



# Two Classes in Embedded Systems Development



(yes, this is a slight simplification)

**Plan Oriented  
Top Down  
Big Systems  
Big Companies**



**Modeling**

**Grown  
Bottom Up  
Small/Medium Systems  
Smaller Companies**



**C Code**

**2**

**Programming  
vs. Modeling**

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**

**3**

**Domain-Specific  
Languages**

## Domain Specific Languages

A DSL is a **focussed, processable language** for describing a specific **concern** when building a system in a specific **domain**. The **abstractions** and **notations** used are natural/suitable for the **stakeholders** who specify that particular concern.

## Domain Specific Languages

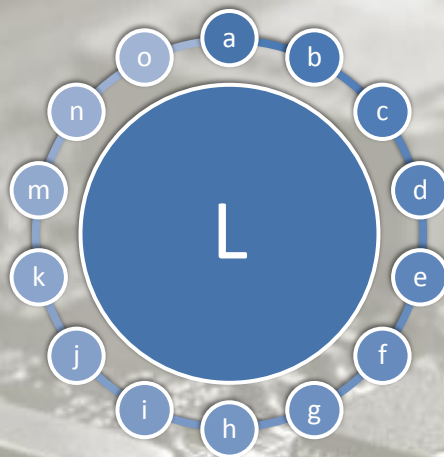
A DSL is a **language** at D that provides **linguistic abstractions** for **common patterns and idioms** of a language at D-1 when used within the domain D.

A **good** DSL does **not** require the use of patterns and idioms to express semantically interesting concepts in D. Processing tools do **not** have to do “semantic recovery” on D programs.

# 4

## Modular Languages

Big Language?



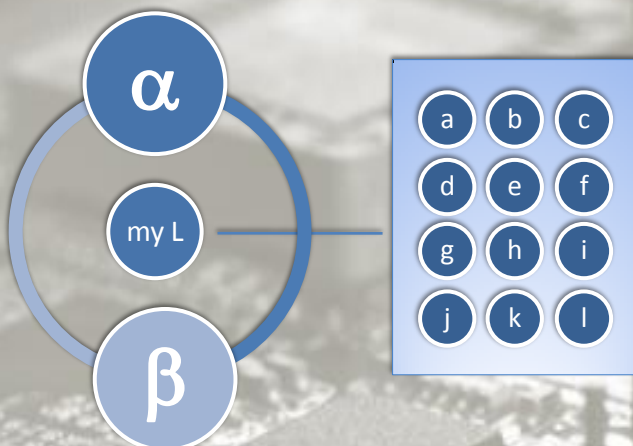
with **many** first class concepts!

