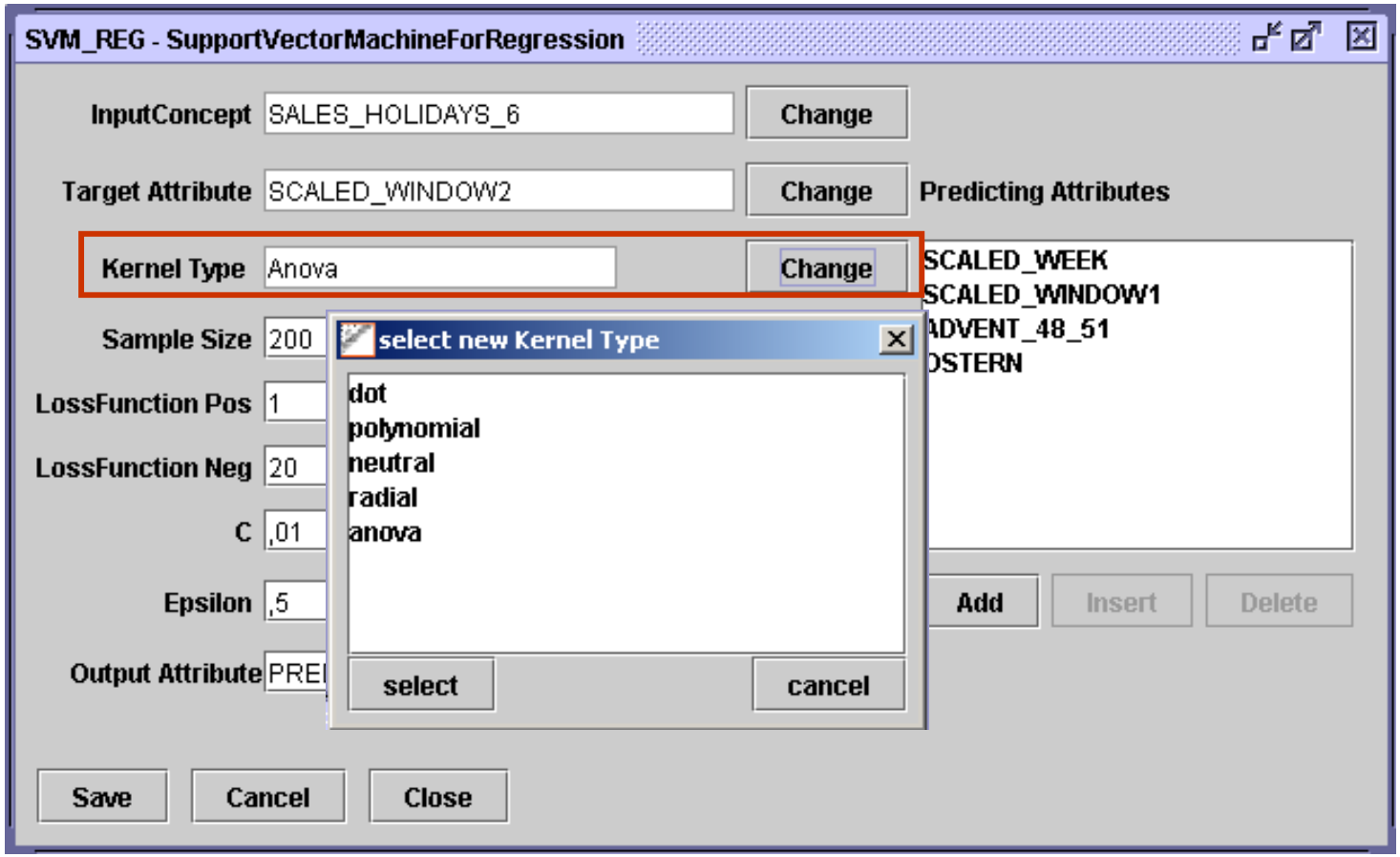


Der Concept Editor

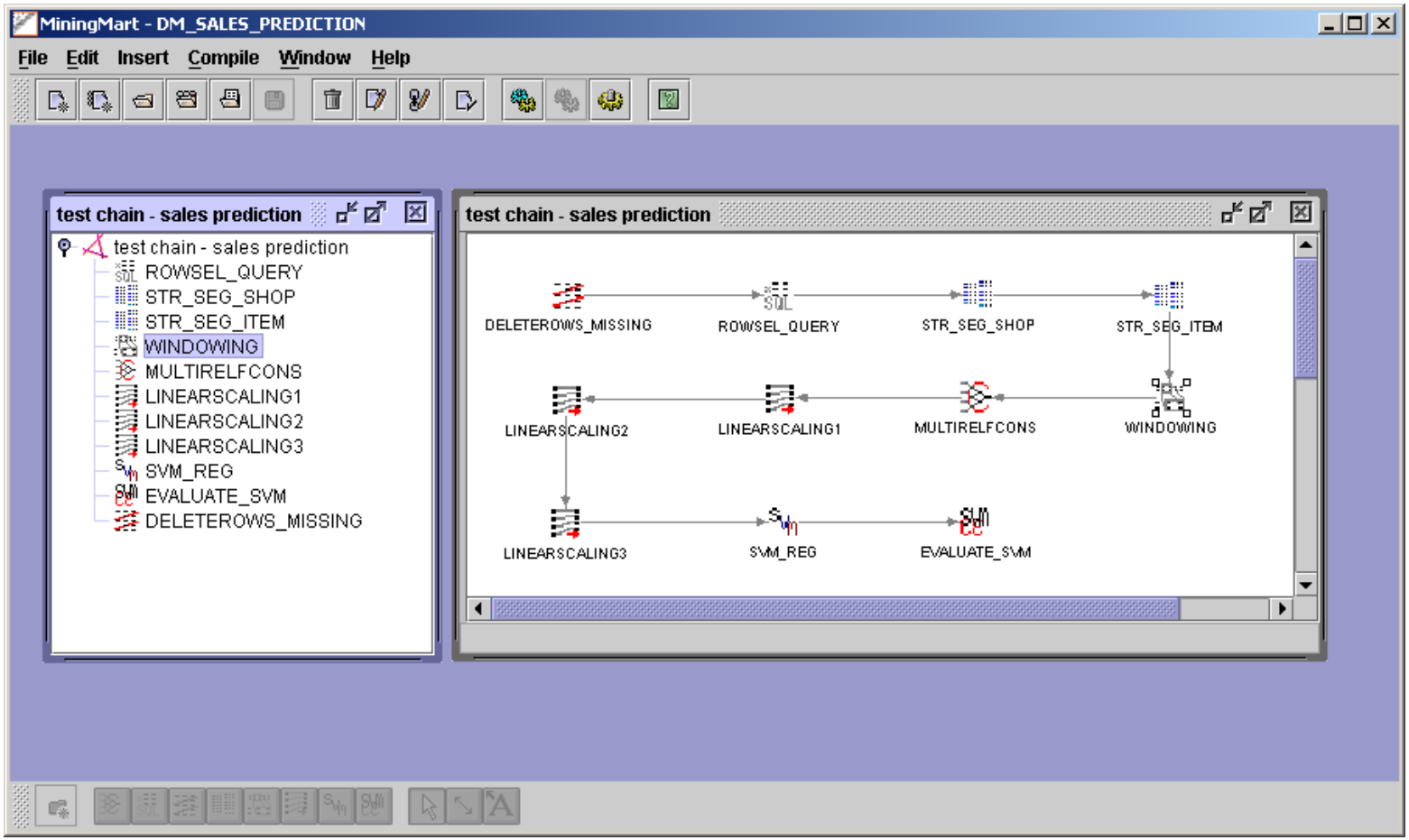


- Definieren und editieren von Begriffen und Relationen
- Abbildung von Begriffen und Relationen auf die Datenbank

