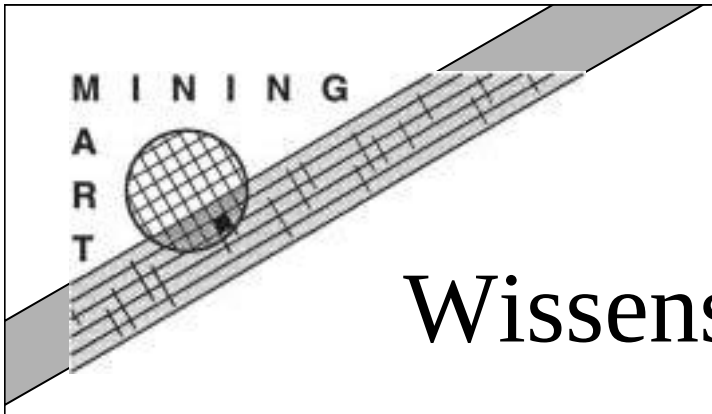


Wissensentdeckung...

…wird laut Gartner Group (1999) in den nächsten 10 Jahren sprunghaft ansteigen: bei n Projekten 1999 werden 2009 schon n^4 Projekte bestehen.

... dient den Firmen vor allem dazu, ihren Kunden besser „zuzuhören" und die Beziehung zu ihnen zu verbessern.

Man kann aber auf der Grundlage vorhandener Daten noch vieles andere verbessern.

Kdnuggets 2002 Poll

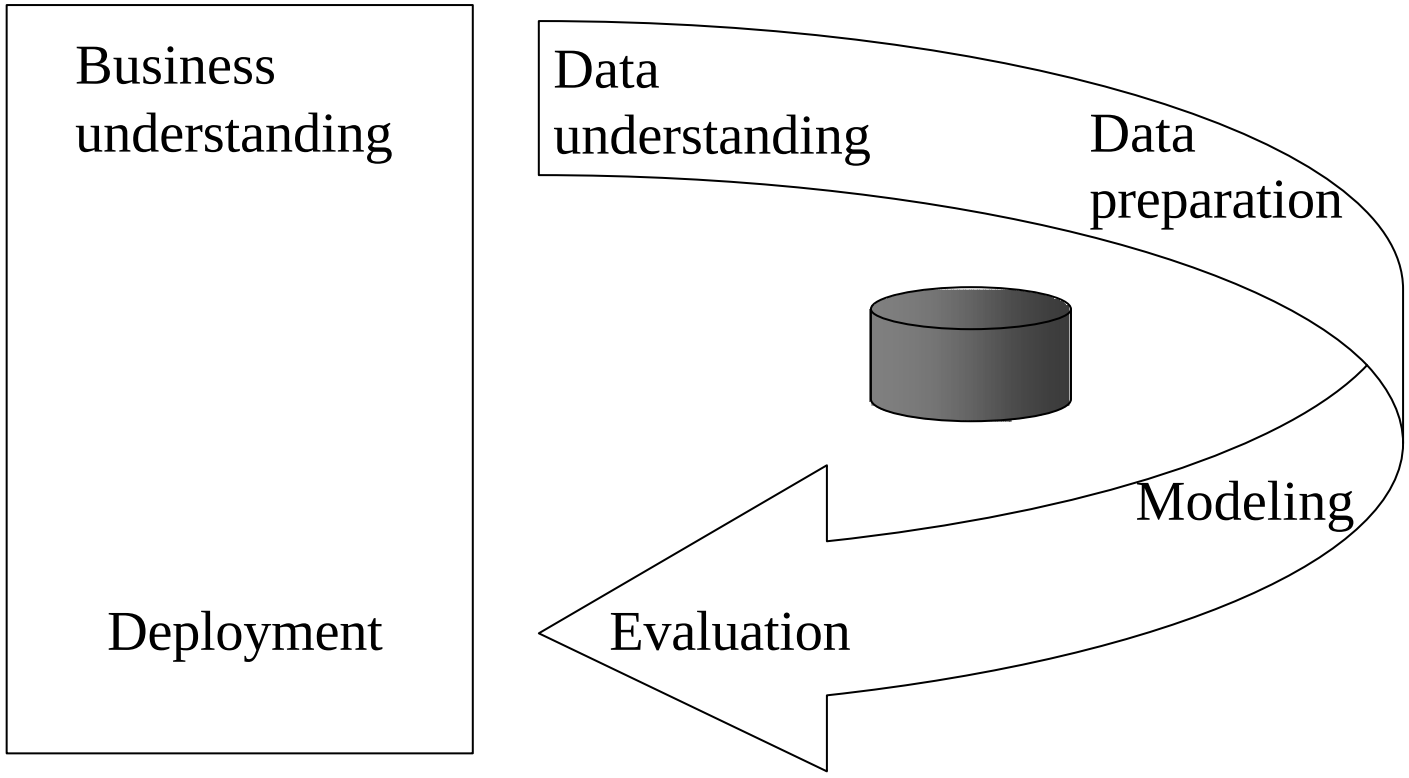
Industries/fields where you currently apply data mining: [608 votes total]	
Banking (77)	13%
Biology/Genetics/Proteomics (32)	5%
Direct Marketing/Fundraising (42)	7%
eCommerce/Web (53)	9%
Entertainment (10)	2%
Fraud Detection (51)	8%
Insurance (36)	6%
Investment/Stocks (17)	3%
Manufacturing (28)	5%
Pharmaceuticals (31)	5%
Retail (36)	6%
Scientific data (51)	8%
Security (14)	2%
Supply Chain Analysis (21)	3%
Telecommunications (56)	9%

