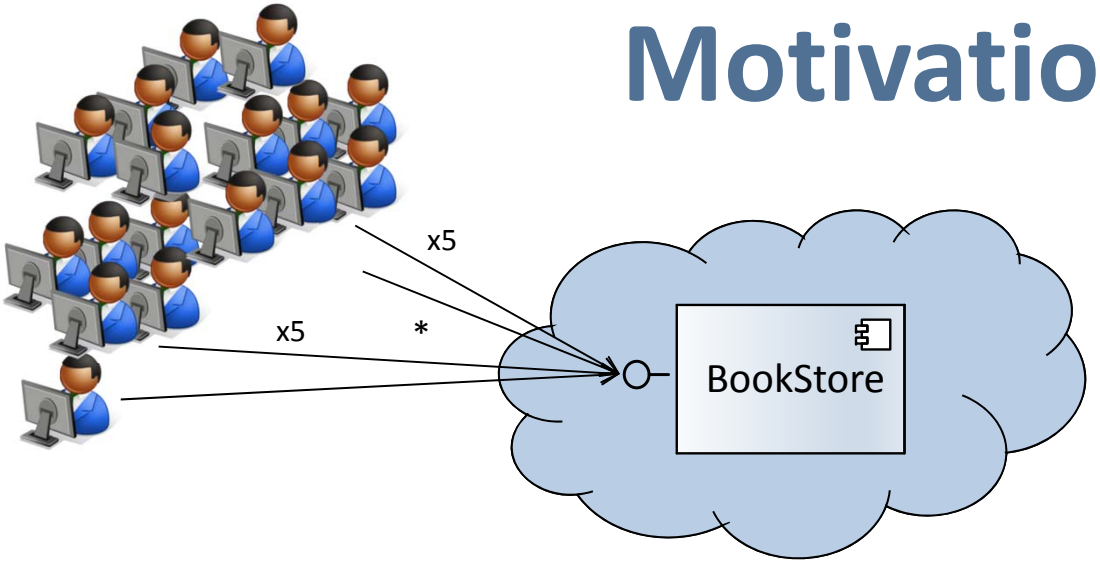
Static Spotter for Scalability Anti-Patterns Detection

Jinying Yu and **Goran Piskachev**

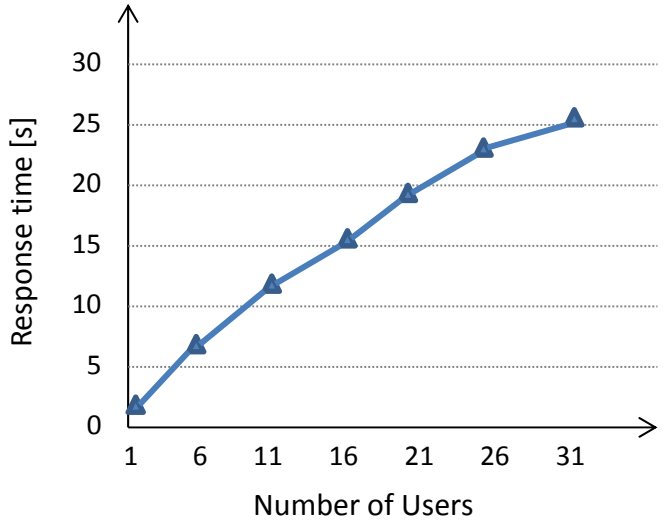
TECHNISCHE UNIVERSITÄT CHEMNITZ

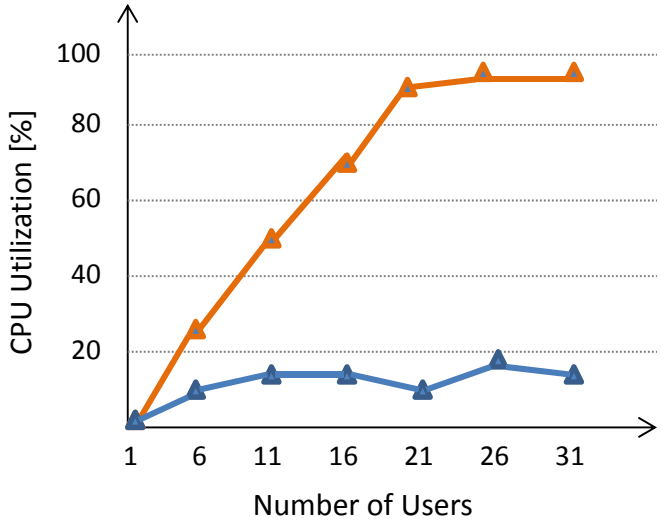
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

