Domain-Specific Languages
The State of the Art in 2014

Markus Völter

&

Bernd Kolb, Dan Ratiu, Zaur Molotnikov,
Domenik Pavletic, Kolja Dumman,
Sascha Lisson, Tamas Szabo, Niko Stotz

voelter { ingenieurbüro für softwaretechnologie // itemis

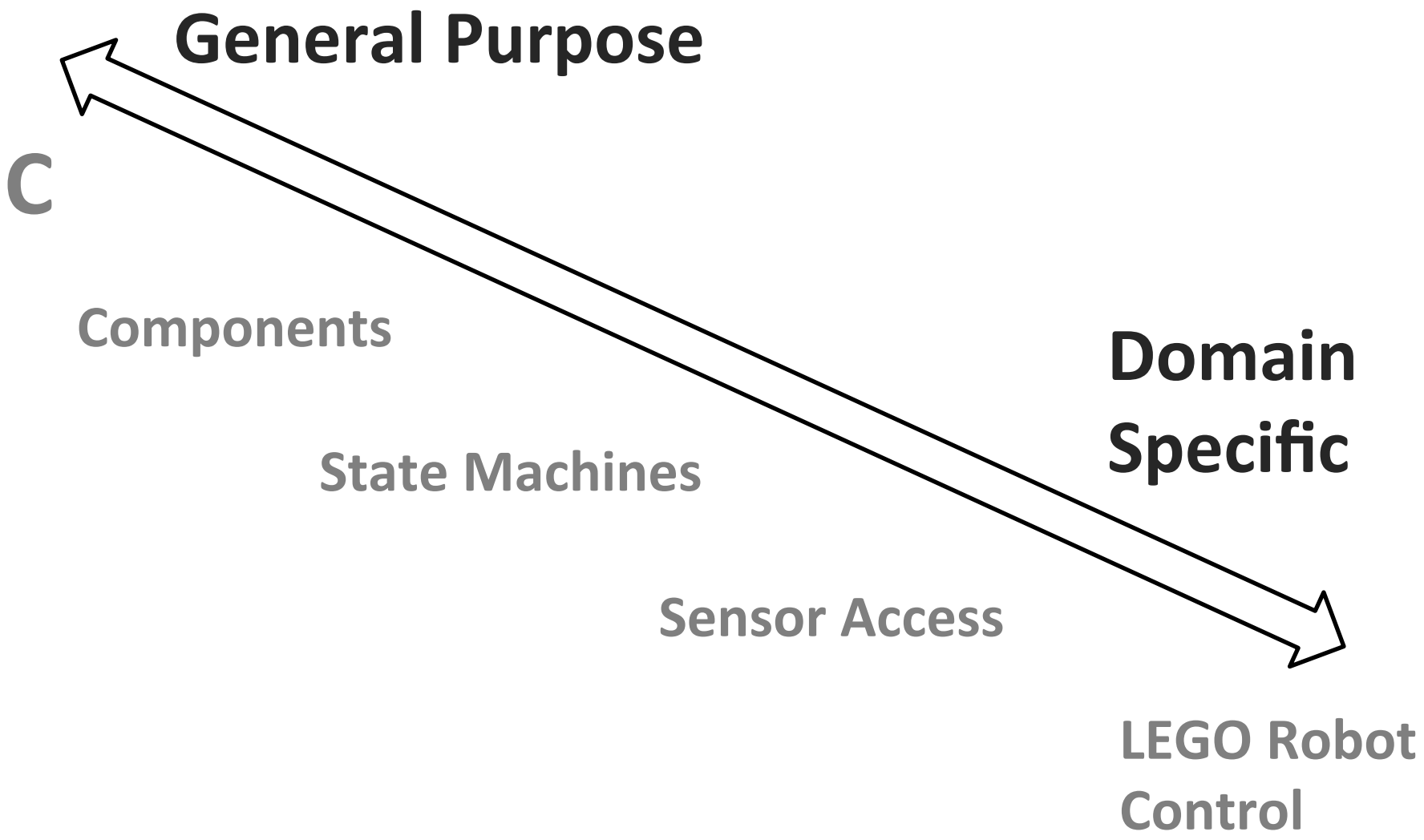
voelter@acm.org
www.voelter.de
@markusvoelter

1 Languages, Models, Programs
2 Language Workbenches
3 JetBrains MPS
4 mbeddr Example
5 Other Examples
6 Summary

1

Languages, Models, Programs

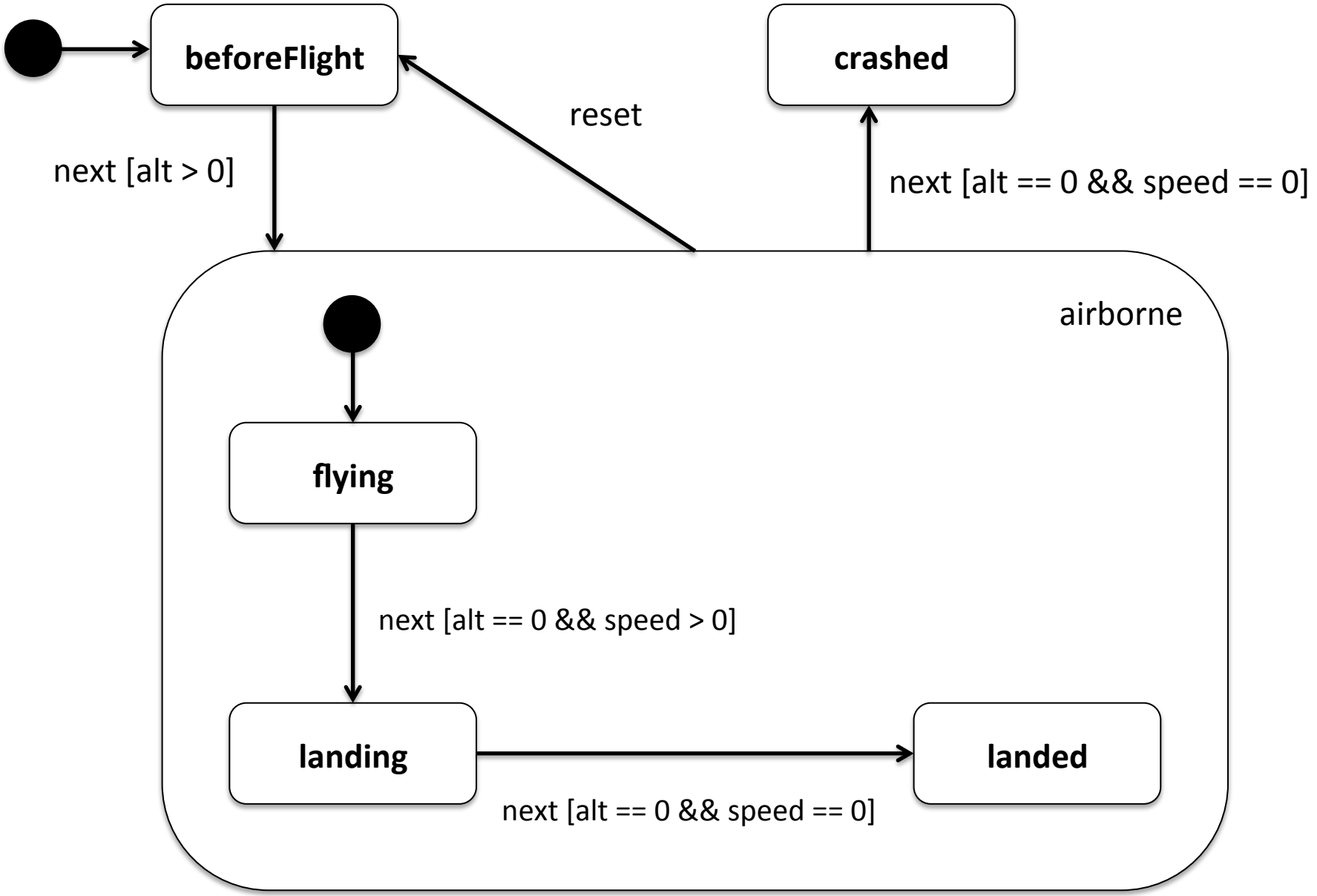
	more in GPLs	more in DSL
Domain Size	large and complex	smaller and well-defined
Designed by	guru or committee	a few engineers and domain experts
Language Size	large	small
Turing-completeness	almost always	often not
User Community	large, anonymous and widespread	small, accessible and local
In-language abstraction	sophisticated	limited
Lifespan	years to decades	months to years (driven by context)
Evolution	slow, often standardized	fast-paced
Incompatible Changes	almost impossible	feasible


General Purpose
C
Components
State Machines
Sensor Access
Domain
Specific
LEGO Robot
Control

Model or Code?

```
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;
}
```

Model or Code?



Model or Code?

```
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 {
  }
}
```

Model or Code?

```
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(); }
  }
}
```

Model or Code?

Does it really matter?

What is the difference?

Who cares?

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

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**

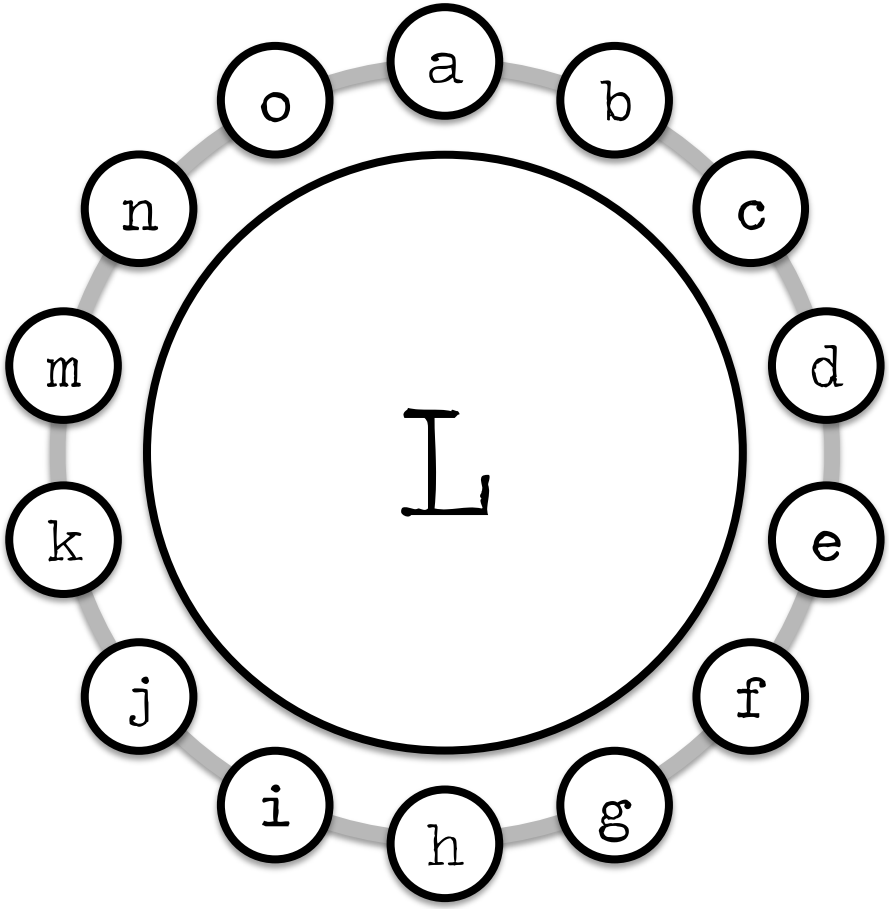
… with suitable **notations**

… **with suitable expressiveness**

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

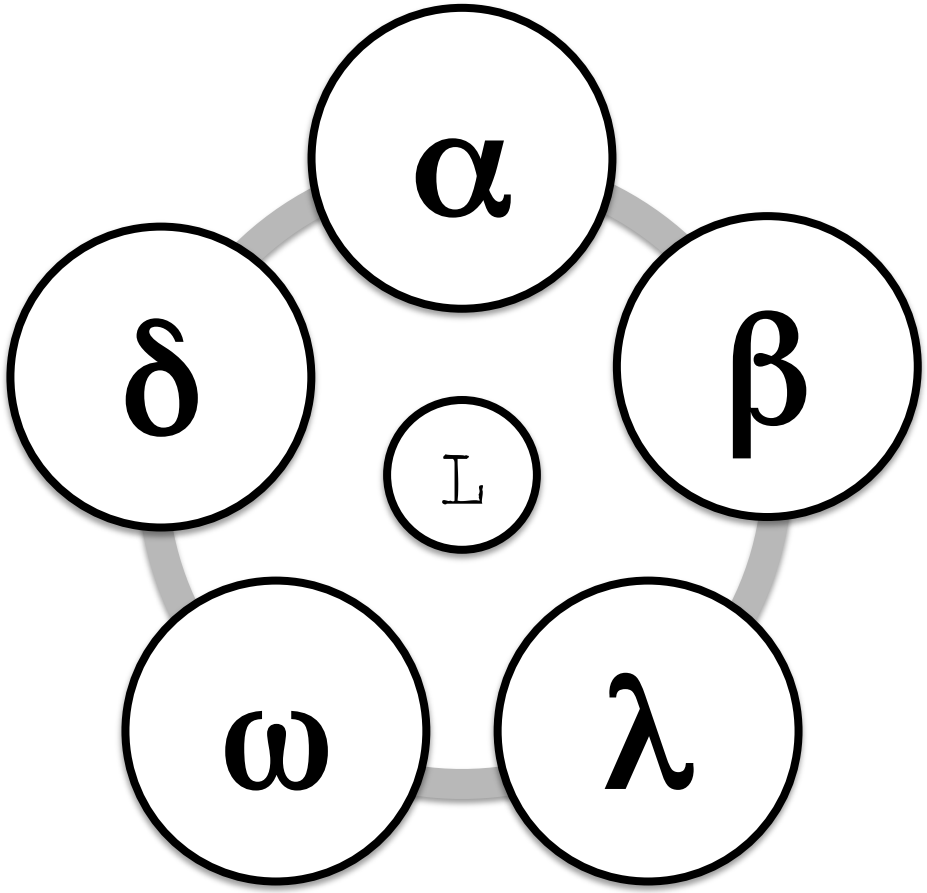
and always:
precise and tool processable

