OOP 2014
SOFTWARE MEETS BUSINESS

COMPLEXITY:
Managing Today's Challenges
München, 03. - 07. Februar 2014

Language? Form? IDE? Application?

Towards Language-Oriented Business Apps

Markus Voelter
independent/itemis
voelter@acm.org
www.voelter.de
@markusvoelter

Bernd Kolb
itemis
kolb@itemis.de
www.itemis.de
@berndkolb

M

1 About mbeddr

2 mbeddr Demo 1

3 JetBrains MPS

4 mbeddr Demo 2

5 Generalization

6 LOBA
why
where we are
what is missing

7 Conclusions

1

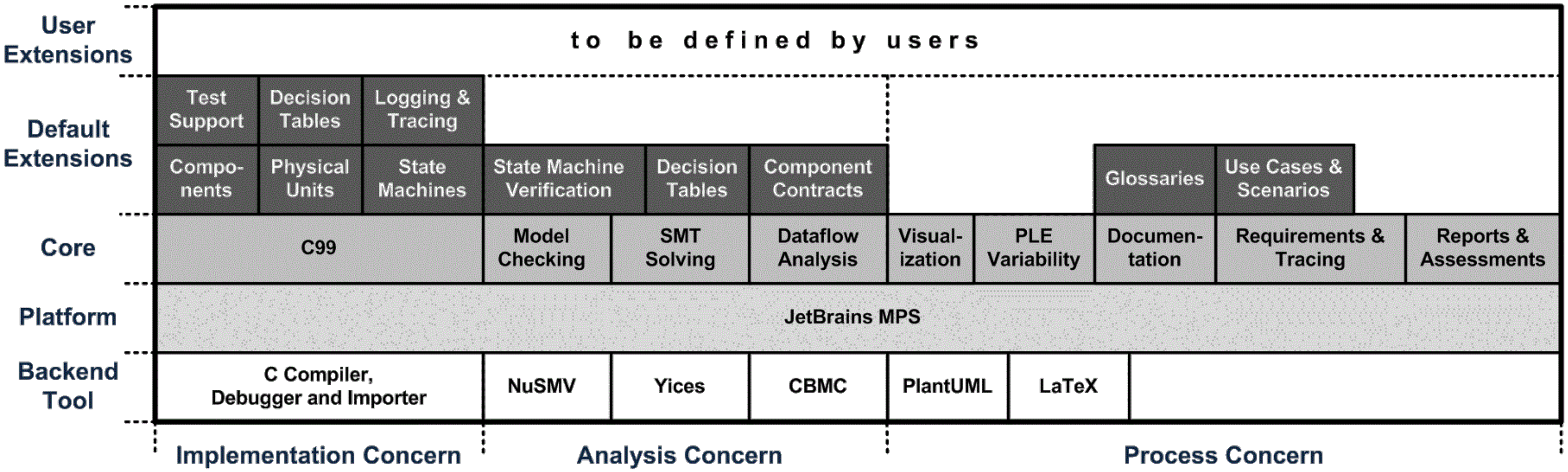
About mbeddr

B

About mbeddr

Language Engineering Embedded Software

An extensible collection of integrated languages for embedded software engineering.



User Extensions: to be defined by users

Default Extensions: Test Support, Decision Tables, Logging & Tracing, Components, Physical Units, State Machines, State Machine Verification, Decision Tables, Component Contracts, Glossaries, Use Cases & Scenarios

Core: C99, Model Checking, SMT Solving, Dataflow Analysis, Visualization, PLE Variability, Documentation, Requirements & Tracing, Reports & Assessments

Platform: JetBrains MPS

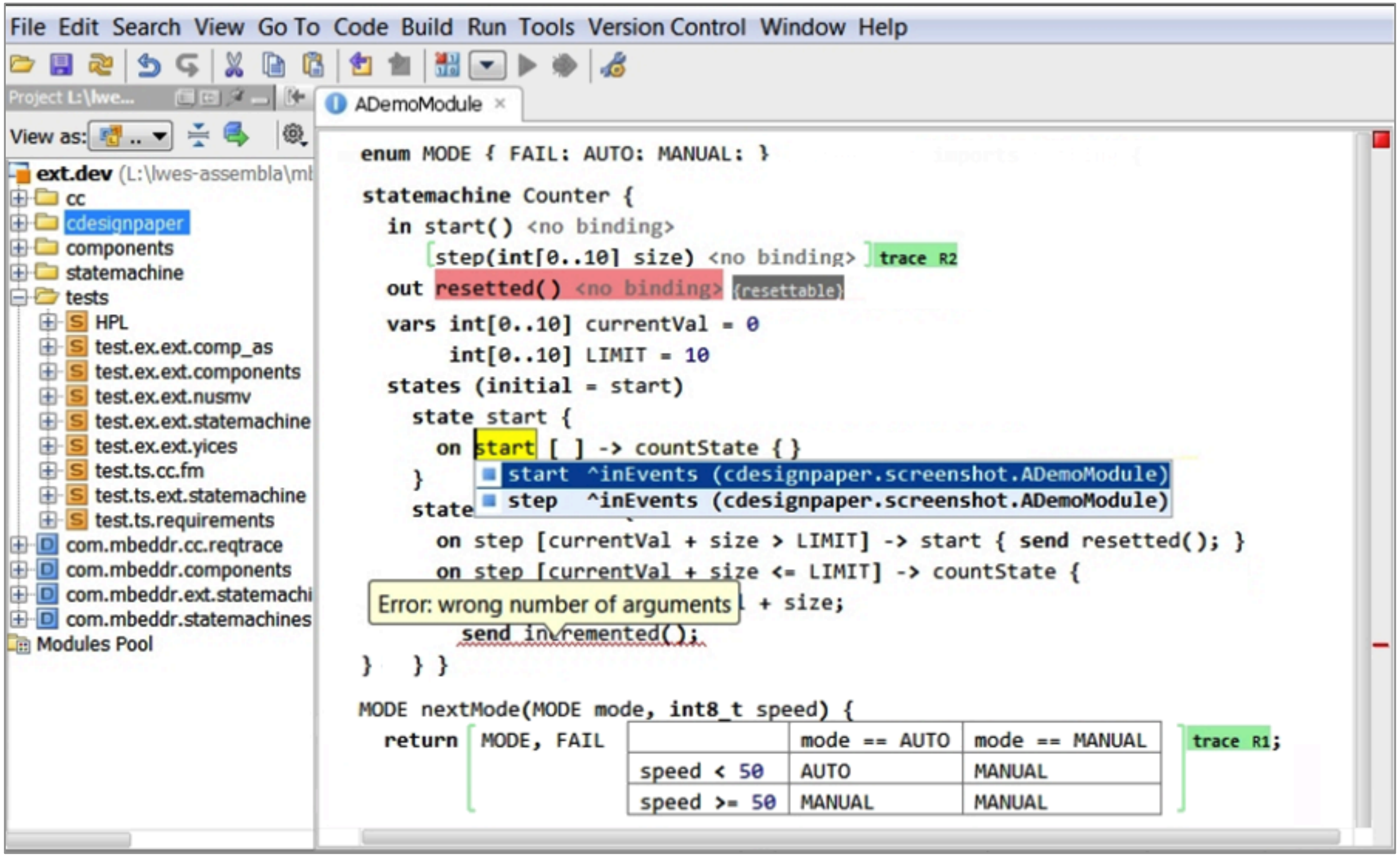
Backend Tool: C Compiler, Debugger and Importer, NuSMV, Yices, CBMC, PlantUML, LaTeX

Implementation Concern, Analysis Concern, Process Concern

About mbeddr

mbeddr

Language Engineering Embedded Software



```
enum MODE { FAIL: AUTO: MANUAL: }

statemachine Counter {
  in start() <no binding>
     [step(int[0..10] size) <no binding> ] trace R2
  out resetted() <no binding> {resettable}
  vars int[0..10] currentVal = 0
       int[0..10] LIMIT = 10
  states (initial = start)
    state start {
      on start [ ] -> countState { }
         start  ^inEvents (cdesignpaper.screenshot.ADemoModule)
    }    step   ^inEvents (cdesignpaper.screenshot.ADemoModule)
    state
      on step [currentVal + size > LIMIT] -> start { send resetted(); }
      on step [currentVal + size <= LIMIT] -> countState {
Error: wrong number of arguments l + size;
        send incremented();
}   } }

MODE nextMode(MODE mode, int8_t speed) {
  return MODE, FAIL
```

	mode == AUTO	mode == MANUAL
speed < 50	AUTO	MANUAL
speed >= 50	MANUAL	MANUAL

trace R1;

An IDE + Debugger for all of them

About mbeddr

mbeddr

Language Engineering Embedded Software

Open Source
Eclipse Public License

eclipse

http://mbeddr.com

http://www.eclipse.org/proposals/technology.mbeddr/

itemis

fortiss

BMW Car IT

Bundesministerium
für Bildung
und Forschung

2

mbeddr Demo I

B

3

JetBrains MPS

About mbeddr

Built on JetBrains MPS

A Language Workbench

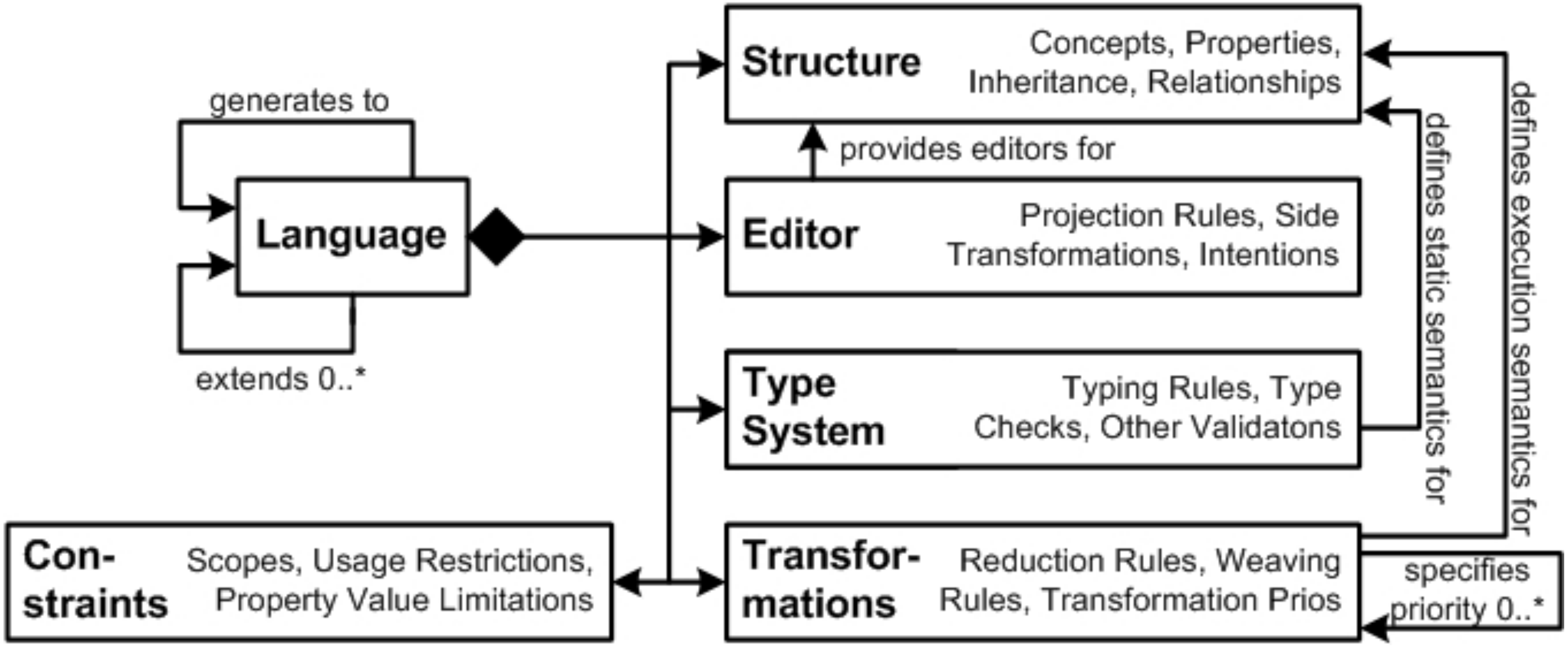
About mbeddr

Built on JetBrains MPS

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

About MPS

Rich Set of Language Aspects

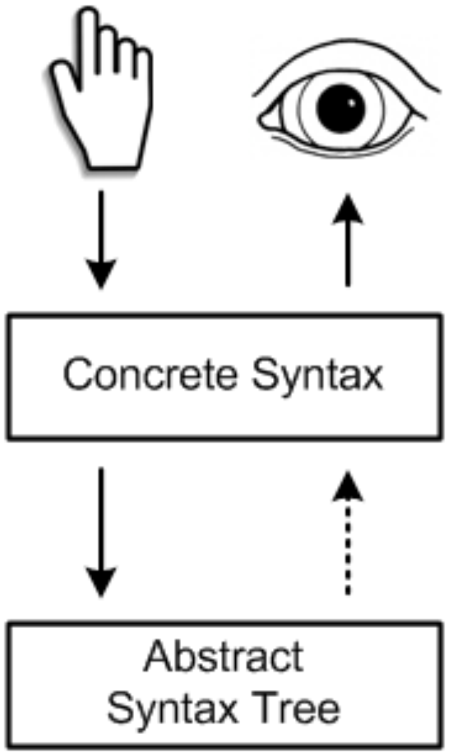

generates to
Language
extends 0..*
Structure
Concepts, Properties, Inheritance, Relationships
provides editors for
Editor
Projection Rules, Side Transformations, Intentions
Type System
Typing Rules, Type Checks, Other Validatons
Con-straints
Scopes, Usage Restrictions, Property Value Limitations
Transfor-mations
Reduction Rules, Weaving Rules, Transformation Prios
defines static semantics for
defines execution semantics for
specifies priority 0..*

