CWI Centrum Wiskunde & Informatica

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE | INRIA

The mechanics of building a DSL using Rascal

Jurgen J. Vinju

IPA spring days

April 17th 2012

Rascal Team

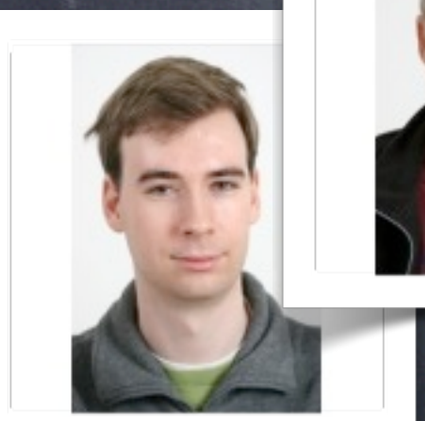
Paul Klint

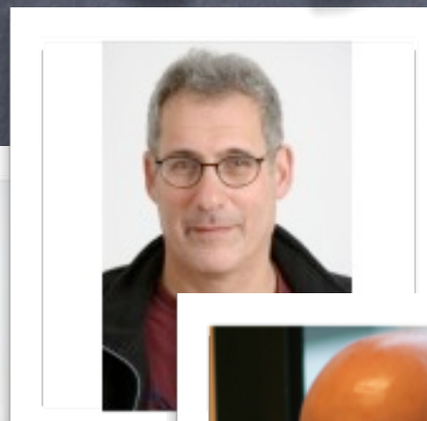
Jurgen Vinju

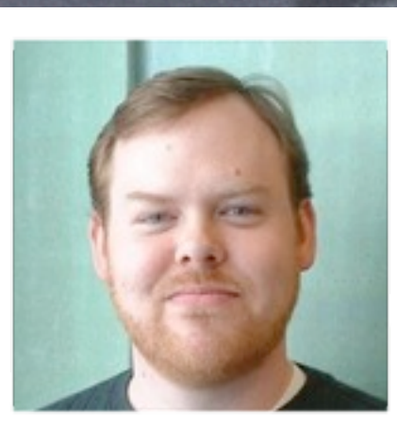
Tijs v/d Storm

Bob Fuhrer

IMP







Credits

How Tijs & I were drafted...

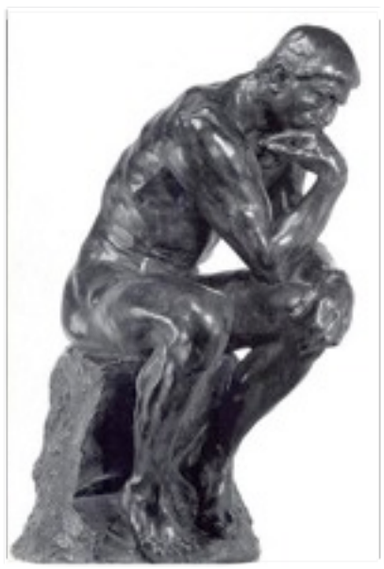
- Esprit: GIPE I & GIPE II (90's)
- ASF+SDF Meta-Environment (00's)
- Eclipse
- IDE Meta Tooling Platform (IMP)
- Rascal is a part of IMP now
- Rascal draws inspiration from countless other projects (see SCAM 2009 paper for references)

"Generation of Interactive Programming Environments"

