Variability in mbeddr

Current State + Challenges

Markus Völter + mbeddr team
An extensible set of integrated languages for embedded software engineering.

Specific Languages
#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 airborne {
    \// Here is a comment on a transition.
    on next [tp->alt == 0 m \&\& tp->speed == 0] \-> crashed
  }
  state landing {
    on next [tp->speed == 0 mps] \-> landed
    [on next [tp->speed > 0 mps] \-> landing { points--; }] \> imp.
  }
  state airborne {
    [tp->alt == 0 m \&\& tp->speed == 0] \-> crashe
    [tp->alt == 0 m \&\& tp->speed > 0 mps] \-> lan
    [tp->speed > 200 mps \&\& tp->speed > 0 mps] \-> lan
    [tp->speed > 100 mps \&\& tp->speed <= 200 mps] \-> airborne
    [tp->speed == 0 mps] \-> landed
    [tp->speed == 0 mps] \-> landing \-> implements Shuttle
  }
}

itemis France: Smart Meter

First significant mbeddr project
cia. 100,000 LoC
about to be finished
great modularity due to components
uses physical units extensively
great test coverage due to special extensions
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.
20+ Projects in various stages
by various “Big Name” companies.
2
The Language Workbench
Open Source
Apache 2.0
http://jetbrains.com/mps
Language Workbench

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

Parsing

Projectional Editing

Concrete Syntax

Abstract Syntax Tree

Concrete Syntax

Abstract Syntax Tree
[Projectional Editing]
Syntactic Flexibility

Regular Code/Text

Mathematical

Tables

Graphical
[Projectional Editing]
Language Composition

Separate Files
Type System
Transformation
Constraints

In One File
Type System
Transformation
Constraints
Syntax
IDE
Version 3.1.2 released recently:
Better Performance, Graphical Editors
Better Tables, Tooltips, Better Console
SLE Paper
Current State
[Feature Models]
Textual Notation
Graphical Visualization
SAT-based checking
Feature Conditions

Expressions over features in feature models

IDE-integration with feature models

SAT-checked for consistency
[Presence Conditions]

Static Variability – pimped #ifdef

Attachable to any program node, any language

IDE support to show the variant

Ref Checking for variable code
[Runtime Variability]
Special Variant-Switch Statement usable in C statement context
Interfaces + Components acting as the black boxes to be configured.
[Composition F/W]

Positive Variability
Merge Operators
Domain-Specific Extensions

experimental
Challenges
Integration with Type System
Integration with Type System

**NOT** ref. integrity or name resolution. This already works.

**BUT** arbitrary constraints such as name uniqueness and actual type calculations
Integration with Type System
Thank you!