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

About MPS

Projectional Editing

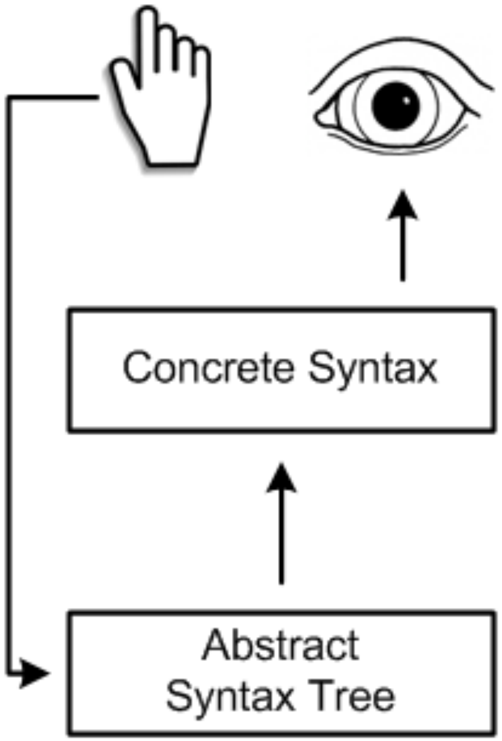
Parsing



Concrete Syntax

Abstract Syntax Tree

Projection



Concrete Syntax

Abstract Syntax Tree

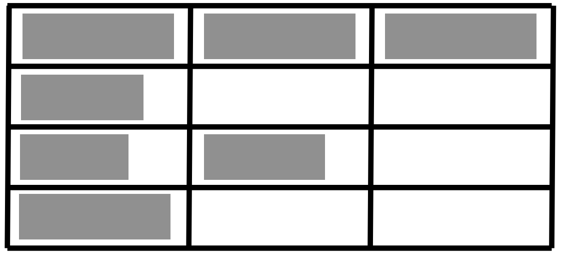
About MPS

Notational Flexibility

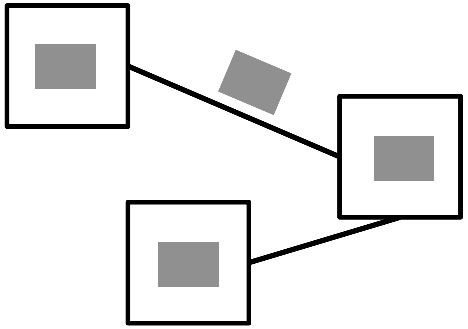
Regular Code/Text

Mathematical

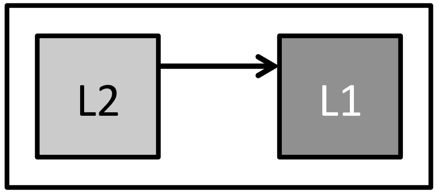
Tables



Graphical

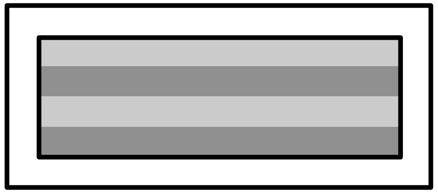


Language Composition



Separate Files

Type System
Transformation
Constraints



In One File

Type System
Transformation
Constraints
Syntax
Editor/IDE

4

mbeddr Demo II

B

5

Generalization

M

Thought Process

From Data Formats To Languages

Structure, Constraints, Semantics

Data Format *+ Syntax + IDE*

Language

Thought Process

Language Engineering

Languages

Language Reuse

Language Modularization

Language Composition

Language Engineering

Languages

Language Engineering

Text Math Graphics

Tables Symbols Forms

Syntactic Diversity

Thought Process

Language Workbenches

Languages

Language Engineering

Syntactic Diversity

But does this really work?

Language Workbenches

Generic Tools, Specific Languages

Ingredients

Specific Languages

- Languages
- Language Engineering
- Syntactic Diversity

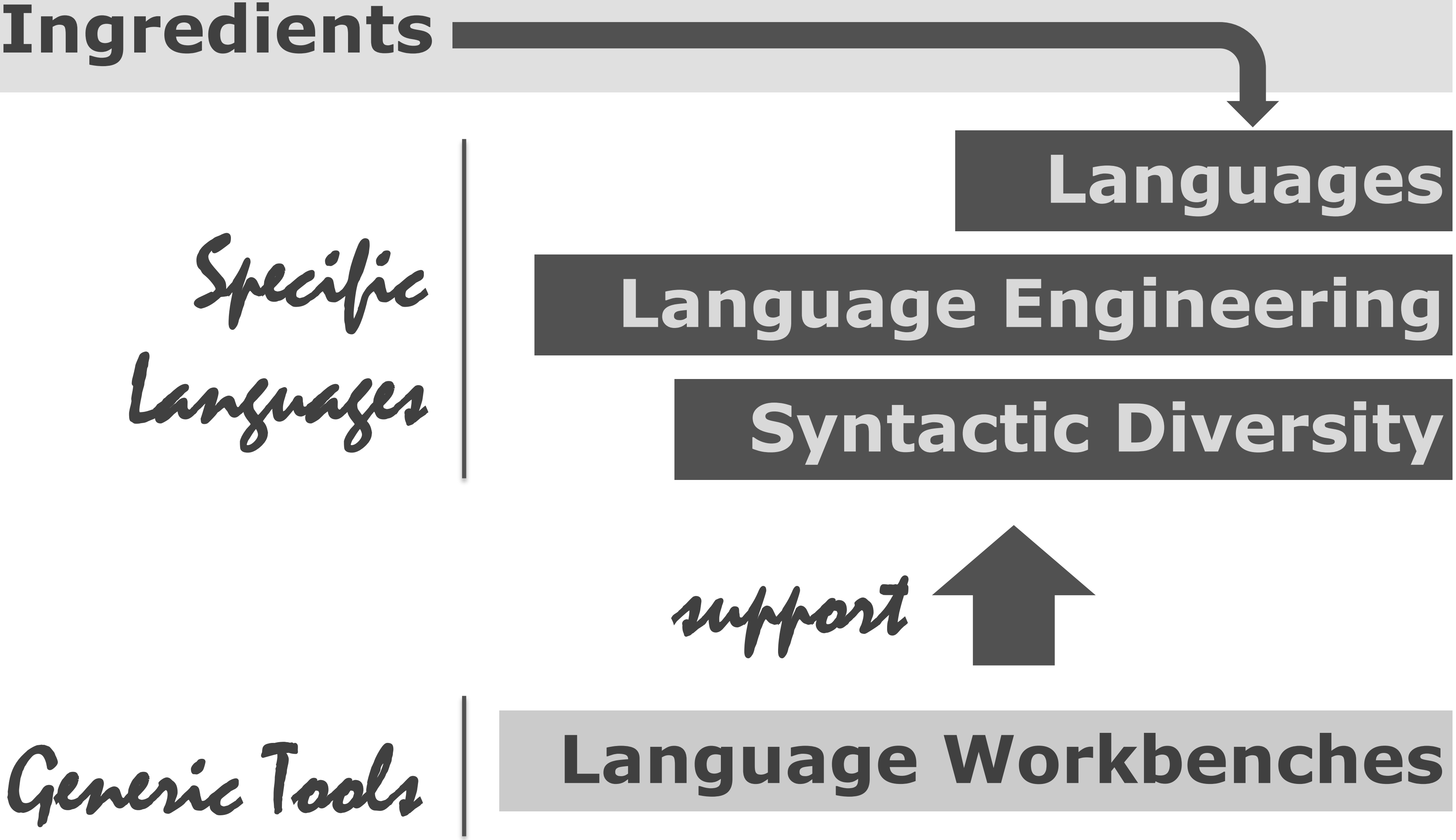
Generic Tools

- Language Workbenches

(we don't have to reimplement editors and synchronizers)

Generic Tools, Specific Languages

Ingredients



Languages

Language Engineering

Syntactic Diversity

support

Language Workbenches

Language Workbenches

Typical Features

Language Definition, Reuse, Extension, Composition

Mixing Notations

Type Systems, Constraints, Transformation, Interpretation

Language Workbenches

Typical Features

Goto Definition/Find Usages

Error Markup/Quick Fixes

Syntax Highlighting

Code Completion

Search/Replace

Refactoring

Debugging

Reporting

Visualization

Version Control

Language Workbenches

Typical Features

for any Language!

Language Workbenches

Typical Features

Language Workbenches act as the foundation for IDEs for any language.

6

LOBA

why
where we are
what is missing

M

6

LOBA

why

where we are

what is missing

M

LOBA: Why

For which kinds of Systems?

Language Workbenches act as the foundation for IDEs for any language.

LOBA: Why

For which kinds of Systems?

Language Workbenches act as the foundation for IDEs for (m)any applications.

LOBA: Why

For which kinds of Systems?

many applications?

Structured or Formalizable

Mathematical

Data-Oriented

Language-y

LOBA: Why

For which kinds of Systems?

many applications?

Data Models

Pricing Calculations

Financial Calculations

Business Rules

Contracts

Highly Structured Requirements

LOBA: Why

[Motivation] Languages!

A language may be hiding behind many of these!

LOBA: Why

[Motivation] Languages!

But: users don‘t want to be programmers!

LOBA: Why

[Motivation] Languages!

Combine the best of

Applications/Forms/UIs

and

Languages and IDEs.

LOBA: Why

[Motivation] Languages!

Applications/Forms/Uis

Structure

User Guidance

Tables

Views

vs.

Languages + IDEs

Expressions

Complex Structures

Code Completion

Type Checking

Debugging

Refactoring

6

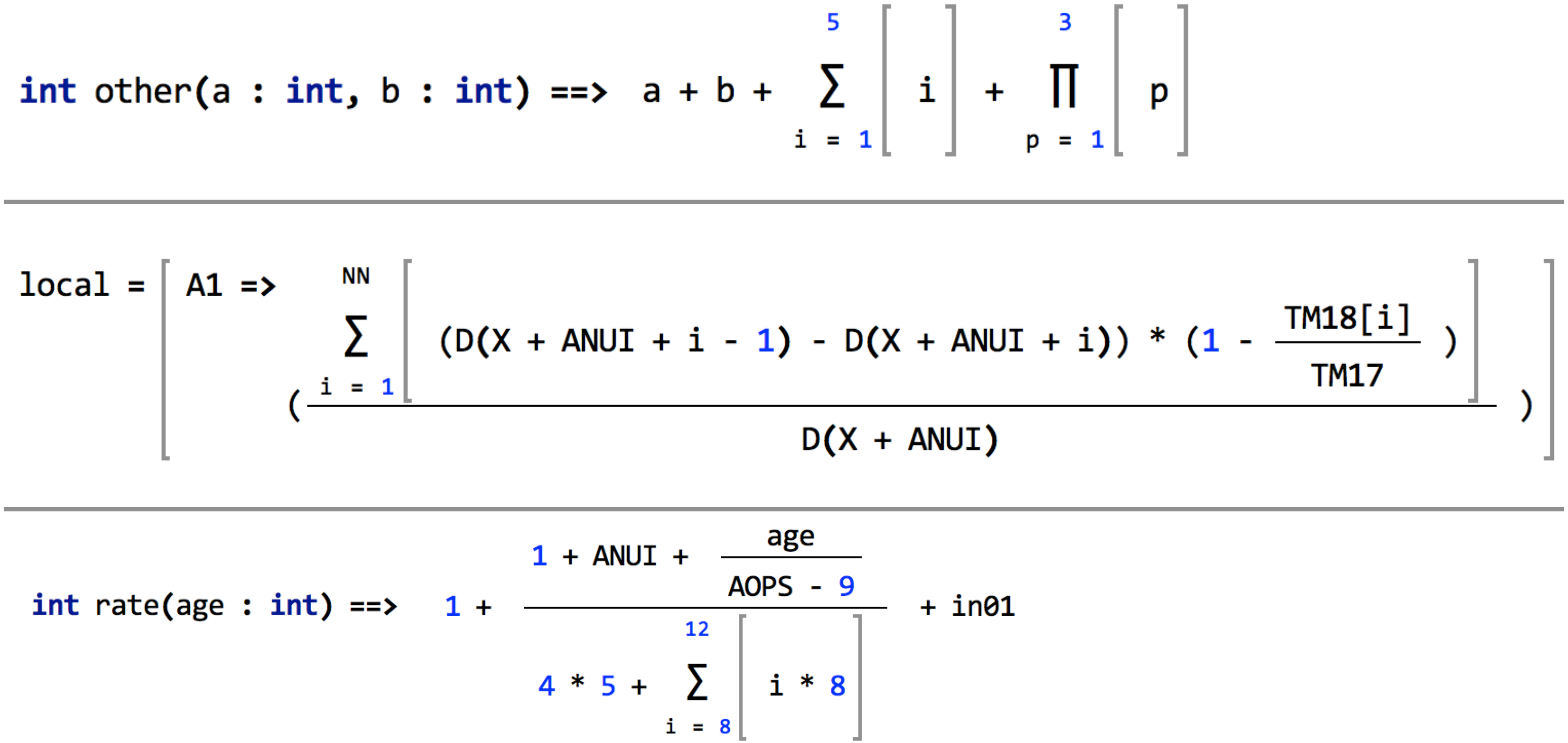
LOBA

why
where we are
what is missing

M

LOBA: Where we are.

[Notation] Math



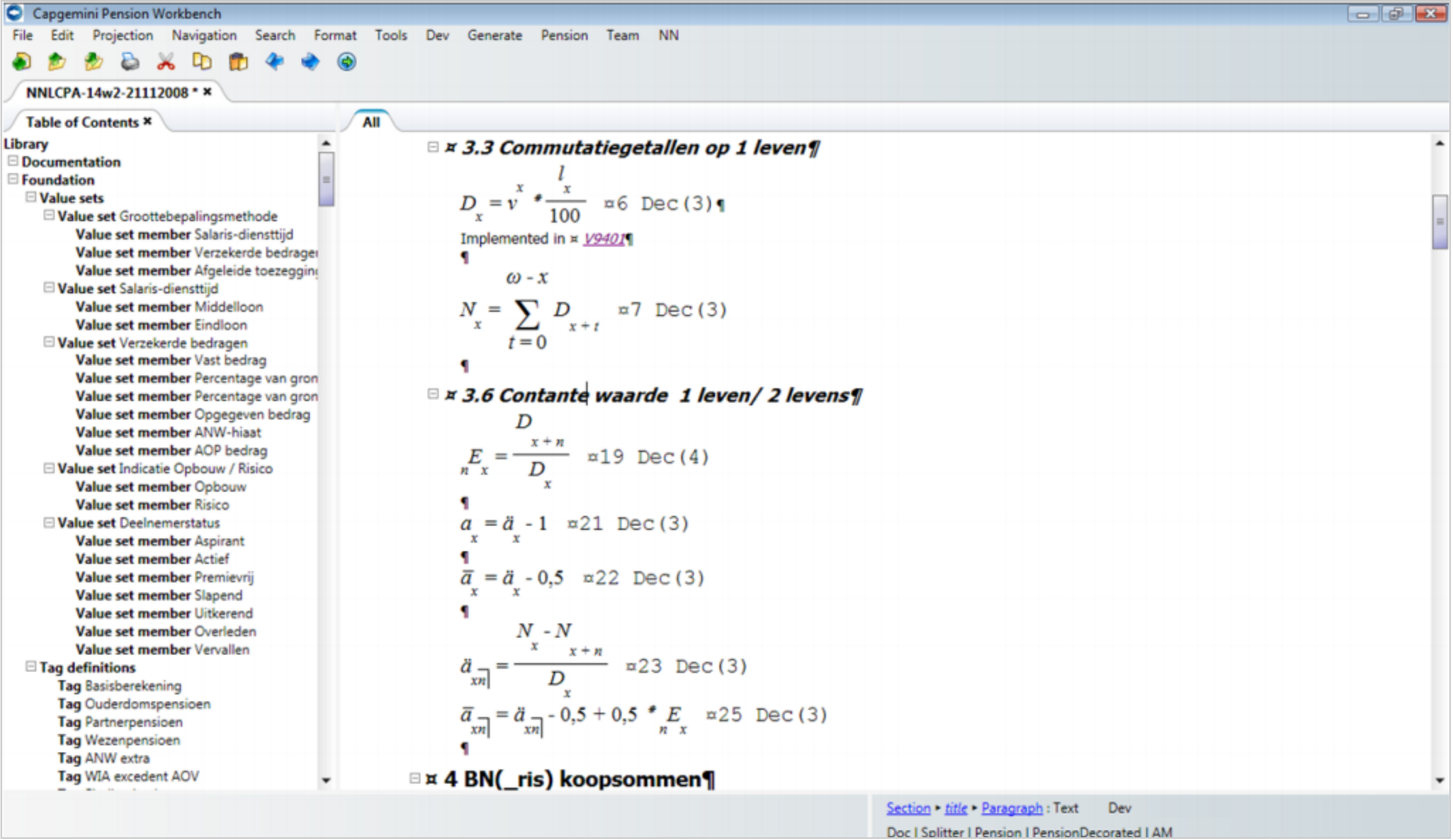
$$\text{int other(a : int, b : int) ==>}\quad a + b + \sum_{i = 1}^{5}\left[\, i \,\right] + \prod_{p = 1}^{3}\left[\, p \,\right]$$