Why we need Rascal @CWInl

Meta Tool

Tools



Research

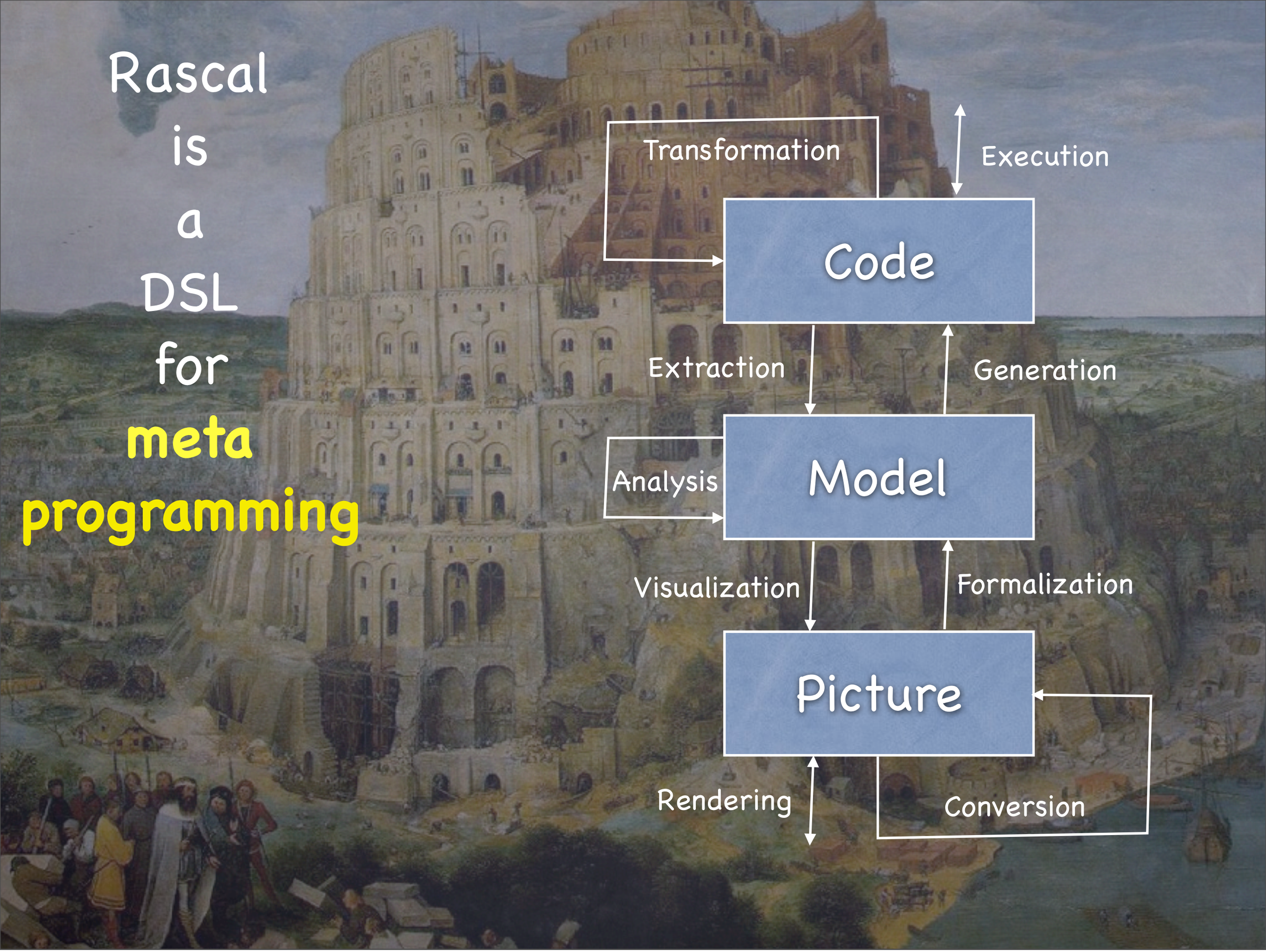
Application

Every week
a new tool
a new DSL

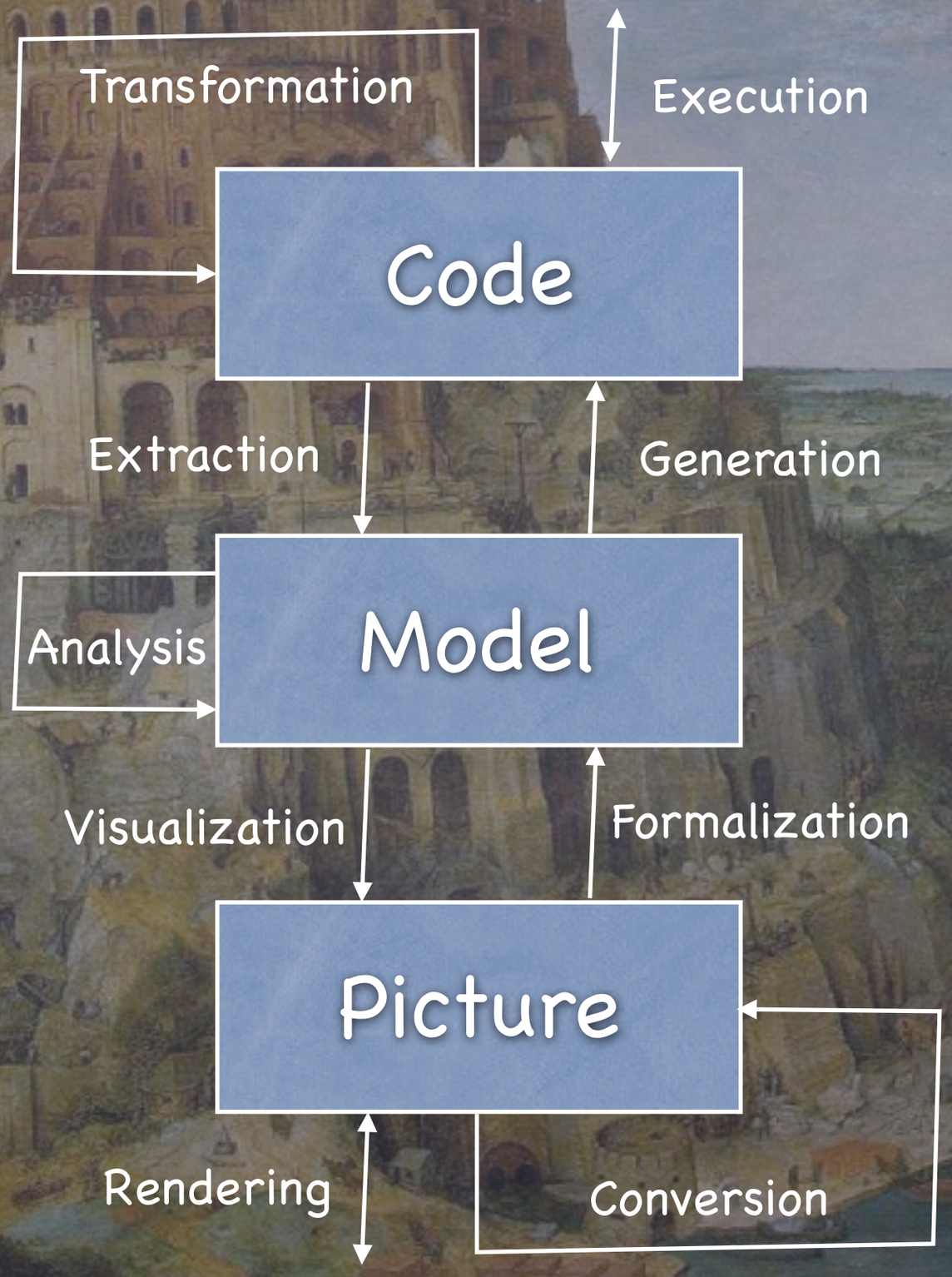
Story

- We focus on
  - textual DSLs (not graphical)
  - external DSLs (not embedded/internal)
  - executability via code generation (not interpreted)
- This talk is about
  - source file in, source file out
  - language prototyping
  - building a DSL using a DSL

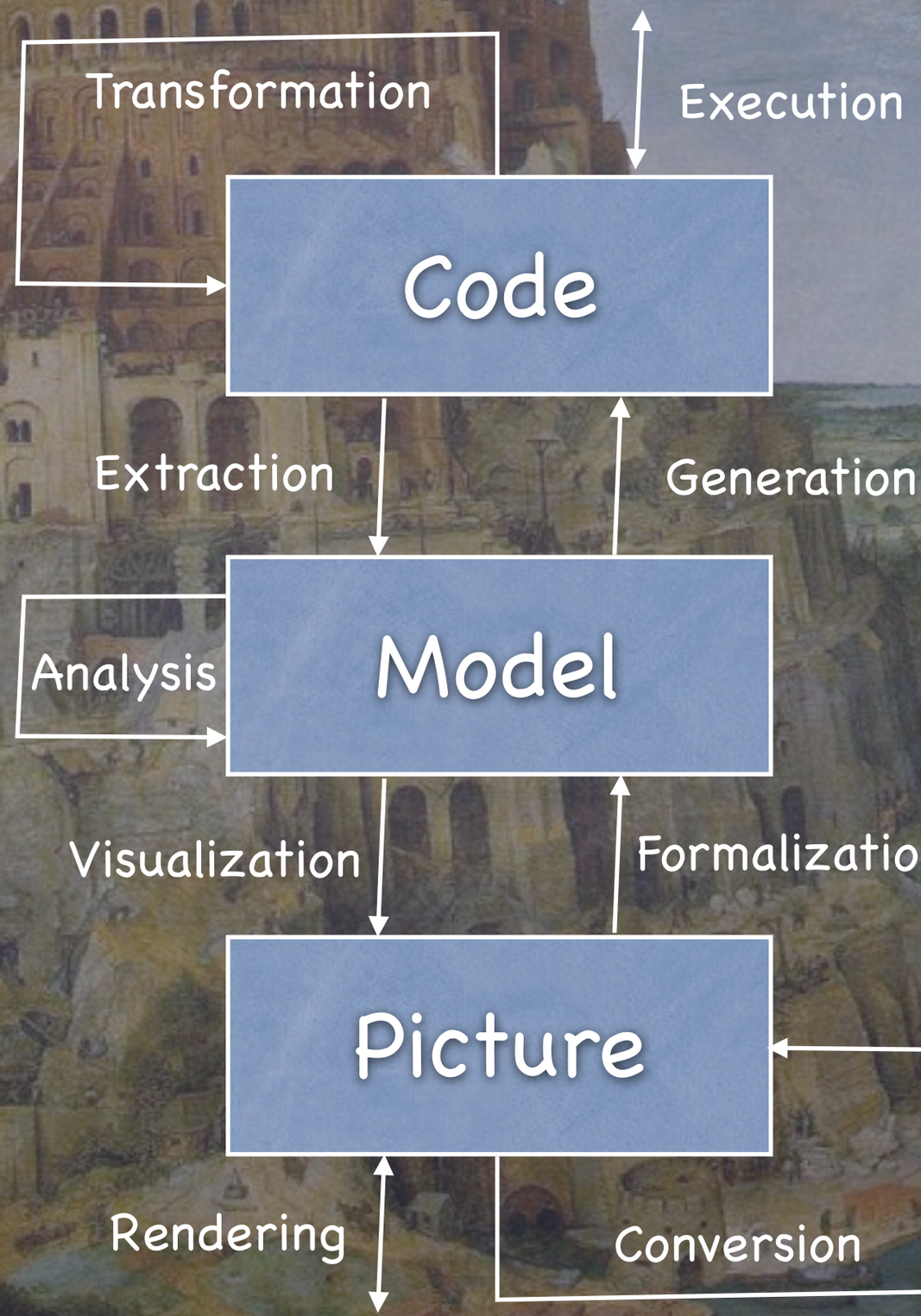
Rascal is a DSL for **meta programming**

