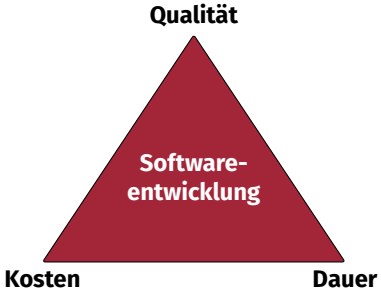
Projekte und Abschlussarbeiten

Institut für Softwaretechnik und Programmiersprachen | 14. Juli 2026

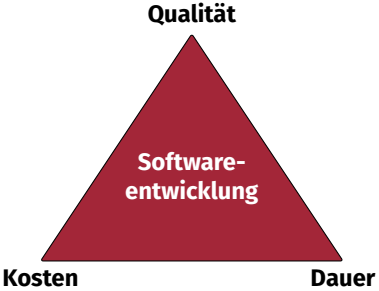
Software Engineering
Programming Languages

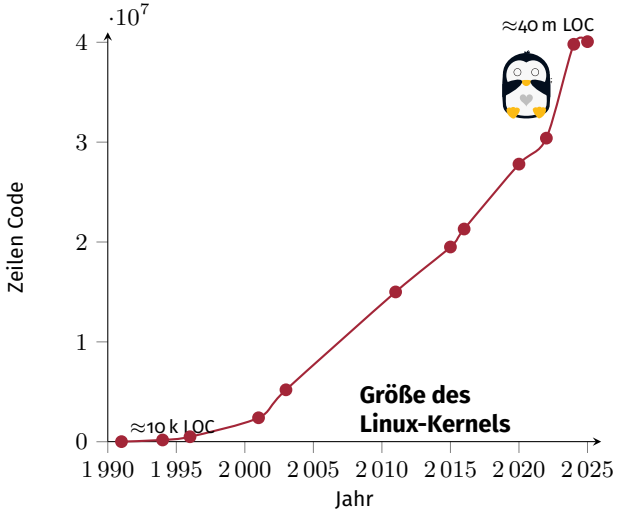
universität ulm

Software Engineering



Software Engineering



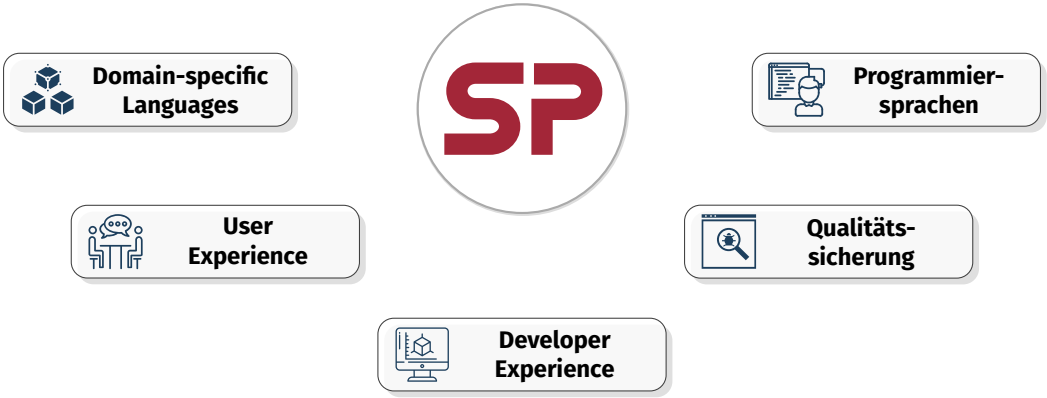


Software Engineering and Programming Languages

Prof. Tichy, Prof. Heinrich, Jun.-Prof. Wiesmayr, Dr. Raschke

Software Engineering and Programming Languages

Prof. Tichy, Prof. Heinrich, Jun.-Prof. Wiesmayr, Dr. Raschke



Software Engineering and Programming Languages

Prof. Tichy, Prof. Heinrich, Jun.-Prof. Wiesmayr, Dr. Raschke

Software Engineering and Programming Languages

Prof. Tichy, Prof. Heinrich, Jun.-Prof. Wiesmayr, Dr. Raschke

Bringt das etwas?

Bringt das etwas?

Domain-specific Languages

Programmier-sprachen

Wie gut ist es denn?

User Experience

Qualitäts-sicherung

Developer Experience

Bringt das etwas?

Domain-specific Languages

Programmier-sprachen

Wie gut ist es denn?

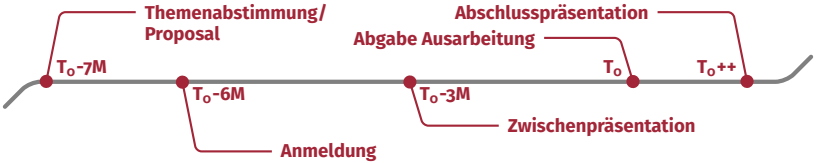
User Experience

Qualitäts-sicherung

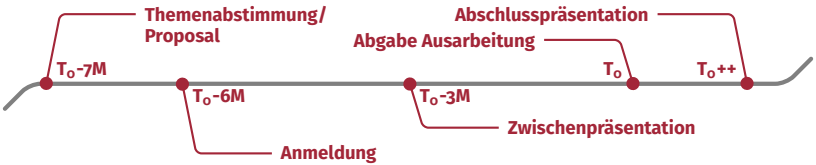
Was benötigen die Nutzer/ Entwickler:innen eigentlich?

Developer Experience

Prozess

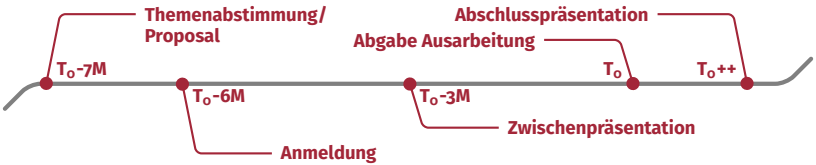

Themenabstimmung/
Proposal
Abschlusspräsentation
Abgabe Ausarbeitung
T0-7M
T0
T0++
T0-6M
T0-3M
Zwischenpräsentation
Anmeldung

Prozess



- Enge Betreuung durch Mitarbeiter:in
  - Thema
  - Methodik
  - Wöchentliche Treffen
  - Regelmäßiges Feedback
  - Kurze Wege

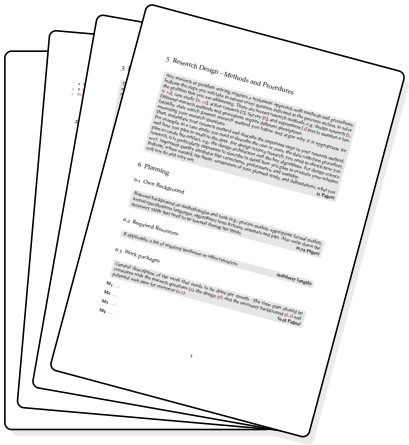
Prozess

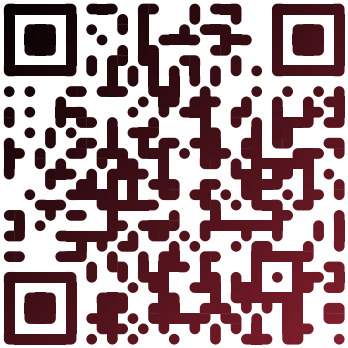


- Enge Betreuung durch Mitarbeiter:in
  - Thema
  - Methodik
  - Wöchentliche Treffen
  - Regelmäßiges Feedback
  - Kurze Wege
- Institutsbüro/Pool für Abschluss- und Projektarbeiten
- Kaffee!

