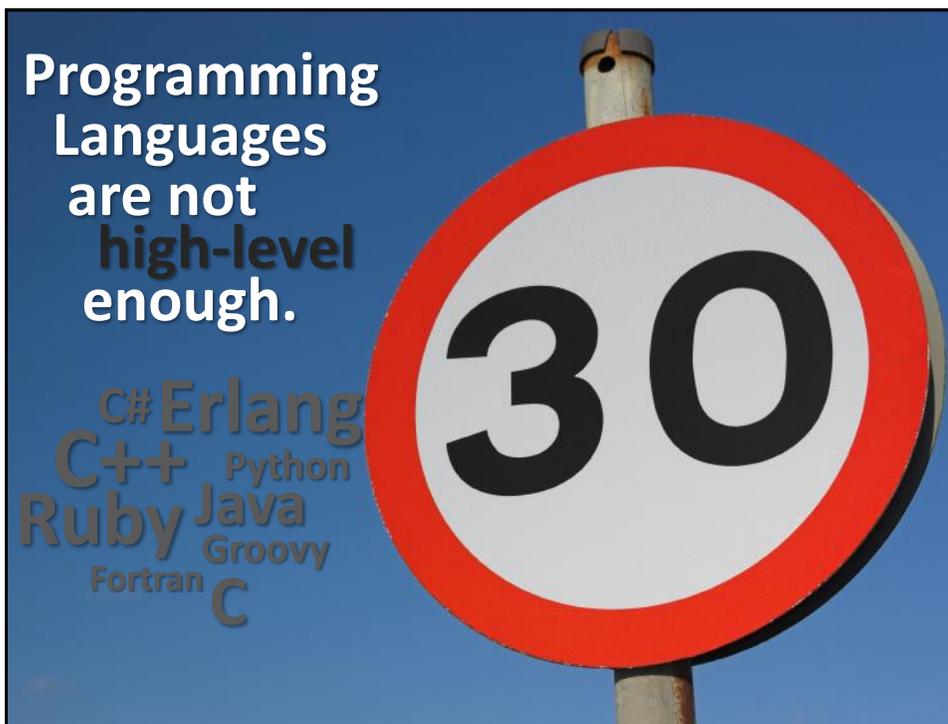


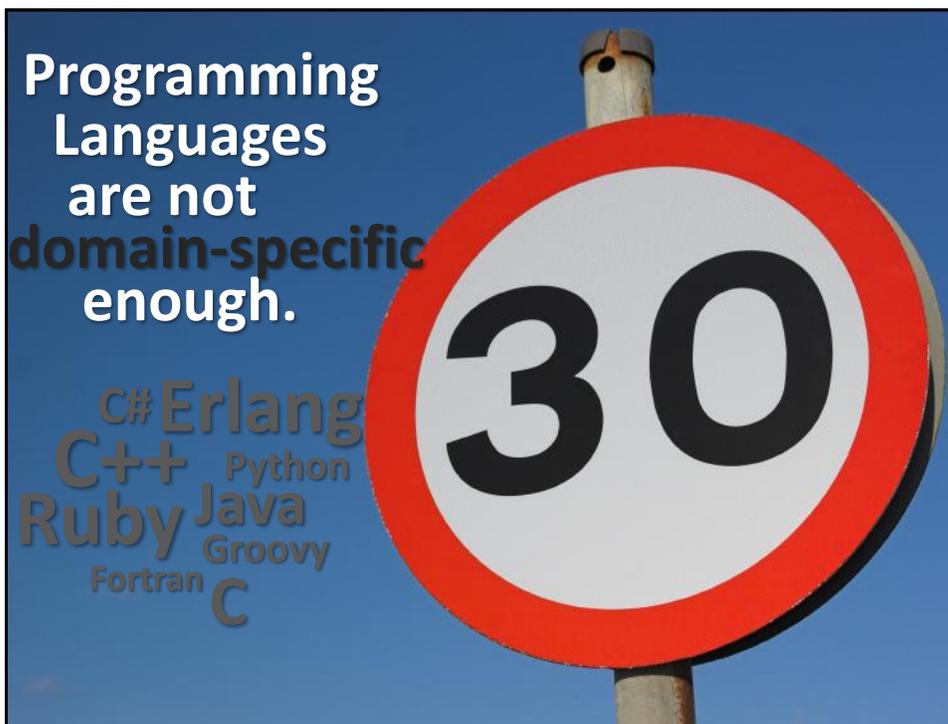
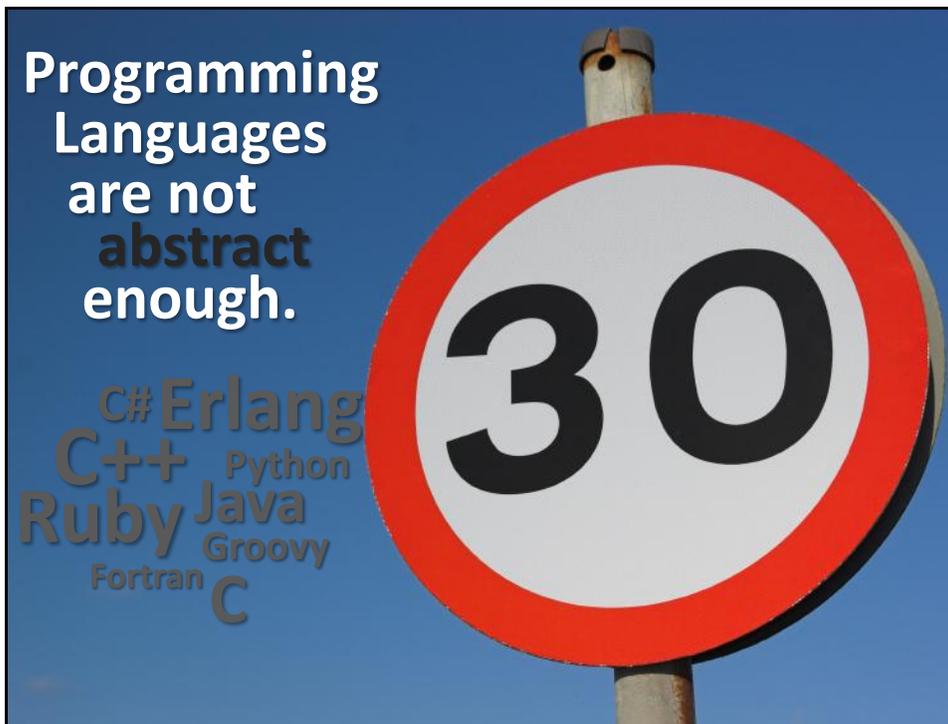
from
to **Programming**
Modeling
... and back again.

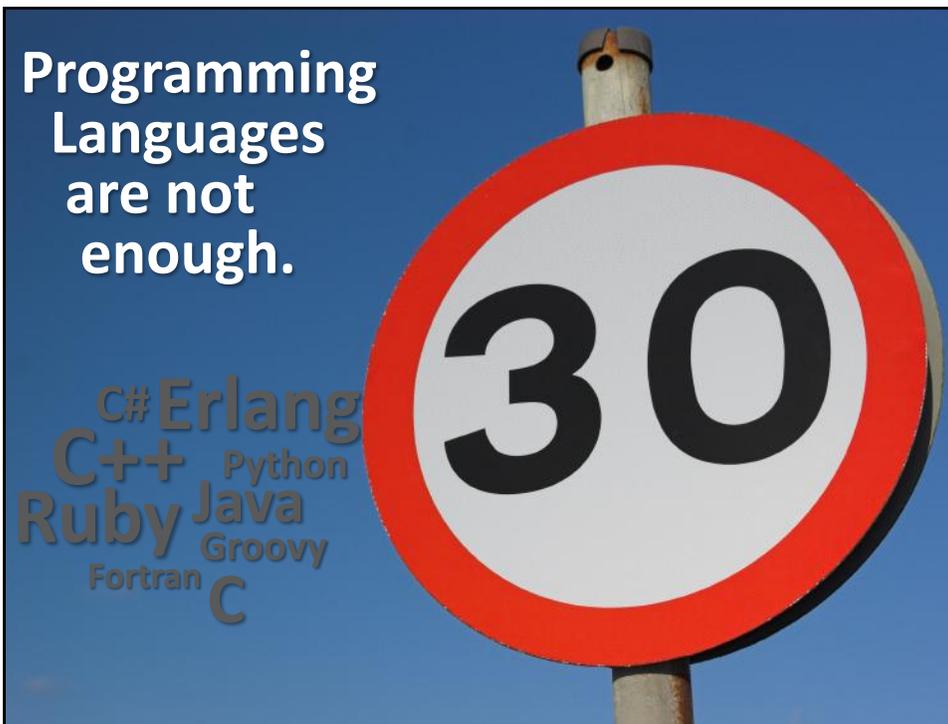
Markus Voelter
Independent/itemis
voelter@acm.org
itemis

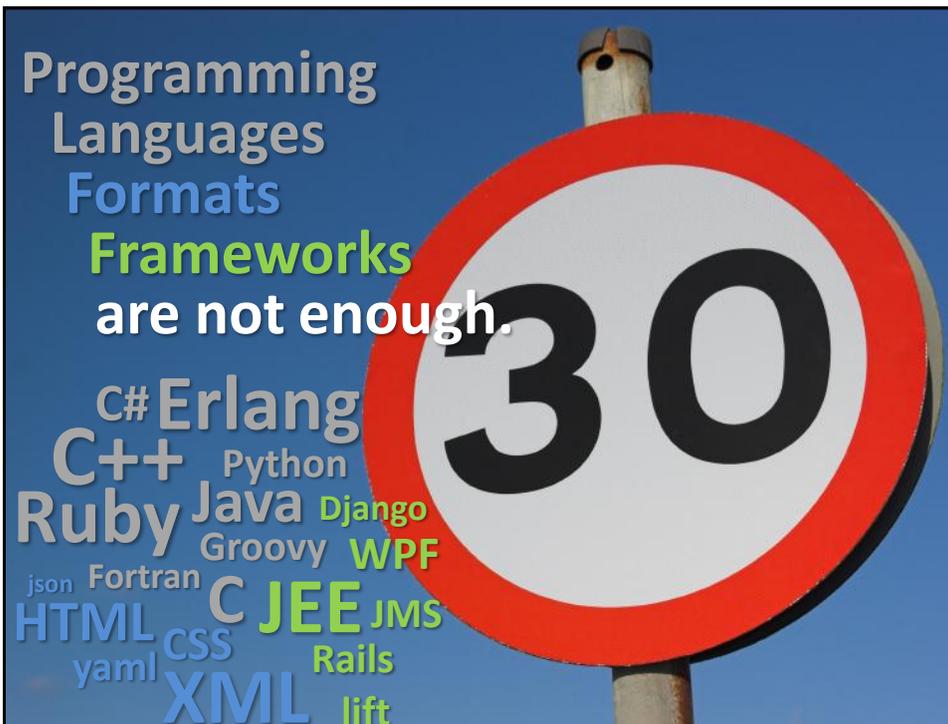
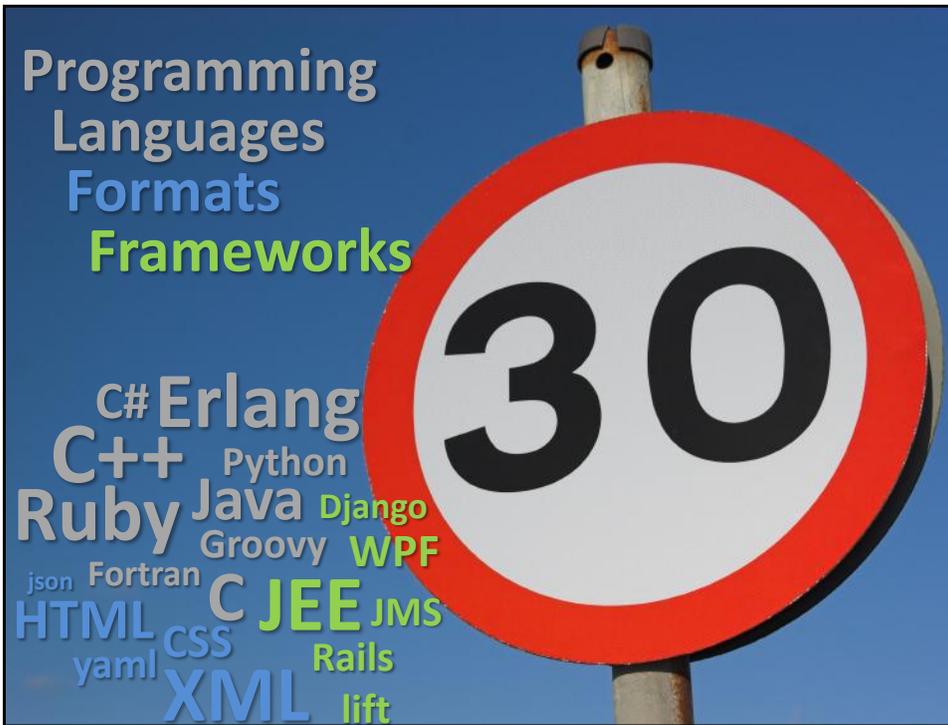
Programming Languages

#Erlang
C++ Python
Ruby Java
Fortran Groovy
C

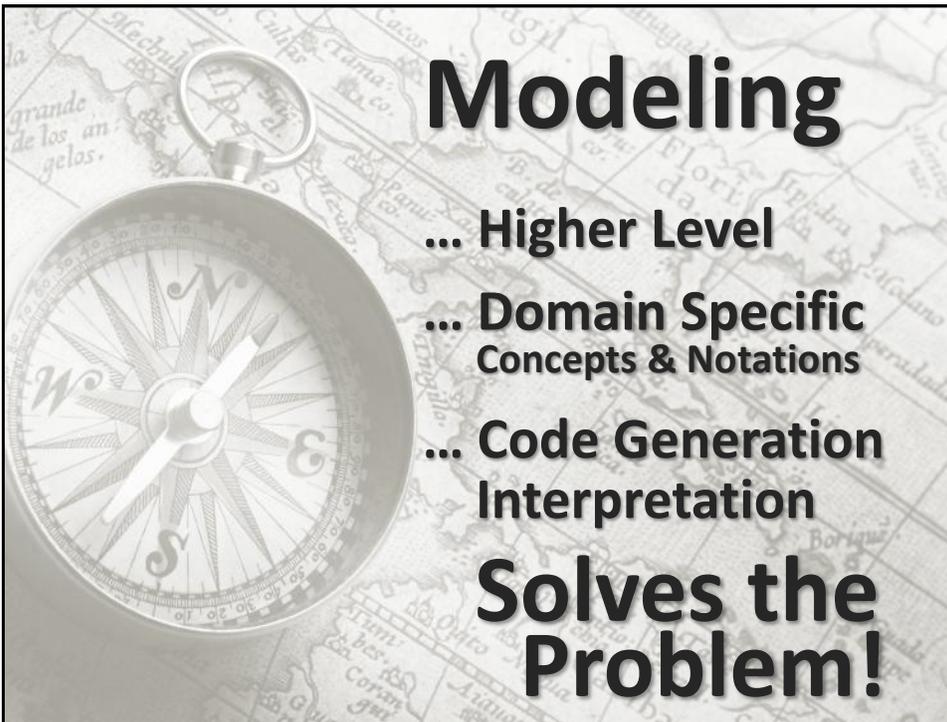








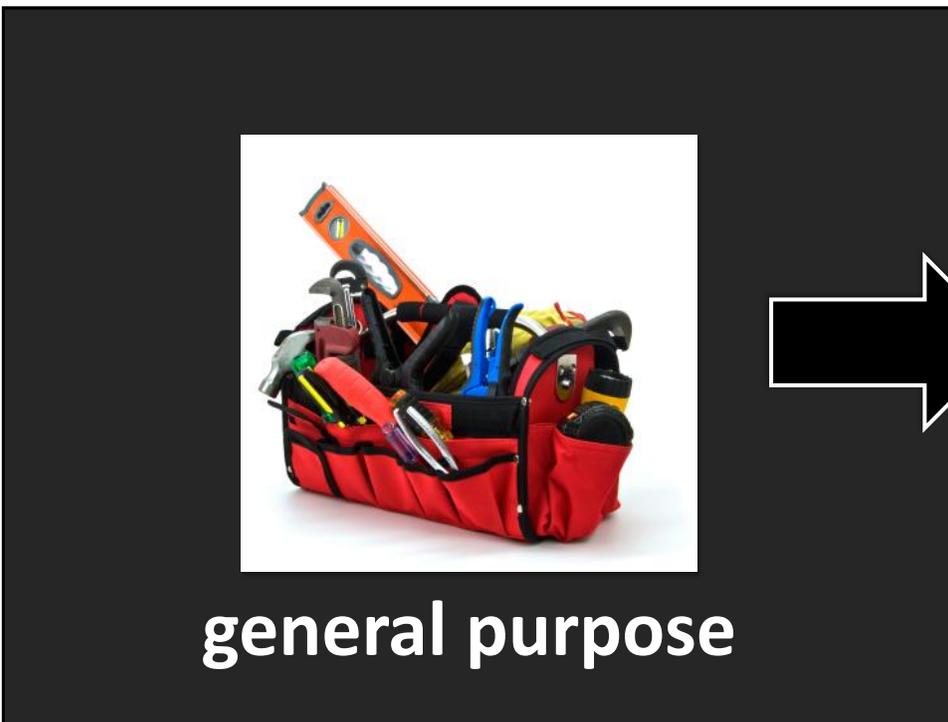
So?