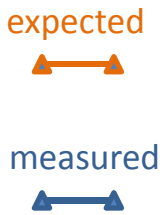
Motivation



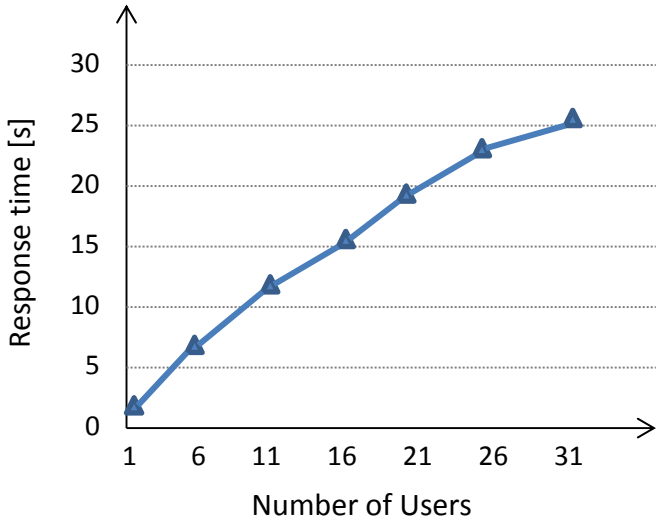

System architect

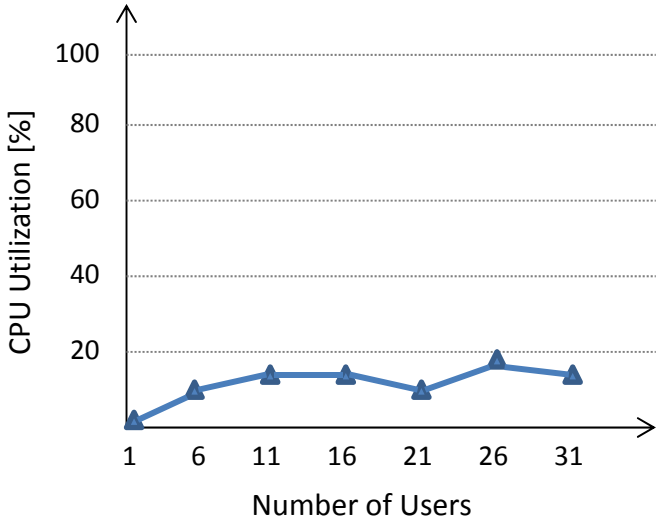




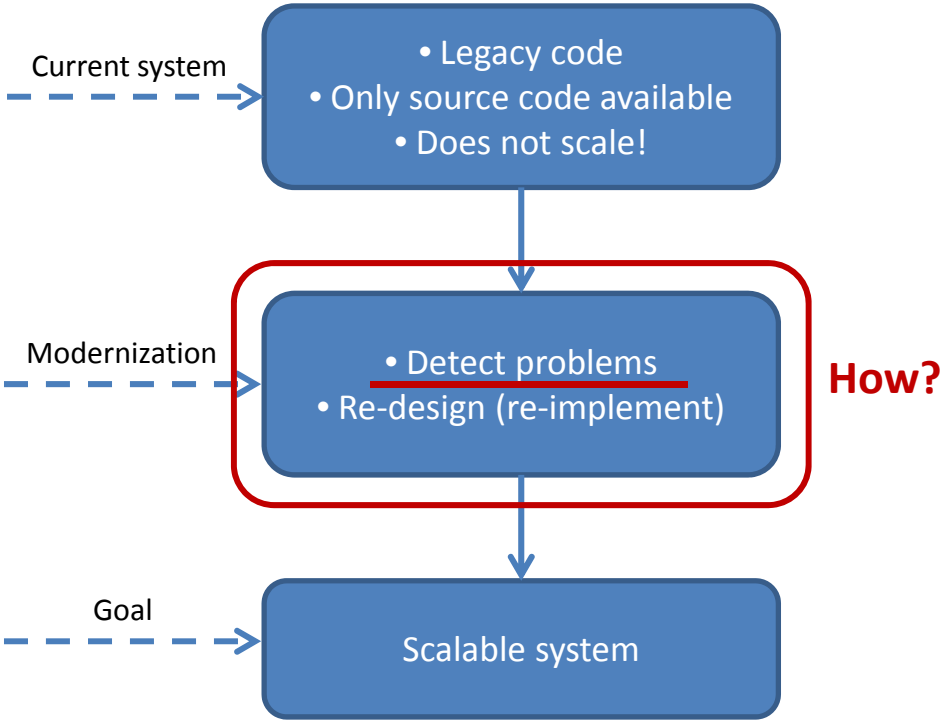


Problem





System architect



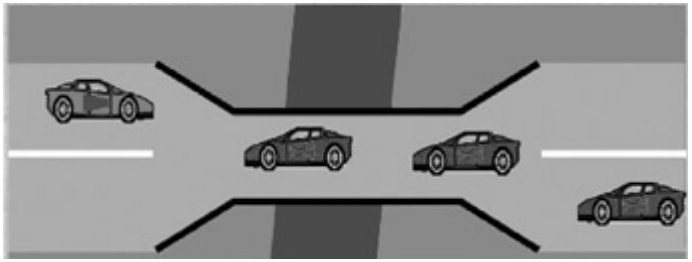
Solution

Problem: **Scalability Problems** → **Static Spotter**

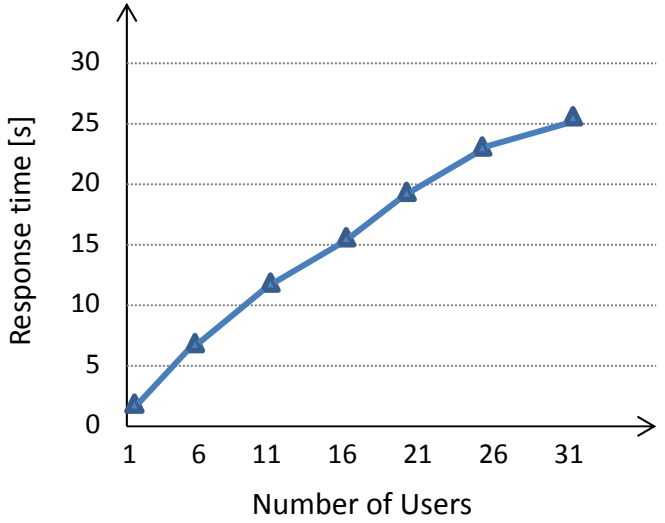
Outline

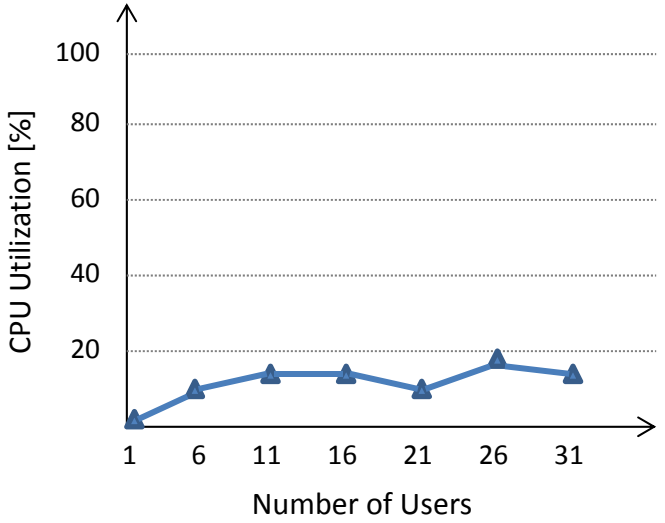
- 1-lane-bridge
- Static Spotter: Overview
- Static Spotter: Specification
- Static Spotter: Detection

