

Software Systems Engineering

Bachelor Project Offerings - 2023

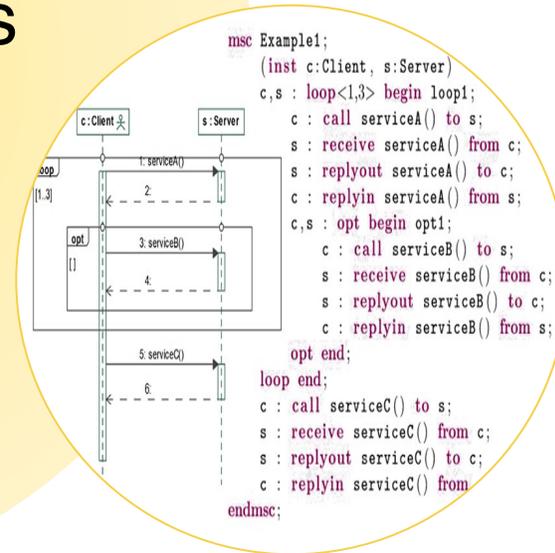


Fundamental Aspects of Computing

Software Systems Engineering



Software and people, things and organizations



Software Engineering

Fundamental Aspects of Computing



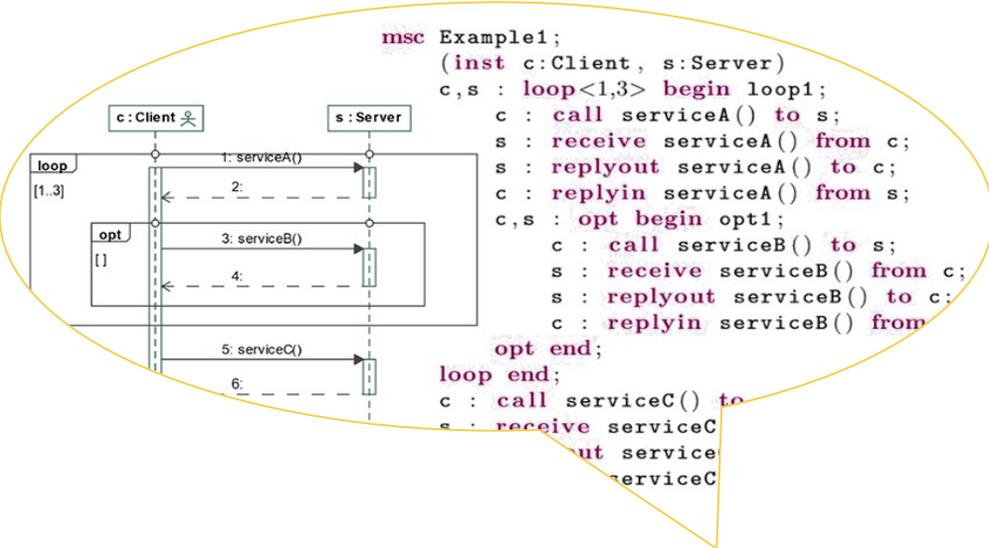
What is it?

We study how software systems behave, and ensure they do not contain errors

Includes: correct and secure software, (specification, modelling, programming) languages, semantics, formal methods, verification, model checking, static analysis, type theory

You might consider this area if you liked any of these courses

Course Id	Name
01241	Computer Science Modelling
02148	Introduction to Coordination in Distributed Applications
02239	Data Security
02242	Program Analysis
02244	Logic for Security
02246	Model Checking
02247	Compiler Construction
02263	Formal Aspects of Software Engineering
02913	Advanced Analysis Techniques
02245	Program Verification



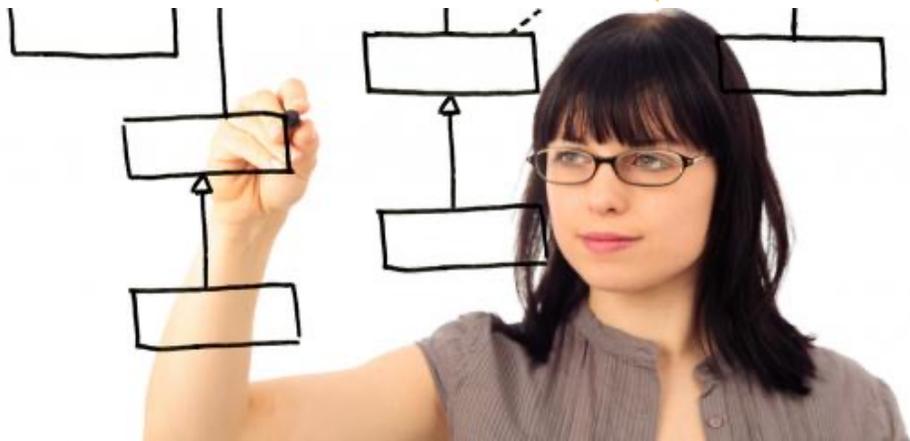
What is it?