A vintage-style map with a compass rose overlaid on it. The map shows various geographical features and text, including "grande de los an gelos." and "Flori da". The compass rose is a detailed illustration of a compass with a needle pointing towards the top. The text "Modeling" is prominently displayed in the upper right, followed by a list of features and the phrase "Solves the Problem!".

Modeling

- ... Higher Level
- ... Domain Specific Concepts & Notations
- ... Code Generation Interpretation

Solves the Problem!





**tailor made
effective++**

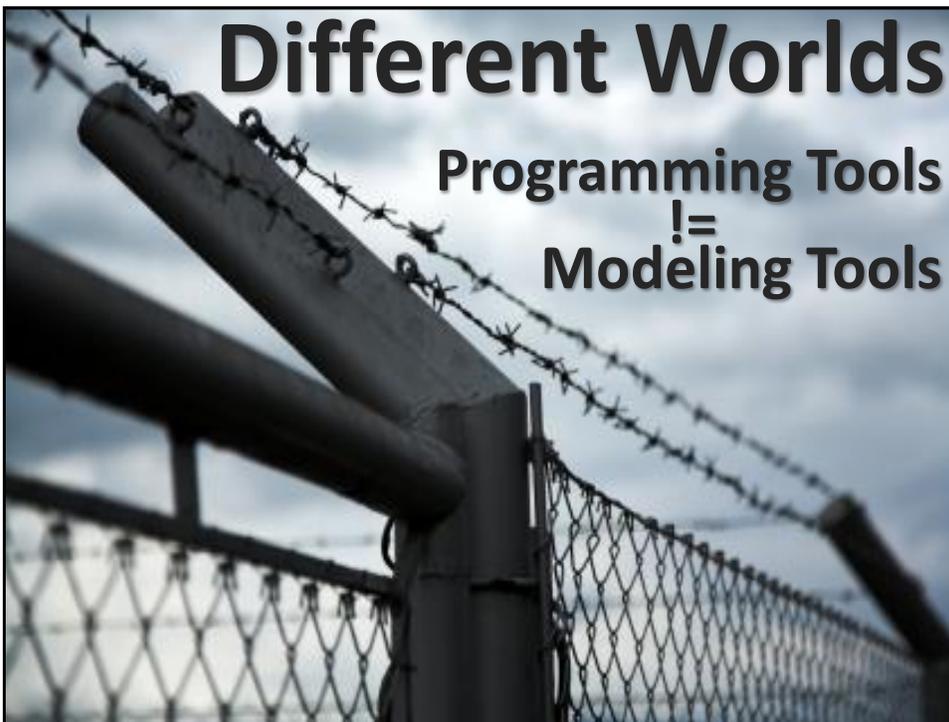
specialized, limited

used by experts

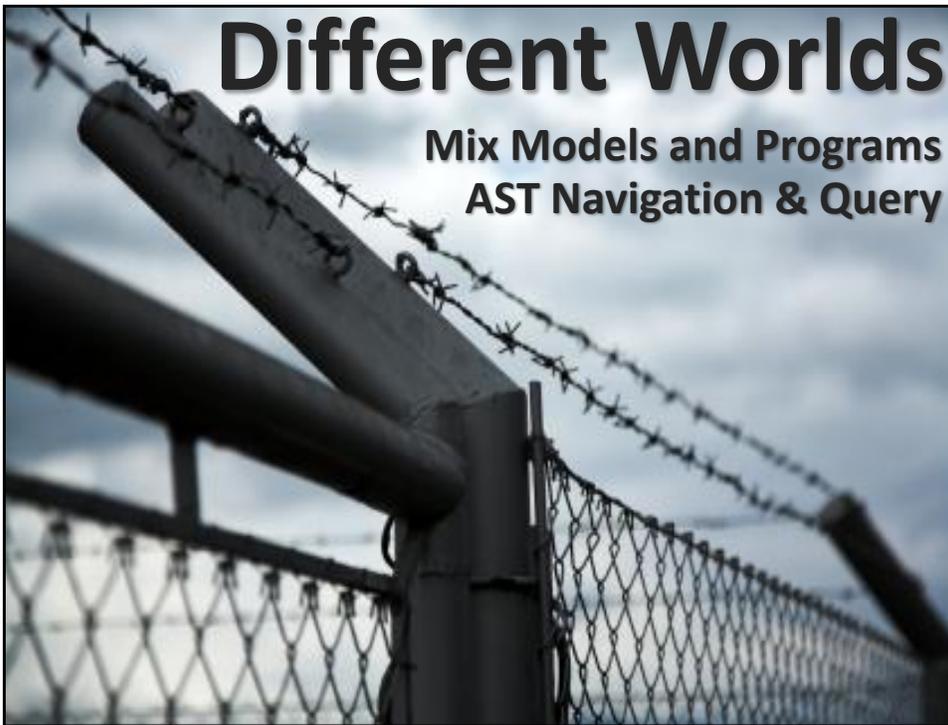
**together with other
specialized tools**

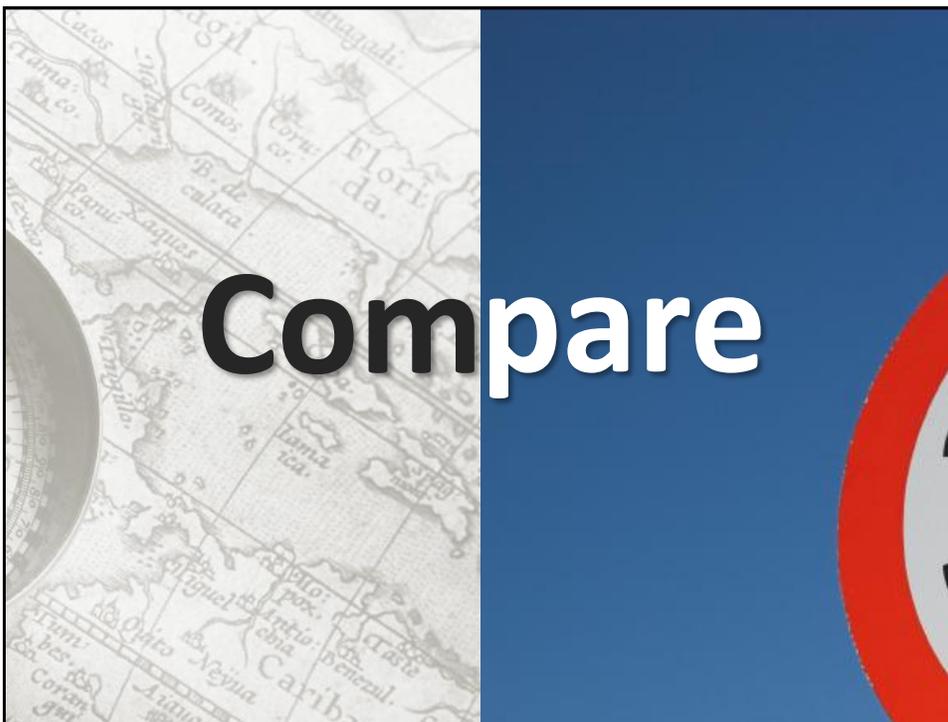


But:









Domain Specific Notations and Abstractions

Flexible!

Limited!

Frameworks
Libraries
(Fluent) APIs

Graphical Textual Forms Tables

Flexible!

Limited!

Textual Trees

**Customize
Generator
or
Interpreter**

Flexible!

Limited!

Reflection
Meta Programs
Open Compilers

**Define custom
Query
or
Navigate
or
Transform**

Flexible!

Limited!

AST APIs
Static Analysis
Regex

**Custom
or Validation
Error Checks**

Flexible!

Limited!

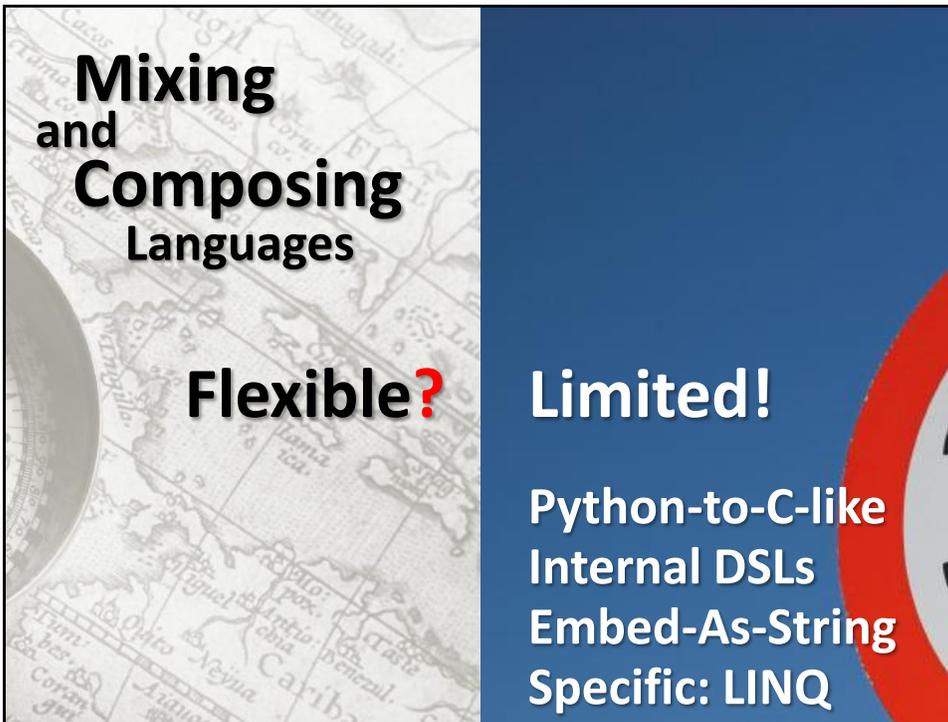
IDE plugins
Static Analysis
Open Compilers

**Different
Representations
and Projections**

Flexible!

Limited!

Text is Text
Code Folding
Tree Views
Visualizations



**Mixing
and
Composing
Languages**

Flexible?

Limited!

Python-to-C-like
Internal DSLs
Embed-As-String
Specific: LINQ



Brittle!

Modeling Tools...!?

**Scalable
Usable
IDE Support**

Mature!

Brittle!

**Debugging
Refactoring
Testing**

Mature!

Brittle!

**Versioning
Diff, Merge
Branching**

Mature!

for some tools...





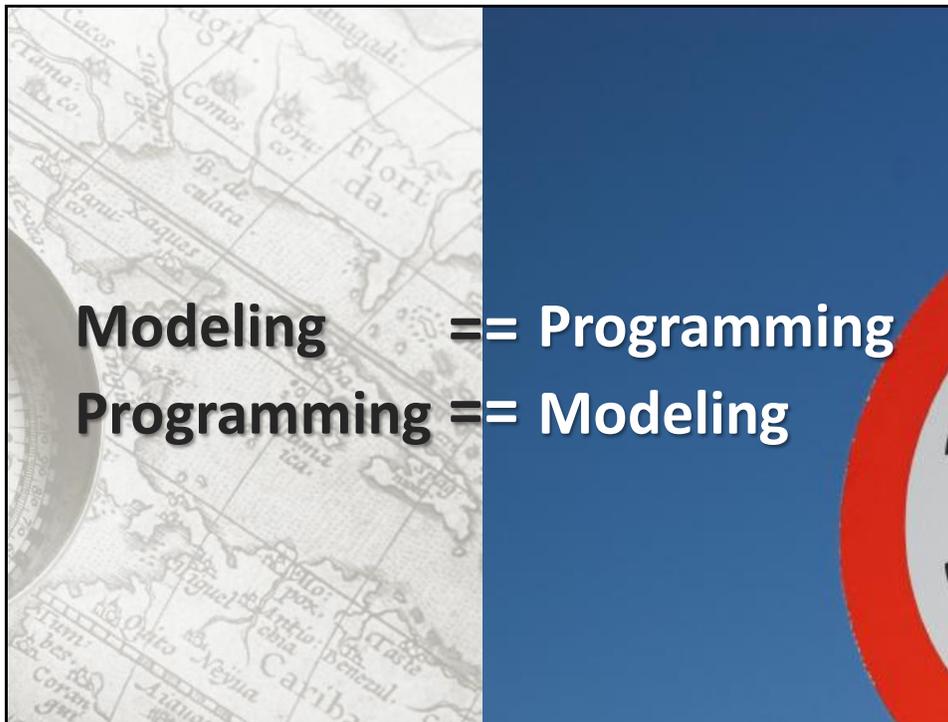
**Why
the difference?**

It is time for ...