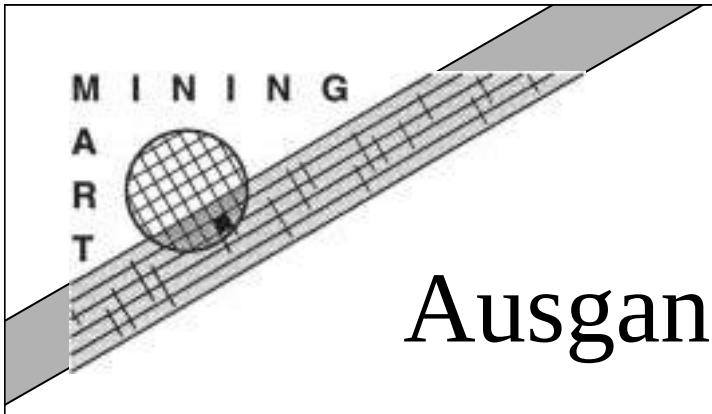
Häufig benutzte Systeme/ Verfahren

- SPSS CLEMENTINE (128)
- Weka (101)
- SAS (100)
- CART MARS (89)
- SAS Enterprise Miner (67)
- IBM Intelligent Miner (35)

- Statistische Verfahren
- Verfahren des maschinellen Lernens
- Data cube und Frequent Itemsets

CRISP-DM Process Model

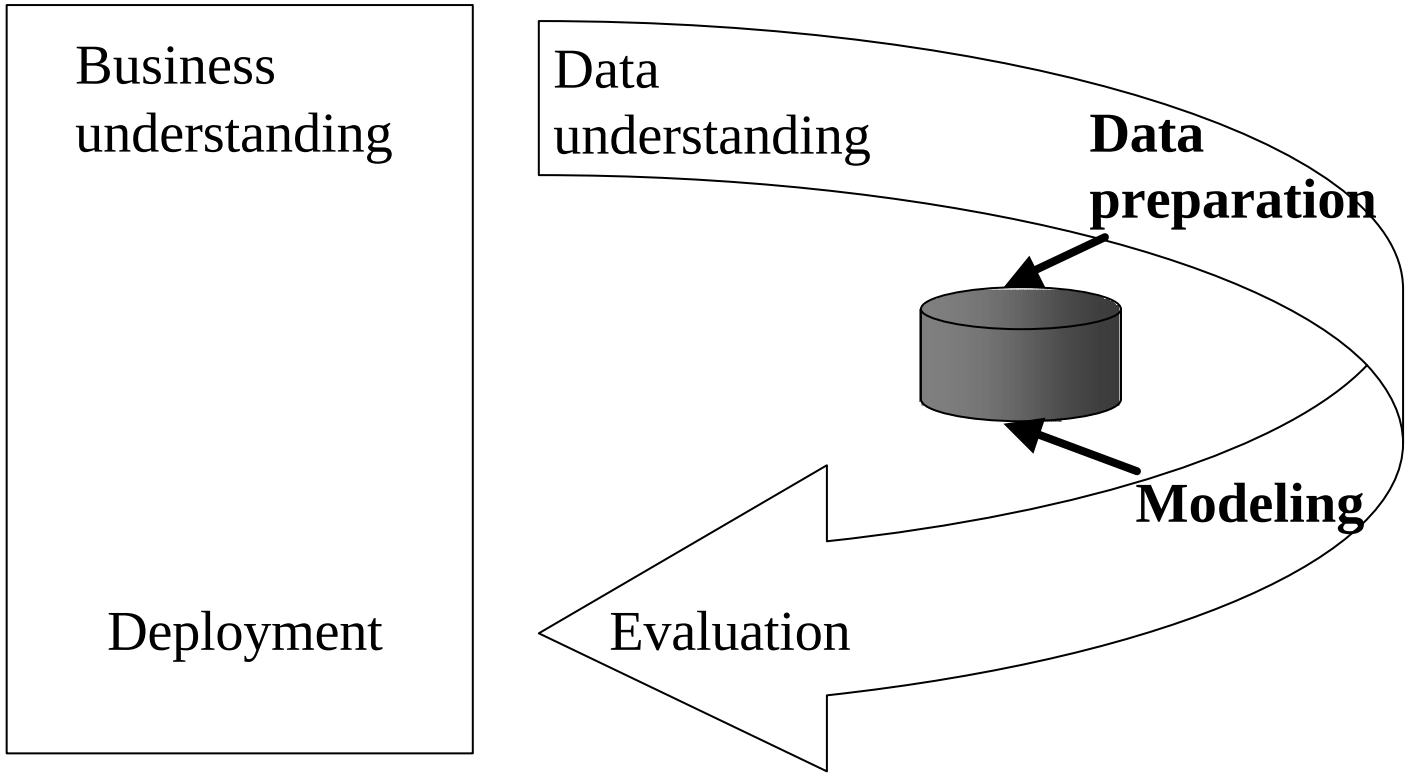




Ausgangssituation

- Datenvorverarbeitung muss besser unterstützt werden
  - 80 % des Aufwands der Datenanalyse sind Vorverarbeitung
  - Bessere Daten bringen bessere Analyseergebnisse
- Dokumentation der Vorverarbeitung fehlt meist
  - Ähnliche Fälle müssen jedesmal neu erstellt werden
  - Erfahrungen der Mitarbeiter werden nicht weitergegeben
- Datenanalyse ist eine Anfragesprache an Datenbanken
  - Datenanalyse sollte in DBMS integriert sein
  - Anfragen für die Datenanalyse müssen hoch optimiert werden

Mining Mart Focus



Mining Mart Ziele

- Operatoren für die Vorverarbeitung
  - direkt auf der Datenbank
  - maschinelles Lernen für die Vorverarbeitung
- Dokumentation
  - der Daten
  - der Fälle
- Wiederverwendung von abstrahierten Fällen

Mining Mart Ansatz

- Metadaten zur Beschreibung von
  - Daten,
  - Operatoren und
  - Fällen (Sequenzen von Operatoren)
- Compiler, der Metadaten in ausführbaren SQL-Code übersetzt oder externe Verfahren aufruft
- Sammlung von Fällen in Form von operationalen Metadaten

