1

Embedded Development
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
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.

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.
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
Concerns & Viewpoints

Viewpoints

suitable abstractions and notations for each
Viewpoints

Integrated references via symbolic or syntactic composition

6 Domain-Specificity
7

The LWES Project
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
http://mbeddr.com

Meta Programming System

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

ITEMS
fortiss
SICK
Sensor Intelligence
LEAR CORPORATION
Using the C Language
8
Language Workbenches

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" modeling
9

Projectional Editing

Parser-based

... 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.
Unlimited 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. Composision
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. Compososition
Version 2.0 August 2011
Version 2.1 early 2012

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

DEMO

Extending
The Language
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