Trends in Programming Languages

ICALEPCS 2011

Markus Voelter
Independent/itemis
voelter@acm.org

A single language to rule them all

An ecosystem of languages

An ecosystem of languages

The Pendulum Swings

An ecosystem of languages

The need for Concurreny

An ecosystem of languages

The need for Productivity

An ecosystem of languages

Separation Platform - Language

An ecosystem of languages

Building Languages is easier.

❶ Interesting GPL Features

Type Inference

Let the compiler figure out types.

Classic Java

```
Map<String, MyType> m =
     new HashMap<String, MyType>();
```

Classic Java

```
Map<String, MyType> m =
     new HashMap<String, MyType>();
```

Scala

```
var m = new HashMap[String, MyType]();
```

Classic Java

```
Map<String, MyType> m =
     new HashMap<String, MyType>();
```

Scala

```
var m = new HashMap[String, MyType]();
```

C# + LINQ

```
Address[] addresses = …

var res = from a in addresses
          select new { name = a.name(),
                       tel = a.telNo()};

foreach (var r in res) {
  Console.WriteLine("Name: {0}, Num: {1}",
     r.name, r.tel);
}
```

Functions

Focus on verbs instead of nouns (objects)

Ruby

```
[1,2,3,4,5,6].each {
      |element| puts (element * 2) }
```

Scala

```
x: Int => x + 1
```

Scala

```
def apply(f: Int => Int, v: Int) => f(v)
```

Pattern Matching

Easily
deconstruct
data structures

F#

```
> type Expr =
  | Op of string * Expr * Expr
  | Var of string
  | Const of int;;

> let rec eval x = match x with
    | Op(op, l, r) ->
          let (lv, rv) = (eval l, eval r)
      if (op = "+") then lv + rv
      elif (op = "-") then lv - rv
      else failwith "Unknonw operator!"
    | Var(var) -> getFromSymbolTable var
    | Const(n) -> n;;
```

Transactional Memory

Declarative
Shared Memory
Concurrency

Similar to GC:

- > Rely on clever compiler and RT system
- > Solution might not always be optimal
- > … but good enough in 99% of cases
- > and much less (error prone) work.

Fortress

```
atomic do
  // the stuff here is executed as if
  // there was only this thread
end
```

Declarative++

Avoid saying things
you don‘t want to say

Java < 5

```
for ( int i=0; i < data.length; i++ ) {
  // do a computation with data[i]
}
```

Java >= 5

```
foreach ( DataStructure ds in data ) {
  // do something with ds
}
```

Java < 5

```
for ( int i=0; i < data.length; i++ ) {
  // do a computation with data[i]
}
```

Java >= 5

```
foreach ( DataStructure ds in data ) {
  // do something with ds
}
```

Fortress

```
for I <- 1:m, j <- 1:n do
  a[i,j] := b[i] c[j]
end
```

Fortress

```
for i <- seq(1:m) do
  for j <- seq(1:n) do
    print a[i,j]
  end
end
```

Message Passing

Shared Memory is BAD

(Joe Armstrong)

Erlang

```
pid = spawn(fun() -> doSomething() end)
Pid ! Message
```

Erlang

```
pid = spawn(fun() -> doSomething() end)
Pid ! Message
```

Erlang

```
loop
  receive
    {add, Id, Name, FirstName} ->
                    ActionsToAddInformation;
    {remove,Id} -> ActionsToRemoveItAgain;
    ...
    after Time -> TimeOutActions
end
```

❷ Domain Specific Languages

Definition

What is a DSL?