Editieren einer SVM Anwendung



The Case Editor



Beispielfall: Abverkaufsprognose

- Vorverarbeitung:
  - Datenbereinigung
  - Beschränkung auf bestimmte Artikel und Geschäfte
  - Abstraktion von Zeitreihen: Windowing
  - Einbringen von Hintergrundwissen: Feiertags-Tabelle
  - Skaliere alle Attribute auf [0...1] für die SVM

Rohdaten des Abverkaufs

Shop_ID	Week	Sale	Item
1	1	1	1
1	1	2	2
1	2	5	3
1	3	6	2
2	1	8	1
2	2	2	1
2	3	1	2
3	1	7	4
3	1		3
3	2	3	1
4	1	2	1
4	1	1	2
5	2	2	6
5	3		2

Schritt1: Operator Delete Missing Values

Delete Missing Values

Shop_ID	Week	Sale	Item	
1	1	1	1	
1	1	2	2	
1	2	5	3	
1	3	6	2	
2	1	8	1	
2	2	2	1	
2	3	1	2	
3	1	7	4	
3	1		3	**Delete**
3	2	3	1	
4	1	2	1	
4	1	1	2	
5	2	2	6	
5	3		2	**Delete**

Schritt 2: Beschränkung auf Artikel 1 und 2

Select Item In (1,2)

Shop_ID	Week	Sale	Item	
1	1	1	1	ok
1	1	2	2	ok
1	2	5	3	delete
1	3	6	2	ok
2	1	8	1	ok
2	2	2	1	ok
2	3	1	2	ok
3	1	7	4	delete
3	2	3	1	ok
4	1	2	1	ok
4	1	1	2	ok
5	2	2	6	delete

Schritt 3: Segment by Shop

Segement by shop

Shop_ID	Week	Sale	Item	
1	1	1	1	ColumnSet1
1	1	2	2	
1	3	6	2	
2	1	8	1	ColumnSet2
2	2	2	1	
2	3	1	2	
3	2	3	1	ColumnSet3
4	1	2	1	ColumnSet4
4	1	1	2	

Schritt 4: Segment by Item

Segment by Item

Shop_ID	Week	Sale	Item	
1	1	1	1	ColumnSet11
1	1	2	2	ColumnSet12
1	3	6	2	ColumnSet13
2	1	8	1	ColumnSet21
2	2	2	1	ColumnSet22
2	3	1	2	ColumnSet23
3	2	3	1	ColumnSet31
4	1	2	1	ColumnSet41
4	1	1	2	ColumnSet42

Schritt 5: Windowing

Windowing ColumSet12 Fenstergröße 2

Shop_ID	Week	Sale	Item	Start	End	Window1	Window2
1	1	2	2	1	2	2	8
1	2	8	2	2	3	8	6
1	3	6	2	3	4	6	1

Beispielhaft für ein Columnset!

Schritt 6: Multirelational Feature Construction

Join with Holidays

Week	Window1	Window2	New Year	Eastern
1	2	8	1	0
2	8	6	0	0
3	6	1	0	0

Schritt 7: Linear Scaling

Week	Window1	Window2	New Year	Eastern
1	0,25	8	1	0
2	1,00	6	0	0
3	0,75	1	0	0

Week	Window1	Window2	New Year	Eastern
1	0,25	1,00	1	0
2	1,00	0,75	0	0
3	0,75	0,13	0	0

Scale Week				
Week	Window1	Window2	New Year	Eastern
0,25	0.25	1.00	1	0
0,50	1.00	0.75	0	0
0,75	0.75	0.13	0	0

Data Mining Schritt: Wende SVM an!

Abstraktionsebenen

- Wissensrepräsentation:
  Wie repräsentiert man Daten und Prozesse von KDD Anwendungen?
  - M4: *Formalismus* zur Repräsentation von KDD-Prozessen
- Systementwickler:
  Wie *implementiert* man den Formalismus für Compiler und GUI?
  - Einbettung in relationale Datenbanken
- Anwender: Repräsentation eines *bestimmten* Sachbereichs, bestimmter Relationen und eines bestimmten Falles
  - Welche Begriffe, Relationen habe ich in einem bestimmten Fall?

Bestandteile vom Metamodell M^4

- Repräsentation von Operatoren:
  - notwendige und optionale Parameter
  - syntaktische / datenbasierte Bedingungen und Zusicherungen
  - Referenz auf ausführbaren (Java-)Code
- Repräsentation abstrahierter KDD Prozesse:
  - Sachbereiche (Begriffe, Relationen)
  - einzelne Schritte im KDD-Prozess
- Operationale Ebene –einzelne KDD Prozesse:
  - Parametersetzungen einzelner Operatoranwendungen
  - Tabellen / Views zu Begriffen des Sachbereichs