Big Language



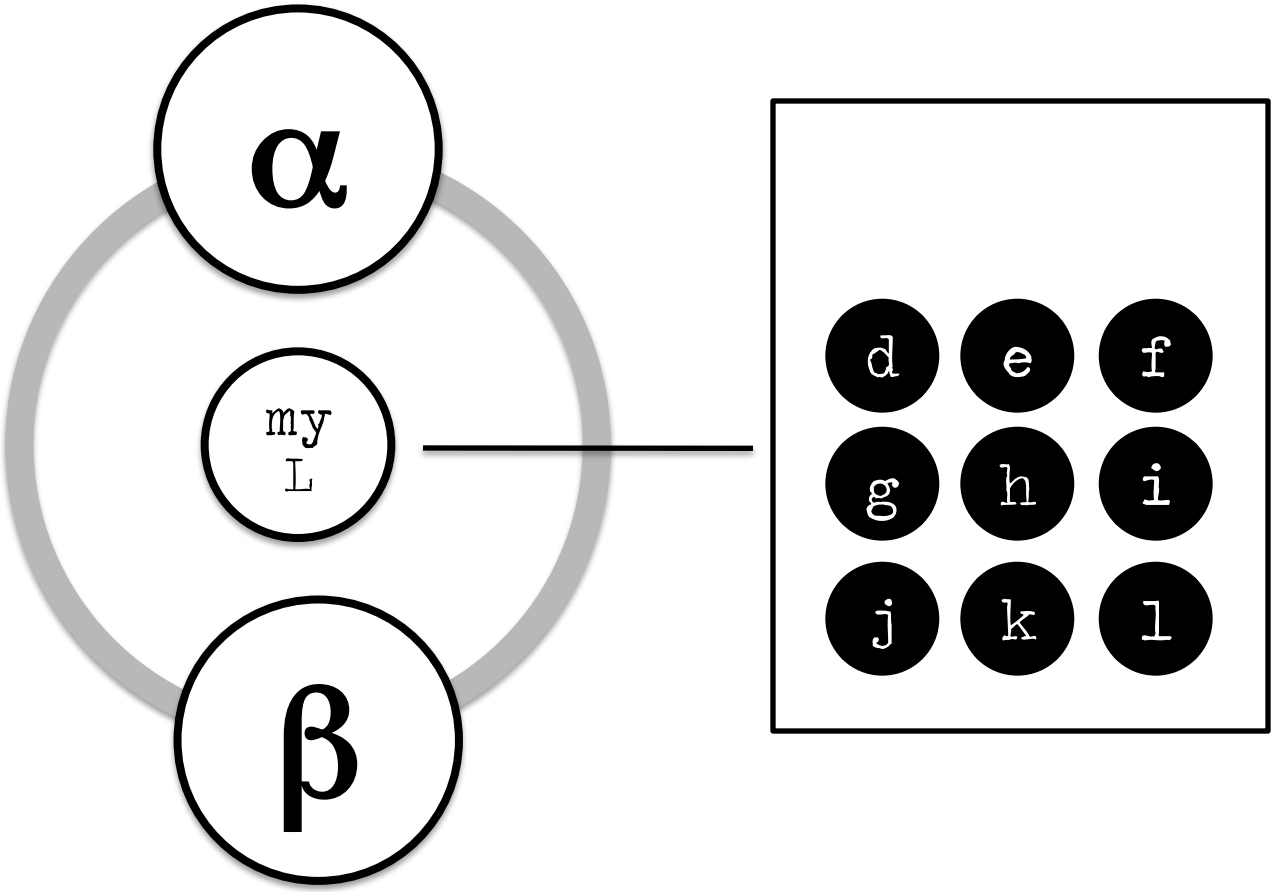
with **many** first class concepts!

Small Language



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

Modular Language



with many optional,
composable modules

2

Language Workbenches

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

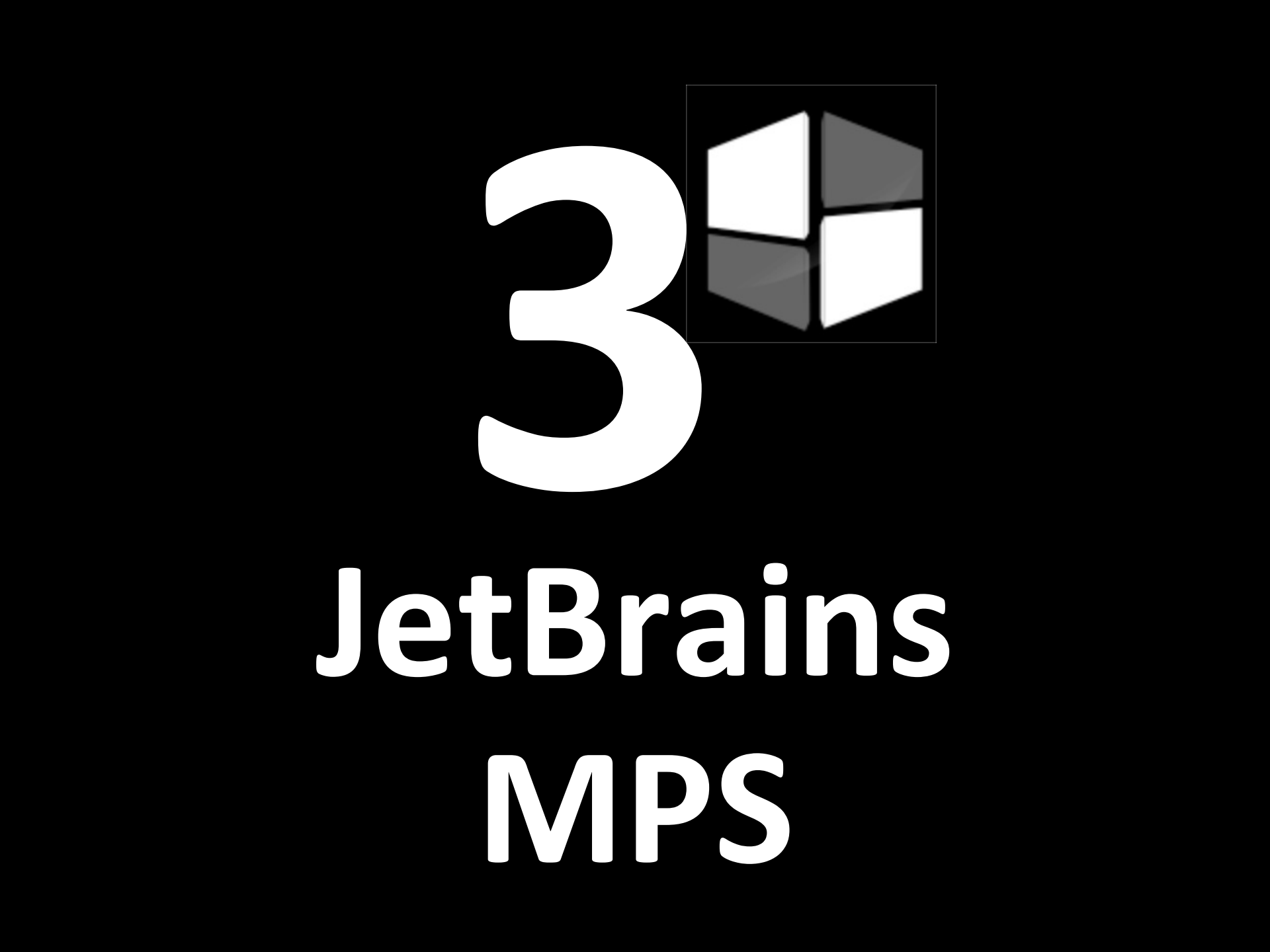
Language Workbench

(Martin Fowler)

support for „classical“ programming

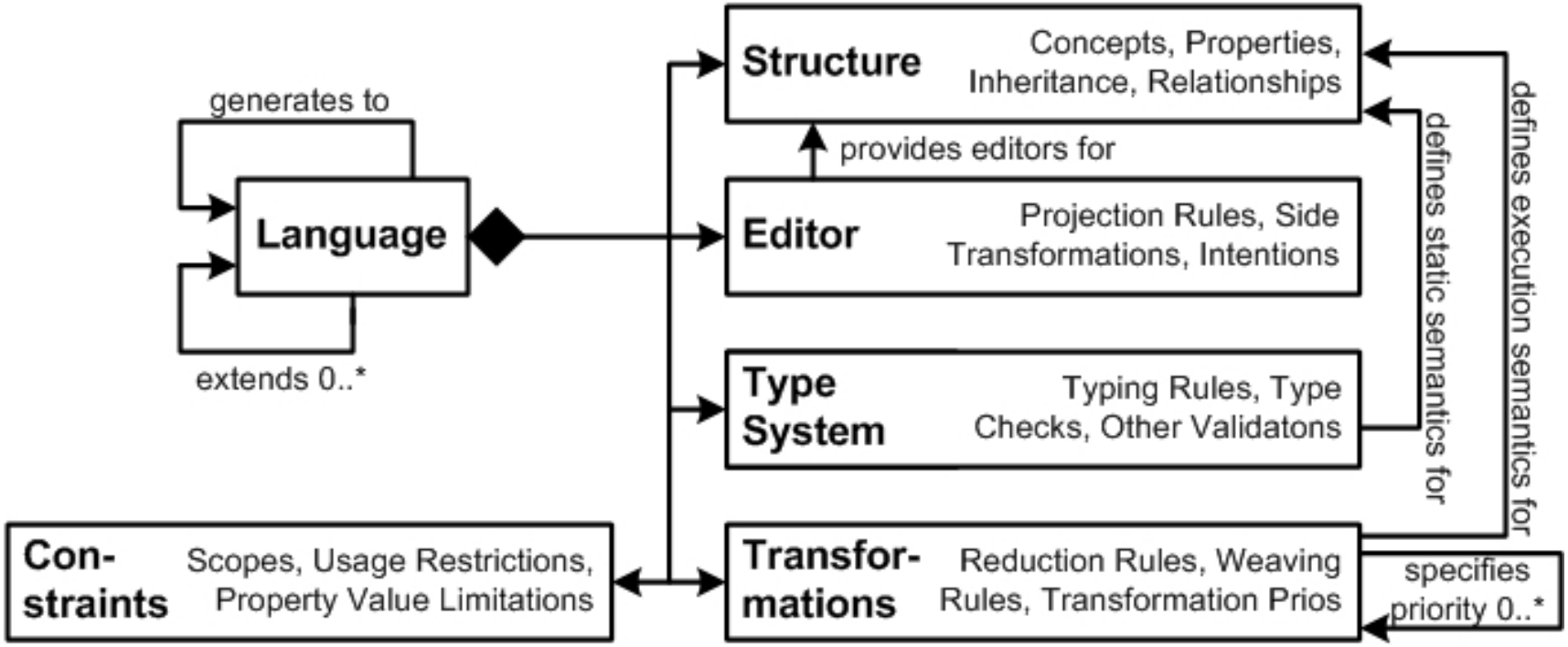
+

„classical“ and modeling


3
JetBrains
MPS

Open Source
Apache 2.0
http://jetbrains.com/mps

[Language Workbench]

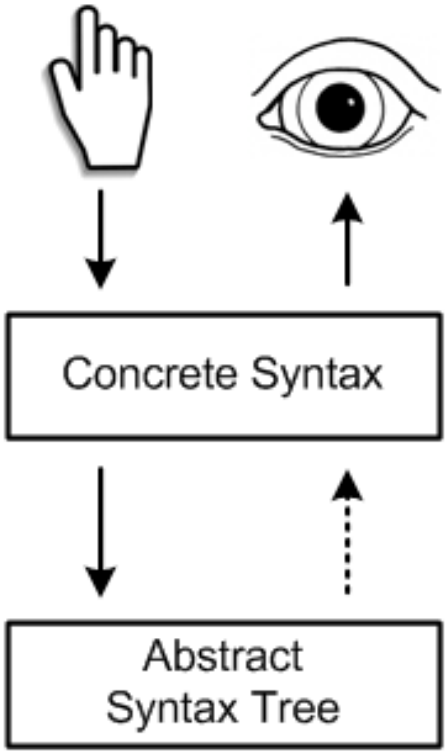


+ Refactorings, Find Usages, Syntax Coloring, Debugging, ...

