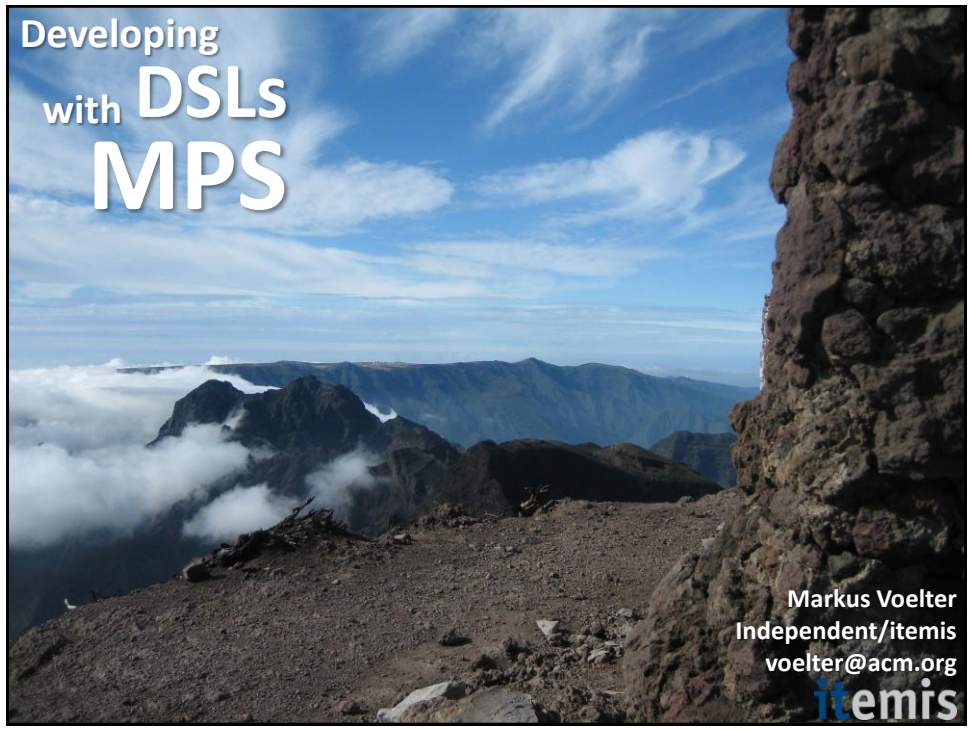


We don‘t want to model, we want to program!

We don‘t want to
model,
we want to
program!

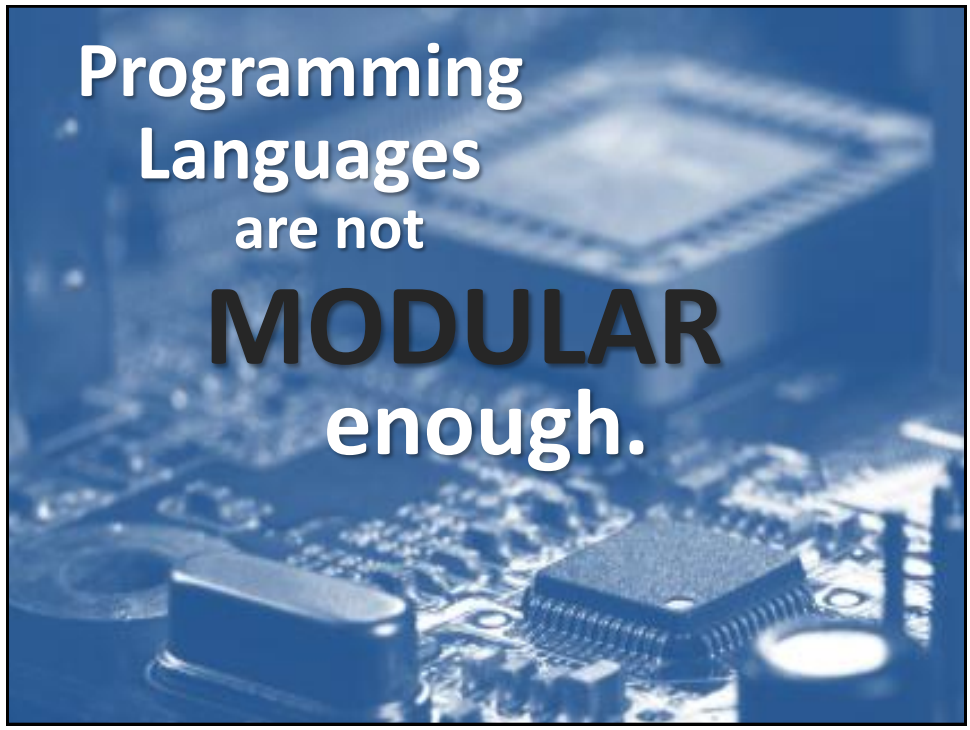
… at different levels of **abstaction**
… from different **viewpoints**
… **integrated**!

We don‘t want to
model,
we want to
program!

