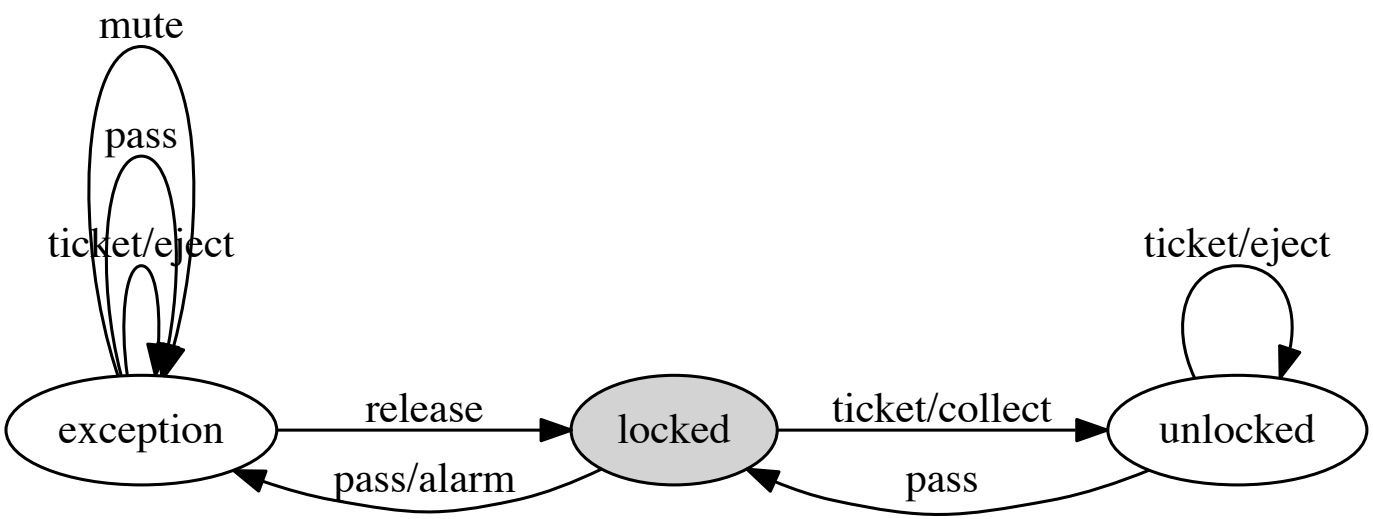
What’s software language engineering?

Ralf Lämmel
Software Languages Team
University of Koblenz
http://www.softlang.org/

What's **software language engineering**?
What's a **software language**?
How to **engineer** such languages?

A finite state machine for a revolving door

(Concrete visual syntax)



A DSL instead of a "general purpose" PL

A finite state machine for a revolving door

(Concrete textual syntax)

```
initial state locked {
   ticket/collect –> unlocked;
   pass/alarm –> exception;
}
state unlocked {
   ticket/eject;
   pass –> locked;
}
state exception {
   ticket/eject;
   pass;
   mute;
   release –> locked;
}
```

A DSL instead of a "general purpose" PL

A finite state machine for a revolving door

(Abstract JSON-based syntax)

```
{"exception": [{
      "initial": false,
      "transitions": {
            "release": [[null, "locked"]],
            "ticket": [["eject", "exception"]],
    "mute": [[null, "exception"]],
    "pass": [[null, "exception"]]}}],
 "locked": [{
       "initial": true,
       "transitions": {
          "ticket": [["collect", "unlocked"]],
          "pass": [["alarm", "exception"]]}}],
 "unlocked": [{
       "initial": false,
       "transitions": {
          "ticket": [["eject", "unlocked"]],
          "pass": [[null, "locked"]]}}]}
```