Proposal

- **Ziele**
  - Abstimmung der Inhalte vor der Anmeldung
  - Risikominimierung für Sie und uns
- **Inhalte**
  - Kontext der Arbeit
  - Betrachtetes Problem / Forschungsfrage(n)
  - Stand der Forschung
  - Lösungsidee
  - Methodik und Evaluationsplan
- $\approx$ 8 Seiten Text (Reuse in der Arbeit!)





uulm.de/in/sp/teaching/topics-for-theses-and-projects/

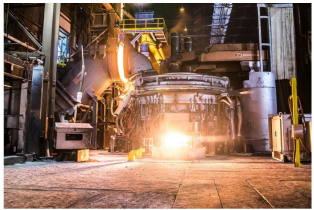
Themenübersicht

1. Software Engineering für die industrielle Automatisierung (B. Wiesmayr)
2. Static and Dynamic Program Analysis for Data Science (F. Sihler)
3. Static and Dynamic Program Analysis for Data Science (O. Gerstl)
4. Privacy and Architecture-based Simulation (P. Leber)
5. Architectural Security Analysis (L. Le)
6. Algorithm Detection (D. Neumüller)
7. Probing Program State in Code Language Models (A. Diera)

— 1 —

Software Engineering für die industrielle Automatisierung (B. Wiesmayr)

Software Engineering für die industrielle Automatisierung



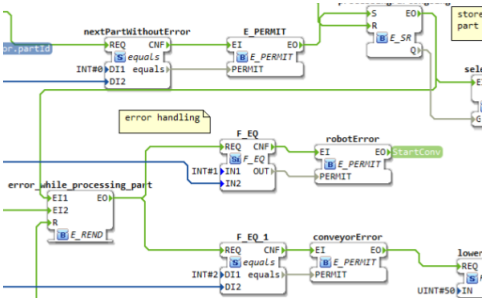
Domänenspezifische Anforderungen an die Softwareentwicklung

- Hohe Komplexität und hohe Qualitätsanforderungen
- Steuerungssoftware wird meist von Domänenexperten entwickelt
- Schwierige Integration von IT und Maschinen
- Ereignisgesteuerte Abläufe

Software Engineering muss Domänenexperten mit geeigneten Methoden und Werkzeugen unterstützen!

Integrating LLM-based Assistance into IDEs for Graphical Languages

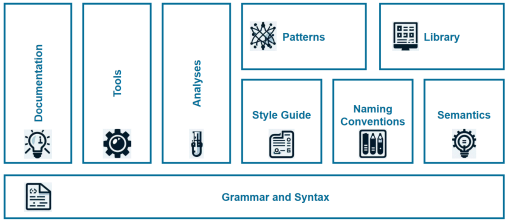
BA MA



- **Problem**
  - Simply supplying screenshots to LLM is expensive and error-prone
- **Goal**
  - Towards a user-friendly copilot for graphical languages
- **Tasks**
  - Identify relevant parts of a program / model depending on current edits, evaluate data exchange formats and supply it to an LLM and/or
  - Implement intelligent assistant that can request information iteratively
- **Needed Skills**: Java

Assessing the maturity of domain-specific language ecosystems

P BA MA

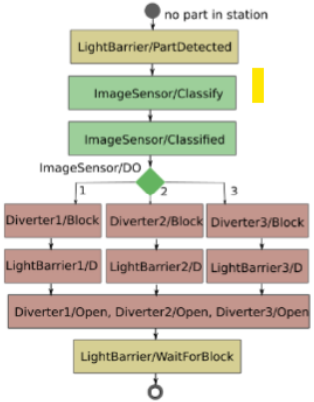


- **Problem**
  - Difficult to identify suitable DSL
- **Goal**
  - Measure the technological readiness of a domain-specific language and its supporting artefacts
- **Tasks**
  - Develop assessment tool and/or
  - Evaluate assessment tools, e.g., in interviews
- **Needed Skills**:
  Willingness to work with literature

S. Greiner et al., "Maturity Evaluation of Domain-Specific Language Ecosystems for Cyber-Physical Production Systems," IEEE ETFA, 2023, doi: 10.1109/ETFA54631.2023.10275624.

From behavior specifications to software in the era of LLMs

BA MA

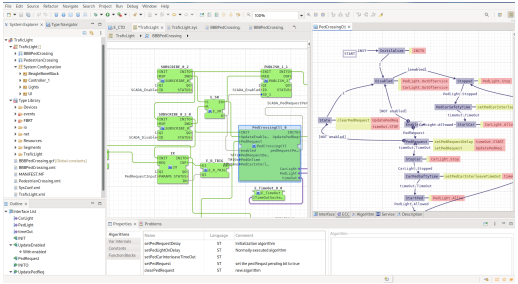


- **Problem**
  - Drawings aren't as helpful as models
- **Goal**
  - Bridge the gap between technical specification documents and software development
- **Tasks**
  - Evaluate methods for reading and interpreting flowcharts
  - Experiment with code generation from the acquired models (LLM-based, conventional,...)

B. Wiesmayr, A. Zoitl, L. Prenzel and S. Steinhorst, "Supporting a Model-driven Development Process for Distributed Control Software," IEEE ETFA, 2022, doi: 10.1109/ETFA52439.2022.9921506.

Developing assistance tools for domain experts within an open-source IDE

BA MA



- **Goal**
  - Help domain experts creating better software faster
- Possible **Tasks**
  - Develop support infrastructure for testing, visualizing, behavior specifications, etc.
  - Evaluate the support tools compared to the state of the art
- **Needed Skills**:
  - Java
  - Model-driven engineering

Eclipse 4diac, an IDE and runtime environment for distributed industrial automation. `https://eclipse.dev/4diac/`.

