Rapid Prototyping of Language Servers with Rascal, VSCode, and Gitpod

Jurgen J. Vinju
Centrum Wiskunde & Informatica
TU Eindhoven
Swat.engineering
There is way too much code...

- Code can be **generated** from DSLs
  - better exploration, testing, verification
  - better maintenance
  - less boring = more fun!
- Code can be **analyzed** (using “open” compilers)
  - evidence-based maintenance
  - debugging
- Code can be **transformed**, source-to-source
  - mass maintenance activities
  - code-to-model, reverse engineering

Really <3 source code but there is too much of it!

Automating software construction and maintenance is the [only] humane solution [T. Girba]
Rascal is for everything meta

- diverse languages everywhere
- parse extract query infer transform
- parse
- extract
- query
- infer
- transform
- operations
- parses
- extracts
- queries
- inferences
- transformations

- visuals
- models
- PL code
- facts
- strings
- graphs
- tables
- trees
- data types

accurate artefact identification is key

accurate artefact identification is key

SWAT - Software Analysis And Transformation
Meta in the IDE

- focus on Domain Specific Languages (DSLs)
- want editor support with all the fanciness
- Microsoft’s Language Server Protocol

UI \leftrightarrow \text{LSP Client} \leftrightarrow \text{LSP Server} \quad \text{Rascal language -servers}

chatty protocol

generates
Meta in the IDE

- want editor support with all the fanciness
- Microsoft’s Language Server Protocol
- GitPod frees the generated IDEs
- Rascal provides **immediate** extensions
YOP

- YOP is a toy language made for highschool students
- Make Your Own Programming Language
- LOGO-like
- Features: parsing, intermediate miniSVG language, code generation to SVG, static checking, jump-to-definition
- ±350 LOC
- Funny variations:
  - colors
  - procedures with recursion
  - smoothening optimizer
```java
herhaal 8 {
  rechts 45
  herhaal 6 {
    herhaal 90 {
      vooruit 2
      rechts 2
    }
  }
  rechts 90
}
```
An open source community; script all your coding and research activities with → Rascal ← http://www.rascal-mpl.org

Amazing immediate full-blown development environments

Co-design and co-maintenance of software portfolios through (domain-specific) language engineering

SWAT: the next steps in programming and software engineering in Amsterdam

Other examples

- DSL with ING: “Rebel” DSL for distributed transaction systems with availability and isolation guarantees
- DSL with NFI: carving fast and slow, generating “undelete” algorithms for recovering destroyed data
- Example research (by Lina Ochoa): measuring impact of breaking changes in APIs and the effective use of semantic versioning on the ecosystem level