Institutsvorstellung

Wintersemester 25/26

Institute of Software Engineering and Programming Languages

Agenda (1/2)

- Configurable Software Systems (Sabrina Böhm)
  - Exploring Complex Feature Interactions in Software Product Lines (BA/MA)
  - Survey on the Analysis of Sampling in Software Product Lines (BA/MA)
- Self-Adaptive Systems (Raphael Straub)
  - Scalable Visualization for State-Graphs (SE/Informatik Projekt)
  - Automated Model Generation for Self-Adaptive Microservices (MA)
- Attack Modelling and Analysis for Secure Software Systems (Lan Le)
  - Mitigation of Attack Propagation using Architectural Analysis and Language Models (BA/MA)
  - Online Modelling and Analysis Tool to investigate Attack Propagation in Software Architectures (SE/Informatik Projekte)

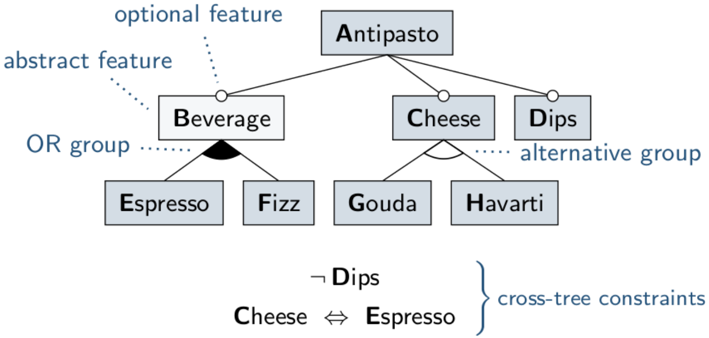
Agenda (2/2)

- Static Program-Analysis (Florian Sihler)
    - Static Program-Analysis for Data Analysis Projects (SE/Informatik Projekte)

- SE/Informatik Projekte (Robert Heinrich in Kooperationen mit Industrie und Forschung)
    - RAG for Public Sector -- A Tool for Enhancing Public Service Delivery by Information Retrieval with LLMs (SE/Informatik Projekt in Kooperation mit DPS Engineering GmbH)
    - BlockchainBench -- An extensible Tool for Modelling and Analysing Blockchain Systems (SE/Informatik Projekt in Kooperation mit KIT/TUM)

Configurable Software Systems

- Identification of Feature Interactions



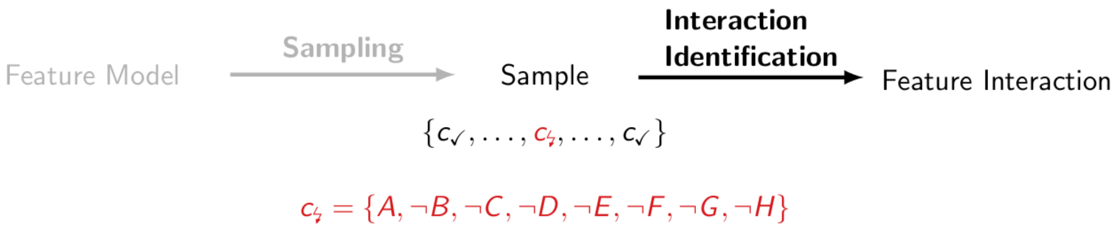
Sample $S_1 = \{$

$c_1 =$
$\{A, \neg B, \neg C, \neg D, \neg E, \neg F, \neg G, \neg H\},$

$c_2 =$
$\{A, B, C, \neg D, E, \neg F, G, \neg H\}\}$

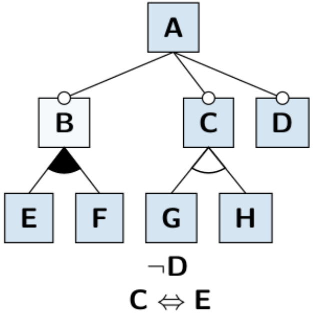
Exploring Complex Feature Interactions in Software Product Lines

BA MA



$\{c_{\checkmark}, \ldots, c_{\lightning}, \ldots, c_{\checkmark}\}$

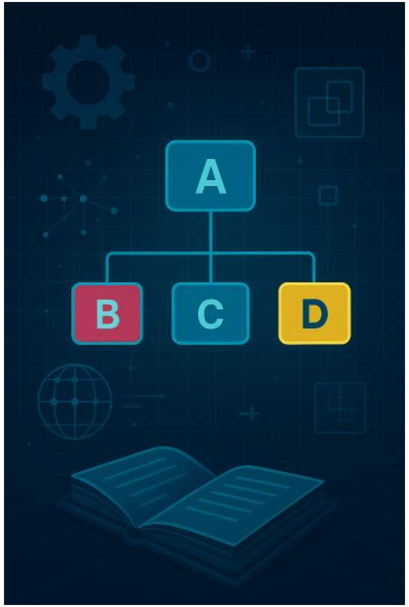
$c_{\lightning} = \{A, \neg B, \neg C, \neg D, \neg E, \neg F, \neg G, \neg H\}$



- Tasks

- Research on occurencies of complex feature interaction bugs and error masking bugs in configurable software systems

