



Die PDM-Bibliothek: Embedded Data Mining

Prudential Systems Software GmbH

www.prudsys.de

info@prudsys.de

Telefon: (03 71) 2 70 93-0

Fax: (03 71) 2 70 93-90

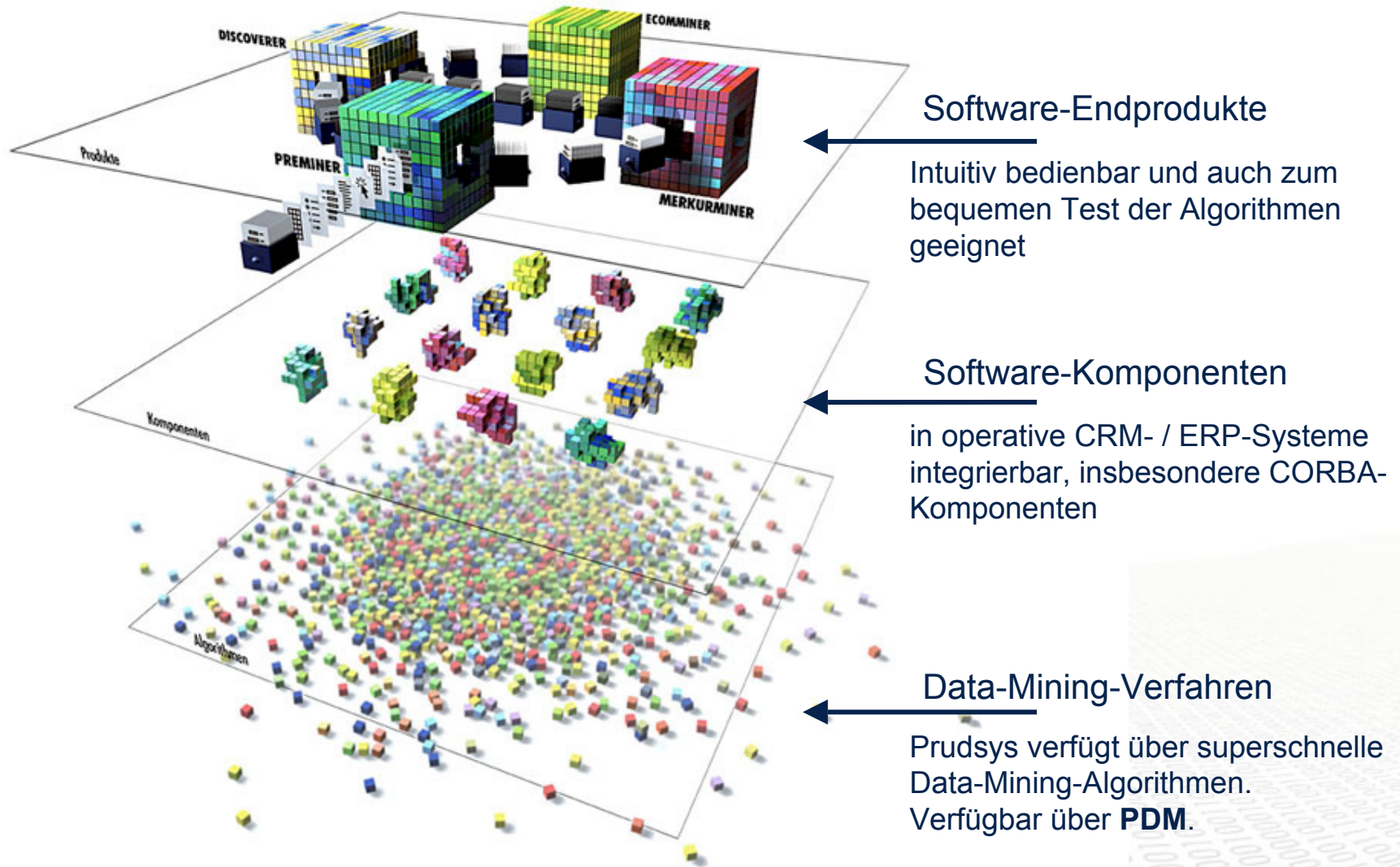
- Aufgabe der PDM
- Umsetzung:
 - CWM for Data Mining
 - Erweiterung für die PDM
 - Implementierung
- Standards
- Nutzung der PDM

