Probabilistic and Possibilistic Graphical Models in Complex Industrial Applications

Prof. Dr. Rudolf Kruse, PD Dr. Jörg Gebhardt
University of Magdeburg

kruse@iws.cs.uni-magdeburg.de
Department of Computer Science in Magdeburg

- 20 full professors
- 60 research assistents
- 50 research assistents in projects
- 1200 students

- Standard Bachelor and Master Courses
  in German Language

- Master Courses in English:
  Data and Knowledge Engineering,
  Computational Visualistics,

http://www.cs.uni-magdeburg.de
Main Research Topic: Intelligent Data Analysis

- Intelligent Data Analysis with Different Methods such as Neuronal Networks, Fuzzy Systems, Evolutionary Algorithms und related Methods
- Development of Freeware such as NEFCLASS and the „InformationMiner“

Current Industrial Projects with former doctorates
- Item Planning with Markov-Networks (Volkswagen, Jörg Gebhardt)
- Information Mining (BMW, Daimler Chrysler)
- Bayes-Methods in Finance (Several German Banks)
- New Data Analysis Methods in Telecommunications (British Telecom)
## Item Planning at VW

### STRATEGY OF VW GROUP

<table>
<thead>
<tr>
<th>Marketing strategy</th>
<th>prefer individual vehicle specifications by customers</th>
<th>bestseller-oriented vehicle specifications by car maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>very large number of possible variants</td>
<td>low number of possible variants</td>
</tr>
</tbody>
</table>

### Vehicle specification

<table>
<thead>
<tr>
<th>Item family</th>
<th>Item</th>
<th>Item family</th>
<th>Item family</th>
<th>Item family</th>
<th>Item family</th>
<th>Item family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>short back</td>
<td>2.8L 150 kW</td>
<td>Type alpha</td>
<td>4</td>
<td>leather</td>
<td>yes</td>
</tr>
<tr>
<td>body variant</td>
<td>engine</td>
<td>radio</td>
<td>door layouts</td>
<td>seat covering</td>
<td>vanity mirror</td>
<td>......</td>
</tr>
</tbody>
</table>
Example: „Golf“ Class of Vehicles

- approximately 200 item families (variables)
- from 2 to 50 items in each family

- i.e. more than \(2^{200}\) possible vehicle specifications
- choice of valid specifications is restricted by RULE SYSTEMS
  (10,000 technical rules, even more marketing-, and production-oriented)

Example (technical rules that restrict validity of item combinations)

\[
\text{if } \text{engine} = e_1 \quad \text{then} \quad \text{transmission} = t_3
\]

\[
\text{if } \text{engine} = e_4 \quad \text{and} \quad \text{auxiliary heater} = h_1 \quad \text{then} \quad \text{generator} \in \{g_3, g_4, g_5\}
\]

How to predict installation rates of item combinations?
Problem Representation

Historical Data

Sample of produced vehicle specifications

(representative choice, context-dependent, f.e. Golf)

System of rules

Rules for the validity of item combinations

(specified for a vehicle class and a planning interval)

If engine = e1 and auxiliary heater = h2 then generator in \{g3,g4,g5\}

Prediction

Planning

predicted / assigned planning data

(production program, demands, installation rates, capacity restrictions, ... bills of material, ...)

... (Golf, short back, 2.8 L 150 kW spark engine, radio alpha, ...) ...
Result of Problem Analysis

- Handling rules: Modelling Constraints
- Handling historical data: Learning from Data
- Combining the different sources: Fusion of Models
- Supporting planning: Belief Change

These types of problems were treated in the three big EC Projects: DRUMS 1, DRUMS 2 and FUSION

Our recommendation in 2001 was to use “decomposable models”, e.g. Probabilistic Graphical Models, to build a Systems that supports the planners (nonstandard solution).
Problem Representation

Historical Data
Sample of produced vehicle specifications
(representative choice, context-dependent, f.e. Golf)

System of rules
Rules for the validity of item combinations
(specified for a vehicle class and a planning interval)

If engine = e1 and auxiliary heater = h2 then generator in {g3,g4,g5}

Prediction Planning
predicted / assigned planning data
(production program, demands, installation rates, capacity restrictions, ...
bills of material, ...)

... (Golf, short back, 2.8 L 150 kW spark engine, radio alpha, ...)
...
Some Basic Ideas: A Toy Example

Example World

- 10 differently equipped cars, 3 attributes.
- One car is chosen at random and examined.
- Inferences are drawn about the unobserved attributes.