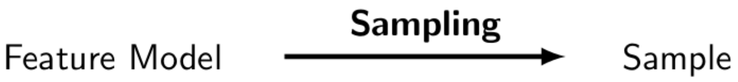
- Detect, compare, and discuss the complex feature interactions



Sabrina (Mail sabrina.boehm@uni-ulm.de)

Survey on the Analysis of Sampling in Software Product Lines

BA MA

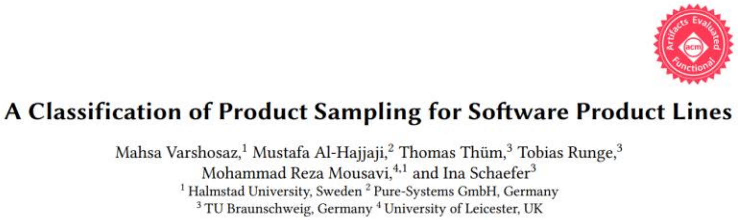


In the paper "A classification of product sampling for software product lines" (SPLC'18), Mahsa et al. propose a classification for product sampling techniques by classifying the existing literature.

- Tasks

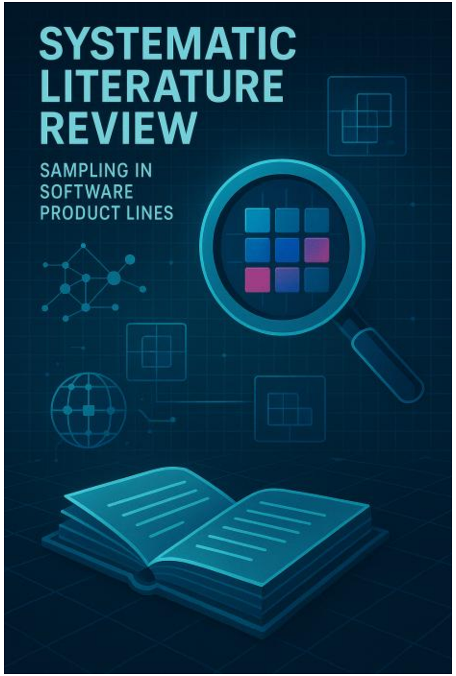
- Literature research

- Investigate and classify new research topics and research gaps of growing interest



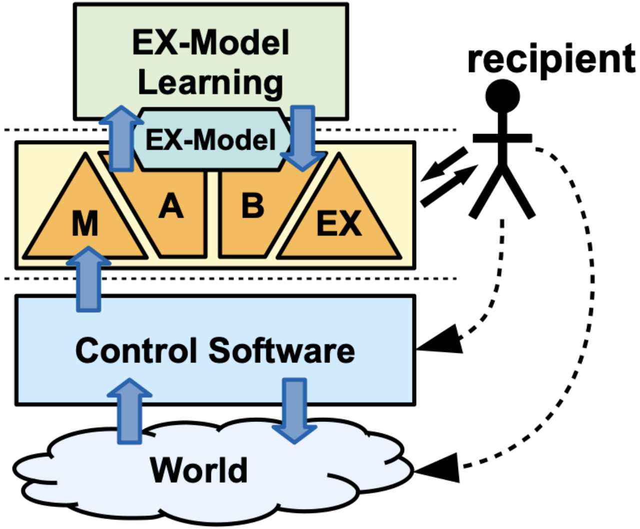
A Classification of Product Sampling for Software Product Lines

Mahsa Varshosaz,[1] Mustafa Al-Hajjaji,[2] Thomas Thüm,[3] Tobias Runge,[3]
Mohammad Reza Mousavi,[4,1] and Ina Schaefer[3]
[1] Halmstad University, Sweden [2] Pure-Systems GmbH, Germany
[3] TU Braunschweig, Germany [4] University of Leicester, UK


SYSTEMATIC LITERATURE REVIEW
SAMPLING IN SOFTWARE PRODUCT LINES

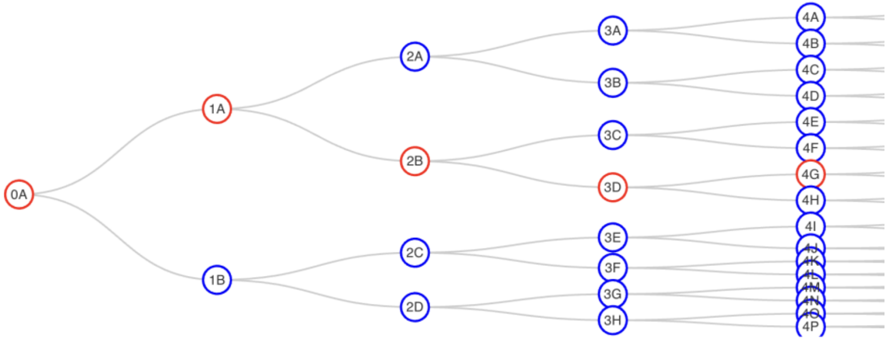
Sabrina (Mail sabrina.boehm@uni-ulm.de)

