

Software Design & Modelling

Software Systems – Design – L1T3

Dr. Vadim Zaytsev aka @grammarware, November 2020

UNIVERSITY
OF TWENTE.

Fm
Formal
Methods
& Tools



Software System

- a system
 - of intercommunicating components
 - based on software
 - forming part of a computer system
 - (a combination of hardware and software)
 - in achieving a common goal
- cf. cyber-physical system

Design of Software Systems

Design of Software

- Design of Software
 - input: requirements spec / architecture description
 - output: prescriptive blueprint / plan
- Design of Software Systems
 - input: problem
 - output: models

(Software) Modelling

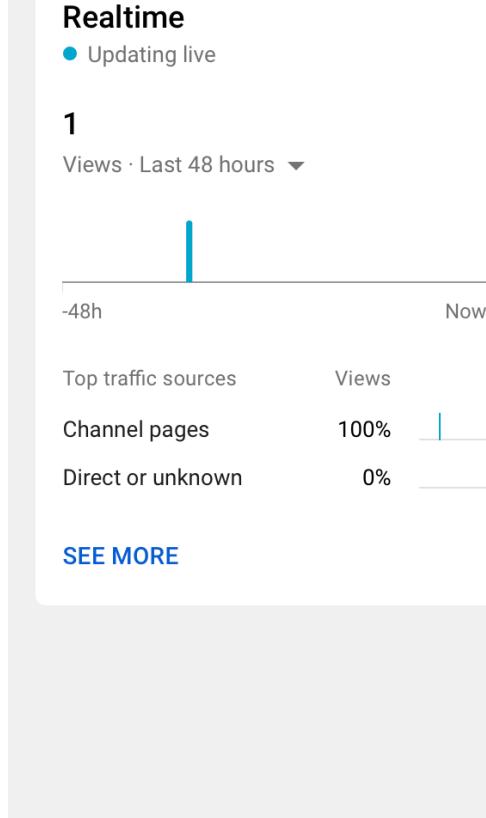
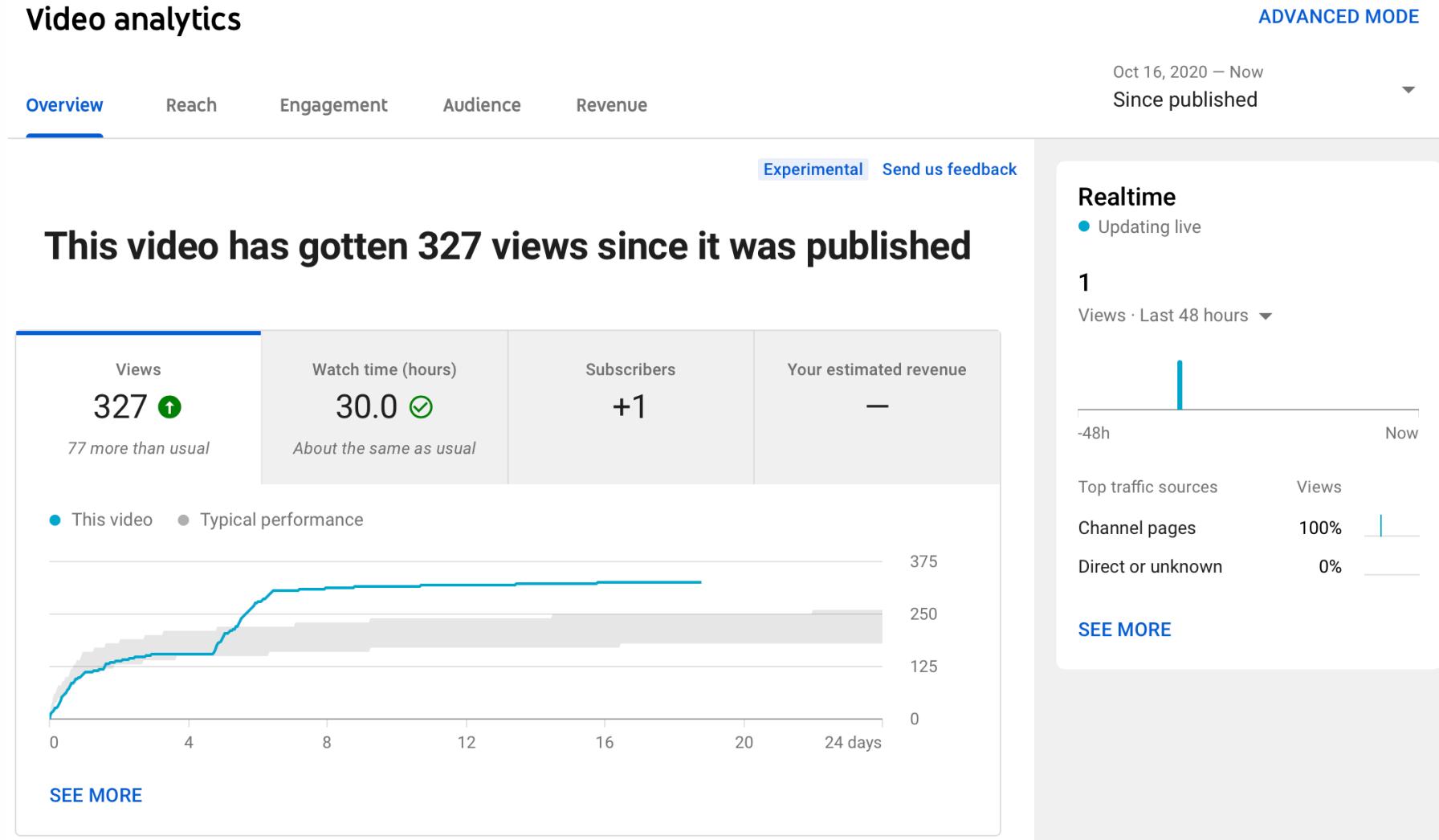
- Model is
 - a simplified representation
 - of a part of the world
 - from a particular view
- Main features:
 - mapping
 - reduction
 - pragmatic