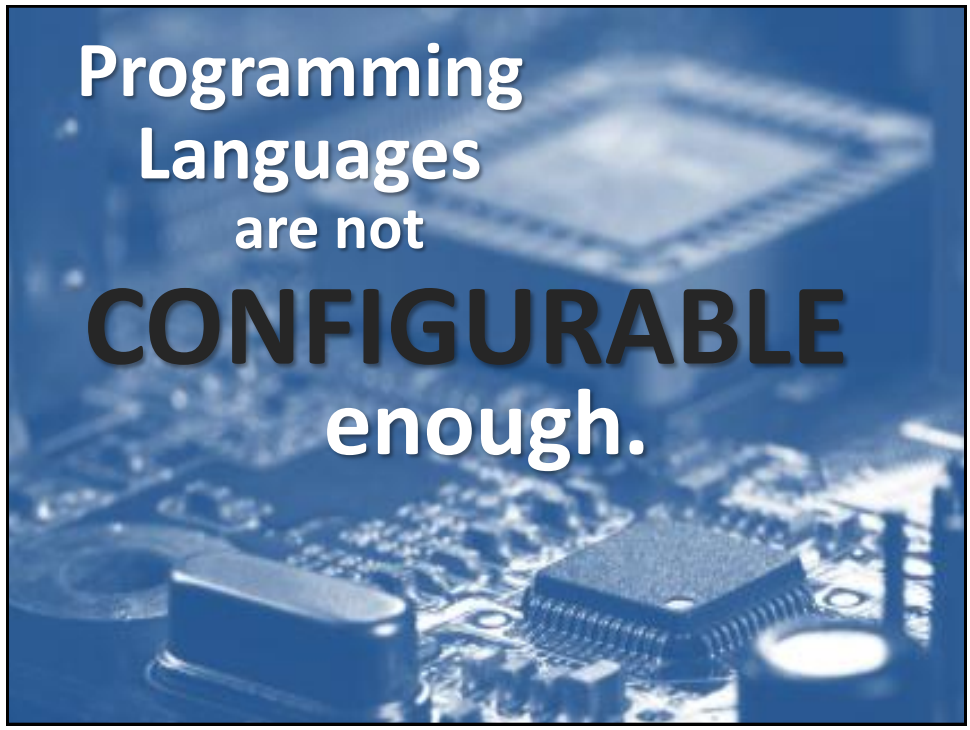
… with different degrees of **domain-specificity**
… with suitable **notations**
… with suitable **expressiveness**

We don‘t want to
model,
we want to
program!

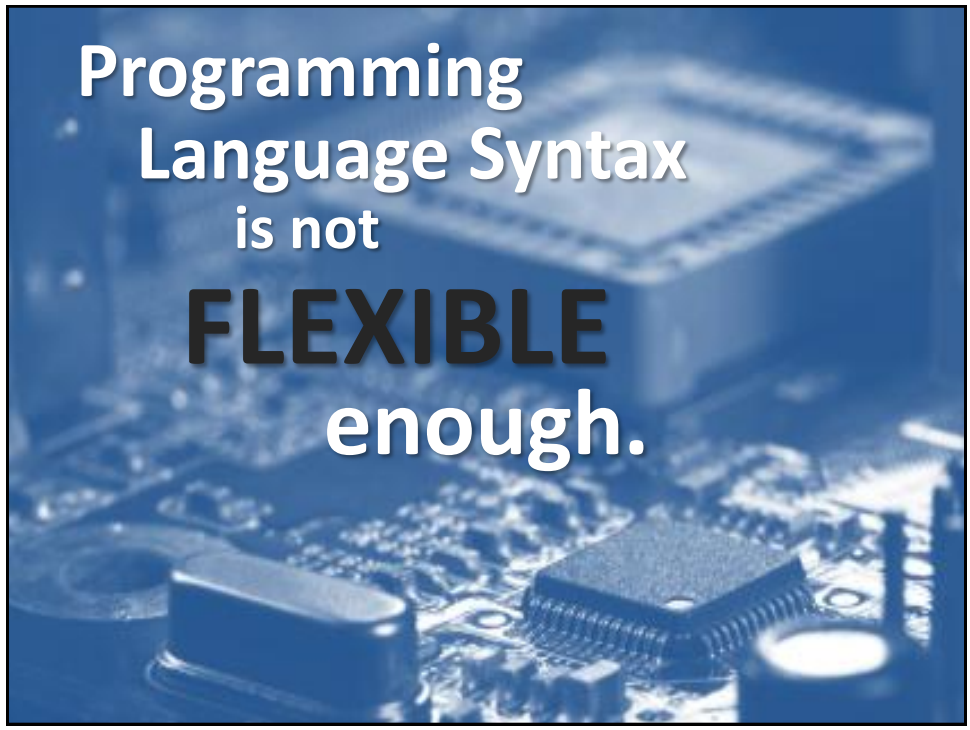
And always:
precise and tool processable


Programming
Languages
are not
MODULAR
enough.

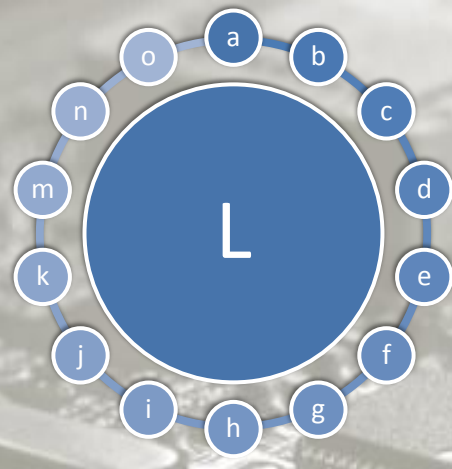
Programming
Languages
are not
COMPOSABLE
enough.