Self-Adaptive Systems

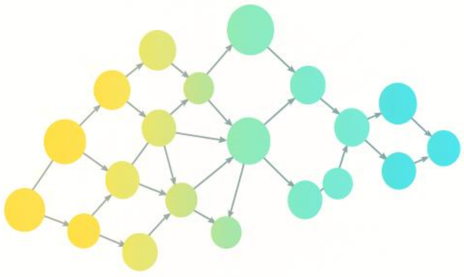


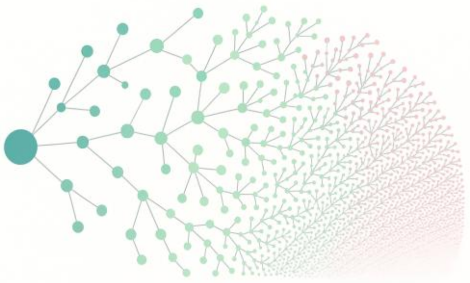
Scalable Visualization for State-Graphs

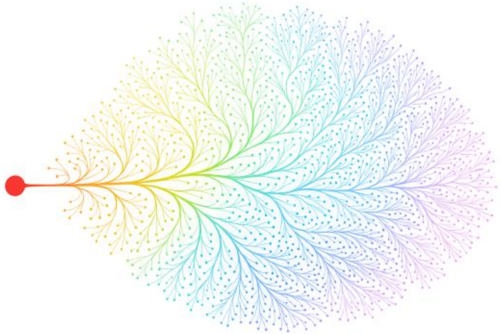
P



- Problem
  - Our state graphs get too large to display
  - We need to visualize all states in a comprehensive and understandable way
  - Ensure the visualization is intuitive and easy to understand
- Tasks
  - Develop visualization concepts for Large State Graphs
  - Implement the developed visualization concepts
  - Develop the solution in React/Typescript







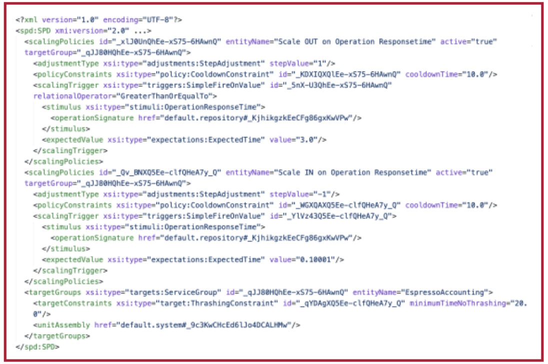
Automated Model Generation for Self-Adaptive Microservices

MA

4-8 Components,
<10 SLOs,
1-3 Policies per
Component

Constraints

Your Prototype



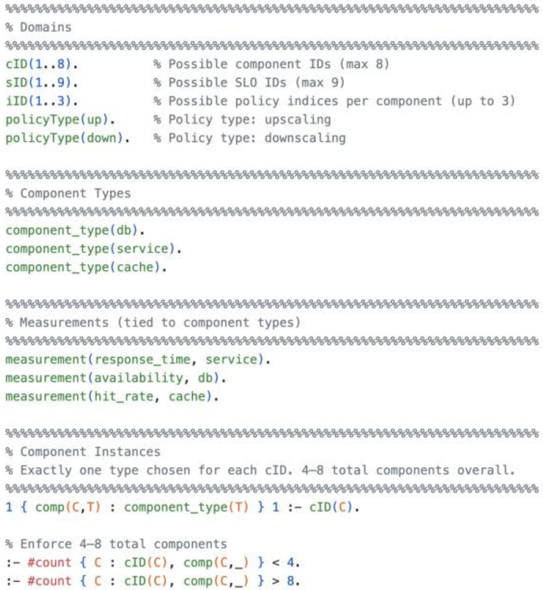
```
<?xml version="1.0" encoding="UTF-8"?>
<spd:SPD xmi:version="2.0" ...>
  <scalingPolicies id="_xlJ0UnQhEe-xS75-6HAwnQ" entityName="Scale OUT on Operation Responsetime" active="true"
  targetGroup="_qJJ80HQhEe-xS75-6HAwnQ">
    <adjustmentType xsi:type="adjustments:StepAdjustment" stepValue="1"/>
    <policyConstraints xsi:type="policy:CooldownConstraint" id="_KDXIQXQlEe-xS75-6HAwnQ" cooldownTime="10.0"/>
    <scalingTrigger xsi:type="triggers:SimpleFireOnValue" id="_5nX-U3QhEe-xS75-6HAwnQ"
    relationalOperator="GreaterThanOrEqualTo">
      <stimulus xsi:type="stimuli:OperationResponseTime">
        <operationSignature href="default.repository#_KjhikgzkEeCFg86gxKwVPw"/>
      </stimulus>
      <expectedValue xsi:type="expectations:ExpectedTime" value="3.0"/>
    </scalingTrigger>
  </scalingPolicies>
  <scalingPolicies id="_Qv_BNXQ5Ee-clfQHeA7y_Q" entityName="Scale IN on Operation Responsetime" active="true"
  targetGroup="_qJJ80HQhEe-xS75-6HAwnQ">
    <adjustmentType xsi:type="adjustments:StepAdjustment" stepValue="-1"/>
    <policyConstraints xsi:type="policy:CooldownConstraint" id="_WGXQAXQ5Ee-clfQHeA7y_Q" cooldownTime="10.0"/>
    <scalingTrigger xsi:type="triggers:SimpleFireOnValue" id="_YlVz43Q5Ee-clfQHeA7y_Q">
      <stimulus xsi:type="stimuli:OperationResponseTime">
        <operationSignature href="default.repository#_KjhikgzkEeCFg86gxKwVPw"/>
      </stimulus>
      <expectedValue xsi:type="expectations:ExpectedTime" value="0.10001"/>
    </scalingTrigger>
  </scalingPolicies>
  <targetGroups xsi:type="targets:ServiceGroup" id="_qJJ80HQhEe-xS75-6HAwnQ" entityName="EspressoAccounting">
    <targetConstraints xsi:type="target:ThrashingConstraint" id="_qYDAgXQ5Ee-clfQHeA7y_Q" minimumTimeNoThrashing="20.
    0"/>
    <unitAssembly href="default.system#_9c3KwCHcEd6lJo4DCALHMw"/>
  </targetGroups>
</spd:SPD>
```