... a Different Perspective





**modeling ==
programming**

modeling == programming

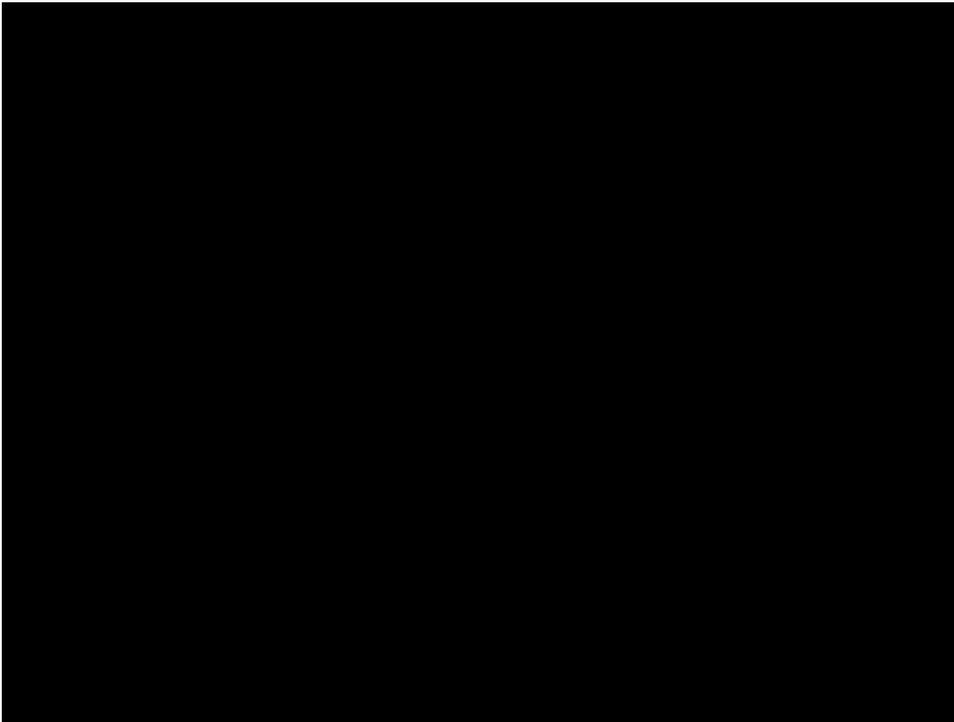
- ... at different levels of **abstraction**
- ... from different **viewpoints**
- ... **integrated!**

modeling == programming

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

**modeling ==
programming**

And always:
precise and tool processable





Language Workbench

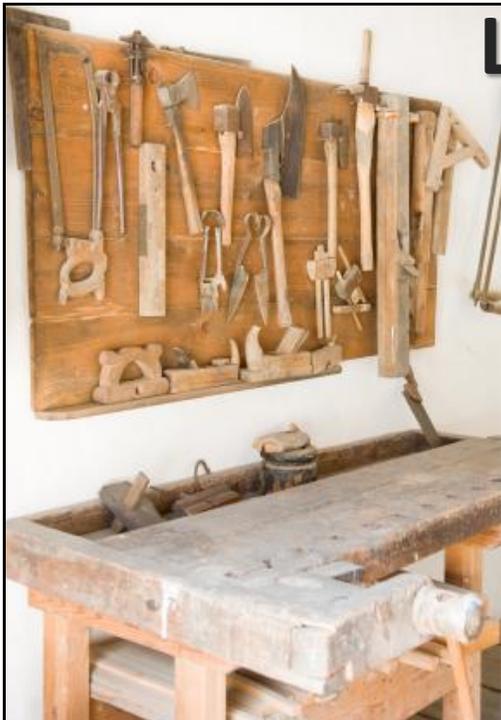
(Martin Fowler)



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)

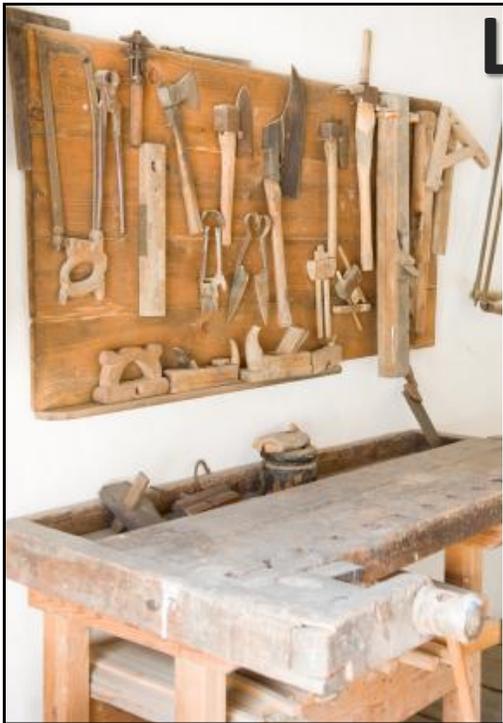
projectional
editing



Language Workbench

(Martin Fowler)

persist
incomplete
or
contradictory
information



Language Workbench

(Martin Fowler)

powerful
editing +
testing
refactoring
debugging
groupware

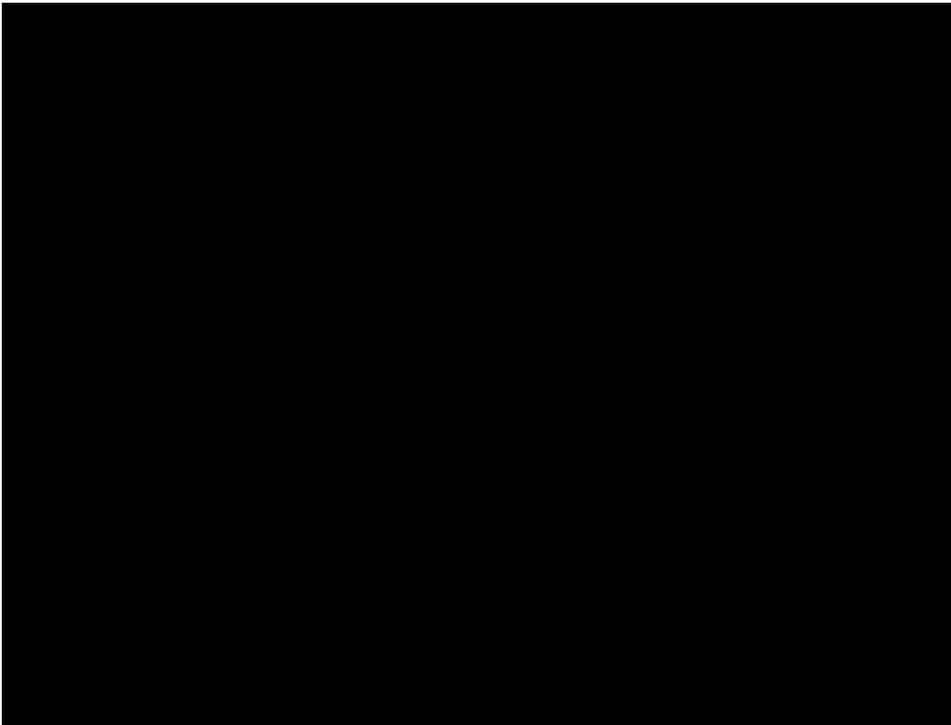
language definition
implies
IDE definition



Language Workbench

(Martin Fowler)

support for +
„classical“
programming
„classical“ and
modeling

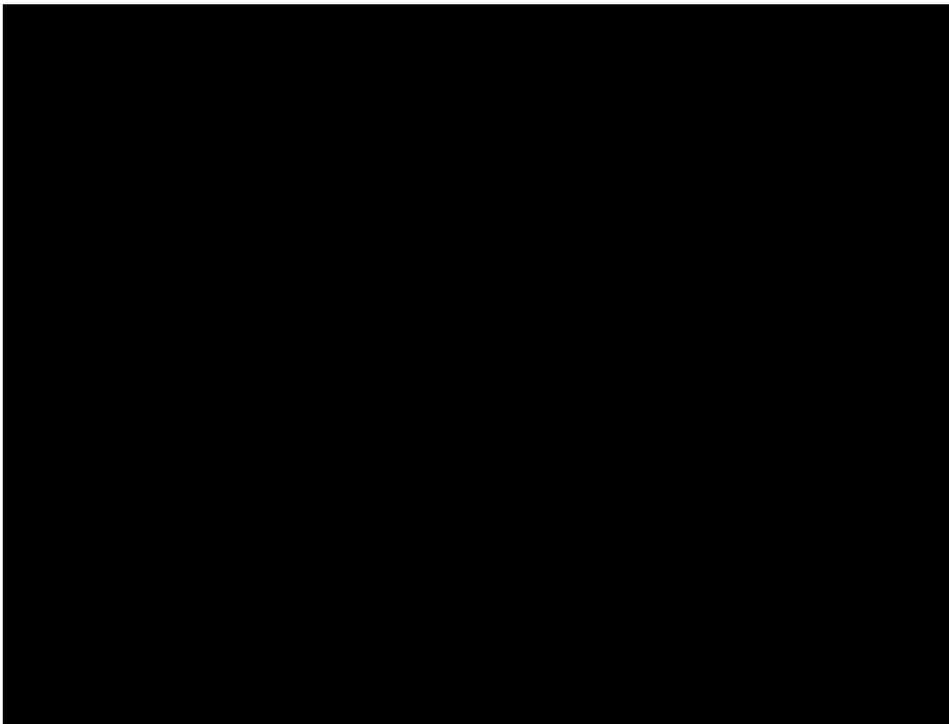
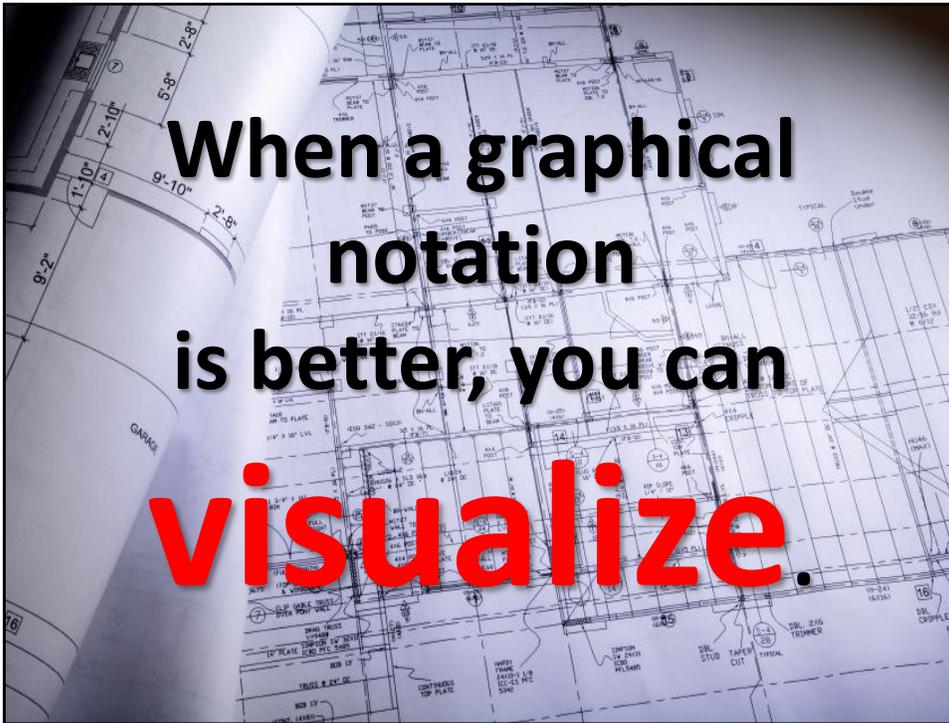


Syntax
primarily
textual

The background of the slide features several mathematical formulas in a light blue, semi-transparent font. The most prominent formula is the Fisher information matrix:
$$\frac{\partial^2}{\partial \theta^2} \ln f_{\theta}(x) = -\frac{\partial}{\partial \theta} \int_{\mathcal{R}_n} T(x) f(x, \theta) dx = \int_{\mathcal{R}_n} \frac{\partial}{\partial \theta} T(x) f(x, \theta) dx$$
 Other visible formulas include
$$\frac{\partial}{\partial a} \ln f_{a, \sigma^2}(\xi) = \frac{(\xi_1 - a)}{\sigma^2}$$
 and
$$f_{a, \sigma^2}(\xi) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(\xi_1 - a)^2}{2\sigma^2}\right\}$$

Syntax
primarily
textual
 with more symbols
 think: mathematics

Syntax
primarily
textual
 sometimes
 box&line style

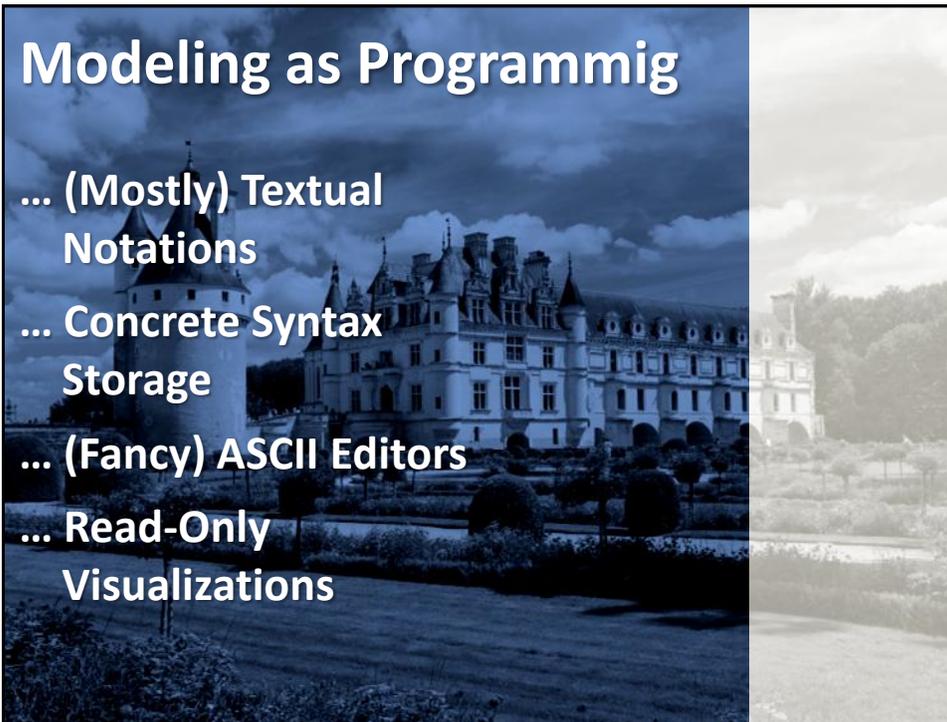


1 Available Tooling

Eclipse
Xtext

Modeling as Programmig

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



Custom Syntax

Graphical
Textual
Symbolic++

IDE Support

Teamwork
Debugging
Custom Editors

Complete Symbolic Integration

**Goto Def
Find Refs
Refactoring**

Infrastructure Integration

- ... storage not text**
- ... diff/merge with existing tools**
- ... existing text work well!**