A DSL instead of a “general purpose” PL

Some real world software languages

- Haskell: The functional programming language Haskell
- Java: The Java programming language
- Python: The dynamic programming language Python
- ANTLR: The grammar notation of the ANTLR technology
- JSON: The JavaScript Object Notation
- JSON Schema: The JSON Schema language
- XML: Extensible Markup Language
- XSD: XML Schema Definition
- Alloy: The Alloy specification language
- CIL: Bytecode of .NET's CLR
- Common Log Format: The NCSA Common log format
- DocBook: The DocBook semantic markup language for documentation
- FOAF: The friend of a friend ontology
- INI file: The INI file format
- Java bytecode: Bytecode of the JVM
- make: The make tool and its language
- OWL: Web Ontology Language
- Prolog: The logic programming language Prolog
- QTFF: QuickTime File Format
- RDF: Resource Description Framework
- RDFS: RDF Schema
- Scala: The functional OO programming language Scala
- Smalltalk: The OO reflective programming language Smalltalk
- SPARQL: SPARQL Protocol and RDF Query Language
- UML: Unified Modeling Language
- XPath: The XML path language for querying
- XSLT: Extensible Stylesheet Language Transformations

Some types of software languages

- data-modeling languages;
- markup languages;
- programming languages;
- specification languages;
- stylesheet languages;
- transformation languages.

Language classification by purpose

Purpose (language)	**Purpose** (element)	**Classifier**	**Example**
Programming	Program	Programming language	Java
Querying	Query	Query language	XPath
Transformation	Transformation	Transformation language	XSLT
Modeling	Model	Modeling language	UML
Specification	Specification	Specification language	Alloy
Data representation	Data	Data format	QTFF (QuickTime file format)
Documentation	Documentation	Documentation language	DocBook
Configuration	Configuration	Configuration language	INI file
Logging	Log	Log format	Common Log Format
…	…	…	…