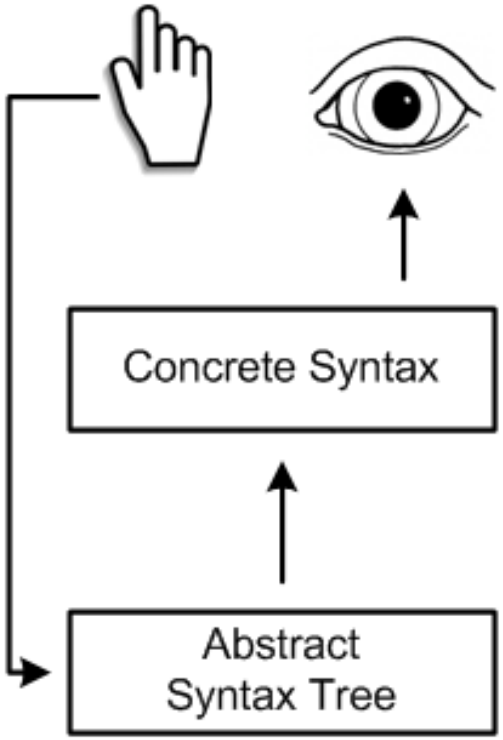
Projectional Editing

[Projectional Editing]

Parsing



Projectional Editing



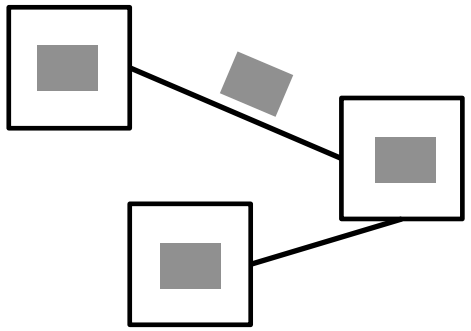
[Projectional Editing]
Syntactic Flexibility

Regular Code/Text

Mathematical

Tables

Graphical



[Projectional Editing]
Syntactic Flexibility

Regular Code/Text

```
//[ 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];
  } for
} aSummingFunction (function)
```

Mathematical

```
double midnight2(int32 a, int32 b, int32 c) {
  return (-b + √(b² - Σ_{i=1}^{4} a * c)) / (2 * a);
} midnight2 (function)
```

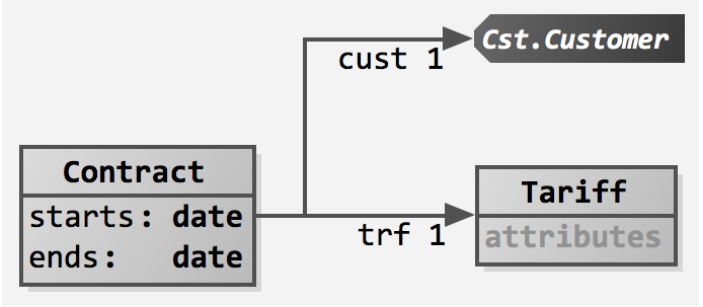
Tables

```
int16 decide(int8 spd, int8 alt) {
  return
```

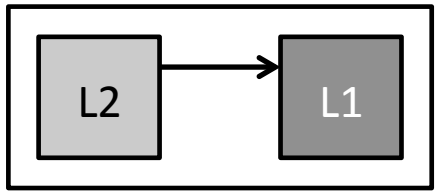
	spd > 0	spd > 100
alt < 0	1	1
alt == 0	10	20
alt > 0	30	40
alt > 100	50	60

```
} decide (function)
```

Graphical



[Projectional Editing]
Language Composition



Separate Files

Type System
Transformation
Constraints

In One File

Type System
Transformation
Constraints
Syntax
IDE

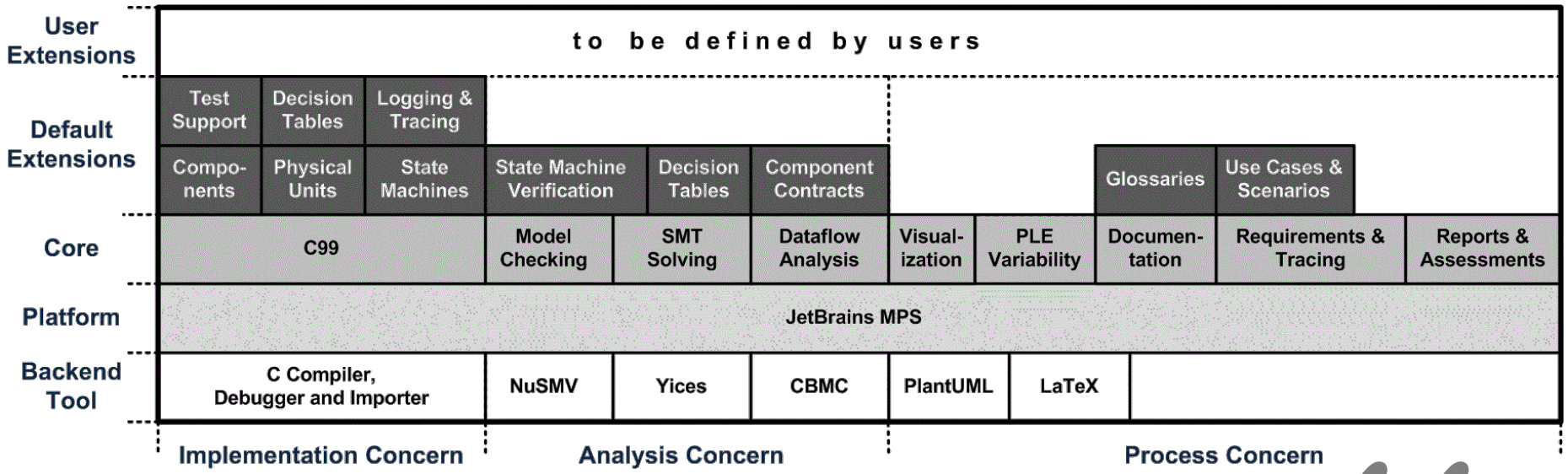
mbeddr

50+ extensions to C
10+ extensions to requirements lang.

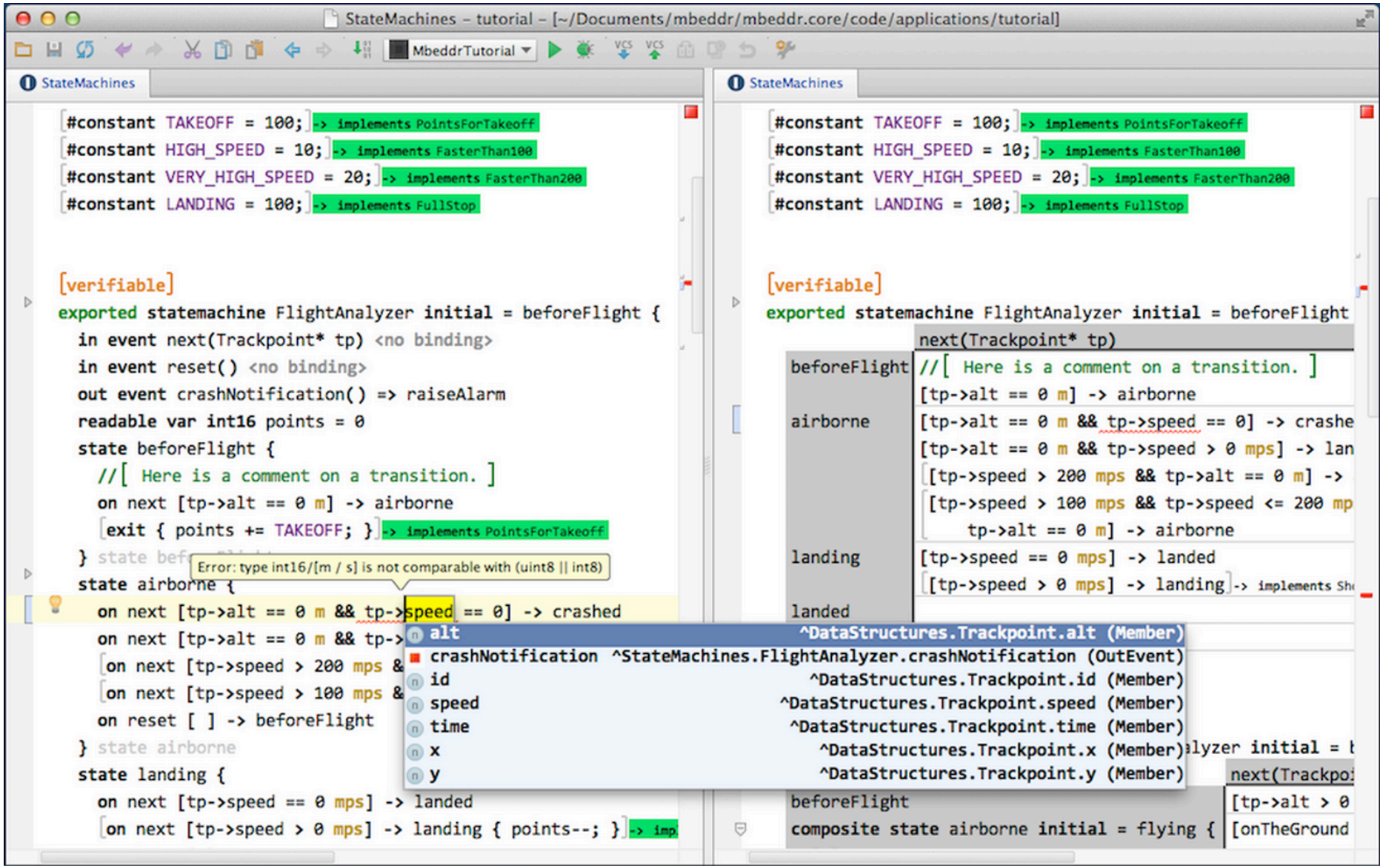

4
mbeddr

mbeddr

An extensible set of integrated languages for embedded software engineering.



“Specific Languages”


mbeddr
StateMachines – tutorial – [~/Documents/mbeddr/mbeddr.core/code/applications/tutorial]
MbeddrTutorial
StateMachines
#constant TAKEOFF = 100; -> implements PointsForTakeoff
#constant HIGH_SPEED = 10; -> implements FasterThan100
#constant VERY_HIGH_SPEED = 20; -> implements FasterThan200
#constant LANDING = 100; -> implements FullStop
[verifiable]
exported statemachine FlightAnalyzer initial = beforeFlight {
in event next(Trackpoint* tp) <no binding>
in event reset() <no binding>
out event crashNotification() => raiseAlarm
readable var int16 points = 0
state beforeFlight {
//[Here is a comment on a transition.]
on next [tp->alt == 0 m] -> airborne
exit { points += TAKEOFF; } -> implements PointsForTakeoff
} state bef
Error: type int16/[m / s] is not comparable with (uint8 || int8)
state airborne {
on next [tp->alt == 0 m && tp->speed == 0] -> crashed
on next [tp->alt == 0 m && tp->
on next [tp->speed > 200 mps &
on next [tp->speed > 100 mps &
on reset [ ] -> beforeFlight
} state airborne
state landing {
on next [tp->speed == 0 mps] -> landed
on next [tp->speed > 0 mps] -> landing { points--; } -> imp
alt ^DataStructures.Trackpoint.alt (Member)
crashNotification ^StateMachines.FlightAnalyzer.crashNotification (OutEvent)
id ^DataStructures.Trackpoint.id (Member)
speed ^DataStructures.Trackpoint.speed (Member)
time ^DataStructures.Trackpoint.time (Member)
x ^DataStructures.Trackpoint.x (Member)
y ^DataStructures.Trackpoint.y (Member)
StateMachines
next(Trackpoint* tp)
beforeFlight
//[Here is a comment on a transition.]
[tp->alt == 0 m] -> airborne
airborne
[tp->alt == 0 m && tp->speed == 0] -> crashe
[tp->alt == 0 m && tp->speed > 0 mps] -> lan
[tp->speed > 200 mps && tp->alt == 0 m] ->
[tp->speed > 100 mps && tp->speed <= 200 mp
tp->alt == 0 m] -> airborne
landing
[tp->speed == 0 mps] -> landed
[tp->speed > 0 mps] -> landing -> implements Sh
landed
beforeFlight
composite state airborne initial = flying {
[tp->alt > 0
[onTheGround

fortiss itemis

BMW Car IT

Bundesministerium
für Bildung
und Forschung

Research Project 2011 - 2013

Open Source @ eclipse.org
Eclipse Public License 1.0
http://mbeddr.com

Commercial Use and Extension

itemis
SIEMENS

Research Platform

itemis France: Smart Meter

First significant mbeddr project
ca. 100,000 LoC
about to be finished
great modularity due to components
uses physical units extensively
great test coverage due to special extensions

ACCEnT Control.Lab

LMS INTERNATIONAL
Researchpark Haasrode 1237 | Interleuvenlaan 68 | B-3001 Leuven [Belgium]
T +32 16 384 200 | F +32 16 384 350 | info@lmsintl.com | www.lmsintl.com

Worldwide
For the address of your local representative, please visit www.lmsintl.com/lmsworldwide

LMS is a leading provider of test and mechatronic simulation software and engineering services in the automotive, aerospace and other advanced manufacturing industries. As a business segment within Siemens PLM Software, LMS provides a unique portfolio of products and services for manufacturing companies to manage the complexities of tomorrow's product development by incorporating model-based mechatronic simulation and advanced testing in the product development process. LMS tunes into mission-critical engineering attributes, ranging from system dynamics, structural integrity and sound quality to durability, safety and power consumption. With multi-domain and mechatronic simulation solutions, LMS addresses the complex engineering challenges associated with intelligent system design and model-based systems engineering. Thanks to its technology and more than 1250 dedicated people, LMS has become the partner of choice of more than 5000 manufacturing companies worldwide. LMS operates in more than 30 key locations around the world.