- Schaffung einer universellen Data-Mining-Bibliothek

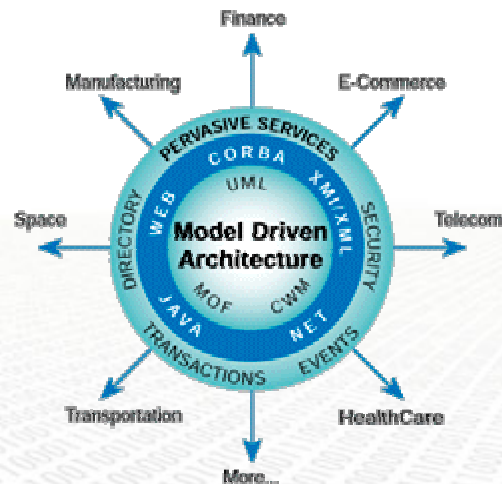
- Ziele:
 - Schaffung einheitlicher Bibliothek für weltweit einzigartige prudsys-Algorithmen
 - Nutzung, um Data-Mining-Modelle in Fremdsoftware anzuwenden
 - Bibliothek als OEM-Lösung in Fremdsoftware integrieren

- Schwerpunkte:
 - Unabhängigkeit von Programmiersprachen und Interfaces
 - Unabhängigkeit von Datenquellen
 - Unterstützung der wesentlichen Data-Mining-Standards
 - Automatisierung der Data-Mining-Verfahren

I Aufgabe: Schaffung eines Baukastensystems



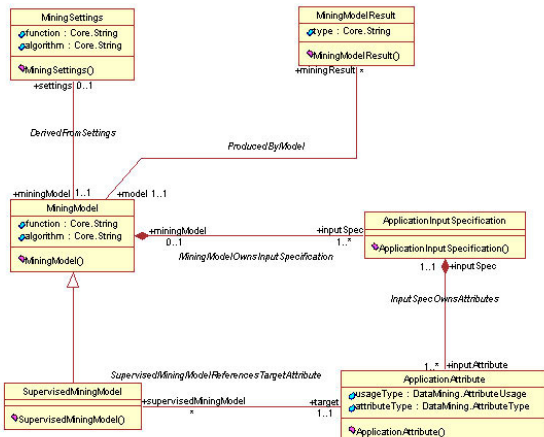
- Unabhängigkeit von Programmiersprachen und Schnittstellen:
 - MDA (Model Driven Architecture)
- Modellierung des Kerns in UML durch Erweiterung des CWM (Common Warehouse Metamodel) for Data Mining (MOF)
- Nutzung der Data-Mining-Modellsprache PMML
- In Zusammenarbeit mit russischem MDA-Spezialisten ZSoft



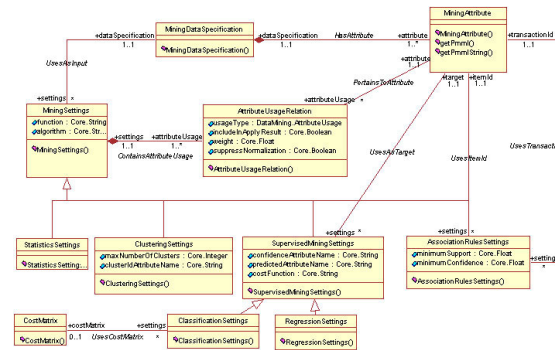
II.1 CWM for Data Mining

- Beschreibung von Data-Mining-Modellen in UML
- 3 UML-Diagramme:
 - Modelle
 - Settings
 - Attribute
- UML-Diagramme:

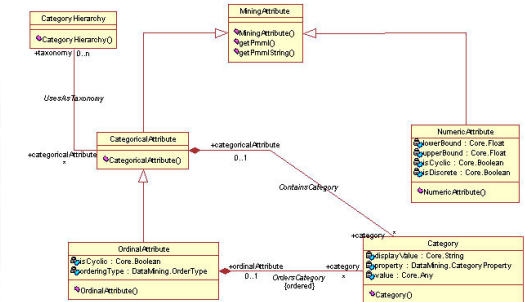
Modell



Settings

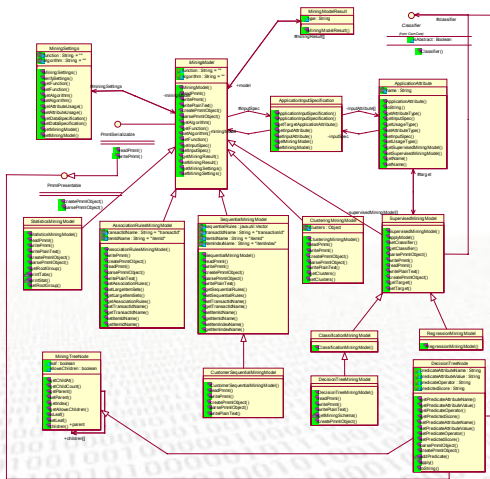


Attribute

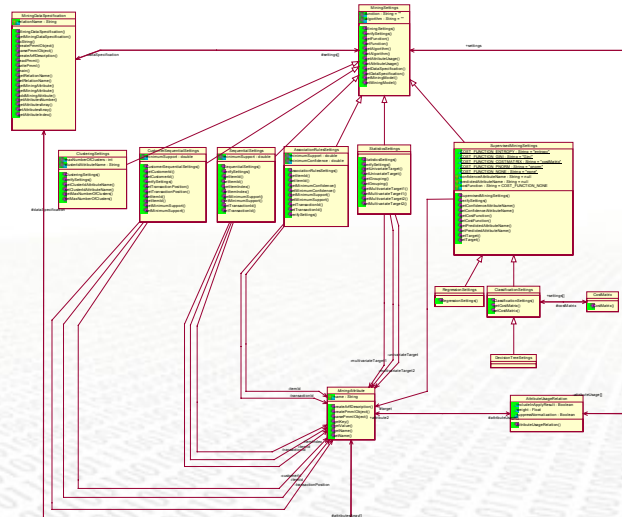


- Erweiterung der CWM-Diagramme um algorithmenspezifische Klassen und PMML-Handling
- Erweiterte UML-Diagramme:

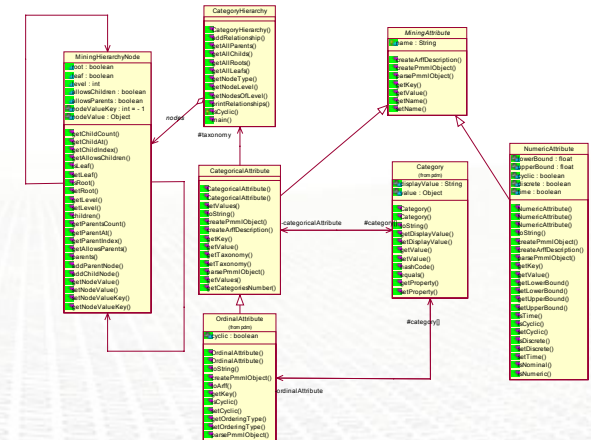
Modelle



Settings



Attribute

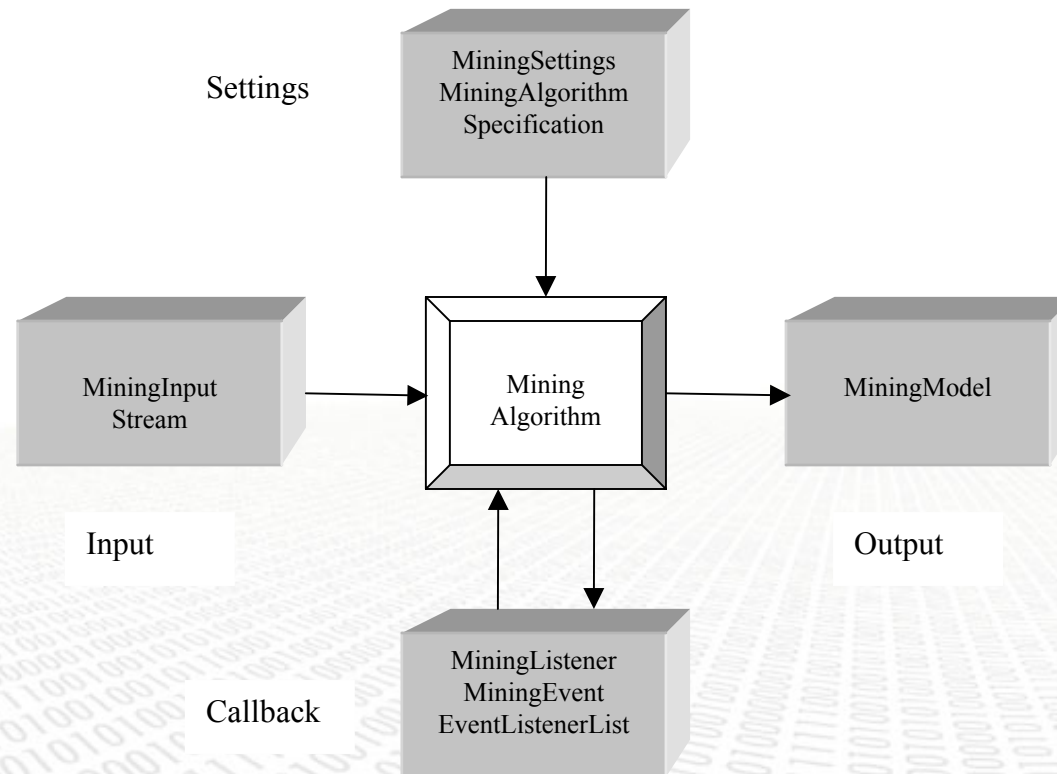


II.2 Erweiterung für die PDM: Data Access

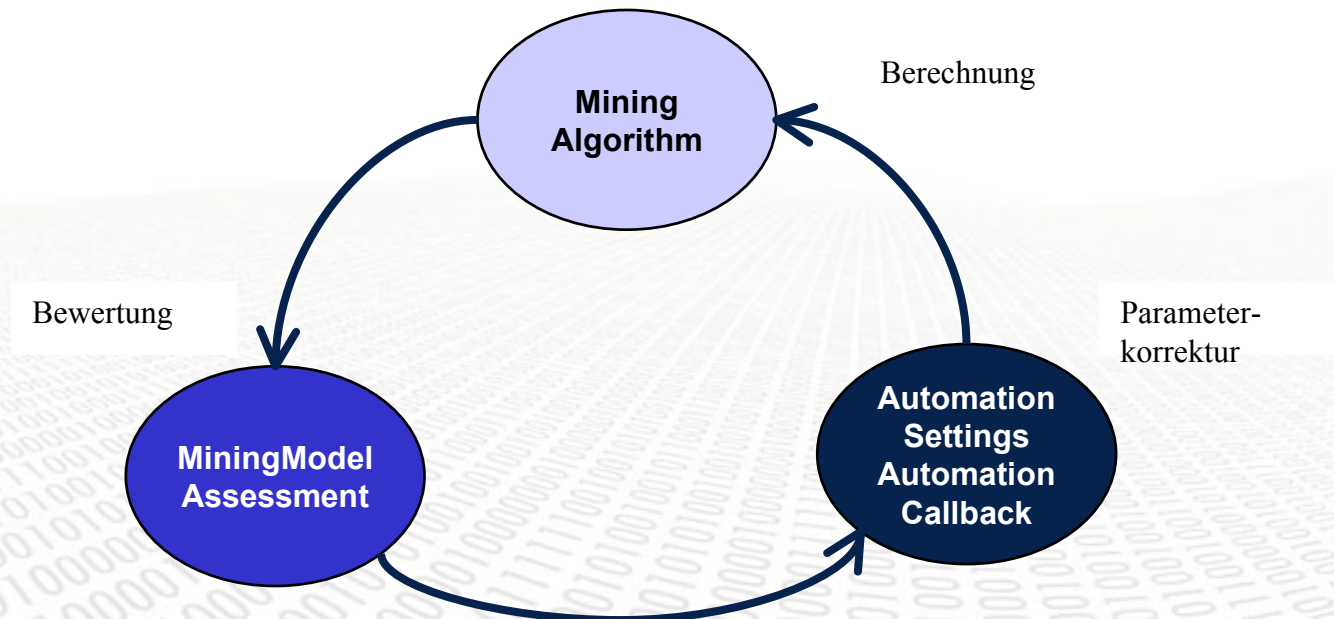
- Modelliert allgemeinen Datenzugriff für Data-Mining-Algorithmen
 - Grundidee: Data-Mining-Algorithmen nutzen 1 Matrix (eventuell sparse)
 - Wird durch Klasse *MiningInputStream* beschrieben:
 - Cursor-basiert, wahlfreier Zugriff, blockweiser Zugriff
 - Nicht alle Zugriffsarten müssen unterstützt werden
 - Erweiterungen von *MiningInputStream* für verschiedene Datenquellen:
 - *MiningArrayStream*: Daten im Speicher
 - *MiningFileStream*: Zugriff auf Dateien
 - *MiningSqlStream*: Zugriff auf relationale Datenbanken
 - *MiningFilterStream*: Transformationen für *MiningInputStream*
 - Nutzer der PDM kann eigene Erweiterungen von *MiningInputStream* schreiben
- ⇒ *Unabhängigkeit der PDM von Datenquellen*
- Vektoren werden durch *MiningVector* beschrieben, Erweiterungen:
 - *MiningSparseVector*: dünnbesetzte Matrizen
 - *MiningBinarySparseVector*: dünnbesetzte Binär-Matrizen