Language Composition

**Grammar composition with
traditional parsers is tough!**

**More advanced parsers
currently research**

**Limited to
Unicode**

**how to handle
non-character symbols**

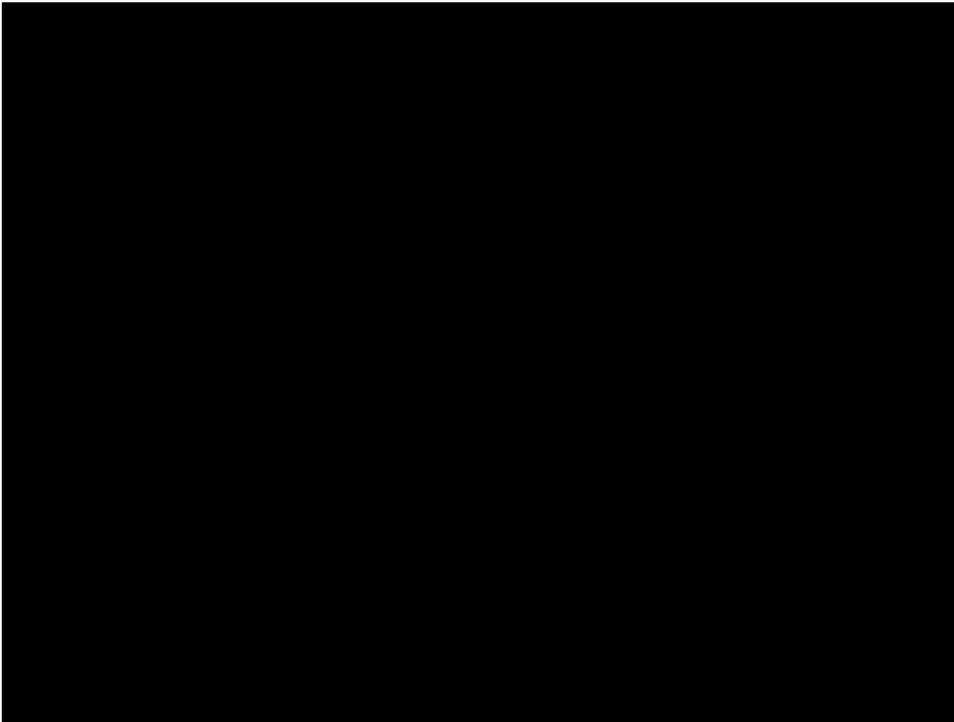
Graphics != Text

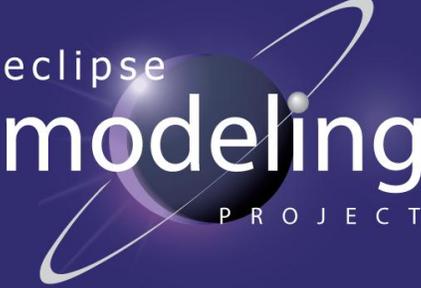
two worlds...

separate editors

... per syntax/viewpoint

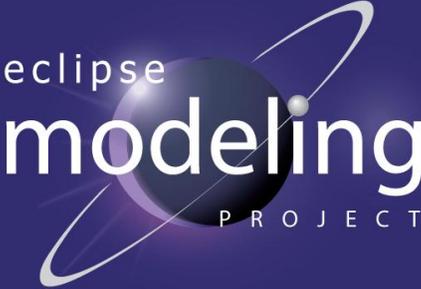
... models can still be ref integrated





eclipse
modeling
PROJECT

<http://eclipse.org/modeling>



xtext

eclipse
modeling
PROJECT

<http://eclipse.org/xtext>

Xtext: Specify Grammar

eclipse
modeling
PROJECT

```

Namespace:
  "namespace" name=ID (featureclause=FeatureClause)? "("
  (usings+=Using)*
  ( subNamespaces+=Namespace |
    components+=Component |
    datatypes+=DataType |
    interfaces+=Interface |
    compositions+=Composition )*
  ")";

Using:
  "using" namespace=[Namespace|qualID];

Component:
  (pointcut=Pointcut)? "component" name=ID (tags=TagsClause)? (featurec
  (ports+=Port)*
  ")";

Port:
  MessagePort | DataPort;

MessagePort:
  ProvidedPort | RequiredPort;

ProvidedPort:
  "provides" name=ID ":" interface=[Interface] (featureclause=FeatureCl

RequiredPort:
  
```

Xtext: Generated Editor

eclipse
modeling
PROJECT

```

namespace com {
  namespace airwizard {
    using com.airwizard.domaintypes
    using com.airwizard.types

    namespace shared {
      struct aaa {
        x : FlightAIDI
        y : Flights
      }

      typedef String FlightID

      struct FlightStatus {
        f : FlightStatus
        eta: FlightID -- com.airwizard.shared
        Flights -- com.airwizard.shared
        String -- com.airwizard.types
        Time -- com.airwizard.domaintypes
        neww aaa -- com.airwizard.shared
        boolean -- com.airwizard.types
        int -- com.airwizard.types
      }
    }
  }
}

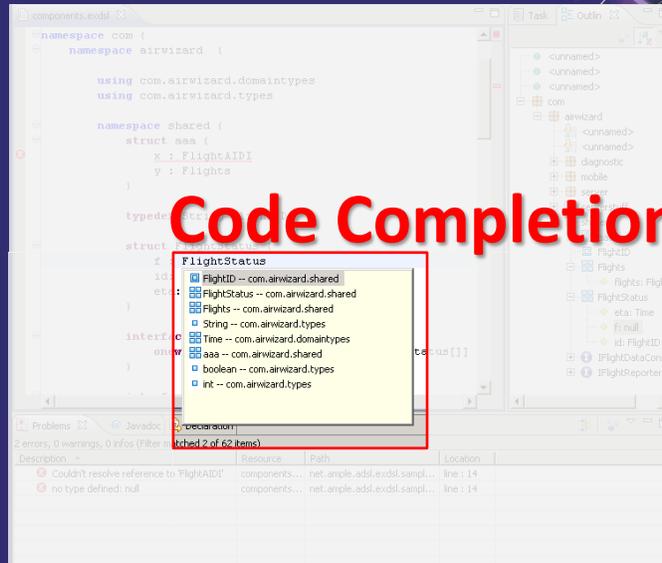
```

Problems 2 errors, 0 warnings, 0 infos (Filter matched 2 of 62 items)

Description	Resource	Path	Location
Couldn't resolve reference to 'FlightAIDI'	components...	net.anple.adsl.exdsl.sampl...	line : 14
no type defined: null	components...	net.anple.adsl.exdsl.sampl...	line : 14

Xtext: Generated Editor

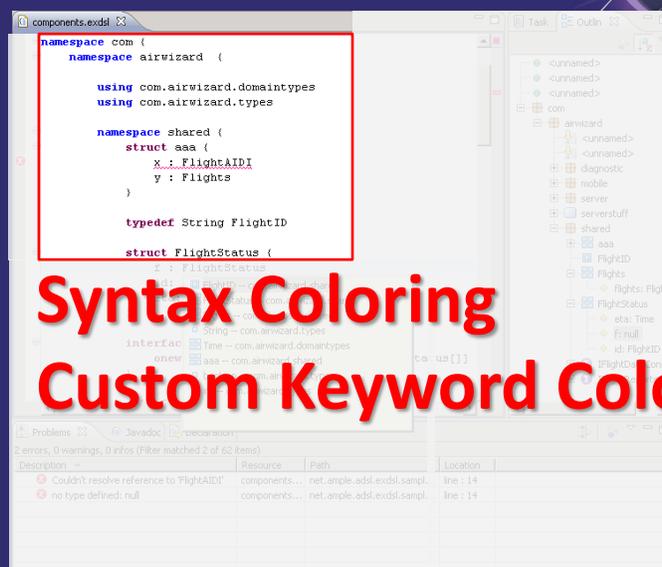
eclipse
modeling
PROJECT



Code Completion

Xtext: Generated Editor

eclipse
modeling
PROJECT



Syntax Coloring
Custom Keyword Coloring

Xtext: Generated Editor



Realtime Constraint Validation

The screenshot shows the Eclipse IDE with an Xtext editor. A red box highlights a line of code: `x: FlightAID1`. Below the editor, the Problems view shows two error messages:

Description	Resource	Path	Location
Couldn't resolve reference to 'FlightAID1'	components...	net.anple.adsl.exdsl.sampl...	line : 14
no type defined: null	components...	net.anple.adsl.exdsl.sampl...	line : 14

Xtext: Generated Editor



Customizable Outlines

The screenshot shows the Eclipse IDE with an Xtext editor. A red box highlights a customized outline in the Outline view. The outline shows a tree structure of the model elements, including namespaces, structures, interfaces, and types. The error messages from the previous screenshot are also visible in the Problems view at the bottom.

Xtext: Generated Editor

eclipse
modeling
PROJECT

Code Folding

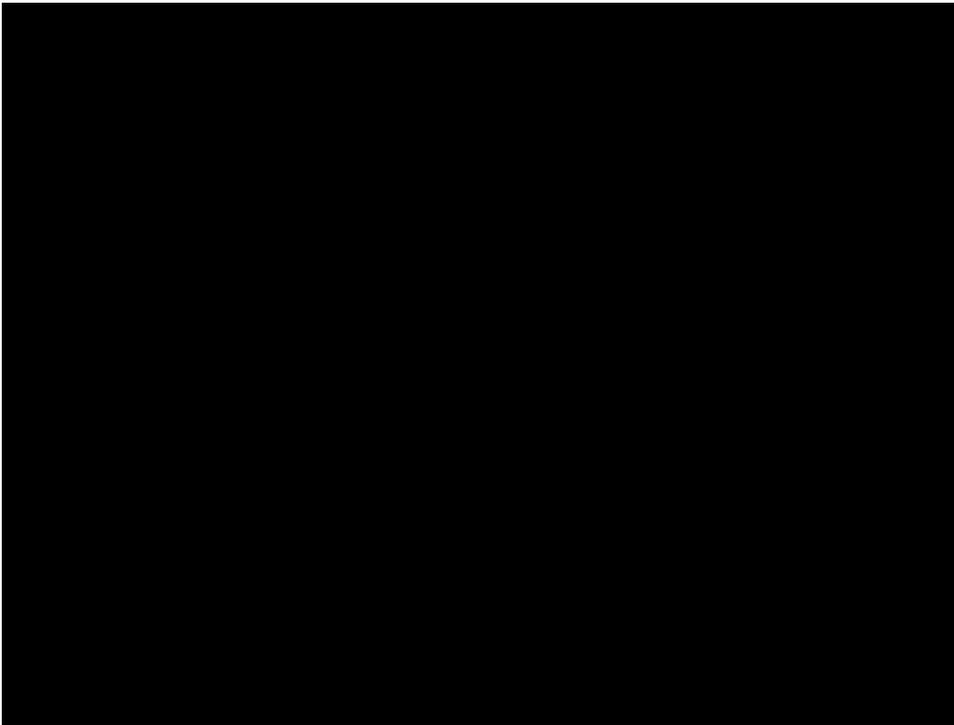
Description	Resource	Path	Location
Couldn't resolve reference to 'FlightAID'	components...	net.eclipse.adsl.exdsl.samp...	line : 14
no type defined: null	components...	net.eclipse.adsl.exdsl.samp...	line : 14

Xtext: Generated Editor

eclipse
modeling
PROJECT

Goto Definition
Find References
Cross-File References
Model as EMF

Description	Resource	Path	Location
Couldn't resolve reference to 'FlightAID'	components...	net.eclipse.adsl.exdsl.samp...	line : 14
no type defined: null	components...	net.eclipse.adsl.exdsl.samp...	line : 14



2 Available Tooling

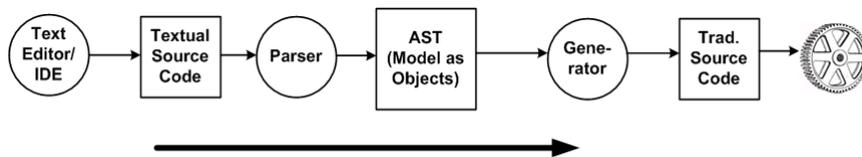
Jetbrains'
Meta
Programming
System

Parser-based

text

... to tree

... to text



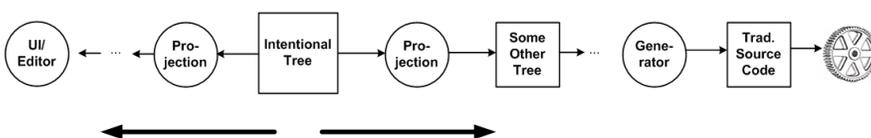
Projectional

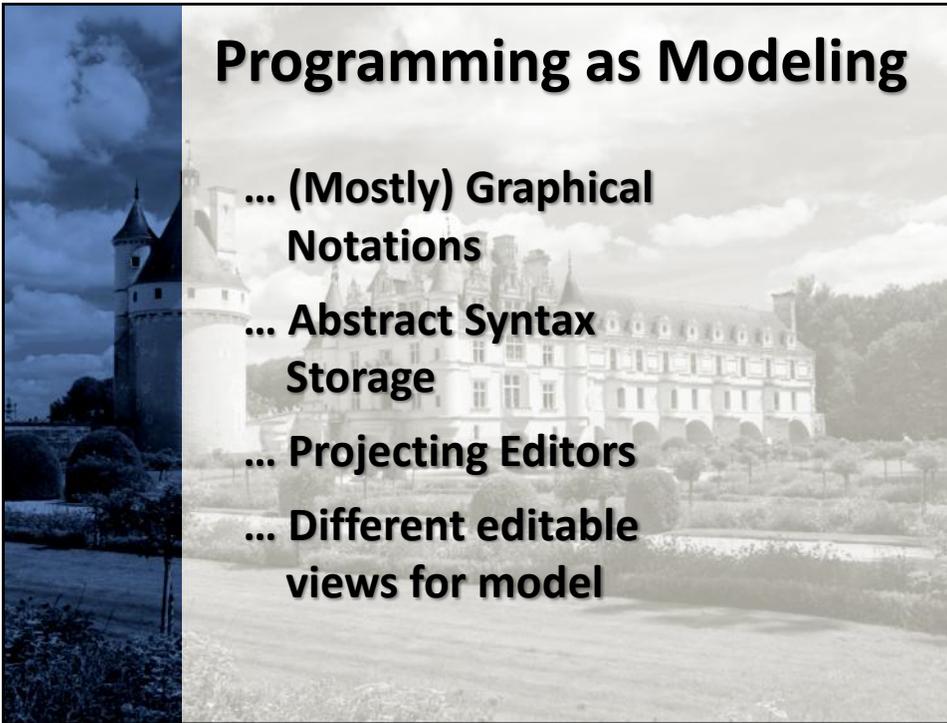
tree

... to text-lookalike (editor)

... to other trees ... [*]