All Models are Wrong



Software Model: YouTube Analytics

Video analytics

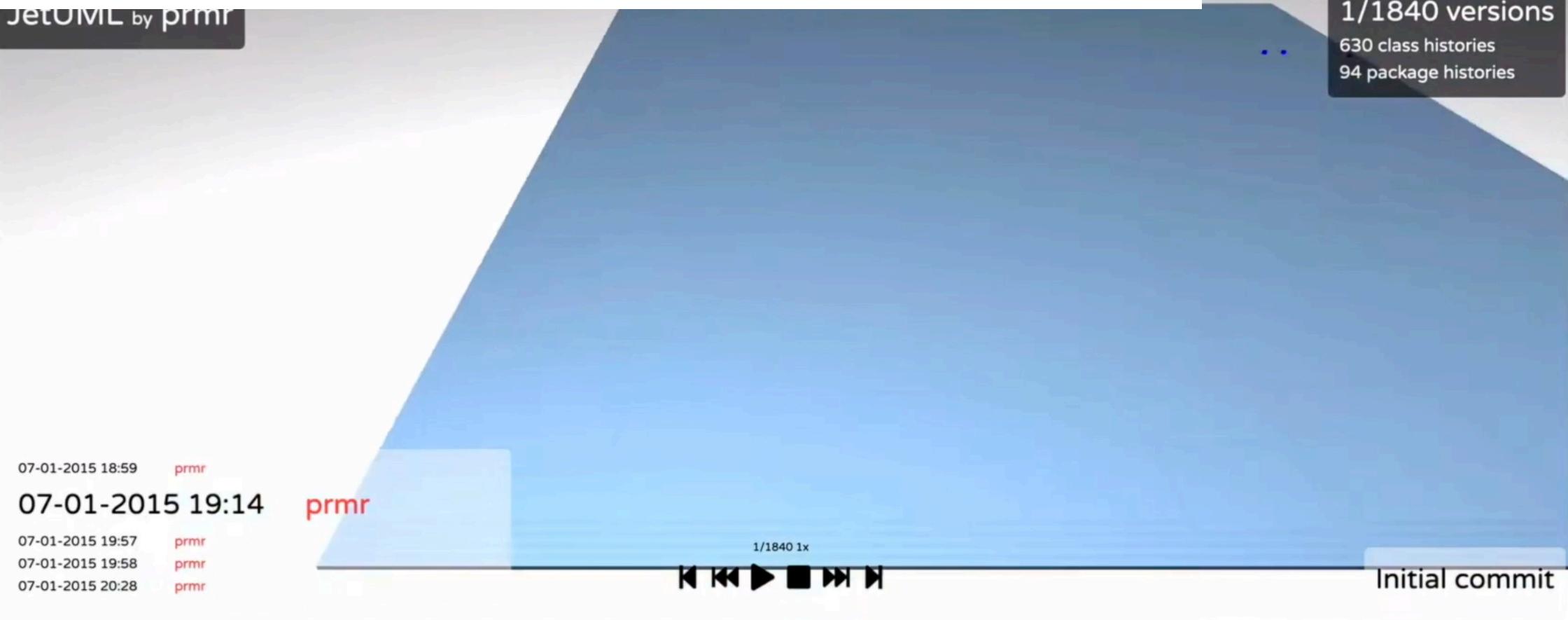


Software Model: Code City

JetUML by prmr

all ▾ Settings ⚙

1/1840 versions
630 class histories
94 package histories



Software Model: Test Coverage

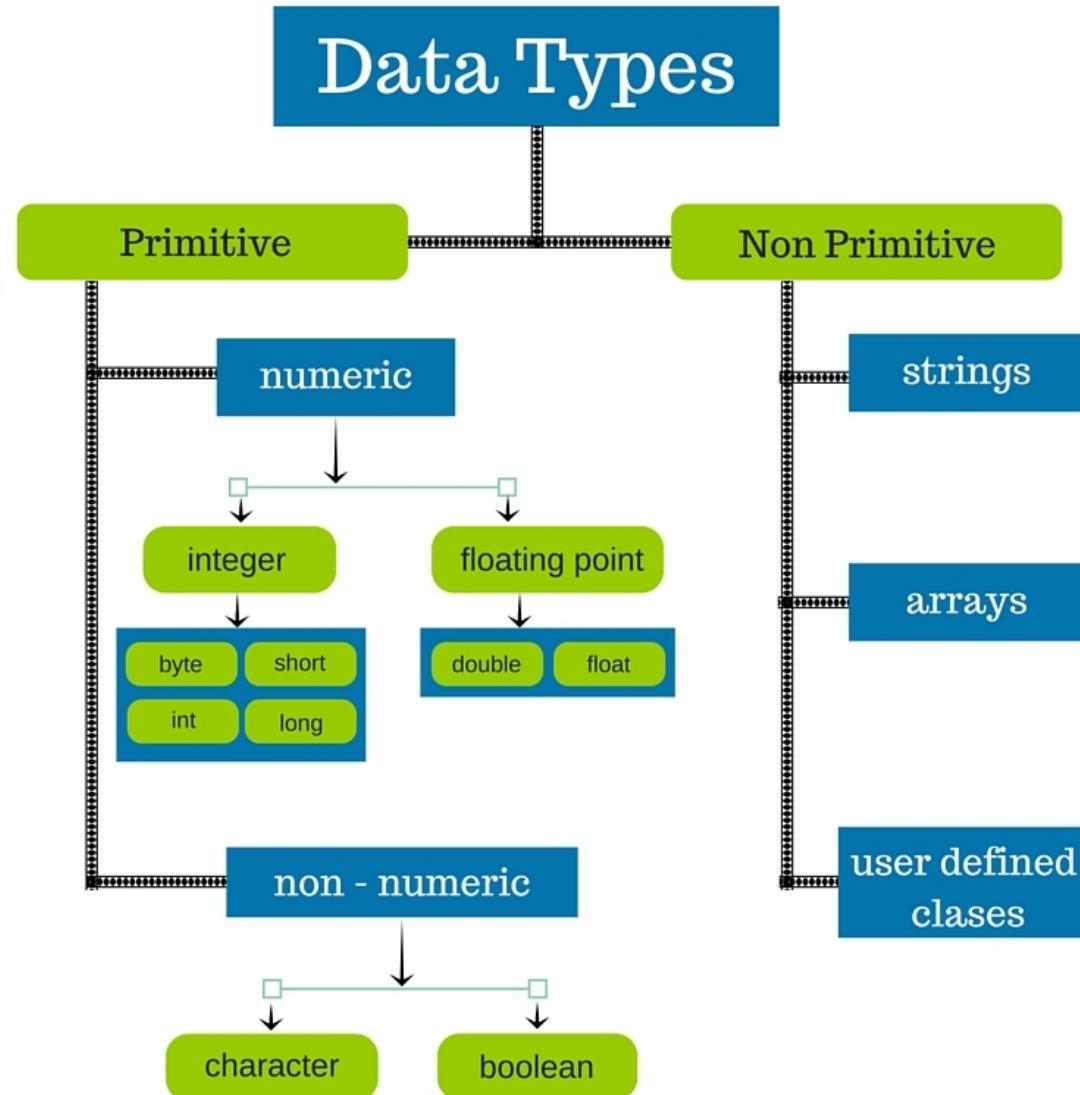
The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** eclipse-workspace - JUnitDemo/src/demo/Counter.java - Eclipse IDE
- Left Sidebar:** Package Explorer (CounterTests [Runner: JUnit 5] (0.674 s))
 - canBeInitialised() (0.022 s)
 - isEmptyByDefault() (0.003 s)
 - canRunAddWord() (0.001 s)
 - canAddTwoSameWords() (0.004 s)
 - topStaysAtThree1() (0.029 s)
 - topStaysAtThree2(int) (0.020 s)
 - topStaysAtThree3(int) (0.003 s)
 - canAddWord() (0.001 s)
 - canAddTwoDifferentWords() (0.000 s)
 - sortsTwoByWeight() (0.002 s)
- Central Area:** Counter.java and CounterTests.java
- Code View (Counter.java):**

```
1 package demo;
2
3 import java.util.ArrayList;
4
5 public class Counter {
6
7     private HashMap<String, Integer> dict = new HashMap<String,
8
9         public static int max(int x, int y) {
10
11         int result = x;
12
13         if (x < y)
14             result = y;
15
16         return result;
17
18     }
19
20     public List<Entry<String, Integer>> top() {
21         var result = new ArrayList(dict.entrySet());
22         result.sort(weightComparator);
23
24         if (result.size() > 3)
25             return result.subList(0, 3);
26     }
27 }
```
- Bottom View:** Coverage (CounterTests (4 Nov 2020 17:09:04))