Siemens PLM Software

SIEMENS

LMS®

A Siemens Business

20+ Projects in various stages

by various “Big Name” companies.

Branching out into the Domains

Finance, Insurance, Requirements

DEMO

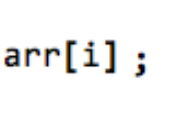
[Embedded Software]

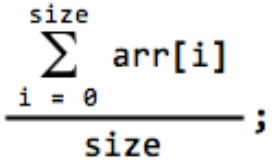
mbeddr

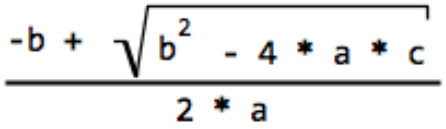
```
struct Trackpoint {
  int8 id;                // sequence ID of the trackpoint
  int8/s/   timestamp; // timestamp as taken from GPS time
  int8/m/   x;            // longtitude, simplified as a number
  int8/m/   y;            // latitude, simplified as a number
  int8/m/   alt;          // altitude as of the GPS
  int8/mps/ speed;        // current speed, if available
};

                        -1
derived unit mps = m s    for velocity
```

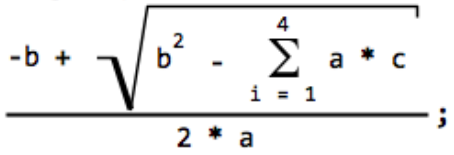
[Embedded Software]

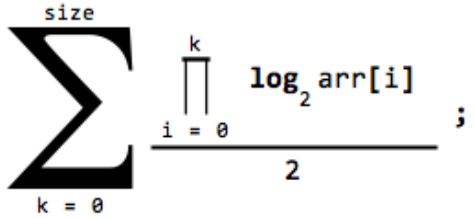
mbeddr

```
int32 sumUpIntArray(int32[] arr, int32 size) {
  return  ;
} sumUpIntArray (function)
```

```
int32 averageIntArray(int32[] arr, int32 size) {
  return  ;
} averageIntArray (function)
```

```
double midnight1(int32 a, int32 b, int32 c) {
  return  ;
} midnight1 (function)
```

```
double midnight2(int32 a, int32 b, int32 c) {
  return  ;
} midnight2 (function)
```

```
double sumOfProductsOfLogs(int32[] arr, int32 size) {
  return  ;
} sumOfProductsOfLogs (function)
```

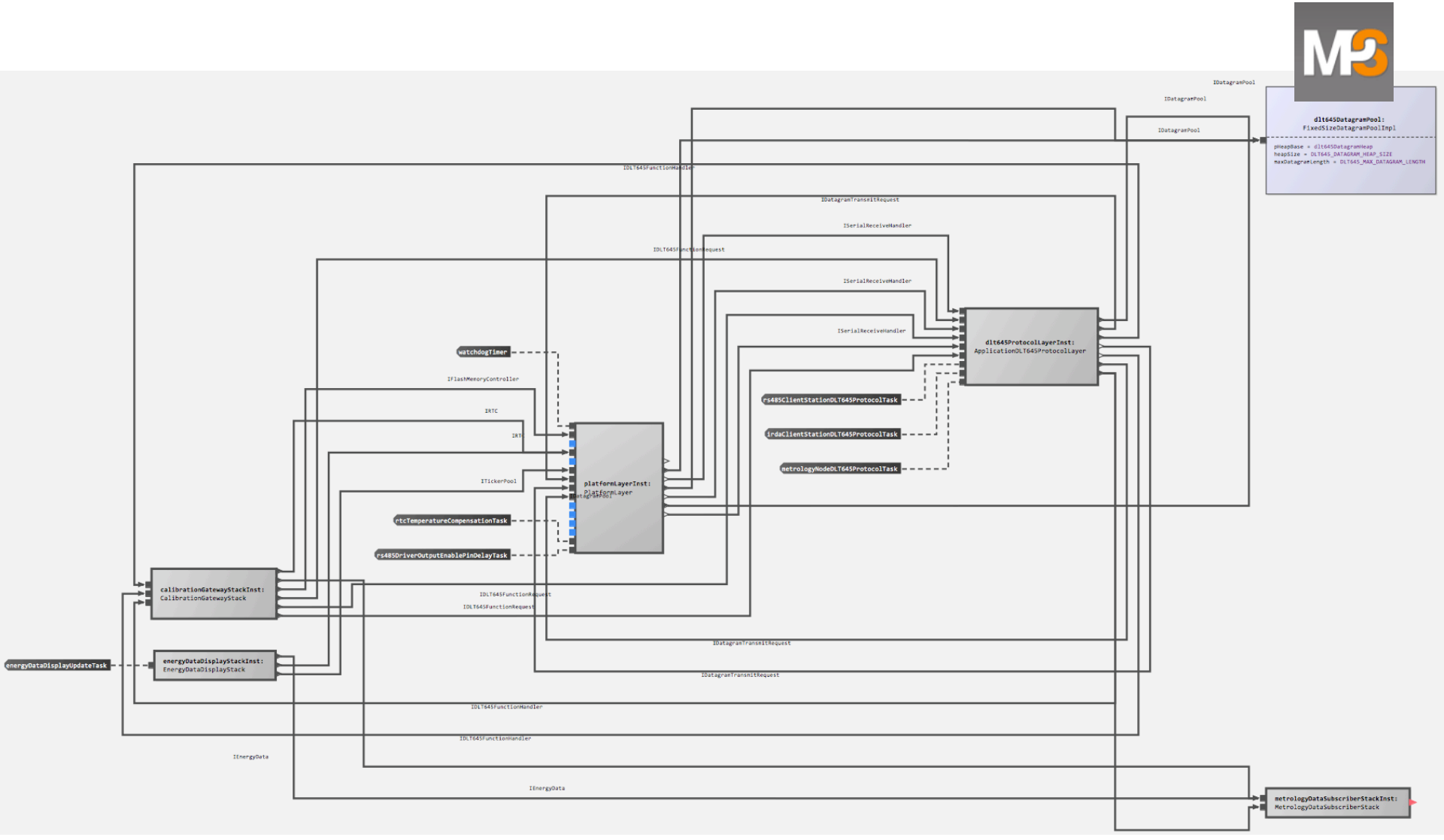
[Embedded Software]

mbeddr

```
exported cs interface TrackpointStore1 {
  void store(Trackpoint* tp)
    pre(0) isEmpty()
    pre(1) tp != null
    post(2) !isEmpty()
    post(3) size() == old(size()) + 1
  Trackpoint* get()
    pre(0) !isEmpty()
  Trackpoint* take()
    pre(0) !isEmpty()
    post(1) result != null
    post(2) isEmpty()
    post(3) size() == old(size()) - 1
  query int8 size()
  query boolean isEmpty()
}
```

[Embedded Software]

mbeddr



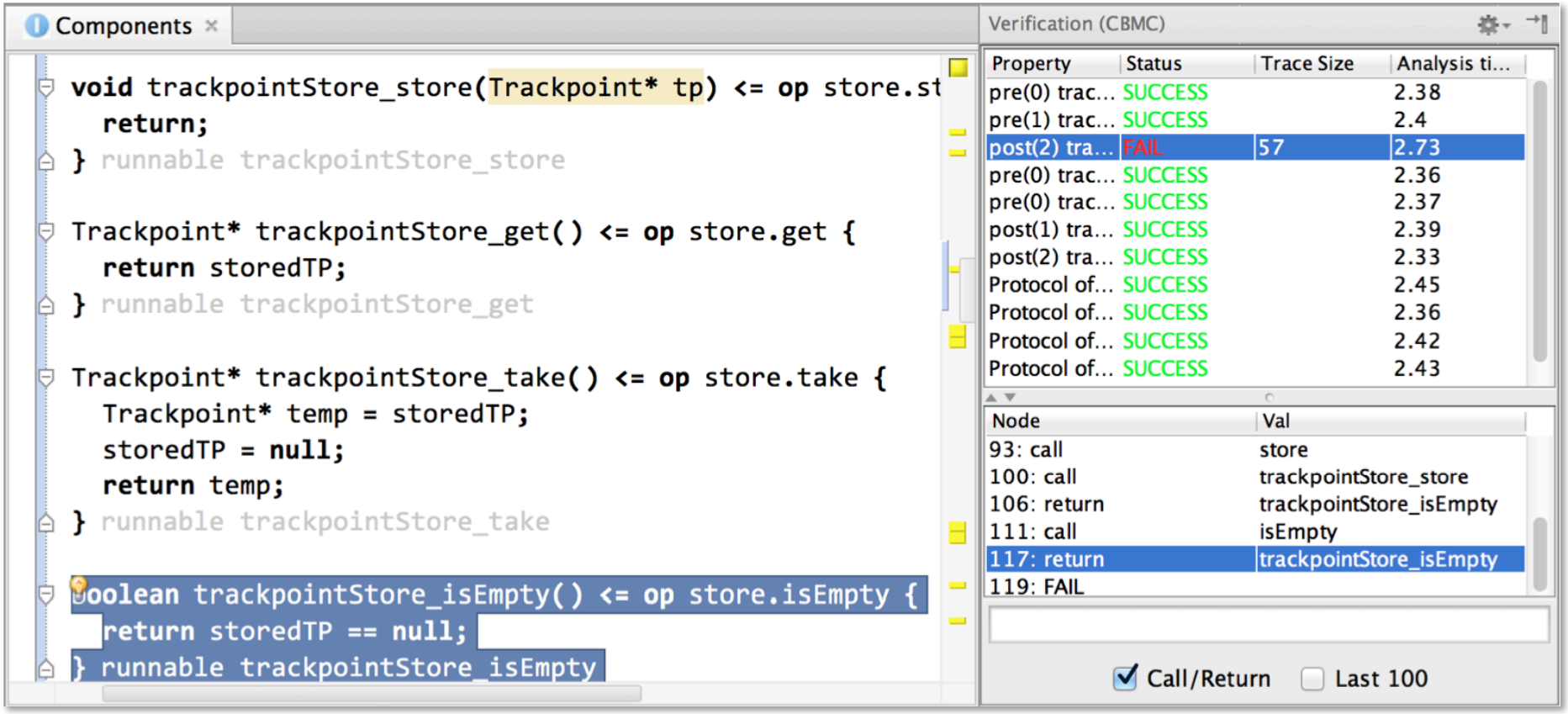
[Embedded Software]

mbeddr

```
mock component StorageMock report messages: true {
  provides TrackpointStore1 store
  Trackpoint* lastTP;
  total no. of calls is 5
  sequence {
    step 0: store.isEmpty return true;
    step 1: store.store {
        assert 0: parameter tp: tp != null
      }
      do { lastTP = tp; }
    step 2: store.isEmpty return false;
    step 3: store.take return lastTP;
    step 4: store.store
  }
}
```

[Embedded Software]

mbeddr



[Embedded Software]

mbeddr

```
exported component Judge extends nothing {
  provides FlightJudger judger
  int16 points = 0;
  void judger_reset() <= op judger.reset {
    points = 0;
  } runnable judger_reset
  void judger_addTrackpoint(Trackpoint* tp) <= op judger.addTrackpoint {
    points += 0
  } runnable judger_addTrackpoint
  int16 judger_getResult() <= op judger.getResult {
    return points;
  } runnable judger_getResult
} component Judger
```

	tp->alt <= 2000 m	tp->alt >= 2000 m
tp->speed < 150 mps	0	10
tp->speed >= 150 mps	5	20

[Embedded Software]

mbeddr

```
statemachine FlightAnalyzer initial = beforeFlight {
  ...
  state crashed {
    entry { raiseAlarm(); }
  }
}
...
void raiseAlarm() {}
```

```
statemachine FlightAnalyzer initial = beforeFlight {
  out crashNotification() => raiseAlarm
  ...
  state crashed {
    entry { send crashNotification(); }
  }
}
```

[Embedded Software]

mbeddr

```
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]
       -> flying { points += VERY_HIGH_SPEED; }
    on next [tp->speed > 100 mps]
       -> flying { points += HIGH_SPEED; }
  }
  state landing {
    on next [tp->speed == 0 mps] -> landed
    on next [ ] -> landing { points--; }
  }
  state landed {
    entry { points += LANDING; }
  }
}
```