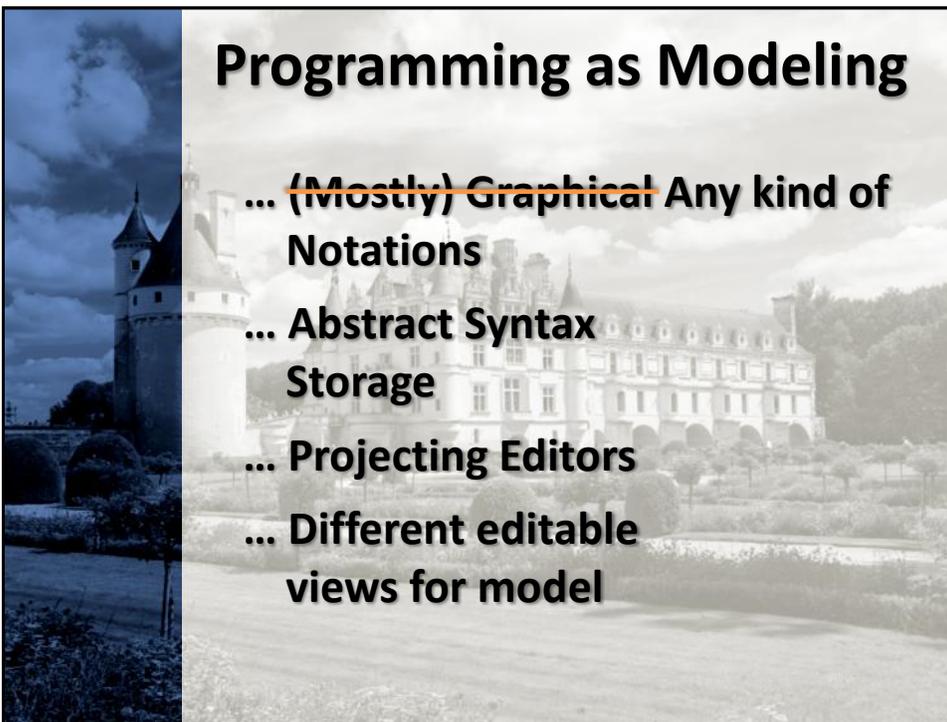
... to text





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

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

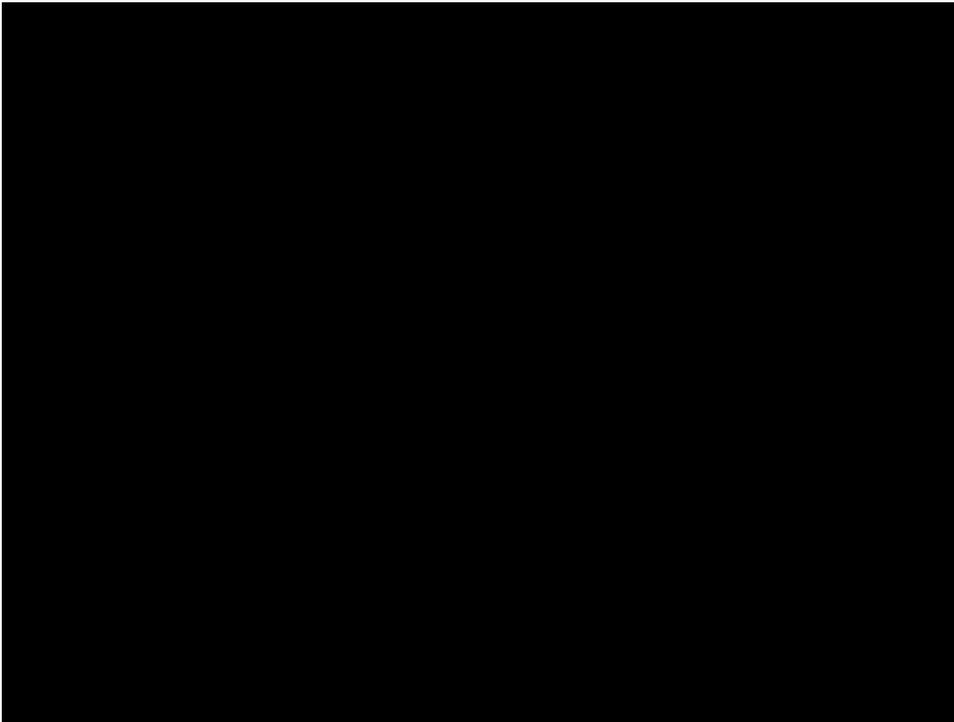
... diff/merge must be in tool

... existing text tools don't work

Proprietary Tools

... no standards

... no interop





licensed under
Apache 2.0

released in
Q2 2009

currently
1.1 RC1

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

(MPS comes with ^{Java++} **BaseLanguage**)

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

The screenshot shows the 'LockStatement' editor window with the 'Typesystem' tab selected. The content is as follows:

```

concept LockStatement extends Statement
  implements <none>

instance can be root: false

properties:
<< ... >>

children:
StatementList body 1 specializes: <none>
Expression lockExpression 1 specializes: <none>

references:
<< ... >>

concept properties:
alias = lock

concept links:
<< ... >>

concept property declarations:
<< ... >>

concept link declarations:
<< ... >>

```

The bottom toolbar includes buttons for Structure, Editor, Constraints, Behavior, Typesystem, Intentions, Find Usages, Data Flow, and Generator.

Structure ♦ Editor ♦ Typesystem ♦ Generator

The screenshot shows the 'LockStatement' editor window with the 'Editor' tab selected. The content is as follows:

```

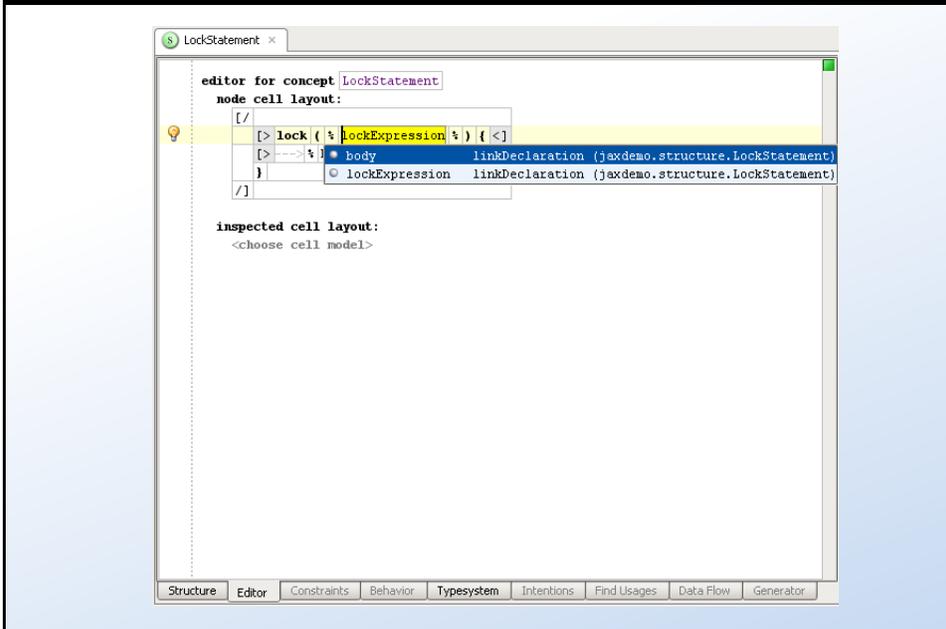
editor for concept LockStatement
node cell layout:
[ /
[> lock ( | % lockExpression % ) { <[
[> ---> % body % <[
]
] /]

inspected cell layout:
<choose cell model>

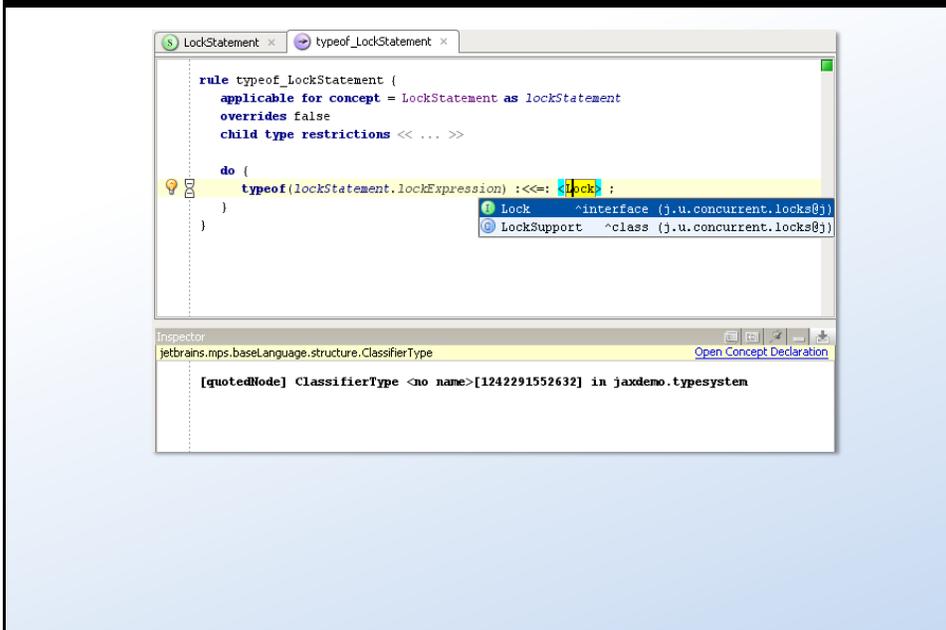
```

The bottom toolbar includes buttons for Structure, Editor, Constraints, Behavior, Typesystem, Intentions, Find Usages, Data Flow, and Generator.

Structure ♦ Editor ♦ Typesystem ♦ Generator



Structure ♦ Editor ♦ Typesystem ♦ Generator



Structure ♦ Editor ♦ Typesystem ♦ Generator

```

main x
<u>conditional root rules:</u>
<< ... >>

<u>mapping rules:</u>
<< ... >>

<u>weaving rules:</u>
<< ... >>

<u>reduction rules:</u>
concept LockStatement --> reduce_LockStatement
inheritors false
condition <always>

<u>abandon roots:</u>
<< ... >>

<u>pre-processing scripts:</u>
<< ... >>

```

Structure ♦ Editor ♦ Typesystem ♦ Generator

```

reduce_LockStatement x
content node:
public class someclass extends <none> implements <none> {
  <<static fields>>

  <<static initializer>>
  <<fields>>
  <<properties>>
  <<initializer>>
  public someclass() {
    <no statements>
  }

  public void somemethod() {
    Lock l = null;
    try {
      $COPY_SRC[1].lock();
      $LOOP$[$COPY_SRC[null:]]
    } finally {
      $COPY_SRC[1].unlock();
    }
  }

  <<static methods>>

  <<static inner classifiers>>
}

```

Structure ♦ Editor ♦ Typesystem ♦ Generator

The screenshot shows an IDE window titled 'reduce_lockStatement'. The main editor displays the following code:

```

content node:
public class someclass extends <none> implements <none> {
  <<static fields>>

  <<static initializer>>
  <<fields>>
  <<properties>>
  <<initializer>>
  public someclass() {
    <<no statements>>
  }

  public void somemethod() {
    lock l = null;
    <TF> try {
      $COPY_SRC$[1].lock();
    } <TF> finally {
      $COPY_SRC$[1].unlock();
    }
  }

  <<static methods>>
  <<static inner class>>
}

```

The Inspector window shows the configuration for the 'copy/reduce node macro':

```

jetbrains.mps.lang.generator.structure.CopySrcNodeMacro
copy/reduce node macro
mapping label : <no label>
mapped node : (node, genContext, operationContext)->node< > {
  node.lockExpression;
}

```

Structure ♦ Editor ♦ Typesystem ♦ Generator

The screenshot shows an IDE window titled 'reduce_lockStatement'. The main editor displays the following code:

```

content node:
public class someclass extends <none> implements <none> {
  <<static fields>>

  <<static initializer>>
  <<fields>>
  <<properties>>
  <<initializer>>
  public someclass() {
    <<no statements>>
  }

  public void somemethod() {
    Lock l = null;
    <TF> try {
      $COPY_SRC$[1].lock();
    } <TF> finally {
      $COPY_SRC$[1].unlock();
    }
  }

  <<static methods>>
  <<static inner class>>
}

```

The Inspector window shows the configuration for the 'loop-macro':

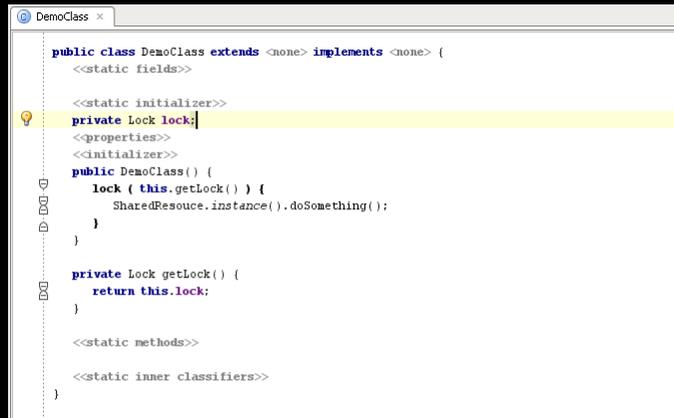
```

jetbrains.mps.lang.generator.structure.LoopMacro
loop-macro
mapping label : <no label>
mapped nodes : (node, genContext, operationContext)->sequence<node< >> {
  node.body.statement;
}

```

Language Extension Example

Result behaves like a native
base language construct



```

public class DemoClass extends <none> implements <none> {
    <<static fields>>

    <<static initializer>>
    private Lock lock;
    <<properties>>
    <<initializer>>
    public DemoClass() {
        lock ( this.getLock() ) {
            SharedResource.Instance().doSomething();
        }
    }

    private Lock getLock() {
        return this.lock;
    }

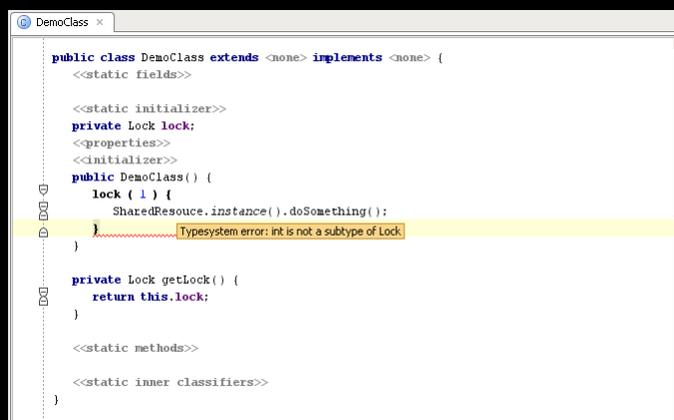
    <<static methods>>

    <<static inner classifiers>>
}

```

Language Extension Example

Result behaves like a native
base language construct



```

public class DemoClass extends <none> implements <none> {
    <<static fields>>

    <<static initializer>>
    private Lock lock;
    <<properties>>
    <<initializer>>
    public DemoClass() {
        lock ( 1 ) {
            SharedResource.Instance().doSomething();
        }
    }

    private Lock getLock() {
        return this.lock;
    }

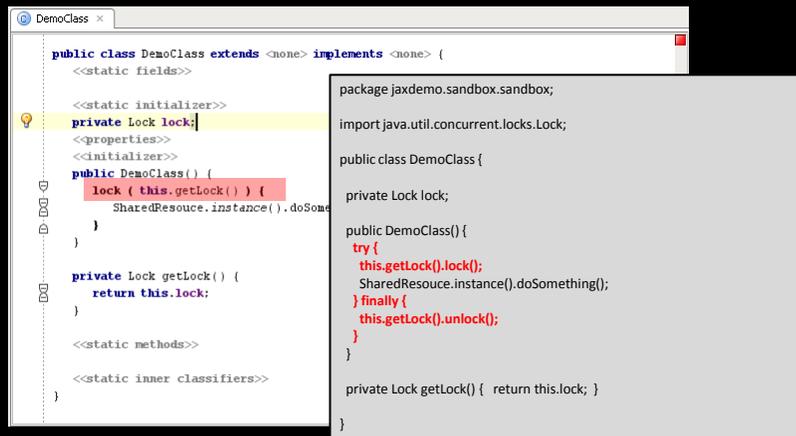
    <<static methods>>

    <<static inner classifiers>>
}