The lifecycle of a software language

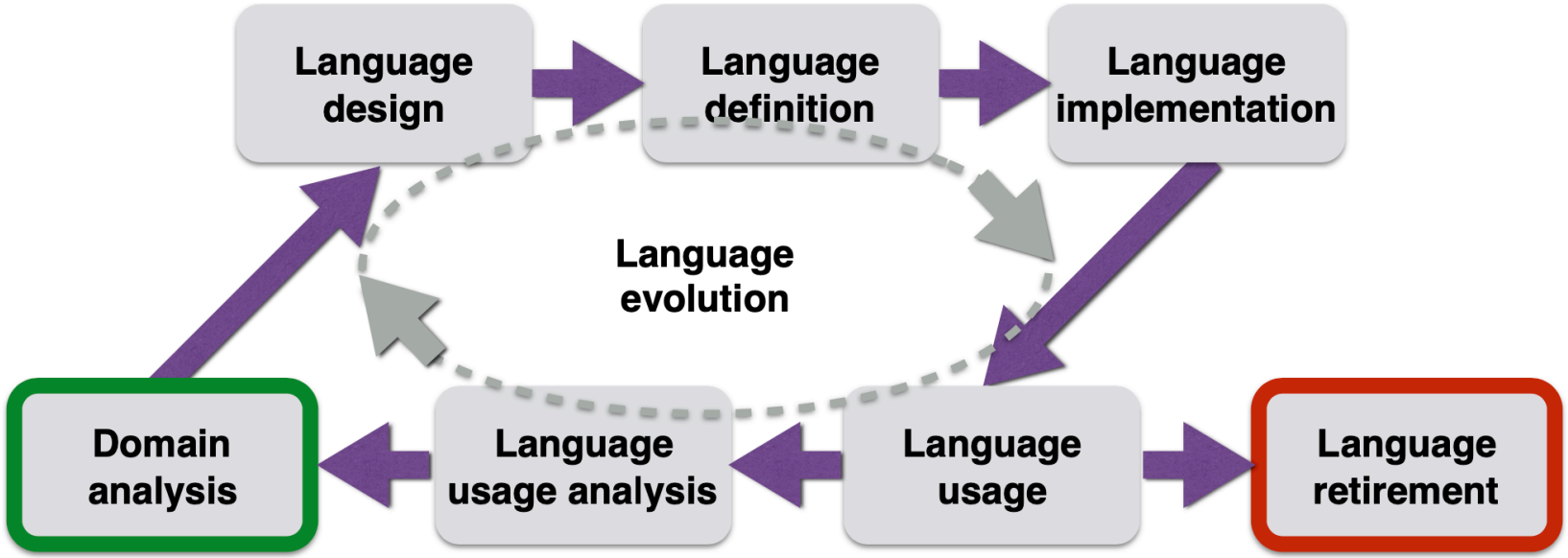


Fig. — The lifecycle of a software language. The nodes denote phases of the lifecycle. The edges denote transitions between these phases. The lifecycle starts with *domain analysis*. The lifecycle ends (theoretically) with *language retirement*. We may enter cycles owing to language evolution.

Typical software language implementations

- Compiler
- Interpreter
- Language processors
  - Parsers / unparsers / preprocessors
  - Software transformations
  - Software analyses
  - Software generators
  - Software translators
  - Software specializers
  - …

Simplified data flow in a compiler

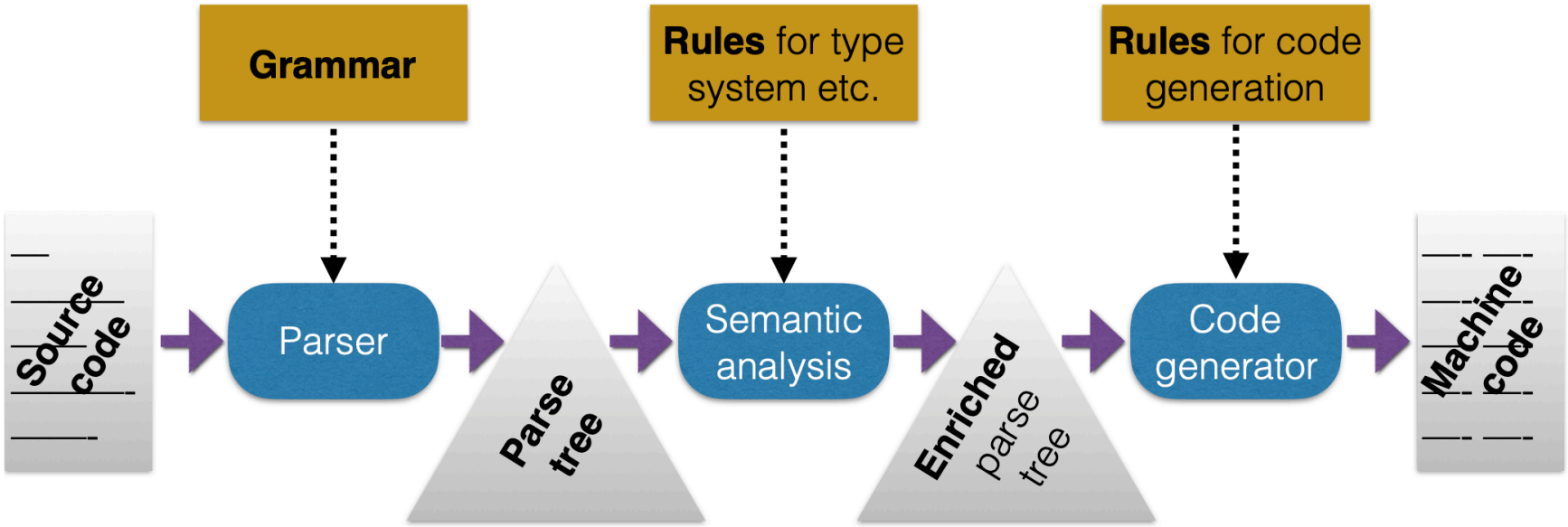


Fig. — Simplified data flow in a compiler. The rectangles with rounded edges represent logical phases of compilation. The remaining nodes (rectangles and triangles) correspond to input and output, as expressed by the direction of the arrows.