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
JUnitDemo	96.3 %	261	10	271

Software Model: Type System



Software Modelling

Informal Modeling

- Emergent graphical notations support communication, collaboration, & understanding.
- Postponement of any structural limitation on the sketched models as long as possible.

Low-Code Software Development

- Inclusion of non-professional developers.
- SE with minimal up-front investment.
- Costs rise in proportion to business value.



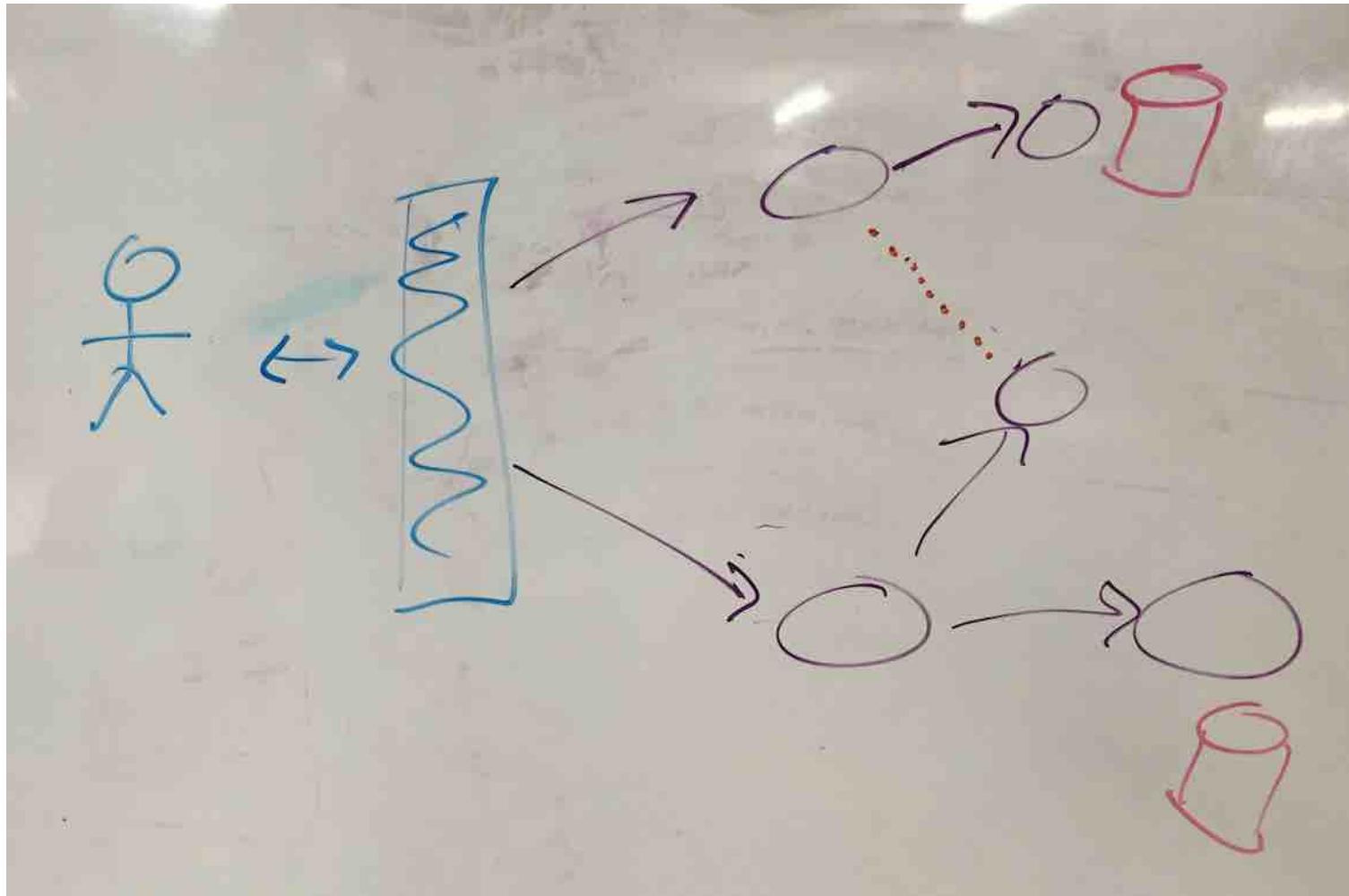
Open Challenges

- AI-based MDE Techniques
- Multi-Paradigm Modelling
- Model for Modelling
- Model Management

Model-Based Systems Engineering

- Communication, understanding and development across experts.
- Stability of languages, methods and tools.
- Industry can rely on educators training future employees.

Natural Models



Formal Models

(E-COND-TRUE)