1-lane-bridge in traffic

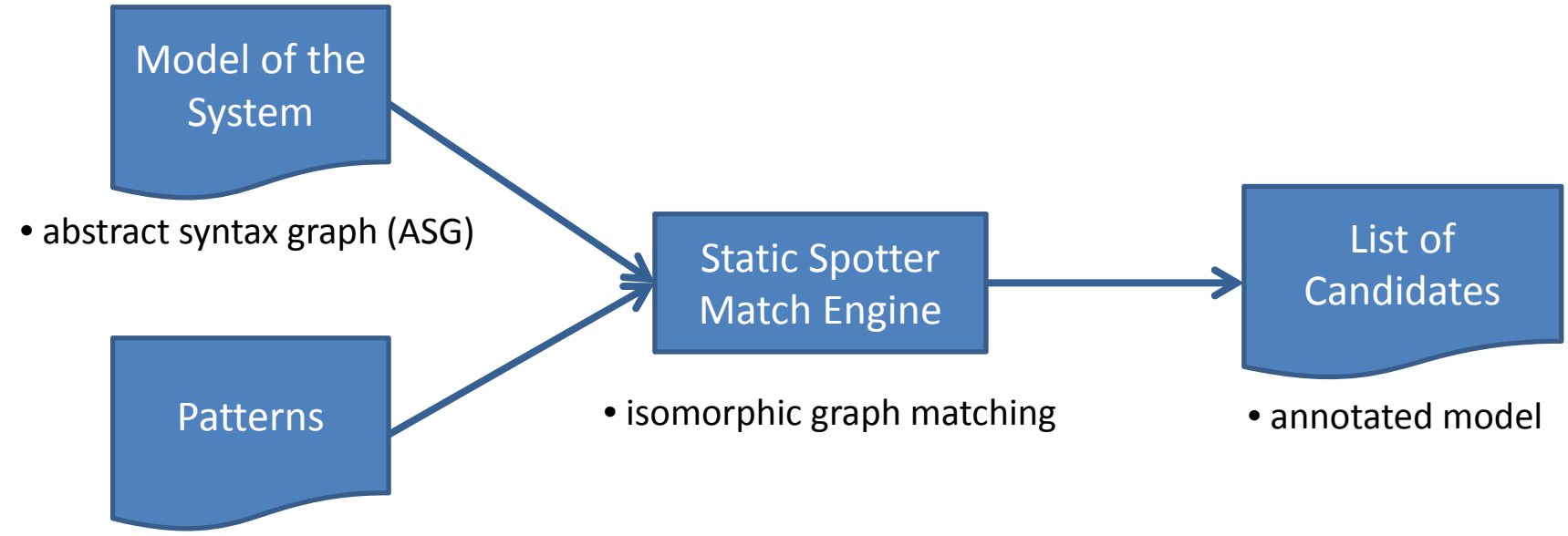


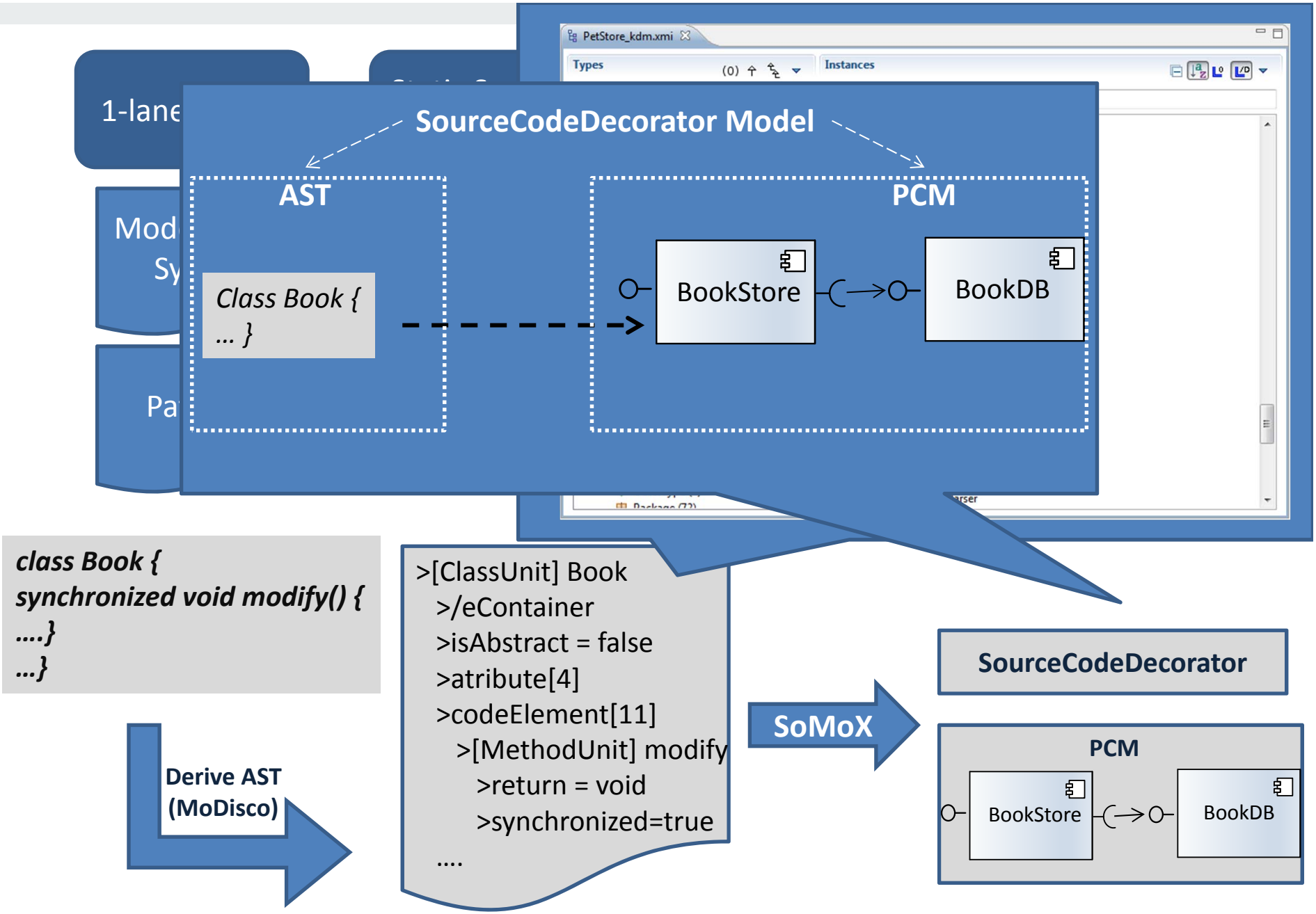
In Software:
- passive resource limits the concurrency
- e.g. synchronization points, database locks
- may cause congestion





1-lane-bridge | Static Spotter: Overview | Static Spotter: Specification | Static Spotter: Detection