## Small Language?



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

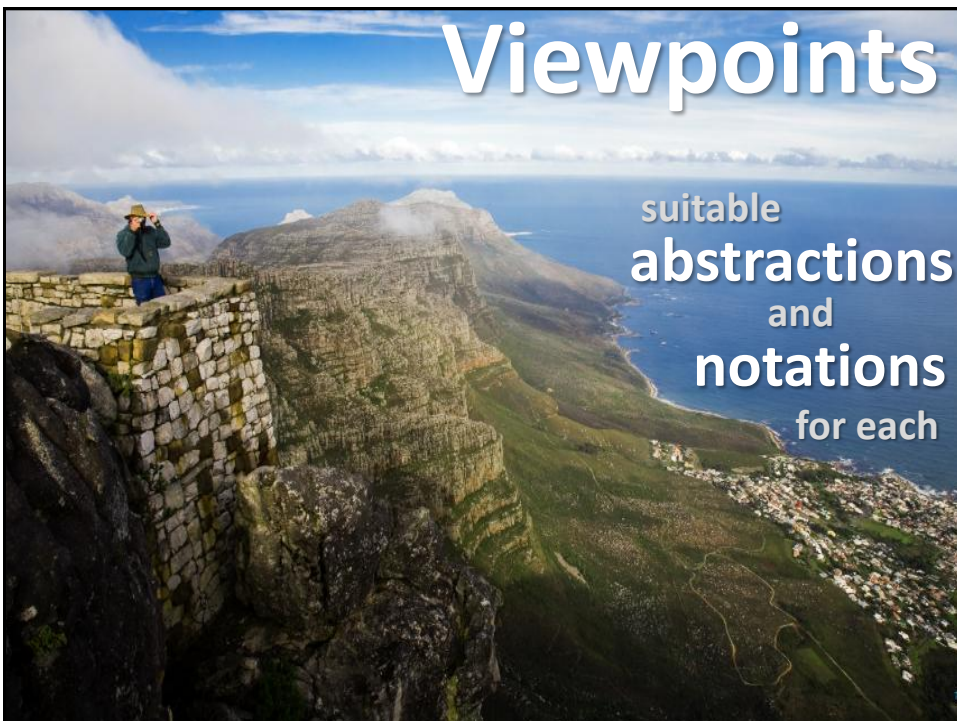
## Modular Language

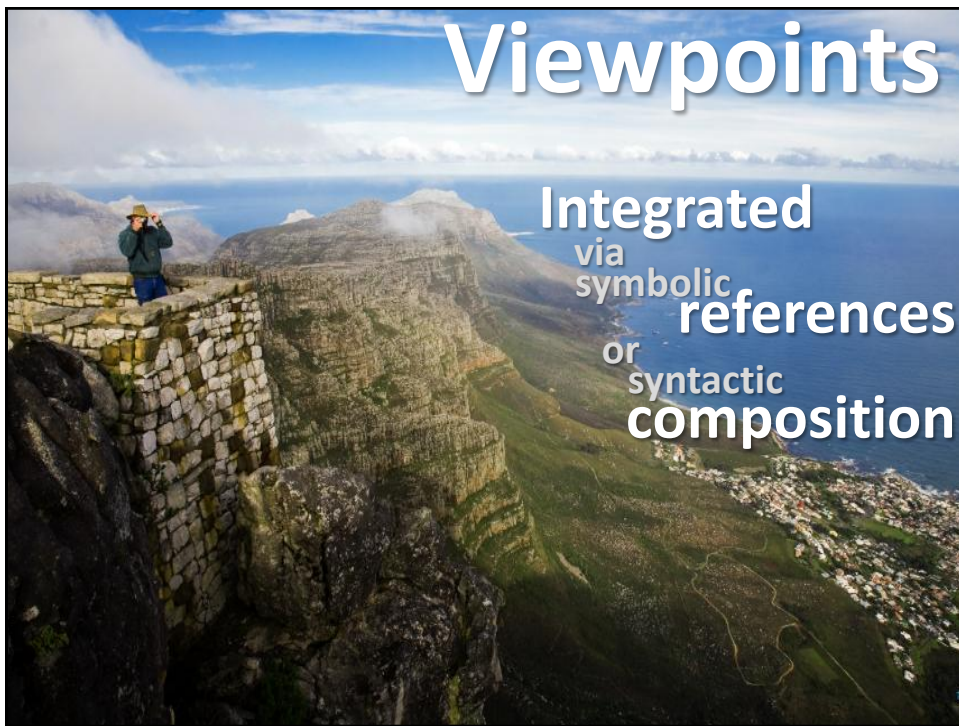


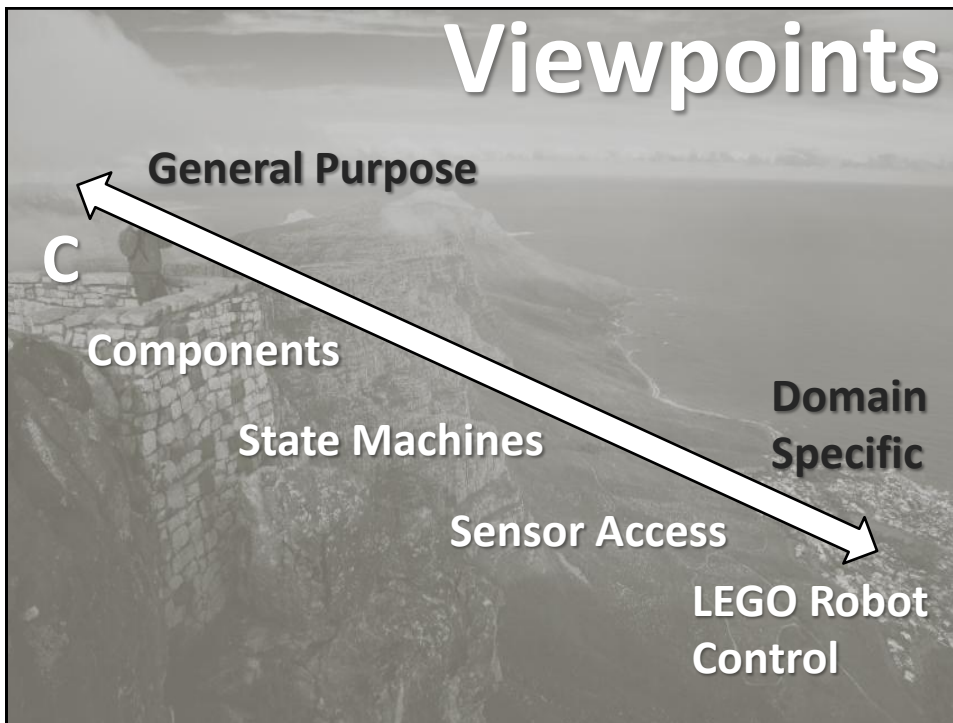
with many **optional,**  
**composable** concepts

