Training Session on RuleML Technology

Loan Processor Suite: Transforming, Visualizing, and Querying Datalog RuleML Decision Rules

Harold Boley
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, Canada

Decision CAMP, 4-6 November 2013
eBay Town Hall, San Jose, CA
How to Process a Loan?

“Loans are processed after a credit application is filled out and turned in to the loan officer. This will start the loan processing course, first by pre-qualifying the buyer by reviewing the credit history and debt load of the applicant.”

http://www.ask.com/question/how-to-process-a-loan
Use Decision Rules and Data Facts!

The Loan Processor Suite is a series of formal documents for testing and learning *Graph inscribed logic (Grailog), RuleML, SVG, XSLT, POSL, OO jDREWW, etc.* They exemplify *Datalog RuleML* decision rules and data facts that are being transformed, visualized, and queried.

See: [http://ruleml.org/papers/Primer](http://ruleml.org/papers/Primer)

Decision Rule / Data Fact Architecture

Decision Rules:
- matchingAssessor
- ratingAssessor
- clientCategory

Data Facts:
- loanProcessor
- riskBrancher
- clientEvaluation

See: [http://www.cs.unb.ca/~boley/Grailog/LoanProcessor/LoanProcessor.txt](http://www.cs.unb.ca/~boley/Grailog/LoanProcessor/LoanProcessor.txt)
Joint Agile Development of Decision Rules and Data Facts

The predicate `loanProcessor` considers requests with $\text{AmountAsk} \in (0, 500000]$, also fixing $\text{RiskLevel} = 1000$ and $\text{RatiMin} = 0.8$ in a call to `riskBrancher`:

Its "$\lt\$" rule invokes `matchingAssessor`, hence `clientCategory`, for table lookup.

Its "$\geq\$" rule uses `ratingAssessor`, hence `clientEvaluation`, for deep analysis.

Possible outcomes are either failure, without $\text{AmountGrant}$ binding, or success, with $\text{AmountGrant}$ bound to the loan.

Predicate definitions can flexibly combine rules and facts using Prolog/Datalog-like POSL syntax (e.g., "$\leftarrow\$" as the "IF" infix):

[Link to source](http://ojs.academypublisher.com/index.php/jetwi/article/view/0204343353)
Decision Rule: loanProcessor

\[\text{loanProcessor}(\text{?Client}, \text{?AmountAsk}, \text{?AmountGrant}) :\]
\[\text{greaterThan}(\text{?AmountAsk}, 0), \]
\[\text{lessThanOrEqual}(\text{?AmountAsk}, 500000), \]
\[\text{riskBrancher}(\text{?Client}, \text{?AmountAsk}, \text{?AmountGrant}, 1000, 0.8). \]

% loanProcessor(In,In,Out)

% Positive loan request

% up to half a million.
Decision Rules: riskBrancher

\[
\text{riskBrancher}(\text{?Client}, \text{?AmountAsk}, \text{?AmountGrant}, \text{?RiskLevel}, \text{?RatiMin}) :- \\
\text{lessThan}(\text{?AmountAsk}, \text{?RiskLevel}), \\
% \text{Deterministic} ... \\
\text{matchingAssessor}(\text{?Client}, \text{?AmountAsk}, \text{?AmountGrant}).
\]

\[
\text{riskBrancher}(\text{?Client}, \text{?AmountAsk}, \text{?AmountGrant}, \text{?RiskLevel}, \text{?RatiMin}) :- \\
\text{greaterThanOrEqual}(\text{?AmountAsk}, \text{?RiskLevel}), \\
% ... \text{branch} \\
\text{ratingAssessor}(\text{?Client}, \text{?AmountAsk}, \text{?AmountGrant}, \text{?RatiMin}).
\]
Decision Rules: matchingAssessor

matchingAssessor(?Client,?AmountAsk,?AmountGrant) :-
  clientCategory(?Client,gold),  
  % ?Client fact matches gold category
  multiply(?AmountGrant,?AmountAsk,0.75).  
  % ?AmountGrant = ?AmountAsk * 0.75

matchingAssessor(?Client,?Amount,?Amount) :-
  IF  
  % ?Amount = ?AmountGrant = ?AmountAsk  
  clientCategory(?Client,platinum).
  % ?Client fact matches platinum category
Decision Rule: ratingAssessor

\[
\text{ratingAssessor}(\text{Client}, \text{AmountAsk}, \text{AmountGrant}, \text{RatiMin}) \leftarrow \\
\text{clientEvaluation}(\text{Client}, \text{AmountAsk}, \text{Rating}), \\
\% \text{ Data analysis binds } \text{Rating} \text{ in } [0,1]. \\
\text{greaterThanOrEqual}(\text{Rating}, \text{RatiMin}), \\
\% \text{ For } \text{Rating} < \text{RatiMin}: \text{not approved} \\
\text{multiply}(\text{AmountGrant}, \text{AmountAsk}, \text{Rating}). \\
\% \text{AmountGrant} = \text{AmountAsk} \ast \text{Rating}
\]
Data Facts: clientCategory