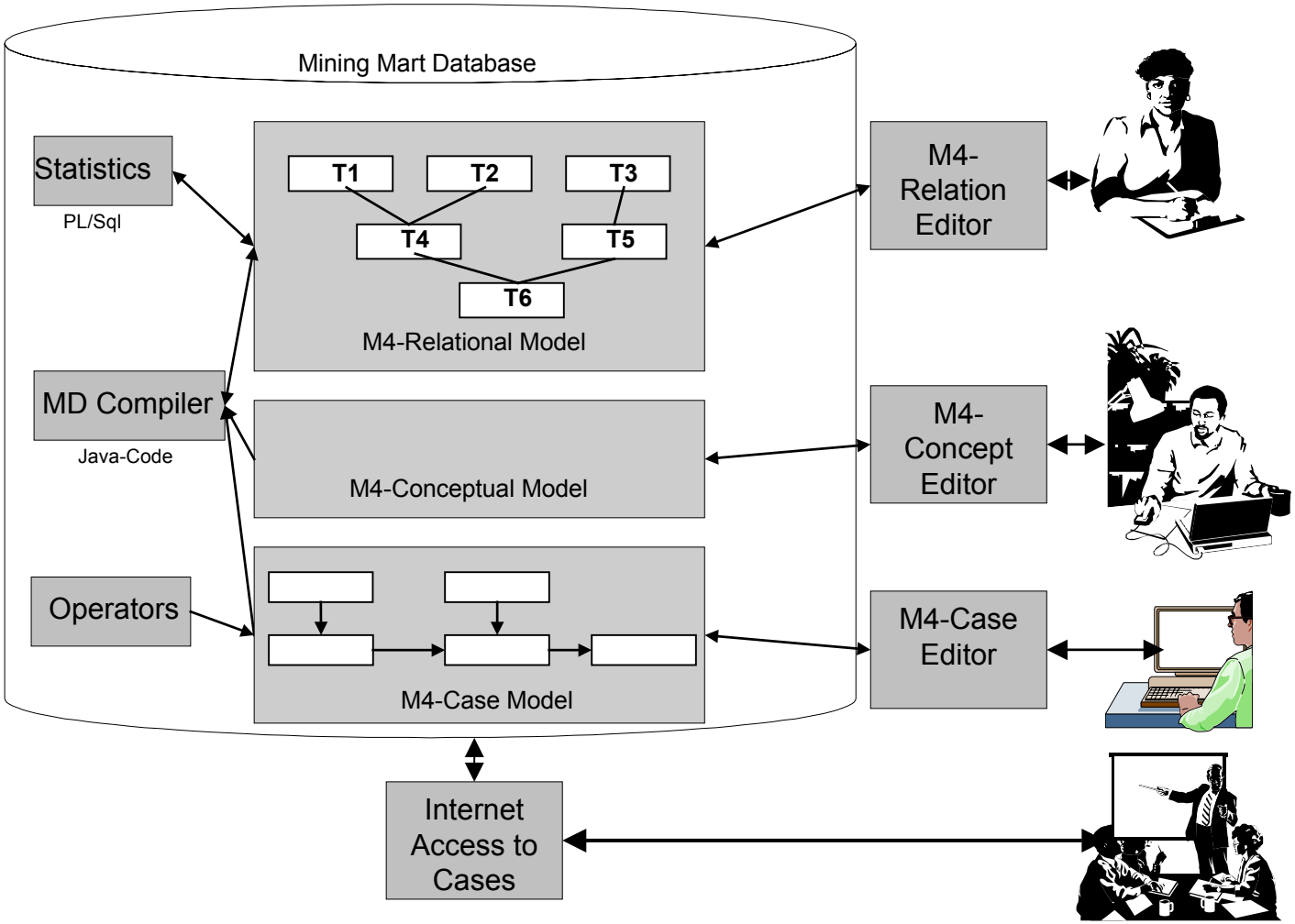
M4 Modell für Metadaten

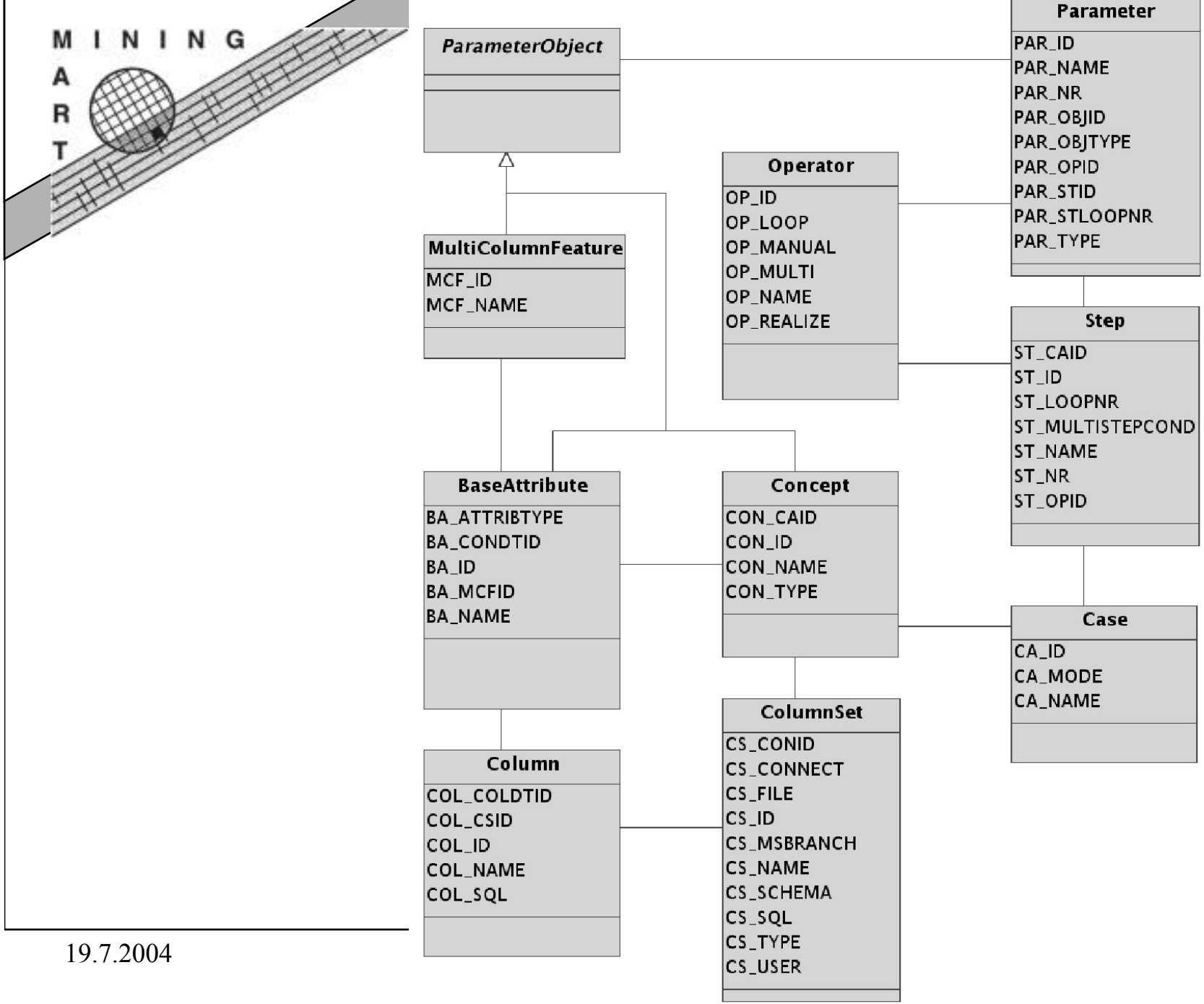
Daten	Daten-Transformationen
Das begriffliche Modell beschreibt die Objekte und Klassen der Anwendung	Das Fallmodell beschreibt Operator-ketten
	Ausführungsmodell: generiert SQL-views / ruft externeVerfahren auf

Erstellung von Metadaten

- Der Datenbankadministrator liefert das relationale Modell.
- Der Anwender liefert das begriffliche Modell.
- Die Datenanalyseexpertin liefert das Fallmodell oder passt es an.
- Die ersten Fälle wurden vom Mining Mart Projekt erstellt.
- MiningMart Systementwickler entwicklen Operatoren und deren Metadaten

Systemarchitektur




ParameterObject
Parameter
PAR_ID
PAR_NAME
PAR_NR
PAR_OBJID
PAR_OBJTYPE
PAR_OPID
PAR_STID
PAR_STLOOPNR
PAR_TYPE
Operator
OP_ID
OP_LOOP
OP_MANUAL
OP_MULTI
OP_NAME
OP_REALIZE
MultiColumnFeature
MCF_ID
MCF_NAME
Step
ST_CAID
ST_ID
ST_LOOPNR
ST_MULTISTEPCOND
ST_NAME
ST_NR
ST_OPID
BaseAttribute
BA_ATTRIBTYPE
BA_CONDTID
BA_ID
BA_MCFID
BA_NAME
Concept
CON_CAID
CON_ID
CON_NAME
CON_TYPE
Case
CA_ID
CA_MODE
CA_NAME
ColumnSet
CS_CONID
CS_CONNECT
CS_FILE
CS_ID
CS_MSBRANCH
CS_NAME
CS_SCHEMA
CS_SQL
CS_TYPE
CS_USER
Column
COL_COLDTID
COL_CSID
COL_ID
COL_NAME
COL_SQL

Relational Data Model

Formalismus zur Beschreibung der Datenbank

- Column
  - Supertype: Attribute
  - Subtypes: None
  - Attributes: name (of the column), dataType
  - Associations: belongsToColumnSet, keys, correspondsToBaseAttribute
- ColumnSet (meist eine Tabelle oder Sicht)
- ColumnStatistics
- ColumnSetStatistics
- Key, PrimaryKey, ForeignKey

Conceptual Data Model

Formalismus zur Beschreibung des Sachbereichs

- Concept
  - Supertype: Class
  - Attributes: name, subConceptRestriction
  - Associations: isA, correspondsToColumnSet, FromConcept, ToConcept Constraints
- Relationship
- FeatureAttribute
- Value
- RoleRestriction
- DomainDataType

Conceptual Data Model cont'd

- BaseAttribute
  - SuperType: FeatureAttribute
  - Associations:
    - domainDataType: is it from the raw data or has it been created?
    - isPartOfMultiColumnFeature: pointer to a set of columns which together form a Feature