$$\frac{}{\langle \text{if}(\text{true}, e_1, e_2), q \rangle \mapsto \langle e_1, q \rangle}$$

(E-TICK)

$$\frac{}{\langle \text{tick}(c, e_0), q \rangle \mapsto \langle e_0, q - c \rangle}$$

(E-MATD-VAL)

$$\frac{v_0 \in \text{Val} \quad v_1 \in \text{Val} \quad \dots \quad v_{m_j} \in \text{Val}}{\langle \text{matd}(C_j(v_0, \langle v_1, \dots, v_{m_j} \rangle), \overrightarrow{C_j(x_0, \langle x_1, \dots, x_{m_j} \rangle).e_j}), q \rangle \mapsto \langle [v_0, v_1, \dots, v_{m_j}/x_0, x_1, \dots, x_{m_j}]e_j, q \rangle}$$

(E-APP-Abs)

$$\frac{v_2 \in \text{Val}}{\langle \text{app}(\lambda(x.e_0), v_2), q \rangle \mapsto \langle [v_2/x]e_0, q \rangle}$$

(E-COND-FALSE)

$$\frac{}{\langle \text{if}(\text{false}, e_1, e_2), q \rangle \mapsto \langle e_2, q \rangle}$$

(E-MATP-VAL)

$$\frac{v_1 \in \text{Val} \quad v_2 \in \text{Val}}{\langle \text{matp}(\text{pair}(v_1, v_2), x_1.x_2.e_1), q \rangle \mapsto \langle [v_1, v_2/x_1, x_2]e_1, q \rangle}$$