$$\text{local} = \left[\ \text{A1 =>}\ \left(\frac{\sum_{i = 1}^{NN}\left[(D(X + ANUI + i - 1) - D(X + ANUI + i)) * \left(1 - \frac{TM18[i]}{TM17}\right) \right]}{D(X + ANUI)} \right) \right]$$

$$\text{int rate(age : int) ==>}\quad 1 + \frac{1 + ANUI + \frac{age}{AOPS - 9}}{4 * 5 + \sum_{i = 8}^{12}\left[\, i * 8 \,\right]} + in01$$

LOBA: Where we are.

[Notation] Math II


Capgemini Pension Workbench
File Edit Projection Navigation Search Format Tools Dev Generate Pension Team NN
NNLCPA-14w2-21112008 *
Table of Contents
All
Section ▸ title ▸ Paragraph : Text Dev
Doc | Splitter | Pension | PensionDecorated | AM

Library
- Documentation
- Foundation
  - Value sets
    - **Value set** Grootteberpalingsmethode
      - **Value set member** Salaris-diensttijd
      - **Value set member** Verzekerde bedrage
      - **Value set member** Afgeleide toezeggin
    - **Value set** Salaris-diensttijd
      - **Value set member** Middelloon
      - **Value set member** Eindloon
    - **Value set** Verzekerde bedragen
      - **Value set member** Vast bedrag
      - **Value set member** Percentage van gron
      - **Value set member** Percentage van gron
      - **Value set member** Opgegeven bedrag
      - **Value set member** ANW-hiaat
      - **Value set member** AOP bedrag
    - **Value set** Indicatie Opbouw / Risico
      - **Value set member** Opbouw
      - **Value set member** Risico
    - **Value set** Deelnemerstatus
      - **Value set member** Aspirant
      - **Value set member** Actief
      - **Value set member** Premievrij
      - **Value set member** Slapend
      - **Value set member** Uitkerend
      - **Value set member** Overleden
      - **Value set member** Vervallen
  - Tag definitions
    - **Tag** Basisberekening
    - **Tag** Ouderdomspensioen
    - **Tag** Partnerpensioen
    - **Tag** Wezenpensioen
    - **Tag** ANW extra
    - **Tag** WIA excedent AOV

¤ *3.3 Commutatiegetallen op 1 leven*¶

$D_x = v^x * \frac{l_x}{100}$ ¤6 Dec(3)¶

Implemented in ¤ *V9401*¶

¶

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

¶

¤ *3.6 Contante waarde 1 leven/ 2 levens*¶

${}_nE_x = \frac{D_{x+n}}{D_x}$ ¤19 Dec(4)

¶

$a_x = \ddot{a}_x - 1$ ¤21 Dec(3)

¶

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

¶

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

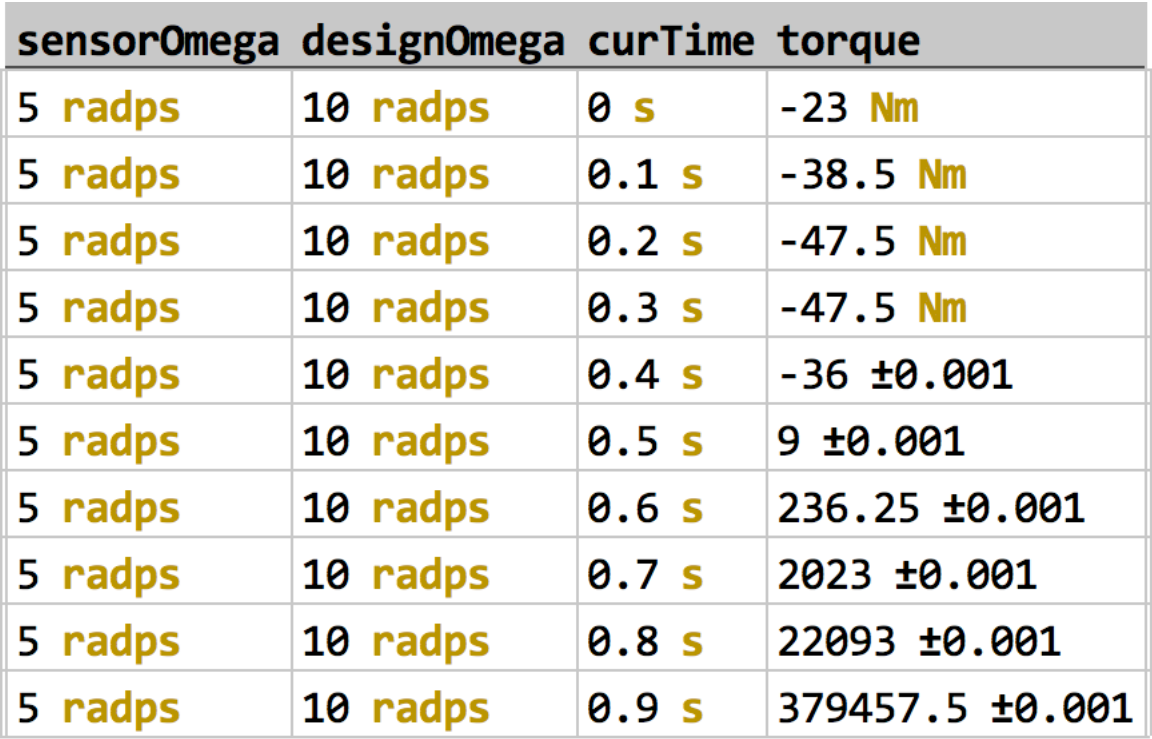
$\bar{a}_{x\overline{n}|} = \ddot{a}_{x\overline{n}|} - 0{,}5 + 0{,}5 * {}_nE_x$ ¤25 Dec(3)

¶

¤ 4 BN(_ris) koopsommen¶

LOBA: Where we are.

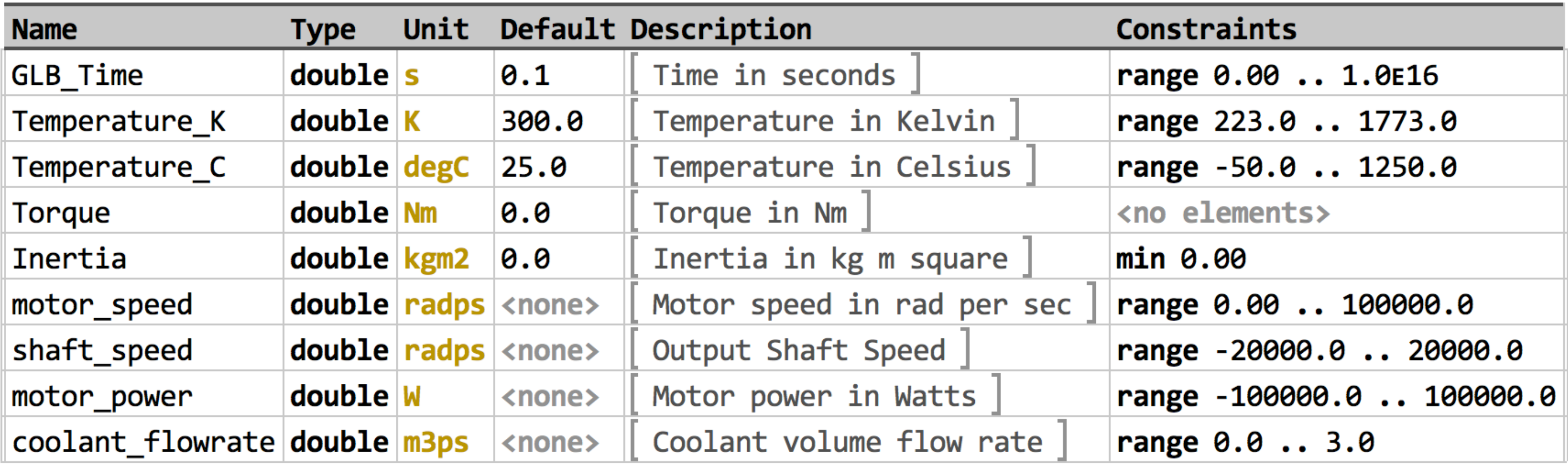
[Notation] Tables



sensorOmega	designOmega	curTime	torque
5 radps	10 radps	0 s	-23 Nm
5 radps	10 radps	0.1 s	-38.5 Nm
5 radps	10 radps	0.2 s	-47.5 Nm
5 radps	10 radps	0.3 s	-47.5 Nm
5 radps	10 radps	0.4 s	-36 ±0.001
5 radps	10 radps	0.5 s	9 ±0.001
5 radps	10 radps	0.6 s	236.25 ±0.001
5 radps	10 radps	0.7 s	2023 ±0.001
5 radps	10 radps	0.8 s	22093 ±0.001
5 radps	10 radps	0.9 s	379457.5 ±0.001

LOBA: Where we are.

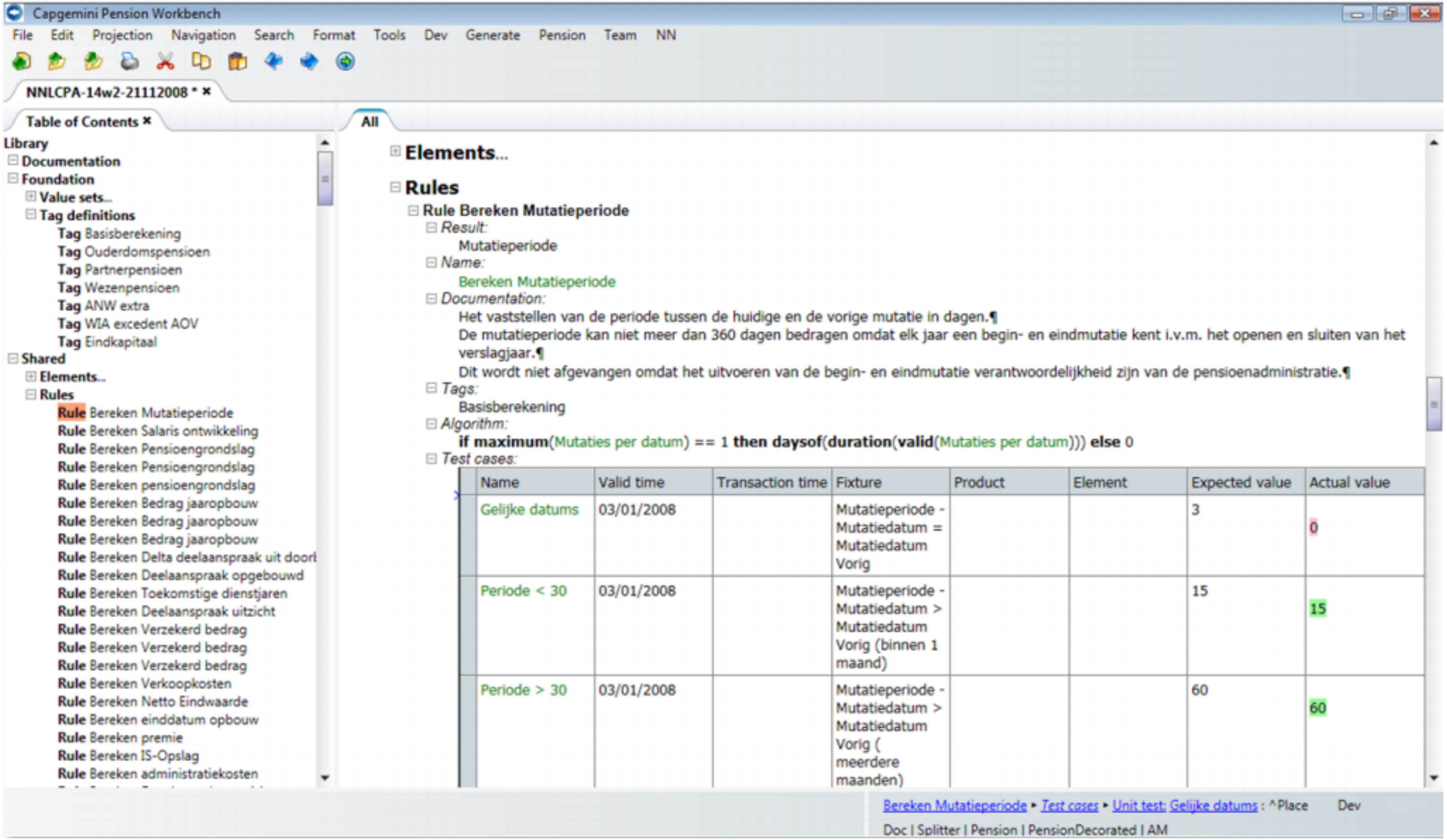
[Notation] Tables II



Name	Type	Unit	Default	Description	Constraints
GLB_Time	double	s	0.1	[Time in seconds]	range 0.00 .. 1.0E16
Temperature_K	double	K	300.0	[Temperature in Kelvin]	range 223.0 .. 1773.0
Temperature_C	double	degC	25.0	[Temperature in Celsius]	range -50.0 .. 1250.0
Torque	double	Nm	0.0	[Torque in Nm]	<no elements>
Inertia	double	kgm2	0.0	[Inertia in kg m square]	min 0.00
motor_speed	double	radps	<none>	[Motor speed in rad per sec]	range 0.00 .. 100000.0
shaft_speed	double	radps	<none>	[Output Shaft Speed]	range -20000.0 .. 20000.0
motor_power	double	W	<none>	[Motor power in Watts]	range -100000.0 .. 100000.0
coolant_flowrate	double	m3ps	<none>	[Coolant volume flow rate]	range 0.0 .. 3.0

LOBA: Where we are.