tailor made

effective++

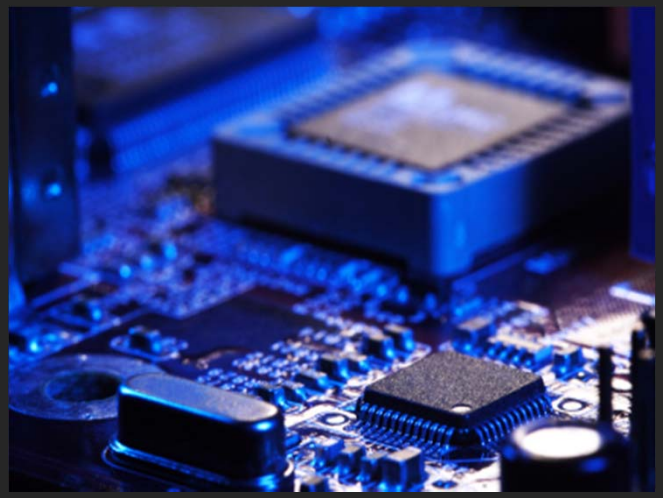
specialized, limited

used by experts

together with other specialized tools

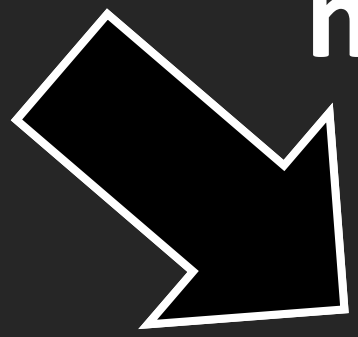
execute?

map



DSL Program
(aka Model)

map



automated!

GPL Program

Example DSLs

Stuff that I have
worked in in the past.

```
procedure writeRegisterNumberZ requestCode 0x29 {
    request: struct request1 {
        int8 acc pattern {
            2:b00;
            6:parentRequestCode;
        };
        int8 registerAddress;
    } ;
    reply: struct dontCareReply {
        int8 statusByte patternref statusByte;
        int8 dontCare patternref defaultReturn;
    } ;
    request: struct request2 {
        int8 registerType pattern {
            4:b0000;
            4:registerType;
        };
        int8 registerAddress;
        int8 registerdata [2];
    } ;
}
```

tests

```
test writeRegisterNumberZ for dip writeRegisterNumberZ {
    send request1 {
        attr registerAddress == reg parameterInstruction;
    };
    expect dontCareReply {
        subattr statusByte # standardStatus == 2;
    };
    send request2 {
        subattr registerType # registerType == 3;
        attr registerAddress == reg parameterInstruction;
        attr registerdata == 0x77;
        subattr registerdata # channelNumber == 5;
    };
}
```

refines

```
register parameterInstruction address 0x37 struct {
    int8 db1 pattern {
     2:b00;
     6:channelNumber;
    };
};
```

Capgemini Pension Workbench

File Edit Projection Navigation Search Format Tools Dev Generate Pension Team NN

NNLCPA-14w2-21112008 * ×

Table of Contents ×

- Library
  - Documentation
  - Foundation
    - Value sets
      - Value set Groottebepalingsmethode
        - 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