— 2 —

Static and Dynamic Program Analysis for Data Science (F. Sihler)

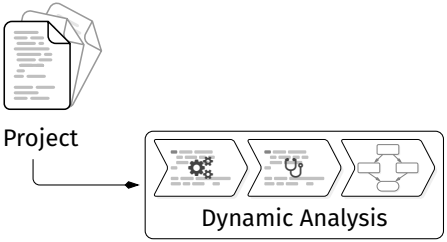
Static and Dynamic Program Analysis for Data Science

Overview

Project

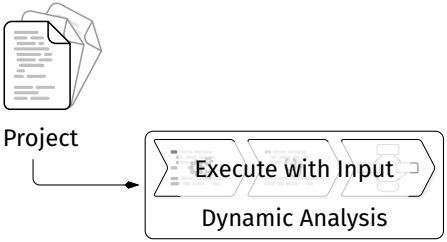
Static and Dynamic Program Analysis for Data Science

Overview



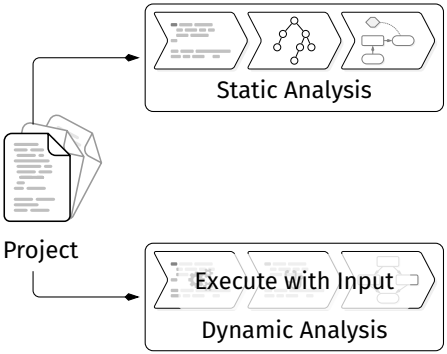
Static and Dynamic Program Analysis for Data Science

Overview



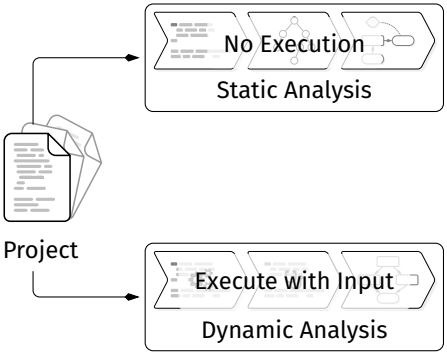
Static and Dynamic Program Analysis for Data Science

Overview



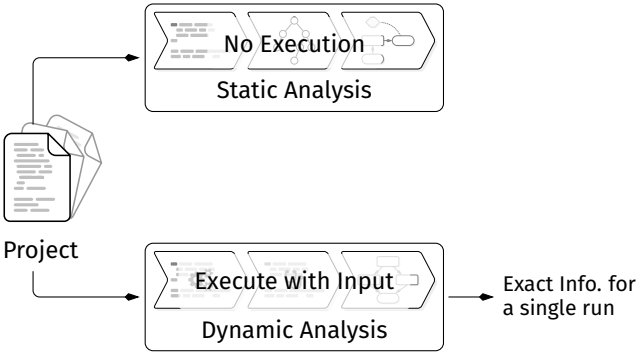
Static and Dynamic Program Analysis for Data Science

Overview



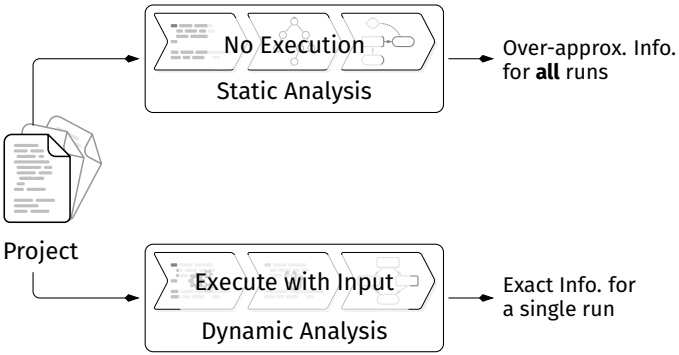
Static and Dynamic Program Analysis for Data Science

Overview



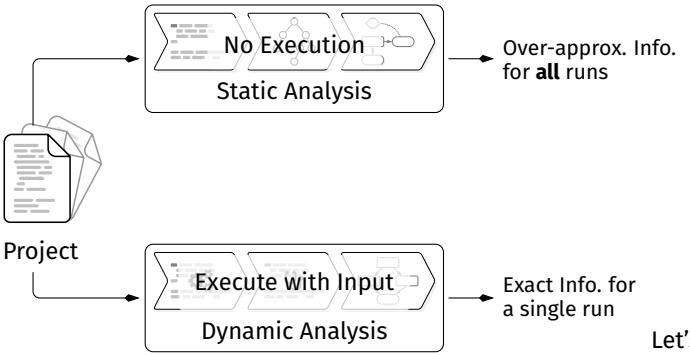
Static and Dynamic Program Analysis for Data Science

Overview



Static and Dynamic Program Analysis for Data Science

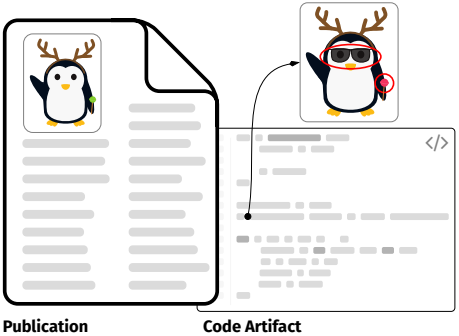
Overview



Let’s dive in!

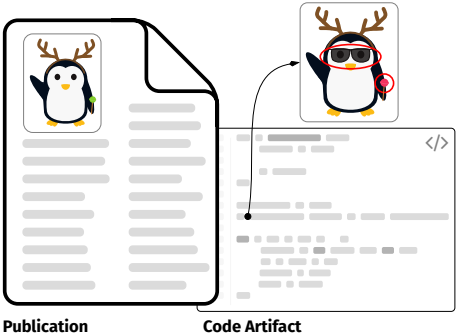
What did they do to my figure?

P BA MA



What did they do to my figure?

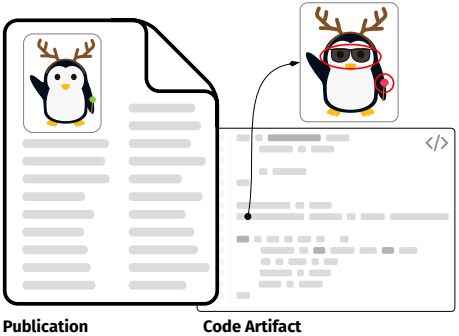
P BA MA



- **Problem**
  - Figures in publications may differ from those produced by the artifact
  - It is hard to link figures to code

What did they do to my figure?

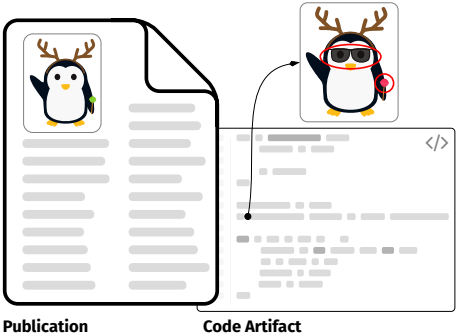
P BA MA



- **Problem**
  - Figures in publications may differ from those produced by the artifact
  - It is hard to link figures to code
- **Goal**
  - Identify changes in figures

What did they do to my figure?

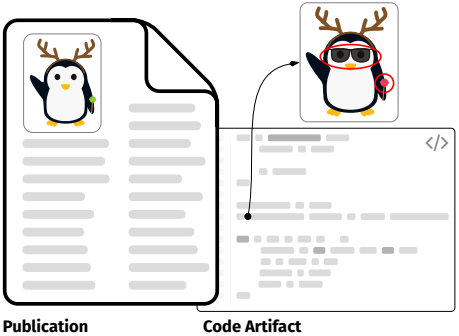
P BA MA



- **Problem**
  - Figures in publications may differ from those produced by the artifact
  - It is hard to link figures to code
- **Goal**
  - Identify changes in figures
- **Tasks**
  - Extract figures and compare them
  - Link figures to code via static analysis
  - Evaluate on real-world papers

What did they do to my figure?

