Language Workbenches
The State of the Art in 2015
0 Languages, Models, Programs
1 Language Workbenches
2 Notational Flexibility
3 Language Modularity
4 Specification Languages
5 LWBs and the Web
6 Miscellaneous
7 Why you should care
Languages, Models, Programs
<table>
<thead>
<tr>
<th></th>
<th>more in GPLs</th>
<th>more in DSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Size</td>
<td>large and complex</td>
<td>smaller and well-defined</td>
</tr>
<tr>
<td>Designed by</td>
<td>guru or committee</td>
<td>a few engineers and domain</td>
</tr>
<tr>
<td></td>
<td></td>
<td>experts</td>
</tr>
<tr>
<td>Language Size</td>
<td>large</td>
<td>small</td>
</tr>
<tr>
<td>Turing-completeness</td>
<td>almost always</td>
<td>often not</td>
</tr>
<tr>
<td>User Community</td>
<td>large, anonymous and</td>
<td>small, accessible and local</td>
</tr>
<tr>
<td></td>
<td>widespread</td>
<td></td>
</tr>
<tr>
<td>In-language abstraction</td>
<td>sophisticated</td>
<td>limited</td>
</tr>
<tr>
<td>Lifespan</td>
<td>years to decades</td>
<td>months to years (driven by</td>
</tr>
<tr>
<td></td>
<td></td>
<td>context)</td>
</tr>
<tr>
<td>Evolution</td>
<td>slow, often standardized</td>
<td>fast-paced</td>
</tr>
<tr>
<td>Incompatible Changes</td>
<td>almost impossible</td>
<td>feasible</td>
</tr>
</tbody>
</table>
Trackpoint* makeTP(uint16 alt, int16 speed) {
    static int8 trackpointCounter = 0;
    trackpointCounter++;
    Trackpoint* tp = ((Trackpoint*) malloc(sizeof Trackpoint));
    tp->id = trackpointCounter;
    tp->timestamp = trackpointCounter;
    tp->alt = alt
    tp->speed = speed
    return tp;
}
statemachine HierarchicalFlightAnalyzer initial = beforeFlight {
    in next()
    in reset()
    out crashNotification() -> raiseAlarm
    state beforeFlight {
        on next [tp->alt > 0 m] -> airborne
    }
    composite state airborne initial = flying {
        on reset [ ] -> beforeFlight
        on next [tp->alt == 0 m && tp->speed == 0 mps] -> crashed
        state flying {
            on next [tp->alt == 0 m && tp->speed > 0 mps] -> landing
            on next [tp->speed > 200 mps] -> airborne
            on next [tp->speed > 100 mps] -> airborne
        }
        state landing {
            on next [tp->speed == 0 mps] -> landed
            on next [ ] -> landing
        }
        state landed {
        }
    }
    state crashed {
    }
}
statemachine HierarchicalFlightAnalyzer initial = beforeFlight {
    in next(Trackpoint* tp)
    in reset()
    out crashNotification() -> raiseAlarm
    readable var int16 points = 0
    state beforeFlight {
        on next [tp->alt > 0 m] -> airborne
        exit { points += TAKEOFF; }
    }

    composite state airborne initial = flying {
        on reset [] -> beforeFlight { points = 0; }
        on next [tp->alt == 0 m && tp->speed == 0 mps] -> crashed
        state flying {
            on next [tp->alt == 0 m && tp->speed > 0 mps] -> landing
            on next [tp->speed > 200 mps] -> airborne { points += VERY_HIGH_SPEED; }
            on next [tp->speed > 100 mps] -> airborne { points += HIGH_SPEED; }
        }
        state landing {
            on next [tp->speed == 0 mps] -> landed
            on next [] -> landing { points--; }
        }
        state landed {
            entry { points += LANDING; }
        }
    }

    state crashed {
        entry { send crashNotification(); }
    }
}
Does it really matter?
What is the difference?
Who cares?
[Model or Code: Notation!]

Textual

Graphical

Tabular

Mathematical

\[ \sum \]

\begin{tabular}{|c|c|c|}
\hline
\textbf{Column 1} & \textbf{Column 2} & \textbf{Column 3} \\
\hline
Row 1 & Row 2 & Row 3 \\
\hline
\end{tabular}
One important aspect of languages.
The other important aspect.