We study how large software systems can be built, maintained, and how they evolve.

Includes: methodologies, architectures, empirical methods, Dev-Ops / ML-Ops, AI for Softw. Eng

You might consider this area if you liked any of these courses

Course Id	Name
02160	Agile Object-oriented Software Development
02161	Software Engineering 1
02162	Software Engineering 2
02170	Database Systems
02239	Data Security
02263	Formal Aspects of Software Engineering
02267	Software Development of Web Services
02268	Process-oriented and Event-driven Software Systems
02291	System Integration
02324	Advanced Programming
02362	Project in Software Development
02245	Program Verification



Software and People, Things and Organizations



What is it?

We study and build software with impact in socio-technical systems

Includes: process aware information systems, Business Process Management, process mining, blockchain, Internet of Things

You might consider this area if you liked any of these courses

Course Id	Name
02148	Introduction to Coordination in Distributed Applications
02239	Data Security
02268	Process-oriented and Event-driven Software Systems
02269	Process Mining
02242	Program Analysis

Example Projects

Software and People, Things and Organizations



Problem: Most processes in organisations are documented via texts, but currently there is lack of traceability between what has been written and how real business processes operate.

Method: This bachelor project integrated knowledge from AI/natural language processing and business process management to identify key components of business processes and their semantics.

Outcome: This thesis improved the state of the art in process extraction from unstructured texts, and a summary of the project was published at the International Conference of Advanced Information Systems Engineering (CAISE). The student continued with an MSc in Data Science.

Software Systems Engineering - People



- + 7 PhDs
- <https://www.compute.dtu.dk/english/research/research-sections/software-systems-engineering/employees>

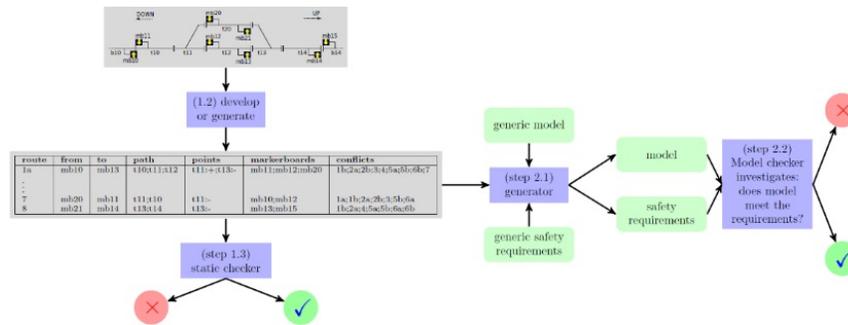


I supervise projects in **Formal Methods**, covering both **Fundamental Aspects of Computing** and **Formal Aspects of Software Engineering**. A project typically covers one or several of the following topics:

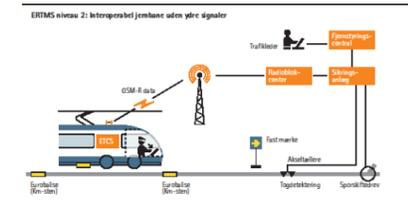
- Design of formal specification & modelling languages and their semantic foundations.
- Formal verification techniques.
- Development of formal method tools.
- Formal modelling and safety verification of cyber-physical systems, e.g. train control systems of the future.



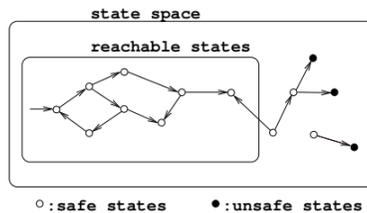
It systems must be safe – avoiding accidents!
Use formal methods to ensure that!



Tools for safety verification.



From Banedanmark.



System models.

$$\Sigma \mid- e1 : t1, rs1, ws1 \quad \Sigma \mid- e2 : t2, rs2, ws2$$

$$\Sigma \mid- (e1, e2) : t1 \times t2, rs1 \cup rs2, ws1 \cup ws2$$

Semantic foundations of languages.