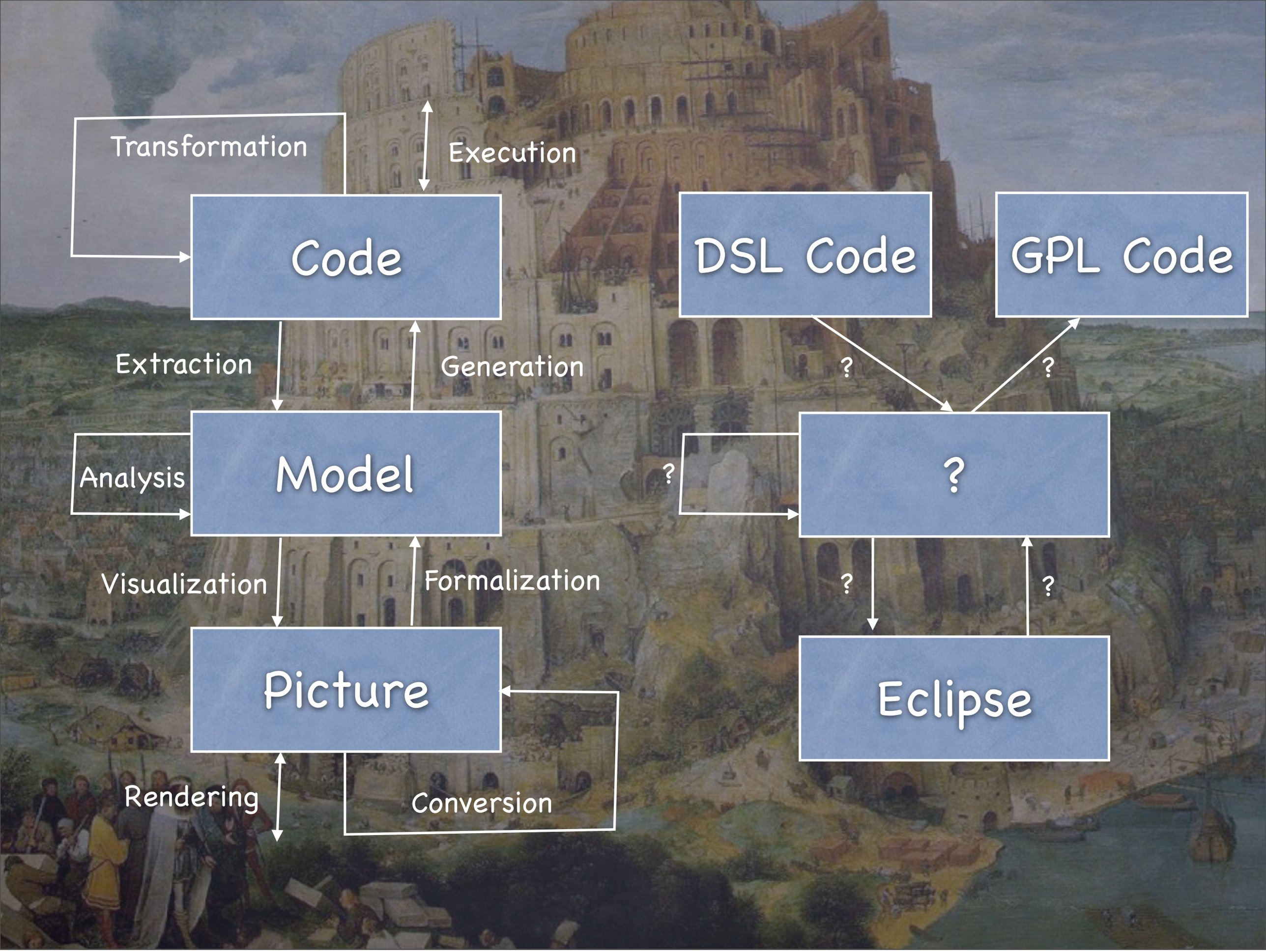


Rascal is a DSL for **meta programming** = moving between representations of source code

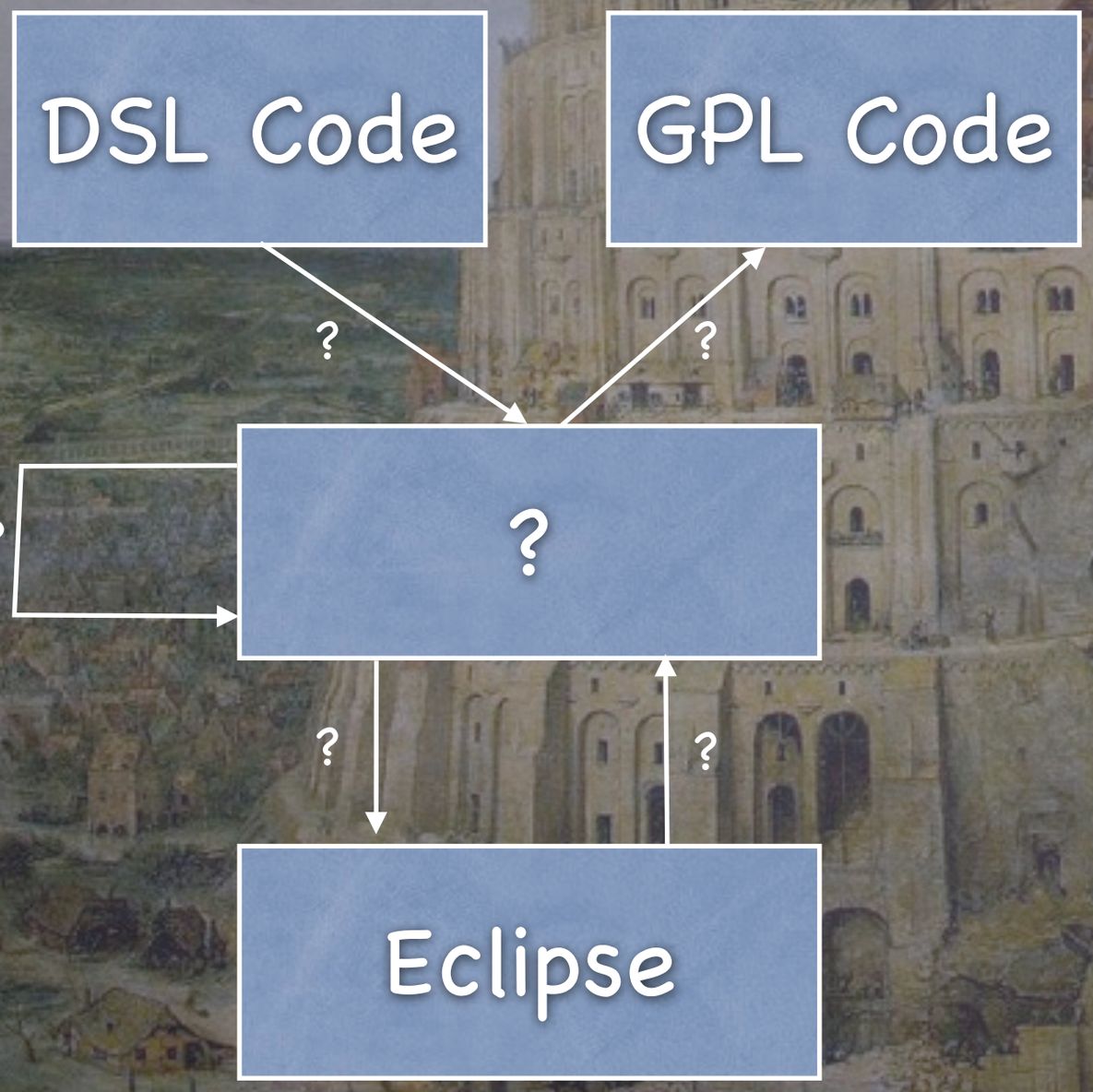


Static analysis
Software metrics
Model extraction
Model-to-model
Model-to-code
Code-to-model
Compilation
Code generation
Visualization
Parsing
Source-to-source
etc.




Transformation
Execution
Code
DSL Code
GPL Code
Extraction
Generation
?
?
Analysis
Model
?
?
Visualization
Formalization
?
?
Picture
Eclipse
Rendering
Conversion

Choices



- **external** design
  - source input
  - source **output**
- **internal** mechanics
  - parsing
  - models
  - stages
  - analysis
  - optimization
  - user feedback
  - IDE

External design is wicked

Rascal
is about making
the mechanics
of the internal design
so easy to manipulate
that you can
experiment with the
external design
without much
punishment;
modularly,
incrementally

Rascal
is about making
the mechanics
of the internal design
so easy to manipulate
that you can
experiment with the
external design
without much
punishment;
modularly,
incrementally

Rascal does **not** solve the external **design** problem, it **does** solve the internal **mechanics** problem

Design Principles (a.k.a. Requirements)

- Scaling up and scaling down
- Expressivity without magic
- One-stop-shop (integrated)
- Open
- Debuggable/Understandable