Facts store database table which captures qualitative categories of previous quantitative analysis

clientCategory(nilper,silver).

clientCategory(bold,gold).

clientCategory(claritum,platinum).
Data Facts: clientEvaluation

(Non-ground) facts cache ratings from deep client data analysis, which could be made conditional on (currently free) ?AmountAsk

clientEvaluation(nilper,?AmountAsk,0.77).
clientEvaluation(bold,?AmountAsk,0.79).
clientEvaluation(claritum,?AmountAsk,0.91).
clientEvaluation(ralcitum,?AmountAsk,0.91).
Query Rules: negTest

\texttt{negTest(1,}\ ?\texttt{AG}) \ :- \texttt{loanProcessor}\left(\textit{nilper},200,\ ?\texttt{AG}\right). \\
\quad \% \textit{Failure since } 200 < \textcolor{red}{1000} \textit{ and } \% \textit{ no silver match} \\

\texttt{negTest(2,}\ ?\texttt{AG}) \ :- \texttt{loanProcessor}\left(\textit{nilper},1100,\ ?\texttt{AG}\right). \\
\quad \% \textit{Failure since } 1100 > \textcolor{red}{1000} \textit{ and } \% \quad 0.77 < 0.8 \\

\texttt{negTest(3,}\ ?\texttt{AG}) \ :- \texttt{loanProcessor}\left(\textit{bold},1100,\ ?\texttt{AG}\right). \\
\quad \% \textit{Failure since } 1100 > \textcolor{red}{1000} \textit{ and } \% \quad 0.79 < 0.8 \\

\texttt{negTest(4,}\ ?\texttt{AG}) \ :- \texttt{loanProcessor}\left(\textit{ralcitum},200,\ ?\texttt{AG}\right). \\
\quad \% \textit{Failure since } 200 < \textcolor{red}{1000} \textit{ and } \% \textit{ no category}
Query Rule: posTest

\texttt{posTest(?AG1,?AG2,?AG3,?AG4) :-
loanProcessor(bold,200,?AG1),
  \% Success with ?AG1 = 150 since 200 < 1000 and
  \% gold match
loanProcessor(claritum,200,?AG2),
  \% Success with ?AG2 = 200 since 200 < 1000 and
  \% platinum match
loanProcessor(claritum,400000,?AG3),
  \% Success with ?AG3 = 364k since 400k > 1k and
  \% 0.91 > 0.8
loanProcessor(ralcitum,500000,?AG4).
  \% Success with ?AG4 = 455k since 500k > 1k and
  \% 0.91 > 0.8}
Transforming POSL to RuleML/XML

\[ \text{loanProcessor}(\text{?Client}, \text{?AmountAsk}, \text{?AmountGrant}) \vdash \]
\[ \quad \text{% loanProcessor}(\text{In}, \text{In}, \text{Out}) \]
\[ \text{greaterThan}(\text{?AmountAsk}, \text{0}), \]
\[ \quad \text{% Positive loan request} \]
\[ \text{lessThanOrEqual}(\text{?AmountAsk}, \text{500000}), \]
\[ \quad \text{% up to half a million.} \]
\[ \text{riskBrancher}(\text{?Client}, \text{?AmountAsk}, \text{?AmountGrant}, \text{1000}, \text{0.8}). \]
\[ \quad \text{% \ldots, ?RiskLevel, ?RatiMin}. \]

\[ \text{OO jDREW 1.0 POSL/RuleML Translator} \]

Complete: [http://www.cs.unb.ca/~boley/Grailog/LoanProcessor/LoanProcessor.xml](http://www.cs.unb.ca/~boley/Grailog/LoanProcessor/LoanProcessor.xml) (View Page Source)
Visualizing RuleML/XML as Grailog/SVG

<Implies>
  <And>
    <Atom>
      <Rel>greaterThan</Rel>
      <Var>AmountAsk</Var>
      <Ind>0</Ind>
    </Atom>
    <Atom>
      <Rel>lessThanOrEqual</Rel>
      <Var>AmountAsk</Var>
      <Ind>500000</Ind>
    </Atom>
    <Atom>
      <Rel>riskBrancher</Rel>
      <Var>Client</Var>
      <Var>AmountAsk</Var>
      <Var>AmountGrant</Var>
      <Ind>1000</Ind>
      <Ind>0.8</Ind>
    </Atom>
  </And>
  <Atom>
    <Rel>loanProcessor</Rel>
    <Var>Client</Var>
    <Var>AmountAsk</Var>
    <Var>AmountGrant</Var>
  </Atom>
</Implies>

Complete: http://www.cs.unb.ca/~boley/Grailog/LoanProcessor/LoanProcessor.svg (View Page Source)
Querying with OO jDREW

See: [OO jDREW 1.0 snapshot (Java Web Start)](https://example.com)
Exercises

1) Give client ralcitum the category platinum and explain what will happen for the positive and negative tests.

2) Augment the decision rules so that clients of category silver (e.g., client nilper) will obtain half of the amount asked. Hint: Model category silver in analogy to category gold.

3) Update the entire LP Suite (transformation, visualization, and querying) using 1) and 2)