Mining Mart Ansatz

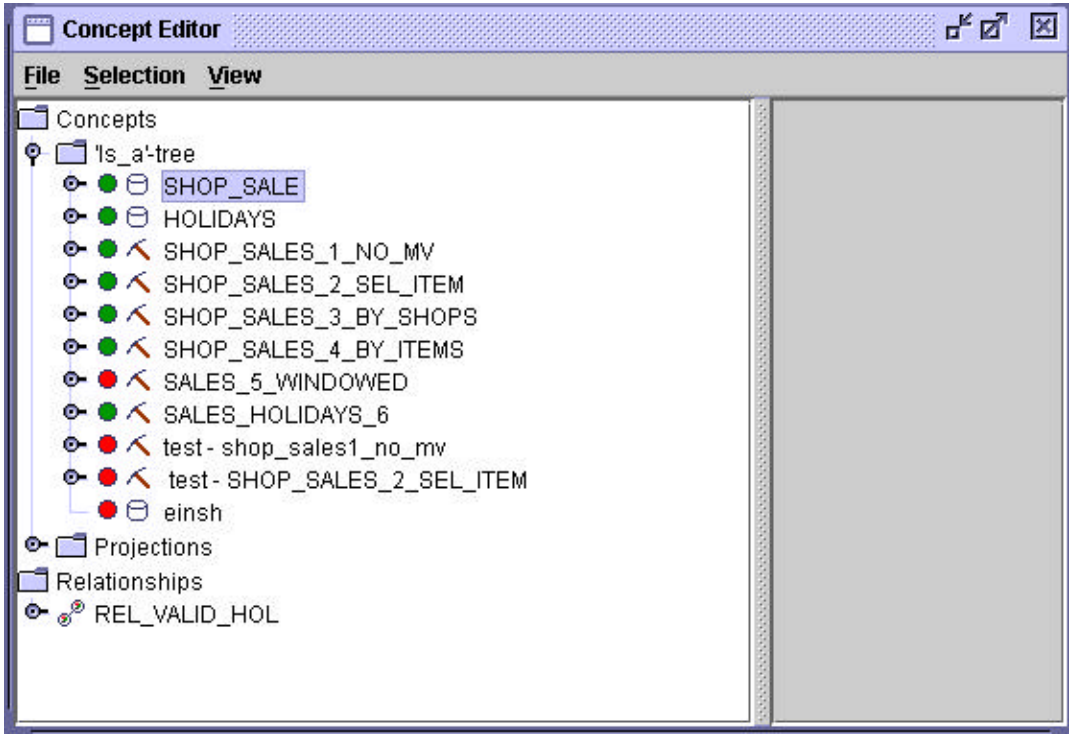
- Unterstützung der Benutzer bei der Vorverarbeitung
  - Erfolgreiche Fälle als Vorlage in Form von Metadaten
  - Anwendbarkeitsbedingungen bei Operatoren
- Offener Ansatz
  - Fälle mit Beschreibung im Internet
  - Erweiterbarkeit durch Einbinden neuer Operatoren
- Publikation der Metadaten – Geheimhaltung der relationalen Geschäftsdaten (incl. DB-Schema)

MINING MART

Meta Modell für Metadaten

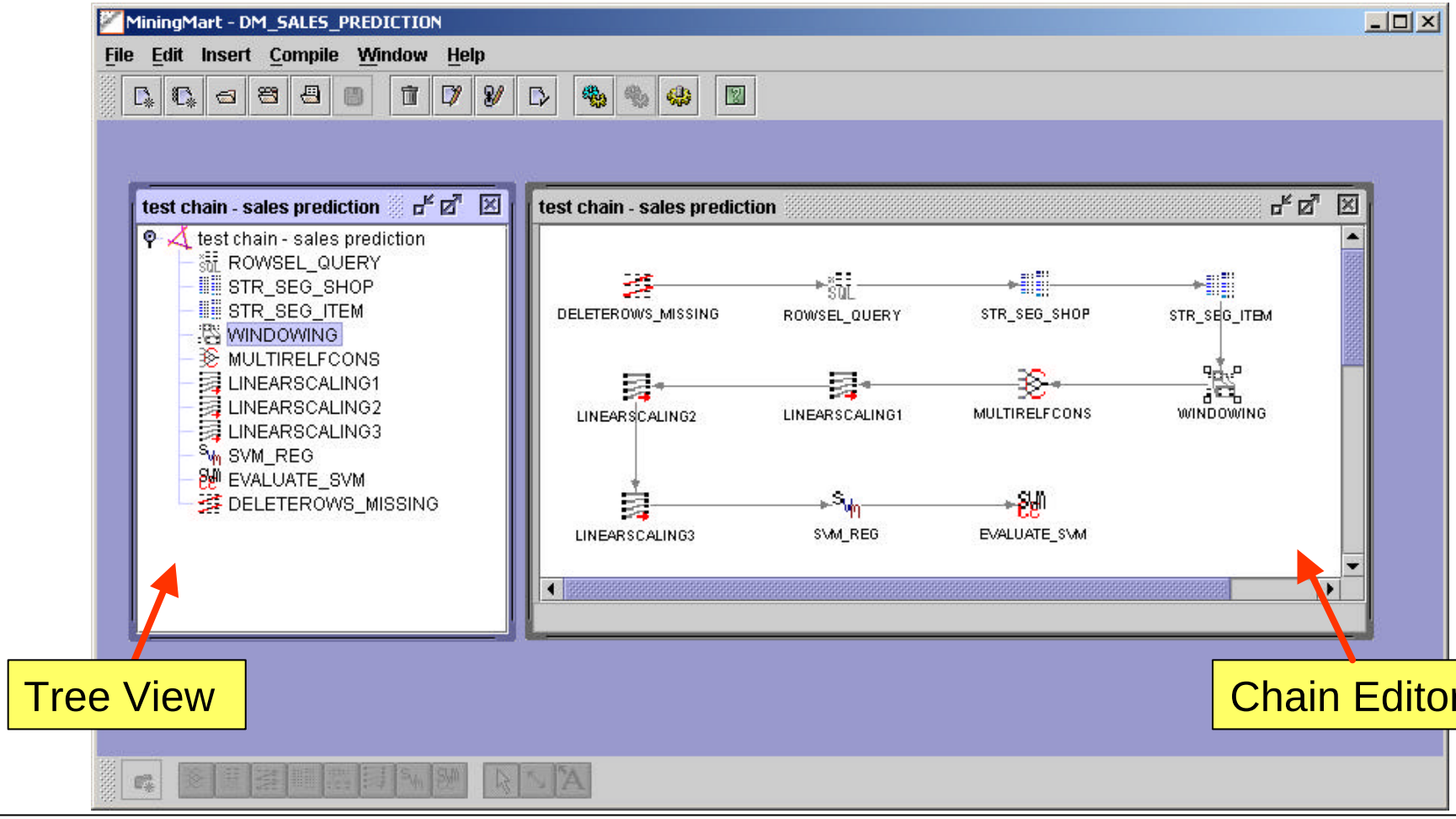
Das begriffliche Modell beschreibt die Objekte und Klassen der Anwendung	Das Fallmodell beschreibt Operator-ketten
Das relationale Modell bescheibt die Datenbank	Das Ausführungsmodell generiert SQL statements oder Aufrufe externer Verfahren