[Embedded Software]

mbeddr

```
[checked]
exported statemachine FlightAnalyzer initial = beforeFlight {
```

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

```
}
```

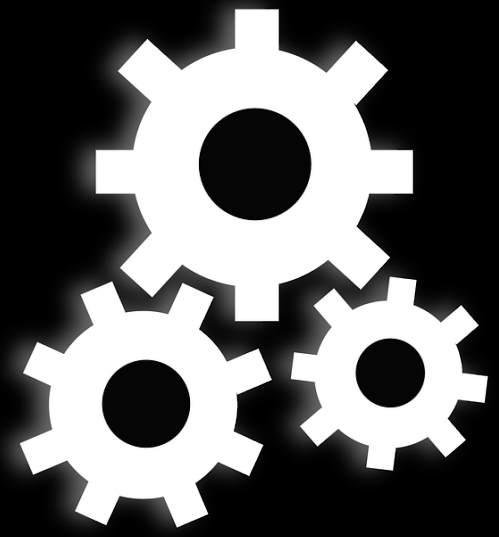
[Embedded Software]

mbeddr

```
section 1.2 existing.comps: Interfaces and Components {

  [ Interfaces declare operations that can be provided or used by components.
  Each operation can also declare pre- and postconditions as well as
  protocols. These can be checked either at runtime or statically. The
  @cm(Components) module contains examples. Below is an interface:

  embed as text Components.TrackpointStore1/

  [ The interfaces, components and thei relationships in a given module can
  also be rendered graphically. An example is shown in @fig(ci)

  ---------------------------------------------------------------------------
  visualize Components.store.TrackpointStore1/
      components + interfaces (grouped) as ci
    location: vis:/
    scaling:  width100
  [The components and their provided (solid lines) and required (dotted lines)
  ports.
  ---------------------------------------------------------------------------

  [ Of course the visualizations are also not just images. In the source to the
  document, we embed references to \code(IVisualizable) instances. In the
  doc, one can select the visualization category, and then, during
  generation, PlantUML automatically rerenders the image.

}
```

5

Other Examples



Insurance Configuration

[Business Application]

Insurance Configuration Tool

Life Product

Example product

1 Overview

1.1 Description

> Lorem ipsum dolor sit amet, consectetur adipiscin
> potenti. Etiam risus ante, bibendum ut mattis ege
> velit. Quisque venenatis faucibus tellus consequa
> quam eu dui dictum sollicitudin.
> Duis tempus justo magna. Nunc lobortis libero sed
> non sagittis sed, vulputate quis nunc. Integer sc
> eros faucibus congue scelerisque, sapien sapien p
> ultricies viverra mauris. Pellentesque pretium du
> sit amet consectetur augue. Aliquam nibh arcu, eg
> lectus a lacus sollicitudin pellentesque et sed m

1.2 Selling Period and Holder

This product can be sold from 9 / 9 / 9 until 9 / 9 / 9
The holder of the product can be a Person
Specifying the beneficiary is optional

1.3 Covers

This product includes the following covers
- Financial cover

[Business Application]

Insurance Configuration Tool

Life Product

English	Example product
Portuguese	Portuguese product text

1 Overview

1.1 Description

Lorem ipsum dolor sit amet, consectetur adipiscing
potenti. Etiam risus ante, bibendum ut mattis eget
velit. Quisque venenatis faucibus tellus consequat
quam eu dui dictum sollicitudin.
Duis tempus justo magna. Nunc lobortis libero sed
non sagittis sed, vulputate quis nunc. Integer sol
eros faucibus congue scelerisque, sapien sapien pl
ultricies viverra mauris. Pellentesque pretium dui
sit amet consectetur augue. Aliquam nibh arcu, ege
lectus a lacus sollicitudin pellentesque et sed me

1.2 Selling Period and Holder

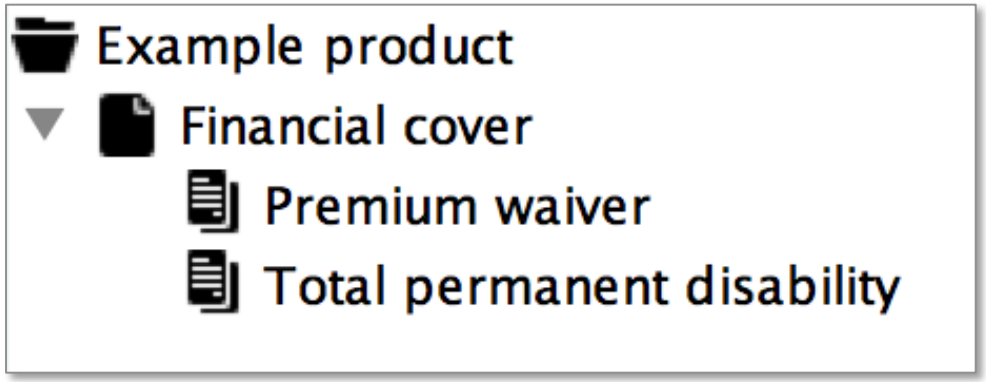
This product can be sold from 9 / 9 / 9 until 9 / 9 / 9
The holder of the product can be a Person
Specifying the beneficiary is optional

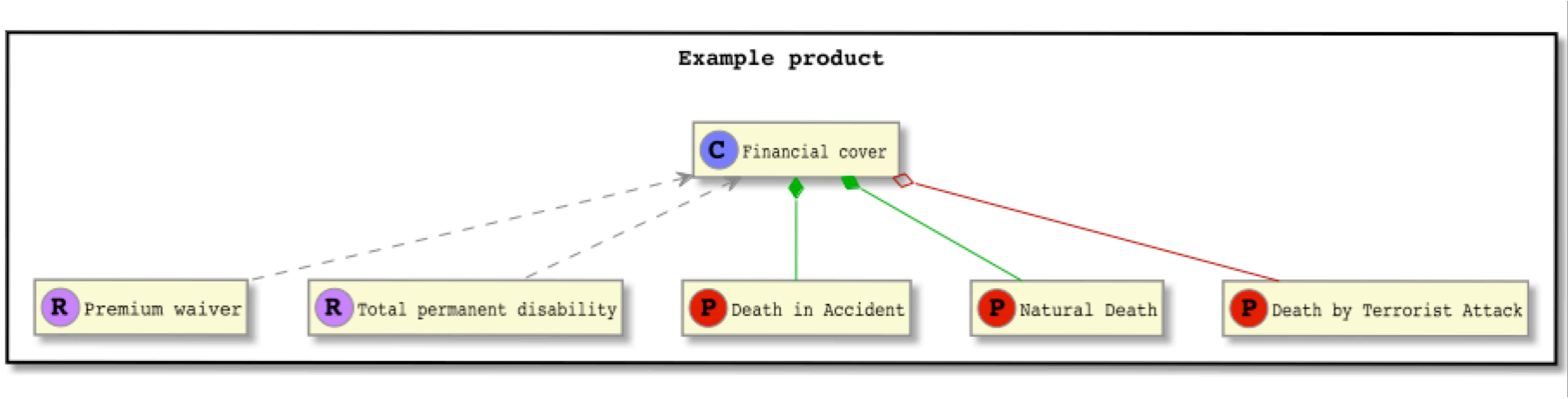
1.3 Covers

This product includes the following covers
- Financial cover

[Business Application]

Insurance Configuration Tool





[Business Application]

Insurance Configuration Tool

[ ]

Formula Library FormulaLibrary01 for Rule Set Type IEEE_RST

[ ]

numeric formulaOne = 3 < 4

$$12 + 7 + CATV - \sum_{in=1}^{6} \left(in + 9 + \sum_{out=1}^{in} (in + out) \right)$$

Testsuite

Show Test Suite ☐

[ ]

numeric dummy = 3 > 2

$$\sum_{index=1}^{6} (index + 1)$$

Testsuite

Show Test Suite ☐

[Business Application]

Insurance Configuration Tool

[ ]

Formula Library FormulaLibrary01 for Rule Set Type IEEE_RST

[ ]

numeric formulaOne = 3 < 4 | $12 + 7 + CATV - \sum_{in=1}^{6} \left(in + 9 + \sum_{out=1}^{in} (in + out) \right)$

Testsuite

Show Test Suite ☑ Run All Clear Evaluation

	Input Variables	Output Variables	
<no name>	**CATV**		**result**
	0	Expected	9
		Result	0

Run Run and Debug

<<Test Case>>

[ ]

numeric dummy = 3 > 2 | $\sum_{index=1}^{6} (index + 1)$

Testsuite

Show Test Suite ☑ Run All Clear Evaluation

	Input Variables	Output Variables	
test			**result**
		Expected	0
		Result	0

Run Run and Debug

<<Test Case>>

[Business Application]

Insurance Configuration Tool

Rule Set Type DemoRuleSetType

Business objects

person : **Person**

Variables:

PRMI	: **int**
FR	: **int**
NN	: **int**
TT	: **int**
J	: **int**
A3	: **int**
G3	: **int**
ANUI	: **int**
X	: **int**

Parent

<no parent>

Libraries

Standard

Extra

[Business Application]

Insurance Configuration Tool

Rule Set Type DemoRuleSetType

Business objects

person : **Person**

Variables:

PRMI : **int**
FR : **int**
NN : **int**
TT : **int**
J : **int**
A3 : **int**
G3 : **int**
ANUI : **int**
X : **int**

Parent

<no parent>

Libraries

Standard
Extra

Rule Set Type DemoRuleSetType

Business objects

<no business objects>

Variables:

<no variables>

Parent

<no parent>

Libraries

<no libraries>

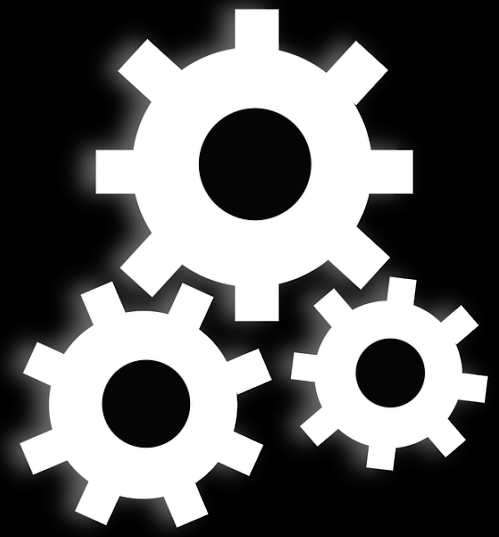
[Business Application]

Insurance Configuration Tool

```
rule set DemoRulseSet2 is of type DemoRuleSetType

EU0    : int                    [ save false print false ]
CATEG  : string                 [ save false print false ]
CATEG1 : double                 [ save true  print true  ]
```

Toggle Information

```
PREMIO = [ A1 > 10   => EU0
           <always>  => FLAG ]

FLAG   = [ CATEG1 equals 60 or CATEG1 equals 63 or CATEG1 equals 64 => 160
           PREMIO equals 0                                          => 162
           CATEG1 > 0 or substr(inga[4], 1, 1) equals "V"           => 163
           <always>                                                 => PREMIO + FLAG ]

PREMIO = [ <always> => round(PREMIO * (1 + factacer), 0) ]
```

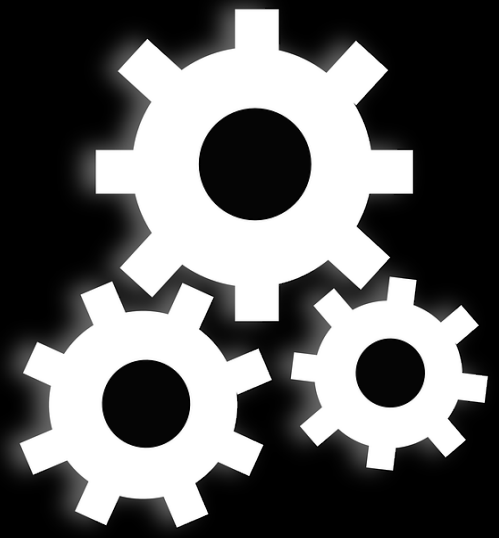


Tax/Benefits

[Tax/Benefits Application]

Dutch Tax Agency (Evaluation)

```
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
}
```



Telco Demo

[Business Application]

