Modern Rule-Based Systems
Future Trends in Knowledge Management and Decision Making in Autonomous Systems

Antoni Ligęza

Katedra Automatyki
Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki
Akademia Górniczo-Hutnicza
Kraków

Zakopane 1.05.2012
Table of contents

1 Introduction

2 State-of-the-Art

3 Important Issues

4 Solutions: Current State and New Trends
   - KR Languages
   - Rule Specialization
   - Inference Control
   - Introducing KB Structure
   - Design
   - User Interface
   - Complex Solutions

5 Concluding Remarks
Wykaz literatury


Wykaz literatury – c.d.

Thinking — What is the Essence of it?
Thinking: Planning, Acting, Monitoring, Reacting, Replanning
Rules and Rule–Based Systems

- Rules – very popular and powerful method for knowledge representation.
  - Usually presented in the **IF...THEN...** form.

  *IF the sun is shining* THEN *it’s warm*

- Rule–based systems – a class of expert/control systems.

  - **Rule 1**: IF $s$ is NB THEN $u_f$ is bigger
  - **Rule 2**: IF $s$ is NM THEN $u_f$ is big
  - **Rule 3**: IF $s$ is Z THEN $u_f$ is medium
  - **Rule 4**: IF $s$ is PM THEN $u_f$ is small
  - **Rule 5**: IF $s$ is PB THEN $u_f$ is smaller
### Rules: Decision Tables

#### Rules void `defineGreeting(App app, int hour)`

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>A1</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>min &lt;= hour</code></td>
<td><code>hour &lt;= max</code></td>
<td><code>app.greeting = greeting;</code></td>
</tr>
<tr>
<td><code>int min</code></td>
<td><code>int max</code></td>
<td><code>String greeting</code></td>
</tr>
<tr>
<td><strong>Hour From</strong></td>
<td><strong>Hour To</strong></td>
<td><strong>Set Greeting</strong></td>
</tr>
<tr>
<td>0</td>
<td>11</td>
<td>Good Morning</td>
</tr>
<tr>
<td>12</td>
<td>17</td>
<td>Good Afternoon</td>
</tr>
<tr>
<td>18</td>
<td>22</td>
<td>Good Evening</td>
</tr>
<tr>
<td>23</td>
<td>24</td>
<td>Good Night</td>
</tr>
</tbody>
</table>

#### Rules void `defineSalutation(App app, Customer c)`

<table>
<thead>
<tr>
<th>Gender</th>
<th>Marital Status</th>
<th>Age Less Than</th>
<th>Set Salutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>Married</td>
<td></td>
<td>Mrs.</td>
</tr>
<tr>
<td>Female</td>
<td>Single</td>
<td></td>
<td>Ms.</td>
</tr>
<tr>
<td>Male</td>
<td></td>
<td>10</td>
<td>Mr.</td>
</tr>
<tr>
<td>Male</td>
<td>Single</td>
<td></td>
<td>Little</td>
</tr>
</tbody>
</table>
Rules: Decision Trees

Introduction

Step 1 Introduction
- phone
- MyQuery
- MyOutput

Step 1
- Select Phone
- Table_vlegend
- Table
- Show Phone

Step 2 Introduction
- plan
- Personal

Step 2
- Select Plan
- Business

Step 3 Introduction
- phone
- plan

Step 3
- BP
- M
- phone

Other text and diagrams are present in the image, but the main focus is on the decision tree structure described above.
## An Idea of Rule-Based System

### Fact Base

<table>
<thead>
<tr>
<th>Fact</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>fact$_1$</td>
<td>#q$_1$</td>
</tr>
<tr>
<td>fact$_2$</td>
<td>#q$_2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>fact$_k$</td>
<td>#q$_k$</td>
</tr>
</tbody>
</table>

where: # stays for negation (¬) or nothing; #p is a **literal**.