PCM Models

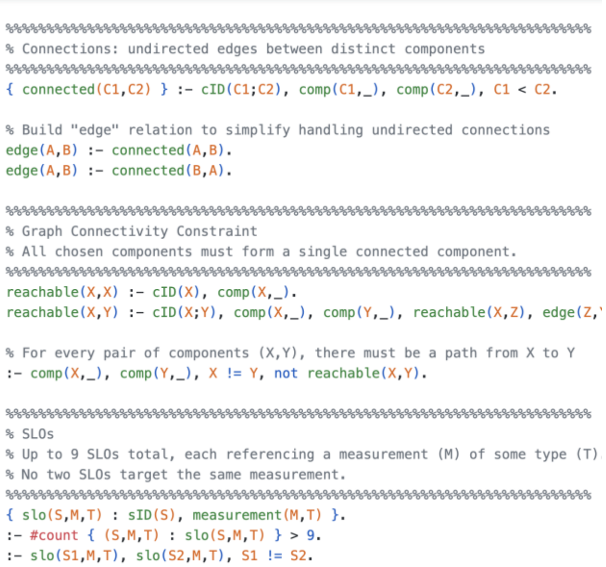
- Problem
  - PCM Models are complex
  - Degrees of freedom have to be identified and possible constraints defined
  - An approach to solve the problem has to be implemented

- Tasks
  - Analyze the PCM Models, define possible constraints, chose an approach for solving the problem
  - Develop a Prototype
  - Evaluate the Prototype

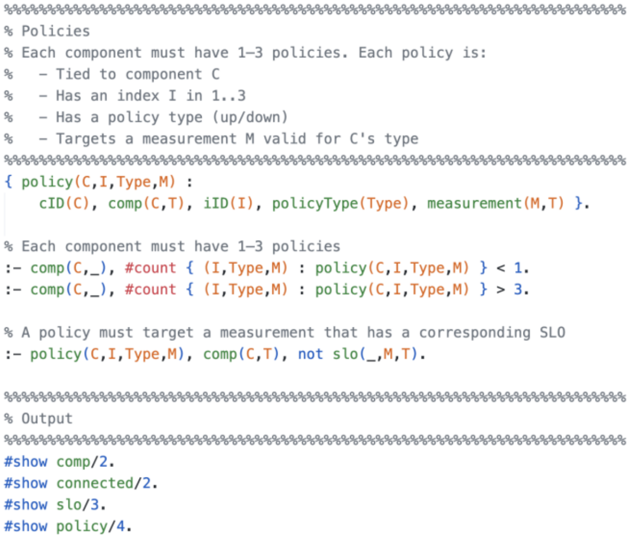
Automated Model Generation: Simple Example



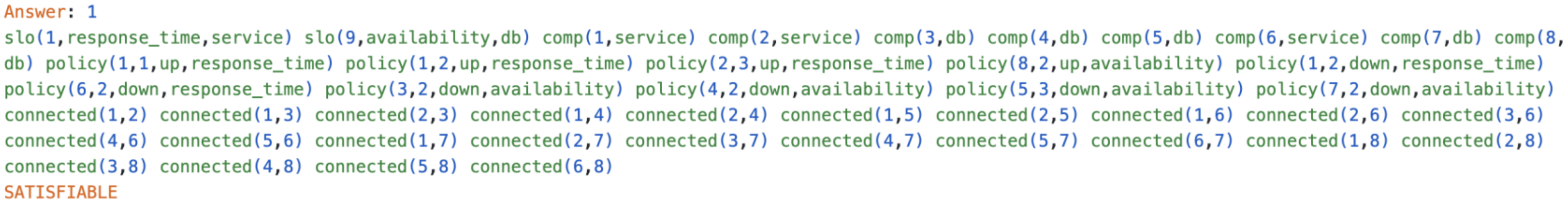
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Domains
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cID(1..8).          % Possible component IDs (max 8)
sID(1..9).          % Possible SLO IDs (max 9)
iID(1..3).          % Possible policy indices per component (up to 3)
policyType(up).     % Policy type: upscaling
policyType(down).   % Policy type: downscaling

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Component Types
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
component_type(db).
component_type(service).
component_type(cache).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Measurements (tied to component types)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
measurement(response_time, service).
measurement(availability, db).
measurement(hit_rate, cache).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Component Instances
% Exactly one type chosen for each cID. 4–8 total components overall.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1 { comp(C,T) : component_type(T) } 1 :- cID(C).