The Concept Editor

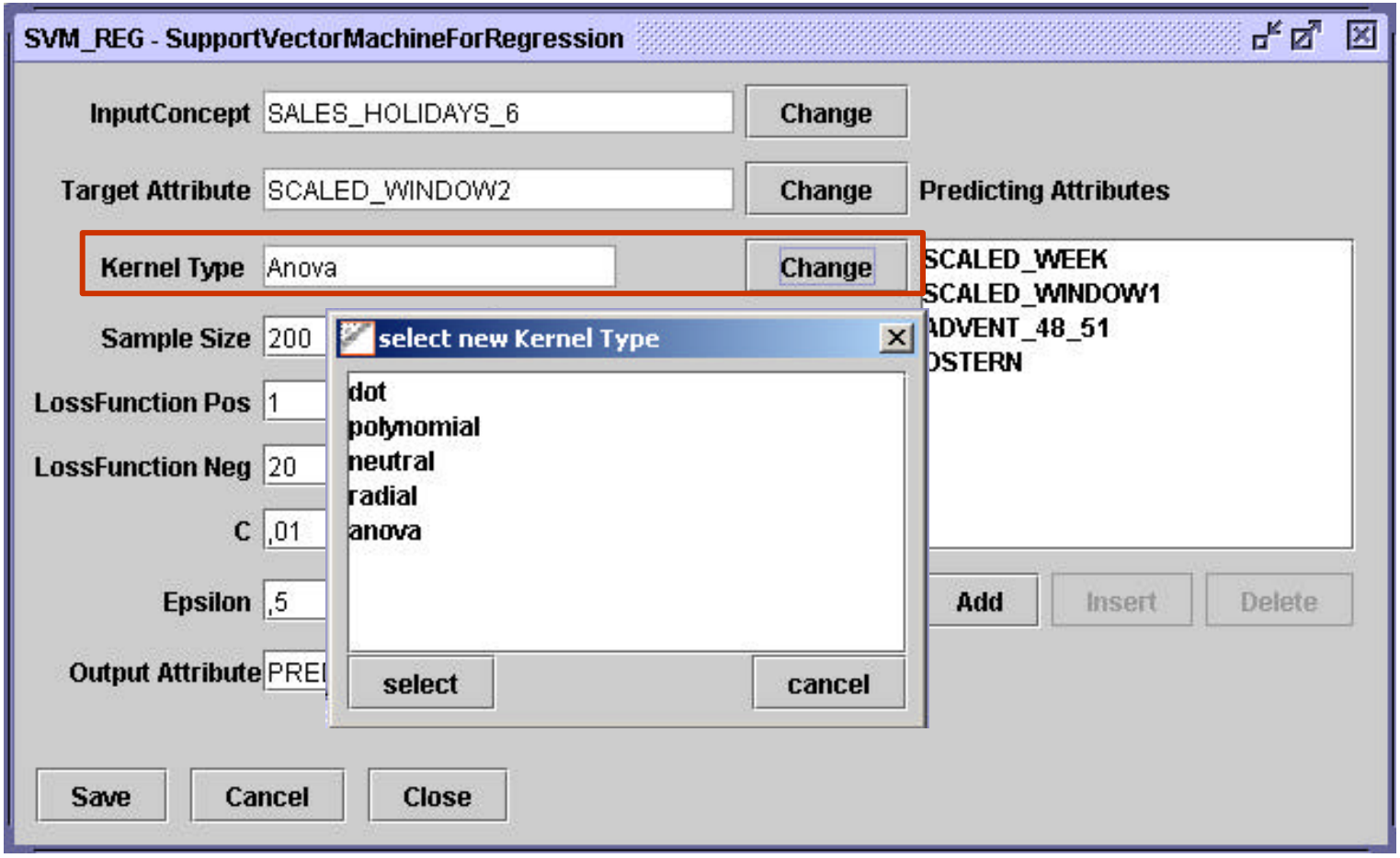


- Definieren und editieren von Begriffen und Relationen
- Abbildung von Begriffen und Relationen auf die Datenbank