Programming
Languages
are not
CONFIGURABLE
enough.

Programming
Languages
are not
ADAPTABLE
enough.

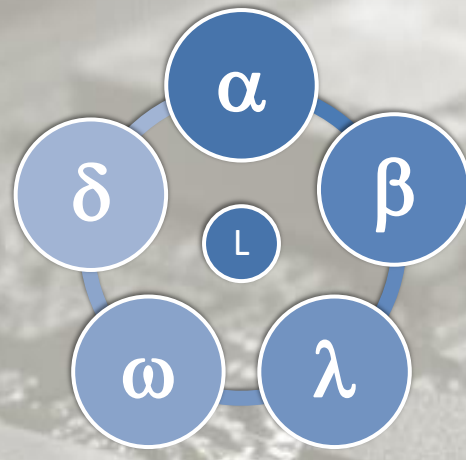

Programming
Language Syntax
is not
FLEXIBLE
enough.

Big Language?



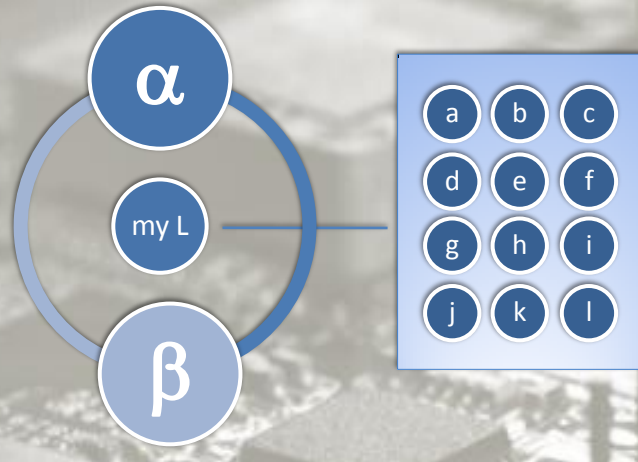
with many first class concepts!

Small Language?



with a few, orthogonal and poweful concepts

Modular Language



with many optional, composable concepts

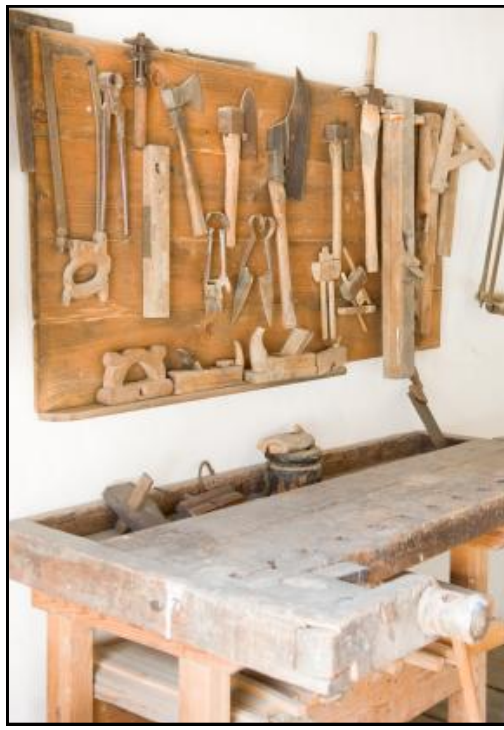
Modular Language

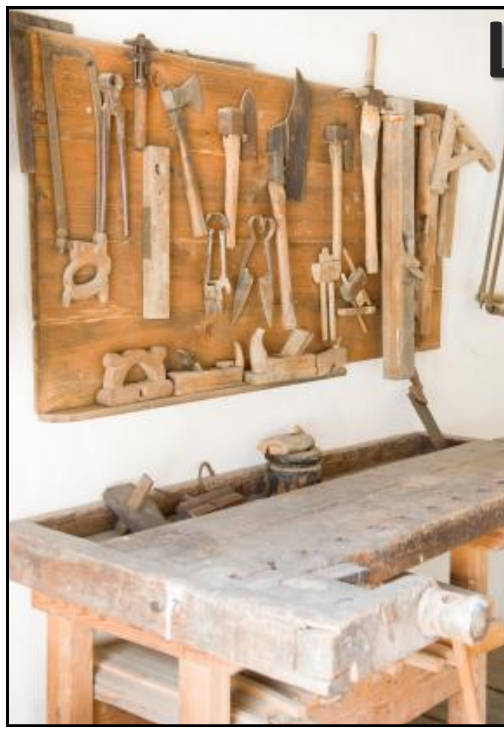
Like frameworks and libraries,

Modular Language

Like frameworks
and libraries,

but with syntax
and IDE support





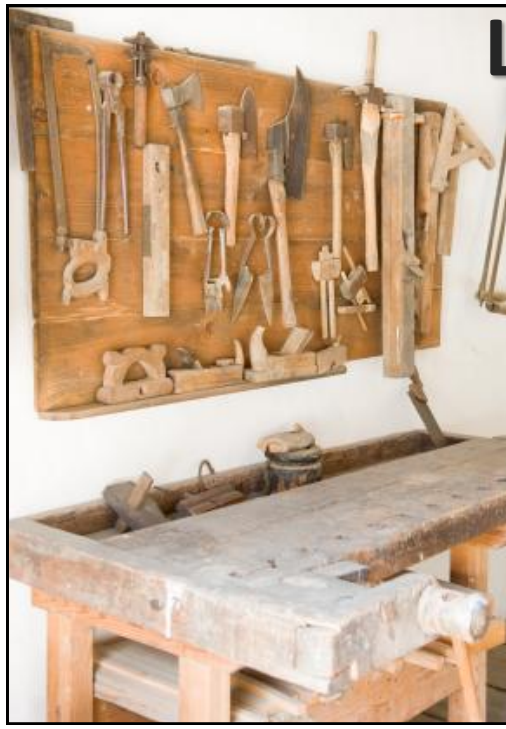
Language Workbench

(Martin Fowler)