II.2 Erweiterung für die PDM: Algorithms

- Modellierung von Data-Mining-Algorithmen
- Zentrales Element: Klasse *MiningAlgorithm*
 - Input, Output, Settings, Callback (Listener-Konzept wie in Java)

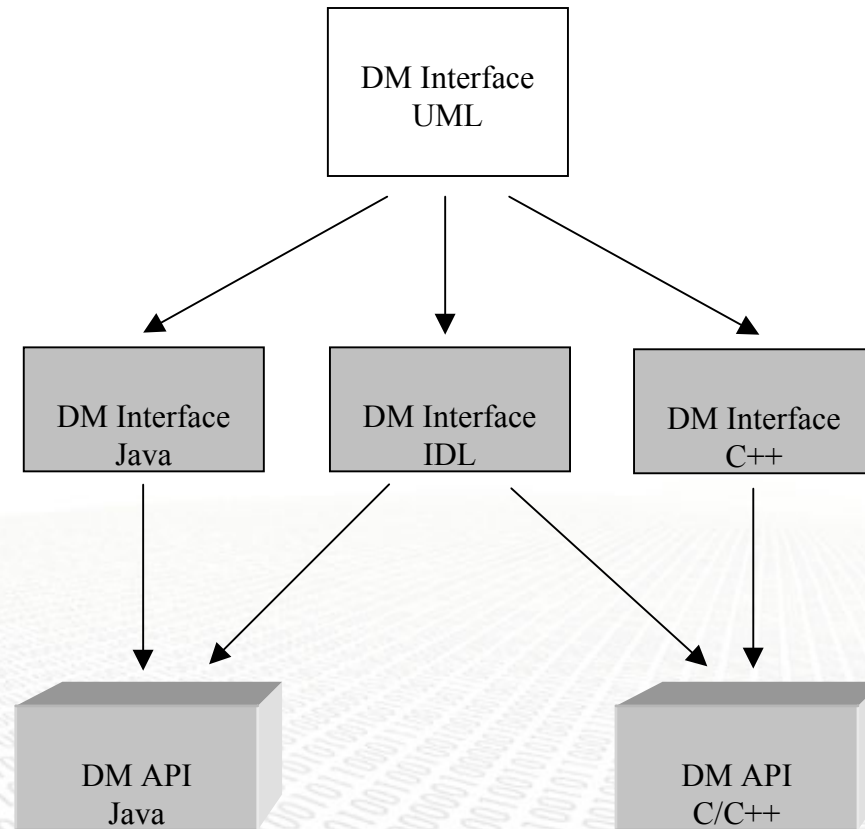


- Erweiterte Algorithmenklassen:
 - *AssociationRulesMiningAlgorithm*: Warenkorbanalyse
 - *ClassificationMiningAlgorithm*: Klassifikationsalgorithmen
 - *ClusteringAlgorithm*: Clusteralgorithmus
- Von diesen werden spezielle Algorithmen abgeleitet
- Automatisierung:
 - Da alle Data-Mining-Komponenten modelliert sind problemlos möglich



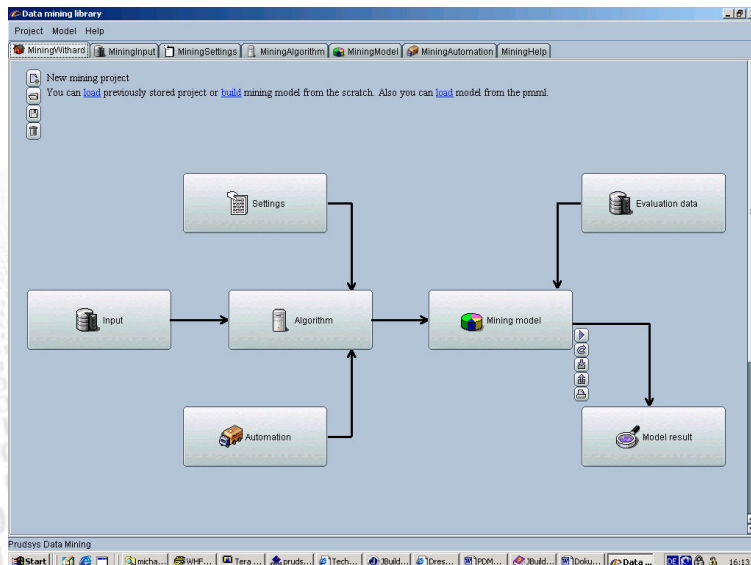
- Im Ergebnis 5 UML-Diagramme:
 - *Model*: Data-Mining-Model
 - *Settings*: Algorithmenparameter
 - *Attribute*: Attribute
 - *DataAccess*: Datenzugriff
 - *Algorithms*: Algorithmen, Automatisierung
- Alle in einem großen UML-Diagramm vereint
- Daraus können Implementierungen abgeleitet werden