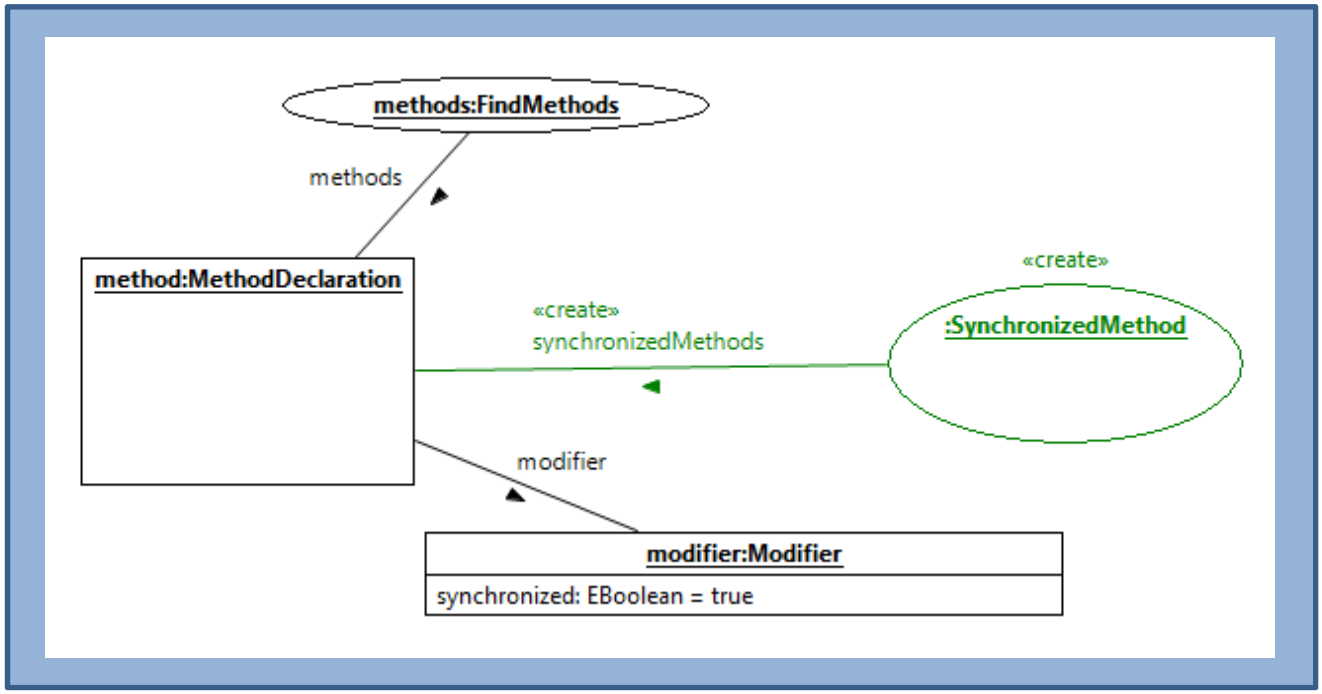



1-lane
Mod
Sy
Pa
PetStore_kdm.xmi
Types
Instances
SourceCodeDecorator Model
AST
PCM
Class Book {
... }
BookStore
BookDB
class Book {
synchronized void modify() {
....}
...}
>[ClassUnit] Book
>/eContainer
>isAbstract = false
>atribute[4]
>codeElement[11]
>[MethodUnit] modify
>return = void
>synchronized=true
....
Derive AST
(MoDisco)
SoMoX
SourceCodeDecorator
PCM
BookStore
BookDB

1-lane-bridge | Static Spotter: Overview | Static Spotter: Specification | Static Spotter: Detection

How to detect candidates for 1-lane-bridge?
e.g. synchronization spots
- synchronized method
- synchronized block