Case Model

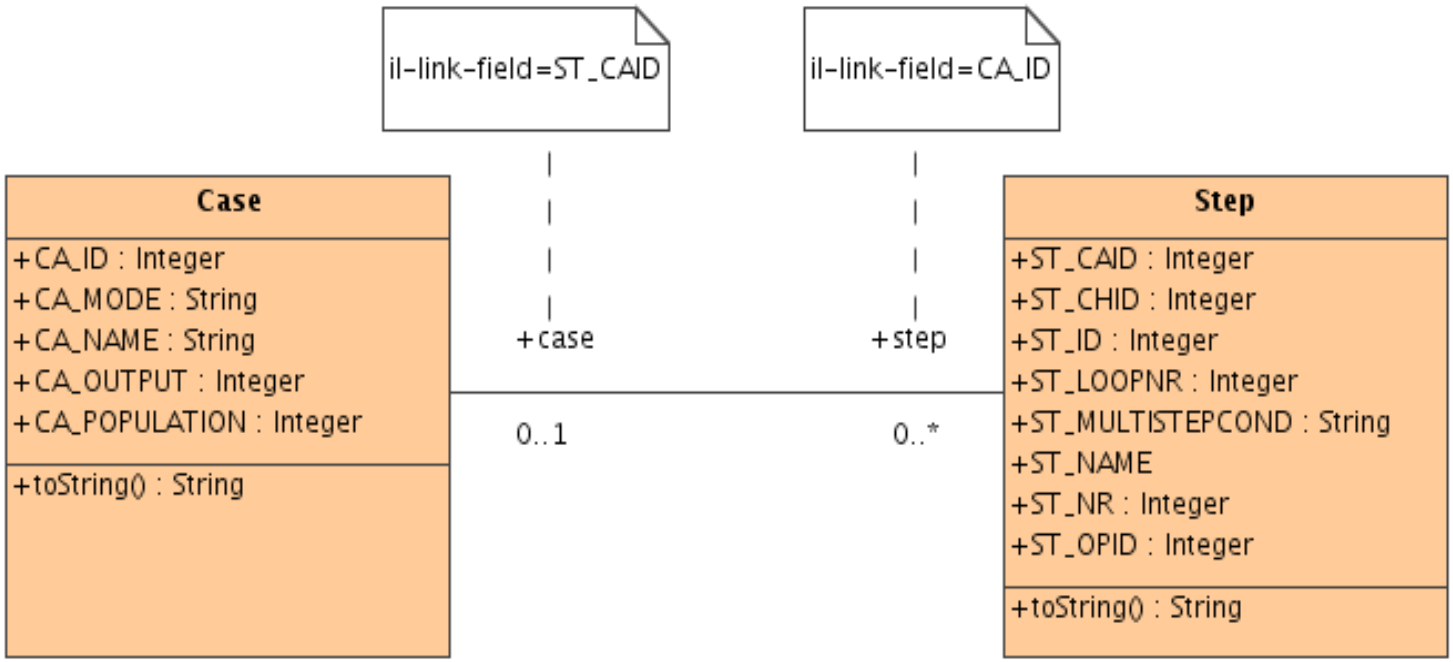
Formalismus zur Beschreibung von KDD-Prozessen

- Case
  - Attributes: name,
    - case mode -- {test, final},
    - caseInput -- list of entities from the conceptual model,
    - caseOutput -- concept, normally the input to the data mining step
    - Documentation – free text
  - Associations:
    - listOfSteps – aggregation of steps,
    - population – concept from caseInput, the one the analysis deals with,
    - targetAttributes – FeatureAttribute to which the data analysis is applied

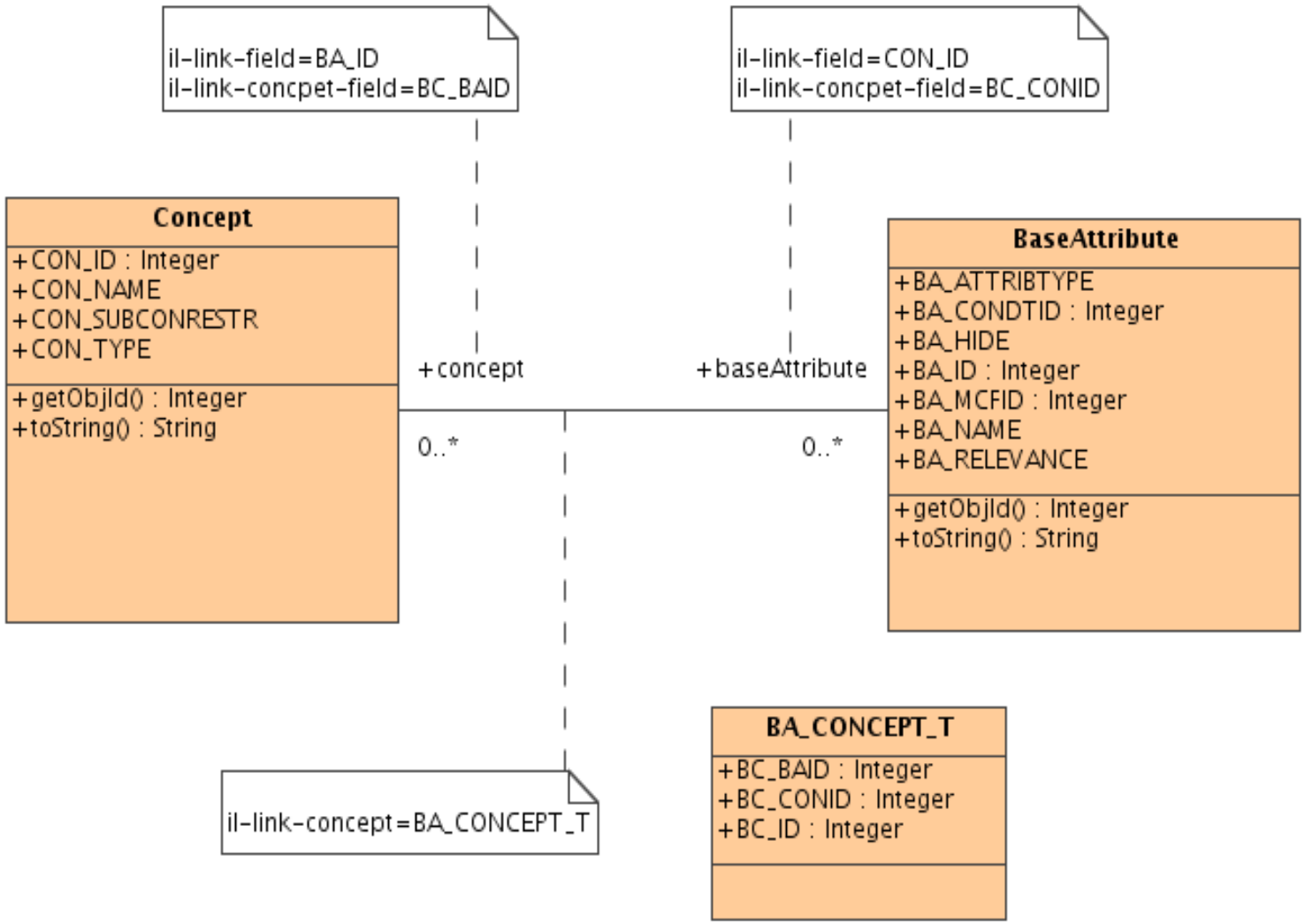
Case Model cont'd

- Step
  - Attributes: name
  - Associations: belongsToCase, embedsOperator, predecessor, successor
- Operator
  - Attributes with values {yes, no}:
    - loopable -- apply operator several times with changed parameters,
    - multi-stepable – operator delivers several results which will be processed separately in parallel,
    - manual – using no external algorithm
  - Associations:
    - parameters forming the input of the operator
    - conditions -- to be checked given the data,
    - constraints -- to be checked without access to data,
    - assertions – will be true after operator execution