Language Workbench

(Martin Fowler)

Freely
define
languages and
integrate
them



Language Workbench

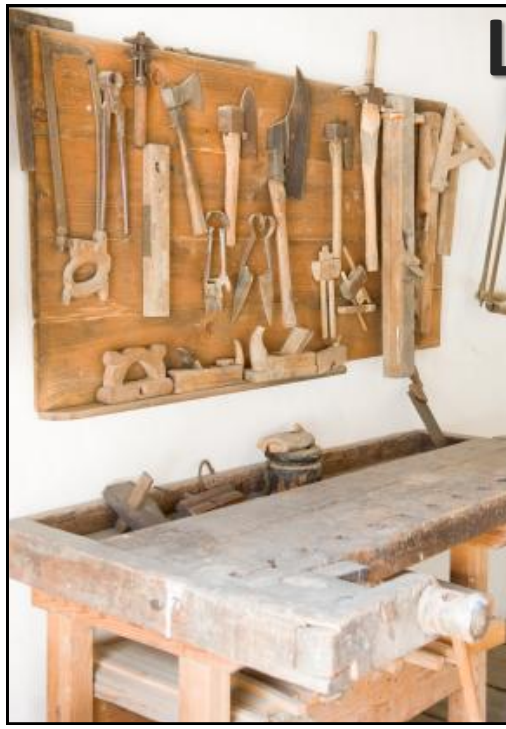
(Martin Fowler)

use
persistent?
abstract?
representation

Language Workbench

(Martin Fowler)

language ::=
schema
+ editors
+ generators



Language Workbench

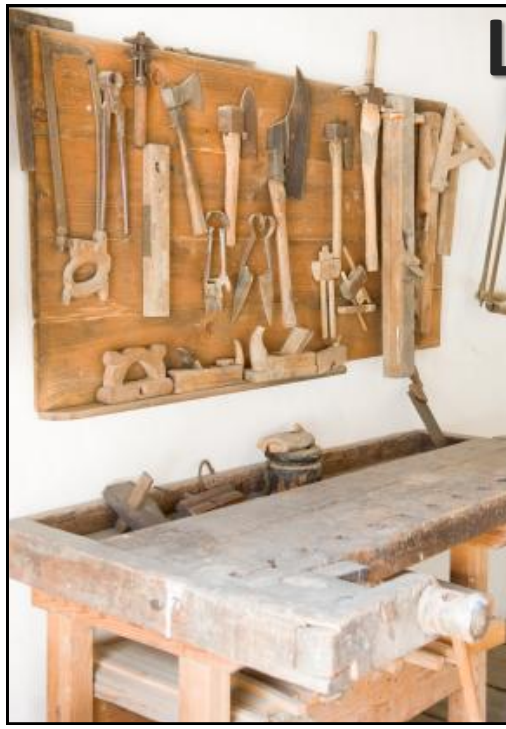
(Martin Fowler)

projectional editing?