P BA MA



- **Problem**
  - Figures in publications may differ from those produced by the artifact
  - It is hard to link figures to code
- **Goal**
  - Identify changes in figures
- **Tasks**
  - Extract figures and compare them
  - Link figures to code via static analysis
  - Evaluate on real-world papers
- **Needed Skills**
  - Programming in a language of choice
  - Understanding of empirical eval.

Program Slicing to Improve Conventional Coverage

BA MA

```
setup_conn ← function() {
   con ← dbConnect(SQLite(), ":memory:")
   dbExecute(con, "CREATE TABLE...")
   return(con)
}

test_that("DBI::dbExecute works", {
   expect_false(is.null(setup_conn()))
})
```

Program Slicing to Improve Conventional Coverage

BA MA

```
setup_conn ← function() {
   con ← dbConnect(SQLite(), ":memory:")
   dbExecute(con, "CREATE TABLE...")
   return(con)
}

test_that("DBI::dbExecute works", {
   expect_false(is.null(setup_conn()))
})
```

- **Problem**
  - Test coverage can be misleading

Program Slicing to Improve Conventional Coverage

BA MA

```
setup_conn ← function() {
   con ← dbConnect(SQLite(), ":memory:")
   dbExecute(con, "CREATE TABLE...")
   return(con)
}

test_that("DBI::dbExecute works", {
   expect_false(is.null(setup_conn()))
})
```

- **Problem**
  - Test coverage can be misleading
- **Goal**
  - Improve test-coverage metrics with program slicing

Program Slicing to Improve Conventional Coverage

BA MA

```
setup_conn ← function() {
   con ← dbConnect(SQLite(), ":memory:")
   dbExecute(con, "CREATE TABLE...")
   return(con)
}

test_that("DBI::dbExecute works", {
   expect_false(is.null(setup_conn()))
})
```

- **Problem**
  - Test coverage can be misleading
- **Goal**
  - Improve test-coverage metrics with program slicing
- **Tasks**
  - Pick a language of choice
  - Identify sensible slicing points
  - Evaluate on real-world projects

Program Slicing to Improve Conventional Coverage

BA MA

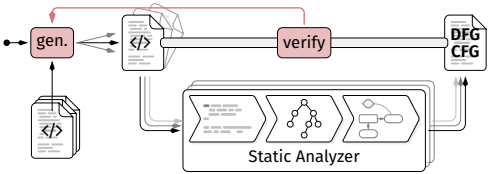
```
setup_conn ← function() {
   con ← dbConnect(SQLite(), ":memory:")
   dbExecute(con, "CREATE TABLE...")
   return(con)
}

test_that("DBI::dbExecute works", {
   expect_false(is.null(setup_conn()))
})
```

- **Problem**
  - Test coverage can be misleading
- **Goal**
  - Improve test-coverage metrics with program slicing
- **Tasks**
  - Pick a language of choice
  - Identify sensible slicing points
  - Evaluate on real-world projects
- **Needed Skills**
  - Understanding of test coverage
  - Programming in a language of choice

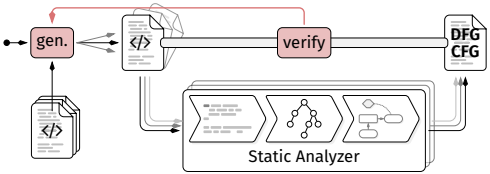
Fully automated Correctness Verification for Static Analysis Tools

BA MA P



Fully automated Correctness Verification for Static Analysis Tools

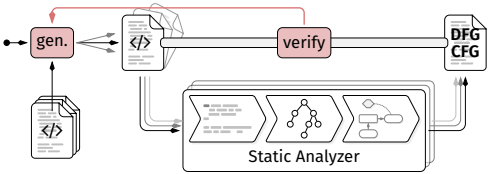
BA MA P



- **Problem**
  - Languages have many, complex and usually underspecified semantics
  - Verifying static analyses is hard

Fully automated Correctness Verification for Static Analysis Tools

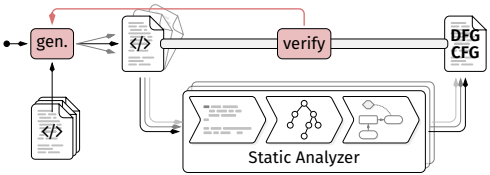
BA MA P



- **Problem**
  - Languages have many, complex and usually underspecified semantics
  - Verifying static analyses is hard
- **Goal**
  - Find automated ways to check correctness properties

Fully automated Correctness Verification for Static Analysis Tools

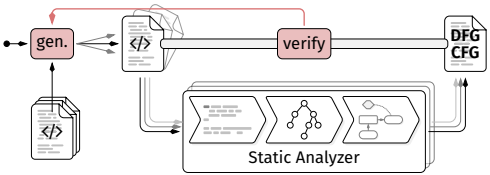
BA MA P



- **Problem**
  - Languages have many, complex and usually underspecified semantics
  - Verifying static analyses is hard
- **Goal**
  - Find automated ways to check correctness properties
- **Tasks**
  - Identify strategies like metamorphic testing or differential testing
  - Use these to generate test cases
  - Evaluate effectiveness

Fully automated Correctness Verification for Static Analysis Tools

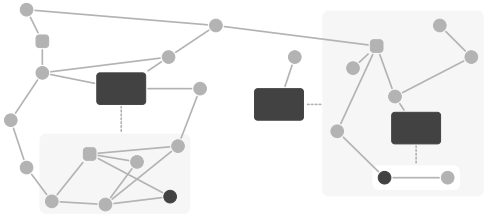
BA MA P



- **Problem**
  - Languages have many, complex and usually underspecified semantics
  - Verifying static analyses is hard
- **Goal**
  - Find automated ways to check correctness properties
- **Tasks**
  - Identify strategies like metamorphic testing or differential testing
  - Use these to generate test cases
  - Evaluate effectiveness
- **Needed Skills**
  - Knowledge of PBT or similar

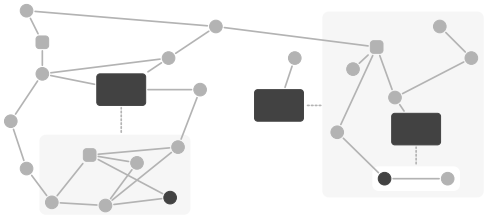
Transitive Packages and Security

P BA MA



Transitive Packages and Security

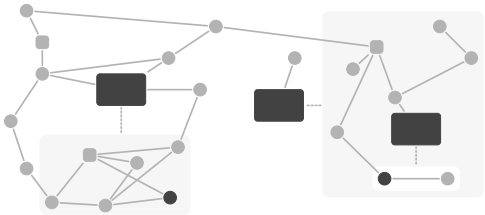
P BA MA



- **Goal**
  - Work on and extend *flowR*, a static analysis framework

Transitive Packages and Security

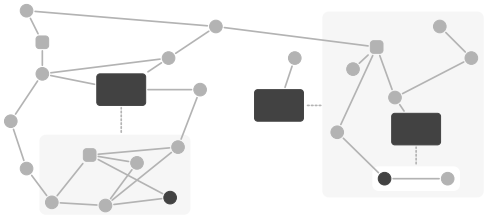
P BA MA



- **Goal**
  - Work on and extend *flowR*, a static analysis framework
- (Possible) **Tasks**
  - Security Analysis
  - Transitive Package Information
  - Dynamic Code Loading
  - Implicit Assumption Inference
  - …

