

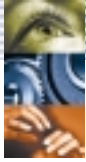
# Mandarax

Ein OpenSource Projekt zur Unterstützung von  
regelbasierten Anwendungen

Java Forum Stuttgart 2002

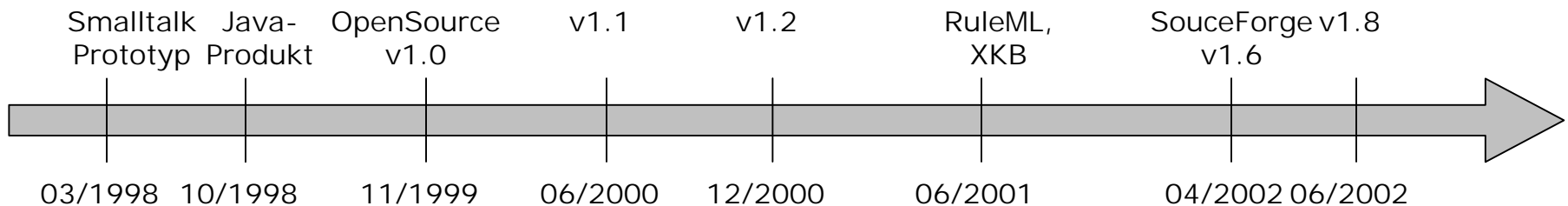
27.06.2002

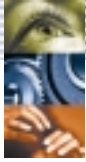
Jochen Hiller, [jochen.hiller@bauer-partner.com](mailto:jochen.hiller@bauer-partner.com)



## Überblick

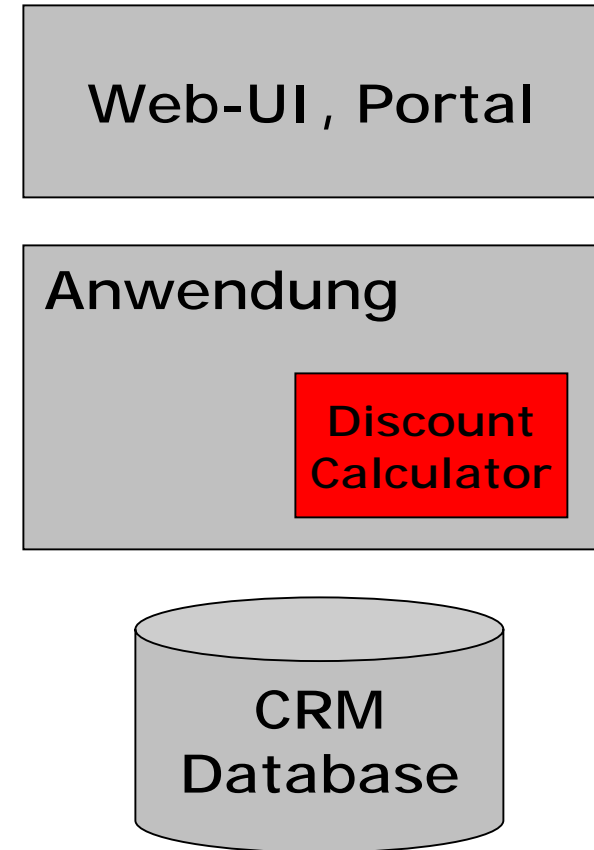
- Projekt-Verantwortliche
  - Jens Dietrich, Polytechnic of Namibia
  - Jochen Hiller, Bauer & Partner
- 1998: Erster Kunde als kommerzielles Produkt
  - DaimlerChrysler USA, Sales Forecasting
- 1999: OpenSource, LGPL-Lizenz
- Einsatz heute: USA, China, Europa, Versicherungen
- <http://www.mandarax.org>,  
<http://sourceforge.net/projects/mandarax>