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
„classical“ and modeling

Syntax primarily textual

Syntax primarily textual

with more symbols

think: mathematics

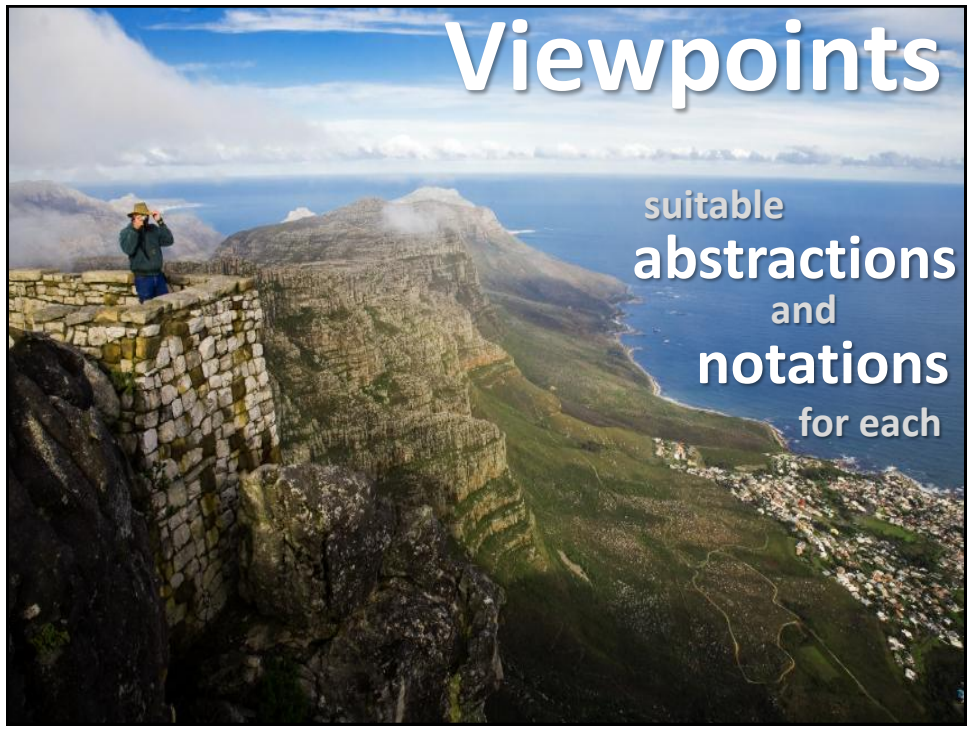
Syntax primarily textual

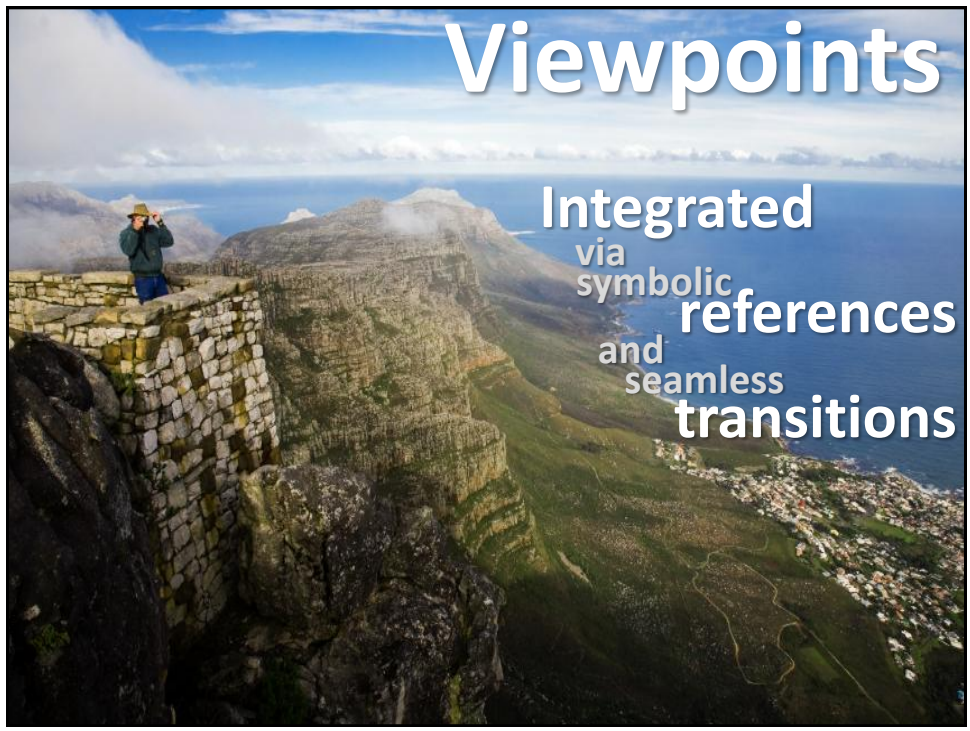
sometimes box&line style

Syntax primarily textual

sophisticated visualizations

Viewpoints


Viewpoints
suitable
abstractions
and
notations
for each


Viewpoints
Integrated
via
symbolic
references
and
seamless
transitions

Viewpoints
Business

custom
purpose-built

create/include

Viewpoints
Business

Custom
Notations

real
business
expert integration

Viewpoints
Business

Custom Notations

real business expert integration

But that's another talk…

Viewpoints
Technical

predefined library

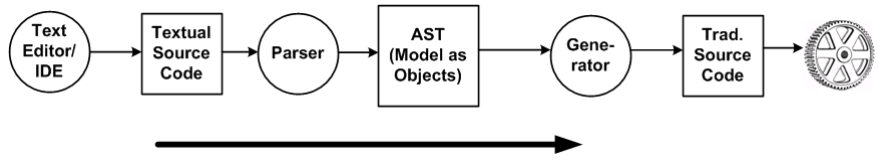
configure

Parser-based

text

… to tree

… to text



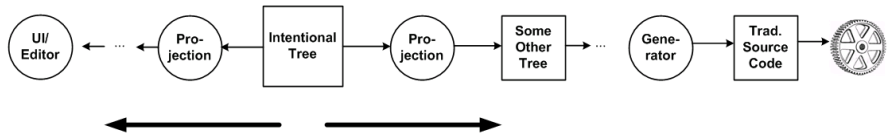
Projectional

tree

… to text-lookalike (editor)

… to other trees … [*]