All

¤ 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¶

Section ▸ title ▸ Paragraph : Text Dev

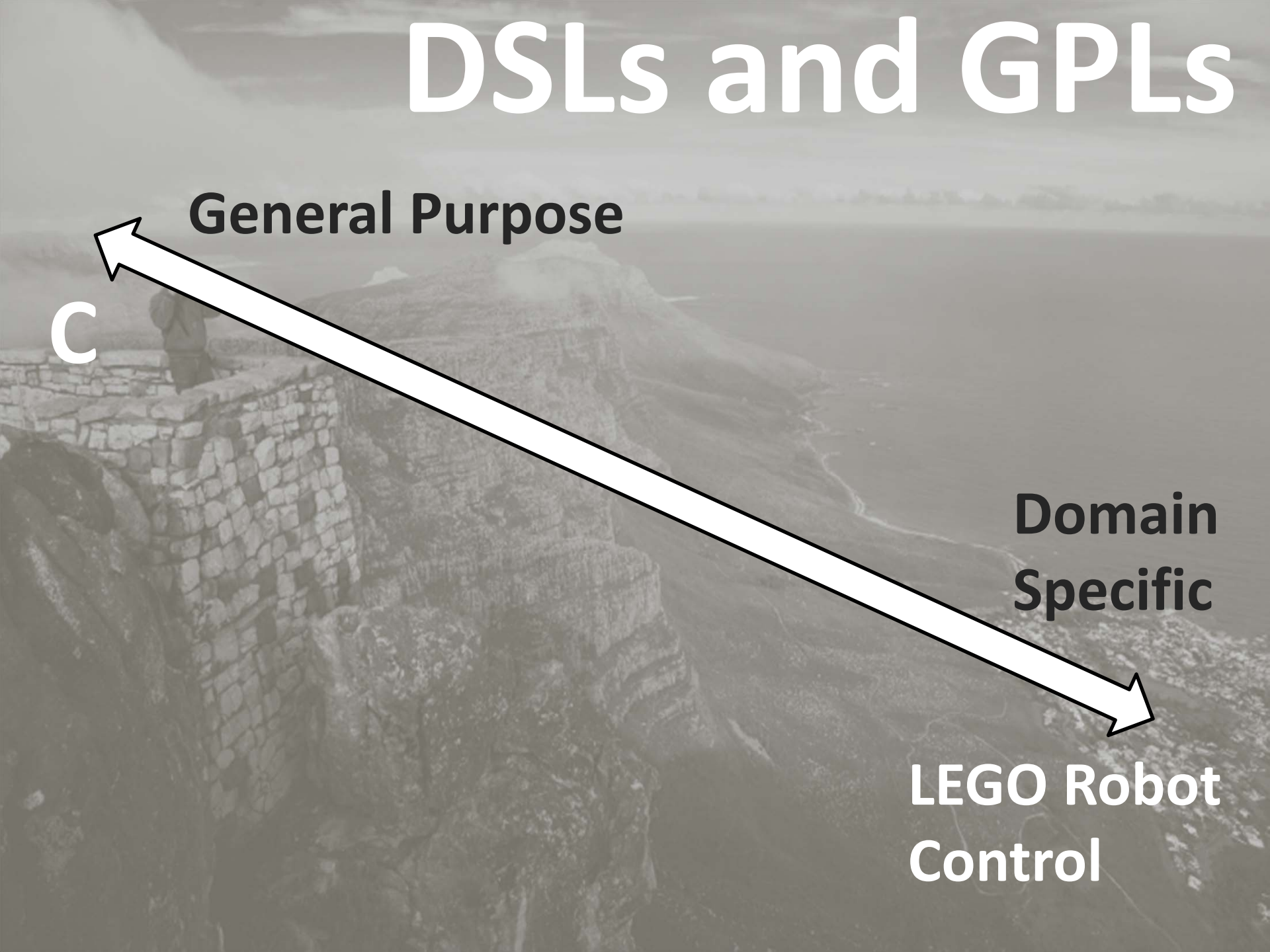
Doc | Splitter | Pension | PensionDecorated | AM

```
pumping program P1 for AtLeastOneZone + WithAlarm +
                        SuperPowerCompartment[f=comp1]  {
    parameter defaultWaterLevel : int
    parameter superWaterLevel: int
    event superPowerTimeout

    init {
        set comp1->targetHeight = defaultWaterLevel
    }

    start:
        on (comp1->needsPower == true) && !(comp1->isPumping) {
            do comp1->pumpOn
        }
        on comp1->enough {
            do comp1->pumpOff
        }
        on comp1.superPumping->turnedOn {
            set comp1->targetHeight = superWaterLevel
            raise event superPowerTimeout after 20
        }
        on comp1.superPumping->turnedOff or superPowerTimeout {
            set comp1->targetHeight = defaultWaterLevel
        }
}
```