Anne Haxthausen

<http://www.imm.dtu.dk/~aeha/>

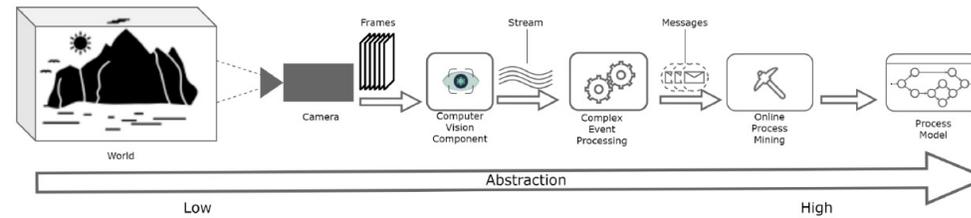
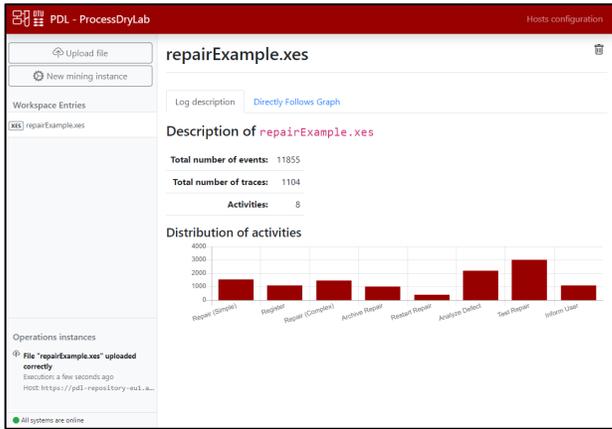
aeha@dtu.dk



Projects in Process Mining

I supervise projects around processes and process mining including

- Streaming process mining
- Software engineering and development aspects of process mining
- Applications of process mining



Online process mining in video streams using computer vision

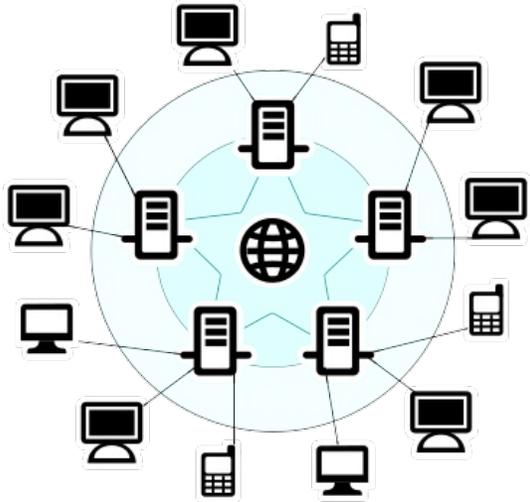
<https://findit.dtu.dk/en/catalog/6216242583fe41235e4fb8af>

<https://beamline.cloud>



Andrea Burattin

<https://andrea.burattin.net/>
andbur@dtu.dk



Concurrent and distributed applications are everywhere, but their development is **challenging**.

We want to make it easier, by creating **tools and libraries based on strong mathematical foundations**, to ensure that distributed programs work as intended.

I supervise projects related to **programming languages** (especially **functional programming**, particularly **Scala 3**), **compilers**, and **formal methods** to model distributed systems.



Programming language

Possible Thesis Topic

Model-based testing of REST or GraphQL applications

Functional effect systems for distributed applications

Metaprogramming for source code analysis and verification

Design and development of **concurrent and distributed runtime systems**

Automatic generation of **monitors for distributed applications**

Development of **user-friendly tools to explain formal methods** (e.g. how type checking works, how a regular expression is matched)

Automatic code generation from models and specifications



Alceste Scalas

<https://people.compute.dtu.dk/alcsc>

alcsc@dtu.dk



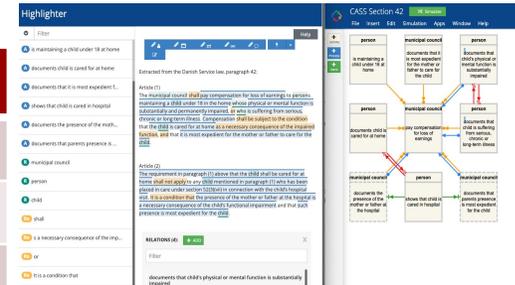
We create theories and tools ensuring that IT applications behave in a lawful way, preserving (human) rights and preserving obligations. The results are formally proven, and empirically validated via case studies and benchmarks.

I supervise projects in **any** intersection between **programming language research, logic/conceptual modelling, process management/mining, natural language processing, and empirical software engineering**

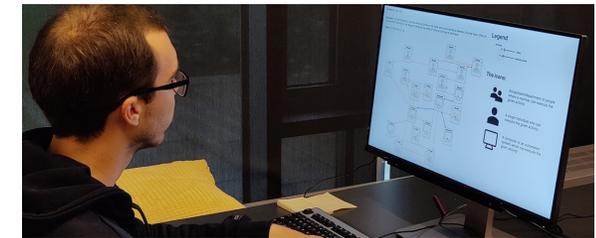
- Possible Thesis Topic**
- NLP and ML techniques for **specification extraction** of legal rights and obligations
- Formal Modelling** of legal contracts
- Process optimization techniques** to reduce carbon footprint in business processes
- Compiler techniques for generation of **safe-by-design distributed systems**.
- Gamification of **Process Simulators**
- Constraint-solving techniques** to identify hidden dependencies in software designs
- Factors affecting the **understandability of Software Engineering Models**

$$\frac{(T-IN) \quad \Gamma_1 \vdash x : ?T.S \quad \Gamma_2, x : S, y : T \vdash P}{\Gamma_1 \circ \Gamma_2 \vdash x?(y).P}$$