# 5

## Concerns & Viewpoints







# 7

## The LWES Project



<http://mbeddr.com>



<http://mbeddr.com>

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



**LWES**  
Language Workbenches  
for Embedded Systems



Bundesministerium  
für Bildung  
und Forschung

<http://mbeddr.com>



Meta Programming System

JBRAINS



**LWES**  
Language Workbenches  
for Embedded Systems



Bundesministerium  
für Bildung  
und Forschung

<http://mbeddr.com>



innovation in software and systems



Sensor Intelligence.



Sensor Intelligence.



CORPORATION

# mbeddr.com

[Home](#)[Overview](#)[Publications](#)[MPS](#)[Showcase](#)[Code](#)[Team](#)

Feeds: Posts Comments

**LWES**  
Language Workbenches  
for Embedded Systems  
<http://mbeddr.com>

gefördert durch das Bundesministerium für Bildung und Forschung, Förderkennzeichen 01|S11014

## First C Code working

July 17, 2011 by mpscmcd

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.

```
void testMain() {  
    int x = 1;  
    int y = 2;  
    int z = x + y;  
    int w = z * 2;  
    int v = w / 2;  
    int u = v - 1;  
    int t = u + 1;  
    int s = t * 3;  
    int r = s / 3;  
    int q = r * 2;  
    int p = q / 2;  
}
```

### ARCHIVES

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

### CATEGORIES

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

### PAGES

# DEMO<sub>1</sub>

## Using the C Language

14

# 8

## Language Workbenches





# Language Workbench

(Martin Fowler)

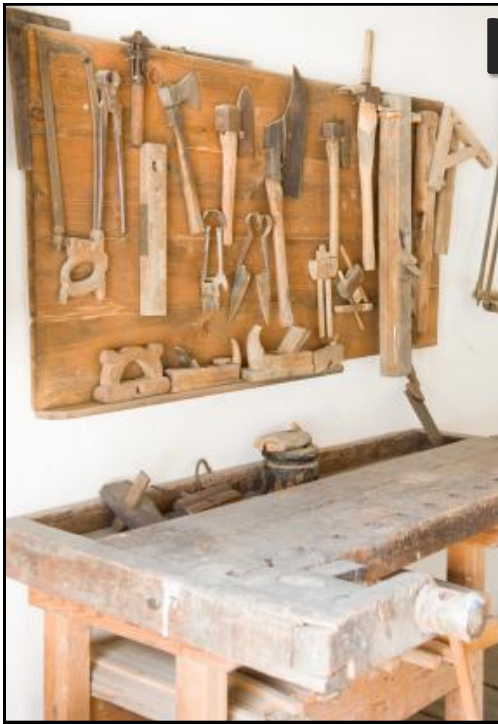
use  
**persistent**  
**abstract**  
representation



# Language Workbench

(Martin Fowler)

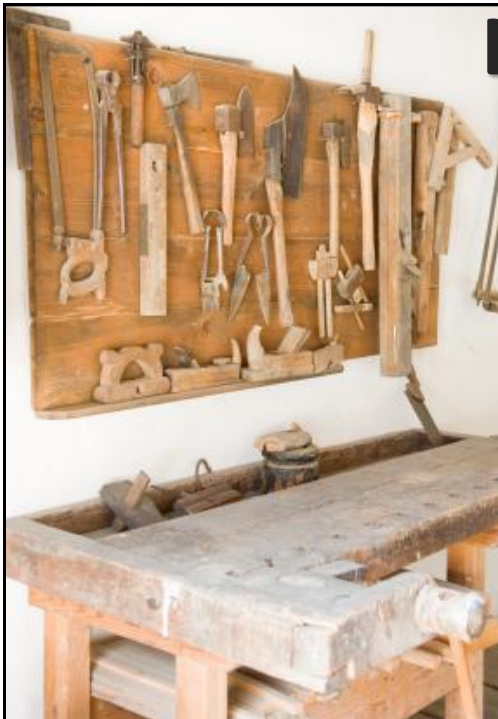
language ::=  
**schema**  
**+ editors**  
**+ generators**



