# grammarware legacy

Vadim Zaytsev, SWAT, CWI, 2010–2013

# output

# PEM Colloquium

- Recovery, Convergence and Documentation of Languages (16/XII/2010)
- Grammar Investigation (24/II/2011)
- Cheating on the Undecidability of Language Equivalence (28/IV/2011)
- Toward an Engineering Discipline for Grammar Recovery (18/VIII/2011)
- Bidirectional Transformations and Grammarware (3/II/2012)
- Tolerance in Grammarware (4/V/2012)
- Experimental Replications (25/V/2012)
- Grammar Composition and Extension (10/VII/2012)
- Negotiated Transformations (18/I/2013)
- Software Engineering Considered Harmful (15/II/2013)
- Modeling Software Structures with GrammarLab (17/V/2013)

# Conferences

- SLE 2011: “Comparison of Context-free Grammars Based on Parsing Generated Test Data”
- Wikimania 2011: “Wiki Migration”
- BX 2012: “Language Evolution, Metasyntactically”
- SAC 2012: “BNF WAS HERE: What Have We Done About the Unnecessary Diversity of Notation for Syntactic Definitions”
- LDTA 2012: “Notation-Parametric Grammar Recovery”
- SATToSE 2012: “Bidirectional Grammar Transformations”
- SFD 2012: “Open Notebook Computer Science”
- SL(E)BOK 2012: “Subatomic Scientific Knowledge Objects”
- XM 2012: “Negotiated Grammar Transformation”
- SATToSE 2013: “Modelling Robustness with Conjunctive Grammars”
- XM 2013: “Pending Evolution of Grammars”
- XM 2013: “Language Support for Megamodel Renarration”
- MoDELS 2013: “Modeling Software Structures with GrammarLab”
- SLE 2013: “Micropatterns in Grammars”

# Other presentations

- Grammar Comparison Techniques (SERG, 5/X/2011)
- The Life Cycle of Grammarware (CWI SM, 3/II/2012)
- Maintenance and Evolution of Grammarware by Grammar Transformation (IPA MDSE, 18/IV/2012)
- Advanced Metaprogramming (SLT, 17/VII/2012)
- Megamodelling Language Design: User Experiences and Ad Hoc Megamodelling (SLT, 18/VII/2012)
- Grammar Convergence (SLT, 25/VII/2012)
- Renarration of Megamodels (SLT, 25/VII/2012)
- Rascal Metaprogramming Language (SoTeSoLa, 20/VII/2012)
- Wiki Loves Monuments Data Recovery and Curation (WMFH, 10/XI/2012)
- A Snappy Introduction to Metaprogramming in Rascal (RedDevCon, 26/I/2012)
- Alphametic Cryptarithms and Polychromatic Coppices in Rascal (SATToSE, 10/VII/2013)

# Posters

- Grammars Matter (CLUS)
- Grammarware Engineering (CLUS)
- Grammar Convergence (CLUS)
- Renarrating Linguistic Architecture (MPM)
- Guided Grammar Convergence (SLE)

# Publications (DBLP)

- Software Quality Journal 19(2), 2011
- EC-EASST 49, 2012
- SLE 2011, 2013
- SAC 2012
- LDTA 2012
- WCRE 2013
- arXiv (CoRR) 1107.4661, 1207.6541, 1212.4446
- XM@MoDELS 2012 ×2

# Publications (ACM DL)

- Software Quality Journal 19(2), 2011
- SLE 2011, [2013]
- SAC 2012
- LDTA 2012
- XM 2012
- MPM 2012

# Publications (submitted)

- Negotiated Grammar Transformation (JOT, second round)
- Grammar Zoo: A Repository of Experimental Grammarware (SCP, second round, reviewers' turn)
- Formal Foundations for Semi-parsing (CSMR-WCRE ERA, written in UvA time, accepted)
- Software Language Engineering by Intentional Rewriting (SQM, under review)
- A Bidirectional View on Parsing (BX, with AHB)

# Publications (rejected)

- (metrics) IFIP Performance 2011
- (mining) Veni 2012, 2013
- (guided) ECMFA’12, ICSM’12, IPL, NWPT’12, POPL’13, ESEC’13
- (mutations) TFP’12, RTA’13, SCAM’13
- (tolerant) SCAM’12, NordiCloud’12, IFM’13
- (recovery) JUCS, ICSM ERA’13
- (visualisation) FSE NIER’12
- (unparsing) FSE NIER’12
- (replications) EMSE
- (renarration) MPM’13