- Application Domain
- Solution Domain
- Application Domain
- Solution Domain
- Application Domain
- Solution Domain

- Contracts
- State Machines
- Java Code
- Bytecode

- ?
- Model?
- Code?
- ???
The other important aspect.

- Contracts
- State Machines
- Java Code
- Bytecode

- Artifact
- Model
- Program
- Prodel
[Relationships]
[Relationships]

representation of

model of

part of

view of
model of Abstraction
part of Hierarchy, Partitioning
view of Concern, Aspect, Viewpoint
representation of Presentation, Notation, Form
INVOLVE STAKEHOLDERS
SEPARET CONCERNS
AUTOMATE WHAT YOU MUST
USE APPROPRIATE ABSTRACTIONS
USE FRONT-LOADING

Model-Based Engineering • Model-Driven
(Software) Development • Domain-Specific
Languages • Language-Oriented Programming
Language Workbench(es)

... are the tools to do:

Model-Based Engineering • Model-Driven (Software) Development • Domain-Specific Languages • Language-Oriented Programming
Language Workbench
(Martin Fowler)

Freely define languages and integrate them
Language Workbench
(Martin Fowler)

use persistent abstract representation
Language Workbench
(Martin Fowler)
Language Workbench
(Martin Fowler)

Persist incomplete or contradictory information
Language Workbench
(Martin Fowler)

- powerful editing
- testing
- refactoring
- debugging
- groupware

language definition implies IDE definition
Language Workbench
(Martin Fowler)
support for “classical” programming and modeling
The State of the Art in Language Workbenches

Conclusions from the Language Workbench Challenge

Sebastian Erdweg, Tijs van der Storm, Markus Völter, Meinte Boersma, Remi Bosman, William R. Cook, Albert Gerritsen, Angelo Hulshout, Steven Kelly, Alex Loh, Gabriël Konat, Pedro J. Molina, Martin Palatnik, Eugen Schindler, Klemens Schindler, Riccardo Solmi, Vlad Vergu, Eelco Visser, Kevin van der Vlist, Guido Wachsmuth, and Jimi van der Woning

1 TU Darmstadt, Germany  2 CWI, Amsterdam, The Netherlands  3 INRIA Lille Nord Europe, Lille, France  4 voelter.de, Stuttgart, Germany  5 DSL Consultancy, Leiden, The Netherlands  6 Sioux, Eindhoven, The Netherlands  7 University of Texas, Austin, US  8 Delphino Consultancy, Best, The Netherlands  9 MetaCase, Jyväskylä, Finland  10 TU Delft, The Netherlands  11 Incentive, Sevilla, Spain  12 Independent, Bologna, Italy  13 Universiteit van Amsterdam
<table>
<thead>
<tr>
<th>Notation</th>
<th>Textual</th>
<th>Más</th>
<th>MetaEdit+</th>
<th>MPS</th>
<th>Onion</th>
<th>Rascal</th>
<th>Spoofax</th>
<th>SugarJ</th>
<th>Whole</th>
<th>Xtext</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Graphical</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Tabular</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Symbols</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>Semantics</td>
<td>Model2Text</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Model2Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Concrete syntax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Interpretative</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Validation</td>
<td>Structural</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Naming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Programmatic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Ensō</td>
<td>Más</td>
<td>MetaEdit+</td>
<td>MPS</td>
<td>Onion</td>
<td>Rascal</td>
<td>Spoofax</td>
<td>SugarJ</td>
<td>Whole</td>
<td>Xtext</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>------</td>
<td>-----------</td>
<td>------</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td><strong>Notation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Textual</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Graphical</td>
<td>●</td>
<td>●</td>
<td>❌</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tabular</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbols</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Semantics</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model2Text</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Model2Model</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concrete syntax</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interpretative</td>
<td>●</td>
<td>●</td>
<td>❌</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Validation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structural</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Naming</td>
<td>❌</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Types</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmatic</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Notation
- **Textual**
- **Graphical**
- **Tabular**