… to text



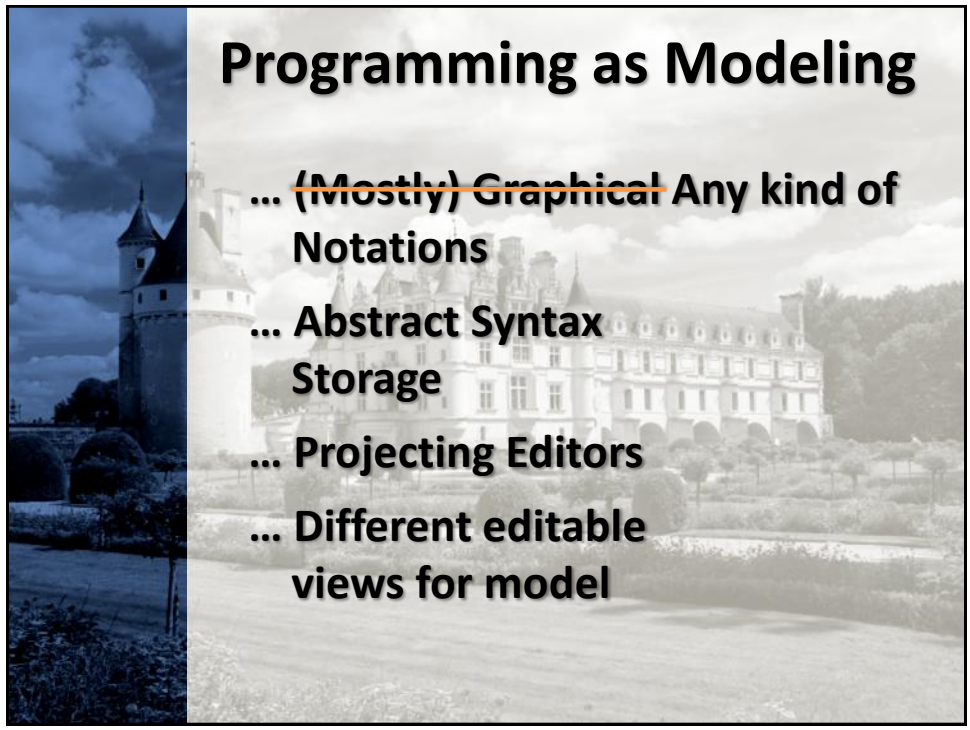
Programming as Modeling

… (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

Custom
Syntax

Graphical
Textual
Symbolic++

IDE
Support

Teamwork
Debugging
Custom Editors

Complete Symbolic Integration

Goto Def
Find Refs
Refactoring

Language Composition

There‘s no parsing.

Unique Language Element Identity.

Unlimted 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

Partial
Projections

… different views
… for different roles/people
… only a particular variant

Storage
!= Schema

… store arbitraty meta data
- change log
- conflicting information
- variability annotations

… independent of language schema!

… „aspects“, overlay

Live
Programs

think: spreadsheet

a change to one part of program can lead to (dependent) changes in other parts

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

also do…

IntelliJ IDEA
Resharper

released in

Q3 2009

currently

V 1.5.1

soon…

V 2.0

JetBRAINS

Open Source

licensed under

Apache 2.0

Build new standalone DSLs

Build new standalone DSLs

Build DSLs that reuse parts of other languages

Build new standalone DSLs

Build DSLs that reuse parts of other languages

(MPS comes with BaseLanguage) Java++

extend base language

Build new standalone DSLs

Build DSLs that reuse parts of other languages

(MPS comes with BaseLanguage) Java++

extend base language

build DSLs that reuse parts of BaseLanguage

Language Extension Example

Language Extension Example

Old

```
ReadWriteLock l = …
l.readLock().lock();
try {
  //code
} finally {
  l.readLock().unlock();
}
```

Language Extension Example

Old

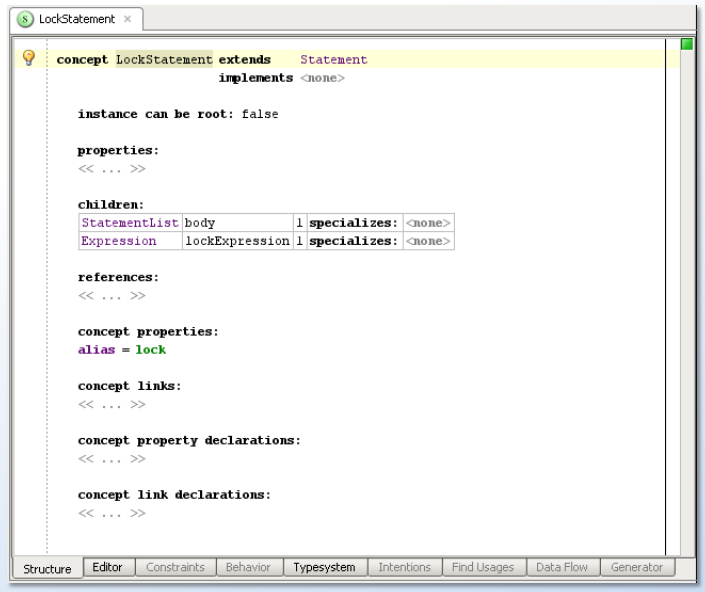
```
ReadWriteLock l = …
l.readLock().lock();
try {
  //code
} finally {
  l.readLock().unlock();
}
```