Telco Demo App

Data Contract

proxy for Customer.Customer

core data entity BillingRegion

code **[key]:**	**string**	references:
name:	**string**	
baseMinPrice:	**float**	
maxRebateFactor:	**float**	

entity Contract

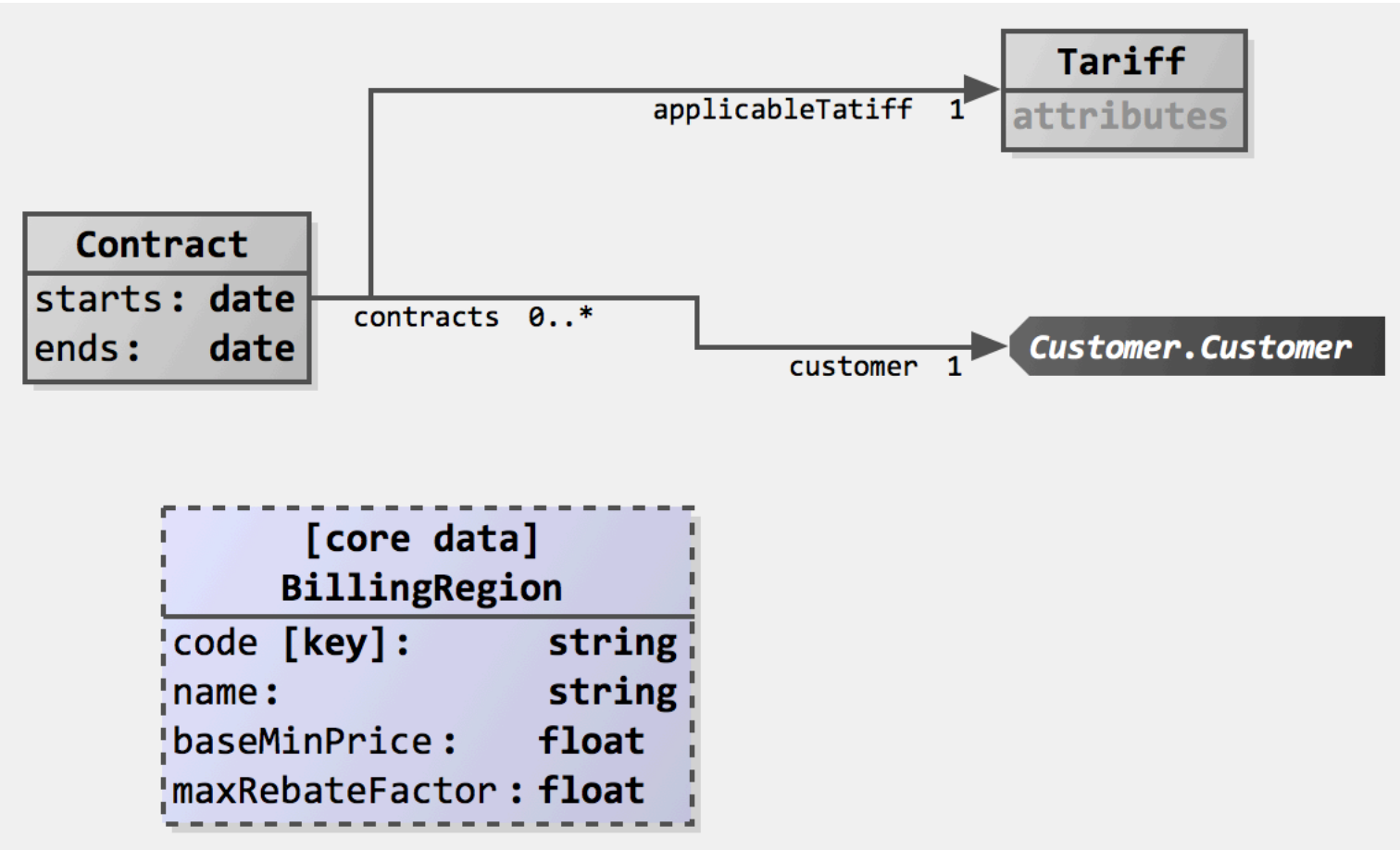
starts: **date**	customer:	***Customer***	1 ☐	<--> contracts 0..*
ends: **date**	applicableTatiff:	***Tariff***	1 ☐	

entity Tariff

attributes:	references:

[Business Application]

Telco Demo App



[Business Application]

Telco Demo App

Core Data DefaultRegions **for entity** BillingRegion

Code	Name	Base Min Price	Max Rebate Factor
BW	Baden Württemberg	0.20	0.8
BY	Bayern	0.20	0.8
BE	Berlin	0.15	0.7
BB	Brandenburg	0.10	0.7
HB	Bremen	0.20	0.7
HH	Hamburg	0.15	0.7
HE	Hessen	0.15	0.7
MV	Mecklenburg-Vorpommern	0.10	0.7
NI	Niedersachsen	0.15	0.7
NW	Nordrhein-Westfalen	0.15	0.7
RP	Rheinland-Pfalz	0.15	0.7
SL	Saarland	0.15	0.7
SN	Sachsen	0.10	0.7
ST	Sachsen-Anhalt	0.10	0.7
SH	Schleswig-Holstein	0.15	0.7
TH	Thüringen	0.10	0.7

[Business Application]

Telco Demo App

Calculations CustomerBasic **for** Customer **imports:** TimeUnits
BusinessRequirements

```
Node:       isRebated [FlagVar]
Kind:       implements
1st Target: Users should be rated
            [ Some users should get cheaper phone calls. The reasons for the rebates are outlined below. ]
```

```
flag isRebated :=  magic of type boolean[T]

[ A couple of statistics about the last month's activity ]
value callsLastMonth :=  entity.calls.where(!it.startTime.isOlderThan(30 day))
flag activeThisMonth :=  !callsLastMonth.isEmpty
value devicesUsedLastMonth :=  callsLastMonth.select(it.sourceDevice).distinct

value totalPriceLastMonth :=  ∑ (i = 0 .. callsLastMonth.size)  callsLastMonth.at(i).price.value

value averageCallPriceLastMonth :=  totalPriceLastMonth / callsLastMonth.size


[ Some random examples. ]
value example :=  all[Call].first.customer.calls.first.startTime
```

[Business Application]

Telco Demo App

Calculations CallCalculations **for** Call **imports:** Σ CustomerBasic

flag isLocal := **magic of type boolean**
flag isLongDistance := **magic of type boolean**
flag isRoaming := **magic of type boolean**

value cust := **entity**.customer
value pricingFactor :=

	isLocal	isLongDistance	isRoaming
cust.isRebated	0.5	0.6	0.8
!cust.isRebated	0.8	0.9	1.0

otherwise 1

Here is a comment added in the gutter, just as in MS Word.
22/09/14 08:19 (13 s ago) by markusvoelter

[Business Application]

Telco Demo App

```
contract BaseContract specializes <no baseContract>          imports: << ... >>
Context Objects:
  c: Customer

[final] assign callsThisMonth
callsThisMonth := c.callsLastMonth

assign amountThisMonth
amountThisMonth := 0

[final] store storeBill
c.bills :+ new MonthlyBill {
             amount := amountThisMonth
           }
```

[Business Application]

Telco Demo App

```
contract FlatrateContract specializes BaseContract          imports: R BusinessRequirements
Context Objects:
  c: Customer

conditional assign overrides BaseContract.amountThisMonth as of 16/8/2014[T]
amountThisMonth := | c.isRebated  | 40 |
                   | otherwise    | 50 |

conditional assign overrides BaseContract.amountThisMonth as of 20/8/2014[T]
amountThisMonth := | c.isRebated  | 40 |
                   | otherwise    | 60 |
```

[Business Application]

Telco Demo App

```
contract FlatrateContract specializes BaseContract          imports: BusinessRequirements
Context Objects:
  c: Customer

[final] assign BaseContract.callsThisMonth
callsThisMonth := c.callsLastMonth

[final] store BaseContract.storeBill
c.bills := new MonthlyBill {
             amount := amountThisMonth
           }
```

```
conditional assign overrides BaseContract.amountThisMonth as of 16/8/2014[T]
amountThisMonth := | c.isRebated  | 40 |
                   | otherwise    | 50 |

conditional assign overrides BaseContract.amountThisMonth as of 20/8/2014[T]
amountThisMonth := | c.isRebated  | 40 |
                   | otherwise    | 60 |
```

[Business Application]

Telco Demo App

```
rule checkStuff
given anything
when [the customer.calls.size is equal to 10] and
     [the call.endTime is smaller than 20    ]
then [set call.price to 20                   ]
     [execute cancelContract with customer   ]
```

[Business Application]

Telco Demo App

```
Assessment: UnusedCode
query:      unused code
sorted: ☒    must be ok: ☒    hide ok ones: ☐
last updated: Sep 18, 2014 (3 days ago) by markusvoelter
```

BaseContract

- ☒ storeBill

CustomerBasic

- ☐ example
- ☐ isMale
- ☐ activeThisMonth

FlatrateContract

- ☐ FlatrateContract.amountThisMonth
- ☐ FlatrateContract.amountThisMonth

total 11, new 0, ok 1

[Business Application]

Telco Demo App

1 | Initially you have no points.

InitialNoPoints /functional: tags

Add Comment | Add Other Data | Add Child Requirement | Add Next Requirement | ⇩

[When the game starts, you have no points.]

---

workpackage inital **scope:** 1 **responsible:** peter **prio:** 1 **effort:** 1 **days**

[ ]

2 | Once a flight lifts off, you get 100 points

PointsForTakeoff /functional: tags

Add Comment | Add Other Data | Add Child Requirement | Add Next Requirement | ⇧ | ⇨ | ⇩

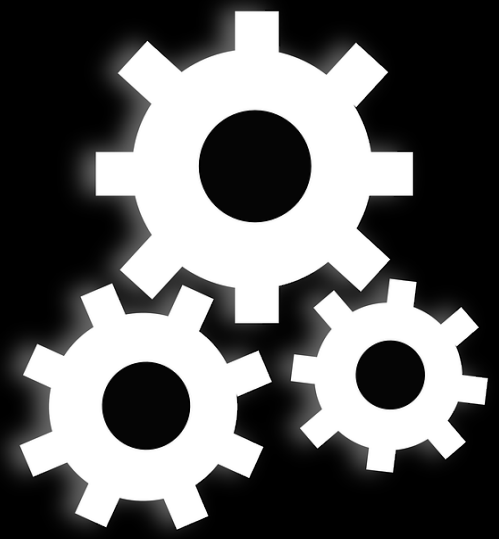
[Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent feugiat enim arcu, ut egestas velit. Suspendisse potenti. Etiam risus ante, bibendum ut mattis eget, convallis sit amet nunc. Ut nec justo sapien, vel condimentum velit. Quisque venenatis faucibus tellus consequat rhoncus.]

3 | The factor of points

PointsFactor /functional: tags

Add Comment | Add Other Data | Add Child Requirement | Add Next Requirement | ⇧ | ⇨ | ⇩

[Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent feugiat enim arcu, ut egestas velit. Suspendisse potenti. Etiam risus ante, bibendum ut mattis eget, convallis sit amet nunc.]



Biological Data Analysis

[Biological Data]

CampagneLab's NYoSh

```
ExecutionTool
Execute a job with a GobyWeb plugin in this environment:  RemoteEnvironment
    with owner  manuele.simi

Job name: MyFirstAlignment
Select the Plugin to execute: STAR22_GOBY

Assign one or more FileSet Instances to each slot:
  INPUT_READS : 849-WT (tag: ULBYRRD )

Options for the job:
  Ambiguity threshold : 10
  GENOME_REFERENCE_ID : build NCBI37.57 for organism mouse
  CHUNK_SIZE : 50000000
```

Submit Clear

[Biological Data]

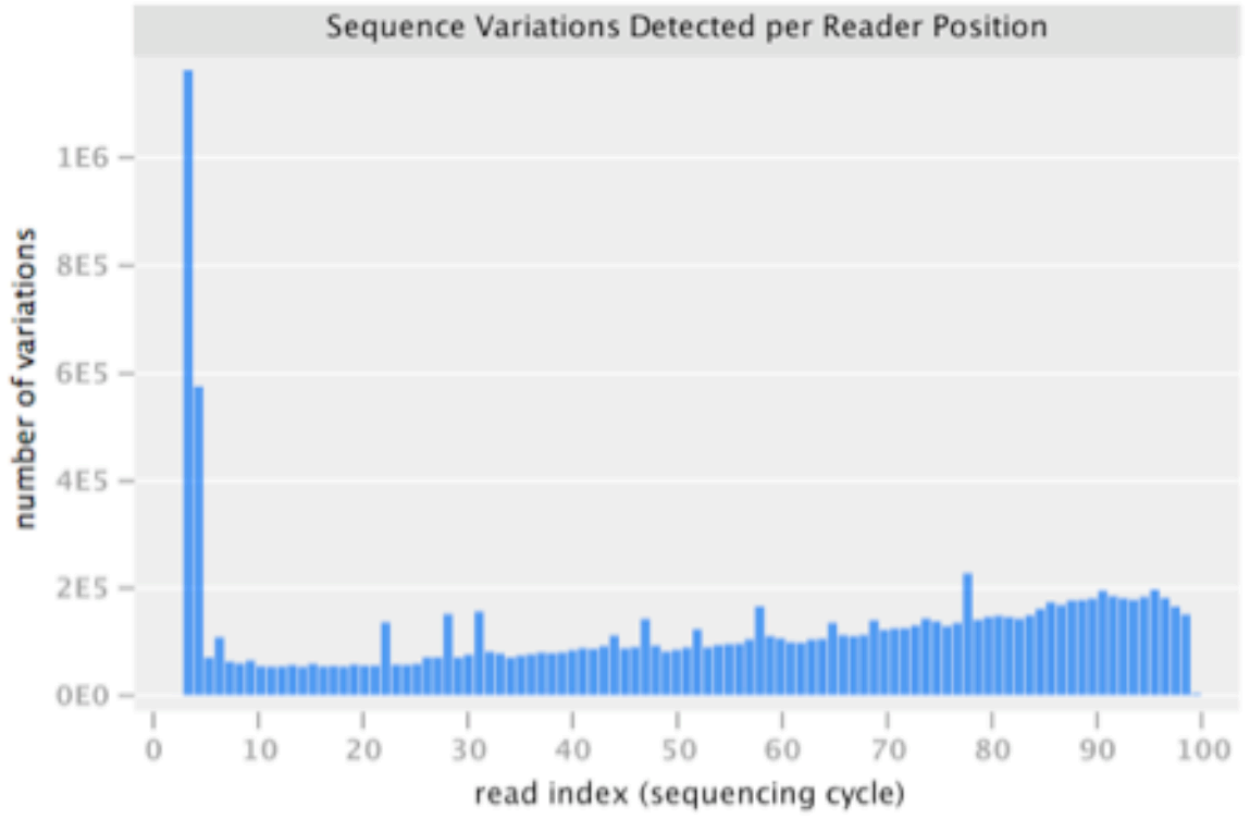
CampagneLab's NYoSh

```
organism set GobyWeb {
  Genomes:
    Homo sapiens ( human )
    Mus musculus ( mouse )
    Rattus norvegicus ( rat )
    Cannis familiaris ( dog )
    Danio rerio ( zebrafish )
    Oryctolagus cuniculus ( rabbit )
    Caenorhabditis elegans ( celegans )
    Xenopus tropicalis ( frog )
  Builds:
    human ( NCBI57 . 37 ) //
    human ( GRCh37 . 70 ) //
    human ( NCBI36 . 54 ) //
    mouse ( NCBI37 . 55 ) // MM9
    mouse ( NCBI37 . 57 ) //
    dog ( BROADD2 . 57 ) //
    zebrafish ( Zv9 . 65 ) // GCA_000002035.2
    rat ( Rnor_5.0 . 76 ) // GCA_000001895.3
    rat ( RGSC3.4.62 . 62 ) //
    rabbit ( oryCun2 . 69 ) //
    celegans ( WBcel215 . 69 ) //
    frog ( JGI_4.2 . 67 ) //
}
```