### Semantics
- **Model2Text**
- **Model2Model**
- **Concrete syntax**

### Validation
- **Structural**
- **Naming**
- **Types**
- **Programmatic**

<table>
<thead>
<tr>
<th></th>
<th>Ensō</th>
<th>Más</th>
<th>MetaEdit+</th>
<th>MPS</th>
<th>Onion</th>
<th>Rascal</th>
<th>Spoofax</th>
<th>SugarJ</th>
<th>Whole</th>
<th>Xtext</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textual</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Graphical</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Tabular</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Symbols</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Model2Text</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Model2Model</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Concrete syntax</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Interpretative</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Structural</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Naming</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Types</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Programmatic</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Enso</td>
<td>Más</td>
<td>MetaEdit+</td>
<td>MPS</td>
<td>Onion</td>
<td>Rascal</td>
<td>Spoofax</td>
<td>SugarJ</td>
<td>Whole</td>
<td>Xtext</td>
</tr>
<tr>
<td>------------------</td>
<td>------</td>
<td>-----</td>
<td>-----------</td>
<td>-----</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td><strong>Notation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Textual</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Graphical</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Tabular</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Symbols</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td><strong>Semantics</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model2Text</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Model2Model</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Concrete syntax</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Interpretative</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td><strong>Validation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structural</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Naming</td>
<td>○</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Types</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Programmatic</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

Legend:
- ●: Mandatory
- ○: Optional
- ⊗: Or
<table>
<thead>
<tr>
<th>Feature</th>
<th>Ensō</th>
<th>Más</th>
<th>MetaEdit+</th>
<th>MPS</th>
<th>Onion</th>
<th>Rascal</th>
<th>Spoofax</th>
<th>SugarJ</th>
<th>Whole</th>
<th>Xtext</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing</td>
<td>DSL testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DSL debugging</td>
<td>●</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DSL prog. debugging</td>
<td>●</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Composability</td>
<td>Syntax/views</td>
<td>●</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Validation</td>
<td>●</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Semantics</td>
<td>●</td>
<td></td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Editor services</td>
<td>●</td>
<td></td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Editing mode</td>
<td>Free-form</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Projectional</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td>●</td>
</tr>
</tbody>
</table>

Legend:
- ● Mandatory
- ○ Optional
- ▲ Or
<table>
<thead>
<tr>
<th>Feature</th>
<th>Enso</th>
<th>Más</th>
<th>MetaEdit+</th>
<th>MPS</th>
<th>Onion</th>
<th>Rascal</th>
<th>Spoofax</th>
<th>SugarJ</th>
<th>Whole</th>
<th>Xtext</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSL testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSL debugging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSL prog. debugging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Composability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax/views</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Editor services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Editing mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Free-form</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Projectional</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enso</td>
<td>Más</td>
<td>MetaEdit+</td>
<td>MPS</td>
<td>Onion</td>
<td>Rascal</td>
<td>Spoofax</td>
<td>SugarJ</td>
<td>Whole</td>
<td>Xtext</td>
</tr>
<tr>
<td>------------------</td>
<td>------</td>
<td>-----</td>
<td>-----------</td>
<td>-----</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSL testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSL debugging</td>
<td>⬤</td>
<td>⬤</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSL prog. debugging</td>
<td>⬤</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Composability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax/views</td>
<td>⬤</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Editor services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Editing mode</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Free-form</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td></td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
</tr>
<tr>
<td>Projectional</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntactic services</td>
<td>Enso</td>
<td>Más</td>
<td>MetaEdit+</td>
<td>MPS</td>
<td>Onion</td>
<td>Rascal</td>
<td>Spoofax</td>
<td>SugarJ</td>
<td>Whole</td>
<td>Xtext</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------</td>
<td>-----</td>
<td>-----------</td>
<td>-----</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>Highlighting</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Outline</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Folding</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Syntactic completion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Diff</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Auto formatting</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Semantic services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reference resolution</td>
<td></td>
<td></td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Semantic completion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Refactoring</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Error marking</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Quick fixes</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Origin tracking</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>Live translation</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
</tbody>
</table>