- Vorgehensweise:
 - Implementierung in Java, C++, IDL-Interface

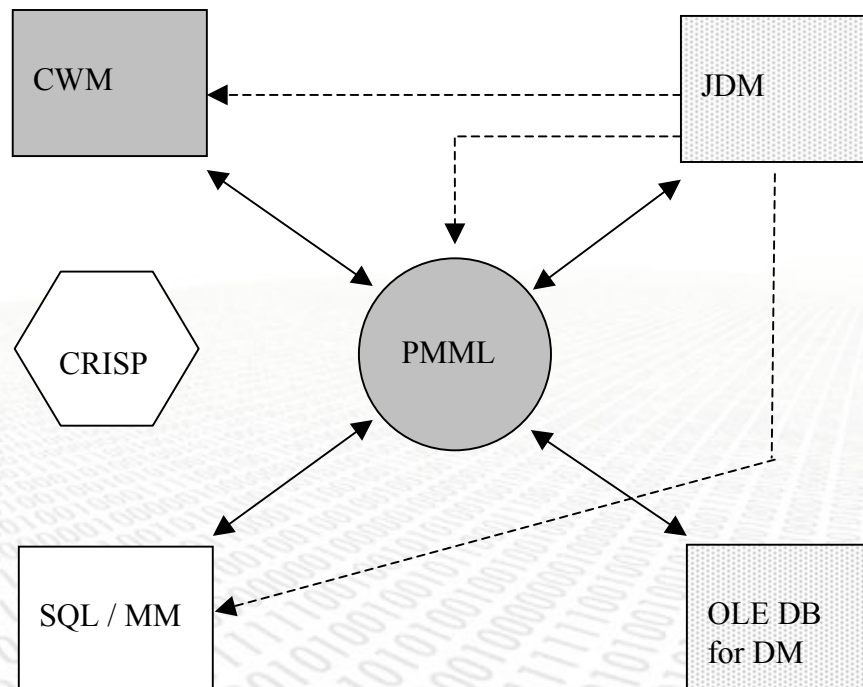


II.3 Implementierung: Java

- Vorgehensweise:
 - UML-Diagramme aus RationalRose in Java-Interfaces konvertiert
 - Zugehöriger Core in Java implementiert
 - Java-Algorithmen in PDM integriert
- Bemerkungen:
 - Alle Java-Klassen als Java Beans implementiert
 - XML-Handling über Data Binding
 - Grafisches Interface zur Demonstration der Bibliothek:



- Derzeit folgende Standards bekannt oder in Ausarbeitung:
 - CWM: UML-basiert für Austausch von Modellen
 - JDM: Java Data Mining Standard
 - OLE DB for DM, SQL / MM: Datenbankbasierte DM Standards
 - PMML: XML-basiert für Austausch von Modellen



- PDM soll alle Standards unterstützen
- Derzeit:
 - CWM nach Konstruktion
 - PMML nach Konstruktion
 - OLE DB for Data Mining über Server
 - (WEKA: populäre Data-Mining-Bibliothek im Hochschulbereich)
- Nach deren Verabschiedung:
 - JDM: basiert ebenfalls auf CWM for Data Mining
 - SQL / MM: ähnlich zu OLE DB for DM

- prudsys-Produkte auf Basis der PDM
- Ergänzung zu prudsys-Produkten für Integration der Data-Mining-Modelle
- Mitte August: Offenlegung für Drittanbieter
- Weitere Entwicklungen:
 - Unterstützung der .NET – Plattform (C# Implementierung)
 - SQL / MM Standard