Graphical DSL for patterns specification



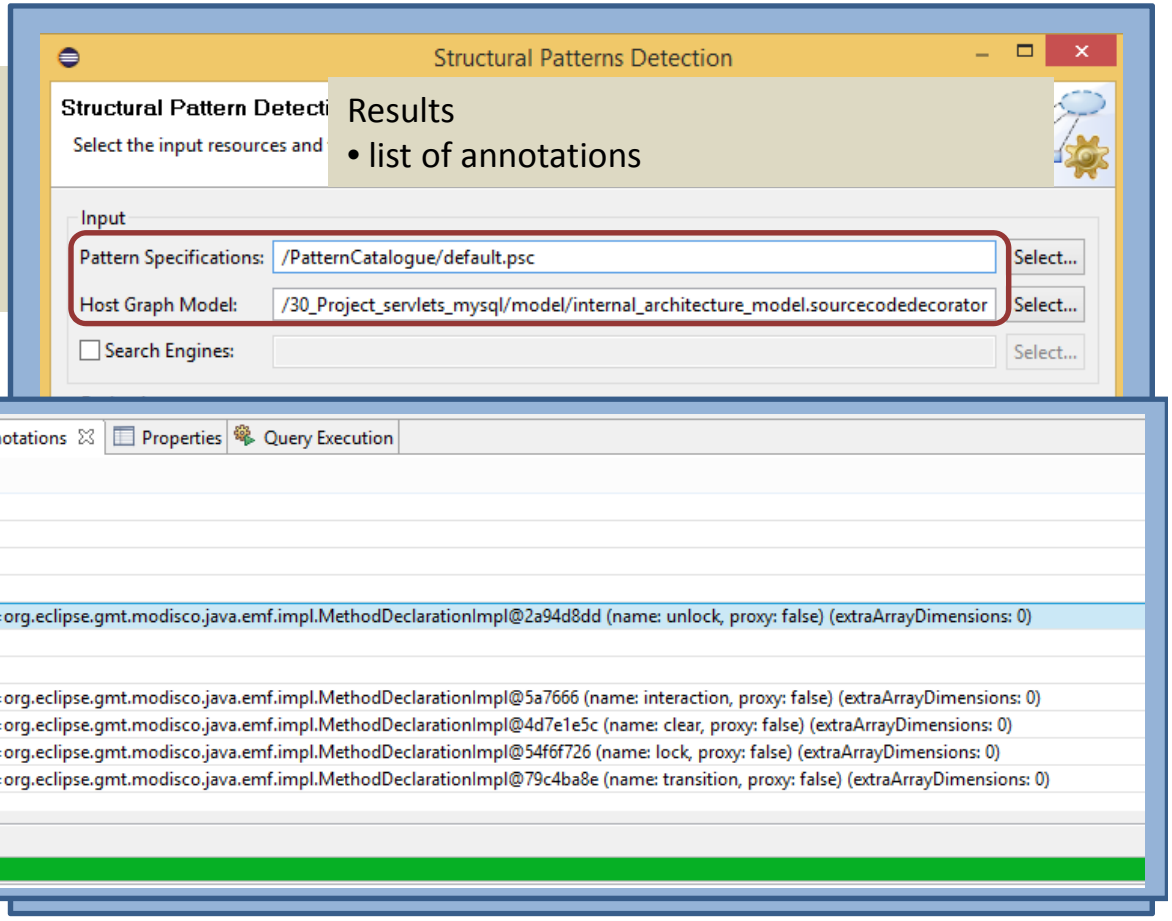
1-lane-bridge

Static Spotter: Overview

Static Spotter: Specification

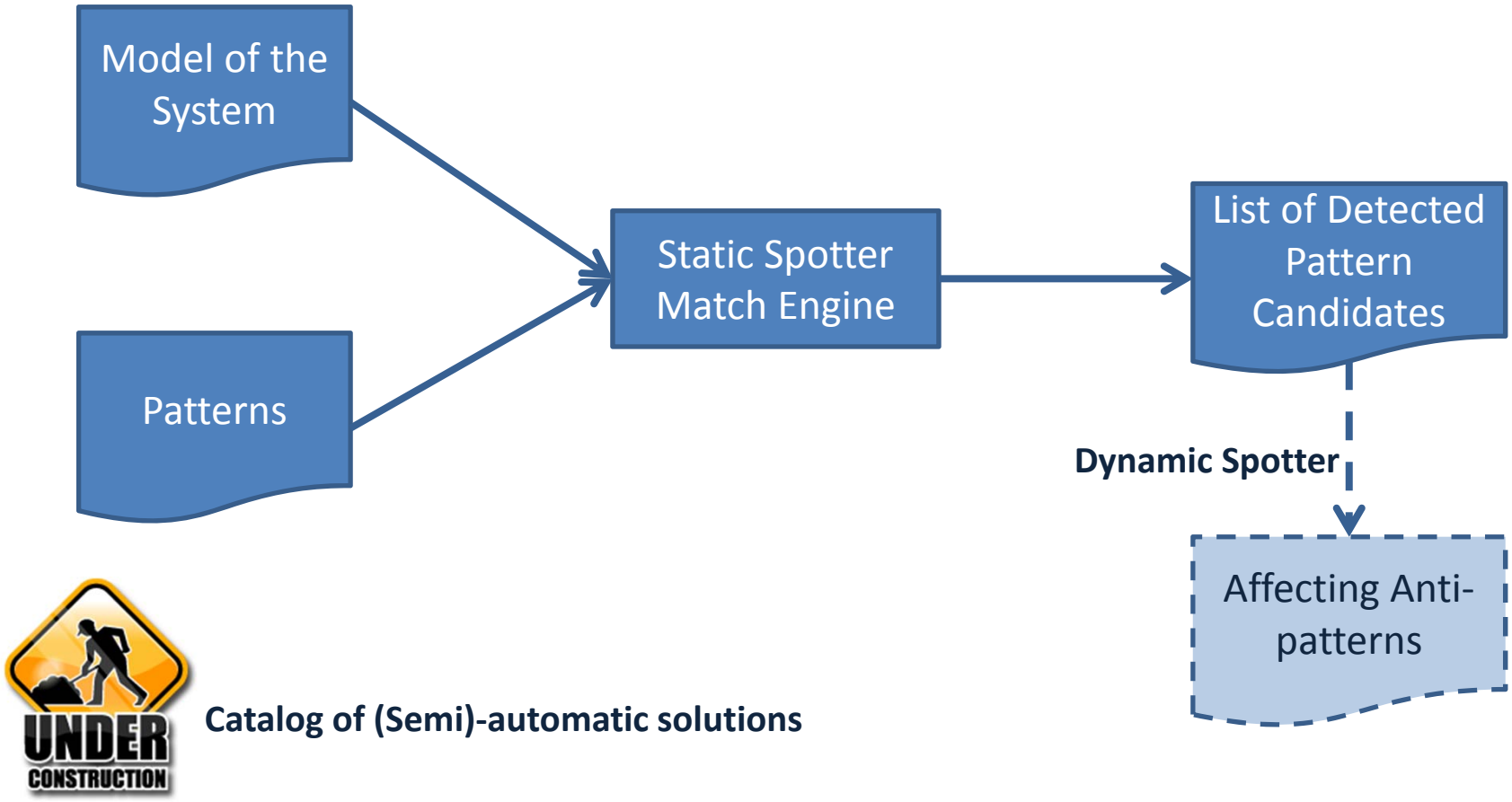
Static Spotter: Detection

Graph matching algorithm
- single root object
- the graph transformation rules are leveled
- combined a bottom-up strategy and a top-down strategy



Tool Demo!

Conclusion



Catalog of (Semi)-automatic solutions

Links

- CloudScale project: http://www.cloudscale-project.eu/
- Tool installation: https://code.google.com/p/reclipse-emf/