### Rule-Base

<table>
<thead>
<tr>
<th>Rule</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>rule$_1$</td>
<td>#p$_1$ ∧ #p$_2$ ∧ ... ∧ #p$_n$ → #h$_1$</td>
</tr>
<tr>
<td>rule$_2$</td>
<td>#p$_1$ ∧ #p$_2$ ∧ ... ∧ #p$_n$ → #h$_2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>rule$_m$</td>
<td>#p$_1$ ∧ #p$_2$ ∧ ... ∧ #p$_n$ → #h$_m$</td>
</tr>
</tbody>
</table>

where: # stays for negation (¬) or nothing; #p is a **literal**.
Structure: Basic components of an Expert System Shell
Structure: Basic components of an Expert System Shell

- **Inference Control Mechanism**
  - Inputs
  - Rule 1
  - Rule 2
  - Rule m
  - Outputs

- **Object**

- **Rule-Based Control System**
Structure: Basic components of an Expert System Shell
Example structure and rules

Start: Context=(aTime='hh:mm' and aLine=n) Output=(aPrice=p)

<table>
<thead>
<tr>
<th>aTime</th>
<th>aPeriod</th>
</tr>
</thead>
<tbody>
<tr>
<td>[08:00-16:00]</td>
<td>bh</td>
</tr>
<tr>
<td>[16:00-20:00]</td>
<td>ah</td>
</tr>
<tr>
<td>[20:00-08:00]</td>
<td>nh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>aPeriod</th>
<th>aSize</th>
<th>aPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>bh</td>
<td>small</td>
<td>87</td>
</tr>
<tr>
<td>bh</td>
<td>medium</td>
<td>77</td>
</tr>
<tr>
<td>bh</td>
<td>large</td>
<td>66</td>
</tr>
<tr>
<td>ah</td>
<td>small</td>
<td>58</td>
</tr>
<tr>
<td>ah</td>
<td>medium</td>
<td>56</td>
</tr>
<tr>
<td>[ah,nh]</td>
<td>large</td>
<td>33</td>
</tr>
<tr>
<td>nh</td>
<td>ANY</td>
<td>37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>aLineNumber</th>
<th>aSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1-10]</td>
<td>small</td>
</tr>
<tr>
<td>[11-30]</td>
<td>medium</td>
</tr>
<tr>
<td>[31-99]</td>
<td>large</td>
</tr>
</tbody>
</table>
## Selected Application Areas

- **society**: co-exitence, ethics, code of honor,
- **law**: orders and prohibitions, taxes,
- **organizations**: army, banks, companies,
- **education and universities**,
- **economy and business**: (BI – Business Intelligence, BR – Business Rules),
- **medicine**: medical procedures,
- **mathematics, physics, chemistry, logics, biology**,
- **technology, management, industry**:
  - monitoring,
  - control,
  - diagnostics,
  - service, repairs,
  - decision support,
  - design and analysis,
  - verification, testing, validation,
  - optimization, adaptation, tuning,
  - configuration.
## Knowledge Engineering: Status of Rule-Based Systems

### Concepts and Foundations: Matured Solutions

- programming languages (e.g. Lisp, Prolog, procedural ones),
- knowledge representation formalisms (attribute logic, FOL),
- formal logic,
- inference,
- inference control.
Concepts and Foundations: Matured Solutions

- programming languages (e.g. Lisp, Prolog, procedural ones),
- knowledge representation formalisms (attribute logic, FOL),
- formal logic,
- inference,
- inference control.

Rule-Based Systems: State-of-the-Art

- matured technology,
- numerous tools — shells,
- numerous successful applications,
- stable market,
- textbook, courses, seminars; research — reincarnation
- embedded solutions, hidden components,
- negative lessons learned.
Rule-Based Systems: Position Statement

- rules are some most successful knowledge representation formalism,
- number of engineering and business applications (hidden or explicit),
- expressive power: high-level declarative knowledge representation,
- *logical independence of applications* — rules are ’data’,
- logical model (in background),
- often combined with other tools,
- visual edition tools (sometimes).
Key Factors for Rule-Based System Success

9 components

- Knowledge representation,
- Inference rules (legal moves),
- Inference control (arrive at the goal; avoid combinatorial explosion),
- Knowledge acquisition:
  - Design,
  - Implementation,
  - Input of Knowledge,
- User interface (picture!), communication, explanations,
- Verification and Validation,
- Modification, extension, adaptation, learning,
- Abstraction, generalization,
- Automated approach.
Knowledge Representation Techniques