[Notation] Tables III


Capgemini Pension Workbench
File Edit Projection Navigation Search Format Tools Dev Generate Pension Team NN
NNLCPA-14w2-21112008 * ×
Table of Contents ×
All

Library
⊟ Documentation
⊟ Foundation
⊞ Value sets...
⊟ Tag definitions
Tag Basisberekening
Tag Ouderdomspensioen
Tag Partnerpensioen
Tag Wezenpensioen
Tag ANW extra
Tag WIA excedent AOV
Tag Eindkapitaal
⊟ Shared
⊞ Elements...
⊟ Rules
Rule Bereken Mutatieperiode
Rule Bereken Salaris ontwikkeling
Rule Bereken Pensioengrondslag
Rule Bereken Pensioengrondslag
Rule Bereken pensioengrondslag
Rule Bereken Bedrag jaaropbouw
Rule Bereken Bedrag jaaropbouw
Rule Bereken Bedrag jaaropbouw
Rule Bereken Delta deelaanspraak uit doorl
Rule Bereken Deelaanspraak opgebouwd
Rule Bereken Toekomstige dienstjaren
Rule Bereken Deelaanspraak uitzicht
Rule Bereken Verzekerd bedrag
Rule Bereken Verzekerd bedrag
Rule Bereken Verzekerd bedrag
Rule Bereken Verkoopkosten
Rule Bereken Netto Eindwaarde
Rule Bereken einddatum opbouw
Rule Bereken premie
Rule Bereken IS-Opslag
Rule Bereken administratiekosten

⊞ **Elements...**

⊟ **Rules**

⊟ **Rule Bereken Mutatieperiode**

⊟ *Result:*
Mutatieperiode

⊟ *Name:*
Bereken Mutatieperiode

⊟ *Documentation:*
Het vaststellen van de periode tussen de huidige en de vorige mutatie in dagen.¶
De mutatieperiode kan niet meer dan 360 dagen bedragen omdat elk jaar een begin- en eindmutatie kent i.v.m. het openen en sluiten van het verslagjaar.¶
Dit wordt niet afgevangen omdat het uitvoeren van de begin- en eindmutatie verantwoordelijkheid zijn van de pensioenadministratie.¶

⊟ *Tags:*
Basisberekening

⊟ *Algorithm:*

```
if maximum(Mutaties per datum) == 1 then daysof(duration(valid(Mutaties per datum))) else 0
```

⊟ *Test cases:*

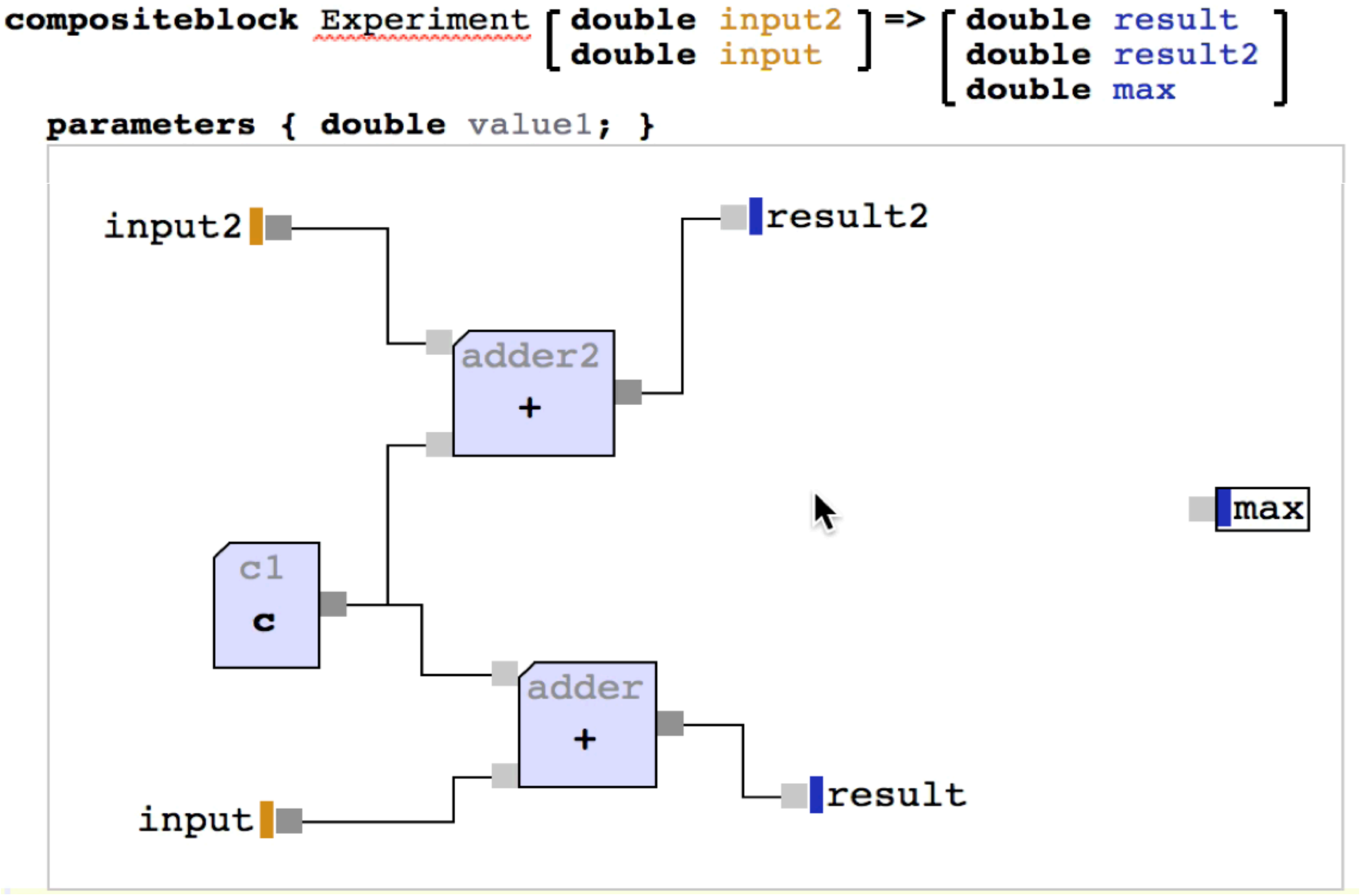
Name	Valid time	Transaction time	Fixture	Product	Element	Expected value	Actual value
Gelijke datums	03/01/2008		Mutatieperiode - Mutatiedatum = Mutatiedatum Vorig			3	0
Periode < 30	03/01/2008		Mutatieperiode - Mutatiedatum > Mutatiedatum Vorig (binnen 1 maand)			15	15
Periode > 30	03/01/2008		Mutatieperiode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

Bereken Mutatieperiode ▸ *Test cases* ▸ Unit test: Gelijke datums : ^Place Dev
Doc | Splitter | Pension | PensionDecorated | AM

LOBA: Where we are.

[Notation] Graphical

D

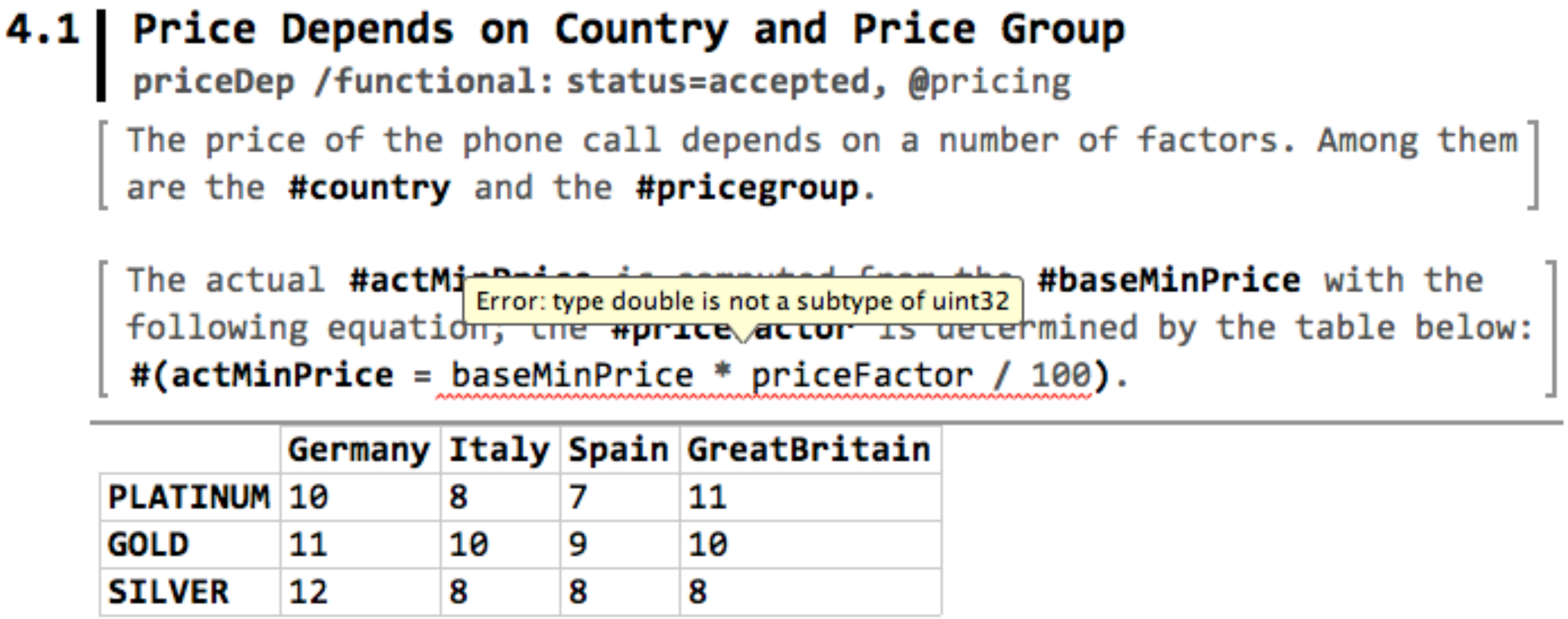


```
compositeblock Experiment [ double input2 ] => [ double result
                            double input  ]      double result2
                                                 double max     ]
  parameters { double value1; }
```

LOBA: Where we are.

[Notation] Mixed Content

D



4.1 Price Depends on Country and Price Group

priceDep /functional: status=accepted, @pricing

The price of the phone call depends on a number of factors. Among them are the **#country** and the **#pricegroup**.

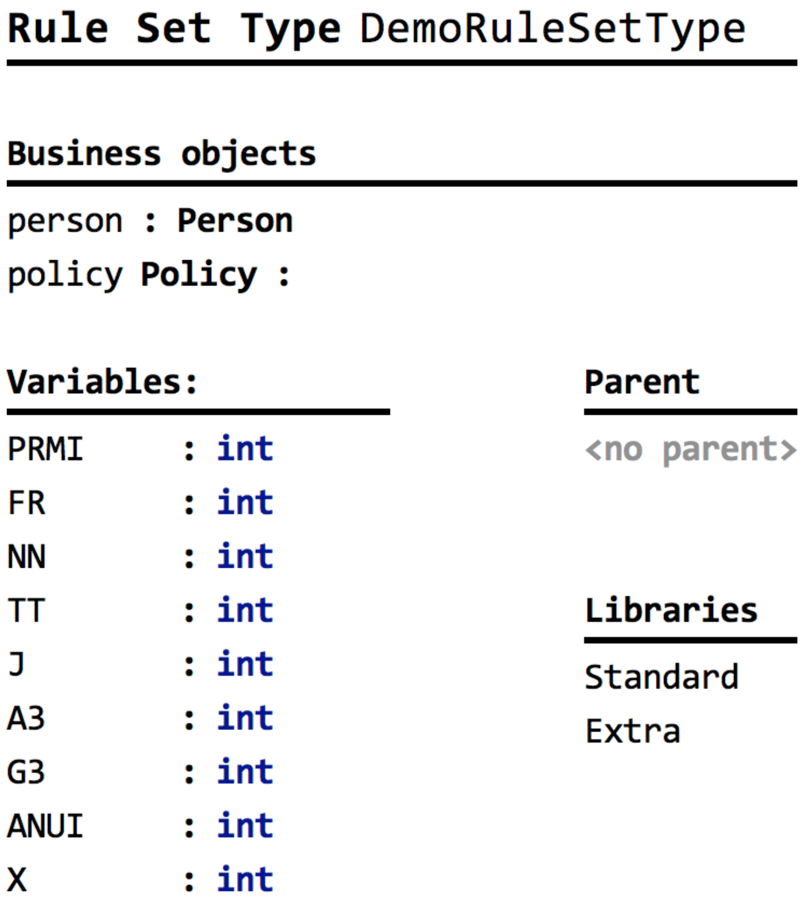
The actual **#actMinPrice** is computed from the **#baseMinPrice** with the following equation, the **#priceFactor** is determined by the table below: **#(actMinPrice** = baseMinPrice * priceFactor / 100**)**.

Error: type double is not a subtype of uint32

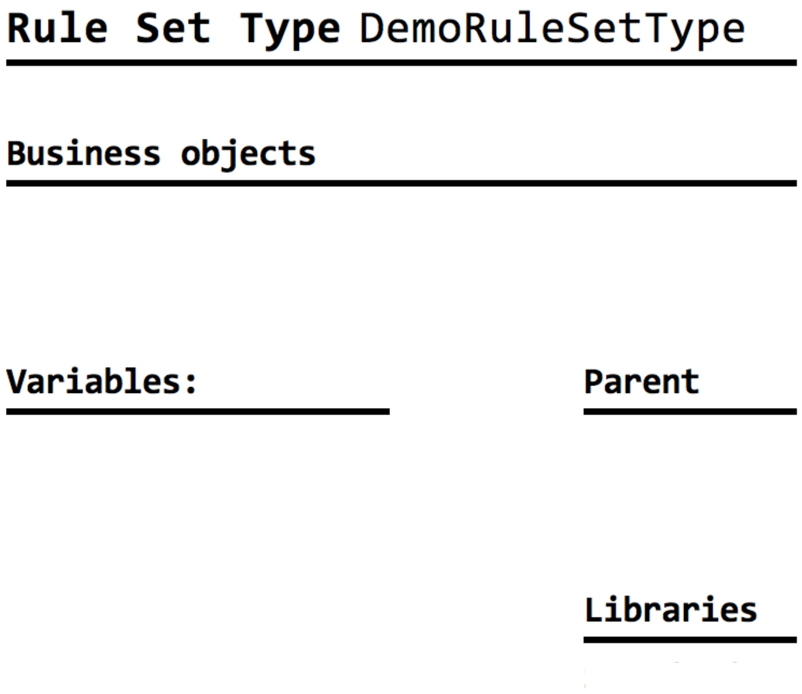
	Germany	Italy	Spain	GreatBritain
PLATINUM	10	8	7	11
GOLD	11	10	9	10
SILVER	12	8	8	8

LOBA: Where we are.

