Xtext vs. MPS: Decision Criteria

Niko Stotz, mail@nikostotz.de
2021-03-03
Textual vs. Projectional: Decision Criteria

Strumenta Community

2021-03-03
Loose distinction: “If you directly edit what’s written on disk, it’s textual.”

<table>
<thead>
<tr>
<th>Textual</th>
<th>Projectional</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ANTLR</td>
<td>• MetaEdit+</td>
</tr>
<tr>
<td>• MontiCore</td>
<td>• MPS</td>
</tr>
<tr>
<td>• Racket</td>
<td>• Sirius</td>
</tr>
<tr>
<td>• Rascal</td>
<td></td>
</tr>
<tr>
<td>• Spoofax</td>
<td></td>
</tr>
<tr>
<td>• Xtext</td>
<td></td>
</tr>
</tbody>
</table>
No killer arguments, but criteria to think about

• Both Xtext and MPS have strengths and weaknesses
• Consider + balance several criteria

→ Use right tool for this job

(Might be different tool for next job)
Targeted User Group

Developers
• used to IDE features
• more similar to other IDEs, especially in source code handling
• better integration in existing tool landscape

Business
• main competitor: Excel
• profit more of MPS features
Tool Environment

- Eclipse
- EMF

- Nothing reusable
Language Properties

- established text-based language
- no tabular structures

- profits from different editors
- requires different viewpoints
- tight integration of separate languages
- unparsable
Input Type

(free) text

• structured
• guided
• pre-defined
• restricted

Examples:
forms, steps / wizards, trees

→ Xtext

→ MPS
Environment

• web (at least as option)
• tight integration with other tools (workflow, arcane versioning, etc.)

• desktop
• loose integration with other tools
Transformations: Model-to-Model

- leverage existing technologies
- re-use existing transformations
- lots of intermediate languages
- extensible transformations
Transformations: Model-to-Text

- tight control of output required
- (close to) target language available
Extensibility

- discouraged
- single tool / language vendor

→ Xtext

- encouraged
- multiple vendors / community

→ MPS
Conceptual Framework / Theory

- Well-researched since 70s
- Based on solid theory
- Several comparable, stable, usable implementations

- Theory: ?
- Usable implementations proprietary*, hard to compare

*although often Open Source