Simple = Simple

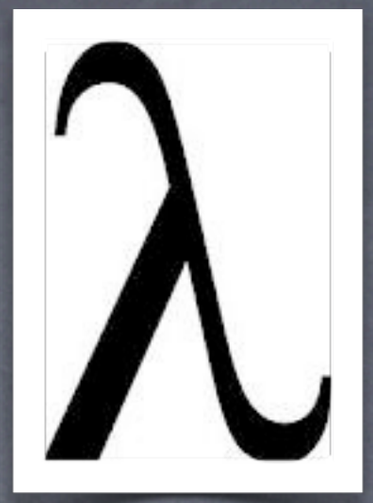
Advanced = Accessible

Meta Programming = Programming

REPL

Integration = Valuable

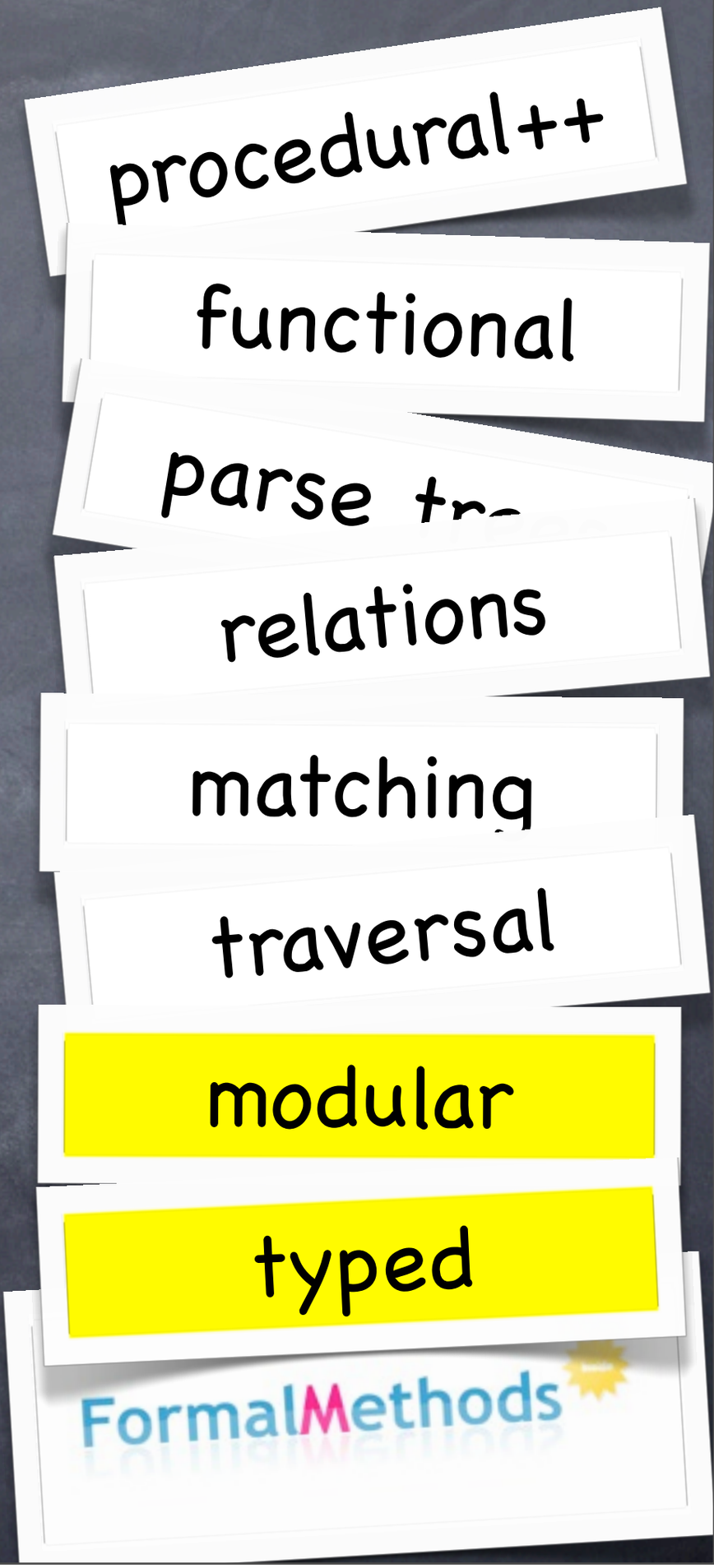
What's good about programming



- Rascal provides primitives for constructing the mechanics of meta programs
- It is a high level **programming** language
- This means that
  - it looks & feels like a programming language
  - and the code can be very concise
  - and you still control what happens
  - and you can still observe what happens at run-time

Ingredients

- Familiar syntax & notation
- Immutable data
- Pattern-based dispatch
- Domain specific data-types
- General context-free grammars
- String templates
- Java back door
- IDE integration via Eclipse IMP



Ingredients

Code	Model	Tools
Source Text Files	Relations	CF Grammars
Parse Trees	Sets	Relational Queries
Abstract Syntax Trees	Maps	Pattern matching
	ADTs	Traversal
		Templates
		Concrete syntax
		Closures

"E.A.Sy" error messages

Source Text File

Grammar

Concrete syntax

Parse Trees

Implode

Abstract Syntax Tree

Error Messages

Queries

Relations

Traversal

Pattern matching

"E.A.Sy" code generation

Source Text File

Parse Trees

Abstract Syntax Tree

Traversal

matching

Templates

Relations

Queries

Error Messages

Target Text File

"E.A.Sy" IDE instantiation

Source Text File

Parse Trees

Abstract Syntax Tree

Relations

Error Messages

Target Text File

Closures

highlights

outline

menus

...

Eclipse IMP callbacks

Eclipse @^$@

Summary of DSLs with Rascal

- **Parse** using context-free grammars
- **Extract** info using patterns, traversals
  - either concrete or abstract syntax
- **Query** using relational operators
- **Generate** code using templates
- callback **Closures** to interact with Eclipse

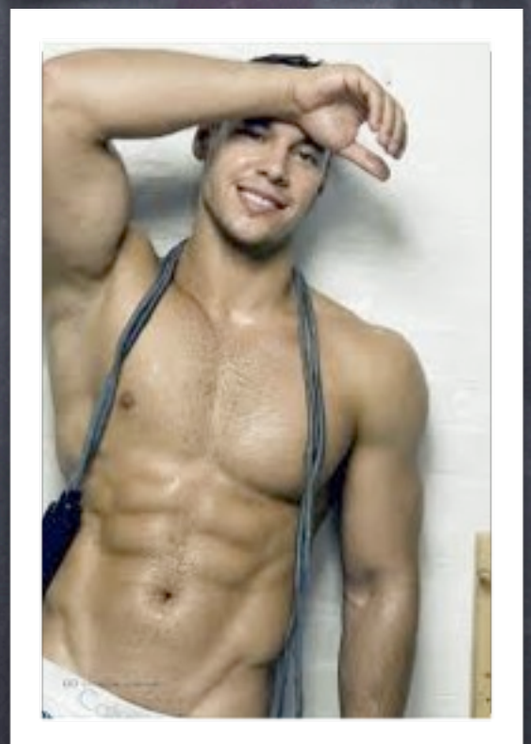
Summary of DSLs with Rascal

- **Parse** using context-free grammars
- **Extract** info using patterns, traversals
  - either concrete or abstract syntax
- **Query** using relational operators
- **Generate** code using templates
- callback **Closures** to interact with Eclipse

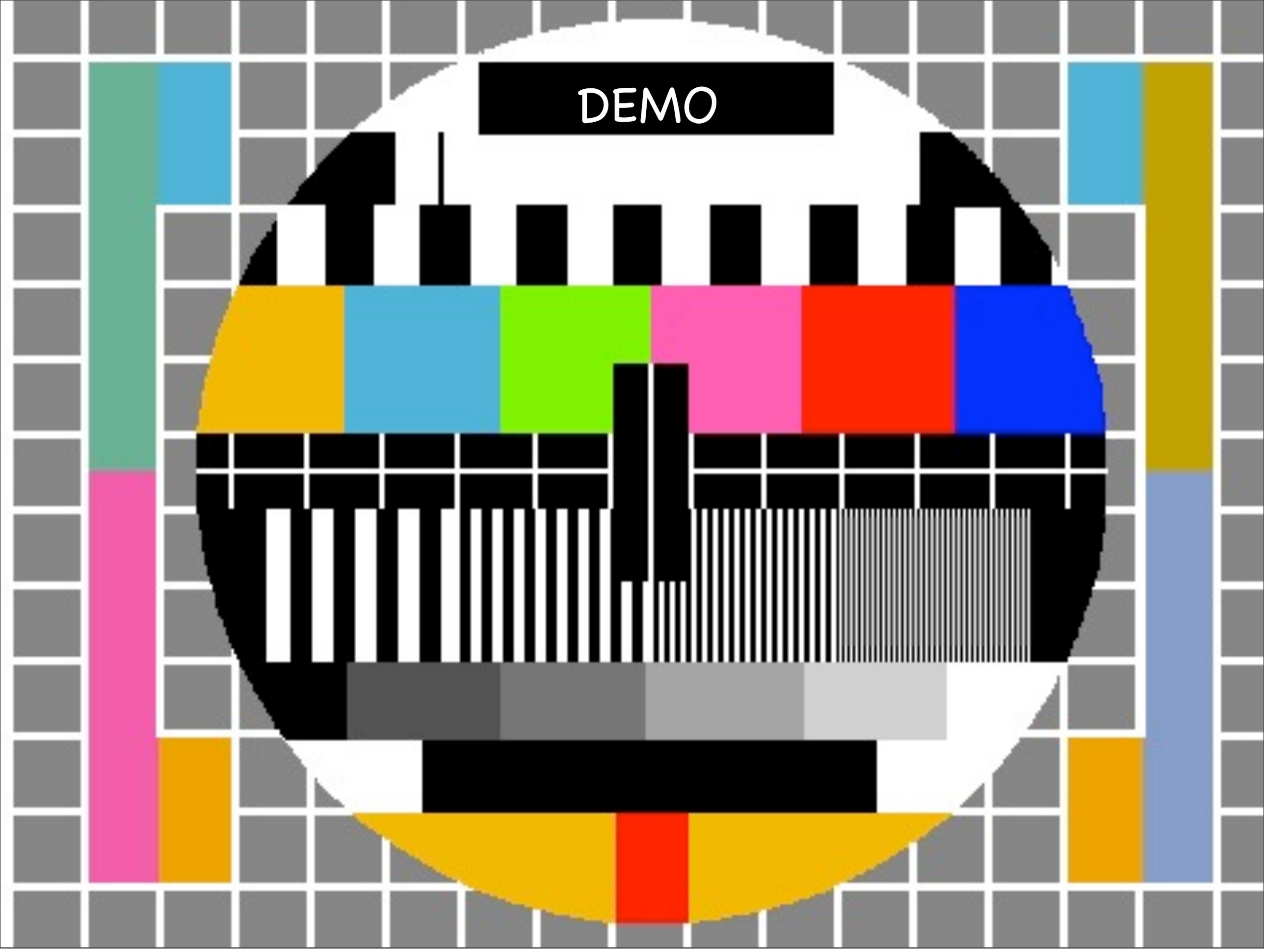
Miss Grant

Martin Fowler

- **State Machine Language**
- Static analysis for error messages



- Code generators for different designs
  - methods are states
  - switch
  - object-oriented (dynamic) representation
- State machine visualization