[Biological Data]

CampagneLab‘s NYoSh

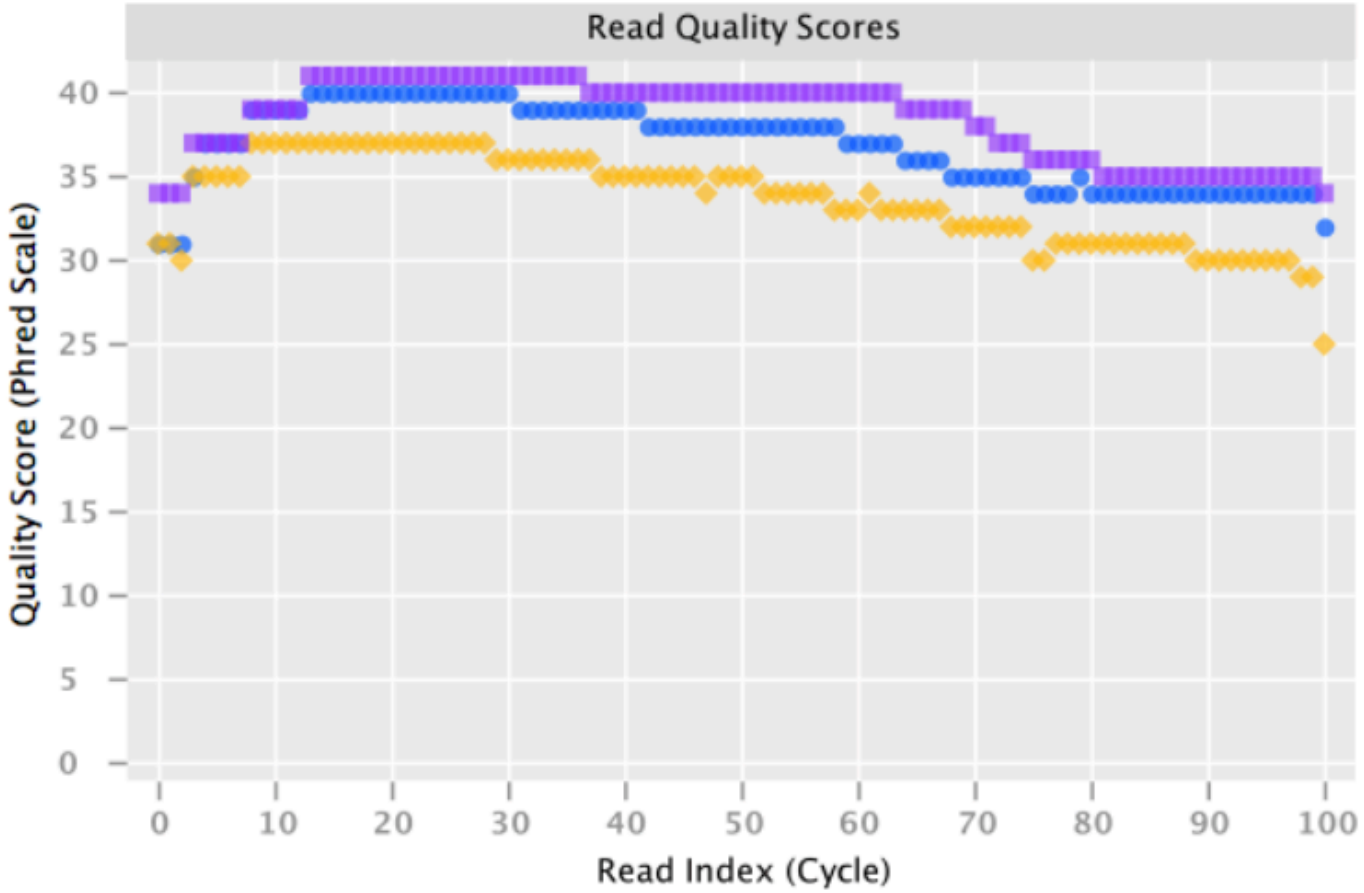
```
goby web alignment 849-WT {
                                                                    tag: VLLYAMY
aligned from sample 849-WT
number alignment entries: 38288002
number of queries: 29019999
```



[Biological Data]

CampagneLab's NYoSh

```
goby 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
}
```



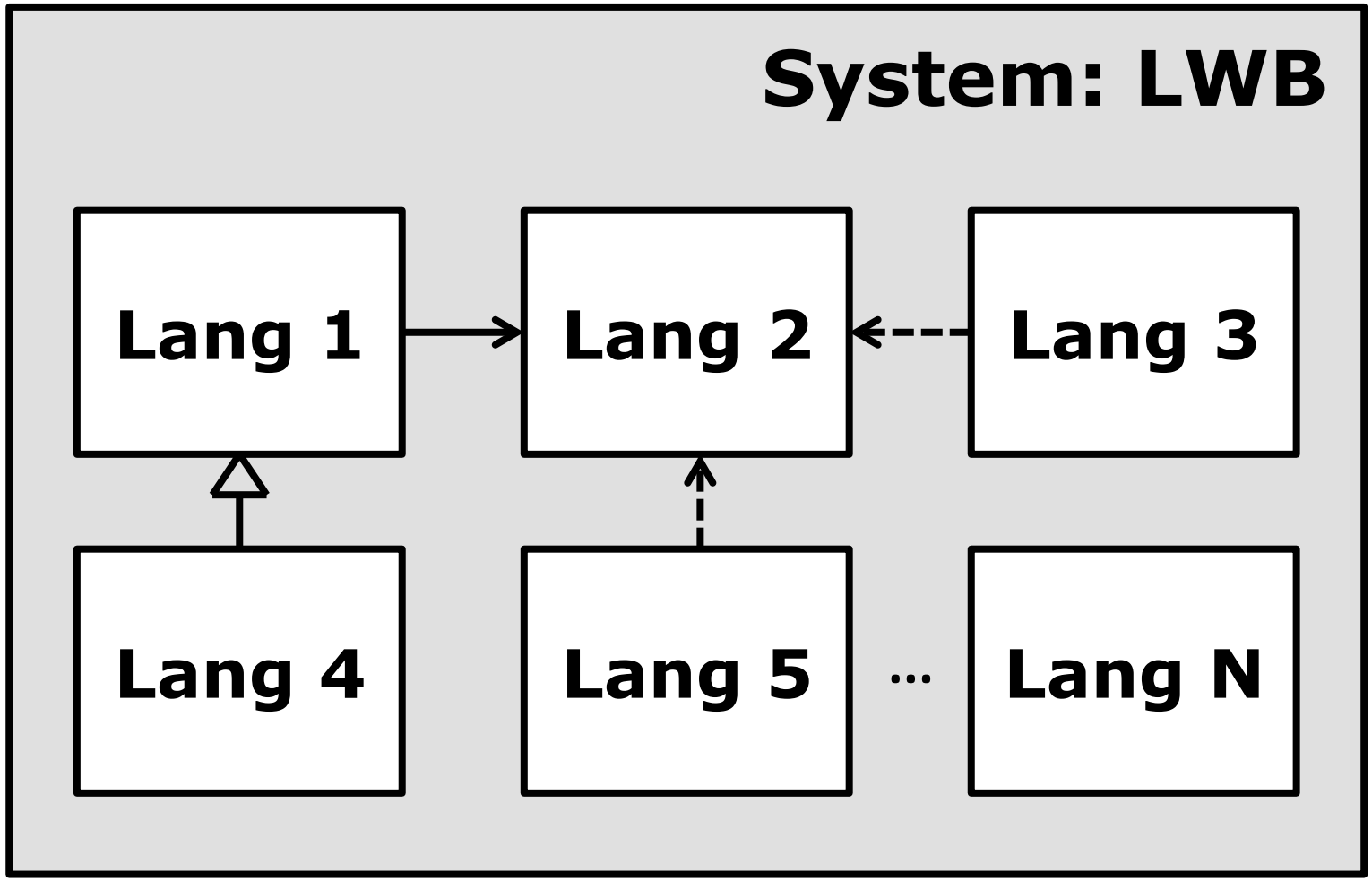
Summary

If you have to build a business app,

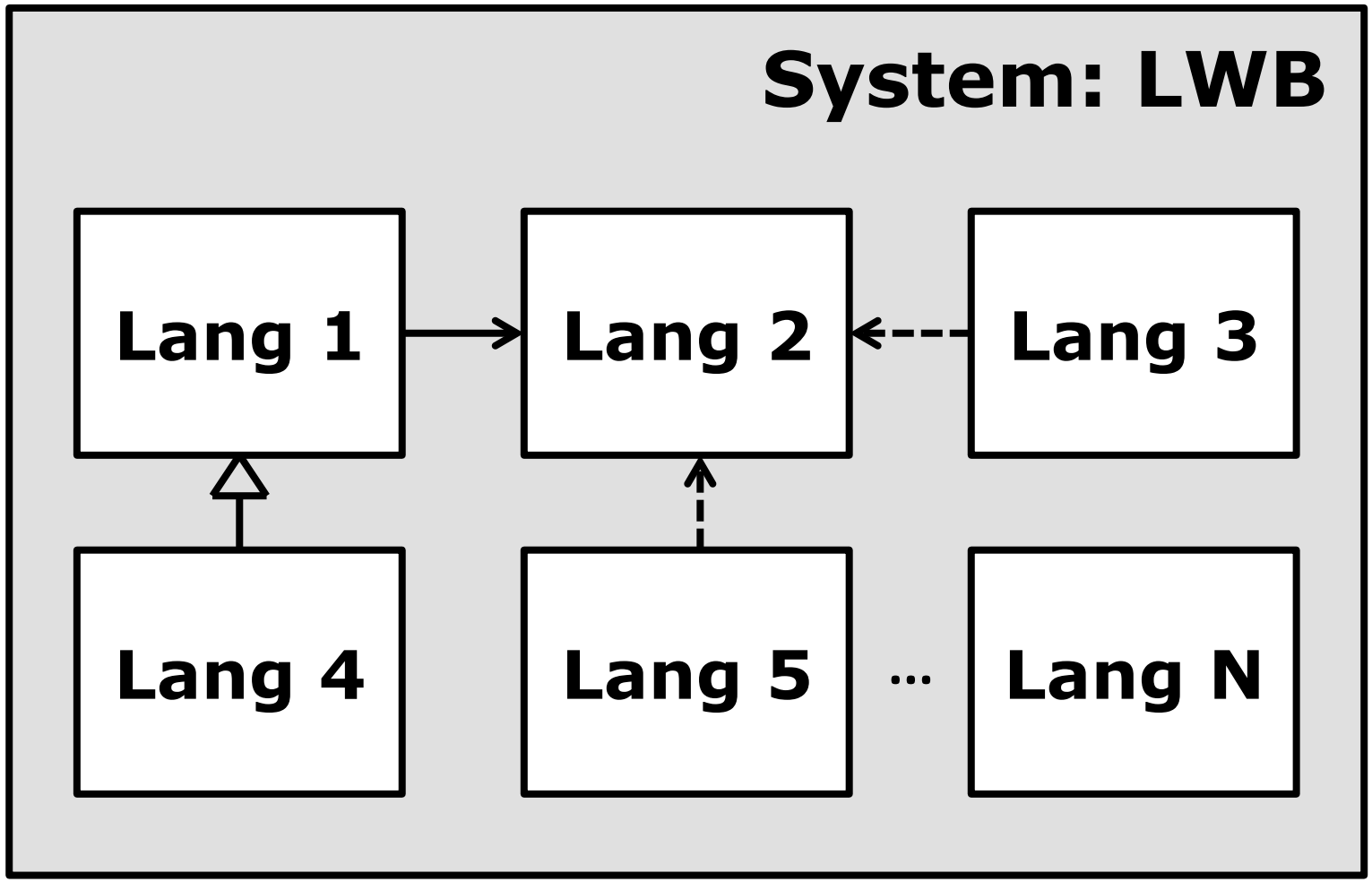
consider using an LWB as the foundation,

and recasting the „application“
as a set of languages.