New

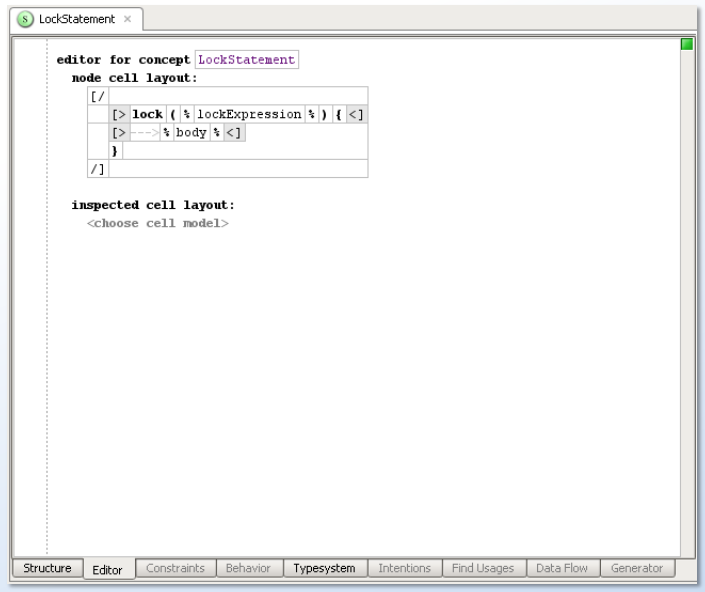
```
ReadWriteLock l = …
lock (l) {
  //code
}
```

Structure ◆ Editor ◆ Typesystem ◆ Generator

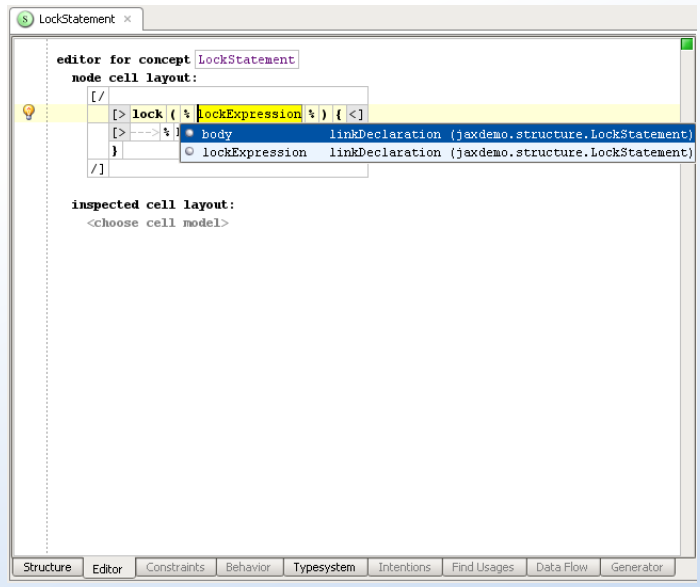
Structure ◆ Editor ◆ Typesystem ◆ Generator



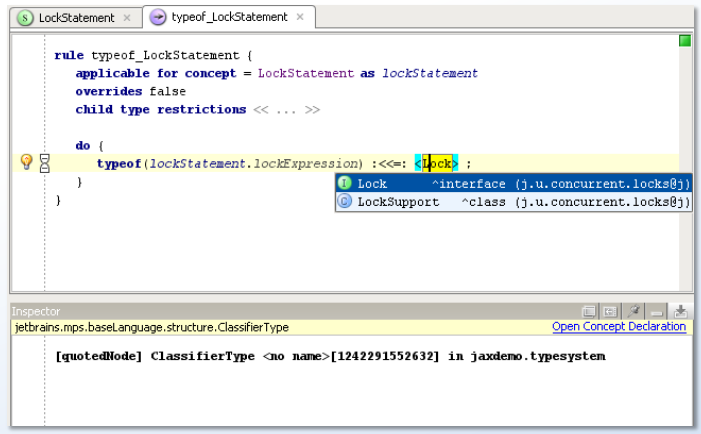
Structure ◆ Editor ◆ Typesystem ◆ Generator



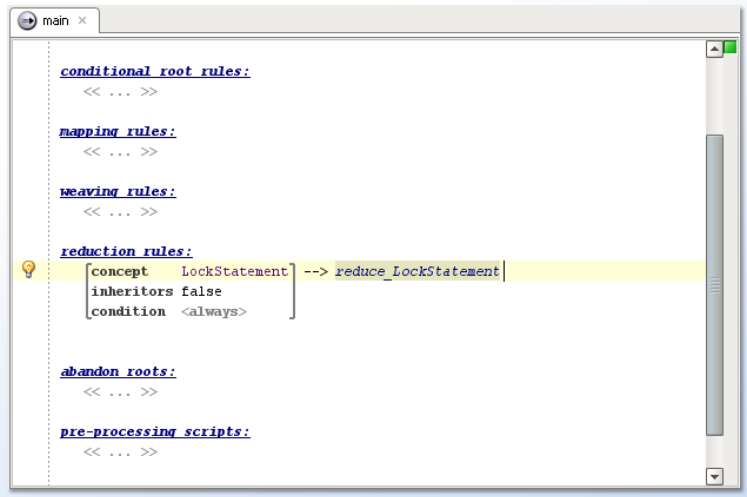
Structure ◆ Editor ◆ Typesystem ◆ Generator



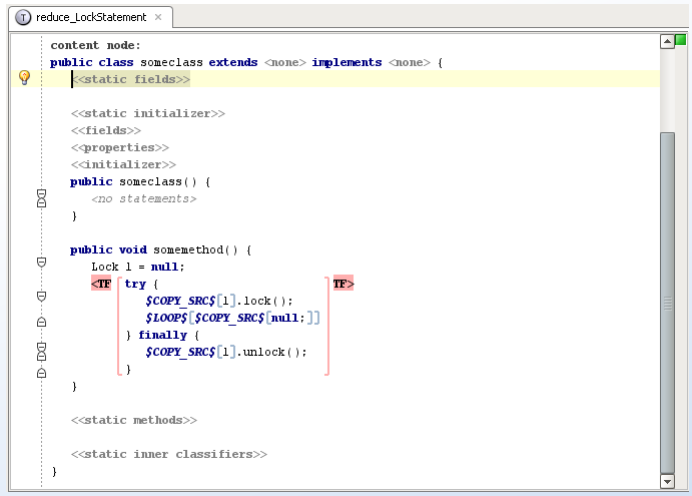
Structure ◆ Editor ◆ Typesystem ◆ Generator