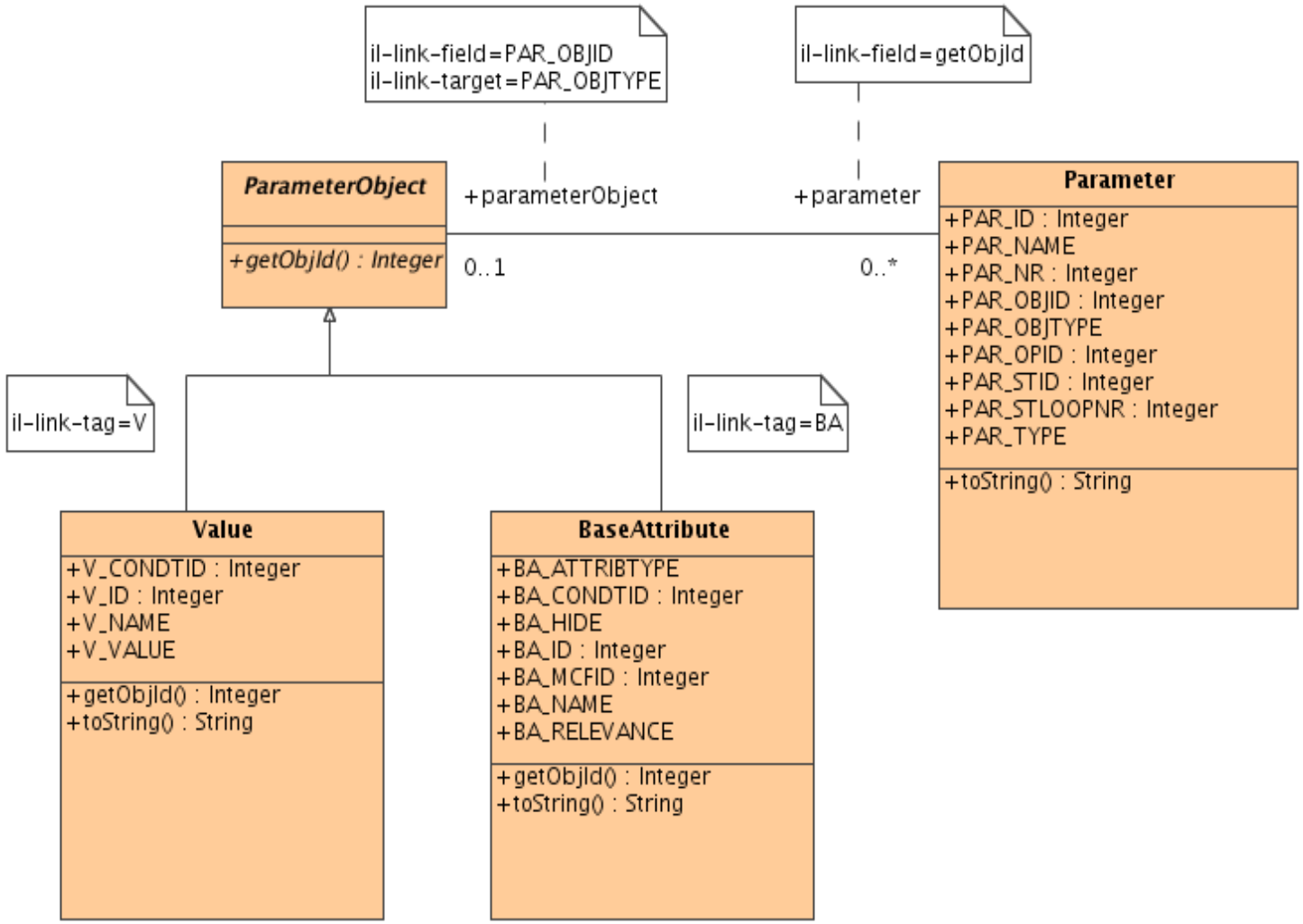
Ein Case enthält mehrere Steps



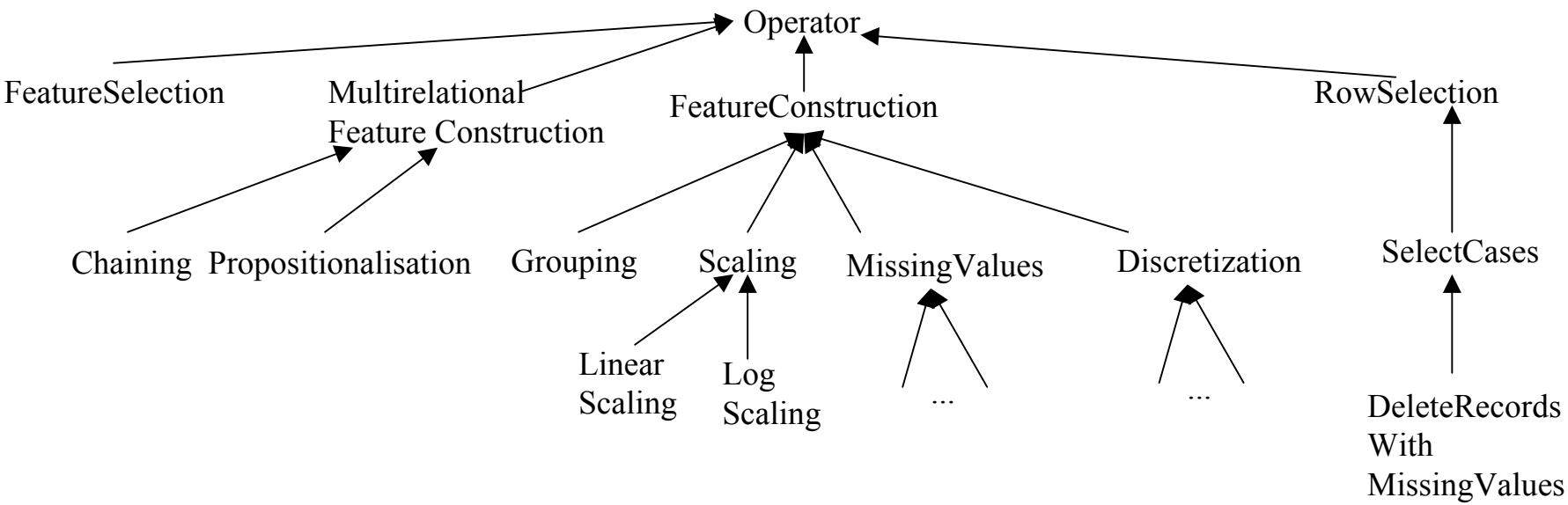
Konzepte und Attribute: n:m Assoziationen



