



Zeeshan Malik

Kevin Shelley

Defining a DSL

- Schema (Abstract Syntax)
 - Abstract Syntax – Vs – Concrete Syntax
- Editor
- Generator

Defining a Language Workbench

- Language Creation and Integration
- Persistent Abstract Representation
- Schema, Editor(s), Generator(s)
- Projectional Editor
- Incomplete or contradictory information in its abstract representation

Model Based Software Development

- UML PIM (Platform Independent Modeling)
- MOF (Meta Object Facility)
- MDD (Model Driven Development)
- LOP (Language Oriented Programming)
through Language Workbenches

Meta Programming System

- MPS – A Language Workbench
 - Structure Language
 - Editor Language
 - Transformation Language
 - Model Query Language (MQL)
 - Template Language
 - Pattern Matching Language
- Other aspects of a language
 - Type System
 - Behavior
 - Constraints
 - Data Flow Analysis

DEMO

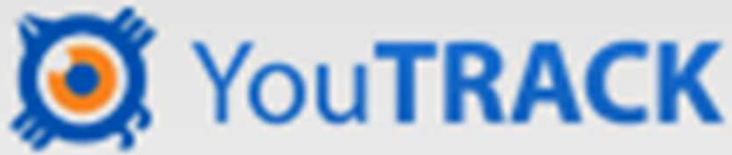
Insights and Lessons Learned

- Projectional Editors
 - Helpful for inexperienced developers
 - Enforces good coding practices
 - Superior Debugging capabilities
 - Prompt for automatic refactoring
 - Code completion, Suggestions, Data Flow Visualization
 - Not all code fits nicely into a template

Insights and Lessons Learned

- MPS is a work in Progress
 - Copy/Paste errors when transferring to/from text editors
 - Error checking can be too strict or buggy
 - Sparse/Dated documentation
- Overall MPS is a very powerful and useful tool
 - Covers all aspects of Language Development
 - Transformations are tricky to work with in the beginning

MPS in Production



YouTrack.JetBrains.com



Mbeddr.WordPress.com



References:

- Language Oriented Programming, The Next Programming Paradigm; Sergey Dmitriev, JetBrains
- Language Workbenches and Model Driven Architecture; Martin Fowler
- Language Workbenches: The killer-App for Domain Specific Languages?; Martin Fowler
- MPS Screencasts, blog and documentation; jetbrains.com/mps