Modeling Rule Systems

Overview:
- Decision tables

Literature:
- chapter 2

Development of Rule based systems

- Target a decision to prototype
- Create decision tables
- Implement using XS shell (CLIPS)
  - Decide on strategy (backward vs. forward chaining)
**Decision Tables**

- What is a decision table?
  - Table representing complete set of conditional expressions
  - where expressions are mutually exclusive
  - in a predefined area

**Example decision table**

<table>
<thead>
<tr>
<th>Condition Stubs</th>
<th>Conditions/Courses of Action</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Employee type</td>
<td>S</td>
<td>H</td>
</tr>
<tr>
<td>Hours worked</td>
<td>&lt;40</td>
<td>&lt;40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action Stubs</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pay base salary</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Calculate hourly wage</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculate overtime</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Produce Absence Report</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Decision Tables

- Consists of three parts
  - Condition rows (stubs)
    - Lists condition relevant to decision
  - Action rows (stubs)
    - Actions that result from a given set of conditions
  - Rules
    - Specify which actions are to be followed for a given set of conditions

What kinds of knowledge?

- Regulations, legislation, ...
- Business rules, corporate policy, ...
  - accept/refuse orders
  - discounts
  - ...
- Expertise
- Classification knowledge
  - types of customers
  - risk categories
  - ...
Why use decision tables?

- Powerful visualisation
- Compact and structured presentation
- Preventing errors is easier
  - Avoid incompleteness and inconsistency
- Modular knowledge organisation
  - Group related rules into single table
  - Combine tables to achieve decision

Decision Table Methodology

1. Identify Conditions & Values
   Find the data attribute each condition tests and all of the attribute’s values.

2. Identify Possible Actions
   Determine each independent action to be taken for the decision or policy.

3. Compute Max Number of Rules
   Multiply the number of values for each condition data attribute by each other.

4. Enter All Possible Rules
   Fill in the values of the condition data attributes in each numbered rule column.

5. Define Actions for each Rule
   For each rule, mark the appropriate actions with an X in the decision table.

6. Verify the Policy
   Review completed decision table with end-users.

7. Simplify the Table
   Eliminate and/or consolidate rules to reduce the number of columns.
Scenario

A company wants to unify the way orders are handled:
- All orders of non star-clients with bad credit should be rejected.
- If there is enough product in stock, orders should be accepted; otherwise order is put in waiting list.

Decision diagram

- Which conditions (variables) influence decision?
- What are the values of those conditions?
  - How to find out/calculate?
- What are the relationships between conditions?
Scenario: conditions and actions

- A company wants to unify the way orders are handled:
  - All orders of non star-clients with bad credit should be rejected.
  - If there is enough product in stock, orders should be accepted; otherwise order is put in waiting list.

Decision diagram example

**Conditions:**
- (good, bad)
- (yes, no)
- (sufficient, insufficient)

**Actions:**
- (accept, reject, waiting-list)

- Determine maximum number of rules
  - Number of rules = values(cond1)*...*values(condN)
  - Rules = 2 * 2 * 2 = 8
### Decision table

8 rows = max number of different rules

<table>
<thead>
<tr>
<th>Credit</th>
<th>Star-client</th>
<th>Stock</th>
<th>Order handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td>Yes</td>
<td>Suf</td>
<td></td>
</tr>
<tr>
<td>Good</td>
<td>No</td>
<td>Insuf</td>
<td></td>
</tr>
<tr>
<td>Good</td>
<td>Yes</td>
<td>Suf</td>
<td></td>
</tr>
<tr>
<td>Good</td>
<td>No</td>
<td>Insuf</td>
<td></td>
</tr>
<tr>
<td>Bad</td>
<td>Yes</td>
<td>Suf</td>
<td></td>
</tr>
<tr>
<td>Bad</td>
<td>No</td>
<td>Insuf</td>
<td></td>
</tr>
<tr>
<td>Bad</td>
<td>Yes</td>
<td>Suf</td>
<td></td>
</tr>
<tr>
<td>Bad</td>
<td>No</td>
<td>Insuf</td>
<td></td>
</tr>
</tbody>
</table>
**Decision table**

<table>
<thead>
<tr>
<th>credit</th>
<th>good</th>
<th>good</th>
<th>good</th>
<th>bad</th>
<th>bad</th>
<th>bad</th>
<th>bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>star-client</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Stock</td>
<td>Suf</td>
<td>Insuf</td>
<td>Suf</td>
<td>Insuf</td>
<td>Suf</td>
<td>Insuf</td>
<td>Suf</td>
</tr>
<tr>
<td>Order handling</td>
<td>Accept</td>
<td>Wait</td>
<td>Accept</td>
<td>Wait</td>
<td>Accept</td>
<td>Wait</td>
<td>Reject</td>
</tr>
</tbody>
</table>

- Check table with experts!
- All conditions?
- All values?
- Correct decisions?

**Simplification**

- Redundant rules?

<table>
<thead>
<tr>
<th>credit</th>
<th>good</th>
<th>good</th>
<th>good</th>
<th>bad</th>
<th>bad</th>
<th>bad</th>
<th>bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>star-client</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Stock</td>
<td>Suf</td>
<td>Insuf</td>
<td>Suf</td>
<td>Insuf</td>
<td>Suf</td>
<td>Insuf</td>
<td>Suf</td>
</tr>
<tr>
<td>Order handling</td>
<td>Accept</td>
<td>Wait</td>
<td>Accept</td>
<td>Wait</td>
<td>Accept</td>
<td>Wait</td>
<td>Reject</td>
</tr>
</tbody>
</table>

- Type of client does not matter
- Stock does not matter
## Decision table simplified

<table>
<thead>
<tr>
<th>credit</th>
<th>good</th>
<th>bad</th>
<th>bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>star-client</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Stock</td>
<td>Suf</td>
<td>Insuf</td>
<td>Suf</td>
</tr>
<tr>
<td>Order handling</td>
<td>Accept</td>
<td>Wait</td>
<td>Accept</td>
</tr>
</tbody>
</table>

## From decision tables to rules

<table>
<thead>
<tr>
<th>credit</th>
<th>good</th>
<th>bad</th>
<th>bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>star-client</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Stock</td>
<td>Suf</td>
<td>Insuf</td>
<td>Suf</td>
</tr>
<tr>
<td>Order handling</td>
<td>Accept</td>
<td>Wait</td>
<td>Accept</td>
</tr>
</tbody>
</table>

R1: IF credit = good AND stock = suf THEN handling = accept
R2: IF credit = good AND stock = insuf THEN handling = waiting-list
R3: IF credit = bad AND star-client = yes AND stock = suf THEN handling = accept
R4: IF credit = bad AND star-client = yes AND stock = insuf THEN handling = waiting-list
R5: IF credit = bad AND star-client = no THEN handling = reject
Rules to CLIPS

- R1: IF credit = good AND stock = suf THEN handling = accept

(defrule credit_rule
  (credit good)
  (stock suf)
  =>
  (assert (handling accept)))

Syntax!!

Combining decision tables

- Not all values are directly determined, some times decision is needed.
- Scenario revisited:
  - When is a client a star-client?
  - How to calculate if stock is sufficient?
  - How to know quality of credit?
  - ...
  - Ex:
    - Star clients have account age of more than 2 years, or have an average order amount higher than 5K
    - Stock is sufficient if requested product is available in the storehouse in a higher quantity than ordered
Extending decision diagram

Combining decision tables
Next lecture: 29 September

Frames/ontologies