Example of re-factoring (re-engineering)

```
void printOwing(double amount) {
        printBanner();
        System.out.println("name: " + name);
        System.out.println("amount: " + amount);
}
```

```
void printOwing(double amount) {
        printBanner();
        printDetails(amount);
}
void printDetails(double amount) {
        System.out.println ("name: " + name);
        System.out.println ("amount: " + amount);
}
```

Data flow for a re-engineering transformation

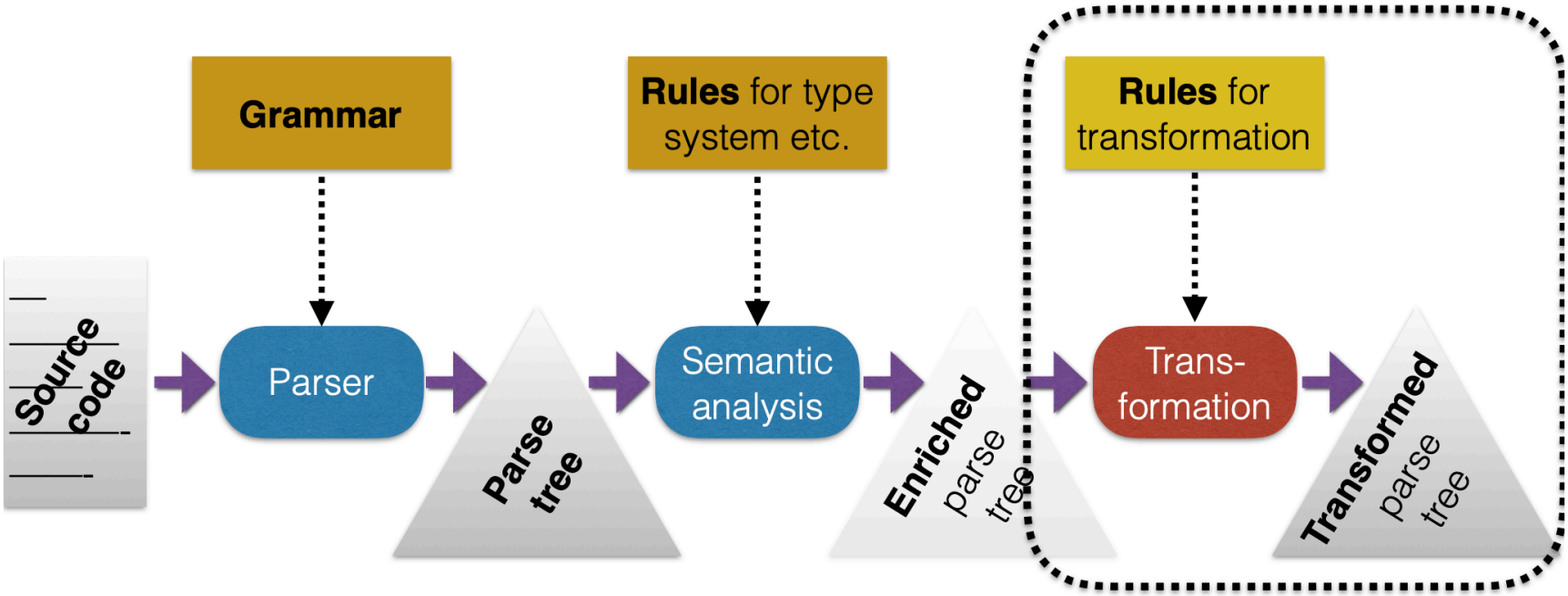


Fig. — Overall data flow for a re-engineering transformation. We have marked the phase which replaces code generation in the standard data flow for compilation.