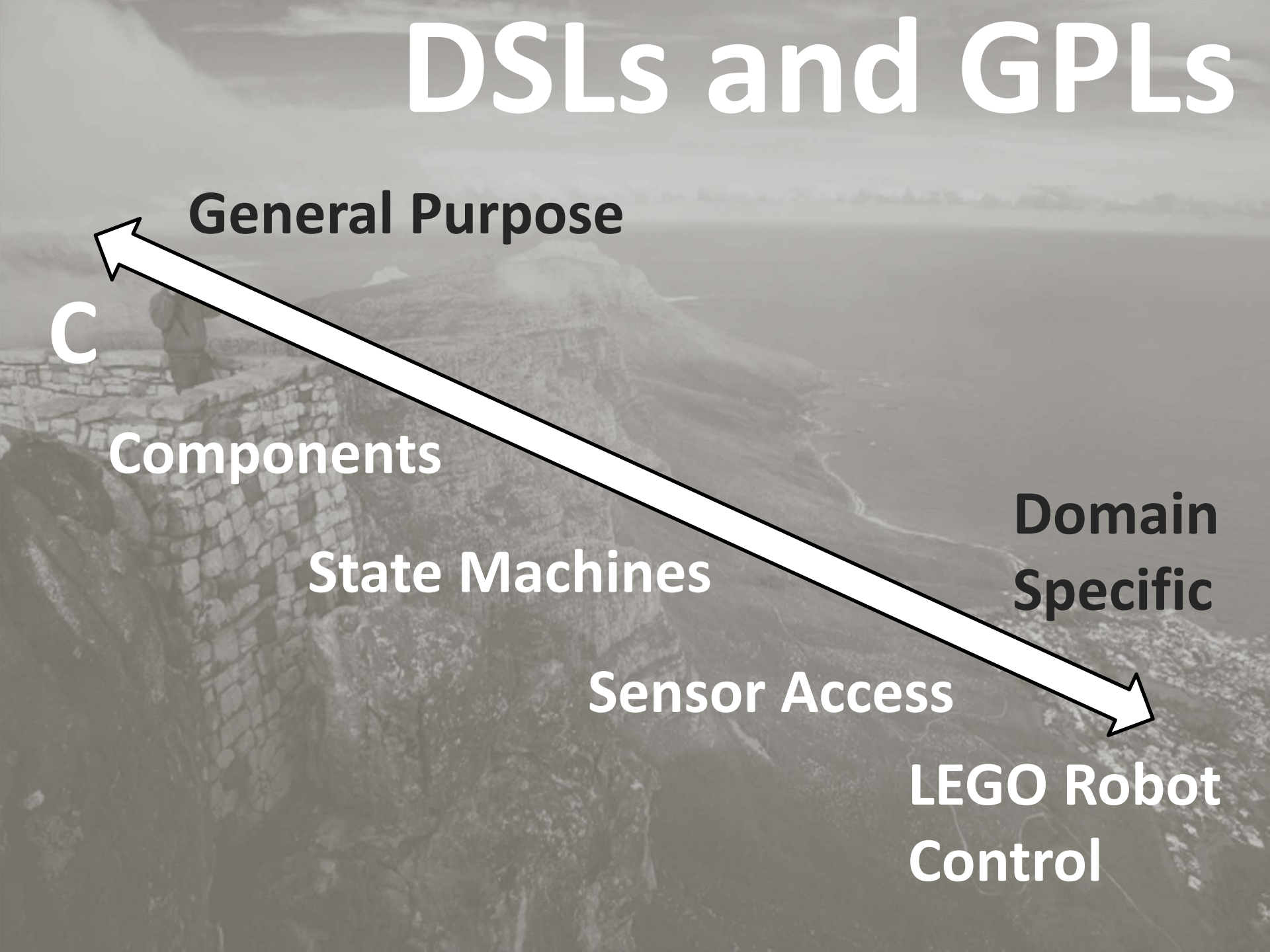
DSLs and GPLs

How can DSLs effectively work together with GPLs?

DSLs and GPLs



DSLs and GPLs

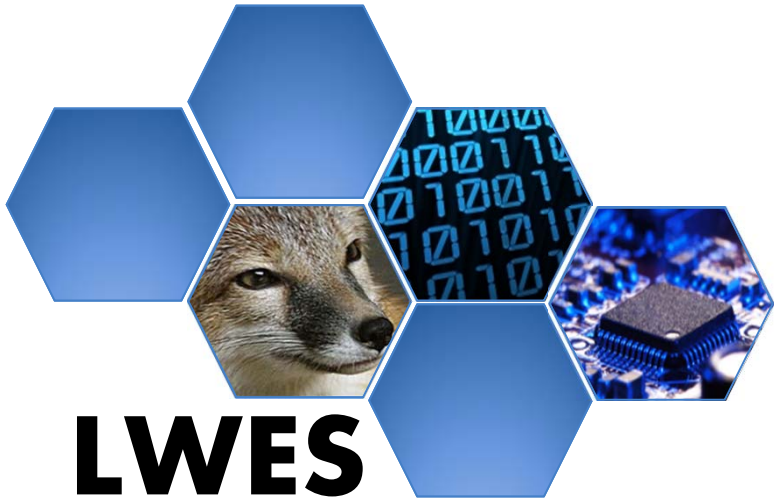


itemis

fortiss
innovation in software and systems

SICK
Sensor Intelligence.