Variety of KR Tools

- numeric (numbers, vectors, matrices, functions, equations),
- qualitative (intervals, symbolic, \{-, 0, +\}),
- algebraic (sets, relations, structures),
- logical formalisms (facts, formulas, rules),
  - propositional logic,
  - attribute logic,
  - Datalog, Prolog, FOL,
  - Description Logic,
- rule-based systems, rules,
- graphs, semantic networks,
- frames, structural (objects),
- pictures (diagrams, schemes, blocks),
- combined (e.g. XTT).
Patterns of inference

- Abstraction (generalization),
- Specialization,
- Pattern Matching,
- Case-Based Reasoning, analogy,
- Logical inference:
  - deduction,
  - abduction,
  - induction,
- **Rule-Based Inference (forward, backward, top-down),**
- Search algorithms,
- Problem reduction (AND-OR graph search),
- Constraint Satisfaction Techniques,
- Consistency-Based Reasoning,
- Graph Transformations, Graph Grammars,
- Numerical Procedures (e.g. optimization),
Inference Control Techniques

Efficient Inference Factors

- KR formalism; search-space selection,
- systematic, blind inference/search,
- heuristic inference/search,
- rule selection, conflict resolution,
- meta-rules, inference control rules,
- problem decomposition, structuring, ordering,
- inference planning,
- using constraints,
- problem reduction,
- constraint relaxation,
- mini-max strategies,
- elimination of cases (tabu search).
Example of Jess/Clips

(bind ?*workdays* (create$ Monday Tuesday Wednesday Thursday Friday))

(bind ?*weekend-days* (create$ Saturday Sunday))

; Rule 1

(defrule workday-if-M-F
 (declare (salience 100))
 (day ?d&:(member$ ?d ?*workdays*))
 ?dow <- (day ?d)
 =>
 (assert (workday))
 (retract ?dow)
 (bind ?*WEEKDAY* ?d)
 (bind ?*WEEKEND* False)
)

(defrule workday-if-M-F2
 (declare (salience 100))
 (day ?d&:(member$ ?d ?*workdays*))
 ?workday <- (workday)
 ?dow <- (day ?d)
 =>
 (assert (workday))
 (retract ?workday)
 (retract ?dow)
 (bind ?*WEEKDAY* ?d)
 (bind ?*WEEKEND* True)
)

(defrule weekend-if-S-S
 (declare (salience 100))
 (day ?d&:(member$ ?d ?*weekend-days*))
 ?dow <- (day ?d)
 =>
 (assert (weekend))
 (retract ?dow)
 (bind ?*WEEKDAY* ?d)
 (bind ?*WEEKEND* True)
)

(defrule weekend-if-S-S2
 (declare (salience 100))
 (day ?d&:(member$ ?d ?*weekend-days*))
 ?workday <- (workday)
 ?dow <- (day ?d)
 =>
 (assert (weekend))
 (retract ?workday)
 (retract ?dow)
 (bind ?*WEEKDAY* ?d)
 (bind ?*WEEKEND* True)
Example of Drools Code

```drools
rule "Rule 1"
when
//$wd : WorkDays()
    //$ttd : ThermostatTimeData( day memberOf $wd.daylist )
    $ttd: ThermostatTimeData ( day == "Monday" || day == "Tuesday" || day == "Wednesday" || day == "Thursday" || day == "Friday", daytype == "")
then
//insertLogical( new WorkDay() );
modify ( $ttd ) {
    setDaytype("workday")
}
end

rule "Rule 2"
when
$ttd: ThermostatTimeData ( day == "Saturday" || day == "Sunday", daytype == "")
then
modify ( $ttd ) {
    setDaytype("weekend")
}
end
```
# Example decision table

### EU-Rent discounts for car rental

<table>
<thead>
<tr>
<th>Car Group</th>
<th>Compact</th>
<th>Mid-size, Full-size</th>
<th>Lux, SUV, Van</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rental period</td>
<td>D, W, M</td>
<td>D</td>
<td>D</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Other</td>
<td>W</td>
<td>W</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>M</td>
<td>M</td>
<td>-</td>
</tr>
<tr>
<td>New club member</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
</tr>
<tr>
<td>&gt;3 days in advance</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
</tr>
<tr>
<td>10%</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>€ 50</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Max (10%, € 50)</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>2-group upgrade</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

**Decision tables are used by several commercial rule engines**

*Example from “Decision Tables and Business Rules”, EBRC 2004 Tutorial, Jan Vanthienen*
Attribute Logic: ALSV(FD)

$A_i$ — an attribute $V_i$ — a subset of $D_i$; $d \in D_i$ — a single element.