- **Mandatory**
- **Optional**
- **Or**
<table>
<thead>
<tr>
<th>Syntactic services</th>
<th>Ensō</th>
<th>Más</th>
<th>MetaEdit+</th>
<th>MPS</th>
<th>Onion</th>
<th>Rascal</th>
<th>Spoofax</th>
<th>SugarJ</th>
<th>Whole</th>
<th>Xtext</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highlighting</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
</tr>
<tr>
<td>Outline</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Folding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntactic completion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auto formatting</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reference resolution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic completion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Refactoring</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
<td>🔄</td>
</tr>
<tr>
<td>Error marking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quick fixes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Origin tracking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Live translation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Enso</td>
<td>Más</td>
<td>MetaEdit+</td>
<td>MPS</td>
<td>Onion</td>
<td>Rascal</td>
<td>Spoofax</td>
<td>SugarJ</td>
<td>Whole</td>
<td>Xtext</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------</td>
<td>-----</td>
<td>-----------</td>
<td>-----</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>Syntactic services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Highlighting</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Outline</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Folding</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Syntactic completion</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Diff</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Auto formatting</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Semantic services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reference resolution</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Semantic completion</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Refactoring</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Error marking</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Quick fixes</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Origin tracking</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Live translation</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Textual</td>
<td>Graphical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Enso</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>● Más</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>● MetaEdit+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>● MPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>● Onion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>● Rascal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>● Spoofax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>● SugarJ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>● Whole</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>● Xtext</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Academic</td>
<td>Industry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ensō</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Más</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MetaEdit+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Onion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rascal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spoofax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SugarJ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Whole</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xtext</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tabular</td>
<td>Graphical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ensō</td>
<td>○</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Más</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MetaEdit+</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPS</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Onion</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rascal</td>
<td>○</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spoofax</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SugarJ</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Whole</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xtext</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

used for real

vs.

research/prototype
<table>
<thead>
<tr>
<th>Textual</th>
<th>Graphical</th>
</tr>
</thead>
<tbody>
<tr>
<td>enso</td>
<td></td>
</tr>
<tr>
<td>mas</td>
<td></td>
</tr>
<tr>
<td>MetaEdit+</td>
<td></td>
</tr>
<tr>
<td>MPS</td>
<td></td>
</tr>
<tr>
<td>onion</td>
<td></td>
</tr>
<tr>
<td>rascal</td>
<td></td>
</tr>
<tr>
<td>spoofax</td>
<td></td>
</tr>
<tr>
<td>sugarj</td>
<td></td>
</tr>
<tr>
<td>whole</td>
<td></td>
</tr>
<tr>
<td>xtext</td>
<td></td>
</tr>
</tbody>
</table>

commercial vs. open source
more
(not part of LWC)

+ JetBrains Nitra
  oomega
EMFText
Eclipse Sirius
Eco
worth looking at

+ JetBrains Nitra
  oomega
  EMFText
  Eclipse Sirius
  Eco
this talk

JetBrains
Nitra
oomega
EMFText
Eclipse
Sirius
Eco
Xtest
eclipse
itemis
2

Notational Flexibility
Projectional Editing
[Projectional Editing]

Parsing

![Diagram showing the process of parsing from concrete syntax to abstract syntax tree.]

Projectional Editing

![Diagram showing the process of projectional editing from abstract syntax tree back to concrete syntax.]
[Projectional Editing]
Syntactic Flexibility

Regular Code/Text

Mathematical

Tables

Graphical
### Projectional Editing

#### Syntactic Flexibility

**Regular Code/Text**

```c
// A documentation comment with references

to @arg(data) and @arg(dataLen)

void aSummingFunction(int8[] data, int8 dataLen) {
    int16 sum;
    for (int8 i = 0; i < dataLen; i++) {
        sum += data[i];
    }
} aSummingFunction (function)
```

**Mathematical**

```latex
\text{double\ midnight2}(\text{int32\ a,\ int32\ b,\ int32\ c})\ \{\nonumber\\
\quad \text{return} \quad \frac{-b + \sqrt{b^2 - \sum_{i=1}^{4} a \times c}}{2 \times a}; \\
\}\ \text{midnight2\ (function)}
```

**Tables**

```c
int16\ \text{decide}(\text{int8\ spd,\ int8\ alt})\ \{\nonumber\\
\quad \text{return} \quad \begin{array}{cc}
\text{spd > 0} & \text{spd > 100} \\
\text{alt < 0} & 1 & 1 \\
\text{alt = 0} & 10 & 20 \\
\text{alt > 0} & 30 & 40 \\
\text{alt > 100} & 50 & 60 \\
\end{array}; \\
\}\ \text{decide\ (function)}
```