DEMO

Design Principles (a.k.a. Requirements)

- Scaling up and scaling down
- Expressivity without magic
- One-stop-shop (integrated)
- Open
- Immediate

No Database

No Coordination

No Diagrams

No Algorithms

Just Control flow

Just Matching

Just EBNF

Just Tracing

Just Trees

Just Profiling

Just Relations

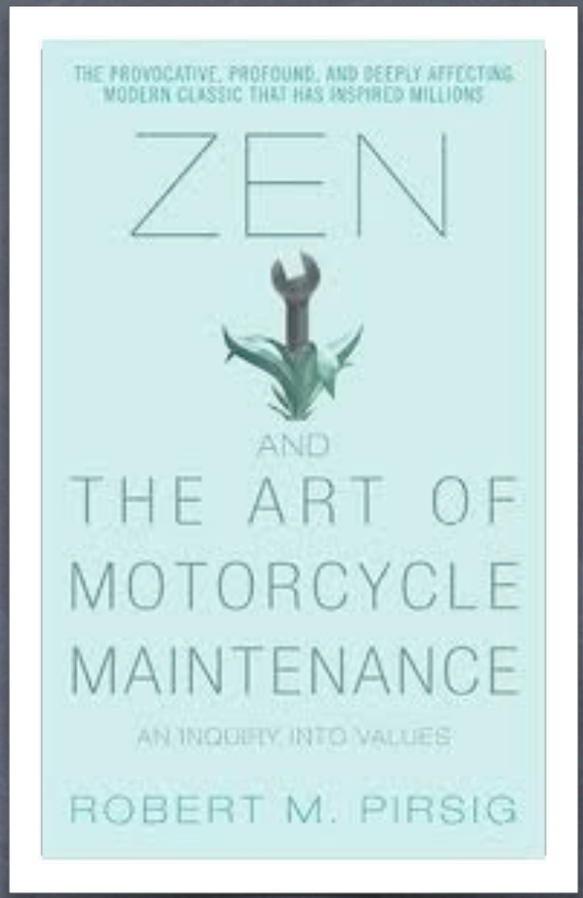
Just Debugging

Just IMP

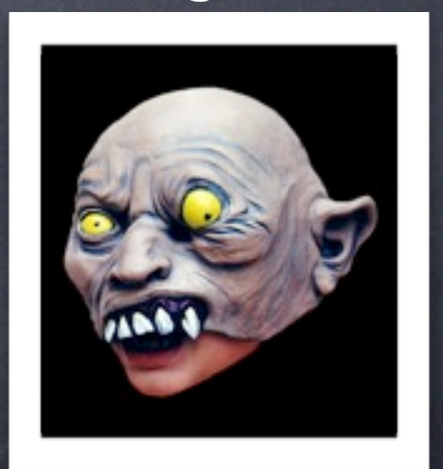
Other applications

- Java refactorings/generics [SCAM2009, N. Izmaylova]
- Source code Visualization
- Oberon-0 compiler in 4 levels [LDTA2011]
- Language Workbench Competitions '11 en '12
- Derric: DSL for CSI (Digital Forensics) [ICSE 2011]
- Visitor 2 Interpreter refactoring [TOOLS 2011]
- PHP backward compat analysis
- Debunking McCabe
- Student projects (Java analysis, SVN analysis, ...)
- Maude/K-framework collaboration
- etc.

Quality of Rascal S.W.O.T.



- [S] simple, powerful, immediate, integral
- [W] alpha/beta quality, slow
- [O] optimization, applications, collaboration, growing team
- [T] the feature creep



Immediate future

- **Applications** in Software Analysis & Transformation
- **Applications** in DSL development and evaluation
- Move to **Eclipse.org**
- More **grammarware** (CASE for grammars)
- More source/model **visualization**
- More front-ends and back-ends for GPLS

Questions?

CWI Centrum Wiskunde & Informatica

http://www.rascal-mpl.org

http://www.eclipse.org/imp

http://www.cwi.nl/sen1

http://tutor.rascal-mpl.org

interactive docs

http://ask.rascal-mpl.org

stackoverflow

Questions?

CWI Centrum Wiskunde & Informatica

http://www.rascal-mpl.org

http://www.eclipse.org/imp

http://www.cwi.nl/sen1

http://tutor.rascal-mpl.org

interactive docs

http://ask.rascal-mpl.org

stackoverflow

Demo Outline

- **First** Entities & Instances languages
  - Immediate IDE: highlighting, folding, error marking, ...
  - Java and SQL generation
  - Online checking and error marking
- **Then** modular extensions
  - Packages: Source-to-source transformation
  - Derived I: expression language extension
  - Derived II: linking to host language using annotations

Rascal - lwc11/input/person.entities - Eclipse Platform

Package Explorer

- lwc11
  - eclipse
  - input
    - car.entities
    - car.package
    - example.entities
    - person.entities
    - person.package
    - product.derived
    - voelter.instances
  - output
  - src
    - lang
      - database
      - derived
      - entities
      - instances
      - packages
    - Plugin.rsc
  - std
- oberon0

Plugin.rsc | person.entities

```
1  entity Person {
2      string name
3      string firstName
4      date birthDate
5      Car2 ownedCar
6  }
7
8  entity Car {
9      string make
10     string model
11 }
```

Undefined reference
Press 'F2' for focus

Outline

- Person
  - name
  - firstName
  - birthDate
  - ownedCar
- Car
  - make
  - model

Problems | Console | Error Log | Progress | Console

Store history | Terminate | Interrupt | Trace

Rascal [lwc11]

```
rascal>import Plugin;
ok
rascal>main();
ok
rascal>
```

Writable | Smart Insert | 5 : 7 | 315M of 475M

Eclipse File Edit Source Release Engineering Navigate Search Project Run Rascal Window Help

Rascal – lwc11/input/person.entities – Eclipse Platform

Rascal SVN Reposit... Java

Package Explorer

- lwc11
  - eclipse
  - input
    - car.entities
    - car.package
    - example.entities
    - person.entities
    - person.package
    - product.derived
    - voelter.instances
  - output
  - src
    - lang
      - database
      - derived
      - entities
      - instances
      - packages
    - Plugin.rsc
  - std
- oberon0

Plugin.rsc | person.entities

```
1  entity Person {
2      string name
3      string firstName
4      date birthDate
5      Ca
6  }
7
8  entity
9      st
10     st
11 }
```

Outline

- Person
  - name
  - firstName
  - birthDate
  - ownedCar
- Car
  - make
  - model

Undo Typing ⌘Z
Revert File
Save ⌘S
Show In ⌥⌘W
Cut ⌘X
Copy ⌘C
Paste ⌘V
Quick Fix ⌘1
Shift Right
Shift Left
Source
Add to Snippets...
Run As
Debug As
Coverage As
Validate
Team
Compare With
Replace With
Pretty Print
WikiText
Preferences...
Entities
Remove from Context ⌥⇧⌘↓

- Generate Java
- Generate XML
- Generate SQL

Problems | Conso

Store history Terminate Interrupt Trace

Rascal [lwc11]

```
rascal>im
ok
rascal>ma
ok
rascal>
```

Writable Smart Insert 5 : 8 315M of 475M

Eclipse File Edit Release Engineering Source Refactor Navigate Search Project Run Rascal Window Help

Rascal - lwc11/output/Person.java - Eclipse Platform

Rascal SVN Reposit... Java

Package Explorer

- lwc11
  - eclipse
  - input
    - car.entities
    - car.package
    - example.entities
    - person.entities
    - person.package
    - product.derived
    - voelter.instances
  - output
    - Car.java
    - car.sql
    - car.xml
    - Car2.java
    - Car3.java
    - Car4.java
    - Person.java
    - person.Person.java
    - person.xml
    - Product.java
  - src
    - lang
      - database
      - derived
      - entities
      - instances
      - packages
    - Plugin.rsc
  - std
- oberon0

person.entities | Person.java

```
1  public class Person {
2
3    private java.lang.String name;
4    public java.lang.String getName() {
5        return this.name;
6    }
7    public void setName(java.lang.String name) {
8        this.name = name;
9    }
10
```