[Guided Editing] Form-Like



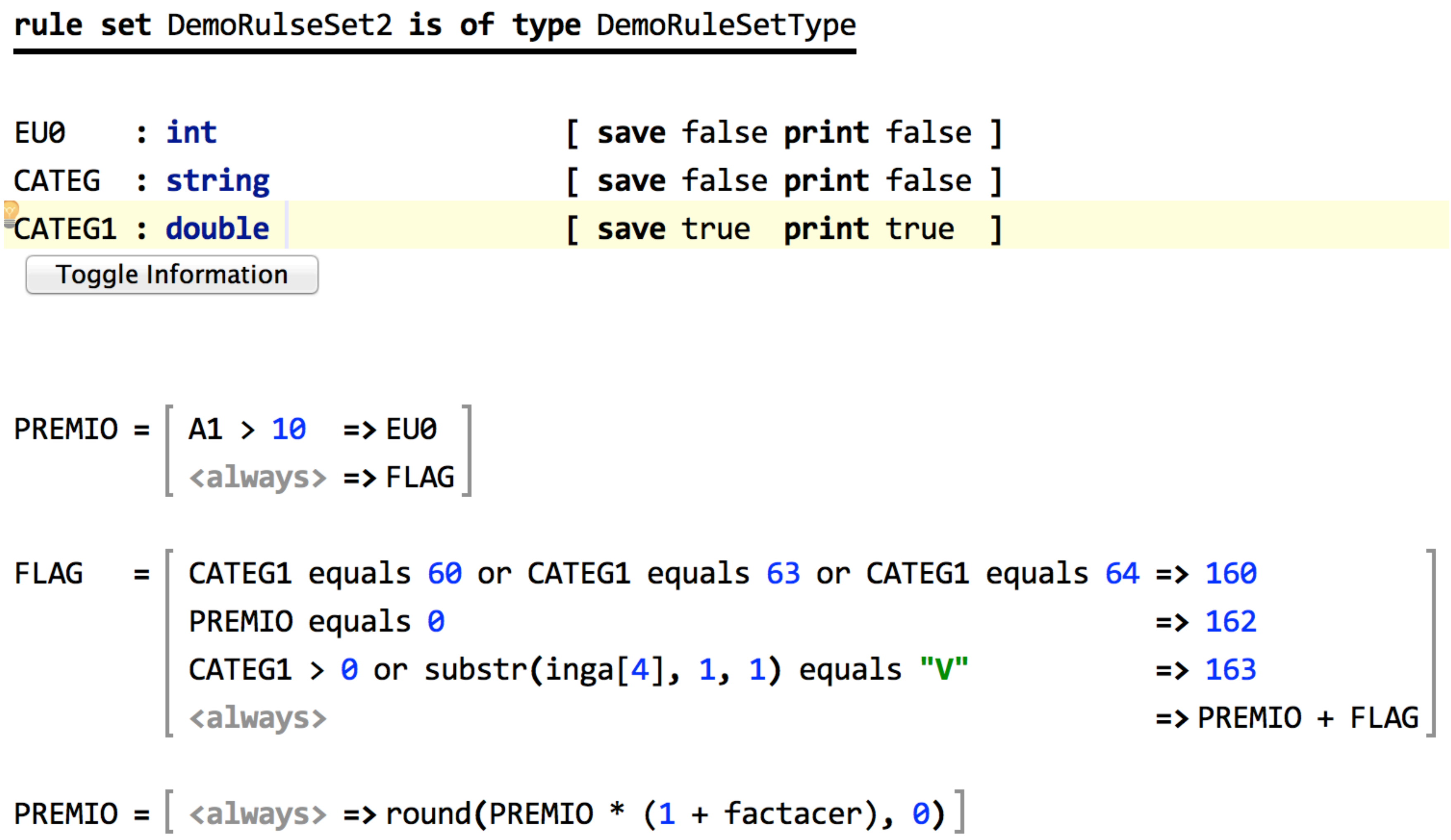
```
Rule Set Type DemoRuleSetType

Business objects
person : Person
policy Policy :

Variables:                       Parent
PRMI      : int                  <no parent>
FR        : int
NN        : int
TT        : int                  Libraries
J         : int                  Standard
A3        : int                  Extra
G3        : int
ANUI      : int
X         : int
```



```
Rule Set Type DemoRuleSetType

Business objects


Variables:                       Parent



                                 Libraries
```

LOBA: Where we are.

[Guided Editing] Form-Like II



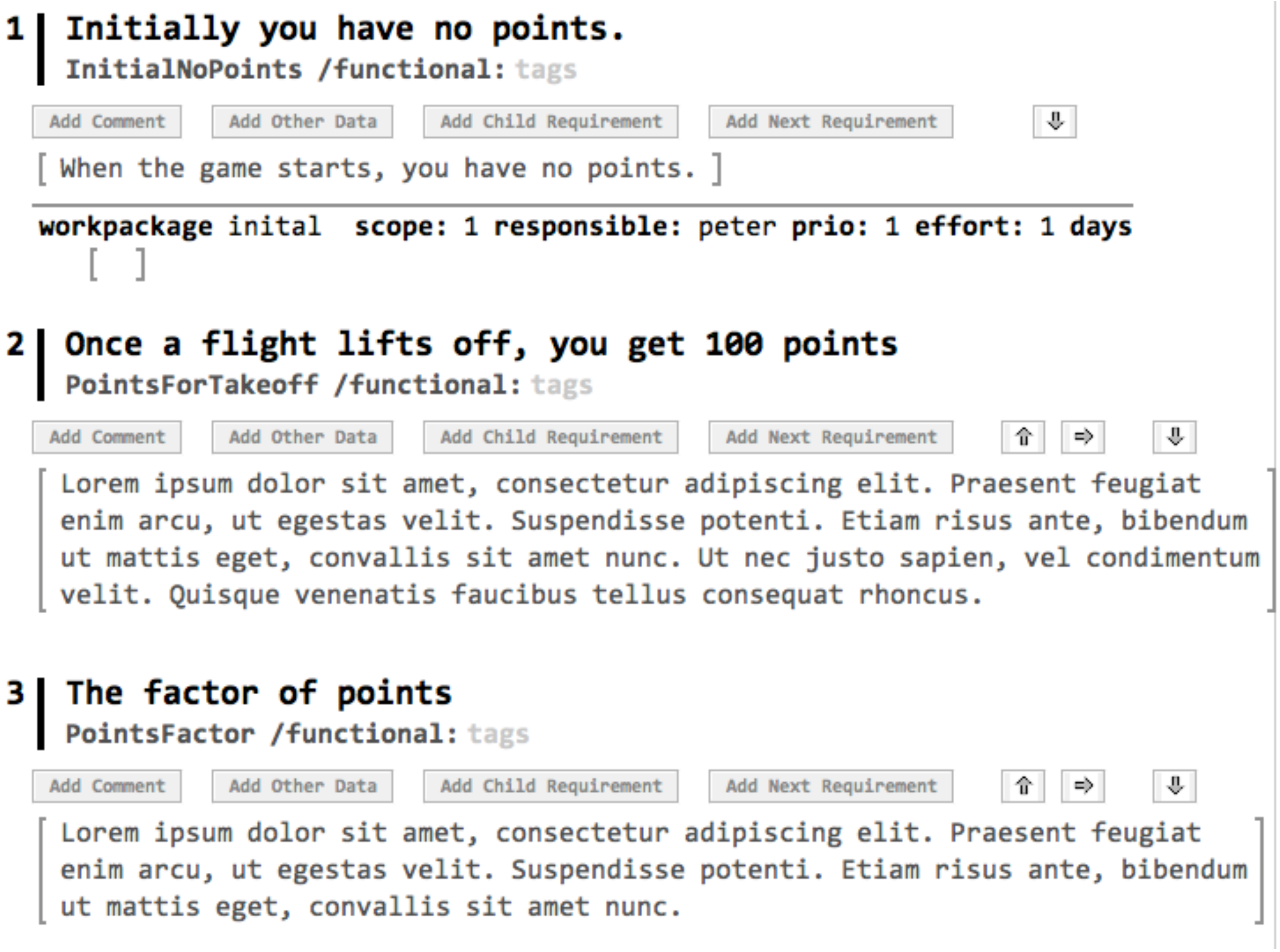
```
rule set DemoRulseSet2 is of type DemoRuleSetType

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

```
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) ]
```

LOBA: Where we are.

[Guided Editing] Editor Buttons

D



1 **Initially you have no points.**

InitialNoPoints /functional: tags

[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

[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.]

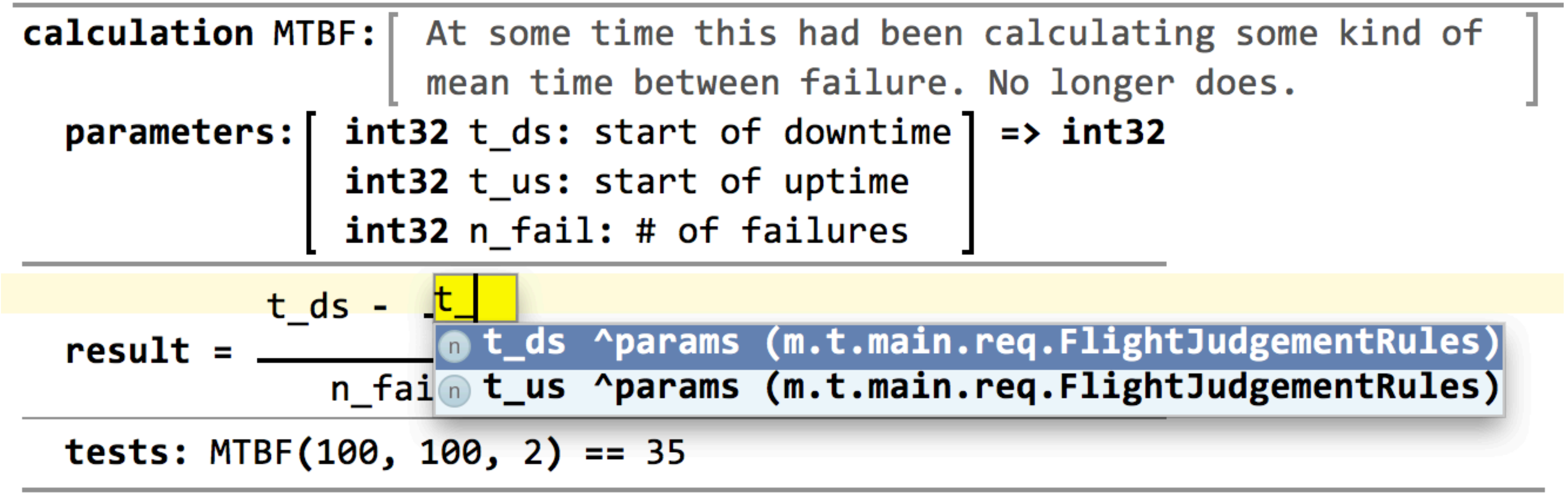
3 **The factor of points**

PointsFactor /functional: tags

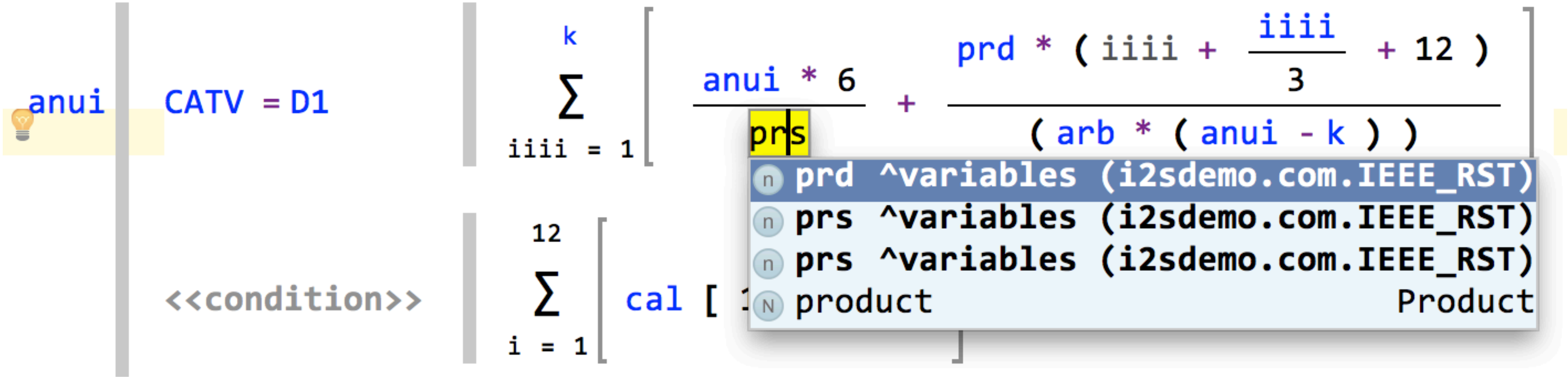
[Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent feugiat enim arcu, ut egestas velit. Suspendisse potenti. Etiam risus ante, bibendum ut mattis eget, convallis sit amet nunc.]

LOBA: Where we are.

[Guided Editing] Code Completion



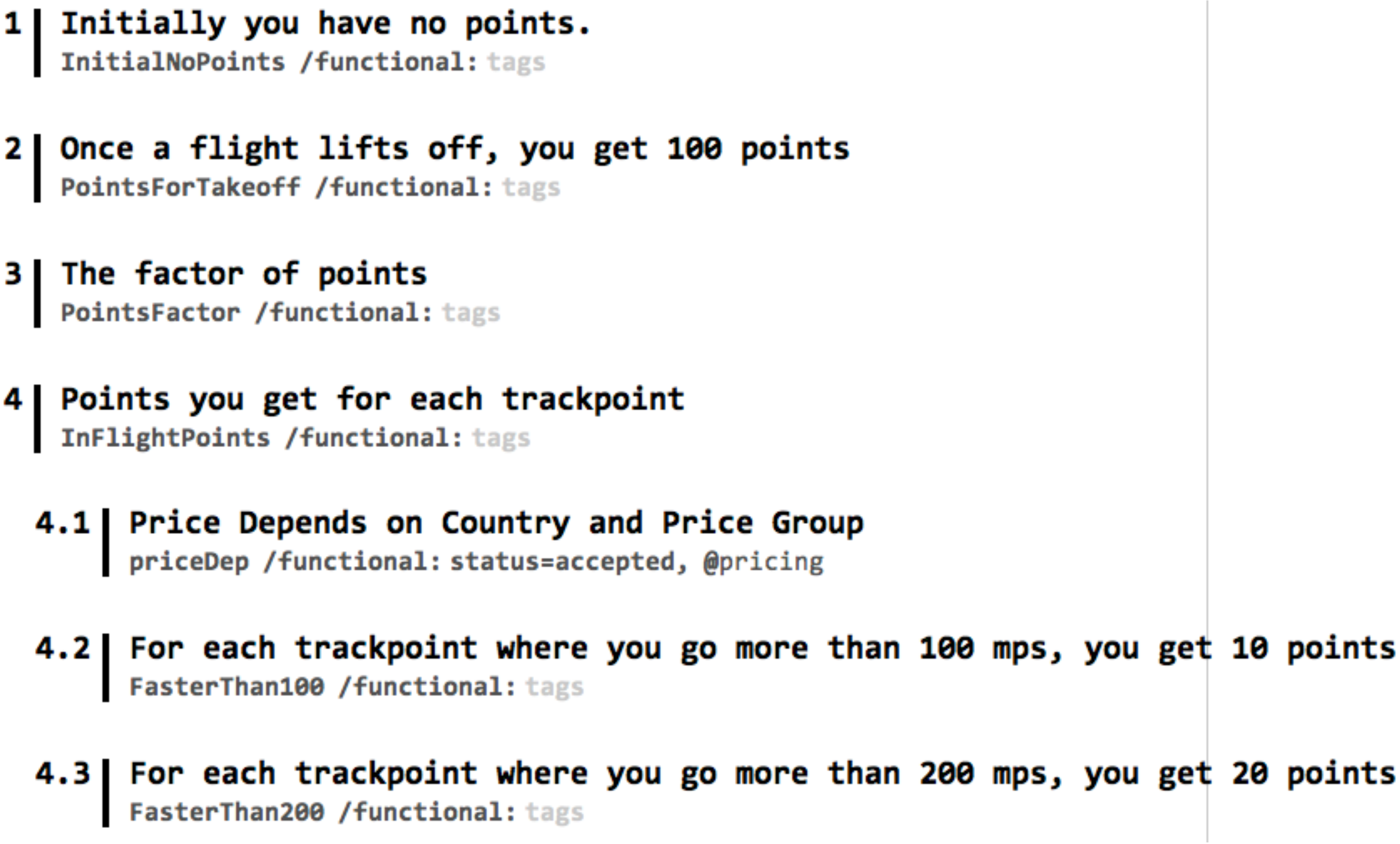
```
calculation MTBF: [ At some time this had been calculating some kind of
                    mean time between failure. No longer does.          ]
  parameters: [ int32 t_ds: start of downtime ] => int32
              [ int32 t_us: start of uptime   ]
              [ int32 n_fail: # of failures   ]