[LOBA]



[LOBA]



[LOBA]

Business Rules
(Financial) Calculations
Data Structures
Mappings or Queries
Validations
Scientific Processes
Contracts

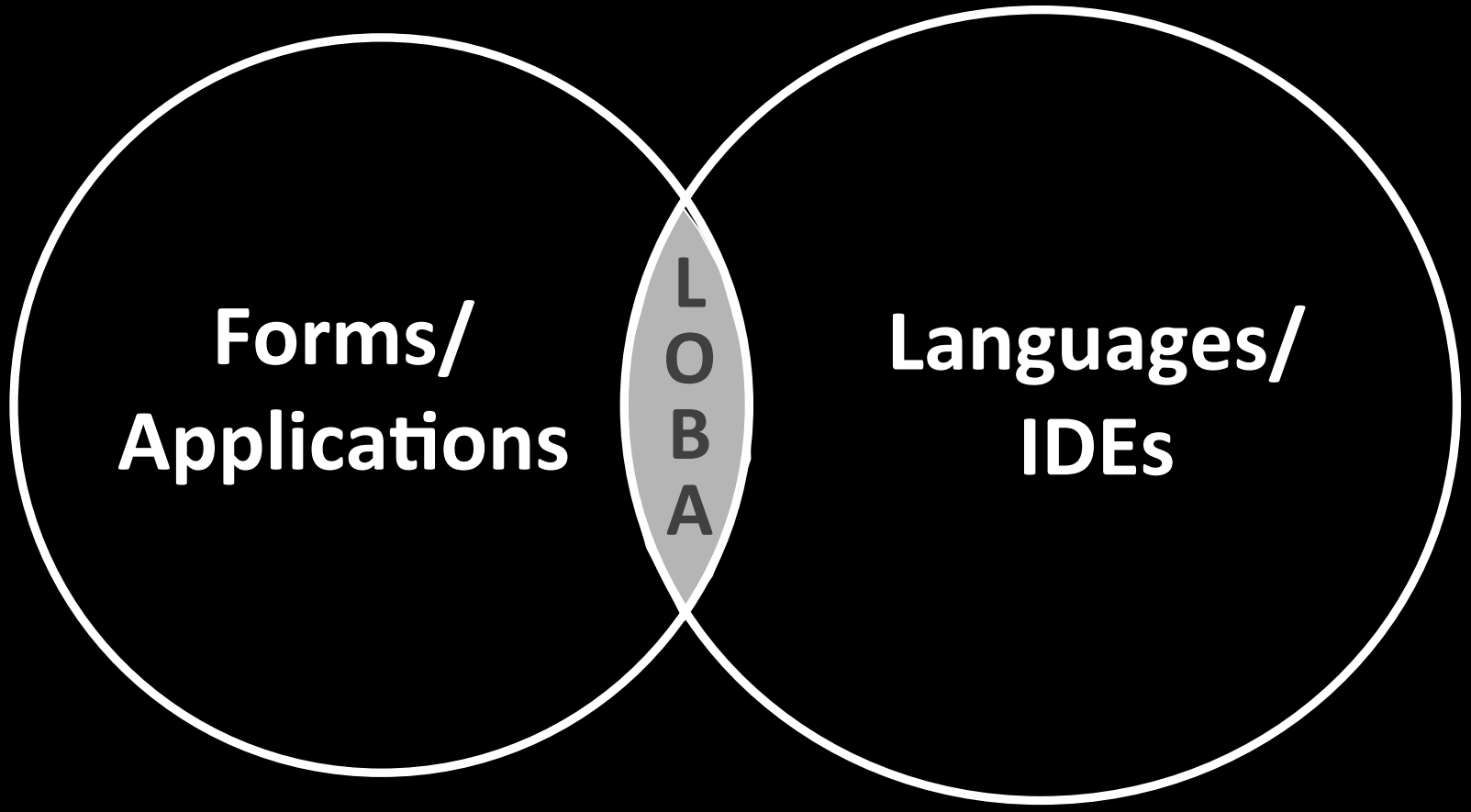
Processes
UI

[LOBA]

Business Rules
(Financial) Calculations
Data Structures
Mappings or Queries
Validations
Scientific Processes
Contracts

Core Business Logic

Processes
UI


Forms/
Applications
LOBA
Languages/
IDEs

Expressivity for Core Domain Knowledge

Build Language for Domain!

User-Friendly Notation

You've seen the demos.

Testing

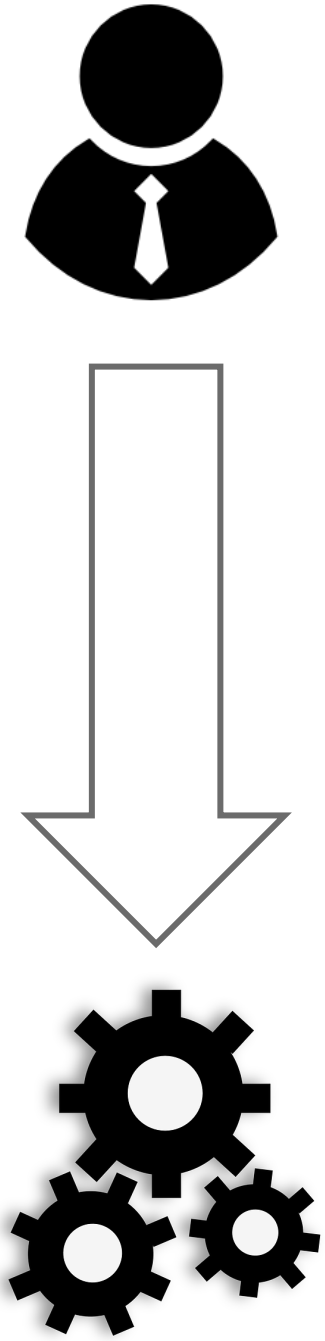
Integrated DSLs for testing.

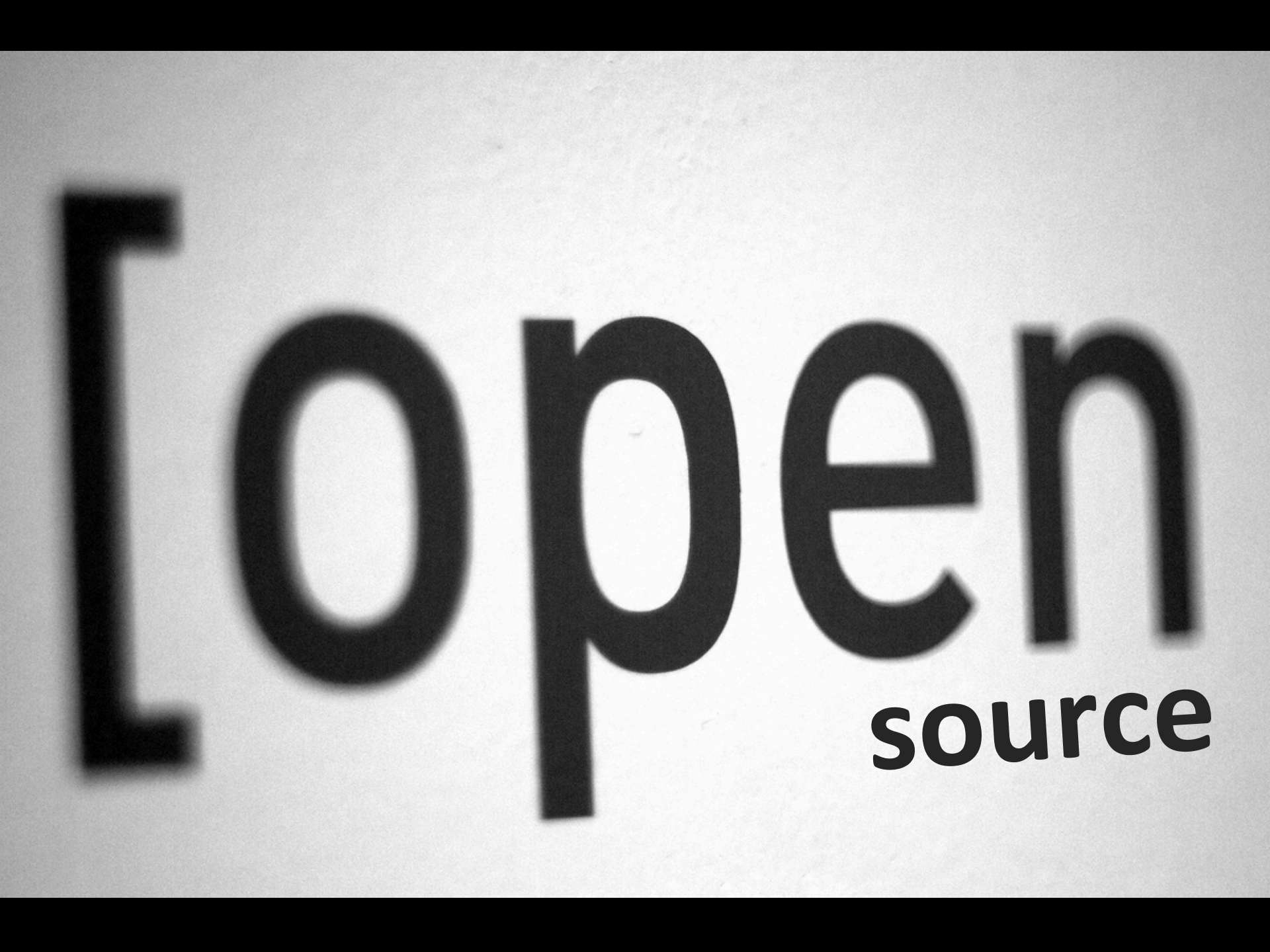
Meaningful Analyses

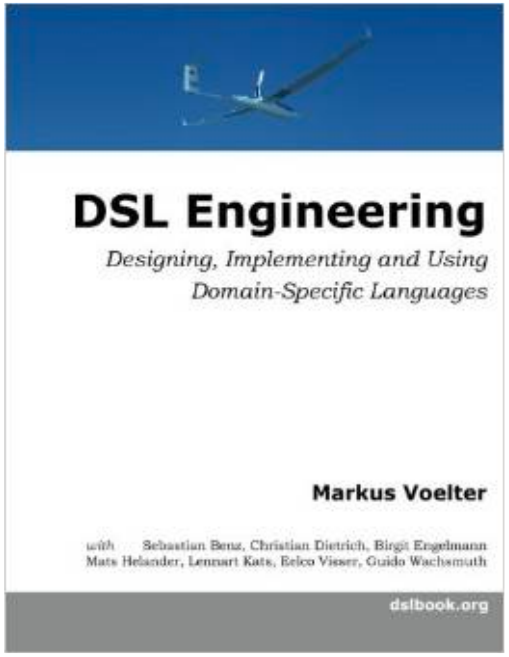
Types, Consistency, Checking

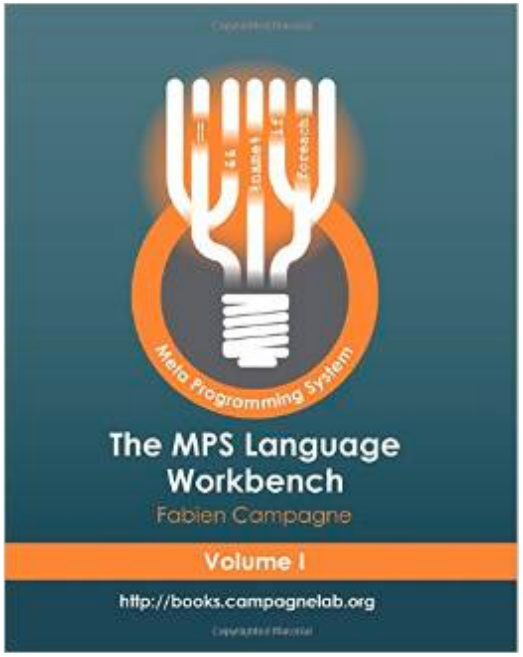
Synthesis of Software

Code Generation.




open
source





mbeddr

MPS Meta Programming System

Language Workbench Challenge

Comparing Tools of the Trade

Check out

`http://languageworkbenches.net`

... for an overview over other Language Workbenches

Thank you!

voelter@acm.org
www.voelter.de
@markusvoelter