Transitive Packages and Security

P BA MA

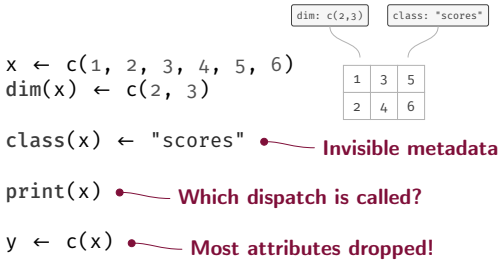


- **Goal**
  - Work on and extend *flowR*, a static analysis framework
- (Possible) **Tasks**
  - Security Analysis
  - Transitive Package Information
  - Dynamic Code Loading
  - Implicit Assumption Inference
  - …
- **Needed Skills**
  - Programming in/with TypeScript

— 3 —
Static and Dynamic Program Analysis for Data Science (O. Gerstl)

Attribute Inference for R Programs

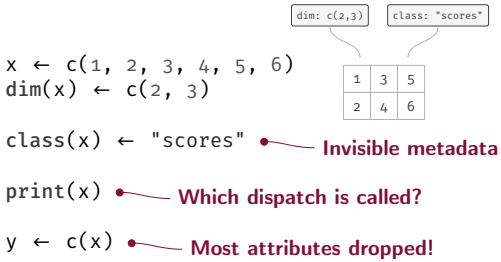
BA MA



- **Problem**
  - Every object in R can have *attributes*
  - Attributes change the behavior of variables and are invisible to analysis

Attribute Inference for R Programs

BA MA



- **Problem**
  - Every object in R can have *attributes*
  - Attributes change the behavior of variables and are invisible to analysis
- **Goal**
  - Statically infer attributes of R values

Attribute Inference for R Programs

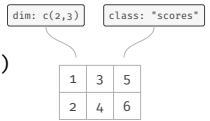
BA MA

```
x ← c(1, 2, 3, 4, 5, 6)
dim(x) ← c(2, 3)

class(x) ← "scores"

print(x)

y ← c(x)
```



Invisible metadata

Which dispatch is called?

Most attributes dropped!

- **Problem**
  - Every object in R can have *attributes*
  - Attributes change the behavior of variables and are invisible to analysis
- **Goal**
  - Statically infer attributes of R values
- **Tasks**
  - Design abstract domains for attributes in R
  - Track attributes though program
  - Evaluation on real-world R scripts

Attribute Inference for R Programs

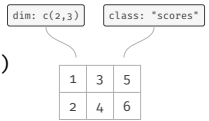
BA MA

```
x ← c(1, 2, 3, 4, 5, 6)
dim(x) ← c(2, 3)

class(x) ← "scores"

print(x)

y ← c(x)
```



- **Problem**
  - Every object in R can have *attributes*
  - Attributes change the behavior of variables and are invisible to analysis
- **Goal**
  - Statically infer attributes of R values
- **Tasks**
  - Design abstract domains for attributes in R
  - Track attributes though program
  - Evaluation on real-world R scripts
- **Needed Skills**
  - Programming in/with TypeScript
  - Interest in abstract interpretation

— 4 —
Privacy and Architecture-based Simulation (P. Leber)

Seminar Privacy in Mobility Systems

It's Official: Cars Are the Worst Product Category We Have Ever Reviewed for Privacy

By **Jen Caltrider, Misha Rykov** and **Zoë MacDonald** | Sept. 6, 2023

[1]

CARS AND TRANSPORTATION | GLOBAL ISSUES

Smart cars, spying fears and security

Kevin Tschierse
04/30/2026

Cameras, sensors, constant connectivity: modern cars collect vast amounts of data.

[2]

Data leak in Switzerland: Open databases at parking enforcement companies

Sensitive owner data and fines for tens of thousands of Swiss drivers were exposed online due to errors by parking enforcement firms.

[3]

We Know Where You Parked

Massive Data Breach at VW Raises Questions about Vehicle Privacy

Already facing significant headwinds, VW has now been hit by a data protection nightmare. Location data from 800,000 electric vehicles and contact info from owners was accessible unprotected on the internet. And the company didn't even know about it.

By **Patrick Beuth**, **Flüpke**, **Max Hoppenstedt Heinrichs**, **Michael Kreil**, **Marcel Rosenbach** und **Rina Wilkin**
03.01.2025, 16.19 Uhr

[4]

Your car's tire sensors could be used to track you

Researchers at IMDEA Networks show standard tire sensors can expose drivers' movements, raising privacy concerns

25 February 2026

[5]

Seminar Privacy in Mobility Systems

What will we do in the seminar?

We want to take a look at current research topics such as:

- Threat Modeling
- Privacy Frameworks
- Privacy Pattern
- Privacy Enhancing Technologies
- Anonymity

Using recent publications, we aim to gain an understanding of this highly relevant and current topic and examine both practical applications and theoretical background.

Contact

- **Who?**
  - Pascal Leber
  - Prof. Dr. Robert Heinrich
- **Email**
  - pascal.leber@uni-ulm.de
  - robert.heinrich@uni-ulm.de
- **Anmeldung**
  - Moodle: Anmeldung zur Verteilung der Seminare im kommenden Wintersemester https://moodle.uni-ulm.de/course/view.php?id=77420

Privacy within the Data Flow Diagram

Data-Hungry Dating Apps Are Worse Than Ever for Your Privacy

By **Jen Caltrider, Misha Rykov** and **Zoë MacDonald** | April 23, 2024

[6]

FTC to Ban BetterHelp from Revealing Consumers' Data, Including Sensitive Mental Health Information, to Facebook and Others for Targeted Advertising

BetterHelp will be required to pay $7.8 million for deceiving consumers after promising to keep sensitive personal data private, agency says

[7]

Camera glasses and data protection: Meta reassures

Meta is reacting to growing criticism of camera glasses and promising new data protection measures. But many questions remain unanswered.

[8]

TikTok faces €530 million EU fine over China data transfers

Dmytro Hubenko with dpa, AP, AFP
05/02/2025

An investigation found the social media giant transferred data to China, violating strict EU privacy rules.

[9]

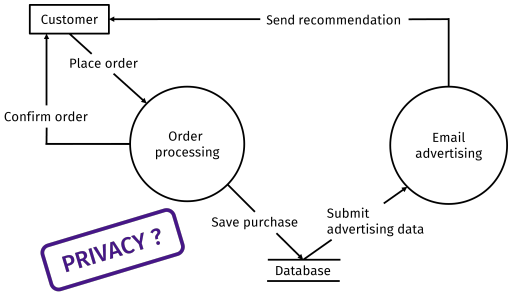
Exploring privacy issues in the age of AI

Published 30 September 2024 | Updated 24 February 2026

[10]

Privacy within the Data Flow Diagram

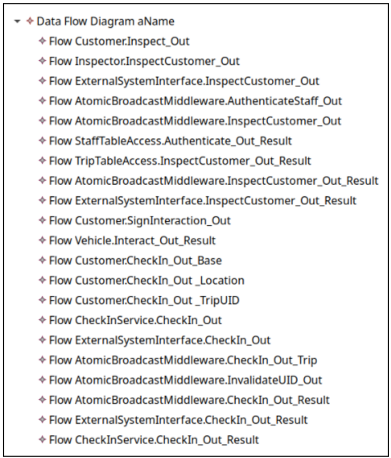
Overview



- **Problem**
  - Privacy violations damage companies' reputations and expose them to costly lawsuits
  - Potential privacy issues should be identified early on, since resolving them later is also expensive
  - A tool for early detection is still lacking
- **Goal**
  - Existing prototypical scientific approaches should be integrated into a unified tool in order to address this gap using a Privacy DFD.