```

Language Extension Example

Translated to regular Java code based on the generator



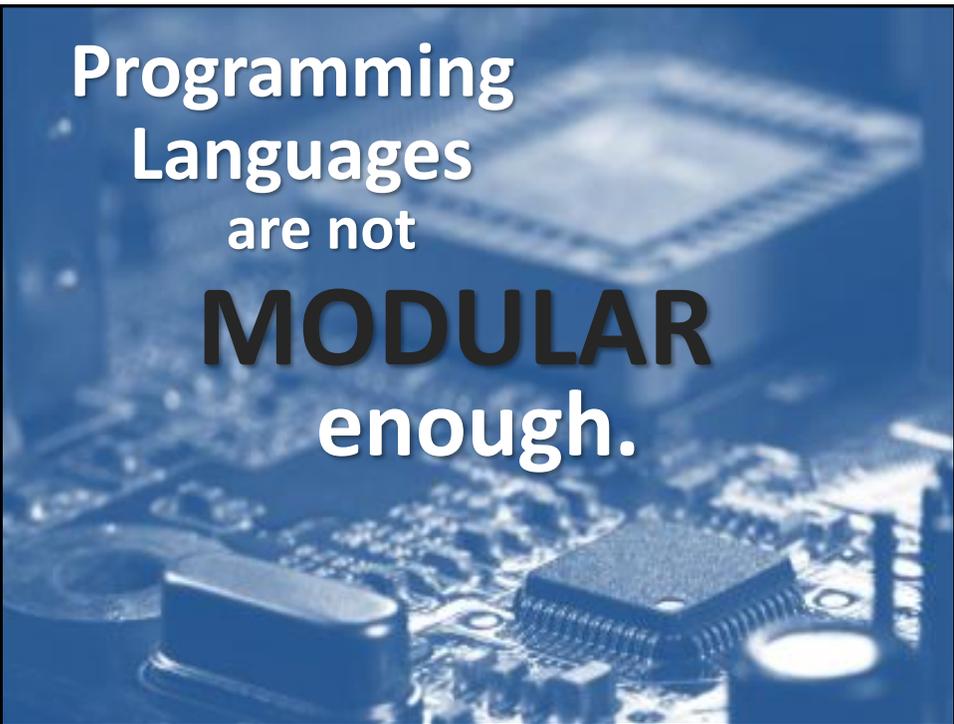
The image shows a side-by-side comparison of code. On the left, a snippet of a language extension is shown within a class structure. On the right, the equivalent regular Java code is displayed.

```
public class DemoClass extends <none> implements <none> {  
    <<static fields>>  
    <<static initializer>>  
    private Lock lock;  
    <<properties>>  
    <<initializer>>  
    public DemoClass() {  
        lock ( this.getLock() ) {  
            SharedResource.instance().doSome  
        }  
    }  
    private Lock getLock() {  
        return this.lock;  
    }  
    <<static methods>>  
    <<static inner classifiers>>  
}
```

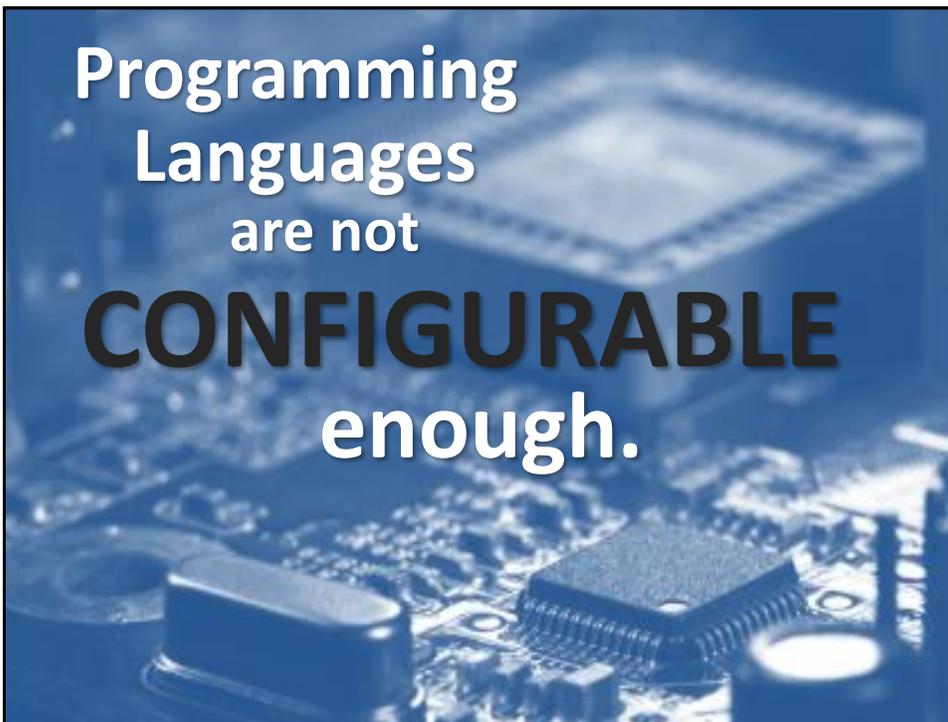
```
package jaxdemo.sandbox.sandbox;  
import java.util.concurrent.locks.Lock;  
public class DemoClass {  
    private Lock lock;  
    public DemoClass() {  
        try {  
            this.getLock().lock();  
            SharedResource.instance().doSomething();  
        } finally {  
            this.getLock().unlock();  
        }  
    }  
    private Lock getLock() { return this.lock; }  
}
```



A vision for Programming



Programming
Languages
are not
MODULAR
enough.





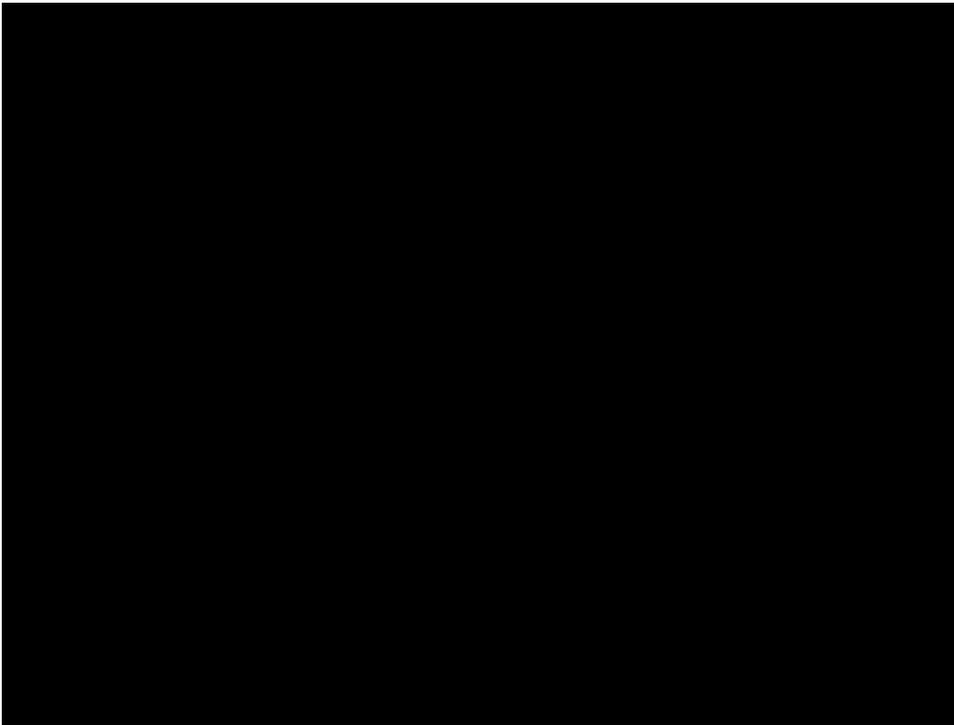
Programming
Languages
are not

ADAPTABLE
enough.



Programming
Language Syntax
is not

FLEXIBLE
enough.

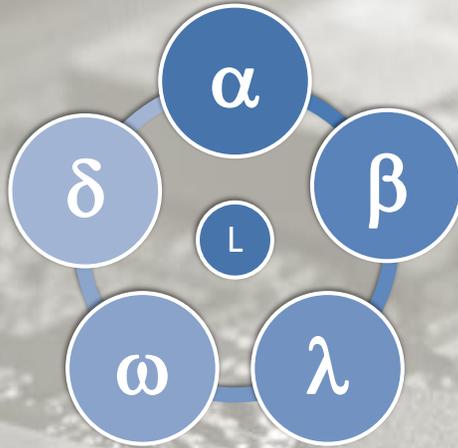


Big Language?

with **many** first class concepts!

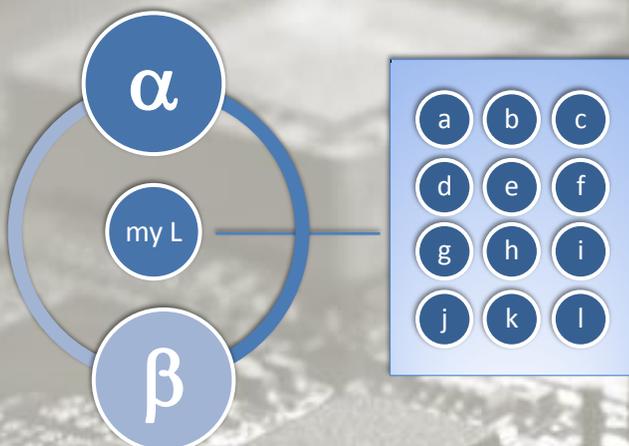
The diagram features a central blue circle containing a white capital letter 'L'. Surrounding this central circle is a ring of 14 smaller blue circles, each containing a lowercase letter from 'a' to 'n' in alphabetical order. The background of the slide is a grayscale image of a computer circuit board.

Small Language?