### Simple Attributes

The legal atomic formulae of ALSV for simple attributes are:

- $A_i = d$,
- $A_i \neq d$,
- $A_i \in V_i$,
- $A_i \notin V_i$.

### Generalized Attributes

The legal atomic formulae of ALSV for generalized attributes are:

- $A_i = V_i$,
- $A_i \neq V_i$,
- $A_i \subseteq V_i$,
- $A_i \supseteq V_i$,
- $A_i \sim V$,
- $A_i \not\sim V_i$. 
### XTT Table

<table>
<thead>
<tr>
<th>Rule</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>...</th>
<th>$A_n$</th>
<th>$H$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\alpha_1 t_{11}$</td>
<td>$\alpha_1 t_{12}$</td>
<td>...</td>
<td>$\alpha_1 t_{1n}$</td>
<td>$h_1$</td>
</tr>
<tr>
<td>2</td>
<td>$\alpha_2 t_{21}$</td>
<td>$\alpha_2 t_{22}$</td>
<td>...</td>
<td>$\alpha_2 t_{2n}$</td>
<td>$h_2$</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>m</td>
<td>$\alpha_m t_{m1}$</td>
<td>$\alpha_m t_{m2}$</td>
<td>...</td>
<td>$\alpha_m t_{mn}$</td>
<td>$h_m$</td>
</tr>
</tbody>
</table>
Example: Thermostat Table

<table>
<thead>
<tr>
<th>Info</th>
<th>Prec</th>
<th>Retract</th>
<th>Assert</th>
<th>Decision</th>
<th>Ctrl</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>aSE</td>
<td>aOP</td>
<td></td>
<td>aTHS</td>
<td>N</td>
</tr>
<tr>
<td>11</td>
<td>spr</td>
<td>dbh</td>
<td>20</td>
<td>1.1</td>
<td>4.12</td>
</tr>
<tr>
<td>12</td>
<td>spr</td>
<td>ndbh</td>
<td>15</td>
<td>1.1</td>
<td>4.13</td>
</tr>
<tr>
<td>13</td>
<td>sum</td>
<td>dbh</td>
<td>24</td>
<td>1.1</td>
<td>4.14</td>
</tr>
<tr>
<td>14</td>
<td>sum</td>
<td>ndbh</td>
<td>17</td>
<td>1.1</td>
<td>4.15</td>
</tr>
<tr>
<td>15</td>
<td>aut</td>
<td>dbh</td>
<td>20</td>
<td>1.1</td>
<td>4.16</td>
</tr>
<tr>
<td>16</td>
<td>aut</td>
<td>ndbh</td>
<td>16</td>
<td>1.1</td>
<td>4.17</td>
</tr>
<tr>
<td>17</td>
<td>win</td>
<td>dbh</td>
<td>18</td>
<td>1.1</td>
<td>4.18</td>
</tr>
<tr>
<td>18</td>
<td>win</td>
<td>ndbh</td>
<td>14</td>
<td>1.1</td>
<td>1.1</td>
</tr>
</tbody>
</table>

rules 11 and 15 can be glued to a single rule, in this case the preconditions would read
\[ aSE \in \{ spr, \, sum \} \land aOP = dbh \]
XTT Prolog

rule(4,11,[f(aSE,atomic,spr),f(aOP,atomic,dbh)],[],[],[f(aTHS,atomic,20)],0,_).

rule(4,12,[f(aSE,atomic,spr),f(aOP,atomic,ndbh)],[],[],[f(aTHS,atomic,15)],0,_).

rule(4,13,[f(aSE,atomic,sum),f(aOP,atomic,dbh)],[],[],[f(aTHS,atomic,24)],0,_).

rule(4,14,[f(aSE,atomic,sum),f(aOP,atomic,ndbh)],[],[],[f(aTHS,atomic,17)],0,_).