Example of analysis (reverse engineering)

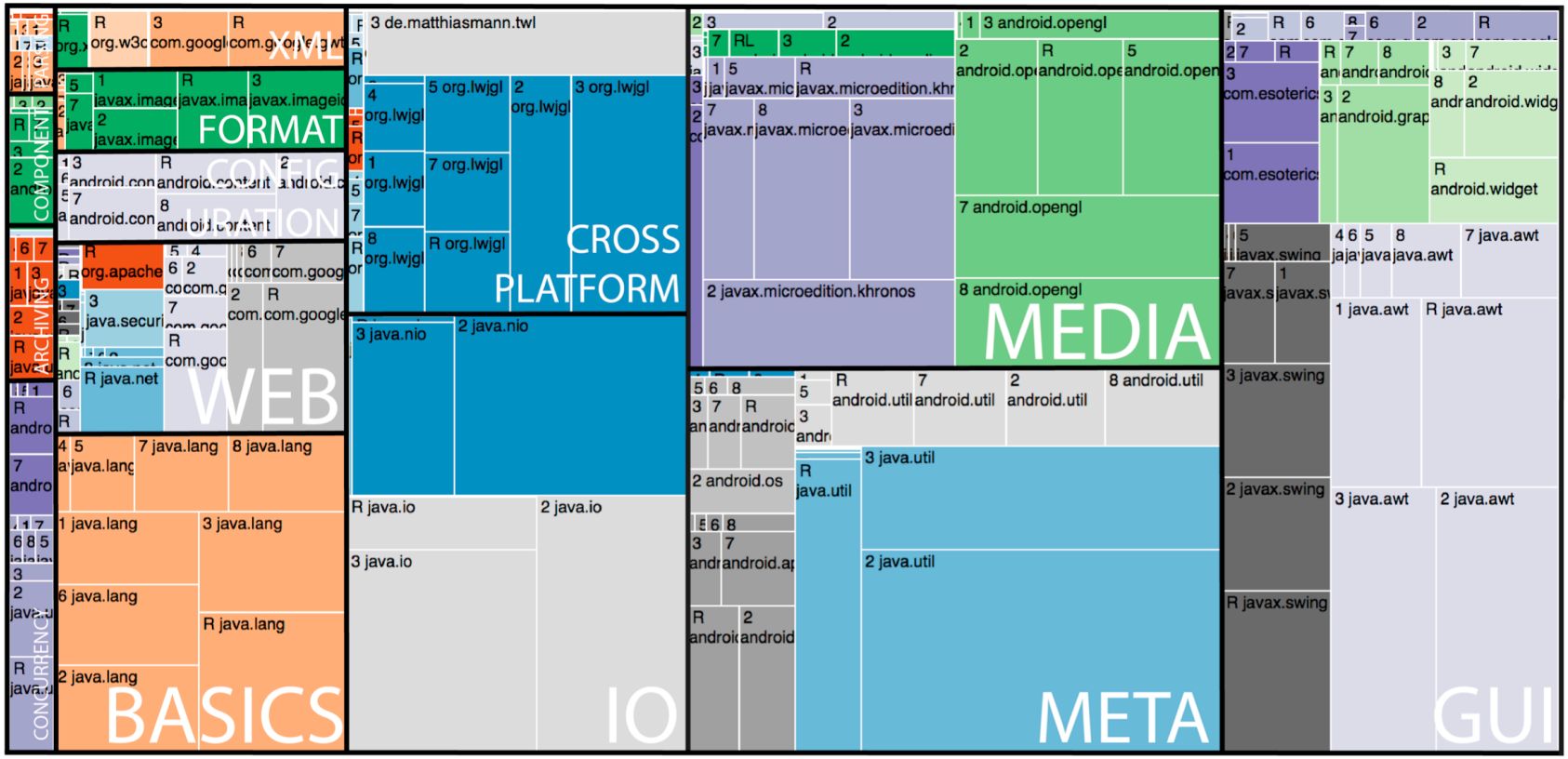


Fig. — An API-usage map for an open-source Java project. The complete rectangle (in terms of its size) models the references to all APIs made by all developers. The nested rectangles partition references by domain (e.g., GUI rather than Swing or AWT). The rectangles nested further partition references by API; one color is used per API. Within each such rectangle, the contributions of distinct developers (1, . . . , 8 for the top-eight committers and "R" for the rest) are shown. Source: [4].

