Flipped Top-Down is Systematic Bottom-Up

Dr. Vadim Zaytsev aka @grammarware

EduSymp 2015

UvA MSc BSc Inf

- ✓ Academic Skills
- ✓ Introduction Programming
- ✓ Architecture & Computer Organisation
- ✓ Programming Languages
- ✓ Discrete Maths & Logic
- ✓ Webprogramming & Databases
- ✓ Linear Algebra
- ✓ Data Structures
- ✓ Automata & Formal Languages
- ✓ Operating Systems
- ✓ Multimedia

https://datanose.nl/#timetable(BSc IN|1,36,0)

Input conditions

- ✓ No MDE
- ✓ No need for MDE
- ✓ Varying levels
- ✓ High expectations
- ✓ Enthusiasm

General setup?

- ✓ 1 month, full time
- ✓ Mostly hacking/engineering
- ✓ Some lectures
- ✓ Some practice hours
- ✓ Week 1: lectures + idea pitch
- ✓ Week 2: MVP + audit
- ✓ Week 3: progress + audit
- ✓ Week 4: dry run + final demo
- ✓ Flipped + mandatory questions

Onderwijsgek, Empty classroom, 2011. CC-BY-SA.

Lecture 1: Intro

- ✓ History of computing?
- ✓ Wicked problems
- ✓ CS vs SE?
- ✓ Life cycle of software?
- ✓ Success stories?
- ✓ . . .

Lecturer is the model

- ✓ You can do . . . as I did
  - ✓ hacking -> system prog & rev eng
  - ✓ railways -> simulations
  - ✓ maths -> formal methods
  - ✓ web -> sep of concerns & mappings
  - ✓ data rec -> databases
  - ✓ query model -> AI
  - ✓ MDE -> OO & …
  - ✓ legacy -> mainframes & legacy
  - ✓ . . .

Lecture 2: Project

- ✓ “Form groups [illegible]”
- ✓ (Minimise grad[illegible] fort)
- ✓ (Larger task[illegible] es[illegible] simplicity)
- ✓ “Wait for[illegible]

Scrum

- ✓ Software dev’t process model
- ✓ Agile manifesto
- ✓ Weekly stand-up meetings
- ✓ Planning poker
- ✓ Roles: scrum master, product owner
- ✓ Emergent roles: backend/frontend, merge&deploy, API design, . . .

Lecture 3: Inside SE

- ✓ Examples o ctivities
- ✓ Case study?
- ✓ Highlights? Wh es?
- ✓ In 2014:
  - ✓ maintena
  - ✓ startups

SWEBoK

- ✓ Do not explain parts
- ✓ Explain everything
- ✓ SWEBoK is the domain model
  - ✓ software construction
  - ✓ software testing
  - ✓ software maintenance
  - ✓ . . .
- ✓ Also, a SWEBoK-based MSc programme

Lecture 4: Paradigms

- ✓ “Programmi Paradigms” course
  - ✓ Recall J
  - ✓ Look at LIS
  - ✓ Look at Prolo
  - ✓ Look at Fort
  - ✓ Look at XS
  - ✓ Look at ltalk
  - ✓ Look at Javascript
  - ✓ . . .