% Enforce 4–8 total components
:- #count { C : cID(C), comp(C,_) } < 4.
:- #count { C : cID(C), comp(C,_) } > 8.
```



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Connections: undirected edges between distinct components
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
{ connected(C1,C2) } :- cID(C1;C2), comp(C1,_), comp(C2,_), C1 < C2.

% Build "edge" relation to simplify handling undirected connections
edge(A,B) :- connected(A,B).
edge(A,B) :- connected(B,A).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Graph Connectivity Constraint
% All chosen components must form a single connected component.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
reachable(X,X) :- cID(X), comp(X,_).
reachable(X,Y) :- cID(X;Y), comp(X,_), comp(Y,_), reachable(X,Z), edge(Z,

% For every pair of components (X,Y), there must be a path from X to Y
:- comp(X,_), comp(Y,_), X != Y, not reachable(X,Y).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SLOs
% Up to 9 SLOs total, each referencing a measurement (M) of some type (T)
% No two SLOs target the same measurement.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
{ slo(S,M,T) : sID(S), measurement(M,T) }.
:- #count { (S,M,T) : slo(S,M,T) } > 9.
:- slo(S1,M,T), slo(S2,M,T), S1 != S2.
```

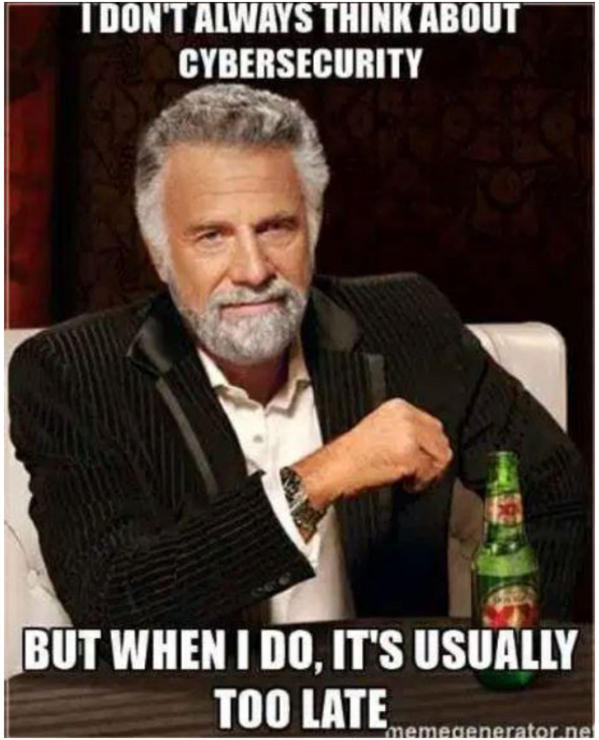


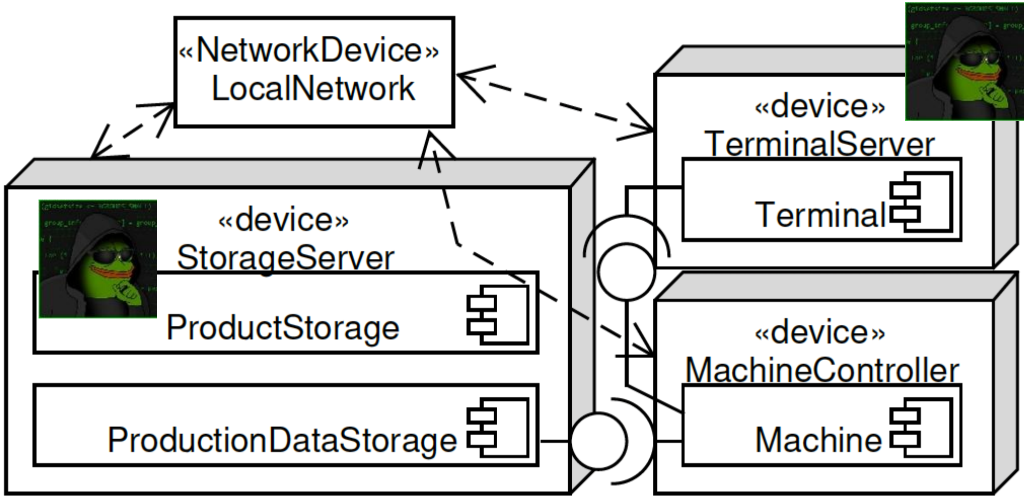
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Policies
% Each component must have 1–3 policies. Each policy is:
%   – Tied to component C
%   – Has an index I in 1..3
%   – Has a policy type (up/down)
%   – Targets a measurement M valid for C's type
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
{ policy(C,I,Type,M) :
    cID(C), comp(C,T), iID(I), policyType(Type), measurement(M,T) }.

% Each component must have 1–3 policies
:- comp(C,_), #count { (I,Type,M) : policy(C,I,Type,M) } < 1.
:- comp(C,_), #count { (I,Type,M) : policy(C,I,Type,M) } > 3.