**Graphical**

```
\text{Cst.Customer}\quad\text{cust\ 1}
```

```
\text{Contract}
\quad\text{starts: date}
\quad\text{ends: date}
\quad\text{tarf\ 1}
```

```
\text{Tariff\ attributes}
```
[Business Application]

Insurance Configuration Tool

Contract
- starts: date
- ends: date

Tariff
- applicableTariff 1
- attributes

Customer.Customer
- customer 1

[BillingRegion]
- code [key]: string
- name: string
- baseMinPrice: float
- maxRebateFactor: float
## Business Application

### Insurance Configuration Tool

Core Data Default Regions for entity BillingRegion

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Base</th>
<th>Min Price</th>
<th>Max</th>
<th>Rebate</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW</td>
<td>Baden Württemberg</td>
<td>0.20</td>
<td></td>
<td></td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>BY</td>
<td>Bayern</td>
<td>0.20</td>
<td></td>
<td></td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>BE</td>
<td>Berlin</td>
<td>0.15</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>BB</td>
<td>Brandenburg</td>
<td>0.10</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>HB</td>
<td>Bremen</td>
<td>0.20</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>HH</td>
<td>Hamburg</td>
<td>0.15</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>HE</td>
<td>Hessen</td>
<td>0.15</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>MV</td>
<td>Mecklenburg-Vorpommern</td>
<td>0.10</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>NI</td>
<td>Niedersachsen</td>
<td>0.15</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>NW</td>
<td>Nordrhein-Westfalen</td>
<td>0.15</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>RP</td>
<td>Rheinland-Pfalz</td>
<td>0.15</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>Saarland</td>
<td>0.15</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>SN</td>
<td>Sachsen</td>
<td>0.10</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>Sachsen-Anhalt</td>
<td>0.10</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>SH</td>
<td>Schleswig-Holstein</td>
<td>0.15</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>TH</td>
<td>Thüringen</td>
<td>0.10</td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
</tbody>
</table>
### Business Application: Insurance Configuration Tool

#### Rule Set Type: DemoRuleSetType

<table>
<thead>
<tr>
<th>Variables</th>
<th>Parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRMI      : int</td>
<td>&lt;no parent&gt;</td>
</tr>
<tr>
<td>FR        : int</td>
<td>&lt;no variables&gt;</td>
</tr>
<tr>
<td>NN        : int</td>
<td>&lt;no variables&gt;</td>
</tr>
<tr>
<td>TT        : int</td>
<td>Libraries</td>
</tr>
<tr>
<td>J         : int</td>
<td>Standard</td>
</tr>
<tr>
<td>A3        : int</td>
<td>Extra</td>
</tr>
<tr>
<td>G3        : int</td>
<td></td>
</tr>
<tr>
<td>ANUI      : int</td>
<td></td>
</tr>
<tr>
<td>X         : int</td>
<td></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Variables</th>
<th>Parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>person : Person</td>
<td>&lt;no business objects&gt;</td>
</tr>
</tbody>
</table>

---

Business objects:

- Person

---

Business objects:

- <no business objects>
bloedverwanten : lijst van Burgers zijn gedefinieerd als {
    Een bloedverwant is een Burger die
    bloedverwant in rechte lijn is of die
    bloedverwant in tweede graad zijlijn is
    Einde declaratie
}
bloedverwanten in rechte lijn : lijst van Burgers zijn gedefinieerd als {
    Een bloedverwant in rechte lijn is een Burger die
    nakomeling is of die
    voorouder is
    Einde declaratie
}
bloedverwanten in tweede graad zijlijn : lijst van Burgers zijn gedefinieerd als {
    Een bloedverwant in tweede graad zijlijn is een ouder.kind met
    ouder.kind ongelijk het actuele voorkomen
    Einde declaratie
    ' dus: broer of zus (incl. erkend kind van ouder)
}
bloed- of aanverwanten in rechte lijn : lijst van Burgers zijn gedefinieerd als {
    Een bloed- of aanverwant in rechte lijn is een Burger die
    bloedverwant in rechte lijn is of die
    aanverwant in rechte lijn is
    Einde declaratie
}
CampagneLab’s NYoSh