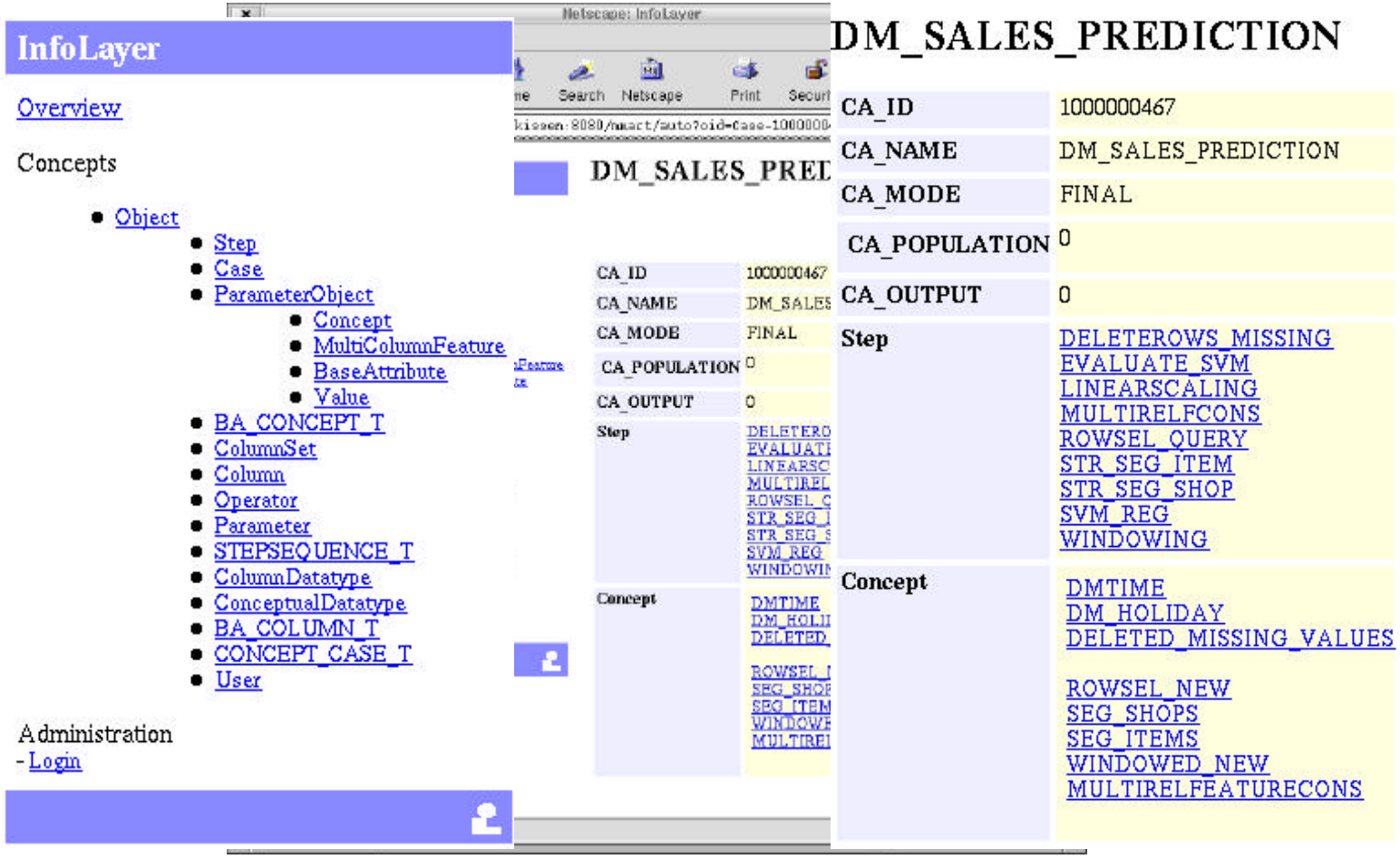
The Case Editor



Setting up an SVM Step



The Internet Case Base



Benutzer des Metamodells

- Der Datenbankadministrator liefert das relationale Modell.
- Der Anwender liefert das begriffliche Modell.
- Die Datenanalyseexpertin liefert das Fallmodell oder passt es an.
  Die ersten Fälle werden vom Mining Mart Projekt erstellt.
- Das MiningMart Projekt entwickelt Operatoren bzw. zu vorhandenen deren Metadaten.

MINING MART

Abstraktionsebenen

- Formalismus zur Repräsentation von Sachbereichsbegriffen, Relationen und KDD-Prozess
  - Was ist ein Begriff, eine Relation, ein KDD-Prozess, ein Operator
- Sprache zur Repräsentation eines bestimmten Sachbereichs, bestimmter Relationen und eines bestimmten Falles
  - Welche Begriffe, Relationen habe ich in einem bestimmten Fall?
- Implementierung des Formalismus und der Sprache als Datenbanktabellen, über denen der Compiler arbeitet und auf die die GUI zugreifen.

Wir sind keine Benutzer!

- Wie sieht das Metamodell aus?
  Welche Repräsentation braucht man, um
  - den allgemeinen KDD-Prozess,
  - die Operatoren mit ihren Parametern, Bedingungen und Zusicherungen,
  - Sachbereiche (Begriffe)

  M4
  - einen bestimmten Fall mit seinen
  - bestimmten Parametersetzungen
  - und einem bestimmten Sachbereich zu beschreiben?

  Instanz
- Wie realisiert man das Metamodell?

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
- UserInput
- 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

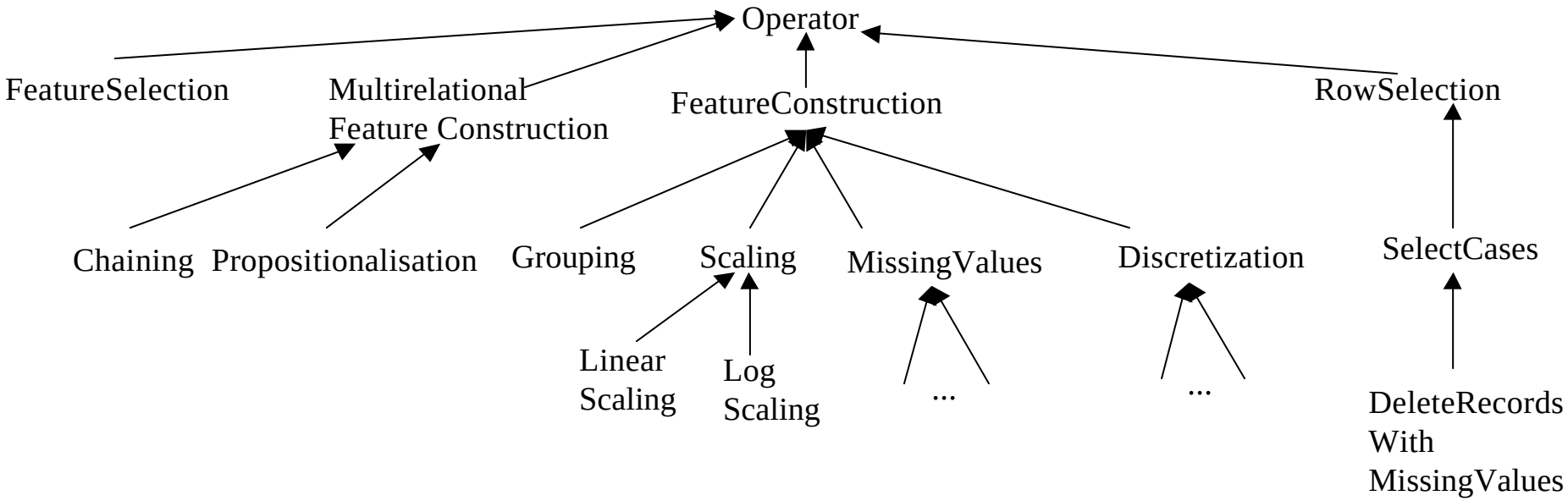
Formalismus zur Beschreibung von KDD-Prozessen