rule(4,15,[f(aSE,atomic,aut),f(aOP,atomic,dbh)],[],[],[f(aTHS,atomic,20)],0,_).

rule(4,16,[f(aSE,atomic,aut),f(aOP,atomic,ndbh)],[],[],[f(aTHS,atomic,16)],0,_).

rule(4,17,[f(aSE,atomic,win),f(aOP,atomic,dbh)],[],[],[f(aTHS,atomic,18)],0,_).

rule(4,18,[f(aSE,atomic,win),f(aOP,atomic,ndbh)],[],[],[f(aTHS,atomic,14)],0,_).
XTT Prolog: HMR

xschm th: [today,hour] ==> [operation].

xrule th/1:
  [today eq workday,
   hour gt 17]
==> [operation set not_bizhours].

xrule th/4:
  [today eq workday,
   hour in [9 to 17]]
==> [operation set bizhours].
XTT Example

<table>
<thead>
<tr>
<th>(&gt;? month</th>
<th>(-&gt;) season</th>
</tr>
</thead>
<tbody>
<tr>
<td>in 1;2;12</td>
<td>winter</td>
</tr>
<tr>
<td>in 3;4;5</td>
<td>spring</td>
</tr>
<tr>
<td>in 6;7;8</td>
<td>summer</td>
</tr>
<tr>
<td>in 9;10;11</td>
<td>fall</td>
</tr>
</tbody>
</table>

Table id: 1 - mc

<table>
<thead>
<tr>
<th>(&gt;? day</th>
<th>(-&gt;) today</th>
</tr>
</thead>
<tbody>
<tr>
<td>in mont;wed;thu;fr</td>
<td>workday</td>
</tr>
<tr>
<td>in sat;sun</td>
<td>weekend</td>
</tr>
</tbody>
</table>

Table id: 2 - dt

<table>
<thead>
<tr>
<th>(&gt;? today</th>
<th>(-&gt;) hour</th>
<th>(-&gt;) operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>= workday</td>
<td>&gt; 17</td>
<td>nBzhrs</td>
</tr>
<tr>
<td>= weekend</td>
<td>= ANY</td>
<td>nBzhrs</td>
</tr>
<tr>
<td>= workday</td>
<td>&lt; 9</td>
<td>nBzhrs</td>
</tr>
<tr>
<td>= workday</td>
<td>in 9,17</td>
<td>bizhrs</td>
</tr>
</tbody>
</table>

Table id: 3 - th

<table>
<thead>
<tr>
<th>(&gt;? operation</th>
<th>(-&gt;) season</th>
<th>(-&gt;) thermostat setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>= nBzhrs</td>
<td>= summer</td>
<td>= 27</td>
</tr>
<tr>
<td>= bizhrs</td>
<td>= summer</td>
<td>= 24</td>
</tr>
<tr>
<td>= nBzhrs</td>
<td>= spring</td>
<td>= 15</td>
</tr>
<tr>
<td>= bizhrs</td>
<td>= spring</td>
<td>= 20</td>
</tr>
<tr>
<td>= nBzhrs</td>
<td>= winter</td>
<td>= 14</td>
</tr>
<tr>
<td>= bizhrs</td>
<td>= winter</td>
<td>= 19</td>
</tr>
<tr>
<td>= nBzhrs</td>
<td>= fall</td>
<td>= 18</td>
</tr>
<tr>
<td>= bizhrs</td>
<td>= fall</td>
<td>= 20</td>
</tr>
</tbody>
</table>

Table id: 4 - os
XTT Example

Table id: tab_2 - ms

Table id: tab_3 - Table2

Table id: tab_4 - Table3

Table id: tab_5 - Table4
Solutions: Current State and New Trends

Rule Specialization

XTT Prolog: Forms of Rules

% rule(<table_number>,
%     <rule_number>,
%     [<precondition_list>],
%     [<retract_list>],
%     [<assert_list>],
%     [<decision_list>],
%     <next_table>,
%     <next_rule in next_table>,
%   ).