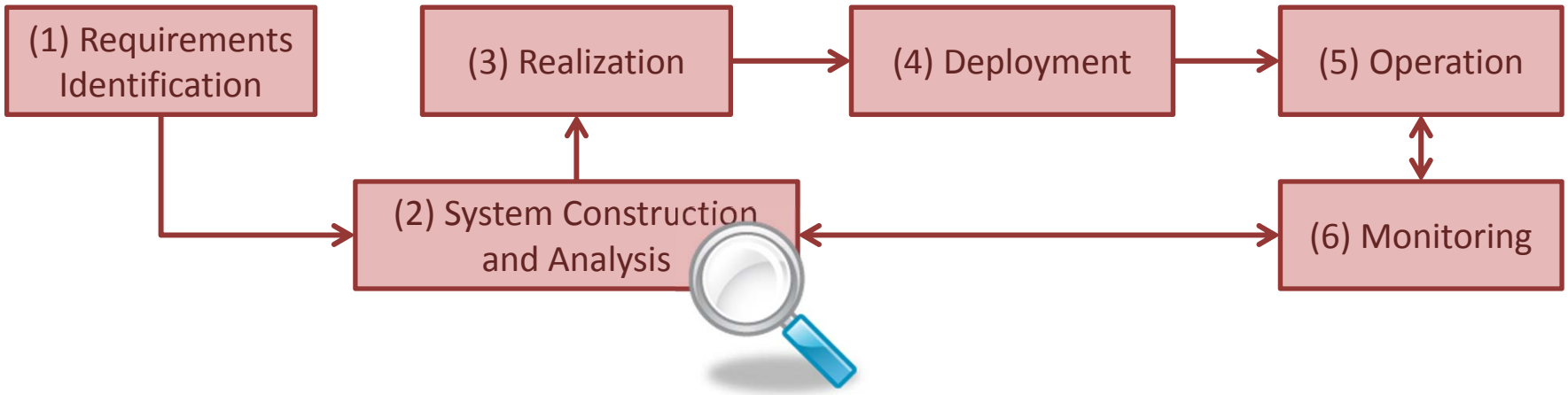
Back-up slides

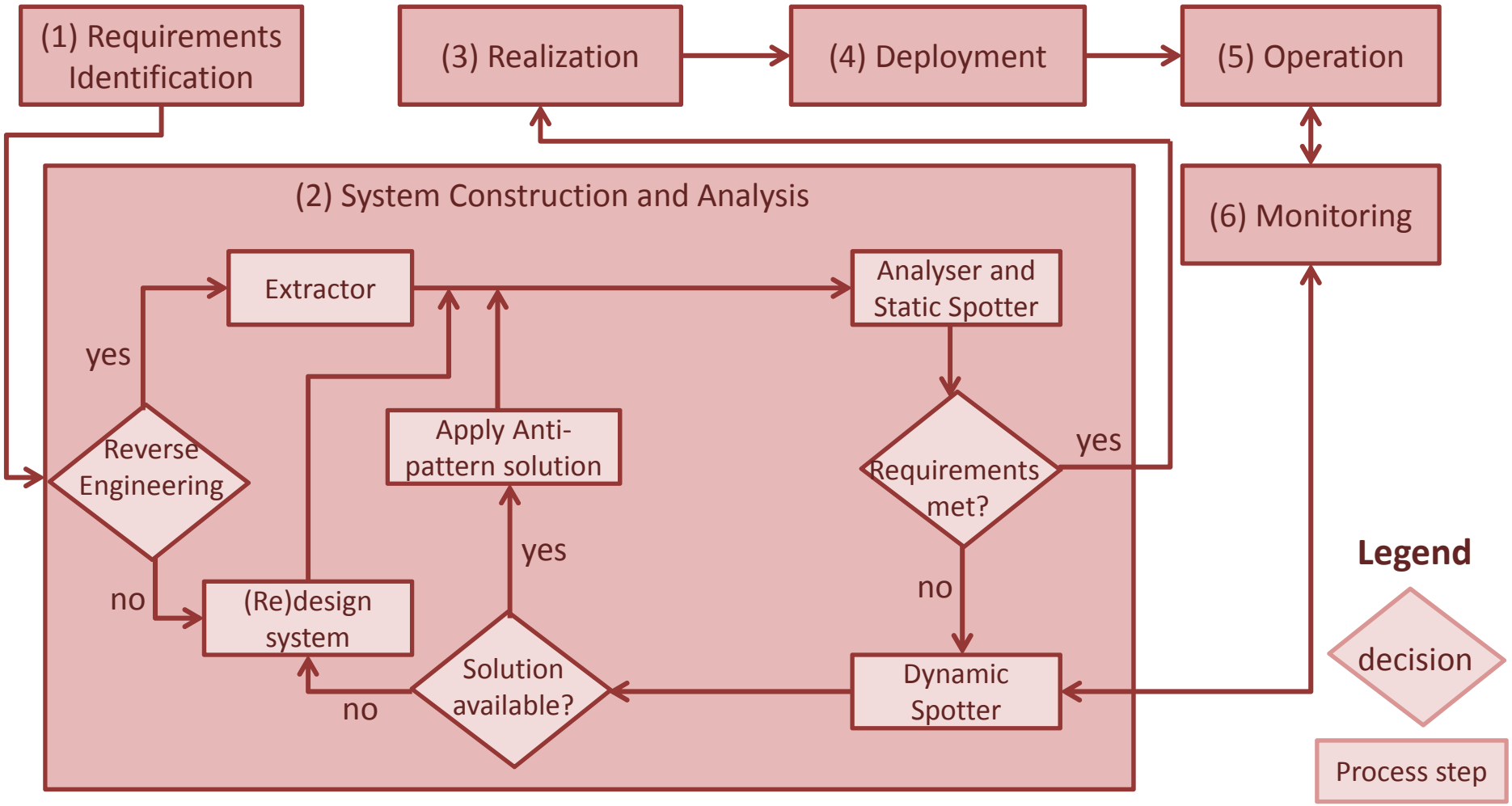
Static Spotter: Pattern Matching

Rule leveling:
A rule depending only on objects in the initial ASG gets number 1. A rule depending on other rules, i.e., whose definition includes annotations created by other rules, gets a higher number consistent with the natural topological order of the rules. Rules included in cycles concerning their dependencies get the same level number and are marked as recursive.

Bottom-up then top-down strategy:
After parsing the source code to create the ASG, the analysis starts in bottom-up mode. Initially, all ASG objects schedule level 1 rules, i.e., those depending on ASG objects only.
When a rule that depends on other rules cannot be applied in bottom-up mode, the algorithm switches to top-down mode, which uses a separate *top-down priority queue. The top-down strategy* tries to make the other rules create the missing annotations based on currently available information.
The rule/context pair at the front of the top-down mode queue is not dequeued if the rule involved schedules other lower-level rules. Instead, pairs added to the top-down queue are queued in *ascending order of their level number. This means that the higherlevel* rule will be reconsidered after the lower-level rules on which it depends (if these succeed). Using a priority queue rather than a stack means that the top-down algorithm goes as far down the ASG as quickly as possible.
The algorithm runs in top-down mode until the top-down queue is empty or a rule in the queue fails with no alternative contexts left to explore.
The overall analysis finishes when the bottom-up queue is empty. In this case the algorithm has analysed all ASG objects and created annotations on the objects for all rules that could be applied.

1-lane-bridge | CloudScale Method | Static Spotter: Overview | Static Spotter: Specification | Static Spotter: Detection




(1) Requirements Identification
(3) Realization
(4) Deployment
(5) Operation
(6) Monitoring
(2) System Construction and Analysis
Extractor
Analyser and Static Spotter
yes
Reverse Engineering
Apply Anti-pattern solution
Requirements met?
yes
yes
no
(Re)design system
Solution available?
Dynamic Spotter
no
Legend
decision
Process step