% A policy must target a measurement that has a corresponding SLO
:- policy(C,I,Type,M), comp(C,T), not slo(_,M,T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#show comp/2.
#show connected/2.
#show slo/3.
#show policy/4.
```



```
Answer: 1
slo(1,response_time,service) slo(9,availability,db) comp(1,service) comp(2,service) comp(3,db) comp(4,db) comp(5,db) comp(6,service) comp(7,db) comp(8,
db) policy(1,1,up,response_time) policy(1,2,up,response_time) policy(2,3,up,response_time) policy(8,2,up,availability) policy(1,2,down,response_time)
policy(6,2,down,response_time) policy(3,2,down,availability) policy(4,2,down,availability) policy(5,3,down,availability) policy(7,2,down,availability)
connected(1,2) connected(1,3) connected(2,3) connected(1,4) connected(2,4) connected(1,5) connected(2,5) connected(1,6) connected(2,6) connected(3,6)
connected(4,6) connected(5,6) connected(1,7) connected(2,7) connected(3,7) connected(4,7) connected(5,7) connected(6,7) connected(1,8) connected(2,8)
connected(3,8) connected(4,8) connected(5,8) connected(6,8)
SATISFIABLE
```

Architectural Security Analysis


I DON'T ALWAYS THINK ABOUT CYBERSECURITY
BUT WHEN I DO, IT'S USUALLY TOO LATE
memegenerator.net



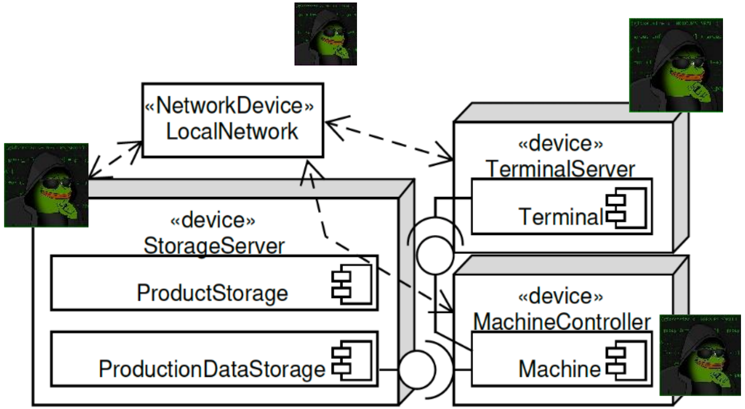
Mitigation of Attack Propagation using Architectural Analysis and Language Models

BA MA

Problem

- An attack can propagate and thus affect the entire cyber-physical systems.

- Selecting appropriate mitigation techniques requires a lot of expert knowledge.



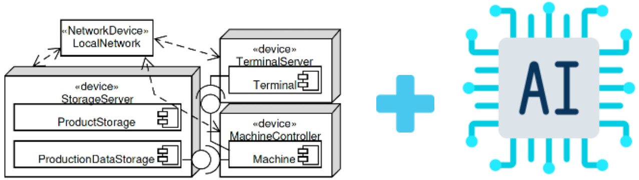
Lan (Mail lan.le@uni-ulm.de)

Tasks

- Research and develop an approach to use architectural analysis and LLM to mitigate an attack.

- Using architecture and asking LLM to identify the vulnerabilities and a suggestion to mitigate the vulnerabilities.

- Analyse the attack propagation with the proposed mitigation.



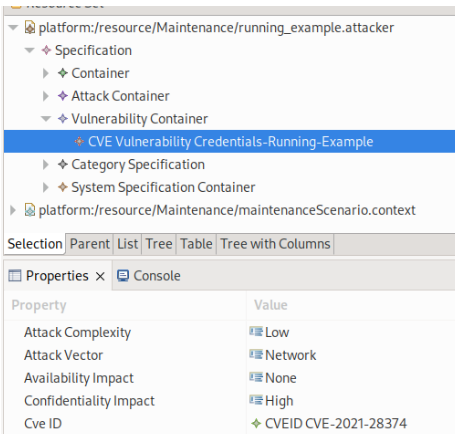
Online Modelling and Analysis Tool to investigate Attack Propagation in Software Architectures

P

Plan to conduct the attacks analysis

Need to install several tools, some tools don't run on your machine.

- Problems



platform:/resource/Maintenance/running_example.attacker