rule(1,1,[f(aDD,set,sWD)],[f(aTD,set,_)],[f(aTD,atomic,wd)],[],2,3
rule(1,2,[f(aDD,set,sWK)],[f(aTD,set,_)],[f(aTD,atomic,wk)],[],2,6
rule(1,1,[f(aDD,set,sWD)],[f(aTD,set,_)],[f(aTD,atomic,wd)],[],2,3
rule(1,2,[f(aDD,set,sWK)],[f(aTD,set,_)],[f(aTD,atomic,wk)],[],2,6
Possible Complex Forms of Rules

\textbf{rule}(n) \quad \textbf{name}(\textbf{parameters})

\textbf{resources} \quad \textbf{resource(s)}

\textbf{if} \quad \textbf{context\_formula}

\textbf{and} \quad \textbf{excluding\_condition(s)}

\textbf{if not} \quad \textbf{precondition(s)}

\textbf{and} \quad \textbf{detailed\_excluding\_condition(s)}

\textbf{then}

\textbf{do} \quad \textbf{action(s)}

\textbf{retract} \quad \textbf{delete\_result(s)}

\textbf{assert} \quad \textbf{add\_result(s)}

\textbf{output} \quad \textbf{message(s)}

\textbf{next} \quad \textbf{rule(s)}

\textbf{else} \quad \textbf{rule(s)}
Types of Rules

Rule types for CEP
An inference algorithm performs three steps:

1. **Pattern Matching**.
2. **Conflict Set Resolution**.
3. **Action Execution**.

*Pattern Matching* is a bottleneck of the inference process.

The naive algorithm is far too slow.

More efficient algorithms: RETE, TREAT, GATOR.
Rule Inference Algorithm – concepts

- **temporal redundancy** – *most* of the rules have RHS influencing *a few* facts only, and only *a few* rules are affected by those changes,
- **structural similarity** – many rules have a similar pattern in their LHP part.
- Facts are stored in the *Working Memory*.
- LHS consists of *patterns*:

![Diagram showing a network structure with nodes and edges]

- *Network* – a tree-like structure consisting of patterns.
- *Working element* – an object with attribute/value pairs describing it.

<table>
<thead>
<tr>
<th>Object 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>property1 := 2</td>
</tr>
<tr>
<td>property2 := 12</td>
</tr>
<tr>
<td>property3 := 7</td>
</tr>
<tr>
<td>property4 := 11</td>
</tr>
</tbody>
</table>
### HeKatE XTT2: Inference Modes

<table>
<thead>
<tr>
<th>Input tables</th>
<th>Middle tables</th>
<th>Output tables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table 1</strong></td>
<td><strong>Table 3</strong></td>
<td><strong>Table 6</strong></td>
</tr>
<tr>
<td>A1 B1</td>
<td>B1 F1</td>
<td>F1 H1 Z1</td>
</tr>
<tr>
<td>A2 B2</td>
<td>B2 F2</td>
<td>F1 H2 Z2</td>
</tr>
<tr>
<td><strong>Table 2</strong></td>
<td><strong>Table 4</strong></td>
<td><strong>Table 7</strong></td>
</tr>
<tr>
<td>C1 D1</td>
<td>B1 H1</td>
<td>H1 J1 Z5</td>
</tr>
<tr>
<td>C2 D2</td>
<td>B2 H2</td>
<td>H1 J2 Z6</td>
</tr>
<tr>
<td><strong>Table 5</strong></td>
<td><strong>Table 5</strong></td>
<td><strong>Table 8</strong></td>
</tr>
<tr>
<td>D1 J1</td>
<td>D1 J1</td>
<td>H2 J1 Z7</td>
</tr>
<tr>
<td>D2 J2</td>
<td>D2 J2</td>
<td>H2 J2 Z8</td>
</tr>
</tbody>
</table>

- **DDI** with assumption that fact **A1** is in knowledge base and **Table 1** is a start table
- **TDI** with assumption that facts **A2** and **C1** are in knowledge base and **Table 7** and **Table 8** are goal tables
- **GDI** with assumption that facts **A1** and **C2** are in knowledge base, and **Table 7** is a goal table

**T** Token sent from one table to another
Drools Flow

- Provides visualisation and a graphical user interface (GUI):