(E-LET-VAL)

$$v_1 \in \text{Val}$$

$$\frac{}{\langle \text{let}(v_1, x.e_2), q \rangle \mapsto \langle [v_1/x]e_2, q \rangle}$$

(E-APP-FIX)

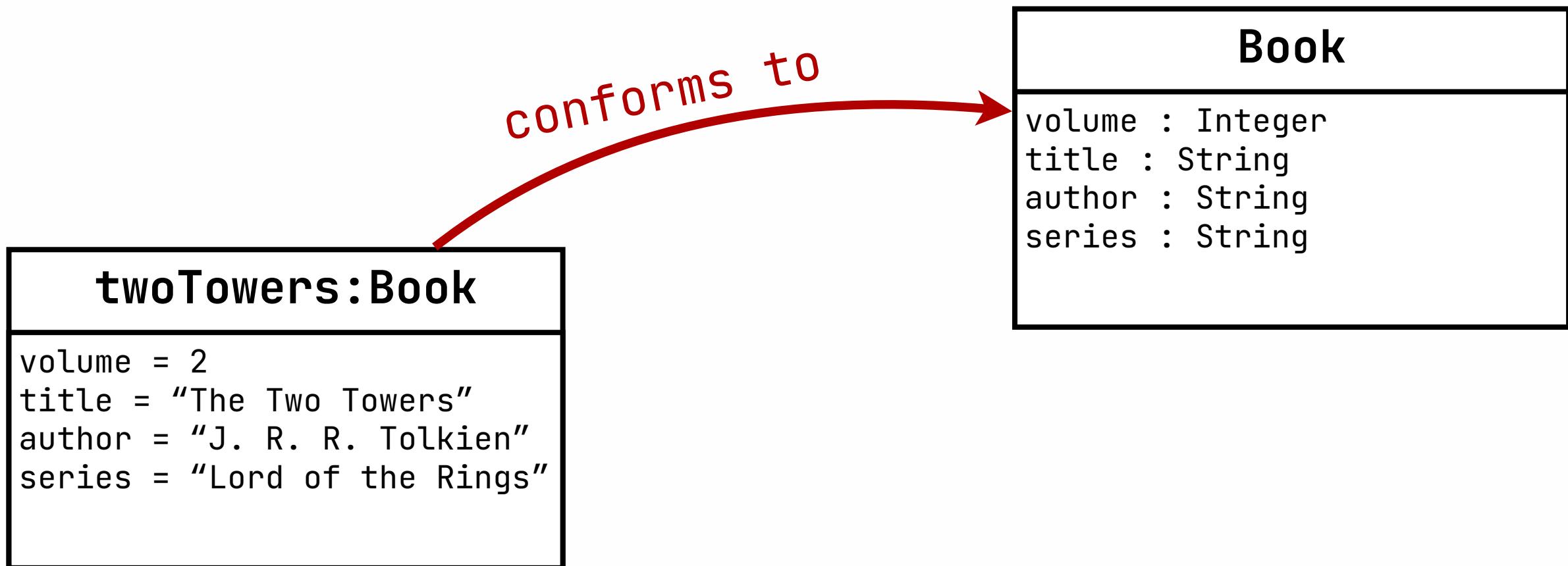
$$\frac{v_2 \in \text{Val}}{\langle \text{app}(\text{fix}(f.x.e_0), v_2), q \rangle \mapsto \langle [\text{fix}(f.x.e_0), v_2/f, x]e_0, q \rangle}$$

Mental Models

- Code is a social construct
- Devs' understanding of a problem
 - incomplete, imprecise, flawed
- Code is a consequence of that
- Complexity accumulates over time
 - [[Lehman1980](#)] [[Ousterhout2018](#)]
- Incremental \Rightarrow hard to control
- Code quality is judged by readers



Conformance



Conclusion

- Fight complexity with models
- Learn UML diagrams
- Flip the classroom
- Request/vote for extra topics
- Unveil software engineering

Topics/slides Disclaimer

- Good ✓

- watch before Q&A
- embrace reality
- try out at labs
- ask for feedback
- apply to project
- dig deeper
- recall from slides

- Bad ✗

- slides over videos
- assumptions
- blanks
- timing