A Graphical Representation for Privacy Data Flow Diagrams

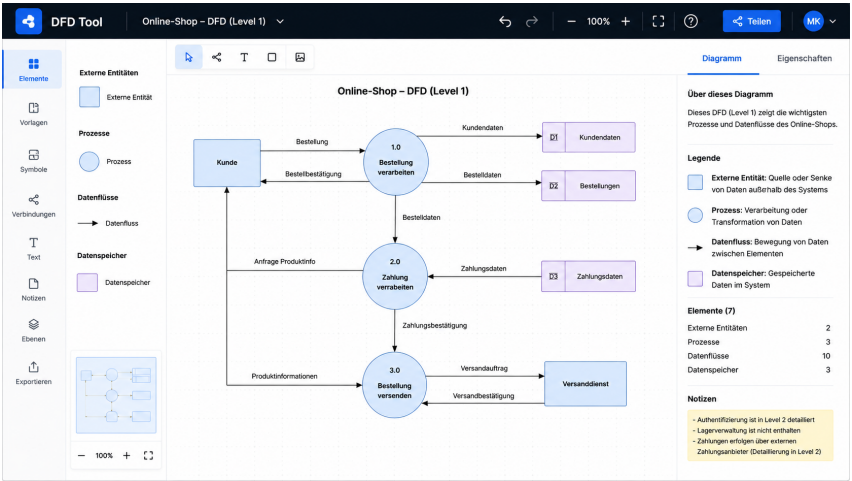
P



- **Problem**
  - Only option for creating a Privacy DFD is the tree editor
  - Complicates the creation of a Privacy DFD
  - Graphical representation of the analysis results is missing
  - Web tool does not support privacy features
- **Tasks**
  - Creating a graphical representation for the Privacy DFD and its analysis results within the desktop application
  - Adapting the web tool

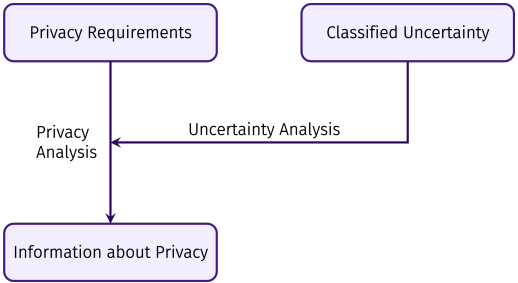
A Graphical Representation for Privacy Data Flow Diagrams

P



Uncertainty-Aware Privacy Analysis within the Privacy DFD

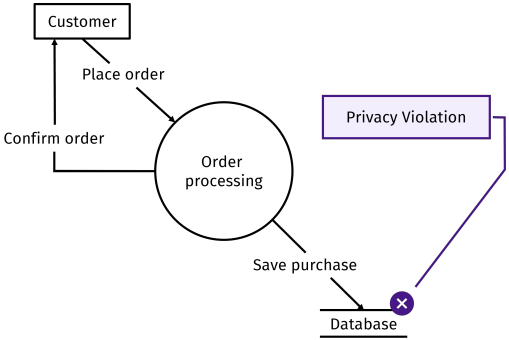
BA MA



- **Problem**
  - Current tool analyzes a Privacy DFD for violations only in the standard way
  - However, systems are always fraught with uncertainties
  - If not addressed, these can lead to erroneous conclusions in analyses
- **Tasks**
  - Classifying the uncertainties that are relevant in relation to the Privacy DFD
  - Examine how uncertainties propagate through the system
  - Develop a mitigation approach to address the uncertainty accordingly

Mitigation of Privacy Violations within the Privacy DFD

BA MA



- **Problem**
  - The Privacy DFD tool does not yet address the mitigation of violations
  - Currently, any detected violations must still be resolved manually
  - This makes the mitigation process error-prone and time-consuming
- **Tasks**
  - Develop a method that automatically resolves violations using discrete optimization techniques
  - Select an appropriate optimization technique, adapt and implement the procedure for the Privacy DFD, and evaluate the results

Contact

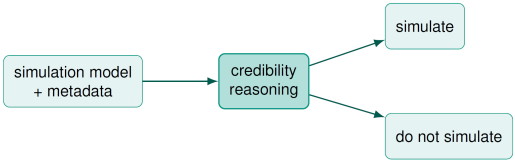
- **Who?**
  - Pascal Leber
  - Prof. Dr. Robert Heinrich
- **Email**
  - pascal.leber@uni-ulm.de
  - robert.heinrich@uni-ulm.de

References

1. Jen Caltrider et al. *It's Official: Cars Are the Worst Product Category We Have Ever Reviewed for Privacy*. Available: https://www.mozillafoundation.org/en/privacynotincluded/articles/its-official-cars-are-the-worst-product-category-we-have-ever-reviewed-for-privacy/. Accessed: Jul. 12, 2026.
2. Kevin Tschierse. *Smart cars, spying fears and security*. Available: https://www.dw.com/en/smart-cars-spying-fears-and-security/video-76995425. Accessed: Jul. 12, 2026.
3. Stefan Krempl. *Data leak in Switzerland: Open databases at parking enforcement companies*. Available: https://www.heise.de/en/news/Data-leak-in-Switzerland-Open-databases-at-parking-enforcement-companies-11310510.html. Accessed: Jul. 12, 2026.
4. Patrick Beuth et al. *Massive Data Breach at VW Raises Questions about Vehicle Privacy*. Available: https://www.spiegel.de/international/business/we-know-where-you-parked-massive-data-breach-at-vw-raises-questions-about-vehicle-privacy-a-4b1cb926-2edb-42ea-92fb-5000cd378fc5. Accessed: Jul. 12, 2026.
5. IMDEA Networks Institute. *Your car's tire sensors could be used to track you*. Available: https://networks.imdea.org/your-cars-tire-sensors-could-be-used-to-track-you/. Accessed: Jul. 12, 2026.
6. Jen Caltrider et al. *Data-Hungry Dating Apps Are Worse Than Ever for Your Privacy*. Available: https://www.mozillafoundation.org/en/privacynotincluded/articles/data-hungry-dating-apps-are-worse-than-ever-for-your-privacy/. Accessed: Jul. 8, 2026.
7. Federal Trade Commission. *FTC to Ban BetterHelp from Revealing Consumers' Data, Including Sensitive Mental Health Information, to Facebook and Others for Targeted Advertising*. Available: https://www.ftc.gov/news-events/news/press-releases/2023/03/ftc-ban-betterhelp-revealing-consumers-data-including-sensitive-mental-health-information-facebook. Accessed: Jul. 12, 2026.
8. Tomislav Bezmalinović. *Camera glasses and data protection: Meta reassures*. Available: https://www.heise.de/en/news/Camera-glasses-and-data-protection-Meta-reassures-11358000.html. Accessed: Jul. 12, 2026.
9. Dmytro Hubenko. *TikTok faces €530 million EU fine over China data transfers*. Available: https://www.dw.com/en/tiktok-faces-530-million-eu-fine-over-china-data-transfers/a-72420728. Accessed: Jul. 12, 2026.
10. Alice Gomstyn and Alexandra Jonker. *Exploring privacy issues in the age of AI*. Available: https://www.ibm.com/think/insights/ai-privacy. Accessed: Jul. 12, 2026.