- Rules are stored in one global knowledge base.
- Rules can be grouped into ruleflow-groups:

```java
rule Żule1
  ruleflow-group "Task1"
  when
    ...
  then
    ...
  end
```

- Only rules from the current ruleflow-group are evaluated and fired.
- Ruleflow-groups determine the order of the rules evaluation and execution.
HeKatE XTT2: Complete Design Process

Conceptual Design

Logical Design

Analysis Verification

Physical Design

PROLOG

XML

Executable

System-specific representation

Other Systems
### User Interface: Decision Tables

<table>
<thead>
<tr>
<th>Objective</th>
<th>Image</th>
<th>Price</th>
<th>Size</th>
<th>Dimensions</th>
<th>Weight</th>
<th>Battery Life</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Direction</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mitsubishi G75</td>
<td>0</td>
<td>59.95</td>
<td>14.63</td>
<td>5.5x1.9x1.4</td>
<td>7.9</td>
<td>3.24</td>
<td>5</td>
</tr>
<tr>
<td>Motorola g520</td>
<td>1</td>
<td>79.95</td>
<td>12.1</td>
<td>5.5x2.0x1.1</td>
<td>6</td>
<td>1.5</td>
<td>3.25</td>
</tr>
<tr>
<td>Ericsson 688</td>
<td>1</td>
<td>99.95</td>
<td>9.69</td>
<td>5.1x1.9x1.0</td>
<td>6.5</td>
<td>1.74</td>
<td>4</td>
</tr>
<tr>
<td>Nokia 5190</td>
<td>1</td>
<td>159.95</td>
<td>11.86</td>
<td>5.2x1.9x1.2</td>
<td>6</td>
<td>3.62</td>
<td>5</td>
</tr>
<tr>
<td>Motorola 6000</td>
<td>1</td>
<td>199.95</td>
<td>11.5</td>
<td>5.0x2.3x0.9</td>
<td>7.2</td>
<td>1.5</td>
<td>3.73</td>
</tr>
<tr>
<td>Ericsson 788</td>
<td>1</td>
<td>349.95</td>
<td>7.01</td>
<td>4.1x1.9x0.9</td>
<td>4.7</td>
<td>1.44</td>
<td>3.33</td>
</tr>
<tr>
<td>Motorola StarTAC 850</td>
<td>1</td>
<td>699.95</td>
<td>8.07</td>
<td>3.9x2.3x0.9</td>
<td>3.5</td>
<td>1.24</td>
<td>2.75</td>
</tr>
<tr>
<td>Bosch World 718</td>
<td>1</td>
<td>389.95</td>
<td>8.8</td>
<td>5.5x2.0 x 0.8</td>
<td>6.7</td>
<td>1.73</td>
<td>3.33</td>
</tr>
</tbody>
</table>
### User Interface: Decision Tables

1. **Do you or have you had a serious, disabling, or life-threatening condition such as stroke, heart, liver or kidney failure, cancer, etc.?**
   - Yes
   - No
   - Unknown/not applicable

2. **Do you have recurring unexplained episodes of any of the following symptoms?**
   - A. Shortness of breath, palpitations, dizziness, or trembling
   - B. Sweating, nausea, choking sensations, or panic attacks
   - C. Numbness, tingling sensations, or unexplained symptoms such as disabling "aches and pains."
   - D. More than one of the above
   - E. None of the above
   - F. Unknown/not applicable

3. **Have you undergone tattooing or body piercing (other than earrings)?**
   - Yes
   - No
   - Unknown/not applicable

4. **Have you seen a psychiatrist, psychologist, or mental health worker in the past six months?**
   - Yes
   - No
   - Unknown/not applicable

5. **Are your symptoms a diagnostic puzzle?**
   - Yes
   - No
   - Unknown/not applicable

6. **Have you ever been treated or hospitalized for depression or a psychiatric condition?**
   - Yes
   - No
   - Unknown/not applicable
### User Interface: Decision Tables

#### Depression Results

<table>
<thead>
<tr>
<th>Condition/Disease</th>
<th>Estimated probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactive Depression</td>
<td>76%</td>
</tr>
<tr>
<td>Major depression</td>
<td>12%</td>
</tr>
<tr>
<td>No significant depression</td>
<td>12%</td>
</tr>
</tbody>
</table>

*Click on any disease for a description.*

#### What do these probabilities mean?

Millions of people visit the Internet daily in search of information about their complaints and conditions. *EasyDiagnosis* offers a novel interactive resource to assist health site visitors **bypass vast amounts of irrelevant medical news and information** offered by traditional medical web sites.