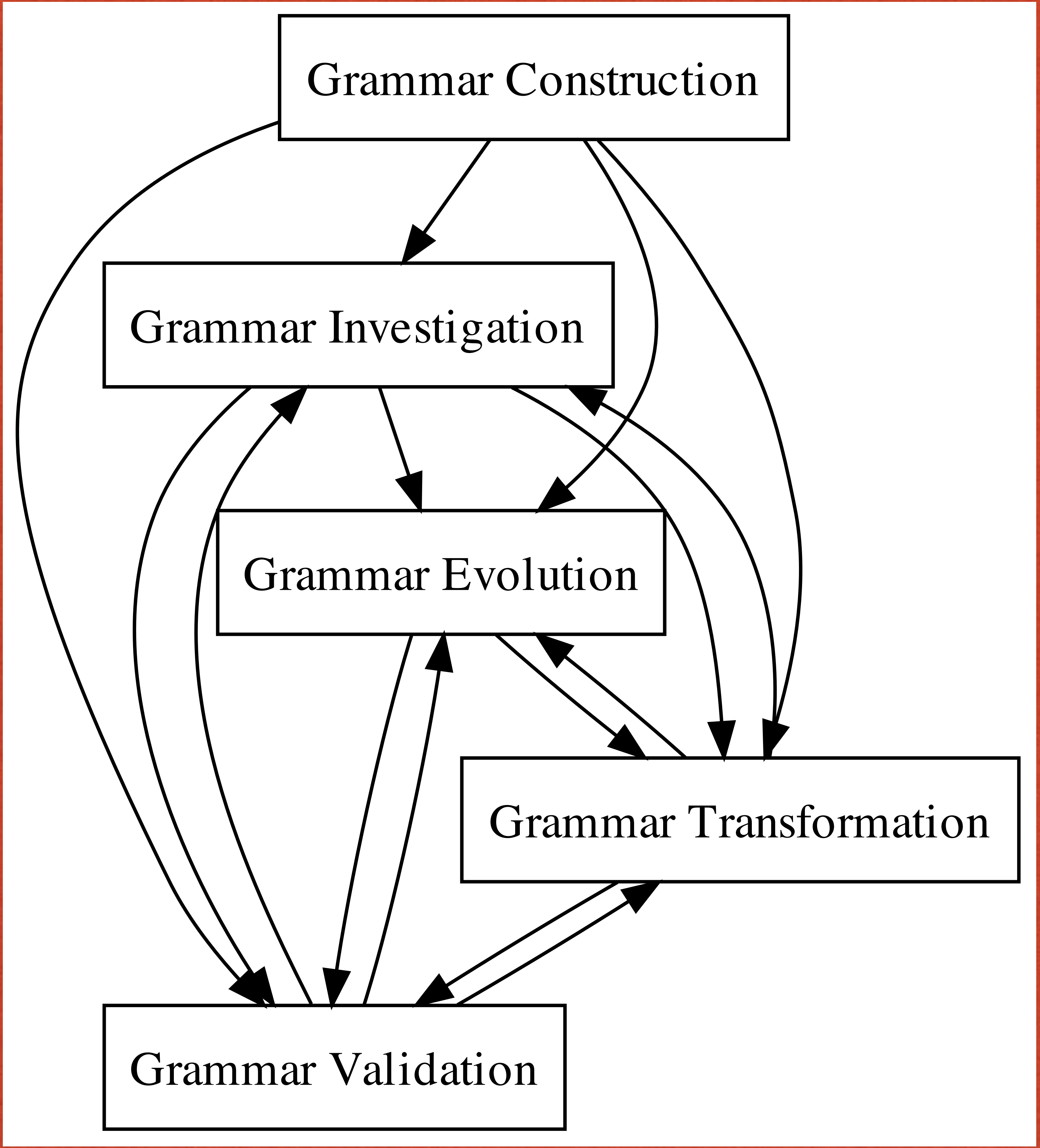
# Event activities

- Org:
  - OpenDataDay 2013, OOPSLE 2013-14, WCN 2011-12 (PC), WCRE 2013 (TDT), SoTeSoLa 2012 (H), SATToSE 2013 (H)
- PC:
  - SCAM 2011-13, WCN 2011-13, SQM 2012-13, LDTA 2012, ACM SRC 2013, XM 2013, WLM 2013
- Pub/Media:
  - SLE 2011, GTTSE 2011, SoTeSoLa 2012, MoDELS 2013
- (Sub)reviewer:
  - LOPSTR 2011, ESEC/FSE TDT 2011, WCRE 2012, IET Software, SCP (×5?), SoSyM, ESEM
- Non-activities:
  - Joy of Coding (×2), Document Freedom Day, BENEVOL 2013

