Language Definition and Extension with MPS

Different Worlds
Programming Tools ≠ Modeling Tools

Different Worlds
Mix Models and Programs

Different Worlds
Modeling Tool ≠ Modeling Tool

Different Worlds
Mix Models and Programs
AST Navigation & Query
Different Worlds
Mix Models and Programs
AST Navigation & Query
Code Constraints

Why the difference?

History?

Why the difference?

It is time for ...

... a Different Perspective
We don’t want to model, we want to program!

... at different levels of abstraction
... from different viewpoints
... integrated!

We don’t want to model, we want to program!

... with different degrees of domain-specificity
... with suitable notations
... with suitable expressiveness

We don’t want to model, we want to program!

And always: precise and tool processable
Syntax primarily textual

When a graphical notation is better, you can visualize.

Syntax primarily textual with more symbols think: mathematics

Syntax primarily textual sometimes box&line style
Modeling as Programming

... (Mostly) Textual Notations
... Concrete Syntax Storage
... (Fancy) ASCII Editors
... Read-Only Visualizations

Programming as Modeling

... (Mostly) Graphical Notations
... Abstract Syntax Storage
... Projecting Editors
... Different editable views for model
Programming as Modeling

... (Mostly) Graphical Any kind of Notations
... Abstract Syntax Storage
... Projecting Editors
... Different editable views for model

Language Composition

There's no parsing.
Unique Language Element Identity.
Unlimited language composition.

Flexible Notations

Textual like ASCII
Graphical box & line
Semi-Graphical mathematical

treated the same can be mixed

Automatic IDE Extension

tool support is inherent for languages build with projectional tools
language definition implies IDE definition

Multiple Notations

... for the same concepts e.g. in different contexts or for different tasks

Tree Editing

... is different from editing text
... try to make it feel like text
... takes some getting used to
but: for more flexible notations a more general editing paradigm is needed
Infrastructure Integration

... storage is not text
... diff/merge must be in tool
... existing text tools don’t work

Proprietary Tools

... no standards
... no interop

Build new standalone DSLs
Build new standalone DSLs
Build DSLs that reuse parts of other languages

licensed under
Apache 2.0
released in
Q2 2009
currently
1.5
Build new **standalone** DSLs
Build DSLs that **reuse** parts of other languages

(MPS comes with **BaseLanguage**)
**extend** base language

Build new **standalone** DSLs
Build DSLs that **reuse** parts of other languages

(MPS comes with **BaseLanguage**)
**extend** base language
build DSLs that **reuse** parts of **BaseLanguage**

**DEMO**

Building/Extending DSLs with JetBrains MPS

**A vision for Programming**
Big Language?

with many first class concepts!

Small Language?

with a few, orthogonal and powerful concepts

Modular Language

with many optional, composable concepts

Modular Language

Like frameworks and libraries,

Modular Language

Like frameworks and libraries,

but with syntax and IDE support
Not a new idea...

Growing A Language

(Guy L. Steele)

THE END.