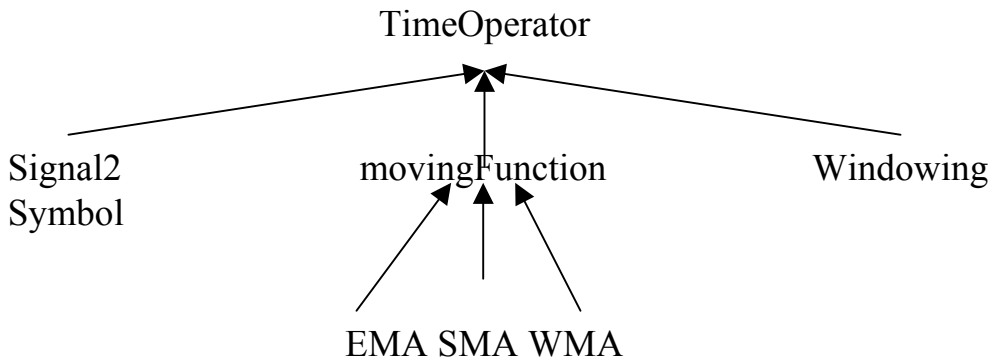
Parameter: Dynamische Assoziationen



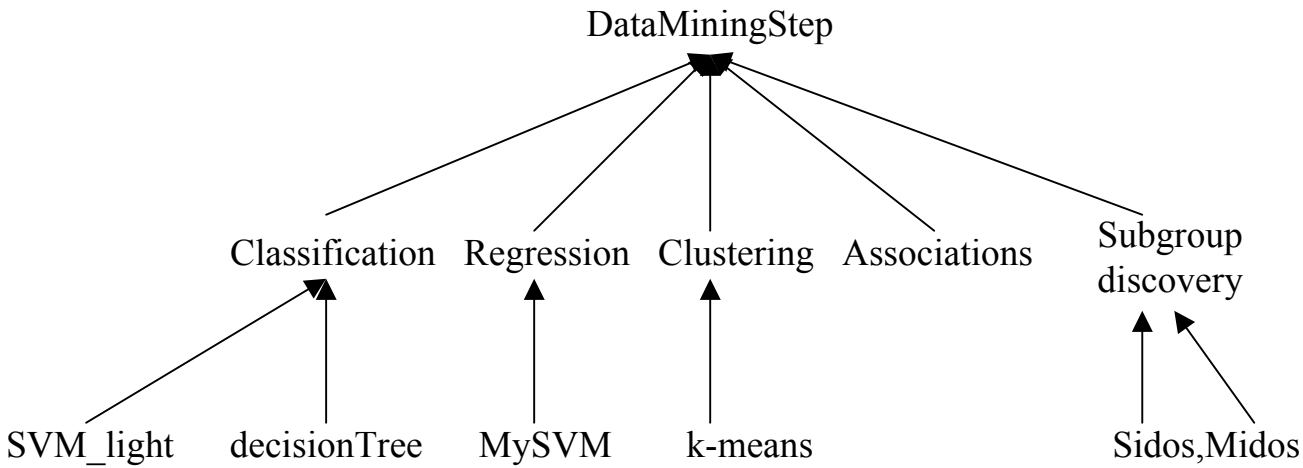
Operatoren des Metamodells



Time Operators in M4



Lernoperatoren des Metamodells



Lernoperatoren sind auch Vorverarbeitungoperatoren!
Beispiel: C4.5 zur Discretisierung oder Ersetzung fehlender Werte.

M4 Modell

- Sie haben Ausschnitte aus dem Formalismus gesehen, der für die Definition eines bestimmten Modells benötigt wird.
- Es ist das Metamodell für die Metadaten .
  In Auszeichnungssprachen (SGML, XML) entspricht dies dem Formalismus der DTD, nicht einer DTD, sondern den formalen Mitteln, eine DTD zu definieren.
- Der Formalismus beinhaltet Klassen mit Vererbung, Attribute und Assoziationen.
- Die Definitionen des Metametamodells sind für alle Sprachen in diesem Formalismus gültig.

Implementierung von M4

- Die M4-Tabellen sind Teil des Systems.
- Die M4 Metadaten werden ebenfalls in Datenbanktabellen gespeichert:
  - Relationenkalkül als bewährter Basisformalismus
  - Konsistenzprüfung und Transaktionsmanagement
- Compiler und GUI setzen auf denselben Tabellen auf.
- Die Tabellen oder Sichten des konkreten Falles
  - kommen von der Anwendung her (Typ: DB)
  - oder werden vom Compiler geschrieben (Typ: Mining).

M4 -- Base Attribute

- Basisattribute entsprechen auf der begrifflichen Ebene einer Spalte der relationalen Ebene.
- Tabelle BASEATTRIB_T hat die Attribute:
  - BA_ID: M4 ID vom Typ Integer
  - BA_NAME: Name vom Typ String
  - BA_CONDTID: Typ des Basisattributs, Fremdschlüssel für die Tabelle CONDATATYPE_T, die die Begriffe des M4-Modells enthält
  - BA_ATTRIBTYPE: ‚DB' (Rohdaten) oder ‚MINING' (durch Preprocessing erzeugte Spalte)
  - BA_MCFID: Fremdschlüssel für die Tabelle MCFEATURE_T, falls dieses Basisattribut gemeinsam mit anderen ein multi-column feature bildet
  - BA_VALID: ‚YES', ‚NO' zeigt an, ob es die entsprechende Spalte in den Rohdaten oder in einer erzeugten Sicht gibt.

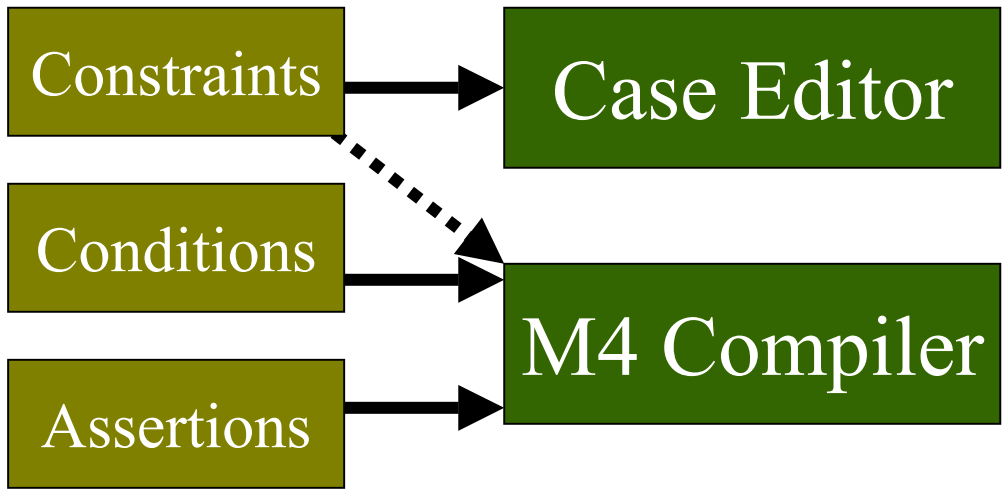
Verbindende Tabellen

- BA_COLUMN_T verbindet Basisattribute mit entsprechenden Spalten des relationalen Modells. Attribute:
  - BAC_ID: ID in dieser Tabelle
  - BAC_BAID: Fremdschlüssel auf BASEATTRIB_T
  - BAC_COLID: Fremdschlüssel auf COLUMN_T des relationalen Modells
- BA_CONCEPT_T verbindet Basisattribute mit ihren Begriffen.
  - BC_ID: ID in dieser Tabelle
  - BC_BAID: Fremdschlüssel auf BASEATTRIB_T
  - BC_CONID: Fremdschlüssel auf CONCEPT_T, die Tabelle der Begriffe

Operatortabellen – für alle Fälle gleich

- OPERATOR_T: allgemeine Definition der Operatoren, OP_PARAMS_T: Eingabe der Operatoren,
- OP_CONSTR_T: Bedingungen für die Anwendung eines Operators, die ohne Daten geprüft werden kann
- OP_COND_T: Bedingungen für die Anwendung eines Operators, die anhand konkreter Daten geprüft werden muss
- OP_ASSERT_T: Aussagen über einen Operator (Nachbedingung)