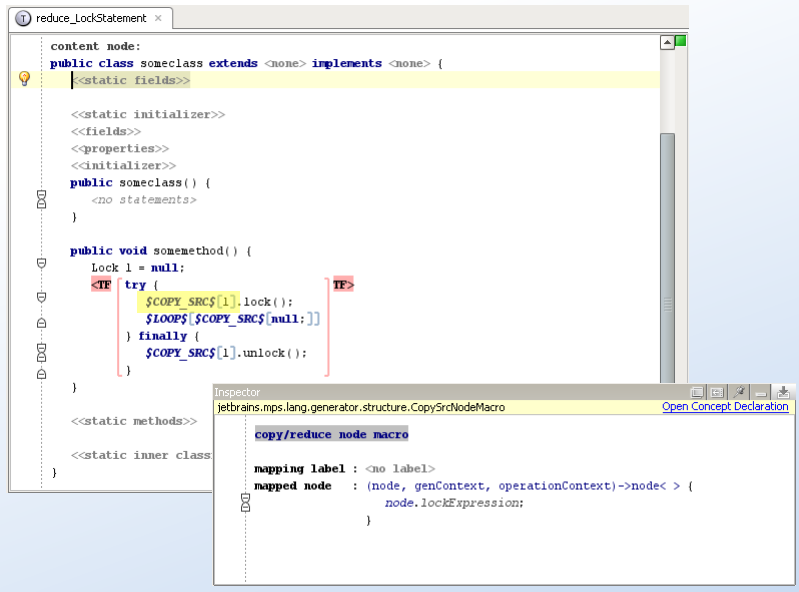
Structure ◆ Editor ◆ Typesystem ◆ Generator



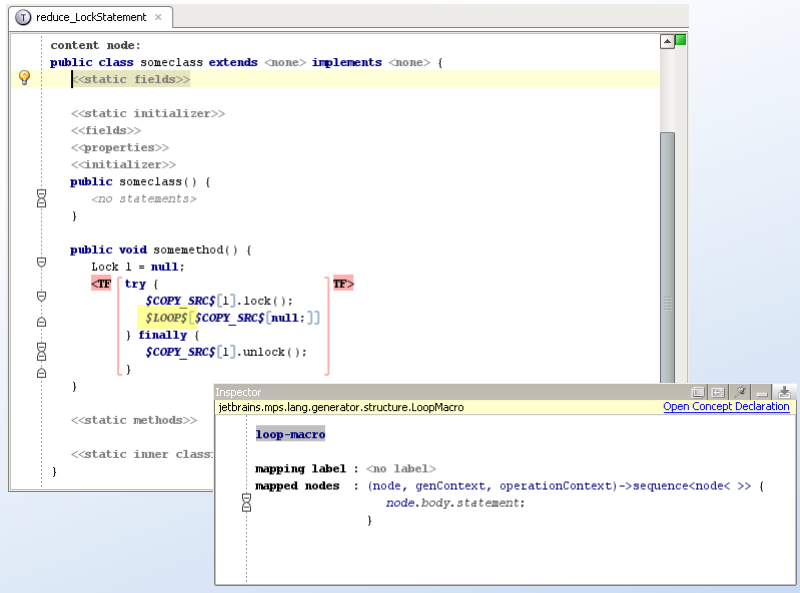
Structure ◆ Editor ◆ Typesystem ◆ Generator



Structure ◆ Editor ◆ Typesystem ◆ Generator



Structure ◆ Editor ◆ Typesystem ◆ Generator



What is planned to be in 2.0?

Tables + Matrixes in the Editor
Diagrams (Box & Line)
Several (separate) Editors per Concept
A Rich Text documentation language
(with embeddable model elements)
Better Build Framework
Stand-Alone DSL Editors
Many Many Fixes :-)

What is planned to be in 2.0?

See also

blogs.jetbrains.com/mps/2010/12/the-mps-future-series-part-1-the-editor

blogs.jetbrains.com/mps/2010/12/the-mps-future-series-part-2-documentation-modularization-et-al

Resources

Website - http://www.jetbrains.com/mps/
User's Guide - http://confluence.jetbrains.net/display/MPSD1/MPS+User's+Guide
Support Forum - http://devnet.jetbrains.net/community/mps

Source Access - http://blogs.jetbrains.com/mps/2010/08/mps-source-code-repository-is-publicly-available-now/
Issue Tracker - http://youtrack.jetbrains.net/dashboard

MPS Tutorial - http://www.jetbrains.com/mps/docs/tutorial.html
LWC 11 Tutorial - http://code.google.com/p/mps-lwc11/wiki/GettingStarted

The End.
.coordinates
web www.voelter.de
email voelter@acm.org
skype schogglad
twitter markusvoelter
xing http://www.xing.com/profile/Markus_Voelter
linkedin http://www.linkedin.com/pub/0/377/a31
itemis