# Hacking

- SLPS
  - http://github.com/grammarware/slps
- GrammarLab
  - http://github.com/cwi-swat/grammarlab
- Zoo (& Tank)
  - merged, currently part of SLPS

# requirements & outcomes


Grammar Construction
Grammar Investigation
Grammar Evolution
Grammar Transformation
Grammar Validation

# Grammar construction

- Grammar notation inference
  - ideally: impossible
  - given the notation: done, coded, published
  - decision points in notational design (WIP)
  - error-tolerant recovery: done, coded, published
  - survey of semi-automated approaches

# Grammar construction

- Grammar inference
  - not done; just discussed
- Grammar extraction
  - many mappers coded (hard to count)
  - XSLT; Python; Prolog; Rascal; …

# Grammar investigation

- Grammar facts
  - easy, unpublishable
- Static grammar analysis
  - many sketches, drafts, no solid result
  - micropatterns: invented, coded, published

# Grammar investigation

- Grammar smells
    - idea by Tijs, draft by me; claimed for SLE2014
- Patterns of semantic action usage
    - not done
    - uninteresting
    - somewhat addressed in semi-parsing

# Grammar evolution

- Advanced grammar diff
  - Levenshtein/Hammond/… – not done
- Suggestive grammar metrics
  - normalisations – done, unpublished
  - refactoring suggestions, etc – not done

# Grammar evolution

- Inference of transformations
  - guided convergence — biggest time waster
  - mutations — useful, not [yet] published
- Grammar synchronisation
  - some bx

# Grammar evolution

- Grammar configuration management
  - versioning, conflict resolution — not done
- Storage aware of grammar knowledge
  - dependencies, product lines — kinda, not really
-

# Grammar transformation

- Views for grammars
  - nothing convincing and/or publishable
- Export to various formats
  - sure, plenty
- Grammar visualisation
  - nothing

# Grammar transformation

- XBGF in Rascal
  - done & redone
- Transformation wizards
  - mutations done, no point-and-click
- Auto-completion
  - nope