Assessing Simulation Model Credibility

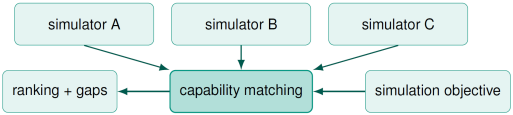
BA MA



- **Problem**
  - The credibility of simulation results is assessed based on experience
  - To address this issue, an ontology has already been developed
  - However, this is currently at the conceptual stage
- **Task**
  - Develop a concept for operationalizing the ontology
  - Implement this concept as a prototype and apply it to sample scenarios for co-simulation
  - Evaluate the approach, e.g., through expert interviews

Metadata-Based Simulator Selection

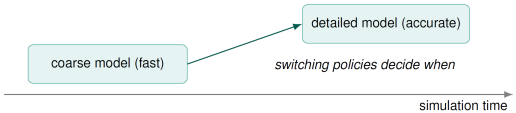
BA MA



- **Problem**
  - Reusing an existing simulator is not necessarily guaranteed
  - A framework was developed to compare simulators against a target
  - However, this is currently at the conceptual stage
- **Task**
  - Refine the metadata schema and design the matching and gap-reasoning procedures
  - Implement the concept as a first prototype
  - Assess the matching quality empirically

When Should a Simulation Switch?

BA MA



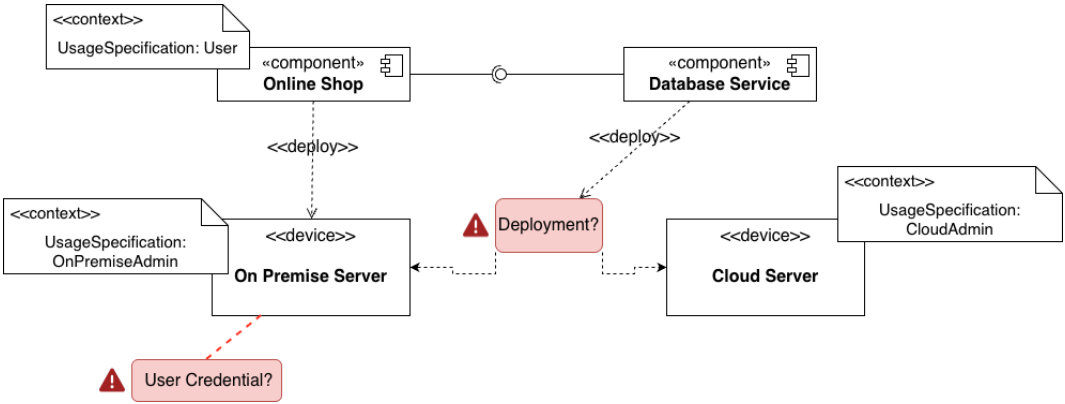
- **Problem**
  - A rough model is used for speed; a detailed model is used for accuracy
  - The switch takes place during runtime and varies with each project
  - A domain-independent specification for switching simulations is missing
- **Task**
  - Provide an overview of switching approaches based on the existing literature
  - Developing a Concept for domain independent switching behavior
  - A prototype needs to be developed to validate the concept

Contact

- **Who?**
  - Prof. Dr. Robert Heinrich
- **Email**
  - robert.heinrich@uni-ulm.de

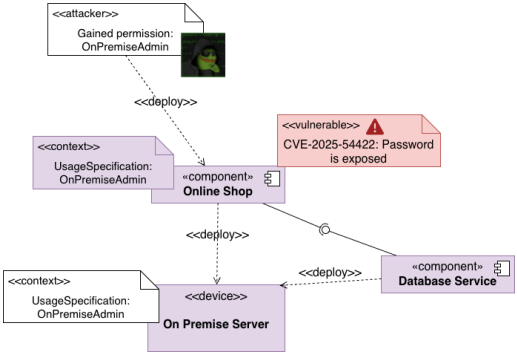
— 5 —
Architectural Security Analysis (L. Le)

Architectural Security Analysis



Mitigation of Attack Propagation using Architectural Analysis and Language Models

BA MA



- **Problem**
  - An attack can propagate and thus affect the entire system.[a]
  - Selecting appropriate mitigation techniques requires a lot of expert knowledge.
  - LLMs are increasingly used in security, so they may help reduce this expert effort.[bc]

[a] Walter et al. "Architectural attack propagation analysis for identifying confidentiality issues." ICSA. IEEE, 2022.

[b] Wang et al. "Shieldgpt: An llm-based framework for ddos mitigation." Proceedings of the 8th asia-pacific workshop on networking. 2024.

[c] Gong et al. "Information security based on llm approaches: A review." arXiv preprint arXiv:2507.18215 (2025).

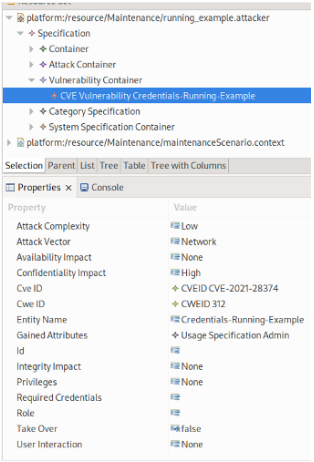
Mitigation of Attack Propagation using Architectural Analysis and Language Models (Cont.)

BA MA

- **Tasks in detail**
  - Build a GraphRAG (Graph-based Retrieval Augmented Generation) based on system architecture.
  - Develop a process that uses a software architecture model and an access control policy model as input for the LLM.
  - The output is the set of possible attacks, how they propagate through the architecture, and the mitigations proposed by the LLM.
- **What you gain**
  - Hands-on experience combining LLMs, GraphRAG, and model-driven security analysis.
  - Work on an active research direction that can lead to a publication.