Car.java

```
1  public class Car {
2
3    private java.lang.String make;
4    public java.lang.String getMake() {
5        return this.make;
6    }
7    public void setMake(java.lang.String make) {
8        this.make = make;
9    }
10
```

Outline

- Person
  - name : String
  - getName() : String
  - setName(String) : v
  - firstName : String
  - getFirstName() : St
  - setFirstName(Strin
  - birthDate : Date
  - getBirthDate() : Da
  - setBirthDate(Date)
  - ownedCar : Car
  - getOwnedCar() : C
  - setOwnedCar(Car)

Problems | Console | Error Log | Progress | Console

Store history Terminate Interrupt Trace

Rascal [lwc11]

```
rascal>import Plugin;
ok
rascal>main();
ok
```

Writable | Smart Insert | 1 : 1 | 317M of 475M

Rascal – lwc11/output/person.sql – Eclipse Platform

Package Explorer

- lwc11
  - eclipse
  - input
    - car.entities
    - car.package
    - example.entities
    - person.entities
    - person.package
    - product.derived
    - voelter.instances
  - output
    - Car.java
    - car.sql
    - car.xml
    - Car2.java
    - Car3.java
    - Car4.java
    - Person.java
    - person.Person.java
    - person.sql
    - person.xml
    - Product.java
  - src
    - lang
      - database
      - derived
      - entities
      - instances
      - packages
    - Plugin.rsc
  - std
- oberon0

person.entities | person.sql

```
1
2  create table Person (
3    _id int primary key,
4    name varchar ,
5    firstName varchar ,
6    birthDate date ,
7    ownedCar int foreign key references Car(_id)
8  );
9
10 create table Car (
11   _id int primary key,
12   make varchar ,
13   model varchar
14 );
15
16
```

Outline

An outline is not availa

Problems | Console | Error Log | Progress | Console

Store history | Terminate | Interrupt | Trace

Rascal [lwc11]

```
rascal>import Plugin;
ok
rascal>main();
ok
```

Writable | Insert | 1 : 1 | 336M of 475M

Rascal - lwc11/src/lang/entities/syntax/Entities.rsc - Eclipse Platform

Rascal | SVN Reposit... | Java

Package Explorer

- lwc11
  - eclipse
  - input
  - output
  - src
    - lang
      - database
      - derived
      - entities
        - ast
        - check
        - compile
        - ide
        - render
        - syntax
          - Entities.rsc
          - Ident.rsc
          - Layout.rsc
          - Types.rsc
        - transform
        - utils
      - instances
      - packages
    - Plugin.rsc
  - std
  - oberon0

person.entities | **Entities.rsc**

```
1  module lang::entities::syntax::Entities
2
3  import lang::entities::syntax::Layout;
4  import lang::entities::syntax::Ident;
5  import lang::entities::syntax::Types;
6
7  start syntax Entities
8      = entities: Entity* entities;
9
10 syntax Entity
11     = @Foldable entity: "entity" Name name "{" Field* "}";
12
13 syntax Field
14     = field: Type Ident name;
```

Outline

- Variables
- Functions
- Types
- Aliases
- Rules
- Annotations
- Tags
- Imports
  - lang::entities::syntax::Layo
  - lang::entities::syntax::Ider
  - lang::entities::syntax::Typ
- Syntax
  - Entities
  - Entity
  - Field

Problems | **Console** | **Error Log** | **Progress** | **Console**

Store history | Terminate | Interrupt | Trace

Rascal [lwc11]

```
rascal>import lang::entities::syntax::Types;
ok
rascal>`entity X { }`
`entity X { }`
Entity: appl(prod([lit("entity"),layouts("LAYOUTLIST"),label("name",sort("N
rascal>
```

332M of 475M

Rascal – lwc11/src/lang/entities/ast/Entities.rsc – Eclipse Platform

Package Explorer

- lwc11
  - eclipse
  - input
  - output
  - src
    - lang
      - database
      - derived
      - entities
        - ast
          - Entities.rsc
        - check
        - compile
        - ide
        - render
        - syntax
          - Entities.rsc
          - Ident.rsc
          - Layout.rsc
          - Types.rsc
        - transform
        - utils
      - instances
      - packages
    - Plugin.rsc
  - std
- oberon0

person.entities | Entities.rsc | Entities.rsc

```
1  module lang::entities::ast::Entities
2
3  data Entities
4      = entities(list[Entity] entities);
5
6  data Entity
7      = entity(Name name, list[Field] fields);
8
9  data Field
10     = field(Type \type, str name);
11
12 data Type
13     = primitive(PrimitiveType primitive)
14     | reference(Name name);
```

Outline

- Syntax
- Imports
- Variables
- Functions
- Aliases
- Rules
- Tags
- Types
  - Entities
  - Entity
  - Field
  - Type
  - Name
  - PrimitiveType
- Annotations
  - anno loc Entities@location
  - anno loc Entity@location;
  - anno loc Field@location;
  - anno loc Type@location;
  - anno loc Name@location;

Problems | Console | Error Log | Progress | Console — Store history | Terminate | Interrupt | Trace

Rascal [lwc11]

```
int: 2
rascal>2 + 2
int: 4
rascal>3 + 3
int: 6
rascal>
```

334M of 475M

Rascal – lwc11/src/lang/entities/compile/Entities2Java.rsc – Eclipse Platform

Package Explorer

- lwc11
  - eclipse
  - input
  - output
  - src
    - lang
      - database
      - derived
      - entities
        - ast
        - check
        - compile
          - Entities2Java.rsc
          - Entities2XML.rsc
        - ide
        - render
        - syntax
        - transform
        - utils
      - instances
      - packages
    - Plugin.rsc
  - std
- oberon0

person.entities | Entities.rsc | Entities.rsc | Entities2Java.rsc

```
10public str entity2java(Entity e) {
11    return "public class <e.name.name> {
12           '<for (f <- e.fields) {>
13           '  <field2java(f)>
14           '<}>
15           '}";
16}
17
18
19public str field2java(Field::field(typ, n)) {
20    <t, cn> = <type2java(typ), capitalize(n)>;
21    return "private <t> <n>;
22           'public <t> get<cn>() {
23           '    return this.<n>;
24           '}
25           'public void set<cn>(<t> <n>) {
26           '    this.<n> = <n>;
27           '}";
28}
29
```

Outline

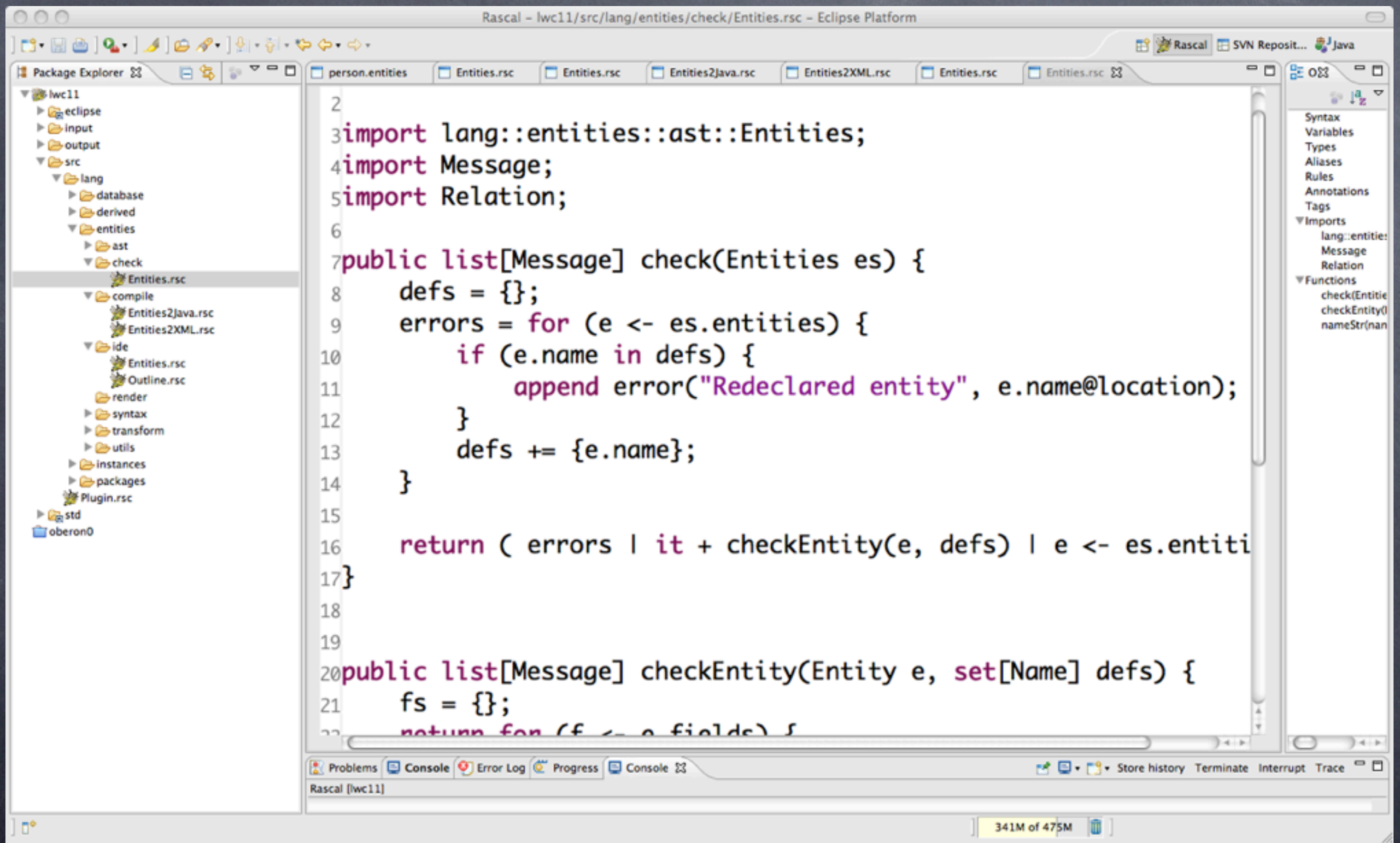
- Syntax
- Variables
- Types
- Aliases
- Rules
- Annotations
- Tags
- Imports
  - lang::entities::ast::Entities
  - String
- Functions
  - entities2java(Entities es)
  - entity2java(Entity e)
  - field2java(Field::field(typ,
  - type2java(primitive(string(
  - type2java(primitive(intege
  - type2java(primitive(boolez
  - type2java(primitive(date())
  - type2java(primitive(curren
  - type2java(reference(name
  - capitalize(str s)