- 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

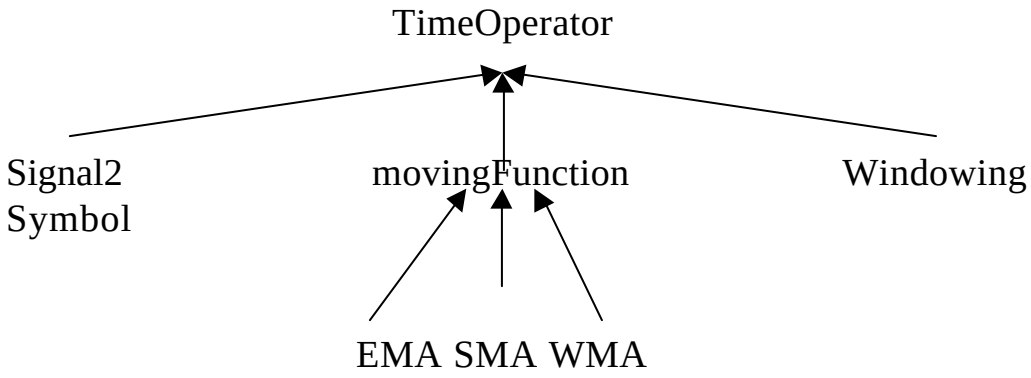
Case Model cont'd

- 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

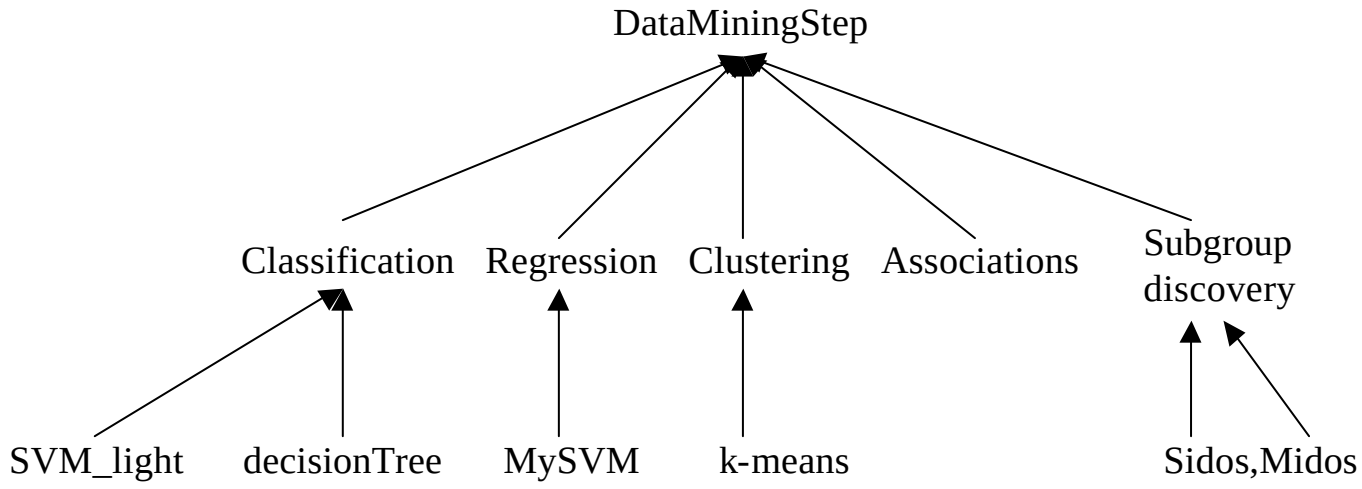
Operatoren des Metamodells



Time Operators in M4

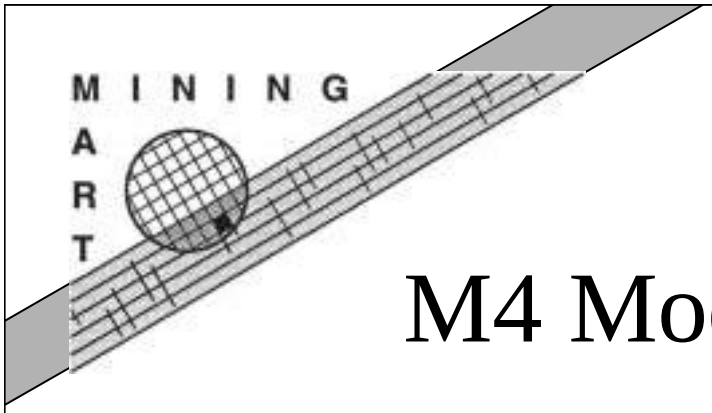


Lernoperatoren des Metamodells



New

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 werden.
- 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

- Sowohl M4 als auch ein bestimmter Fall wird in Datenbanktabellen gespeichert.
- Die M4-Tabellen sind Teil des Systems. Sie steuern den Compiler und an sie schließen die GUI an.
- Die Tabellen des konkreten Falles werden jeweils per GUI oder per Datenbankzugriff gefüllt.