              t_ds - t_
  result = ---------------
              n_fai
                 t_ds ^params (m.t.main.req.FlightJudgementRules)
                 t_us ^params (m.t.main.req.FlightJudgementRules)

  tests: MTBF(100, 100, 2) == 35
```



```
anui | CATV = D1 |   k     [ anui * 6       prd * ( iiii + iiii/3 + 12 ) ]
     |           |   Σ     [ --------  +  ------------------------------ ]
     |           | iiii = 1[   prs            ( arb * ( anui - k ) )     ]
     |           |             prd ^variables (i2sdemo.com.IEEE_RST)
     |           |             prs ^variables (i2sdemo.com.IEEE_RST)
     | <<condition>> | 12      prs ^variables (i2sdemo.com.IEEE_RST)
     |           |   Σ   [ cal [ 1  product                   Product
     |           |  i = 1
```

Business Apps

[Context Aware] Different Projections



1 Initially you have no points.
InitialNoPoints /functional: tags

2 Once a flight lifts off, you get 100 points
PointsForTakeoff /functional: tags

3 The factor of points
PointsFactor /functional: tags

4 Points you get for each trackpoint
InFlightPoints /functional: tags

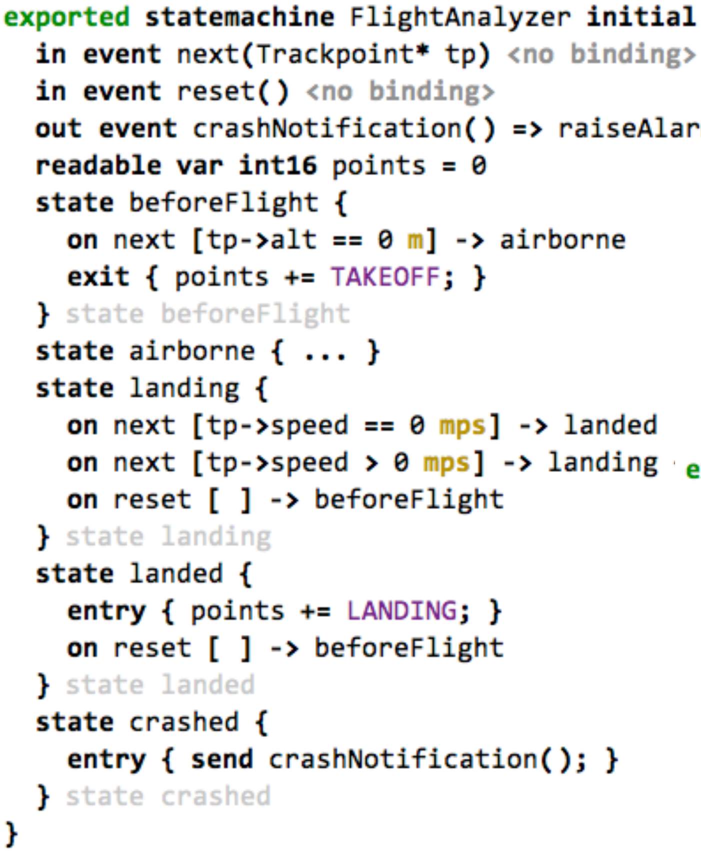
4.1 Price Depends on Country and Price Group
priceDep /functional: status=accepted, @pricing

4.2 For each trackpoint where you go more than 100 mps, you get 10 points
FasterThan100 /functional: tags

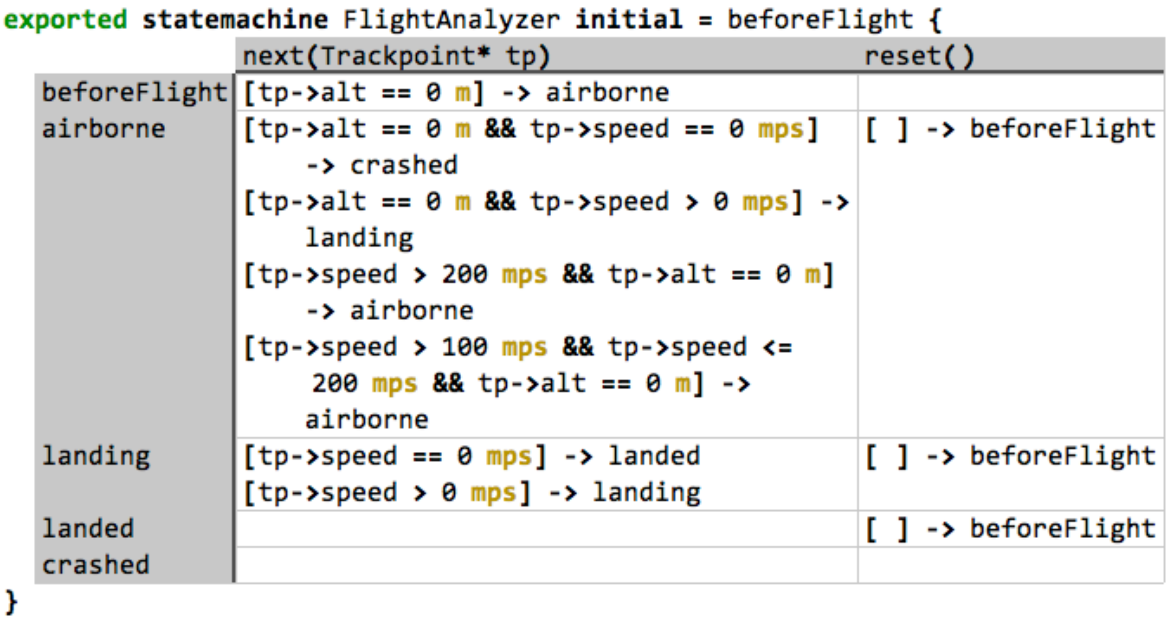
4.3 For each trackpoint where you go more than 200 mps, you get 20 points
FasterThan200 /functional: tags

Business Apps

[Context Aware] Different Projections



```
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 {
    on next [tp->alt == 0 m] -> airborne
    exit { points += TAKEOFF; }
  } state beforeFlight
  state airborne { ... }
  state landing {
    on next [tp->speed == 0 mps] -> landed
    on next [tp->speed > 0 mps] -> landing
    on reset [ ] -> beforeFlight
  } state landing
  state landed {
    entry { points += LANDING; }
    on reset [ ] -> beforeFlight
  } state landed
  state crashed {
    entry { send crashNotification(); }
  } state crashed
}
```



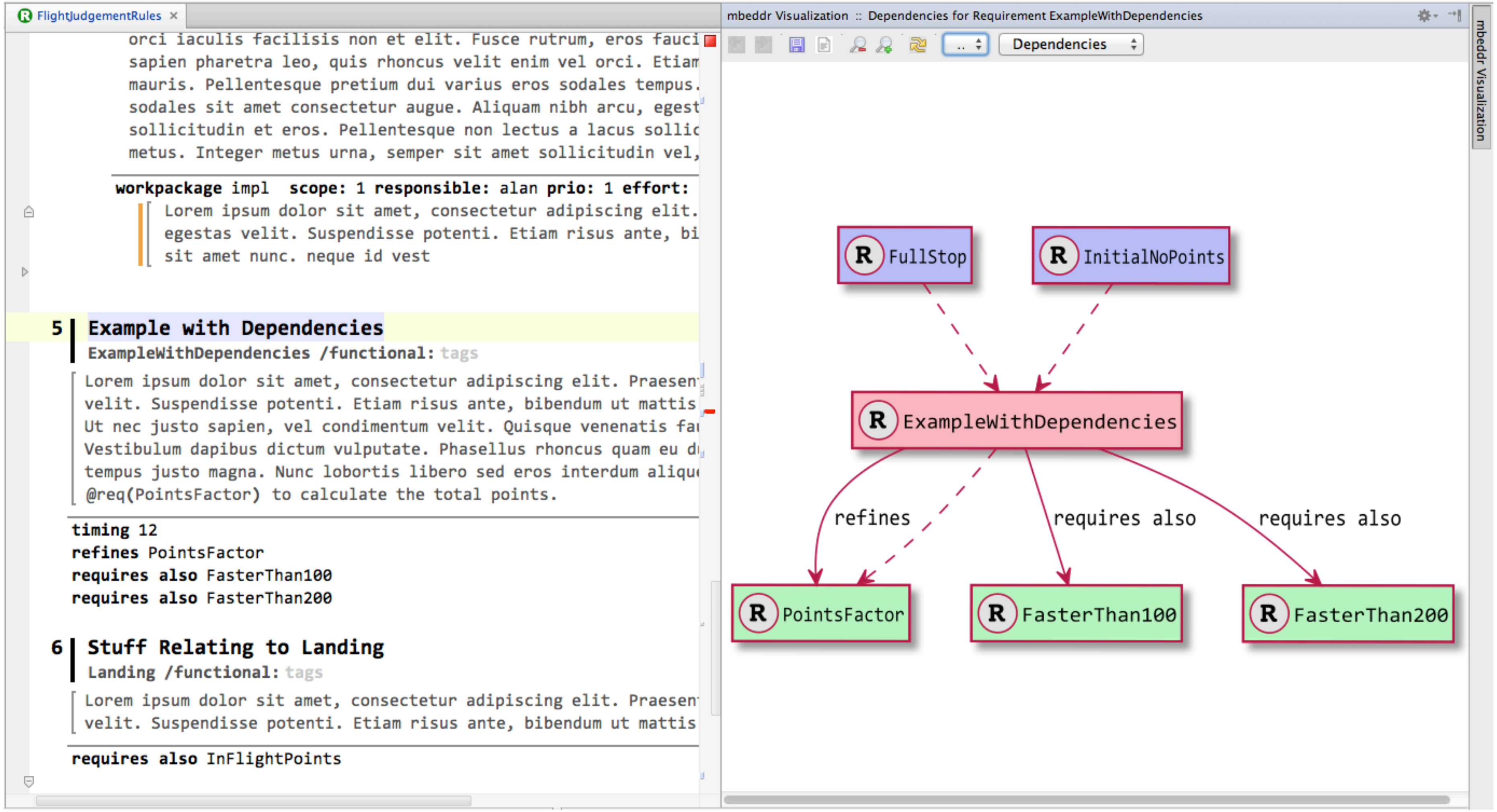
```
exported statemachine FlightAnalyzer initial = beforeFlight {
```

	next(Trackpoint* tp)	reset()
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 [tp->speed > 100 mps && tp->speed <= 200 mps && tp->alt == 0 m] -> airborne	[ ] -> beforeFlight
landing	[tp->speed == 0 mps] -> landed [tp->speed > 0 mps] -> landing	[ ] -> beforeFlight
landed		[ ] -> beforeFlight
crashed		