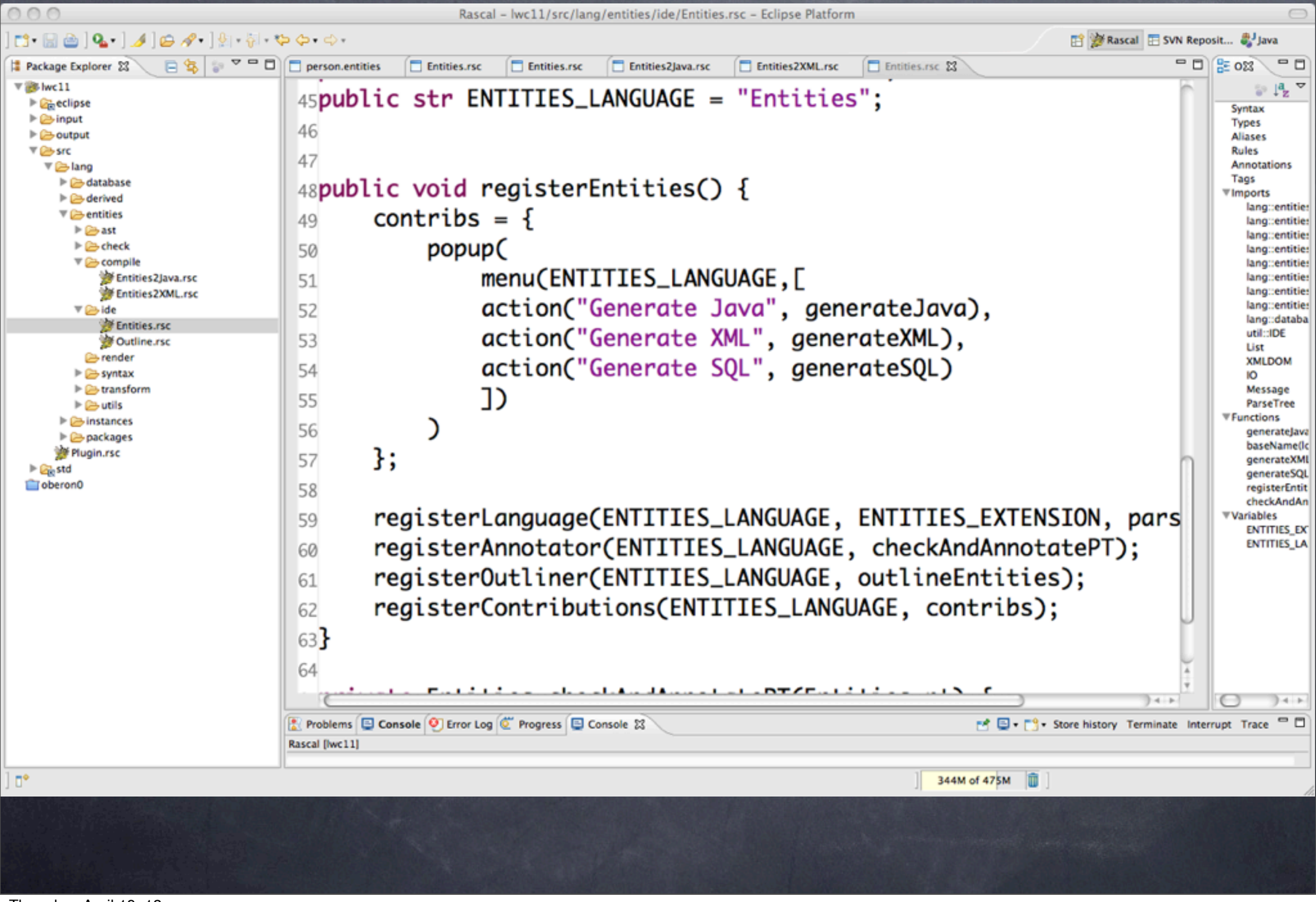
Problems | Console | Error Log | Progress | Console

Store history | Terminate | Interrupt | Trace

Rascal [lwc11]

335M of 475M


Rascal - lwc11/src/lang/entities/check/Entities.rsc - Eclipse Platform
Rascal
SVN Reposit...
Java
Package Explorer
lwc11
eclipse
input
output
src
lang
database
derived
entities
ast
check
Entities.rsc
compile
Entities2Java.rsc
Entities2XML.rsc
ide
Entities.rsc
Outline.rsc
render
syntax
transform
utils
instances
packages
Plugin.rsc
std
oberon0
person.entities
Entities.rsc
Entities.rsc
Entities2Java.rsc
Entities2XML.rsc
Entities.rsc
Entities.rsc
2
3import lang::entities::ast::Entities;
4import Message;
5import Relation;
6
7public list[Message] check(Entities es) {
8 defs = {};
9 errors = for (e <- es.entities) {
10 if (e.name in defs) {
11 append error("Redeclared entity", e.name@location);
12 }
13 defs += {e.name};
14 }
15
16 return (errors | it + checkEntity(e, defs) | e <- es.entiti
17}
18
19
20public list[Message] checkEntity(Entity e, set[Name] defs) {
21 fs = {};
Syntax
Variables
Types
Aliases
Rules
Annotations
Tags
Imports
lang::entitie
Message
Relation
Functions
check(Entitie
checkEntity(
nameStr(nan
Problems
Console
Error Log
Progress
Console
Store history
Terminate
Interrupt
Trace
Rascal [lwc11]
341M of 475M


Rascal – lwc11/src/lang/entities/ide/Entities.rsc – Eclipse Platform
Rascal
SVN Reposit...
Java
Package Explorer
lwc11
eclipse
input
output
src
lang
database
derived
entities
ast
check
compile
Entities2Java.rsc
Entities2XML.rsc
ide
Entities.rsc
Outline.rsc
render
syntax
transform
utils
instances
packages
Plugin.rsc
std
oberon0
person.entities
Entities.rsc
Entities.rsc
Entities2Java.rsc
Entities2XML.rsc
Entities.rsc
public str ENTITIES_LANGUAGE = "Entities";
public void registerEntities() {
contribs = {
popup(
menu(ENTITIES_LANGUAGE,[
action("Generate Java", generateJava),
action("Generate XML", generateXML),
action("Generate SQL", generateSQL)
])
)
};
registerLanguage(ENTITIES_LANGUAGE, ENTITIES_EXTENSION, pars
registerAnnotator(ENTITIES_LANGUAGE, checkAndAnnotatePT);
registerOutliner(ENTITIES_LANGUAGE, outlineEntities);
registerContributions(ENTITIES_LANGUAGE, contribs);
}
Syntax
Types
Aliases
Rules
Annotations
Tags
Imports
Functions
Variables
Problems
Console
Error Log
Progress
Console
Store history
Terminate
Interrupt
Trace
Rascal [lwc11]
344M of 475M