with a **few, orthogonal**
and **powerful** concepts

Modular Language



with many **optional,**
composable concepts

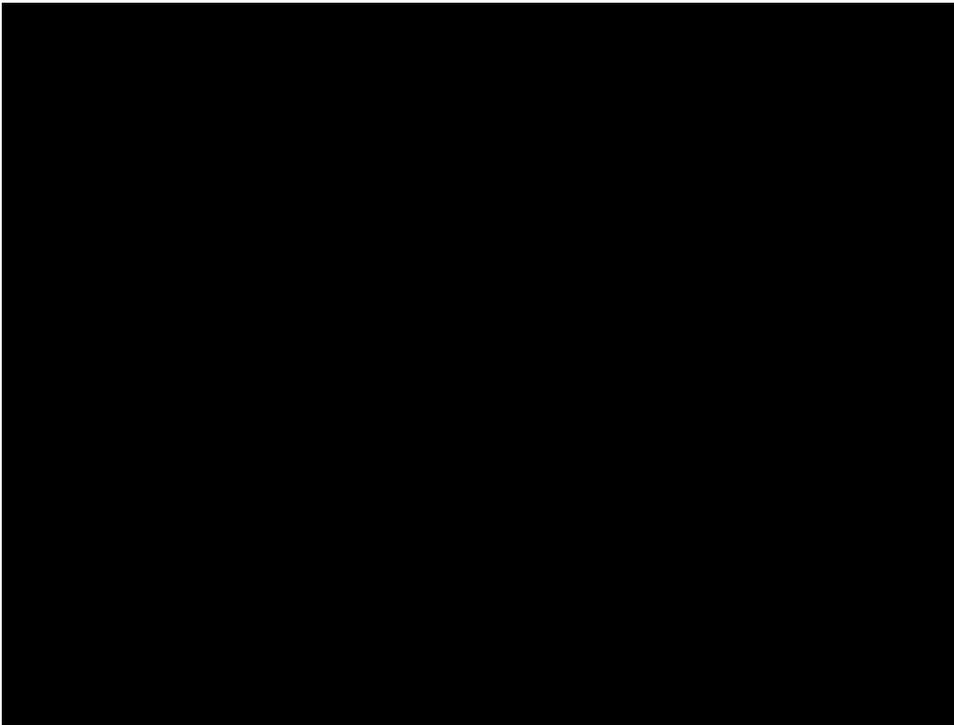
Modular Language

Like frameworks
and libraries,

Modular Language

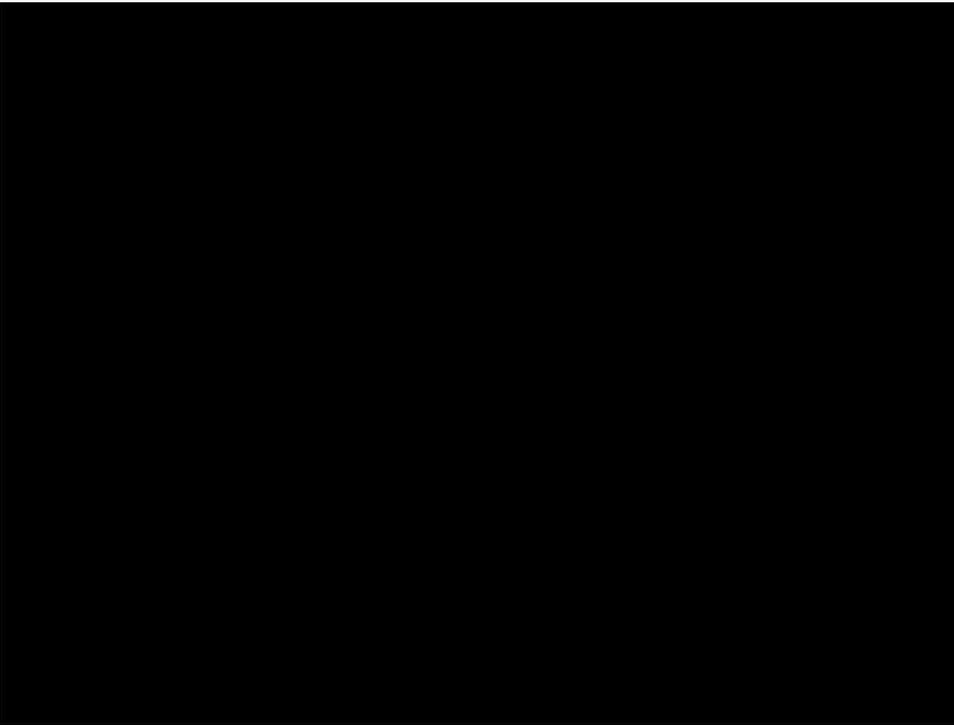
Like frameworks
and libraries,

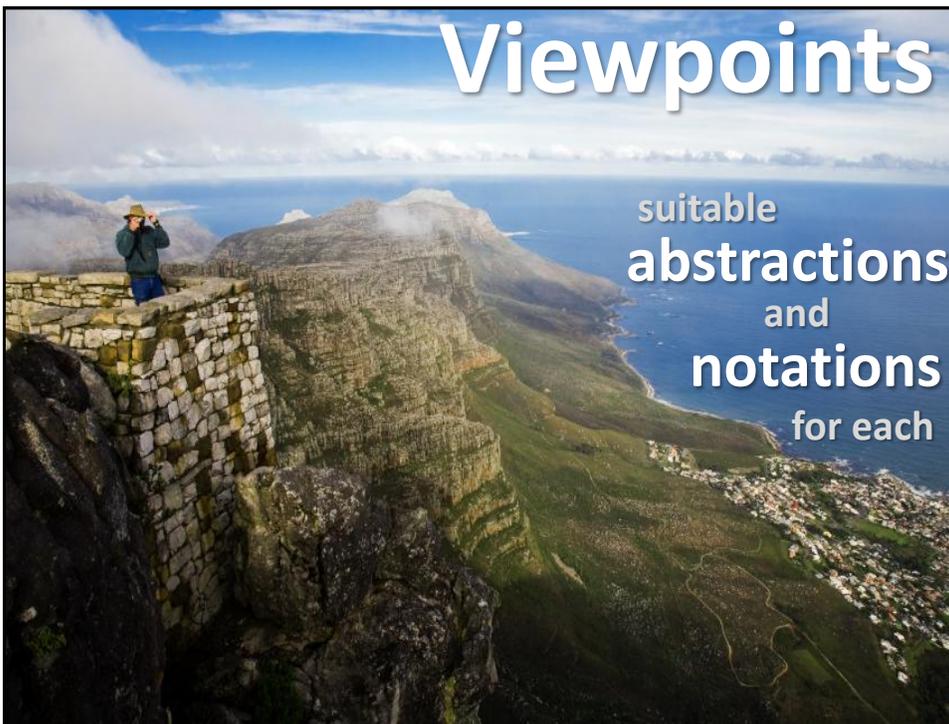
but with syntax
and IDE support

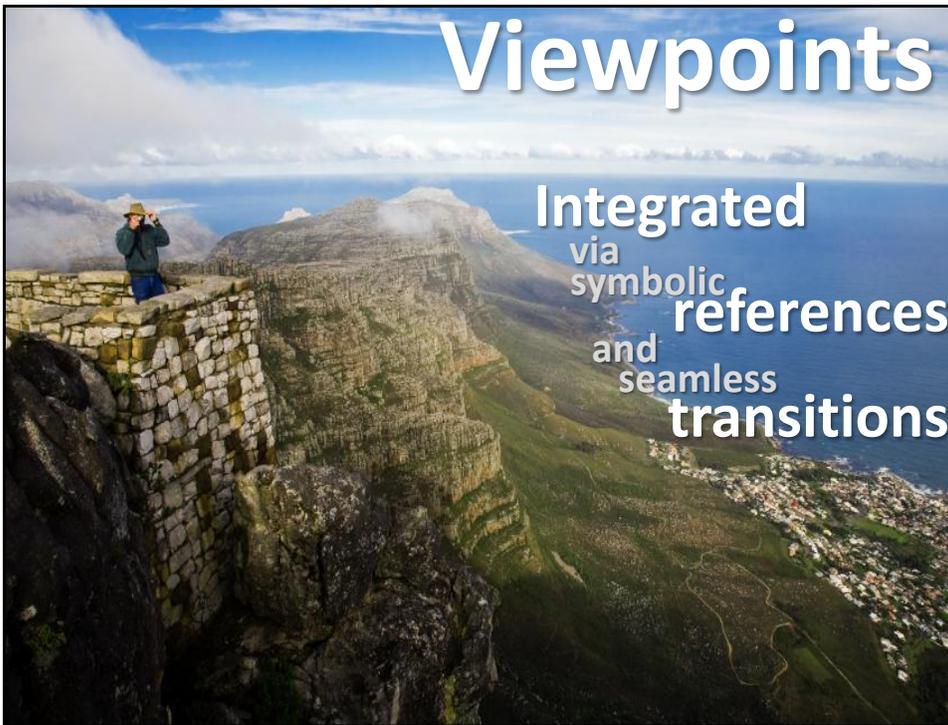


Not a new idea...

Growing A Language
(Guy L Steele)







Viewpoints Business

Custom
Notations

real
business
expert integration

A photograph of various surgical instruments, including forceps, scissors, and probes, laid out on a blue cloth. The instruments are arranged in a neat, organized manner, typical of a surgical tray.

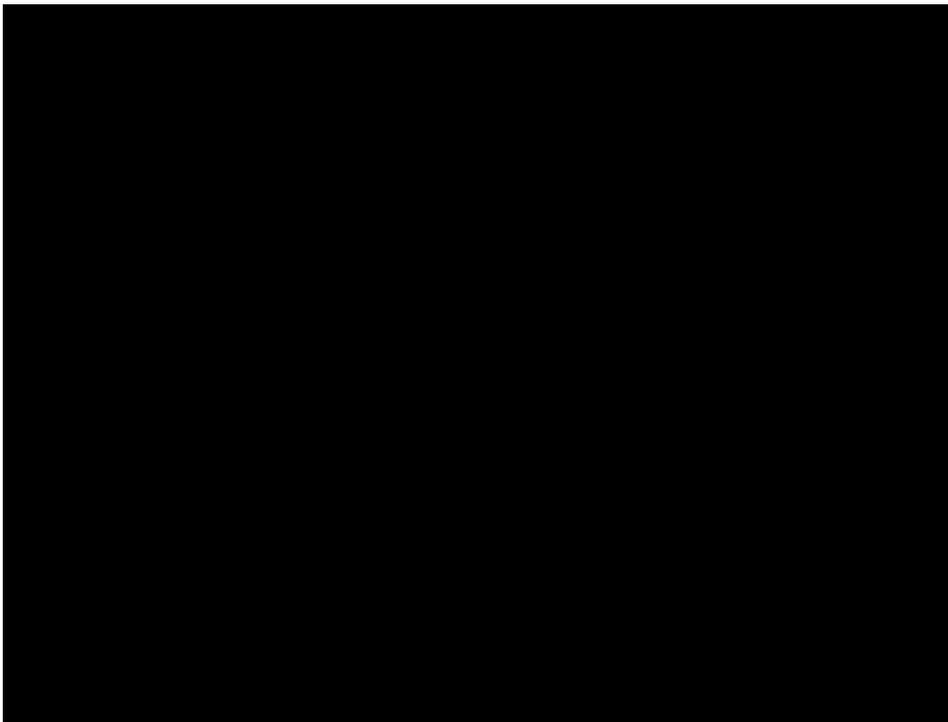
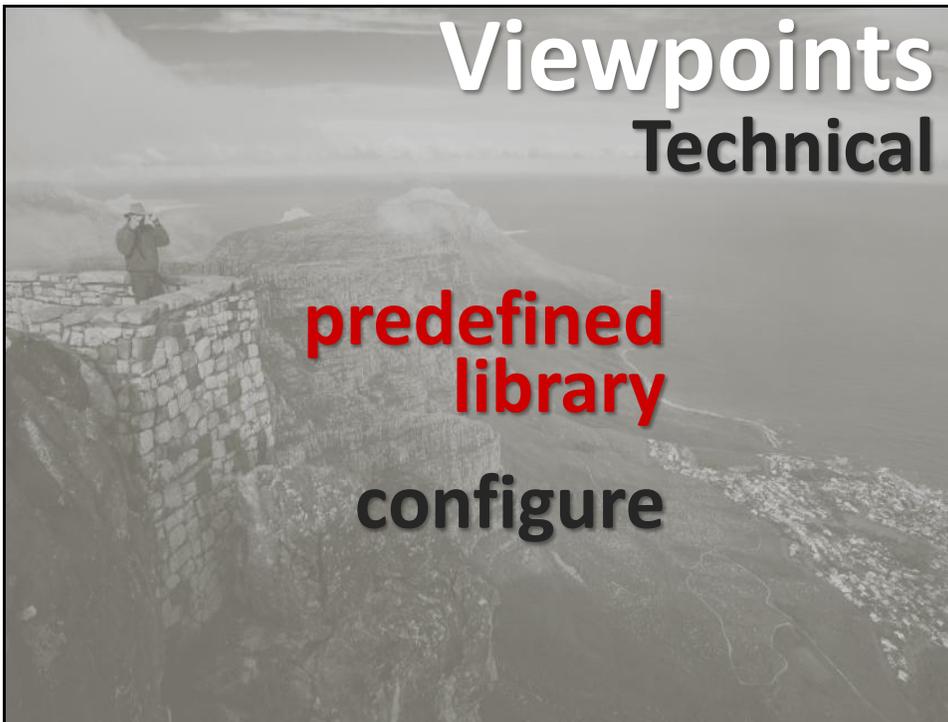
Viewpoints Business

Custom
Notations

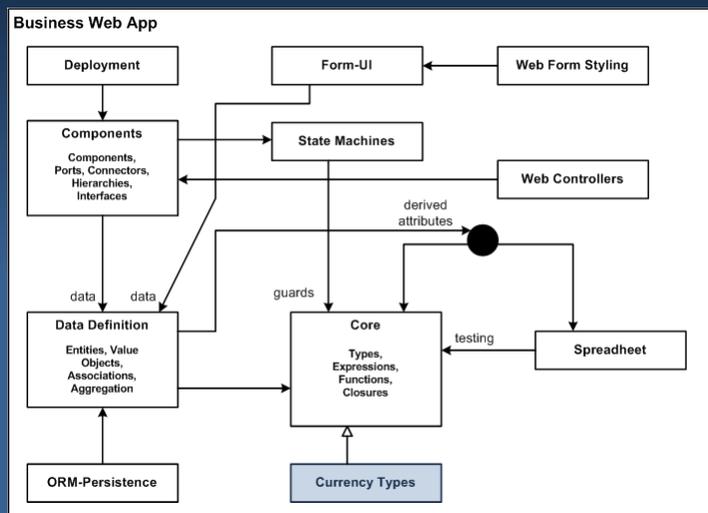
real
business
expert integration

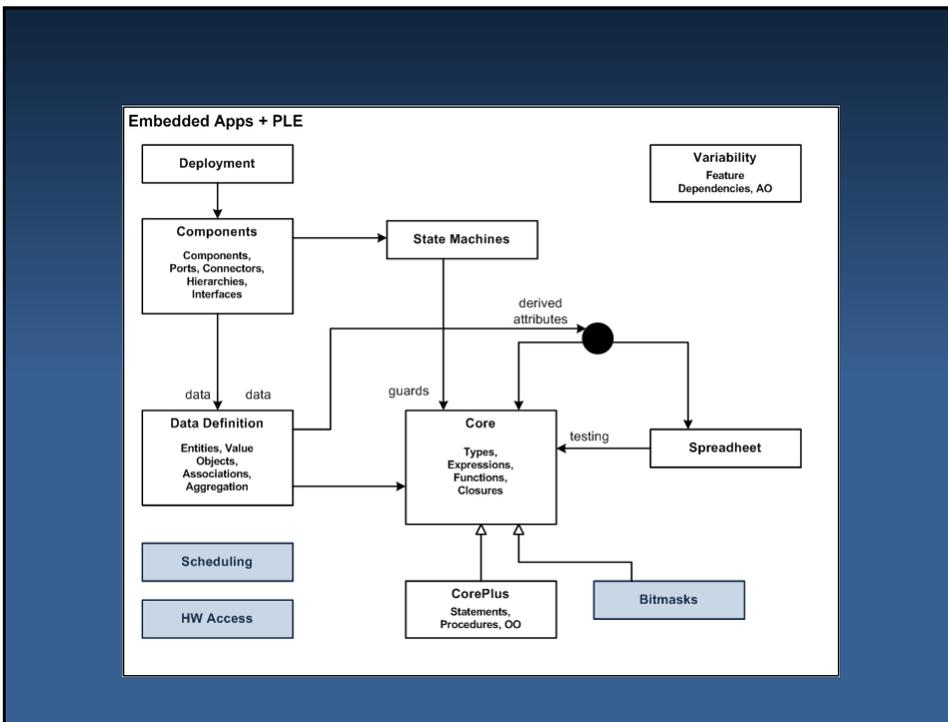
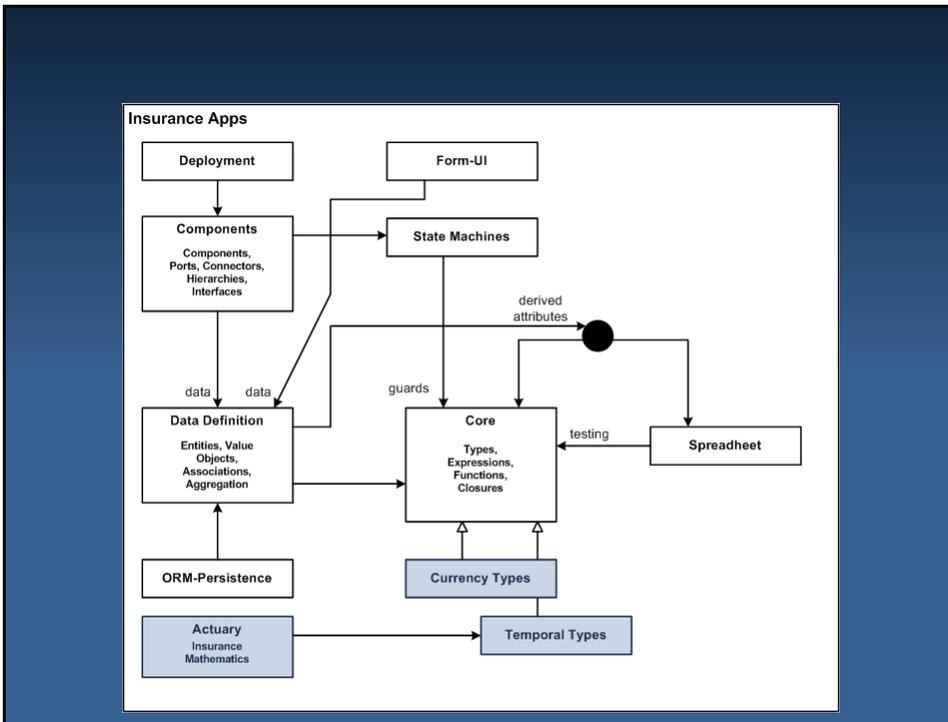
But that's another talk...

A photograph of various surgical instruments, including forceps, scissors, and probes, laid out on a blue cloth. The instruments are arranged in a neat, organized manner, typical of a surgical tray.



Example Languages







(Seemingly)
Simple Example

Adding
matrices
to C in an
embedded
environment.

Currently:

- 1 • Declare Data Structures in XML
- 2 • Generate Headers
- 3 • Implement manually in C

Currently:

Matrices
not supported
in XML format
and generator

Currently:

Tool team

would have to
do the extension

- ... a lot of work
- ... busy
- ... one central tool

Currently:

No real

compiler support
in the resulting C code

- ... type checks
- ... operator overloading
- ... generics (matrix<int>)
- ... matrix syntax?