<table>
<thead>
<tr>
<th>color</th>
<th>speed</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>low</td>
<td>compact</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>compact</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>limo</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>limo</td>
</tr>
</tbody>
</table>
## Item Combinations

### Relation

<table>
<thead>
<tr>
<th>color</th>
<th>speed</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>low</td>
<td>compact</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>compact</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>limo</td>
</tr>
<tr>
<td>yellow</td>
<td>medium</td>
<td>coupe</td>
</tr>
<tr>
<td>green</td>
<td>medium</td>
<td>coupe</td>
</tr>
<tr>
<td>green</td>
<td>high</td>
<td>coupe</td>
</tr>
<tr>
<td>green</td>
<td>high</td>
<td>limo</td>
</tr>
</tbody>
</table>

### Geometric Interpretation

Each cube represents one tuple/car.
Projections
Rule Systems: Use Material Implication

Rules:

\[ r_1 : \quad \text{if } A = a_1 \quad \text{then } B \in \{b_3, b_4, b_5\} \]

\[ \wedge r_2 : \quad \text{if } B = b_2 \quad \text{then } A = a_2 \]

Rule scheme: \((A,B)\)

\[
\begin{array}{cccccc}
\text{Dom}(A) & = & \{a_1, a_2, a_3\} & \quad & \text{Dom}(B) & = & \{b_1, b_2, b_3, b_4, b_5\}
\end{array}
\]
Cylindrical Extensions and Their Intersection

Intersecting the cylindrical extensions of the projection to the subspace formed by color and speed and of the projection to the subspace formed by speed and type yields the original three-dimensional relation.
Focussing

Let it be known (e.g. from an observation) that the given car is green. This information considerably reduces the space of possible value combinations.

From the prior knowledge it follows that the given car must be

- either medium or high speed and
- either a coupe or a limo.
Focussing with Projections

The same result can be obtained using only the projections to the subspaces without reconstructing the original three-dimensional space:

This justifies a network representation:
Relational Graphical Model

≡ Decomposition + Local Model

Example

Operations: Focussing, Revision, Updating…
Transformation into Hypertree Structure

Interpretation as a conditional independence graph:

\[(X, Y) \not\in \mathcal{E} \iff X \perp Y \mid V - \{X, Y\}\]
Transformation into Hypertree Structure

Triangulation

Loss of some information:

\[(X, Y) \notin \mathcal{E} \implies X \perp Y \mid V - \{X, Y\}\]

Skeleton of Tree of Cliques
(hypertree structure)
Tree Of Cliques (Pablo’s VW Bora)

186 variables
174 cliques
max. 9 dimensions
Problem Representation

**Historical Data**

Sample of produced vehicle specifications

*(representative choice, context-dependent, f.e. Golf)*

... *(Golf, short back, 2.8 L 150 kW spark engine, radio alpha, ...)*

... 

**System of rules**

Rules for the validity of item combinations

*(specified for a vehicle class and a planning interval)*

If engine = e1 and auxiliary heater = h2 then generator in {g3,g4,g5}

**Prediction**

predicted / assigned planning data

*(production program, demands, installation rates, capacity restrictions, ... bills of material, ...)*
Planning Problem: Prediction of Parts Demand

Variants-related bill of material

Installation condition: disjunction of item combinations

Installation rates at installation point sum up to 1

EXAMPLE: > 100,000 item combinations needed in „Golf“ class
Choice of the Uncertainty Calculus (Installation rates)

- crisp single-valued
- crisp set-valued relational
- uncertain probabilistic random sets
- One-point-coverage
- Approximation by aggregation
- possibilistic
Probabilistic Graphical Model

Probabilistic Graphical Model: Decomposition + Local Models

Decomposition: Hypergraph on Variables

Local Models: Marginal Distributions of A, B and B, C that „fit together“
**Item Combinations**

<table>
<thead>
<tr>
<th>color</th>
<th>speed</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>low</td>
<td>compact</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>compact</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>limo</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>coupe</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>limo</td>
</tr>
</tbody>
</table>

Now rates (probabilities) for the cubes are added – this gives a finite probability space.
A Probability Distribution

The numbers state the probability of the corresponding value combination.
### Reasoning: Computing Conditional Probabilities

Using the information that the given car is green.