Biological Data

go by web read sample 849–WT {
  tag: NGBOXSJ
  number of reads: 29230382
  is paired: false
  is bisulfite: false
  is ready to align: true
  min length: 101
  max length: 101
  organism: mus_musculus
  platform: Illumina
  lib protocol preserve strand: true
}
int32 sumUpIntArray(int32[] arr, int32 size) {
    return \sum_{i=0}^{size} \text{arr}[i];
} sumUpIntArray (function)

int32 averageIntArray(int32[] arr, int32 size) {
    return \frac{\sum_{i=0}^{size} \text{arr}[i]}{size};
} averageIntArray (function)

double midnight1(int32 a, int32 b, int32 c) {
    return \frac{-b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a};
} midnight1 (function)

double midnight2(int32 a, int32 b, int32 c) {
    return \frac{-b + \sqrt{b^2 - \sum_{i=1}^{4} a \cdot c}}{2 \cdot a};
} midnight2 (function)

double sumOfProductsOfLogs(int32[] arr, int32 size) {
    return \sum_{k=0}^{size} \frac{\log_2 \text{arr}[i]}{2};
} sumOfProductsOfLogs (function)
```c
[checked]
exported statemachine FlightAnalyzer initial = beforeFlight {

    next(Trackpoint* tp) {
        [tp->alt > 0 m] -> airborne
        [tp->alt == 0 m && tp->speed == 0 mps] -> crashed
        [tp->speed > 200 mps && tp->speed > 0 mps] -> landing
        [tp->speed > 100 mps && tp->speed <= 200 mps && tp->alt == 0 m] -> airborne
        { points += VERY_HIGH_SPEED; }
        [tp->speed == 0 mps] -> landed
        [tp->speed > 0 mps] -> landing
        { points--; }
        [ ] -> beforeFlight
    }

    reset() {
        [ ] -> beforeFlight
    }
}```
3 Language Modularity
[Projectional Editing]
Language Composition

- Separate Files
  - Type System
  - Transformation
  - Constraints

- In One File
  - Type System
  - Transformation
  - Constraints
  - Syntax
  - IDE

50+ extensions to C
10+ extensions to requirements lang.
[Projectional Editing]

Modular Language Composition

No change to definition of \( L_1 \) or \( L_2 \) in order to use them together.

Embedding

\[ L_{\text{Host}} + L_{\text{Adapt}} + L_{\text{Emb}} = \]

Extension

\[ L_{\text{Base}} + L_{\text{Ext}} = \]

Extension Composition

\[ L_{\text{Base}} + L_{\text{Ext1}} + L_{\text{Ext2}} = \]
[Projectional Editing]
Growing a Language towards Domain

Domains Naturally form a hierarchy. Languages can „grow“ along.
[Projectional Editing]
Growing a Language towards Domain

automotive
embedded software

avionics

all programs
Growing a Language towards Domain

automotive

embedded software

avionics

all programs

[Projectional Editing]

MBEDDR + AUTOSAR

MBEDDR

C
[Projectional Editing]

DSL Styles and Trade-Offs

Classic External DSL

Nicely analyzable (if done right)

Great IDE Support

Limited Expressiveness
[Projectional Editing]

DSL Styles and Trade-Offs

Classic External DSL  External + Expressions

Can still be nicely analyzable
Great IDE Support
Better expressiveness b/c Expressions
[Projectional Editing]
DSL Styles and Trade-Offs

- Classic External DSL
- External + Expressions
- Prog Lang Extension

Harder to analyze (GPL influence)
Great IDE Support
Very expressive (Fallback to GPL)
[Projectional Editing]
DSL Styles and Trade-Offs

Classic External DSL
External + Expressions
Prog Lang Extension

Great IDE Support
[Projectional Editing]
DSL Styles and Trade-Offs

Great IDE Support
[Projectional Editing]

DSL Styles and Trade-Offs

- Great IDE Support
  - Built via Meta Programming
  - Built with LWB
- Internal DSL
  - Ruby
  - Scala
  - Xtend

Prog Lang Extension
[Projectional Editing]

DSL Styles and Trade-Offs

Proxg Lang

Extension

Built with LWB

Great IDE Support

Built via Meta Programming

Type System

Compiler

IDE

Ruby

Scala

Xtend

Internal DSL
[Projectional Editing]

DSL Styles and Trade-Offs

Potentially similar syntax, but quite different UX
Specification Languages
From a Talk
Lessons Learned from Building mbeddr
No direct support for detecting semantic interactions between languages