```
}
```

LOBA: Where we are.

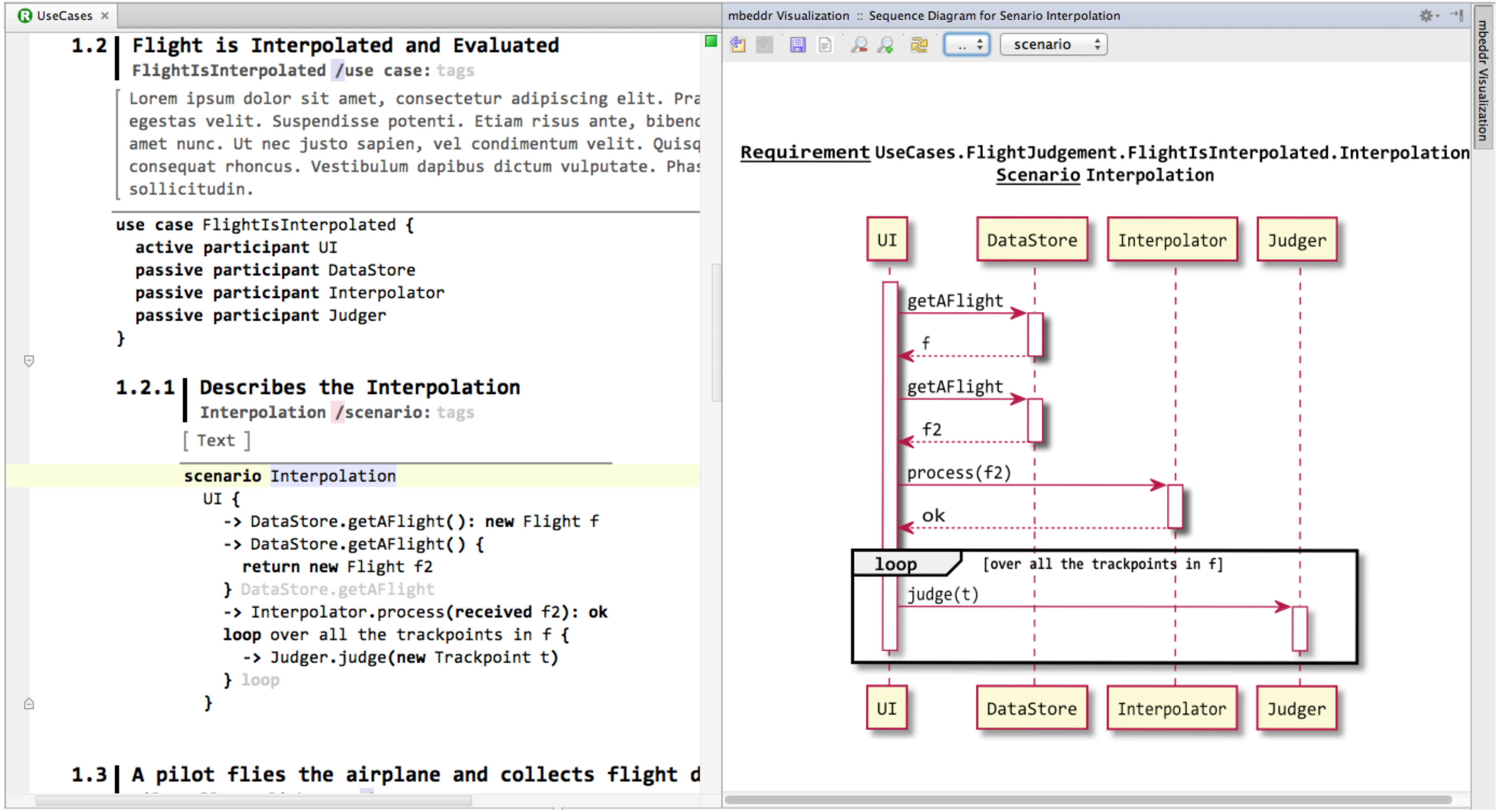
[Context Aware] Visualization



```
orci iaculis facilisis non et elit. Fusce rutrum, eros fauci
sapien pharetra leo, quis rhoncus velit enim vel orci. Etiam
mauris. Pellentesque pretium dui varius eros sodales tempus.
sodales sit amet consectetur augue. Aliquam nibh arcu, egest
sollicitudin et eros. Pellentesque non lectus a lacus sollic
metus. Integer metus urna, semper sit amet sollicitudin vel,

workpackage impl  scope: 1 responsible: alan prio: 1 effort:
   Lorem ipsum dolor sit amet, consectetur adipiscing elit.
   egestas velit. Suspendisse potenti. Etiam risus ante, bi
   sit amet nunc. neque id vest

5 | Example with Dependencies
    ExampleWithDependencies /functional: tags
   Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesen
   velit. Suspendisse potenti. Etiam risus ante, bibendum ut mattis
   Ut nec justo sapien, vel condimentum velit. Quisque venenatis fa
   Vestibulum dapibus dictum vulputate. Phasellus rhoncus quam eu d
   tempus justo magna. Nunc lobortis libero sed eros interdum aliqu
   @req(PointsFactor) to calculate the total points.
  timing 12
  refines PointsFactor
  requires also FasterThan100
  requires also FasterThan200

6 | Stuff Relating to Landing
    Landing /functional: tags
   Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesen
   velit. Suspendisse potenti. Etiam risus ante, bibendum ut mattis
  requires also InFlightPoints
```

LOBA: Where we are.

[Context Aware] Visualization II



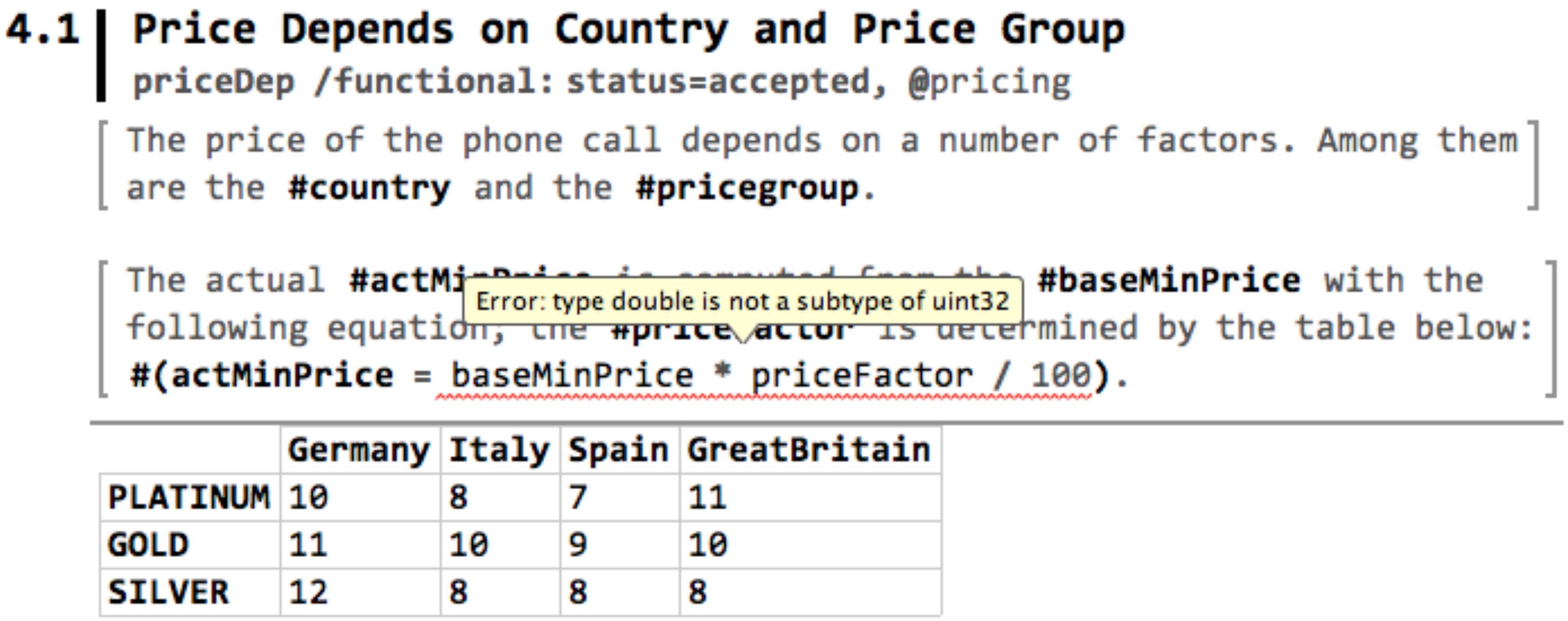
```
1.2 Flight is Interpolated and Evaluated
    FlightIsInterpolated /use case: tags
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pra
    egestas velit. Suspendisse potenti. Etiam risus ante, bibenc
    amet nunc. Ut nec justo sapien, vel condimentum velit. Quisq
    consequat rhoncus. Vestibulum dapibus dictum vulputate. Phas
    sollicitudin.
    use case FlightIsInterpolated {
      active participant UI
      passive participant DataStore
      passive participant Interpolator
      passive participant Judger
    }

    1.2.1 Describes the Interpolation
          Interpolation /scenario: tags
          [ Text ]
          scenario Interpolation
            UI {
              -> DataStore.getAFlight(): new Flight f
              -> DataStore.getAFlight() {
                return new Flight f2
              } DataStore.getAFlight
              -> Interpolator.process(received f2): ok
              loop over all the trackpoints in f {
                -> Judger.judge(new Trackpoint t)
              } loop
            }

1.3 A pilot flies the airplane and collects flight d
```

mbeddr Visualization :: Sequence Diagram for Senario Interpolation

Requirement UseCases.FlightJudgement.FlightIsInterpolated.Interpolation
Scenario Interpolation

LOBA: Where we are.

[Live Code] Error Checking



4.1 Price Depends on Country and Price Group

priceDep /functional: status=accepted, @pricing

The price of the phone call depends on a number of factors. Among them are the **#country** and the **#pricegroup**.

The actual **#actMinPrice** is computed from the **#baseMinPrice** with the following equation, the **#priceFactor** is determined by the table below: **#(actMinPrice** = baseMinPrice * priceFactor / 100**)**.

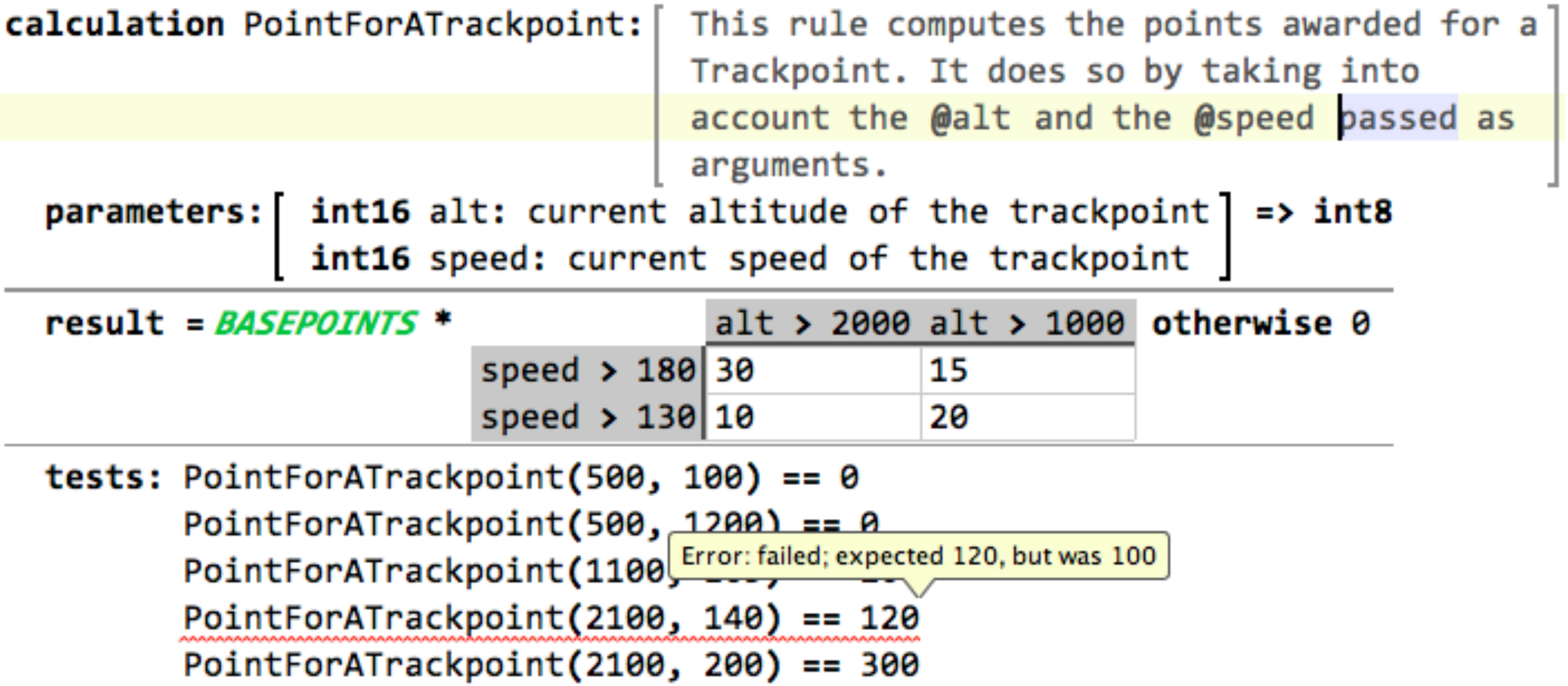
Error: type double is not a subtype of uint32

	Germany	Italy	Spain	GreatBritain
PLATINUM	10	8	7	11
GOLD	11	10	9	10
SILVER	12	8	8	8

LOBA: Where we are.

[Live Code] Interpreted Tests

D