![Conditional Probabilities Diagram]

<table>
<thead>
<tr>
<th></th>
<th>limo</th>
<th>coupe</th>
<th>compact</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>medium</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>low</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>hi</th>
<th>me</th>
<th>lo</th>
</tr>
</thead>
<tbody>
<tr>
<td>limo</td>
<td>572</td>
<td>364</td>
<td>64</td>
</tr>
<tr>
<td>coupe</td>
<td>257</td>
<td>242</td>
<td>21</td>
</tr>
<tr>
<td>compact</td>
<td>572</td>
<td>364</td>
<td>64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>cmp</th>
<th>cpe</th>
<th>lim</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>29</td>
<td>257</td>
<td>286</td>
</tr>
<tr>
<td>medium</td>
<td>61</td>
<td>242</td>
<td>61</td>
</tr>
<tr>
<td>low</td>
<td>32</td>
<td>21</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>limo</th>
<th>coupe</th>
<th>compact</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>medium</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>low</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Probabilistic Decomposition

- As for relational networks, the three-dimensional probability distribution can be decomposed into projections to subspaces, namely the marginal distribution on the subspace formed by color and speed and the marginal distribution on the subspace formed by speed and type.

- The original probability distribution can be reconstructed from the marginal distributions using the following formulae \( \forall i, j, k \):

\[
P(a_i^{\text{(color)}}, a_j^{\text{(speed)}}, a_k^{\text{(type)}}) = P(a_i^{\text{(color)}}, a_j^{\text{(speed)}}) \cdot P(a_k^{\text{(type)}} \mid a_j^{\text{(speed)}})
\]

\[
= P(a_i^{\text{(color)}}, a_j^{\text{(speed)}}) \cdot \frac{P(a_j^{\text{(speed)}}, a_k^{\text{(type)}})}{P(a_j^{\text{(speed)}})}
\]

- These equations express the conditional independence of attributes color and type given the attribute speed, since they only hold if \( \forall i, j, k \):

\[
P(a_k^{\text{(type)}} \mid a_j^{\text{(speed)}}) = P(a_k^{\text{(type)}} \mid a_i^{\text{(color)}}, a_j^{\text{(speed)}})
\]
Reasoning in Join/Junction Trees

- Reasoning in join trees follows the same lines as shown in the simple example.
- Multiple pieces of evidence from different branches may be incorporated into a distribution before continuing by summing/marginalizing.

![Diagram showing the process of reasoning in join/junction trees with data tables and calculations.]
Relation to Bayesian Networks

Directed dependency network

Rule → conditional dependency

Hypergraph representation

Rule → constraint
Graphical Model

- **Historical data**
  - context-dependent sample of produced vehicle specifications

- **System of rules**
  - context-dependent rules for the validity of item combinations

- **Decomposition**
  - Learning
    - Probabilistic Graphical Model

- **Composition**
  - Modify representation
    - Relational Graphical Model

- **Fusion**

- **Graphical Model**
  - fused consistent Markov network

Context: vehicle class, planning interval
Direct Test for Decomposability: Relational

1. \text{color} \quad \text{speed} \quad \text{type}

2. \text{color} \quad \text{speed} \quad \text{type}

3. \text{color} \quad \text{speed} \quad \text{type}

4. \text{color} \quad \text{speed} \quad \text{type}

5. \text{color} \quad \text{speed} \quad \text{type}

6. \text{color} \quad \text{speed} \quad \text{type}

7. \text{color} \quad \text{speed} \quad \text{type}

8. \text{color} \quad \text{speed} \quad \text{type}

For bigger nets use e.g. Hartley Information Measure
# Direct Test for Decomposability: Probabilistic

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>color</strong></td>
<td>speed</td>
<td>type</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>color</strong></td>
<td>speed</td>
<td>type</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>color</strong></td>
<td>speed</td>
<td>type</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>color</strong></td>
<td>speed</td>
<td>type</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Upper numbers: The Kullback-Leibler information divergence of the original distribution and its approximation.

Lower numbers: The binary logarithms of the probability of an example database (log-likelihood of data).
# Learning Graphical Models

<table>
<thead>
<tr>
<th>known structure</th>
<th>unknown structure</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://via.placeholder.com/150" alt="Diagram" /></td>
<td><img src="https://via.placeholder.com/150" alt="Diagram" /></td>
</tr>
</tbody>
</table>

### known structure
- A
- B
- C
- \( \langle a_4, \ b_3, \ c_1 \rangle \)
- \( \langle a_3, \ b_2, \ c_4 \rangle \)

### Statistical Parametric Estimation (closed from eq.):**
- statistical parameter fitting,
- ML Estimation,
- Bayesian Inference, ...