# Language Workbench

(Martin Fowler)

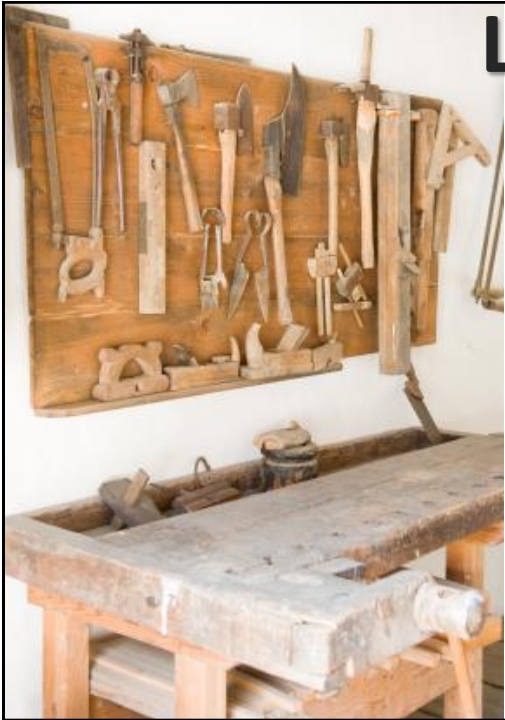
projectional  
editing



# 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

## 9

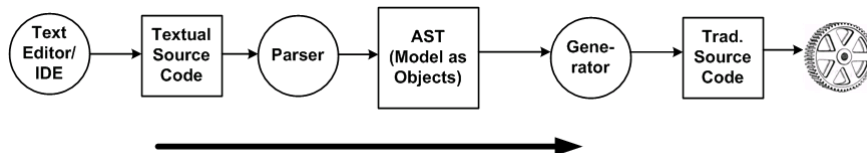
# Projectional Editing

## Parser-based

**text**

**... to tree**

**... to text**



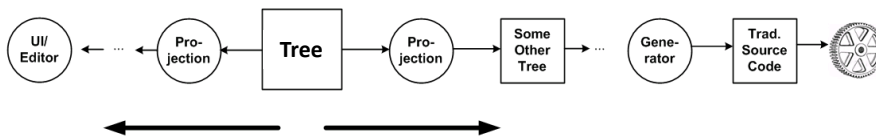
# Projectional

**tree**

... to text-lookalike (editor)

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

... to text



# Language Composition

**There's no parsing.**

**Unique Language Element Identity.**

**Unlimted language composition.**

# Flexible Notations

**Textual**

like ASCII

**Graphical**

box & line

**Semi-Graphical**

mathematical



treated the same  
can be mixed

# Automatic IDE Extension

**tool support is inherent  
for languages build with  
projectional tools**

language definition  
implies  
IDE definition

# Multiple Notations

**... for the same concepts**

**e.g. in different contexts  
or for different tasks**

# Partial Projections

**... different views**

**... for different roles/people**

**... only a particular variant**

# Tree Editing

**... is different from editing text**

**... try to make it feel like text**

**... takes some getting used to**

but: for more flexible notations  
a more general editing paradigm  
is needed

# Infrastructure Integration

**... storage is not text**

**... diff/merge must be in tool**

**... existing text tools don't work**

# Proprietary Tools

... no standards


... no interop

# 10

**JetBrains**  
**MPS**



**<http://jetbrains.com/mps>**



**Open Source (Apache 2.0)**  
**Projectional Editor**  
**Very good at lang. Composition**  
**Version 2.0 August 2011:**

- Improved performance**
- Unified generate/compile/build**
- Debug MPS in MPS**
- Tables in the editor**



**Open Source (Apache 2.0)**

**Projectional Editor**

**Very good at lang. Composition**

**Version 2.0 August 2011**

**Version 2.1 early 2012**

**Graphical Editors, Several Editors per Concept,  
Wiki-Language and more.**



**DEMO<sub>2</sub>**

**Extending  
The Language**

