Development and Verification of Rule Based Systems - a Survey of Developers

Valentin Zacharias
FZI
Karlsruhe, Germany
Results from an Online Survey about Rule Base Development

**Agenda**

- Participants
- Development Methodology
- Tools
  - Development
  - Debugging
  - V&V
- Comparison to OO Development & Issues Hindering Dev.

- Comparison to last surveys (from 18 years ago)
76 Participants
Languages and Systems?
## Size

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM for entire software</td>
<td>59</td>
<td>15</td>
<td>148</td>
</tr>
<tr>
<td>PM for rule base</td>
<td>9</td>
<td>5.5</td>
<td>15</td>
</tr>
<tr>
<td>Number of rules</td>
<td>1969</td>
<td>120</td>
<td>8693</td>
</tr>
<tr>
<td>Size of average rule</td>
<td>9.3</td>
<td>5</td>
<td>17</td>
</tr>
<tr>
<td>Size of largest rule</td>
<td>24</td>
<td>11</td>
<td>39</td>
</tr>
<tr>
<td>Rule developers</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Other software developers</td>
<td>3</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Domain experts that created rules</td>
<td>1.5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Domain experts as consultants</td>
<td>1.9</td>
<td>1</td>
<td>2.5</td>
</tr>
<tr>
<td>Domain experts for V&amp;V</td>
<td>1.7</td>
<td>1</td>
<td>2.4</td>
</tr>
<tr>
<td>Others</td>
<td>0.6</td>
<td>0</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Standard Deviation</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------</td>
<td>--------------------</td>
<td></td>
</tr>
<tr>
<td>PM for entire software</td>
<td>59</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>PM for rule base</td>
<td>9</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Number of rules</td>
<td>1969</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>Size of average rule</td>
<td>9,3</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Size of largest rule</td>
<td>24</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Rule developers</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Other software developers</td>
<td>3</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Domain experts that created rules</td>
<td>1,5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Domain experts as consultants</td>
<td>1,9</td>
<td>2,5</td>
<td></td>
</tr>
<tr>
<td>Domain experts for V&amp;V</td>
<td>1,7</td>
<td>2,4</td>
<td></td>
</tr>
<tr>
<td>Others</td>
<td>0,6</td>
<td>1,6</td>
<td></td>
</tr>
</tbody>
</table>
Prototype, also used by others than the developer(s)

Prototype/Experiment, only used by the developer(s)

In development - deployment planned

Deployed Research

Deployed Commercial

Use?
Methodologies

“CommonKADS is *the* methodology to build rule based systems”
Methodology

- Prototype driven (e.g. MOKA)
- Knowledge engineering (e.g. CommonKADS)
- Other Iterative and Incremental processes (e.g. RUP)
- Waterfall model / Specification driven
- Agile (e.g. ABRD, Xtreme Programming)
- Did not use a specific development process
Prototype driven (e.g. MOKA)

Knowledge engineering (e.g. CommonKADS)

Other Iterative and Incremental processes (e.g. RUP)

Waterfall model / Specification driven

Agile (e.g. ABRD, Xtreme Programming)

Did not use a specific development process
Methodology – only 10PM+

- Prototype driven (e.g. MOKA)
- Knowledge engineering (e.g. CommonKADS)
- Other Iterative and Incremental processes (e.g. RUP)
- Waterfall model / Specification driven
- Agile (e.g. ABRD, Xtreme Programming)
- Did not use a specific development process

Medical Best Practices
Integration of Car Information
Scheduling (Logistics)
Credit Allocation
Methodology – only 10PM+

Prototype driven (e.g. MOKA)

Knowledge engineering (e.g. CommonKADS)

Other Iterative and Incremental processes (e.g. RUP)

Waterfall model / Specification driven

Agile (e.g. ABRD, Xtreme Programming)

Did not use a specific development process

18 Years Ago:

“The most frequent (40%) life cycle model used was the cyclic model [...]. However 22% of the respondents stated that no model was followed”
Tools used for Development

“I doubt that manually created rule bases will serve as the basis for the semantic web or other application areas”
Development Tools

- Text Editor
- Textual Rule Editor
- Constraint / Business Language
- Graphical Rule Editor
- Spreadsheets
- Decision Trees
- Rule Learner
- IDE
- Other