M4 -- Base Attribute

- Basisattribute entsprechen auf der begrifflichen Ebene einer Spalte der relationalen Ebene.
- Tabelle BASATTRIB_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 éntsprechenden 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)

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

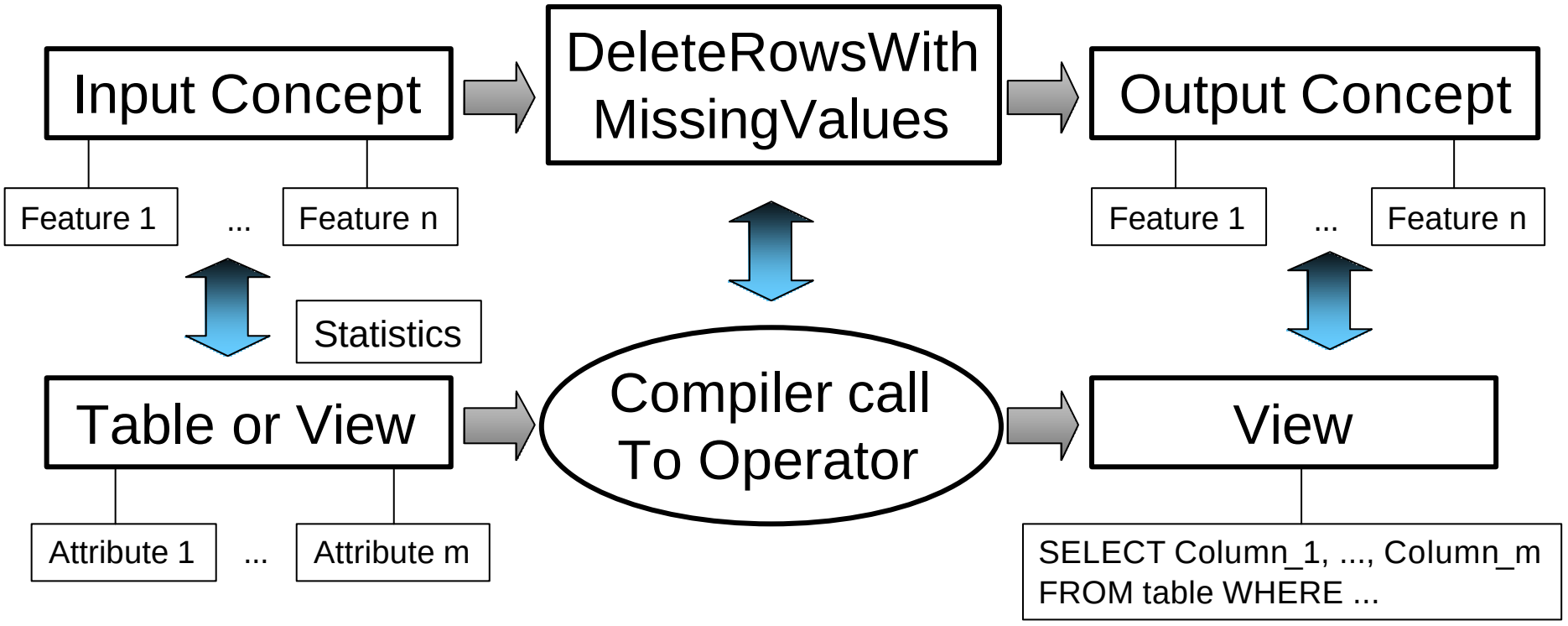
PAR_ID	PAR_OBJ_ID	PAR_OBJ_TYPE	PAR_OP_ID	PAR_ST_ID	...
	Foreign link to *	{BA, MCF, CON,..}	Foreign link to OPERATOR_T	Foreign link to STEP_T	