Data flow for reverse engineering

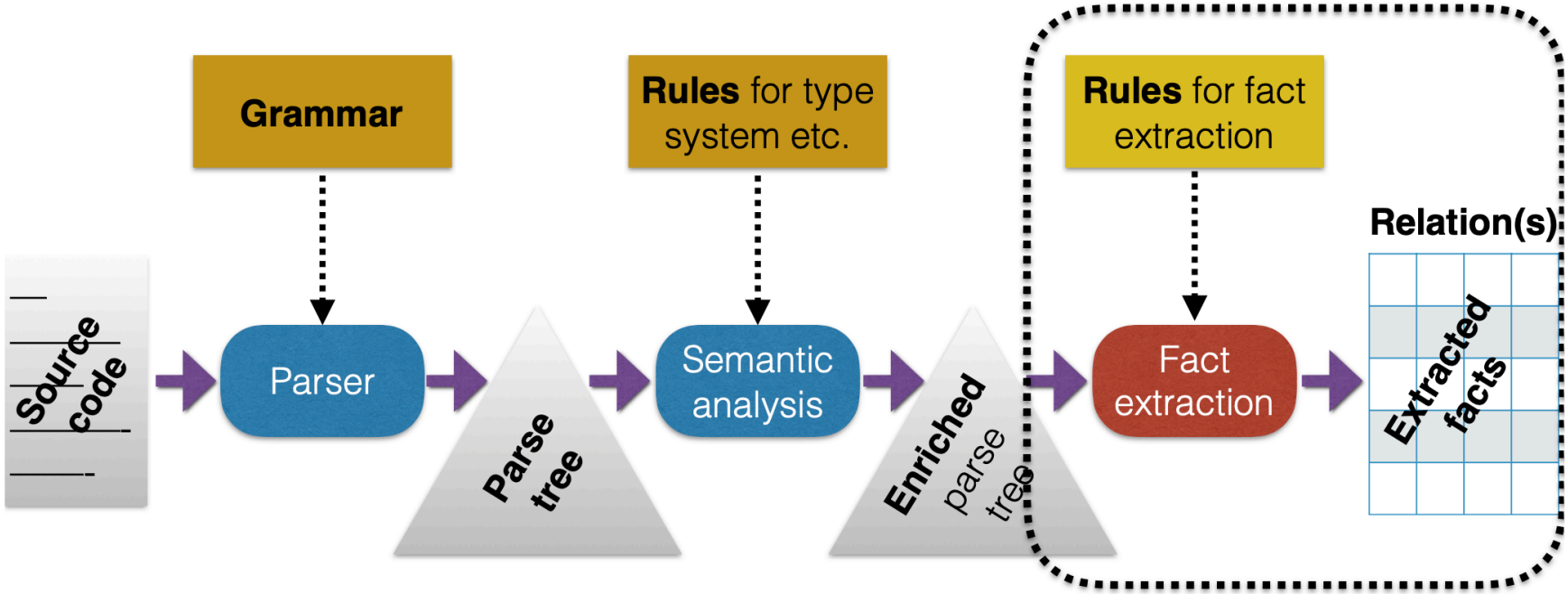


Fig. — Overall data flow for fact extraction in reverse engineering. We have marked the phase which replaces code generation in the standard data flow for compilation. The resulting tables represent "projections" of the source code, for example, call relationships between functions.

Technological spaces

Grammarware string, grammar, parsing, CST, AST, term, rewriting, …

XMLware XML, XML infoset, DOM, DTD, XML Schema, XPath, XQuery, XSLT, …

JSONware JSON, JSON Schema, …

Modelware UML, MOF, EMF, class diagram, modeling, metamodeling, model transformation, MDE, …