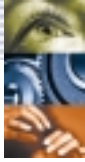




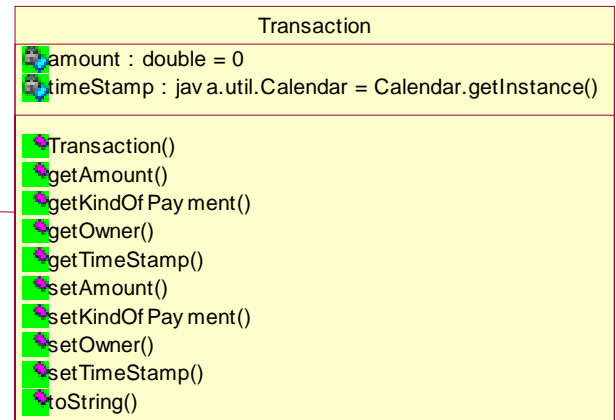
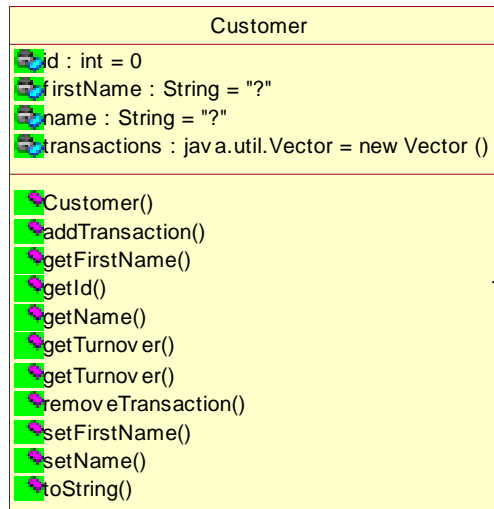
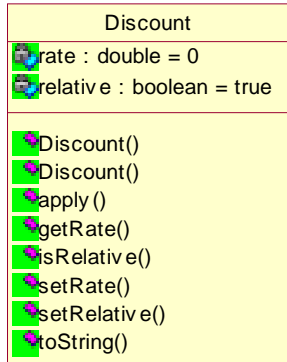
## Ein Beispiel: CRM - Discount Calculator

- Existierende Java Anwendung
  - Existierende Datenhaltung
  - Häufige Änderungen an den Regeln zur Preisfindung
    - Kundenspezifisch
    - Marketing-Kampagnen
    - Bundling
    - Cross-Selling
- ➔ ***Komplexe, änderbare Regeln***



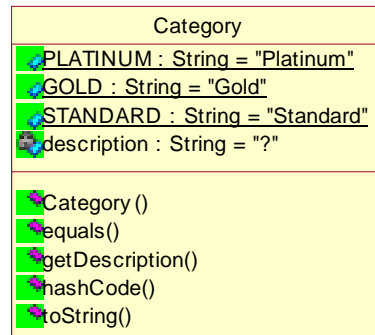


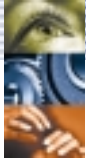
# Customer Relationship Management – Business Model



-owner

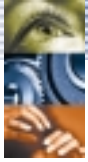
\* kindOf Payment





## Logical Objects – Rules, Facts, Predicates, Terms

**If the turnover of a customer was more than 100\$ in the year 2002 then the customer gets a discount of 5%**



## Logical Objects – Rules, Facts, Predicates, Terms

### Rule

If

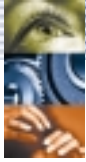
Body

the turnover of a customer was more than  
100\$ in the year 2002

then

the customer gets a discount of 5%

Head



## Logical Objects – Rules, Facts, Predicates, Terms

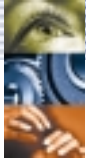
If **Fact**  
the turnover of a customer was more than  
100\$ in the year 2002

**Predicate** more than **Complex Term**  
the turnover of a customer in the  
year 2002

the turnover  
a customer **Variable Term**  
in the year 2002

100\$  
**Constant Term**

...