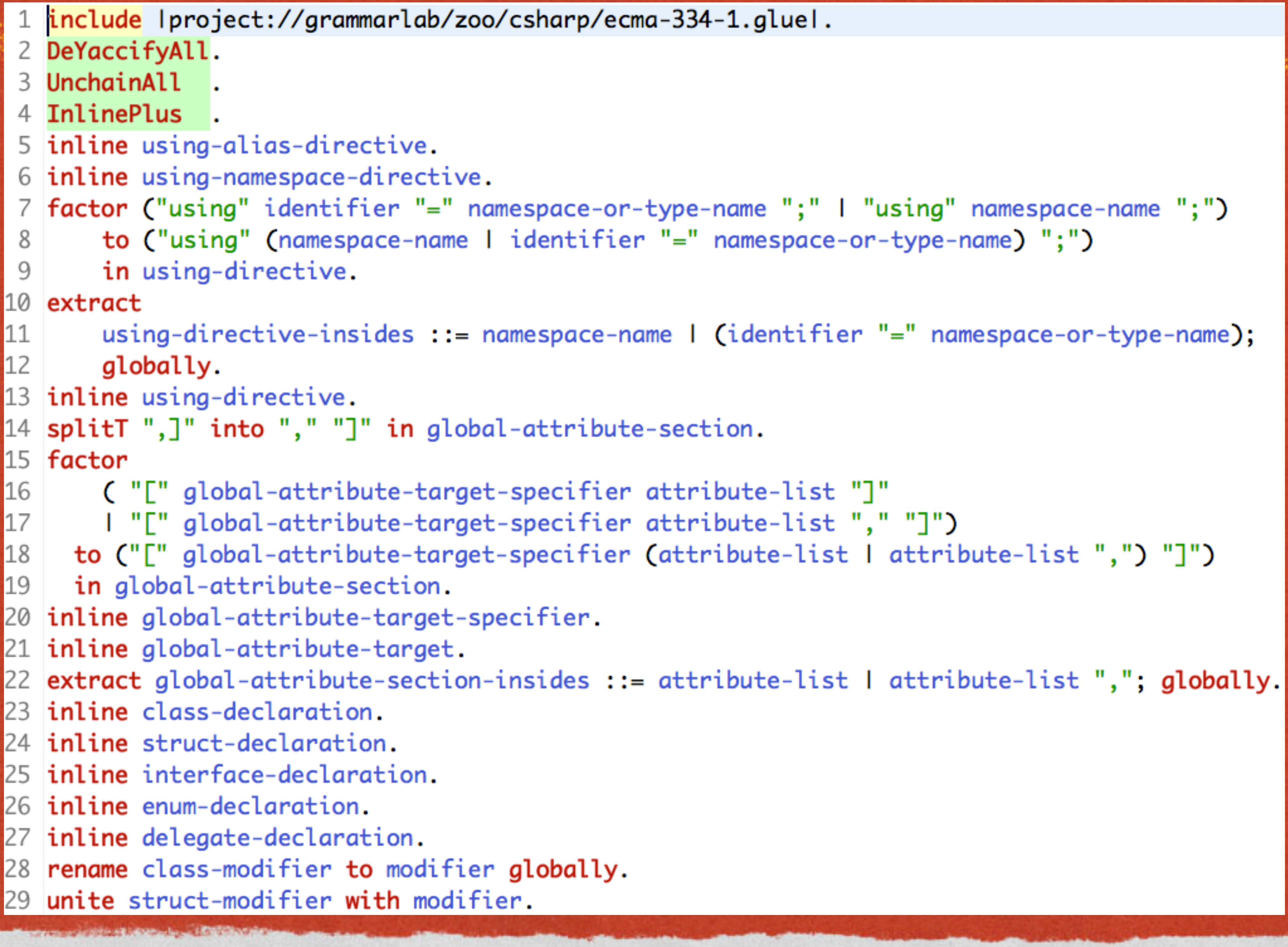
# Grammar validation

- Combinatorial test generation
    - done, but not in Rascal
- Grammar testing infrastructure
    - parts of Grammar Lab — sure; no framework
- Test coverage analysis
    - done, published

# Grammar validation

- Assertions / contracts
  - no
- Grammar verification
  - proving properties — not done
  - transformation verification — also no
- Parsing visualisation
  - no

# Grammar Lab



```
include |project://grammarlab/zoo/csharp/ecma-334-1.glue|.
DeYaccifyAll.
UnchainAll  .
InlinePlus  .
inline using-alias-directive.
inline using-namespace-directive.
factor ("using" identifier "=" namespace-or-type-name ";" | "using" namespace-name ";")
    to ("using" (namespace-name | identifier "=" namespace-or-type-name) ";")
    in using-directive.
extract
    using-directive-insides ::= namespace-name | (identifier "=" namespace-or-type-name);
    globally.
inline using-directive.
splitT ",]" into "," "]" in global-attribute-section.
factor
    ( "[" global-attribute-target-specifier attribute-list "]"
    | "[" global-attribute-target-specifier attribute-list "," "]")
  to ("[" global-attribute-target-specifier (attribute-list | attribute-list ",") "]")
  in global-attribute-section.
inline global-attribute-target-specifier.
inline global-attribute-target.
extract global-attribute-section-insides ::= attribute-list | attribute-list ","; globally.
inline class-declaration.
inline struct-declaration.
inline interface-declaration.
inline enum-declaration.
inline delegate-declaration.
rename class-modifier to modifier globally.
unite struct-modifier with modifier.
```

# Software Language Processing Suite

- Lots of stuff
- Hard to navigate
- Vastly multilinguistic
- Makefiles & command-line oriented
- Scarcely compatible with Eclipse ecosystem

# Grammar Lab

- CWI repository at GitHub
- Slowly growing, ~8K Rascal, ~4K GLUE atm
- Not everything is migrated
- Everything is uniform
- EBNF, XBGF, SLEIR, …

# Conclusion