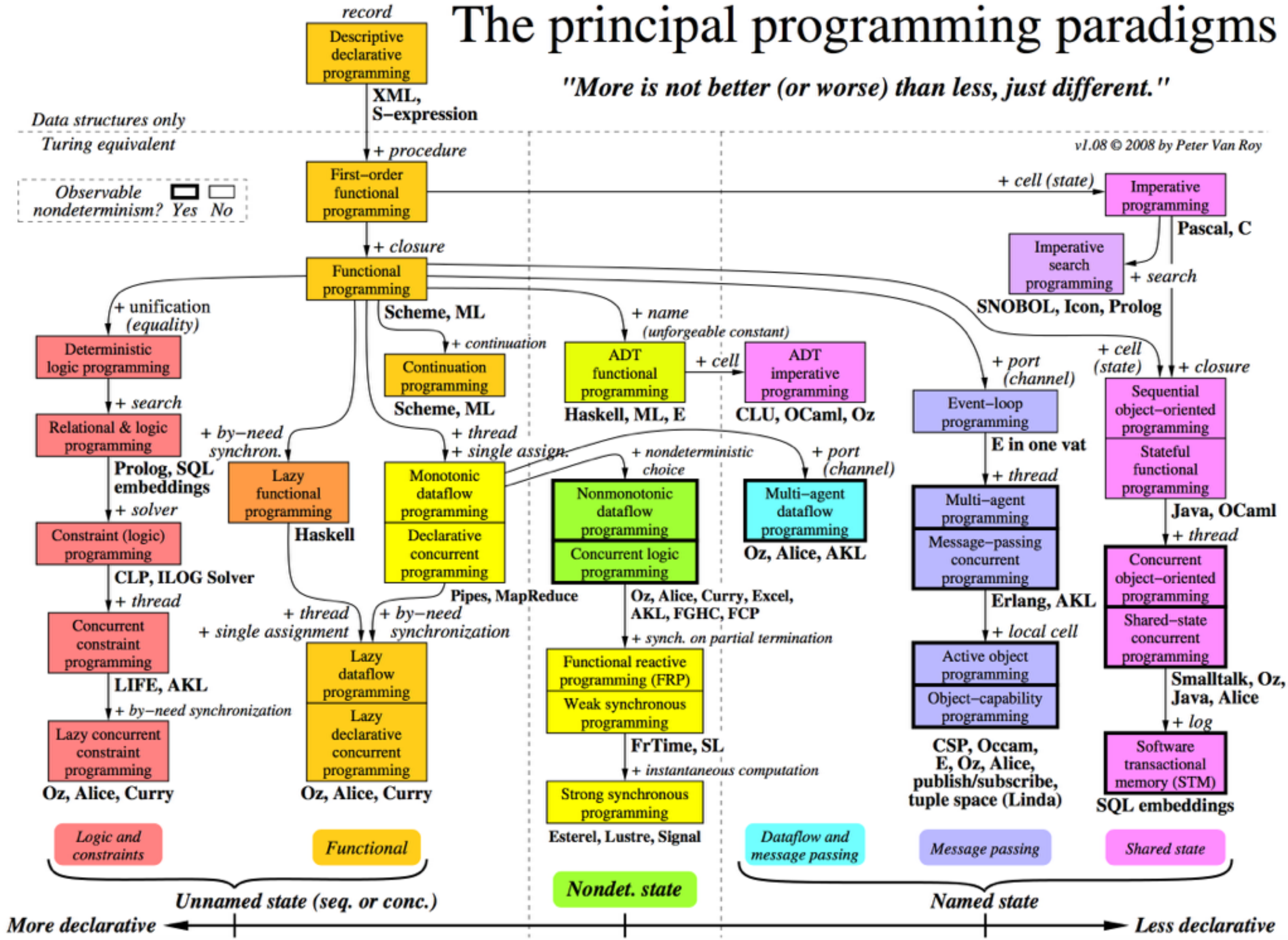
ALL Paradigms

- ✓ Show ALL paradigms at once
- ✓ Connected in a megamodel
- ✓ Renarrate the megamodel

Peter Van Roy, https://www.info.ucl.ac.be/~pvr/paradigms.html

The principal programming paradigms

"More is not better (or worse) than less, just different."



Lecture 4’: Practices

- ✓ Go to considered harmful
- ✓ Liberal on input, conservative on output
- ✓ Favour composition over inheritance
- ✓ Less is more
- ✓ Keep it simple
- ✓ Don’t repeat yourself
- ✓ Loops affect performance
- ✓ Respect naming conventions
- ✓ Put opening curlies on the next line
- ✓ . . .

Patterns as models

- ✓ Good practices eventually become
  - ✓ Design patterns
  - ✓ Architectural patterns
  - ✓ Language constructs
- ✓ Bad practices can be detected
  - ✓ Code smells
  - ✓ Antipatterns
  - ✓ Convention violations

Lecture 5: Choose!

- ✓ Search-based SE
- ✓ Software language engineering
- ✓ Language X / framework Y
- ✓ Practical FP
- ✓ Testing
- ✓ Metaprogramming
- ✓ Reverse engineering
- ✓ Cracking
- ✓ Nothing

Functional Thinking

- ✓ Homework
  - ✓ Neal Ford’s video
- ✓ Classroom
  - ✓ Expression problem
- ✓ Industrial examples
- ✓ Code
  - ✓ in Java 8, Haskell, Scala, Clojure, Groovy, F#, Python, Racket, Erlang, Elixir

Neal Ford, Functional Thinking: Paradigm over Syntax, 2014.

Finale

- ✓ Active involvement
- ✓ High grades
- ✓ Product delivered
  - ✓ by each of 7 teams
- ✓ Good evaluation
- ✓ Big effort
  - ✓ 1000+ intermediate grades

Techniques

- ✓ Goal: introduce SE
  - ✓ lecturer as the model
  - ✓ SWEBoK as the domain model
- ✓ Show relevance of MDE
  - ✓ constantly facing the complexity
- ✓ Connect to the audience
  - ✓ constant feedback
- ✓ Auditors for projects
  - ✓ students >>> lecturers

Lessons learnt

- ✓ Replicable experience?
  - ✓ certainly demanding
  - ✓ could have been harder
- ✓ Refined material?
- ✓ Ad hoc lectures?
- ✓ “Tweetable lectures” failed?
- ✓ Feedback?