### incomplete data
- A
- B
- C
- \( \langle a_4, \ ? , \ c_1 \rangle \)
- \( \langle a_3, \ b_2, \ ? \rangle \)

### Parametric Optimization:*
- EM,
- gradient descent, ...

### discrete optimization over structures (discrete search):*
- likelihood scores,
- MDL

**Problem:**
- search complexity → heuristics

### complete data
- A
- B
- C
- \( \langle a_4, \ b_3, \ c_1 \rangle \)
- \( \langle a_3, \ b_2, \ c_4 \rangle \)

### Discrete Optimization over Structures (discrete search):*
- likelihood scores,
- MDL

**Problem:**
- search complexity → heuristics

### Combined Methods:
- structured EM
- only few approaches

**Problems:**
- criterion for fit?
- new variables?
- local maxima?
- fuzzy values?
Application at the DaimlerChrysler AG

Improvement of Product Quality by Finding Weaknesses

- Learn decision trees or inference network for vehicle properties and faults.
- Look for unusual conditional fault frequencies.
- Find causes for these unusual frequencies.
- Improve construction of vehicle.

Improvement of Error Diagnosis in Garages

- Learn decision trees or inference network for vehicle properties and faults.
- Record properties of new faulty vehicle.
- Test for the most probable faults.
Analysis of Daimler/Chrysler Database

- Database: ~ 18,500 passenger cars
  > 100 attributes per car

- Analysis of dependencies between special equipment and faults.

- Results used as a starting point for technical experts looking for causes.
Analysis of Daimler/Chrysler Database

Fictitious example:
There are significantly more faulty batteries, if both air conditioning and electrical roof top are built into the car.
Example Subnet

Influence of special equipment on battery faults:

<table>
<thead>
<tr>
<th>(fictitious) frequency of</th>
<th>air conditioning</th>
</tr>
</thead>
<tbody>
<tr>
<td>battery faults</td>
<td>with</td>
</tr>
<tr>
<td>electrical sliding roof</td>
<td>with</td>
</tr>
<tr>
<td></td>
<td>without</td>
</tr>
</tbody>
</table>

- significant deviation from independent distribution
- hints to possible causes and improvements
- here: larger battery may be required, if an air conditioning system and an electrical sliding roof are built in

(The dependencies and frequencies of this example are fictitious)
Problems in Structure Learning of PGM

Complexity of learning problem

- Exhaustive graph search in „poor“ classes
- Greedy search (heuristics) in „richer“ classes

Dependency analysis (CI-Tests)

- Handling „soft“ dependencies
- Unsufficient quality of results, need for controllable search strategies

Probability maximization (Bayesian-Dirichlet)

- Integrability of structure knowledge
Information Fusion

- **Historical data**
  - context-dependent sample of produced vehicle specifications

- **System of rules**
  - context-dependent rules for the validity of item combinations

---

- **Estimate prior distribution of installation rates**
- **Use cond. independencies (Composition)**

---

- **Quantitative Learning**
  - PGM (Markov network) having the structure of the relational network

- **Modify representation**
  - Transformation into a relational network with hypertree structure

---

- **context:** vehicle class, planning interval
Planning Models

Typical complexity:

- 200 item families
- 150 cliques
- 5 to 7 dimensions (typical)
- max. dimensions: 11 to 14
- 100 vehicle model groups
- 20 to 40 planning intervals
  (i.e. 2000 to 4000 networks)
Planning Operation : Conditioning (Focussing)

- **Input Data** : item combination (set of variable instantiations)

- **Operation** : Calculate the conditioned network distribution and the probability of the given item combination (propagation).

- **Application** : Calculation of part demands
  Compute the installation rate of item combination \((m_3, t_4, g_5)\).