LEAR CORPORATION



LWES
Language Workbenches *for* Embedded Systems

Bundesministerium für Bildung und Forschung

http://mbeddr.com

Incremental Extension of C with DSLs for Embedded Systems, integrated with Formal Methods and support for PLE and Requirements Tracing.

mbeddr.com

Home | Overview | Publications | MPS | Showcase | Code | Team

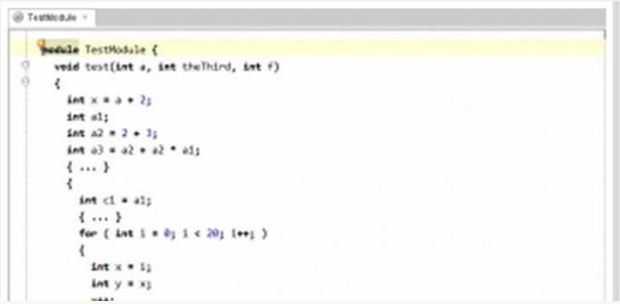
Search

Feeds: Posts Comments

First C Code working

July 17, 2011 by mpscmed

As you may know, our project relies on the idea of extending the C programming language with domain specific extensions. For that to work, we first have to make C available in MPS. While we had done this to some extent in our proof of concept, we are now implementing C much more thoroughly. As you can see in the screenshot below, some essential things are already working.



Archives

- July 2011 (3)
- June 2011 (2)
- January 2011 (2)
- July 2010 (1)
- June 2010 (2)

Categories

- code (3)
- demos'n'stuff (4)
- dev progress (1)
- news (4)
- Uncategorized (2)

Pages

TestModule

```
module TestModule {
  void test(int a, int theThird, int f)
  {
    int x = a + 2;
    int a1;
    int a2 = 2 + 3;
    int a3 = a2 + a2 * a1;
    { ... }
    {
      int c1 = a1;
      { ... }
      for ( int i = 0; i < 20; i++; )
      {
        int x = i;
        int y = x;
        x++;
      }
    }
    int sum = add(12, 3);
  }

  int add(int a, int b) { ... }
}
```

```
module Tests imports nothing
{
  exported test case Adding {
    int x = 0;
    int y = 1;
    int s = add(x, y);
    assert(0) s == 1;
  } Adding(test case)

  int add(int a, int b) {
    return a + b;
  } add(function)

  int main() {
    logInfo(0) "Hello, World!";
    return test Adding;
  } main(function)
}
```

```
Build Configuration
  compiler: gcc
  compiler options: -std=c99

program HelloWorld isTest: true {
  HelloWorld
  Tests
}
```

```
module ClosureTests from test.ex.core.pointers imports ClosureUtils, stdlib {

  typedef (int)=>(int, int) as ftype;

  exported test case testClosures {
    assert(0) aHOF([a, b|a + b;]) == 7;
  } testClosures(test case)

  int aHOF(ftype fun) {
    return fun(4, 3);
  } aHOF (function)
}
```

```
module CompModule from test.ex.ext.components.comptest imports nothing {

  exported enum TLCommand { stop; go; }

  exported c/s interface ITrafficLights {
    int setColor(TLCommand cmd)
  }

  c/s interface IDriver {
    int setDriverValue(int addr, int value)
  }

  exported atomic component Driver extends <no baseComponent> {
    ports:
      provides optional IDriver cmd [0..1]
    contents:
      int setDriverValue(int addr, int value) <- opcall cmd.setDriverValue {
        return 0;
      }
  }
```

```
  exported atomic component TrafficLights extends <no baseComponent> {
    ports:
      requires mandatory IDriver driver [1..1]
      provides mandatory ITrafficLights tl [1..1]
    contents:
      int setColor(TLCommand cmd) <- opcall tl.setColor {
        int xx = 10;
        return driver.setDriverValue(1, 0);
      }
  }
}
```

```
module UtilsTest from test.ex.core.utils imports nothing {

  int filter(int x) {
    return gswitch int {
      case x == 0: 0
      case x < 10: 1
      case x < 20: 2
      default: 5
    };
  } filter (function)

  int decide(int x, int y) {
    return int, 0
```