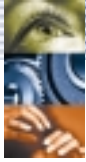


## Logical Objects - Clause Sets

- Logische Programmierung: Rules, Facts → **Clause**
- **ClauseSet** ist eine Menge von **Clause** → Basis der **KnowledgeBase**
- **ClauseSet** kann zur Laufzeit (Query-Time) evaluiert werden
- Über ein **ClauseSet** kann iteriert werden:  
**ClauseSetIterator clauses ();**

Nutzbar für:

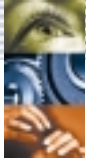
- Proxy-Verhalten gegenüber vorhandenen „Faktenbasen“
  - Java Objekte, SQL-Datenbank, EJB, proprietäre
- AutoFacts: Dynamische Generierung von Faktenbasen
- Verteilte Wissensbasen: Zugriff auf andere Wissensbasen
- Kollaborative Agenten: Anfragen an Agenten



## Maintaining the Knowledge Base

- **KnowledgeBase** enthält Fakten, Regeln (**ClauseSet**)
- Pflege der **KnowledgeBase** (**add**, **remove**, **removeAll**)
- ChangeListener für Änderungen an der **KnowledgeBase**
- **ExtendedKnowledgeBase**, Priorisierung von **ClauseSet**

```
public interface KnowledgeBase {  
    void add(ClauseSet c);  
    void remove(ClauseSet c);  
    void removeAll(ClauseSet c);  
    List getClauseSets();  
    void addKnowledgeBaseChangeListener(...);  
    void removeKnowledgeBaseChangeListener(...);  
}
```



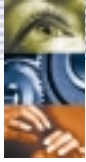
## Issuing Queries

Q: How much discount gets the customer x ?

A: X gets 5% discount.

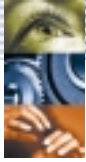
```
public interface InferenceEngine {  
    ResultSet query (Clause query, KnowledgeBase kb,  
        int cardinalityConstraint, int exceptionPolicy)  
}
```

- Eine Query ist ein Fact Objekt mit Variablen
- Queries werden an eine InferenceEngine gestellt
- Ergebnisse werden in Form eines ResultSet geliefert



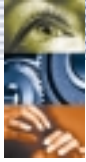
## Inference Engine - Backward vs. Forward Reasoning

	Backward Reasoning	Forward Reasoning
Implementations	Prolog, Mandarax	JRules, CLIPS, JESS (Rete Algorithm)
Architecture	Pull model Push through daemon	Push
Distribution	Distributed knowledge bases, Agents	Difficult
Timeliness	Dynamic ClauseSets	Propagate changes, Observables
Knowledge base size	>Facts (e.g. direct in database) <Rules	>Facts >Rules
Performance	Comparable for small set of rules	Good, proof tree in memory
Arbitrary sources	Extendable ClauseSet	Additional software required (database ?)



## Integrating Databases

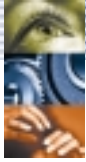
- Ab Mandarax v1.6: SQL Unterstützung von ClauseSet
- **SQLPredicate**
  - DataSource: **javax.sql.DataSource**
  - Name
  - Query (z.B. „SELECT \* FROM customer\_transaction“)
  - TypeMapping (SQL → Java Mapping)
- **SQLClauseSet**: **SQLPredicate** + WHERE-Clause
  - Caching einstellbar (NO, YES, Timeout)
- **SQLFunction**
  - Gibt einen Wert zurück, damit Logik in der Datenbank abbildbar



## Integrating Databases

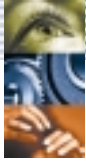
### Code Snippet

```
DataSource dataSource = ...;  
SQLFunction function = new SQLFunction ();  
function.setDataSource (dataSource);  
function.setQuery („SELECT SUM(amount) FROM  
customer_transactions WHERE customer=?“);  
function.setObjectRelationalMapping (new  
OneColumnMapping (Double.class));  
function.setName („turnover of all transactions of a  
customer“);
```