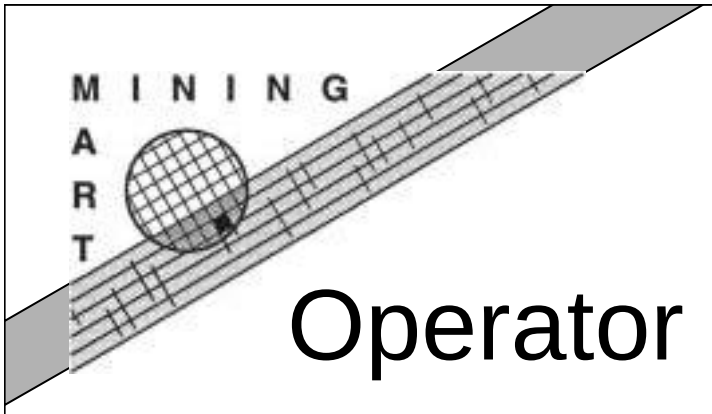
* ∈ BASEATTRIB_T
MCFEATURE_T
VALUE_T
CONCEPT_T
RELATION_T

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

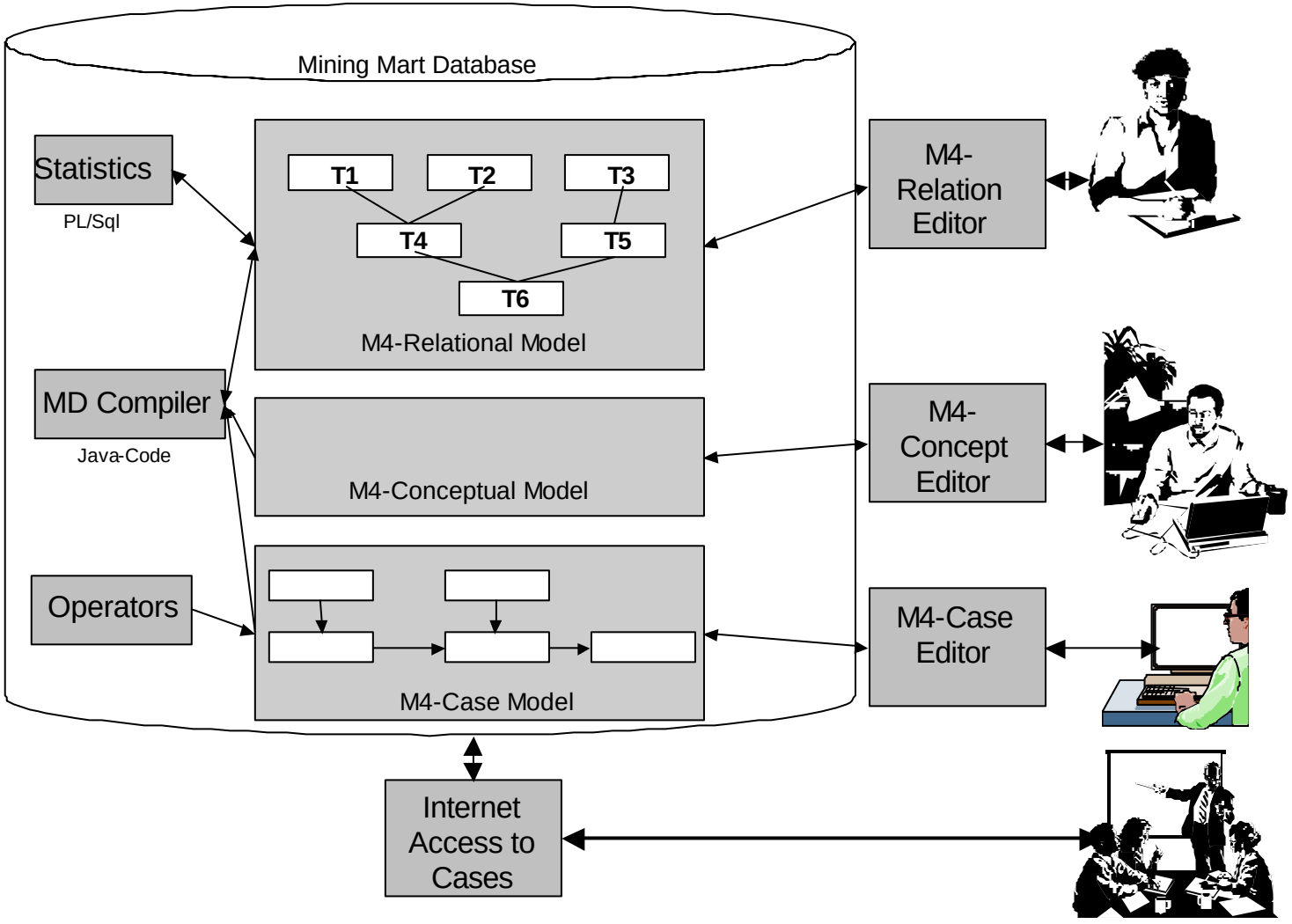


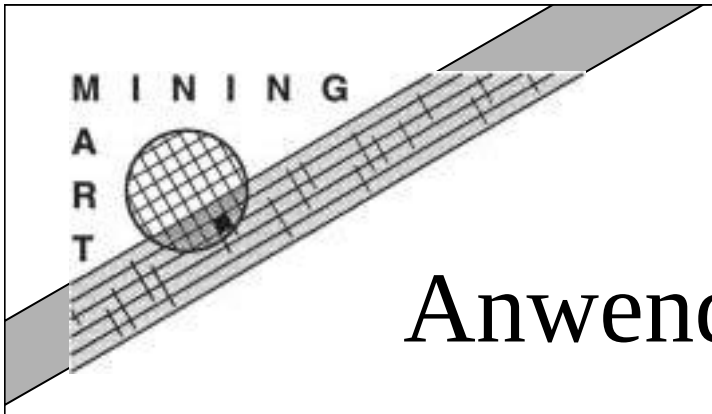


Operator Applicability Information

- Case Editor
  - preparing output for operator application
  - property window for entering parameter values
  - ensures applicability of operator chains
- M4 Compiler
  - checks if inputs/parameters are available and valid
  - required data properties for applicability at runtime
  - assertions: avoid unnecessary database scans

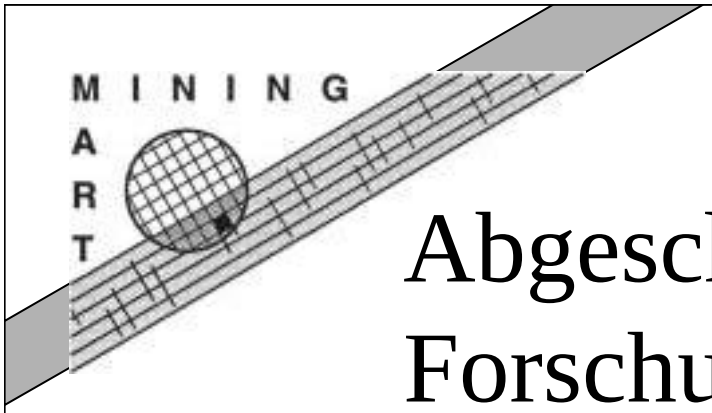
Systemarchitektur





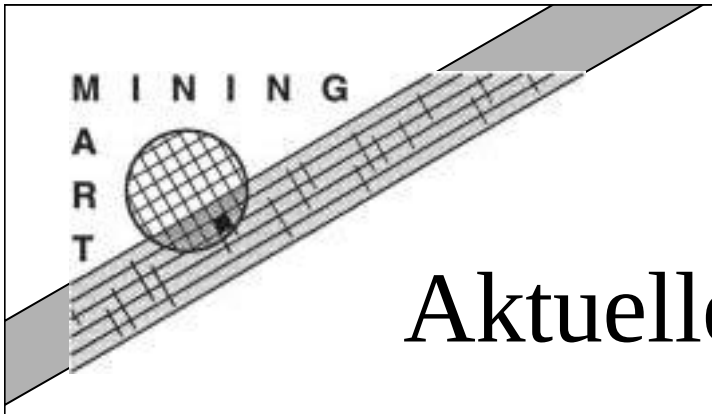
Anwendungen

- 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