Debugger definition separate from generator; leads to duplication
many aspects of language definition too „procedural" and hence hard to analyze.
Declarative!
templates

Definition.Function = <
   <Type> <ID>((<Param*>; separator="","">) { <Statement*; separator="\n">

>

Statement.If = <
   if(<Exp>) <Statement>
   else <Statement>
>

Statement.Return = <return <Exp>;>

Exp.Add = <<Exp> + <Exp>>

Exp.Var = <<ID>>
binding rules

Param(t, name) :
    defines Variable name

Var(name) :
    refers to Variable name

Function(t, name, param*, s) :
    defines Function name
    scopes Variable

Call(name, exp*) :
    refers to Function name
type rules

Call(name, exp*) : t
where definition of
    name : (t*, t)
    and exp* : t_exp*
    and t_exp* == t*

Everything is wonderful. Right?
Well, not quite.
Declarative?

Express real, „messy“ languages?
Can „normal“ people use it?
Procedural escapes?
Do they destroy the whole story?
Not clear. Needs Research.

→ Spoofax@TU Delft
5 LWBs on the Web
Of course ...
Language Workbench
(Martin Fowler)

Freely define languages and integrate them
Freely define languages

... but demonstrate editing technology.

web-based IDEs or modeling tools do not count ...

Language Workbench
(Martin Fowler)
Language Workbench
(Martin Fowler)

Freely define languages
and: initial tools will only use languages in the browser, not define them.
LWBs on the Web

Xtext and MPS teams are working on it

It will be months to years until these are finished/robust.

Other web-LWBs (prototypes) exist: http://languageworkbenches.net
Editor Widget in the Browser
JVM on the Server
Initial Services:
  Syntax Highlighting, Validation, Code Completion, Hovering, Formatting
More Services based on User Input
Use existing Xtext lang def
Q3 2015 earliest

Some Details:

vimeo.com/53935743
xrobots.itemis.de
Projectional Editor in Browser
  incl. Hybrid Editing
No Server Component
No Dependency on MPS
Projectional Editor in Browser
incl. Hybrid Editing

No Server Component
No Dependency on MPS

Initial Services:
Editing, Syntax Highlighting,
Code Completion, Validation
Initially not based on MPS langs

New language stack to be back-ported eventually

Middle/Late 2015

Longer Term:

Full Language Stack (Scopes, TS)

„OCL-like“ as default base language

Modular Language Aspects
LWBs on the Web

It will be years until these are as robust and feature-rich as rich client variants.
Or maybe ...

Make Web-IDEs Extensible?
Miscellaneous Trends
UI vs. Infrastructure

Keep the Client, Webify Infrastructure?
UI vs. Infrastructure

Monto:
Disintegrated DE, IDE Middleware
Python-based
IDE Independence

Xtext on IntelliJ
JetBrains Nitra

DSLs for Visual Studio

```java
syntax NitraType
{
    Typing() : NType * bool;
    | PredefinedType
    |   override Typing
    |      def value = match (GetText(this.PredefinedType))
    |          | "void" => NPredefinedType.Void
    |          | "int" => NPredefinedType.Int
    |          | "string" => NPredefinedType.String
    |          | _ => assert3(false)
    |} (NType.Predefined(value), false);

    | QualifiedType = (NitraTypeName; ".")+
```
Why you should care
Language Engineering solves Real Problems!
Language Engineering solves Real Problems!

Improved Robustness, Testability, Maintainability and Performance in Embedded Software

Preliminary Experience of using mbeddr for Developing Embedded Software

Language Engineering solves Real Problems!

Improved Security for C-based Software.

Improving Software Security using Language Engineering and mbeddr C

Language Engineering solves Real Problems!

Better Integration of Stakeholders and more Efficiency in Business Applications.

Projecting a Modular Future

http://voelter.de/data/pub/projectingModuleFuture.pdf
Language Engineering solves Real Problems!

More formal, checkable and hence more useful requirements and specifications.

Projecting a Modular Future

http://voelter.de/data/pub/projectingModuleFuture.pdf
Thank you!