```
calculation PointForATrackpoint: [ This rule computes the points awarded for a  ]
                                 [ Trackpoint. It does so by taking into        ]
                                 [ account the @alt and the @speed passed as    ]
                                 [ arguments.                                   ]
  parameters: [ int16 alt: current altitude of the trackpoint ] => int8
              [ int16 speed: current speed of the trackpoint  ]
```

result = *BASEPOINTS* * (decision table) otherwise 0

	alt > 2000	alt > 1000
speed > 180	30	15
speed > 130	10	20

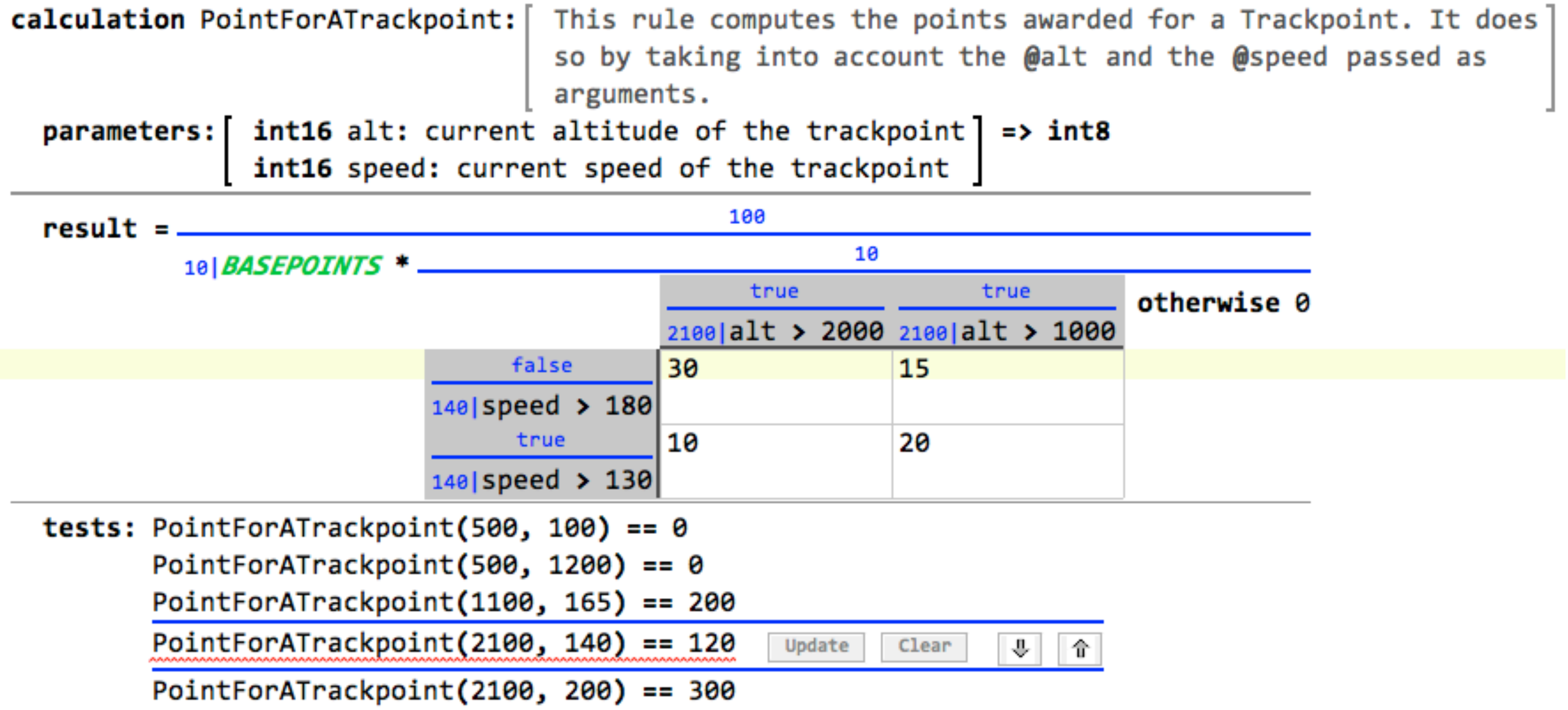
```
  tests: PointForATrackpoint(500, 100) == 0
         PointForATrackpoint(500, 1200) == 0
         PointForATrackpoint(1100,
         PointForATrackpoint(2100, 140) == 120
         PointForATrackpoint(2100, 200) == 300
```

Error: failed; expected 120, but was 100

LOBA: Where we are.

[Live Code] Debugging

D



```
calculation PointForATrackpoint: [ This rule computes the points awarded for a Trackpoint. It does ]
                                 [ so by taking into account the @alt and the @speed passed as     ]
                                 [ arguments.                                                      ]
  parameters: [ int16 alt: current altitude of the trackpoint ] => int8
              [ int16 speed: current speed of the trackpoint  ]
-------------------------------------------------------------------------------
  result =                           100
           ____________________________________________________________________
             10|BASEPOINTS *                       10
                       ________________________________________________________
                                     |     true        |     true        | otherwise 0
                                     | 2100|alt > 2000 | 2100|alt > 1000 |
                   false             | 30              | 15              |
                   140|speed > 180   |                 |                 |
                   true              | 10              | 20              |
                   140|speed > 130   |                 |                 |
-------------------------------------------------------------------------------
  tests: PointForATrackpoint(500, 100) == 0
         PointForATrackpoint(500, 1200) == 0
         PointForATrackpoint(1100, 165) == 200
         PointForATrackpoint(2100, 140) == 120
         PointForATrackpoint(2100, 200) == 300
```

Language Workbenches

All the IDE Support We Expect

Goto Definition/Find Usages

Error Markup/Quick Fixes

Syntax Highlighting

Code Completion

Search/Replace

Refactoring

Debugging

Reporting

Visualization

Version Control

6

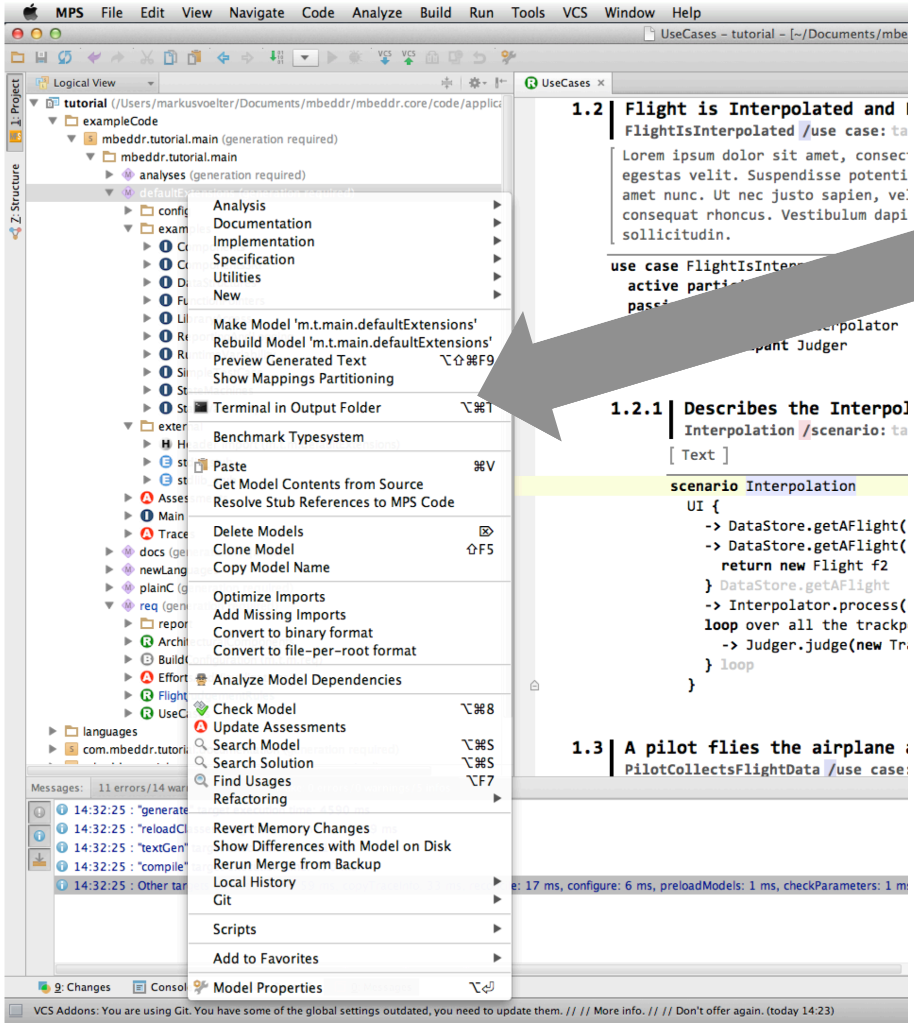
LOBA

why
where we are
what is missing

B

LOBA: What is missing/Challenges

Apparent Tool Complexity


MPS File Edit View Navigate Code Analyze Build Run Tools VCS Window Help
UseCases – tutorial – [~/Documents/mbe
Logical View
1: Project
Z: Structure
UseCases
tutorial (/Users/markusvoelter/Documents/mbeddr/mbeddr.core/code/applica
exampleCode
mbeddr.tutorial.main (generation required)
mbeddr.tutorial.main
analyses (generation required)
config
exam
exter
Asses
Main
Trace
docs
newLang
plainC
req
repor
Archi
Build
Effort
Flight
UseC
languages
com.mbeddr.tutori
Analysis
Documentation
Implementation
Specification
Utilities
New
Make Model 'm.t.main.defaultExtensions'
Rebuild Model 'm.t.main.defaultExtensions'
Preview Generated Text ⌥⇧⌘F9
Show Mappings Partitioning
Terminal in Output Folder ⌥⌘T
Benchmark Typesystem
Paste ⌘V
Get Model Contents from Source
Resolve Stub References to MPS Code
Delete Models ⌦
Clone Model ⇧F5
Copy Model Name
Optimize Imports
Add Missing Imports
Convert to binary format
Convert to file-per-root format
Analyze Model Dependencies
Check Model ⌥⌘8
Update Assessments
Search Model ⌥⌘S
Search Solution ⌥⌘S
Find Usages ⌥F7
Refactoring
Revert Memory Changes
Show Differences with Model on Disk
Rerun Merge from Backup
Local History
Git
Scripts
Add to Favorites
Model Properties ⌥⏎
Messages: 11 errors/14 war
9: Changes
Consol

```
1.2 | Flight is Interpolated and
      FlightIsInterpolated /use case: ta
    [ Lorem ipsum dolor sit amet, consec
      egestas velit. Suspendisse potenti
      amet nunc. Ut nec justo sapien, ve
      consequat rhoncus. Vestibulum dapi
      sollicitudin.

    use case FlightIsInter
      active parti
      pass
                          rpolator
                    pant Judger

1.2.1 | Describes the Interpol
        Interpolation /scenario: ta
      [ Text ]

      scenario Interpolation
        UI {
          -> DataStore.getAFlight(
          -> DataStore.getAFlight(
            return new Flight f2
          } DataStore.getAFlight
          -> Interpolator.process(
          loop over all the trackp
            -> Judger.judge(new Tr
          } loop
        }

1.3 | A pilot flies the airplane
      PilotCollectsFlightData /use case:
```

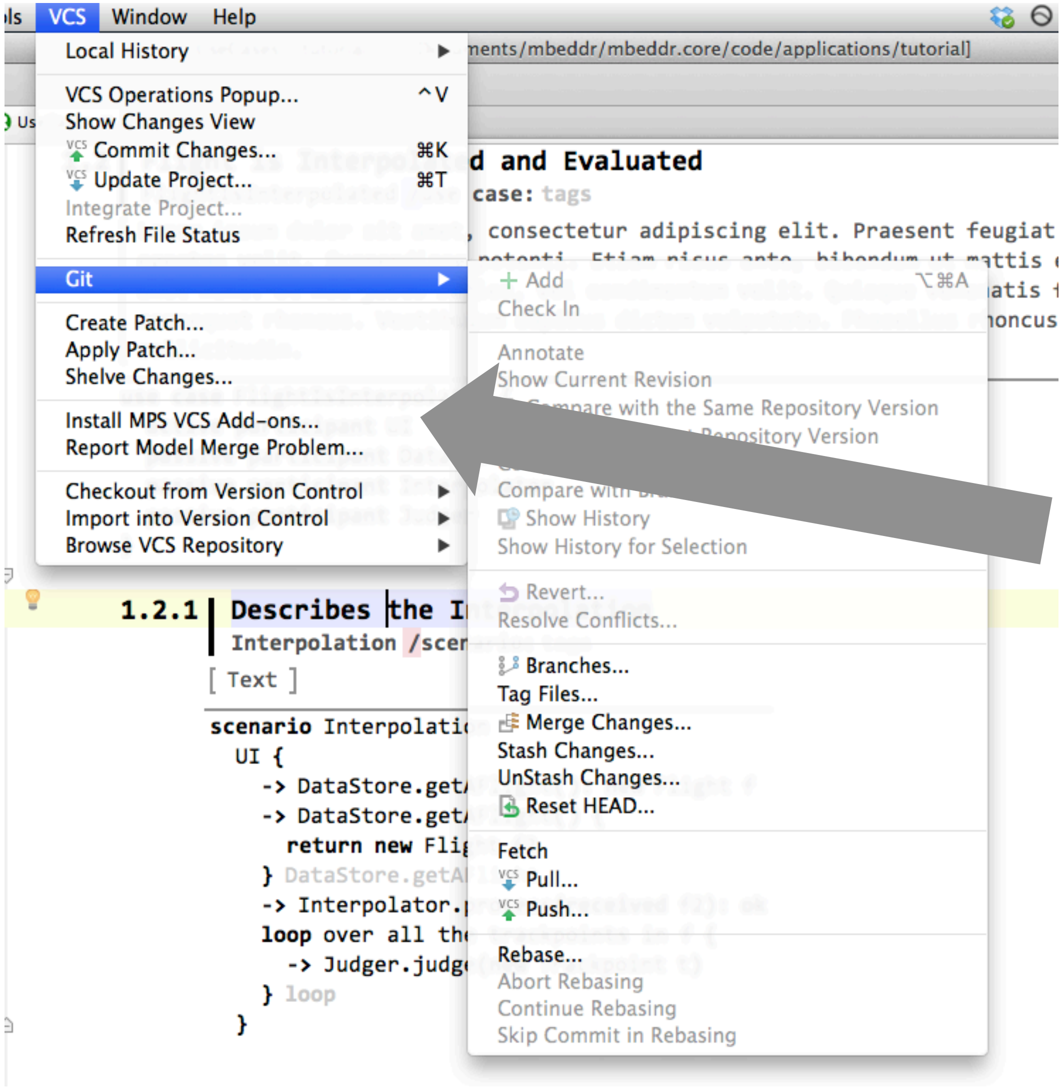
```
14:32:25 : "generate
14:32:25 : "reloadCl
14:32:25 : "textGen"
14:32:25 : "compile"
14:32:25 : Other tar                e: 17 ms, configure: 6 ms, preloadModels: 1 ms, checkParameters: 1 m
```

VCS Addons: You are using Git. You have some of the global settings outdated, you need to update them. // // More info. // // Don't offer again. (today 14:23)

Too many (too big) menus and buttons

LOBA: What is missing/Challenges

Need for Simplified Version Control



```
d and Evaluated
case: tags
, consectetur adipiscing elit. Praesent feugiat
potenti. Etiam risus ante, bibendum ut mattis
atis
oncus

1.2.1 | Describes the I
        Interpolation /scen
      [ Text ]

      scenario Interpolati
        UI {
          -> DataStore.get
          -> DataStore.get
            return new Fli
        } DataStore.getA
        -> Interpolator.
        loop over all the
          -> Judger.judge
        } loop
      }
```

Too many options.

Locking?

Realtime?

LOBA: What is missing/Challenges

Some Shortcomings in MPS

Cross-model generation

Projection of computed collections

Better Graphical Editing

Type System Performance

Some Editor Usability

Adressed by JetBrains in 2014.

LOBA: What is missing/Challenges

Training

Users may not be used to this approach.

Training is important.

Productivity more useful than Learnability.

LOBA: What is missing/Challenges

SE Best Practices

Modularity, Reuse, Injeritance, ...

Users may not know about these things, but they may still be necessary for efficiency reasons.

7

Conclusions

B

Conclusions

Applications hide Languages

Limited Tool Support for them

LWBs are useful alternative

Connect Business & IT

Language? Form? IDE? Application?

Towards Language-Oriented Business Apps

THE END.

Markus Voelter
independent/itemis
voelter@acm.org
www.voelter.de
@markusvoelter

Bernd Kolb
itemis
kolb@itemis.de
www.itemis.de
@berndkolb