Online Modelling and Analysis Tool to investigate Attack Propagation in Software Architectures

P

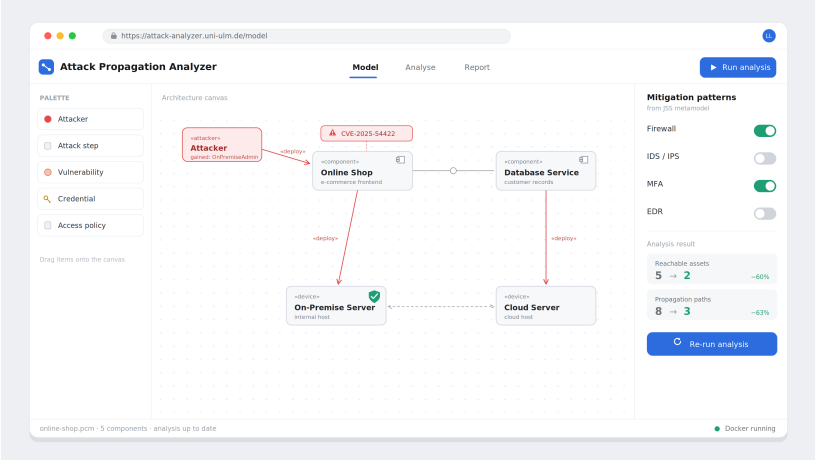


- **Today**
  - A powerful attack propagation and mitigation analysis already exists, but only inside Eclipse.[a]
  - No web interface, local install required.
- **We need**
  - A browser-based tool, no local install, to model attacks and analyse mitigations.

[a] `https://fluidtrust.github.io/attack-propagation-doc/index.html`

Online Modelling and Analysis Tool to investigate Attack Propagation in Software Architectures (Cont.)

Expected result



Online Modelling and Analysis Tool to investigate Attack Propagation in Software Architectures (Cont.)

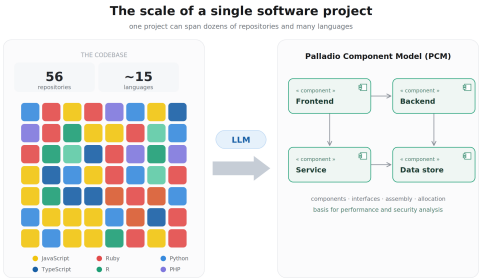
Work packages

- **A. Web modelling editor**
  - Import an architectural model and model the attacker with a graphical drag-and-drop editor.
- **B. Analysis backend**
  - Integrate the existing attack propagation analysis as a backend service and ship it as a Docker package.
- **C. Mitigation integration**
  - Integrate the existing mitigation metamodel and analysis, then apply patterns and compare propagation before vs. after.
- **D. Attack graph & interop**
  - Render and export an attractive attack graph, and round-trip models with the Eclipse-based tool.

- **What you gain**
  - This is an open-source project, and the tool will be used by the community.
  - You can put a real, citable contribution on your CV, not just a course exercise.
  - You gain hands-on experience turning a research prototype into a usable product.

Reverse Engineering Architectural Models from Source Code using Language Models

BA MA



- **Problem**
  - Real research software is large and polyglot. E.g. Chemotion (ComPlat, KIT) spans 56 repositories and around 15 programming languages.[a]
  - No up-to-date architectural model exists, so understanding the overall system structure is hard.
  - Our analyses, such as performance and security analysis, need an architectural model as input, but building one by hand is infeasible.

[a] `https://github.com/ComPlat` and `https://chemotion.net/docs`

Reverse Engineering Architectural Models from Source Code using Language Models (Cont.)

BA MA

- **Goal**
  - Use an LLM to reverse engineer an architectural model in the Palladio Component Model (PCM) [a] from a real codebase.
  - The recovered model serves as a basis for both performance and security analysis.

[a] Reussner, Ralf H., et al. "Modeling and simulating software architectures: The Palladio approach." MIT Press, 2016.

- **Current research**
  - LLM-based software architecture recovery.[a]
  - Existing work mostly targets UML or C4 diagrams for human reading, not analysis-ready models such as PCM.

[a] Amalfitano et al. "Automated Software Architecture Design Recovery from Source Code Using LLMs." ECSA. Springer, 2025.

Reverse Engineering Architectural Models from Source Code using Language Models (Cont.)

BA MA

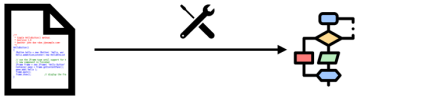
- **Tasks in detail**
  - Use an LLM to map the code from codebase onto PCM elements, such as components, interfaces, assembly, and allocation.
  - Produce a PCM model that loads into the existing analysis tooling.
- **What you gain**
  - Hands-on experience combining LLMs with model-driven engineering on a large, real open-source system.
  - Work on an active research direction that can lead to a publication.

Contact

- **Who?**
  - Lan Le
  - Prof. Dr. Robert Heinrich
- **Email:**
  - lan.le@uni-ulm.de
  - robert.heinrich@uni-ulm.de

— 6 —
Algorithm Detection (D. Neumüller)

Algorithm Detection



Analysis of Source Code Base

Detected algorithms and involved components

Motivation

- Developers spend more than 50% of their time reading and understanding code.

Software Comprehension Problems

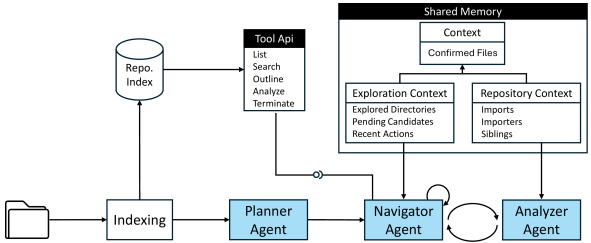
- Tedious, time consuming and often manual process.
- Out of date documentation.
- Colleagues and experts not available.

Goal

- Support software comprehension by automatically detecting algorithms contained in a code base.

Agentic Algorithm Detection

BA MA



Goal

- Improve the performance of the detection

Example Tasks

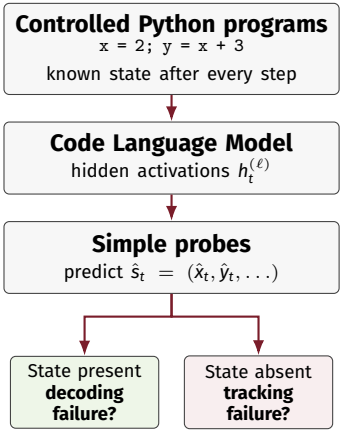
- More Tools for the Navigator
  e.g. find symbol tool
- Compare different prompting styles for the analyzer
  e.g. few-shot, debate
- Add another agent
  e.g. high-level consistency checker

— 7 —

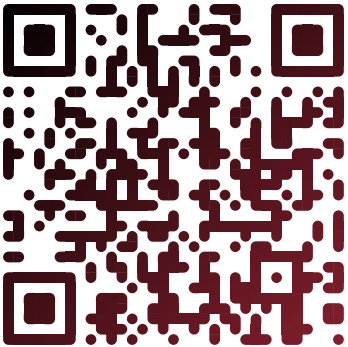
Probing Program State in Code Language Models (A. Diera)

Probing Program State in Code Language Models

BA MA



- **Problem**
  - Code language models often fail to compute exact execution values, but it is unclear whether they lose track of the program state or merely fail to decode it.
- **Task**
  - Generate synthetic Python programs with known intermediate variable values.
  - Collect activations from an open-source language model by layer and execution step.
  - Train simple probes to predict the program state from those activations.



uulm.de/in/sp/teaching/topics-for-theses-and-projects/

Bis bald beim Institut für Softwaretechnik und Programmiersprachen!

Vielen Dank für Ihre Aufmerksamkeit.