	x == 0	x > 0
y == 0	0	1
y > 0	1	2

```
;
  } decide (function)

  exported test case TestDecTab {
    assert(0) decide(0, 0) == 0;
    assert(1) decide(0, 1) == 1;
    assert(2) decide(1, 0) == 1;
    assert(3) decide(1, 1) == 2;
  } TestDecTab(test case)

  exported test case TestGSwitch {
    assert(0) filter(0) == 0;
    assert(1) filter(2) == 1;
    assert(2) filter(15) == 2;
    assert(3) filter(42) == 5;
  } TestGSwitch(test case)

  int main(string[ ] args) {
    return test TestGSwitch, TestDecTab;
  } main (function)
```

Core Extensible C implementation will be Open Sourced in November 2011!

http://mbeddr.com

Tools

Which tools can you use to build your own DSLs?

xtext

MPS
Meta Programming System

INTENTIONAL®
SOFTWARE

Open Source (EPL)

Eclipse-based, Eclipse Project

Very flexible, very popular!

Current Version 2.0:

- **improved performance**
- **Xbase: expressions for reuse**
- **Xtend2: „Better Java“, with support for Xpand-like templates**

Open Source (Apache 2.0)

Projectional Editor

Very good at lang. Composition

Current Version 2.0:

- Improved performance
- Unified generate/compile/build
- Debug MPS in MPS
- Tables in the editor
- (Diagrams planned for 2.1)

Commercial Tool.

Projectional Editor

Very flexible notations

Version 1.8 is current

Way More:

Spoofax

Rascal

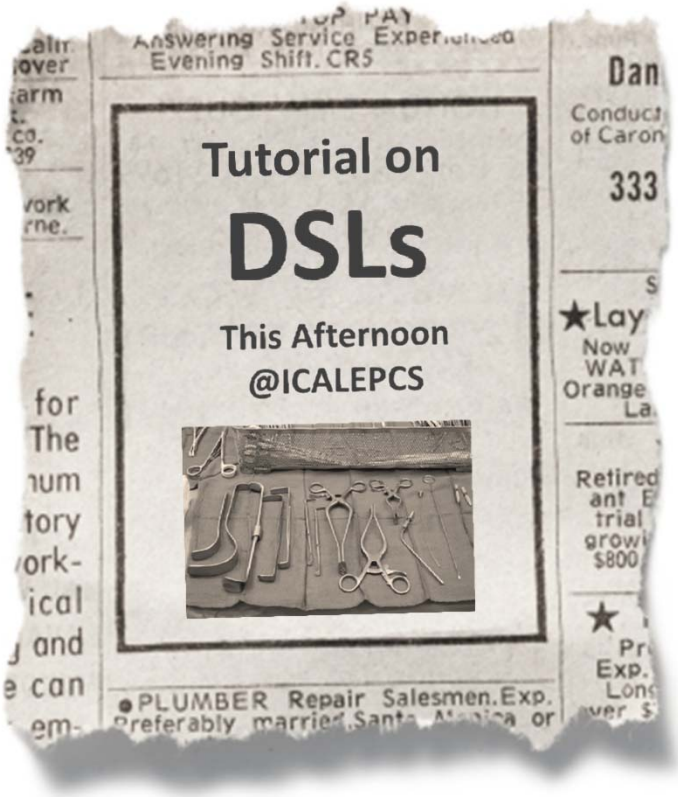
oomega

The Whole Platform

see also

http://languageworkbenches.net

(Commercial Break)



THE END.

.coordinates

web	www.voelter.de
email	voelter@acm.org
skype	schogglad
twitter	markusvoelter

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