SQLware table, SQL, relational model, relational algebra, …

RDFware resource, triple, Linked Data, WWW, RDF, RDFS, OWL, SPARQL, …

Objectware objects, object graphs, object models, state, behavior, …

Javaware Java, Java bytecode, JVM, Eclipse, JUnit, …

Technological spaces

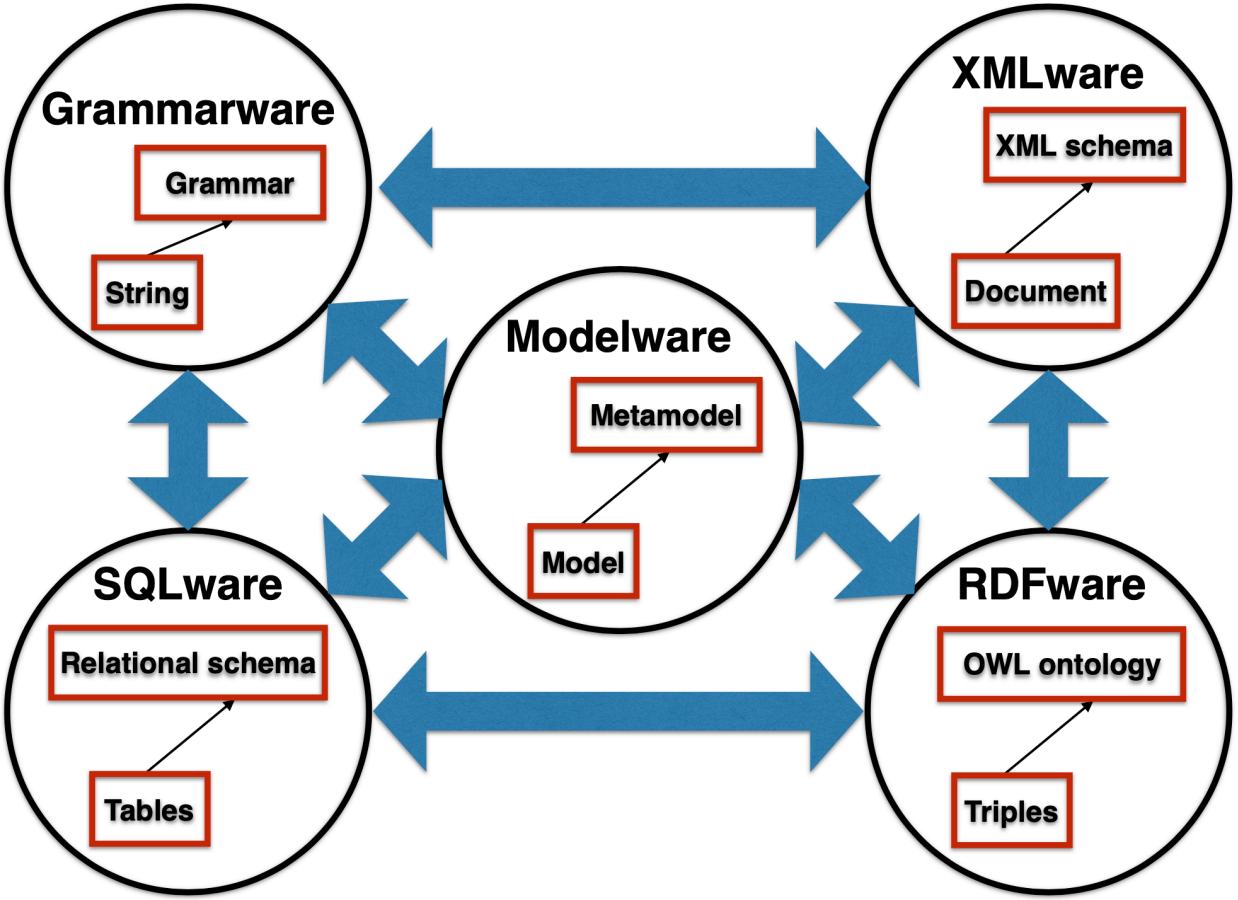


Fig. — A few technological spaces with their instance (data) level and the schema level. The thin arrows model "conformance" such as an XML document conforming to an XML schema. The thick arrows hint at expected "bridges" between the spaces as needed for technological space travel. Inspired by [161].

Online resources

YAS (Yet Another SLR (Software Language Repository))

http://www.softlang.org/yas

YAS' GitHub repository contains all code.

The Software Languages Book

http://www.softlang.org/book