- Specification
  - Container
  - Attack Container
  - Vulnerability Container
    - CVE Vulnerability Credentials-Running-Example
  - Category Specification
  - System Specification Container

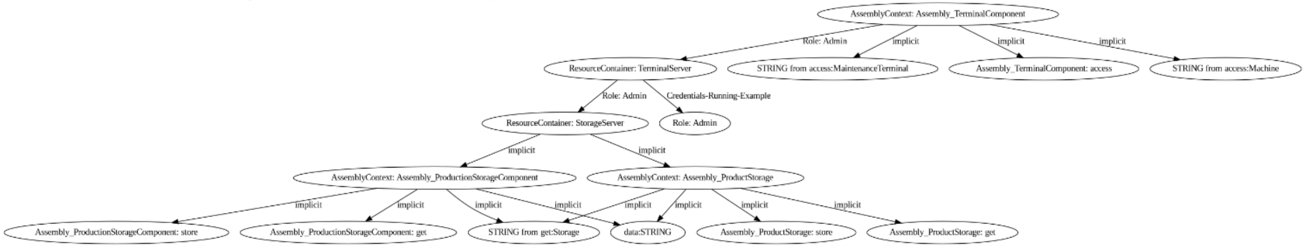
platform:/resource/Maintenance/maintenanceScenario.context

Property	Value
Attack Complexity	Low
Attack Vector	Network
Availability Impact	None
Confidentiality Impact	High
Cve ID	CVEID CVE-2021-28374

- The attack propagation analysis tool is Eclipse-based.

- A graphical tool to model an attacker is needed.

- We need a tool can run online or can be deployed by using Docker.



AssemblyContext: Assembly_TerminalComponent
Role: Admin
implicit
implicit
implicit
ResourceContainer: TerminalServer
STRING from access:MaintenanceTerminal
Assembly_TerminalComponent: access
STRING from access:Machine
Role: Admin
Credentials-Running-Example
ResourceContainer: StorageServer
Role: Admin
implicit
implicit
AssemblyContext: Assembly_ProductionStorageComponent
AssemblyContext: Assembly_ProductStorage
implicit
implicit
implicit
implicit
implicit
implicit
implicit
implicit
Assembly_ProductionStorageComponent: store
Assembly_ProductionStorageComponent: get
STRING from get:Storage
data:STRING
Assembly_ProductStorage: store
Assembly_ProductStorage: get

Lan (Mail lan.le@uni-ulm.de)

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

P

- Tasks

- Develop an online modelling tool to allow modelling an attacker by using a website.

- Develop the function for the online tool to allow conducting an attack propagation analysis.

- Develop the extract function of the online tool to deliver an attractive attack graph.

- Results

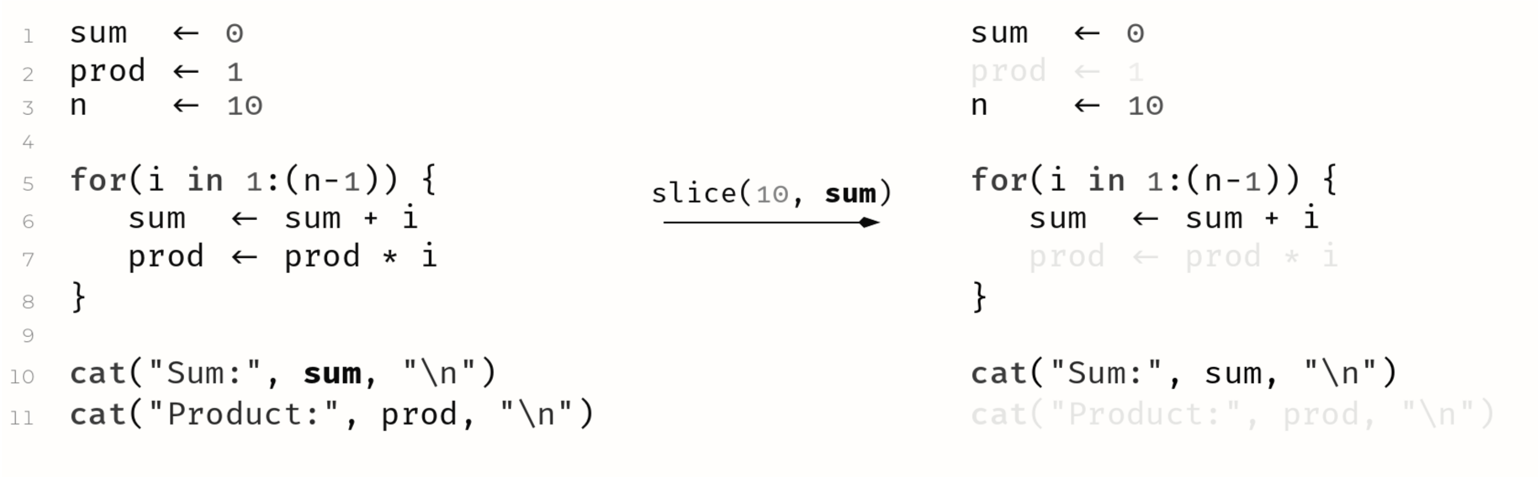
- A website that allows users to conduct an attack propagation analysis.

- The online tool can be shipped as a dockerized package.

Lan (Mail lan.le@uni-ulm.de)

Static Program Analysis

Poking programs to gain some answers



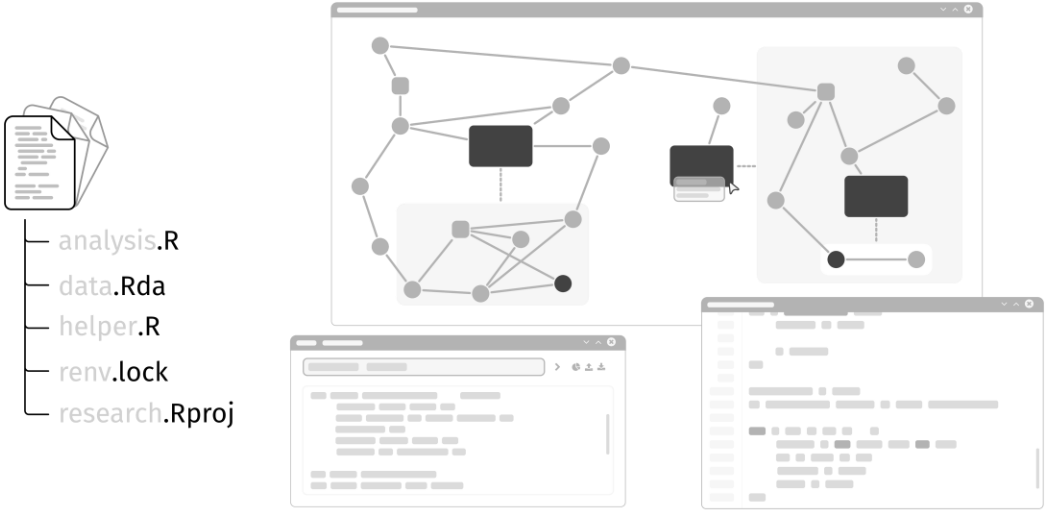
```
1   sum   ← 0
2   prod  ← 1
3   n     ← 10
4
5   for(i in 1:(n-1)) {
6       sum   ← sum + i
7       prod  ← prod * i
8   }
9
10  cat("Sum:", sum, "\n")
11  cat("Product:", prod, "\n")
```

slice(10, **sum**)