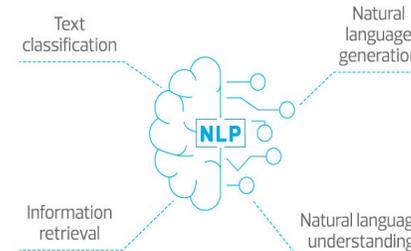
Computer Science



Model-Driven SWEng



Empirical SWEng



Machine Learning/Data Science



Hugo-Andrés López

<http://lopezacosta.net/>

hulo@dtu.dk



We create methodologies and tools for ensuring that programs perform as they are supposed to, using rigorous mathematics and automated reasoning tools.

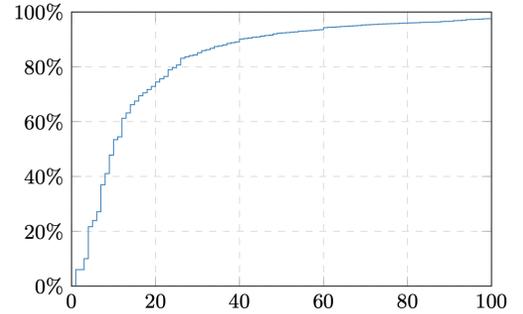
I supervise projects that are broadly concerned with **program verification**, **Rust** programming, verification methodologies, semantics, **probabilistic systems**, model checking, and **decision procedures**.

Exemplary Thesis Topics

- A Pure Backend for Rust Verifiers
- Modular Specifications for Probabilistic Programs
- Automatic Evaluation of Unsafe Code Usage in Large Rust Projects
- Decision Procedures for Quantitative Separation Logic
- Reasoning about Rust Programs with Ghost Code

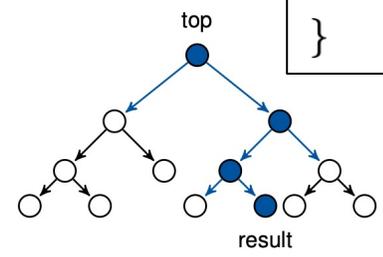
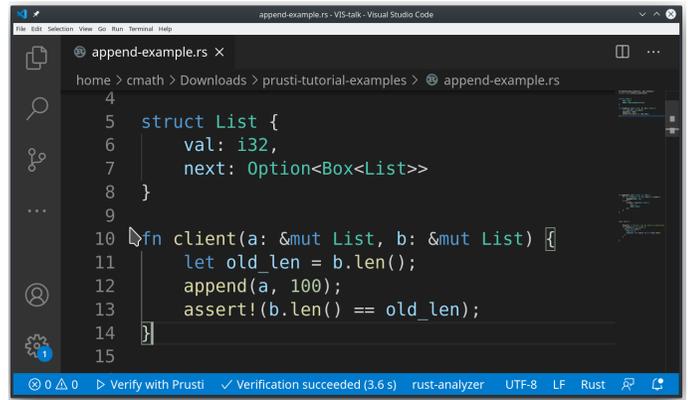
```

rleaf(root) {
  l := <root>;
  r := <root + 1>;
  if (l = 0 and r = 0) {
    out := root
  } else {
    n := 1/2 · ⟨l⟩ + 1/2 · ⟨r⟩;
    out := rleaf(n)
  }
}
    
```



How are programming languages used? Here: *Unsafe Rust blocks*

Automated Program Verification
Here: *Rust verifier Prusti*



Reasoning about randomized algorithms
Here: *meldable heaps*



Christoph Matheja
<https://cmath.eu>
chmat@dtu.dk



Almost like the game of chess:

- Mathematical models of security protocols, of what an intruder can do, and what security goals we try to achieve
- Automated analysis techniques for quickly finding attacks or proving their absence

Possible Thesis Topic

Modeling protocols and **analyzing** them with one or two tools

Post-**compromise** security and multi-factor authentication

Modeling **privacy** and **zero-knowledge proofs**

Accountability: ensure we have somebody to blame when things go wrong

Analysis techniques: work on new model-checking or static analysis techniques

Language design: design new domain-specific languages

Algebraic reasoning: implement algorithms for reasoning about properties of crypto



Sebastian Mödersheim

<https://compute.dtu.dk/~samo/>

samo@dtu.dk