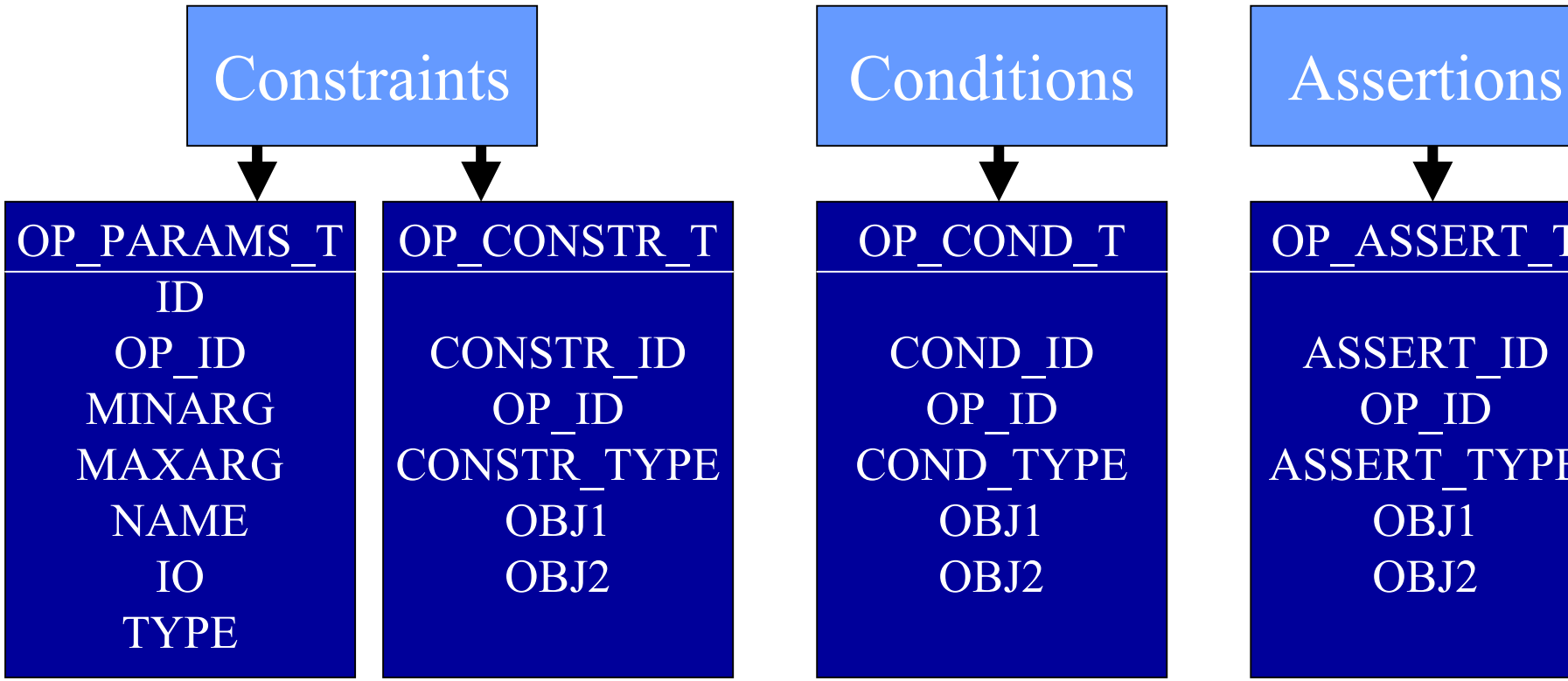
Eigenschaften von Operatoren



Metadaten zu Operatoren

- Case Editor
  - bereitet Ausgaben von Operatoranwendungen vor
  - Parameter werden über geeignetes Fenster editiert
  - Stellt Validität von Operatorsequenzen sicher
- M4 Compiler
  - testet ob Eingaben und Parameter vorhanden und gültig sind
  - Laufzeiteigenschaften der Eingaben werden überprüft
  - Zusicherungen: Vermeidung überflüssiger Datenbankzugriffe

M4 Tabellen zur Operatorbeschreibung



Example: Constraints SVM (1)

- "loopable", nicht "multistepable", nicht "manual"
- genau ein Eingabekonzept
- genau ein Zielattribut
- genau ein Ausgabeattribut
- Vorhersageattribute:
  - Teilmenge der Attribute des Eingabekonzepts
  - Keine fehlenden Werte
  - Typ: SCALAR

Beispiel: Constraints SVM (2)

- Zielattribut:
  - Teil des Eingabekonzepts
  - Typ: skalar
- Parameter vom Typ skalar und positiv:
  "C", "Epsilon", "LossFunctionPos", "LossFunctionNeg"
- Mögliche Werte vom Parameter "KernelType":
  "dot", "polynomial", "radial", "neural", "anova"

Constraints SVM (3)

OPERATOR_T

OP_ID	OP_NAME	OP_LOOP	OP_MULTI	OP_MANUAL
53	SUPPORT_VECTOR_MACHINE	YES	NO	NO

OP_PARAMS_T

ID	OP_ID	MINARG	MAXARG	NAME	IO
1018	53	1	1	TheInputConcept	IN
1019	53	1	NULL	PredictingAttributes	IN
1020	53	1	1	C	IN
1021	53	1	1	Epsilon	IN
1022	53	1	1	LossFunctionPos	IN
1023	53	1	1	LossFunctionNeg	IN
1024	53	1	1	KernelType	IN
1025	53	1	1	TargetAttribute	IN
1026	53	1	1	OutputAttribute	OUT

Constraints SVM (4)

OP_CONSTR_T				
CONSTR_ID	OP_ID	CONSTR_TYPE	OBJ1	OBJ2
1035	53	IN	PredictingAttributes	TheInputConcept
1036	53	TYPE	PredictingAttributes	SCALAR
1037	53	IN	TargetAttribute	TheInputConcept
1038	53	TYPE	TargetAttribute	SCALAR
1039	53	TYPE	C	SCALAR
1040	53	TYPE	LossFunctionPos	SCALAR
1041	53	TYPE	LossFunctionNeg	SCALAR
1042	53	TYPE	Epsilon	SCALAR
1043	53	GT	C	0
1044	53	GT	LossFunctionPos	0
1045	53	GT	LossFunctionNeg	0
1046	53	GT	Epsilon	0
1047	53	ONE_OF	Kernel_Type	"dot, polynomial, .."
1048	53	IN	OutputAttribute	TheInputConcept

Conditions SVM

- Mindestens ein fehlender Wert im Zielattribut
- Keine fehlenden Werte in den Vorhersageattributen

OP_COND_T				
COND_ID	OP_ID	COND_TYPE	OBJ1	OBJ2
1050	53	HAS_NULLS	TargetAttribute	NULL
1051	53	NOT_NULL	ThePredictingAttributes	NULL

Assertions for SVM

- Nach Anwendung kein fehlender Wert im Zielattribut

OP_ASSERT_T				
ASSERT_ID	**OP_ID**	**ASSERT_TYPE**	**OBJ1**	**OBJ2**
1052	53	NOT_NULL	TargetAttribute	NULL

Realisierung der Operatordefinition MissingValues

Input: TheConcept, TargetAttribute, PredictingAttributes
Output: FilledAttribute
Condition: TargetAttribute is a BA with missing values
Constraint: TargetAttribute and PredictingAttributes belong to TheConcept
Assertion: FilledAttribute belongs to TheConcept, FilledAttribute is a BA wo. MV

OP_PARAMS_T

PARAM-ID	OP_ID	...	IO	TYPE
P001	O001		IN	Concept
P002	O001		IN	BaseAttribute
P003	O001		IN	BaseAttributes