# Integrating the Java Object Model

- Mandarax kann auf beliebige Java Objekte zugreifen
  - Einsatz von Reflection (`org.mandarax.kernel.meta`)
- **JFunction**
  - Wrapper auf Java Methoden
  - `String.indexOf (txt) →`  
`new JFunction (String.class.getMethod („indexOf“, new Class[]`  
`{ String.class }, „indexOf“)`
- **JPredicate**
  - Wrapper auf Methoden, die boolsche Werte zurückliefern
  - `String.equals (other) →`  
`new JPredicate (String.class.getMethod („equals“, new Class[]`  
`{ String.class }, „equals“)`
- **AutoFact**
  - Liste von Predicates, Liste von Typen (Java Klassen), Liste von Objekt-Instanzen
  - Kreuzprodukt: Wo Predicate anwendbar, werden Clauses erzeugt



## Mandarax Libraries

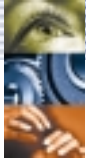
Vorgefertigte Libraries für Standard-Funktionalitäten

`org.mandarax.lib.math.IntArithmetic`

`org.mandarax.lib.math.DoubleArithmetic`

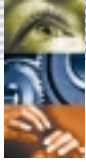
`org.mandarax.lib.text.StringArithmetic`

`org.mandarax.lib.date.DateArithmetic`



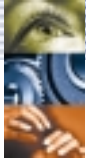
## The XML Interface

- Package: XML Knowledge Base (XKB)
    - XKBManager: Generischer Manager für beliebiger Driver
    - XKBDriver: Manager plugin-Komponente
- ```
org.jdom.Document importKnowledgeBase ();  
void exportKnowledgeBase (org.jdom.Document doc);
```
- RuleML v0.8
    - Built-In Support, enge Abstimmung mit RuleML Group (Dagstuhl Seminar)
  - Mandarax Standard-Treiber v1.0



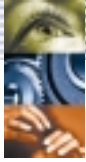
## Ausblick – zukünftige Features ?

- Z.Zt. stärkere Projektunterstützung
- Ausbau SourceForge als Arbeitsplattform
- Integration in Bauer & Partner Wissensmanagement Plattform Top-Logic™
- Integration von EJB Faktenbasen
- Verteilte Wissensbasen (RemoteClauseSets)
- Integrationsfähig in J2EE Umgebungen, Unterstützung JSR-094 (Rule-Engine-API)
- Event-Condition-Action (ECA) Unterstützung
- Integration in Agentenumgebung
- Mandarax als deduktive Datenbank (JDBC compliant ?)



## ORYX – The productivity pack for Mandarax

- 04/2002: v2.0, basiert auf Mandarax v1.6
- Repository Ansatz
- Interaktive Bearbeitung der KnowledgeBase
- Div. Editoren, erweiterbar
- Zukünftig: Web-basierte Benutzer-Oberfläche
  
- Frei für universitäre Zwecke
- Kommerzielle Versionen erhältlich
- Siehe <http://www.jbdietrich.de>



## Referenzen

- Integrating RuleML and OO, Dagstuhl Seminar 02061, 02/2002, Jens Dietrich, <http://www.mandarax.org/docs/dagstuhl.pdf>
- Mandarax Manual v1.8, Jens Dietrich, [http://www.mandarax.org/docs/mandarax\\_1\\_8.pdf](http://www.mandarax.org/docs/mandarax_1_8.pdf)
- RuleML, Rule Markup Language, <http://www.dfki.uni-kl.de/ruleml/>
- JESS, Java Expert System Shell, Ernest J. Friedman-Hill, <http://herzberg.ca.sandia.gov/jess>
- Mandarax – Ein OpenSource Ansatz zur Verwaltung und Verarbeitung von Geschäftsregeln, NetObjectDays 2001, Jochen Hiller, [http://www.netobjectdays.org/pdf/01/papers/ind\\_invited/hiller\\_dietrich.pdf](http://www.netobjectdays.org/pdf/01/papers/ind_invited/hiller_dietrich.pdf)
- Agententechnologien in Java, Jochen Hiller, Java Forum Stuttgart 2001, <http://www.jfs2001.de/folien/E2.ppt>