**SUBSCRIBE**  
Why subscribe?

**Links**
Drools5

- More than a classic expert system shell – provides a platform for integration of processes and rules.
- Consists of four modules:
  - Drools Guvnor – knowledge base repository.
  - Drools Expert – rule engine.
  - Drools Flow – workflow modelling.
  - Drools Fusion (event processing/temporal reasoning).
- Only provides forward chaining.
- Inference engine uses a RETE-based algorithm.
- Knowledge represented as rules in Drools5 format:

```plaintext
rule ŻuleName
  // conditions
then
  // actions
end
```
### XTT Example

#### Rule-Based Systems

**Complex Solutions**

---

**A.Ligęza (GEIST - LI - KIS - WEAIE)**

**Solutions: Current State and New Trends**

---

**XTT Example**

<table>
<thead>
<tr>
<th>mth</th>
<th>season</th>
</tr>
</thead>
<tbody>
<tr>
<td>January, February, December</td>
<td>Summer</td>
</tr>
<tr>
<td>March, April, May</td>
<td>Autumn</td>
</tr>
<tr>
<td>June, July, August</td>
<td>Winter</td>
</tr>
<tr>
<td>September, October, November</td>
<td>Spring</td>
</tr>
</tbody>
</table>

**Table: tab_2 - ms**

<table>
<thead>
<tr>
<th>day</th>
<th>today</th>
<th>hour</th>
<th>oper</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>today</td>
<td>workday</td>
<td>in [9, 17]</td>
<td>during_business_hours</td>
<td>20</td>
</tr>
<tr>
<td>today</td>
<td>workday</td>
<td>&lt; 9</td>
<td>not_during_business_hours</td>
<td>15</td>
</tr>
<tr>
<td>today</td>
<td>workday</td>
<td>&gt; 17</td>
<td>not_during_business_hours</td>
<td>24</td>
</tr>
<tr>
<td>today</td>
<td>weekend</td>
<td>any</td>
<td>not_during_business_hours</td>
<td>27</td>
</tr>
</tbody>
</table>

**Table: tab_3 - Table2**

**Table: tab_4 - Table3**

---

**Table: tab_5 - Table4**

---

**AGH**

---

A.Ligęza (GEIST - LI - KIS - WEAIE)

Rule-Based Systems

Zakopane 1.05.2012 44 / 48
Determining season

Determining workday

Determining operation hours

Determining thermostat settings

month in {1,2,12}

set season to summer

month in {3,4,5}

set season to autumn

month in {6,7,8}

set season to spring

month in {9,10,11}

set season to winter
HeKatE XTT2: Complete Design Process
Rule-Based Systems

Examples

- OPS5, OPS 83,
- CLIPS, JESS,
- Drools,
- G2 (Gensym),
- Sphinx/PC-Shell,
- BizTalk Rules Engine,
- XpertRules, ILOG JRULES, Soar.

Active Areas

- Datalog, Prolog, Erlang,...
- Constraints: CLP, ECLIPSE-CLP, CHOCO,
- Answer Set Programming,
- RuleML initiative,
- Semantic Web Stack: Datalog,
- Business Rules.
The End

Many Thanks for your attention!

The Solution of Every Problem

- **DOES IT WORK?**
  - YES
  - NO

- **YOU'RE AN IDIOT**
  - YES
  - NO

- **CAN YOU BLAME SOMEONE?**
  - YES
  - NO

- **SOMEONE KNOWS?**
  - YES
  - NO

- **YOU'RE FUCKED**
  - YES
  - NO

- **DO NOT TOUCH!**
  - YES
  - NO

- **DID YOU TOUCH IT?**
  - YES
  - NO

- **DO WE HAVE ANY ISSUE?**
  - YES
  - NO

- **GO HAEAD!**
  - YES

- **ALL IS FINE**
  - YES
  - NO