CONST_ID	CONST_OP_ID	TYPE	OBJ_1	OBJ_2
CS01	O001	IN	P002	P001

OPERATOR_T

OP_ID	OP_NAME	...	OP_MANUAL
O001	Missing Values		YES

COND_ID	COND_OP_ID	TYPE	OBJ_1	OBJ_2
CD01	O001	HAS_NULLS	P002	

Operator als Schritt in einem KDD-Prozess

- CASE_T: ID, Name, Validity
- STEP_T: verbindet CASE_T und OPERATOR_T
  zeigt auf eine Tabelle mit konkreten Parametern für diesen Schritt (PARAMETER_T)

PARAMETER_T

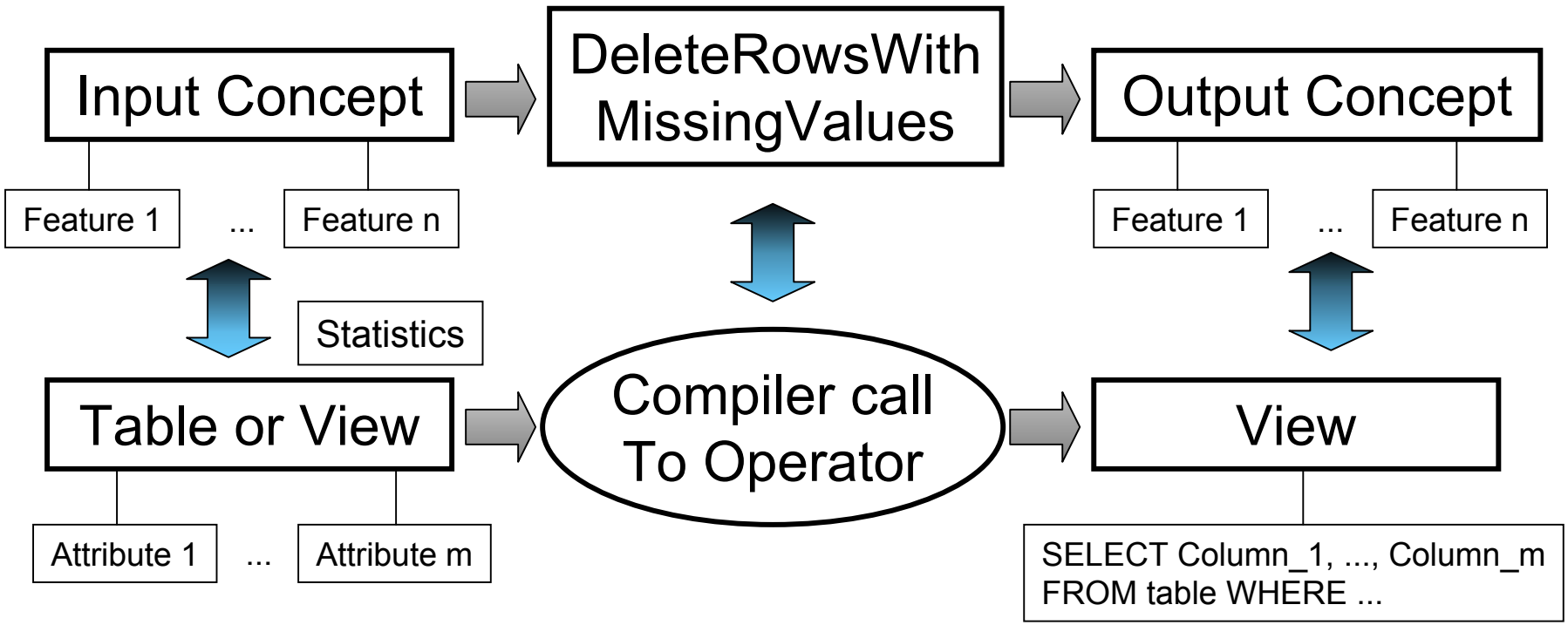
PAR_ID	PAR_OBJ_ID	PAR_OBJ_TYPE	PAR_OP_ID	PAR_ST_ID	...
	Foreign link to *	{BA, MCF, CON,..}	Foreign link to OPERATOR_T	Foreign link to STEP_T	

* ∈ BASEATTRIB_T
MCFEATURE_T
VALUE_T
CONCEPT_T
RELATION_T

Was wissen Sie jetzt?

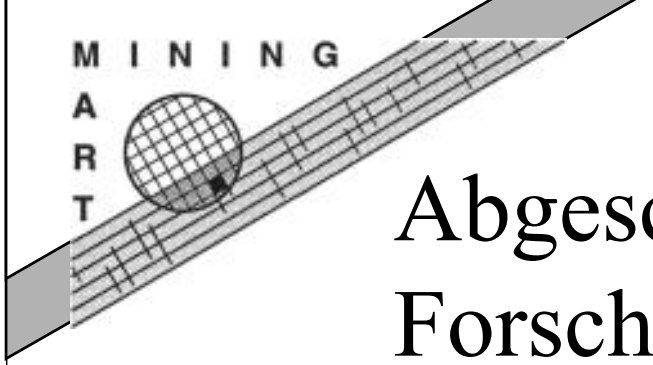
- Das Metamodell für KDD-Prozesse und Sachbereiche ist ein Formalismus, mit dem man bestimmte Prozesse und Begriffe definieren kann.
- Das Metamodell ist in Form von Datenbanktabellen gespeichert, die durch Fremdschlüsselbeziehungen verknüpft sind.
- KDD-Prozesse werden als Folge von Schritten beschrieben, wobei ein Schritt auf einen Operator zeigt.
- Ein Operator ist allgemein durch Tabellen beschrieben, die per Fremdschlüssel auf ihn zeigen. Beispiel: OP_PARAMS_T
- Ein Schritt zeigt auf einen Operator und wird durch PARAMETER_T beschrieben. PARAMETER_T zeigt auf Zeilen in Tabellen für Begriffe und Attribute.

Compiler verbindet begriffliche und Datenbankebene



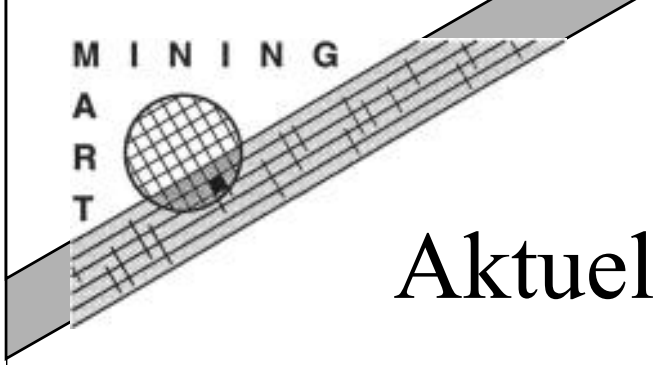
Anwendungen

- Versicherung (SwissLife)
  - Direktes Marketing
  - Analyse der Rückkäufe
- Telekommunikation (TILab, NIT)
  - Unbezahlte Rechnungen, Betrugvorhersage
  - Unterstützung des Call Centers für Telekommunikationsdienste
- Handel (DM, holländische Zeitungen)
  - Abverkaufsprognose



Abgeschlossene europäische Forschungsprojekte

- CRITIKAL Client-Server Rule Induction Technology for Industrial Knowledge Acquisition from Large Databases (1996 – 1998)
- KESO Knowledge Extraction
- CRISP-DM Cross Industry Standard Process for Data Mining (1997 – 2000)



Aktuelle Forschung

- Andere Daten: Zeit, Raum, Genomsequenzen, Sätze
- Engere Anbindung an die Datenbank, Anfrageoptimierung
- Stärkere Unterstützung der frühen Phasen: Dateninspektion, Datenvorverarbeitung
- Metadaten zur Wiederverwendung von DM-Prozessen
- EU-Networks: Sol-EU-net, KDnet
- EU-Projekte: MetaL, Mining Mart, SPIN!, Cinq