```
sum   ← 0
prod  ← 1
n     ← 10

for(i in 1:(n-1)) {
    sum   ← sum + i
    prod  ← prod * i
}

cat("Sum:", sum, "\n")
cat("Product:", prod, "\n")
```

Statische Programm-Analyse für Projekte

P



analysis.R
data.Rda
helper.R
renv.lock
research.Rproj

- flowR
  - Static program analysis for R
  - Support code comprehension, ...
  - Available for VS Code, RStudio, Docker, ...

- Goal
  - Extend flowR with project support
  - Resolve package dependencies
  - Support incremental updates

github.com/flowr-analysis/flowr

Mail to: robert.heinrich@uni-ulm.de

BlockchainBench
An extensible Tool for Modelling and Analysing Blockchain Systems

P

- Blockchain technology enables the operation of distributed ledgers, a form of replicated databases
- Key requirements relate to degree of decentralization (DoD), scalability, and security
- The quality of a given system configuration can be investigated using model-based analysis [ICBC25]

- **Task:** Provide tool support in form of a modeling and analysis workbench for blockchain systems
  - facilitate modeling a blockchain system configuration based on system parameters by developing a Wizard concept
  - allow for selecting several types of analyses and trigger the execution of existing analyses
  - Develop visualization concepts to provide appropriate views to visualize the analysis results
  - Support identifying Pareto-optimal configuration candidates to answer design questions regarding degree of decentralization (DoD), scalability, and security

source: pixabay.com

Yannik Sproll, Robert Heinrich, Lan Bao Quang Le, Niclas Kannengießer. SM-SIM: A Simulator for Analyzing Selfish Mining Attacks in Blockchain Systems. IEEE International Conference on Blockchain and Cryptocurrency. IEEE. 2025.

source: pixabay.com

Retrieval Augmented Generation for Public Sector

DPS
IT FOR FINANCE, RETAIL AND PUBLIC SECTOR

P

- Public service offices struggle with high query volumes, causing delays for citizens and overburdened workforce.
- AI applications could address this issue, however…
  - they tend to “hallucinate” and
  - fall under GDPR-regulations if to be used in public sector.
- The company DPS has an existing AI application but seeks innovative enhancements.
- A team of students will collaborate with the company DPS to improve their app.

Mail to: robert.heinrich@uni-ulm.de

Tasks: Web Development

P

1. **Redesign the User Interface (UI) for Self-Service Devices**
   1. Enhance the layout and design to make it intuitive for citizens, ensuring responsiveness across screen sizes.
   2. Incorporate accessibility features like larger buttons and high-contrast text.
   3. Add visual cues or guided workflows to simplify navigation.
2. **Implement Multi-Language Support**
   1. Enable query handling and responses in multiple languages for diverse demographics.
   2. Add a language detection feature or selection option on the UI.
   3. Localize the document corpus and AI responses accordingly.
3. **Develop an Admin Dashboard**
   1. Build a web-based dashboard for staff to monitor and manage devices.
   2. Include usage analytics, device status, and troubleshooting tools.
   3. Secure it with role-based access control (e.g., for admins and technicians).

Tasks: RAGs and LLMs

P

1. **Fine-Tune the Language Model on Domain-Specific Data**
    1. Use public sector documents (e.g., policies, FAQs) to improve response accuracy.
    2. Test various fine-tuning methods to better handle public service contexts.
2. **Experiment with Different Embedding Models**
    1. Test embedding models (e.g., Sentence-BERT, Universal Sentence Encoder) to boost retrieval accuracy.
    2. Assess trade-offs between size, latency, and precision to select the best model.
3. **Optimize Prompt Engineering**
    1. Refine prompts to produce more relevant and concise responses.
    2. Develop dynamic prompts based on user input for broader query coverage.

Mail to: robert.heinrich@uni-ulm.de

Tasks: Cloud Deployment

P

1. **Set Up a Serverless Architecture**
    1. Shift components (e.g., APIs, AI inference) to serverless functions (e.g., AWS Lambda) for efficiency.
    2. Ensure seamless integration with the existing cloud setup.
2. **Optimize the Database Schema and Queries**
    1. Improve the schema and queries for faster access to data like the document corpus.
    2. Use indexing, caching, or partitioning to enhance performance.
3. **Implement Auto-Scaling**
    1. Configure auto-scaling to manage demand spikes in public service offices.
    2. Add monitoring and alerts to maintain performance without downtime.