Rascal – lwc11/src/lang/packages/syntax/Packages.rsc – Eclipse Platform

Package Explorer

- lwc11
  - eclipse
  - input
    - car.entities
    - car.package
    - example.entities
    - person.entities
    - person.package
    - product.derived
    - voelter.instances
  - output
  - src
    - lang
      - database
      - derived
      - entities
      - instances
      - packages
        - ast
        - check
        - compile
        - ide
        - resolve
        - syntax
          - Packages.rsc
        - utils
      - Plugin.rsc
  - std
  - oberon0

Entities.rsc | Entities.rsc | Entities2Java.rsc | Entities2XML.rsc | Entities.rsc | Entities.rsc | person.package | car.package | Packages.rsc

```
1  module lang::packages::syntax::Packages
2
3  import lang::entities::syntax::Entities;
4  import lang::entities::syntax::Layout;
5  import lang::entities::syntax::Types;
6  import lang::entities::syntax::Ident;
7
8  start syntax Package
9      = package: "package" Ident name "{" Import* imports Entities entities "}";
10
11 syntax Import
12     = imp: "import" Ident name;
13
14 // Extension
15 syntax Name
16     = qualified: Ident "." Ident;
```

Outline

- Variables
- Functions
- Types
- Aliases
- Rules
- Annotations
- Tags
- Imports
  - lang::entities
  - lang::entities
  - lang::entities
  - lang::entities
- Syntax
  - Package
  - Import
  - Name

Problems | Console | Error Log | Progress | Console

Store history | Terminate | Interrupt | Trace

Rascal [lwc11]

350M of 475M

Rascal – lwc11/src/lang/packages/resolve/Packages.rsc – Eclipse Platform

Entities.rsc | Entities2Java.rsc | Entities2XML.rsc | Entities.rsc | Entities.rsc | person.package | car.package | Packages.rsc | Packages.rsc

```
15//   entities are declared out of order and may have cyclic deps
16//   if a package imports 2 packages that export the same name it is an error
17//    (here we assume this has already been checked for)
18//    (resolve will actually resolve them, hiding the error)
19
20
21public WorkingSet resolve(WorkingSet pkgs) {
22    return { <k, success(l, resolvePkg(pkg, pkgs))> | <k, success(l, pkg)> <-
23}
24
25private Package resolvePkg(Package pkg, WorkingSet pkgs) {
26    imps = imports(pkg) + {pkg.name};
27    return visit (pkg) {
28        case Name n:name(str x) => qualified(ip.name, x)[@location=n@location]
29            when i <- imps, <i, success(_, Package ip)> <- pkgs, x in exports(
30    }
31}
32
33
```

Problems | Console | Error Log | Progress | Console

Store history | Terminate

Rascal [lwc11]

355M of 475M

Rascal – lwc11/src/lang/packages/compile/Package2Java.rsc – Eclipse Platform

Package Explorer

- lwc11
  - eclipse
  - input
    - car.entities
    - car.package
    - example.entities
    - person.entities
    - person.package
    - product.derived
    - voelter.instances
  - output
  - src
    - lang
      - database
      - derived
      - entities
      - instances
      - packages
        - ast
        - check
        - compile
          - Package2Java.rsc
          - Package2XML.rsc
        - ide
        - resolve
          - Packages.rsc
        - syntax
          - Packages.rsc
        - utils
    - Plugin.rsc
  - std
  - oberon0

Entities2Java.rsc | Entities2XML.rsc | Entities.rsc | Entities.rsc | person.package | car.package | Packages.rsc | Packages.rsc | Package2Java.rsc

```
2
3import List;
4
5import lang::packages::ast::Packages;
6extend lang::entities::compile::Entities2Java;
7
8public rel[str, str] package2java(Package pkg) {
9    return { <"<pkg.name>.<e.name.name>", packagedEntity2Java(pkg, e)> | e <- p
10}
11
12public str packagedEntity2Java(Package pkg, Entity e) {
13    return "package <pkg.name>;
14           '<for (i <- pkg.imports) {>
15           'import <i.name>.*;
16           '<}>
17           '<entity2java(e)>";
18}
19
20public str type2java(reference(qualified(str pkg, str name))) = "<pkg>.<name>";
```

Problems | Console | Error Log | Progress | Console

Store history | Terminate

Rascal [lwc11]

347M of 475M

Rascal – lwc11/input/product.derived – Eclipse Platform

Rascal | SVN Reposit...

ge Explorer

```
:11
eclipse
input
  car.entities
  car.package
  example.entities
  person.entities
  person.package
  product.derived
  voelter.instances
output
src
  lang
    database
    derived
    entities
    instances
    packages
      ast
      check
      compile
        Package2Java.rsc
        Package2XML.rsc
      ide
      resolve
        Packages.rsc
      syntax
        Packages.rsc
      utils
  Plugin.rsc
std
eron0
```

Packages.rsc | Packages.rsc | Package2Java.rsc | product.derived | »9

```
1  entity Product {
2      string name
3      currency price
4      currency vat = 0.19 * price
5
6      @host("tax.utils.VAT.computeVAT")
7      currency vat2
8  }
```

Outline

- Product
  - name
  - price
  - vat
  - vat2

Problems | Console | Error Log | Progress | Console

IMP Runtime

```
generating item allocations
computing priority and associativity filter
computing lookahead sets
optimizing lookahead automaton
printing the source code of the parser class
```

352M of 475M

Rascal – lwc11/output/Product.java – Eclipse Platform

Explorer

- car.entities
- car.package
- example.entities
- person.entities
- person.package
- product.derived
- voelter.instances
- Car.java
- car.sql
- car.xml
- Car2.java
- Car3.java
- Car4.java
- Person.java
- person.Person.java
- person.sql
- person.xml
- Product.java
- lang
- database
- derived
- entities
- instances
- packages
  - ast
  - check
  - compile
    - Package2Java.rsc
    - Package2XML.rsc
  - ide
  - resolve
    - Packages.rsc
  - syntax
    - Packages.rsc
  - utils

Packages.rsc | Package2Java.rsc | product.derived | Product.java | »10

```
10
11  private java.util.Currency price;
12  public java.util.Currency getPrice() {
13      return this.price;
14  }
15  public void setPrice(java.util.Currency price) {
16      this.price = price;
17  }
18
19  public java.util.Currency getVat() {
20      return (0.19 * getPrice());
21  }
22
23  public java.util.Currency getVat2() {
24      return tax.utils.VAT.computeVAT(this);
25  }
26
27 }
```

Outline

- Product
  - name : String
  - getName() : Stri
  - setName(String
  - price : Currency
  - getPrice() : Curr
  - setPrice(Curren
  - getVat() : Curre
  - getVat2() : Curr

Problems | Console | Error Log | Progress | Console

IMP Runtime

```
generating item allocations
computing priority and associativity filter
computing lookahead sets
optimizing lookahead automaton
printing the source code of the parser class
```

Writable | Smart Insert | 1 : 1 | 342M of 475M

```
10 syntax Field
11     = derived: Type Ident name  "=" Expression
12     | annotated: Annotation Type Ident name
13     ;
14
15 syntax Annotation
16     = @category="MetaVariable" host: "@host" "(" Str arg ")"
17     ;
18
19 syntax Expression
20     = const: Value value
21     | field: Ident var
22     | bracket Bracket: "(" Expression exp ")"
23     | neg: "-" Expression arg
24     >
25     left (
26         mul: Expression lhs "*" Expression rhs
27         | div: Expression lhs "/" Expression rhs
28     )
29     >
30     left (
31         add: Expression lhs "+" Expression rhs
```

Rascal – lwc11/src/lang/derived/compile/Derived2Java.rsc – Eclipse Platform

Entities.rsc | Entities.rsc | person.package | car.package | Packages.rsc | Packages.rsc | Package2Java.rsc | product.derived | Product.java

```
public str field2java(derived(t, n, exp)) {
    return getter(t, n, exp2java(exp));
}

public str field2java(annotated(host(a), t, n)) {
    method = substring(a, 1, size(a) - 1);
    return getter(t, n, "<method>(this)");
}

private str getter(Type t, str n, value exp) {
    <tn, cn> = <type2java(t), capitalize(n)>;
    return "public <tn> get<cn>() {
           '    return <exp>;
           '}";
}

public str exp2java(const(integer(n)))          = "<n>";
public str exp2java(const(float(f)))            = "<f>";
public str exp2java(const(boolean(b)))          = "<b>";
public str exp2java(const(string(s)))           = replaceAll(s, "\n", "\\
public str exp2java(const(date(d, m, y)))       = "new java.util.Date(<y>
public str exp2java(Expression::field(n))       = "get<capitalize(n)>()";
```

358M of 475M

Summary

- 4 languages
- 1+4 IDEs (Rascal's + Dynamically installed)
- 3 checkers
- 3 code generators to Java
  - 1 SQL code generator
  - 2 XML code generators
- Total lines of code: 950 LOC