**Simulation**
Analyze customers' preferences with respect to those persons who buy a navigation system in a VW Polo.
Knowledge Propagation in Trees of Cliques

1. Local computations w.r.t. cliques

2. Collect information

3. Distribute information

Local Operation: Conditioning

Lauritzen, Spiegelhalter, 1988
Shafer, Shenoy, 1988
Effizienz gain with HUGIN

Example: Markov Net for VW „Bora“

- Installation Rates for 460,000 attribute combinations
- Reduction of RAM from 600 MB to 16 MB (Divisor = 38)
- Reduction of computing time from „infeasible“ to 250 sec (Divisor = 80,000)
### Gain with efficient operations

#### Example: Markov Net for Volkswagen Sharan

<table>
<thead>
<tr>
<th></th>
<th>First Prototyping (HUGIN)</th>
<th>... today ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. number dimensions</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>Number tuples</td>
<td>500.000</td>
<td>20.000.000.000</td>
</tr>
<tr>
<td>Valid Tuples</td>
<td>100.000</td>
<td>1.000.000</td>
</tr>
<tr>
<td>Full Network Propagation (Pentium 1.5 GHz)</td>
<td>1 sec</td>
<td>30 ms</td>
</tr>
</tbody>
</table>
Planning Model based on Belief Change

- **Historical data**: context-dependent sample of produced vehicle specifications
- **System of rules**: context-dependent rules for the validity of item combinations
- **Estimate prior distribution of installation rates**
- **Use cond. independencies (Composition)**
- **Quantitative Learning**: PGM (Markov network) having the structure of the relational network
- **Modify representation**: Transformation into a relational network with hypertree structure
- **Revision**: Adaption of installation rates of item combinations that change from valid to invalid
- **Fusion**
- **Updating**: Find referential for item combinations that change from invalid to valid
- **Planning Model**: fused consistent Markov network for item planning
Planning Operation : Updating

■ Input Data : Set of item combinations that will change from invalid to valid; set of valid referential combinations

■ Operation : Copy dependency structure (cross-product ratios) from referential combination to input combination and initialize with $\mathcal{E}$-probabilities.

■ Application : Technical modifications

The combination of engine $e_1$ and transmission $t_3$ changes from invalid to valid, and it adapts the quantitative dependencies from $(e_2, t_3)$.
New Results: Knowledge-based theory and algorithms for revision in MN
(applicable to a family of competing fully / partly specified marginal / conditional probability distributions / absolute settings)

Efficient implementation (Revision takes for hundreds of settings only a few seconds of computation time)
Planning Operation : Revision

- **Input Data**: Family of marginal / conditional probability distributions

- **Operation**: Calculate Markov network with same structure that satisfies all input distributions and is conform to the principle of minimal change.

- **Application**: Marketing stipulations
  Installation probability of item *air condition* increases by 10% in case of *Golf all-wheel drive in France*.

**Logistic restrictions**

The *maximum availability* of engine $e_1$ in week 32/05 is 1.000.
## Specification of Planning Data

<table>
<thead>
<tr>
<th><strong>Name:</strong></th>
<th>Golf - No. 02/07/05 - 17</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vehicle class:</strong></td>
<td>Golf</td>
</tr>
<tr>
<td><strong>Market:</strong></td>
<td>Germany</td>
</tr>
<tr>
<td><strong>Planning interval:</strong></td>
<td>36/05</td>
</tr>
<tr>
<td><strong>Revision scheme:</strong></td>
<td>Engines</td>
</tr>
<tr>
<td><strong>Revision context:</strong></td>
<td>Short back, Comfort</td>
</tr>
<tr>
<td><strong>Context scheme:</strong></td>
<td>Body, Equipment</td>
</tr>
</tbody>
</table>

### Partitioning:

<table>
<thead>
<tr>
<th><strong>Group of 1,8L spark engines</strong></th>
<th>5,79</th>
<th>9,00</th>
<th>≤ 500</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Diesel engine X1 (single item)</strong></td>
<td>2,13</td>
<td>3,00</td>
<td></td>
</tr>
<tr>
<td><strong>Diesel engine X2 (single item)</strong></td>
<td>21,07</td>
<td>[18,20]</td>
<td></td>
</tr>
<tr>
<td><strong>Rest</strong></td>
<td>71,01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Current State of Software Development

• Client-Server System
  (current state: software implementation and test environment for users)

• Server on 6-8 Machines (16 GB each)

• 4-Processor AMD Opteron system

• Terabyte storage device

• Operating System: Linux

• up to 15 system developers

• Programming language: JAVA

• WebSphere Application Developer, Eclipse

• DB-System: Oracle

• Worldwide rollout: now
Need for Theory / Efficient Algorithms

- Efficient transformation of logical rule systems into a relational network, techniques for complexity reduction and inconsistency management
- Consistent quantitative fusion of a prior Markov network with a dependencies modifying relational network to a new Markov network
- Handling generalized constraints
- Efficient algorithms for revision and updating
Is Decomposition Always Possible?