Percent of Participants that said to be using a particular tool
IDE without debugging
Hard coded rules
Schema aware XML Editor
'this [Text Editor] is only for viewing the results
SWRL Tab
Verification & Debugging

“The correctness of rule bases is ensured with formal verification”
Debugging Paradigms for Rule Based Systems

• Procedural Debugging
  – Stepwise execution of inference engine

• Algorithmic Debugging
  – System identifies fault by asking user (oracle) about correctness of results of subcomputations

• Explanations
  – Concise NL or graphical representation of justification for a result

• Why-Not Explanations
  – Explanation also for missing conclusions

• Automatic Theory Revision
  – Automatic correction of rule base

Debugging Tools

- Command Line Procedural Debugger
- Graphical Procedural Debugger
- Algorithmic Debugger
- Explanations
- Why-Not Explanations
- Automatic theory revision & comp.
- None

% of responses
Comparison To ‘Conventional Programming’ and Issues Hindering Development

“Runtime Performance and Expressivity are the problems hindering the development of rule based systems”
Comparison – Question

• How does the rule base and its development process compare to a ‘conventional’ program (created with procedural/object oriented languages) of similar size?

1. Ease of change and maintenance
   Rule base superior; Comparable; Conventional program superior; Don’t know

2. [...]

Computer average with:
* Rule base superior = 1
* Comparable = 0
* Conventional program sup= -1
* Ignore ‘Don’t know’
Issues Hindering Development – Question

• What were the most important issues in the development of the rule base?

  1. Rule expressivity – could not (easily) represent what was needed
     Not an issue; Annoyance; Hindered development
  2. [...]
## Issues Hindering Development

<table>
<thead>
<tr>
<th>Issue</th>
<th>Average</th>
<th>Not an Issue</th>
<th>Annoyance</th>
<th>Hinderance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debugging</td>
<td>1</td>
<td>12</td>
<td>28</td>
<td>12</td>
</tr>
<tr>
<td>Determining completeness</td>
<td>0.76</td>
<td>18</td>
<td>27</td>
<td>6</td>
</tr>
<tr>
<td>Supporting tools missing/immature</td>
<td>0.67</td>
<td>26</td>
<td>17</td>
<td>9</td>
</tr>
<tr>
<td>Editing of rules</td>
<td>0.66</td>
<td>24</td>
<td>23</td>
<td>6</td>
</tr>
<tr>
<td>Determining test coverage</td>
<td>0.65</td>
<td>25</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>Inexperienced developers</td>
<td>0.58</td>
<td>31</td>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>Rule expressivity</td>
<td>0.5</td>
<td>33</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>Keeping rules base up to date</td>
<td>0.5</td>
<td>30</td>
<td>19</td>
<td>4</td>
</tr>
<tr>
<td>Understanding the rule base</td>
<td>0.47</td>
<td>31</td>
<td>19</td>
<td>3</td>
</tr>
<tr>
<td>Runtime performance</td>
<td>0.41</td>
<td>35</td>
<td>14</td>
<td>4</td>
</tr>
<tr>
<td>Organizing collaboration</td>
<td>0.41</td>
<td>35</td>
<td>14</td>
<td>4</td>
</tr>
</tbody>
</table>
Issues Hindering Development

18 Years Ago:

1. Completeness of Knowledge Base
2. Correctness of Knowledge Base
3. System does not present possible opportunities to the user
4. System is hard to use
5. Results difficult to interpret
6. System presents incorrect opportunities to the user
7. Difficult to sequence rules correctly
Conclusions
Meta Conclusion

„This paper is not a (normal) scientific paper, it summarizes a survey [...]“

• Little empirical data about the challenges facing actual rule base developers
• Little interest in the academic rule community at identifying and tackling practical problems?
Conclusion

• Little academic interest in relevant (in particular agile) rule base development methodologies
• Debugging and finding faults as *the challenge*
• Tool support found wanting, possible motivation for rule interchange
• still most rules created manually, text editors widespread
Thanks for your Attention

Valentin Zacharias
zach@fzi.de
http://vzach.de

Attribution

- Pool Rules by Joe Shlabotnik on Flickr
- Waterfall by Sachman75 on Flickr
- Tools by docman on Flickr
- Bug by Chewy Chua on Flickr
- Obstacle by Guillaume Lemoine on Flickr
- Sun Conclusion by ecstaticist on Flickr
- End by bondywhat on Flickr