Software Language Engineering Code generation

Tijs van der Storm

CWI
Centrum Wiskunde & Informatica

university of groningen

Recap

- Grammar -> Parser -> Parse Tree -> AST
- Name resolution: recover referential structure
- Checking: find errors not captured by syntax
- Today:
    - semantics
    - compilation/code generation

Transformation

- Translation
- Restructuring
- Generation
- Optimization
- …

Compilation

- Translation from high-level to low-level
  (= lowering the level of abstraction)
  - Java -> JVM byte code
  - C -> x86 machine code
  - JVM byte code -> x86 machine code
  - QL -> HTML + Javascript
- So not, e.g., Java to C# translation

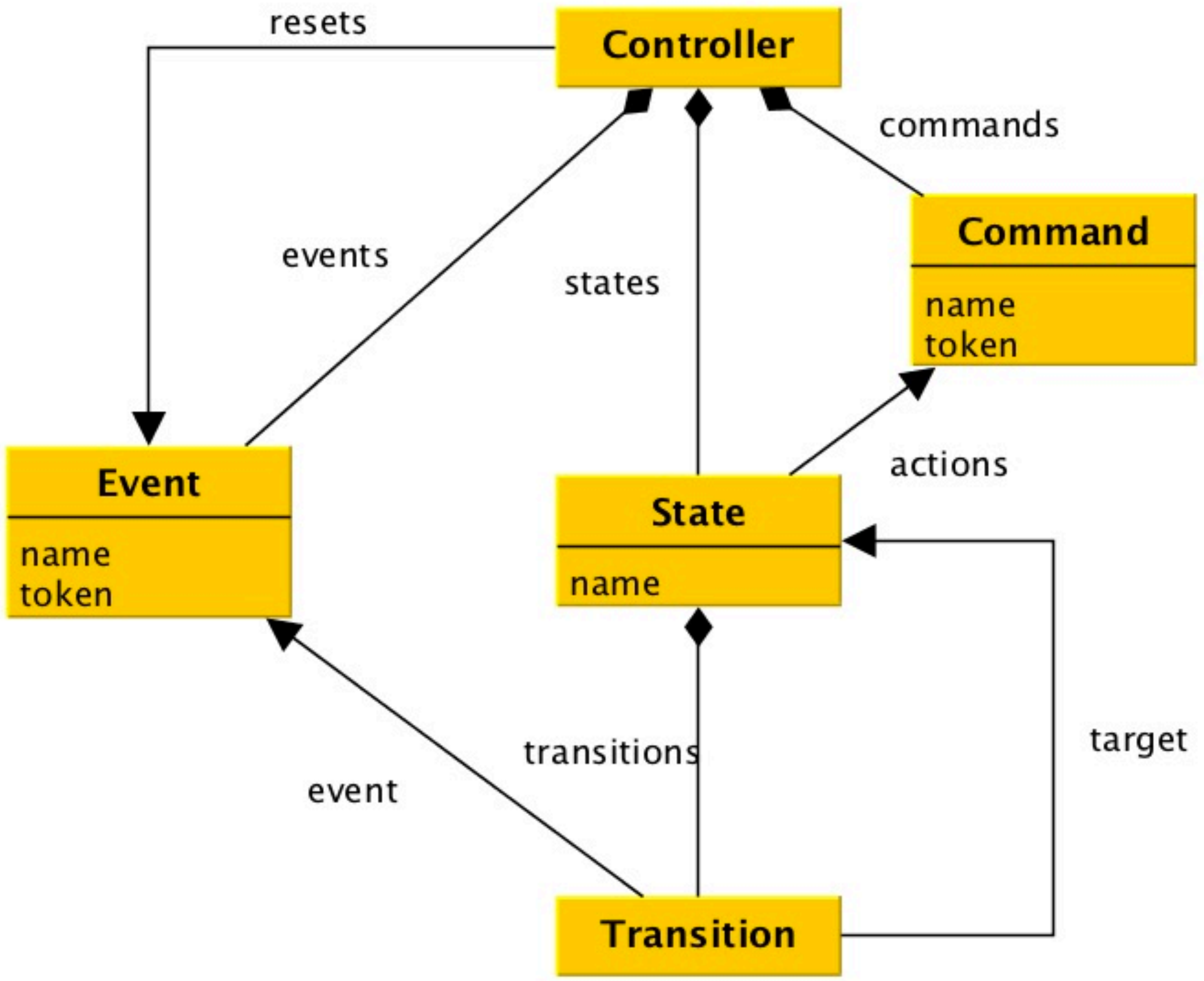
Compiler pipe line

- Simplification:
    - desugar: unless (x) S -> if (!x) S
    - “lowering”: if (x) S -> if (x) S else ;
- Source level optimization
    - e.g. if (true) S -> S, 0 * x -> 0, etc.
- Intermediate representation
    - example: SSA

Source code generation

- AST based: transform trees, format at the end
- String-based: generate source code directly
  - e.g. using template frameworks

State machines


resets
Controller
commands
events
states
Command
name
token
actions
Event
name
token
State
name
transitions
target
event
Transition