Better Solution

```

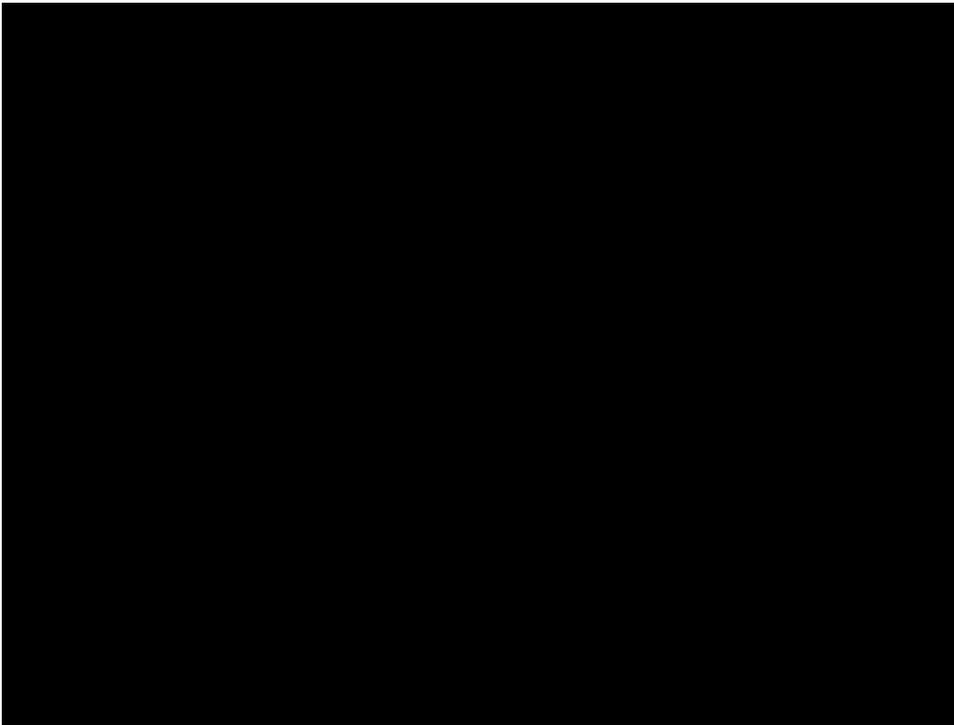
                                par1 * nengine ... 0
qmatrix<int16>[3,3] ModellMatrix = par2 * nengine * q 0 ...
                                ...           ... ...
vector<int16>[3] modellvektor = temp1
                                temp2
                                temp3

vector<int16> gausszerlege( qmatrix<int16> A, vector<int16>  $\vec{b}$  ) {
    // gausszerlegung... à la A x = b => x = A \ b
    return  $\vec{x}$ ;
}

vector<int16> erg = ModellMatrix * modellvektor;

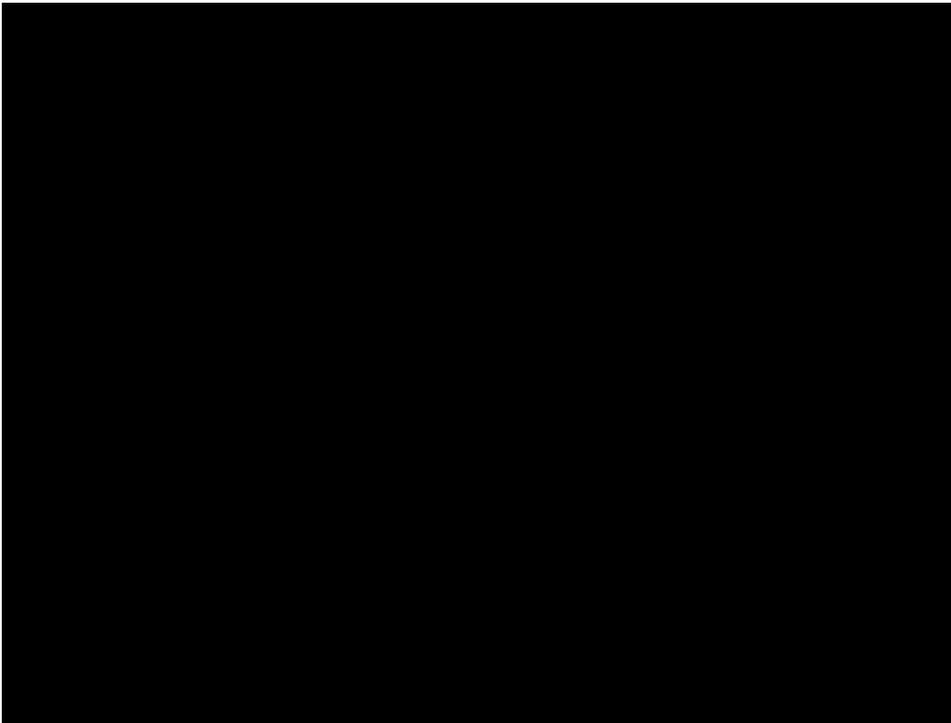
int16 det( qmatrix<int16> A ) {
    int n = size(A);
    return  $\sum_{i \in S_n} (\text{sgn}(i) \prod_{k=1}^n a_{k,i(k)})$ ;
}

```



4 Available
Tooling

**Intentional Software
Domain
Workbench**



 **INTENTIONAL[®]**
SOFTWARE

**Commercial
Product**

Eval available
upon request

A rectangular box with a light blue gradient background and a thin black border. In the top-left corner is the Intentional Software logo, which consists of a blue square containing a white stylized 'I' and 'S' followed by the text 'INTENTIONAL[®]' and 'SOFTWARE' on two lines. The main text in the center of the box reads 'Commercial Product' in a large, bold, black font, and 'Eval available upon request' in a smaller, bold, black font below it.

Pension Workbench Example

The screenshot shows the 'Caggenini Pension Workbench' application. The main window displays the 'Library NN LC PA' with a 'Documentation' section. The '1 Inleiding' (Introduction) section is selected, containing text about the scope of the pension plan and the methods used for valuation. The left sidebar shows a 'Table of Contents' with various value sets and tag definitions.

1 Inleiding

In dit onderdeel wordt uiteengezet hoe de wijze van groottebepaling van toezeggingen plaatsvindt binnen het NN Comfort Pensioen. Dit wordt bepaald door de Groottebepalingsmethode.

Binnen het NN Comfort Pensioen worden de volgende methoden gebruikt:

- Salaris-diensttijd;
- Verzekerde bedragen;
- Afgeleide toezegging;

Daarnaast is er sprake van een onderscheid per toezegging van toezeggingen met waardeopbouw en toezeggen op risicobasis. Welke groottebepalingsmethoden van toepassingen kunnen zijn en op welke wijze deze worden verwerkt, verschilt tussen opbouwtoezeggingen en risicotoezeggingen. Het onderscheid wordt gemaakt met de Indicate Opbouw / Risico.

2 Opbouwtoezeggingen

Binnen het NN Comfort Pensioen zijn de toezeggingen in de basisregeling allen met waarde-opbouw te sluiten. Zowel het Ouderdomspensioen, het Partnerspensioen als het Wezenpensioen. In de basisregeling bestaat ook de mogelijkheid het Partnerspensioen als risico te verzekeren, waarbij ook het Wezenpensioen op basis van éénjarig risico wordt verzekerd.

De opbouw wordt vastgelegd aan de hand van de volgende attributen:

- Bedrag jaaropbouw;
- Delta deelaanspraak uit mutatie;
- Delta deelaanspraak uit doorbouw;
- Deelaanspraak uitzicht;
- Deelaanspraak opgebouwd;
- Deelaanspraak gefinancierd;
- Verzekerd bedrag (*);

*1.144. Voorwaarden hebben invloed op de berekening van de waarde van de opbouwtoezeggingen. Het onderscheid wordt gemaakt door de Partnameinstellingen.

Text Editing Domain

Pension Workbench Example

The screenshot shows the 'Caggenini Pension Workbench' application. The main window displays the 'Library NN LC PA' with a 'Documentation' section. The '3.3 Commutatietallen op 1 leven' (Commutation values on 1 life) section is selected, showing mathematical formulas for calculating commutation values. The left sidebar shows a 'Table of Contents' with various value sets and tag definitions.

3.3 Commutatietallen op 1 leven

$$D_x = v \cdot \frac{l_x}{100} \quad \approx 6 \text{ Dec (3)}$$

Implemented in **1040**

$$N_x = \sum_{t=0}^{\omega-x} D_{x+t} \quad \approx 7 \text{ Dec (3)}$$

3.6 Contante waarde 1 leven/ 2 levens

$$E_x = \frac{x-n}{D_x} \quad \approx 19 \text{ Dec (4)}$$

$$a_x = a_x - 1 \quad \approx 21 \text{ Dec (3)}$$

$$\ddot{a}_x = a_x - 0,5 \quad \approx 22 \text{ Dec (3)}$$

$$\ddot{a}_{\overline{N}|x} = \frac{N_x - N_{x+n}}{D_x} \quad \approx 23 \text{ Dec (3)}$$

$$\ddot{a}_{\overline{m}|} = \ddot{a}_{\overline{m}|} - 0,5 + 0,5 \cdot E_x \quad \approx 25 \text{ Dec (3)}$$

4 BN(_ris) koopsoorten

Insurance Mathematics Domain

Pension Workbench Example

The screenshot shows the Pension Workbench application window. The main area displays a table with columns: Name, Valid time, Transaction time, Future, Product, Element, Expected value, and Actual value. The table contains three rows of data for different periods.

Name	Valid time	Transaction time	Future	Product	Element	Expected value	Actual value
Geleijde datums	03/01/2008		Mutateperiode - Mutatedatum = Mutatedatum Vorig			3	0
Periode < 30	03/01/2008		Mutateperiode - Mutatedatum > Mutatedatum Vorig (binnen 1 maand)			15	15
Periode > 30	03/01/2008		Mutateperiode - Mutatedatum > Mutatedatum Vorig (meerdere maanden)			60	60

Pension Contract Rules Domain

Pension Workbench Example

The screenshot shows the Pension Workbench application window with documentation and mathematical formulas. The documentation includes sections for 'Groottebepaling' and 'Opbouwtoe'. The mathematical formulas are for commutation factors and annuities.

3.3 Commutatietallen op 1 leven

$$D_x = v \cdot \frac{i}{1+i} = 6 \text{ Dec (3)}$$

$$N_x = \sum_{t=0}^{\infty} D_{x+t} = 7 \text{ Dec (3)}$$

3.6 Contant waarde 1 leven/ 2 levens

$$E_{\overline{n}|x} = \frac{1 - v^n}{i} = 19 \text{ Dec (4)}$$

$$a_x = \frac{1 - v}{i} = 21 \text{ Dec (3)}$$

$$\ddot{a}_x = \frac{1 - v}{i} = 22 \text{ Dec (3)}$$

$$\ddot{a}_{\overline{n}|x} = \frac{1 - v^n}{1 - v} = 23 \text{ Dec (3)}$$

$$\ddot{a}_{\overline{n}|x} = \frac{1 - v^n}{1 - v} = 25 \text{ Dec (3)}$$

4 BN(ris) koopsommen

All in one Document

Pension Workbench Example

Library NN LC PA

Documentation

- Groottebepaling**

In dit onderdeel wordt uiteengezet hoe de wijze van groottebepaling van toezeggingen plaatsvindt bepaald door de Groottebepalingsmethode.

Binnen het NN Comfort Pensioen worden de volgende methoden gebruikt:

 - Salaris-diensttijd;
 - Verzekerde bedragen;
 - Afgeleide toezegging;

Daarnaast is er sprake groottebepalingsmethode met risicotoezeggingen.
- 2 Opbouwtoe**

Binnen het NN Comfort het Partnerpensioen is ook het Wezenpensioen.

De opbouw wordt vast:

 - Bedrag jaaropbouw;
 - Delta deelaanspraak;
 - Delta deelaanspraak;
 - Deelaanspraak uitzi;
 - Deelaanspraak opge;
 - Deelaanspraak gefin;
 - Verzekerd bedrag (*

3.3 Commutatietallen op 1 leven

$$D_x = v^x \cdot \frac{l_x}{100} \approx 6 \text{ Dec (3)}$$

Implemented in **1940**

$$\omega - x$$

$$N_x = \sum_{t=0}^{\omega-x} D_{x+t} \approx 7 \text{ Dec (3)}$$

3.6 Contant waarde 1 leven/ 2 levens

$$\frac{D}{N_x} = \frac{x-n}{D_x} \approx 19 \text{ Dec (4)}$$

$$\bar{a}_x = \bar{a}_x - 1 \approx 21 \text{ Dec (3)}$$

$$\bar{a}_x = \bar{a}_x - 0,5 \approx 22 \text{ Dec (3)}$$

$$\frac{N_x - N_{[x:n]}}{D_x} \approx 23 \text{ Dec (3)}$$

$$\bar{a}_{[x:n]} = \bar{a}_x - 0,5 + 0,5 \cdot \frac{E_x}{N_x} \approx 25 \text{ Dec (3)}$$

4 BN(ris) koopsommen

Name	Valid time	Transaction time	Fixture
Gelijke datums	03/01/2008		Mutatatieperiode - Mutatiedatum - Mutatiedatum Vorig
Periode < 30	03/01/2008		Mutatatieperiode - Mutatiedatum - Mutatiedatum Vorig (Binnen 1 maand)
Periode > 30	03/01/2008		Mutatatieperiode - Mutatiedatum - Mutatiedatum Vorig (meerdere maanden)

Symbolically integrated

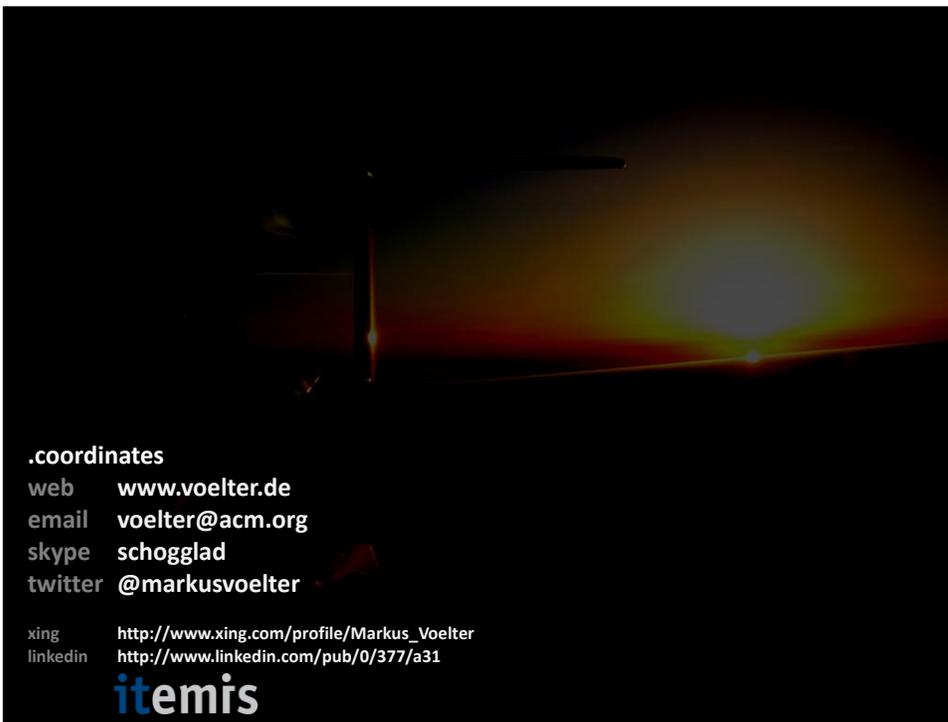




.coordinates
web www.voelter.de
email voelter@acm.org
skype [schogglad](https://www.skype.com/people/schogglad)
twitter [@markusvoelter](https://twitter.com/markusvoelter)

xing http://www.xing.com/profile/Markus_Voelter
linkedin <http://www.linkedin.com/pub/0/377/a31>

itemis



.coordinates
web www.voelter.de
email voelter@acm.org
skype [schogglad](https://www.skype.com/people/schogglad)
twitter [@markusvoelter](https://twitter.com/markusvoelter)

xing http://www.xing.com/profile/Markus_Voelter
linkedin <http://www.linkedin.com/pub/0/377/a31>

itemis

.coordinates

web www.voelter.de
email voelter@acm.org
skype [schogglad](https://www.skype.com/people/schogglad)
twitter [@markusvoelter](https://twitter.com/markusvoelter)

xing http://www.xing.com/profile/Markus_Voelter
linkedin <http://www.linkedin.com/pub/0/377/a31>

itemis