Biography
Ansgar Fehnker is an Associate Professor (Teaching and Leadership) in the School of Computing at Macquarie University. He was previously an Associate Professor at the University of Twente, a professor in CS/IS at the University of the South Pacific, and a researcher at National ICT Australia in Sydney. His research interest is the use of automated system and software verification tools, in particular model checking and static analysis, and their application in the design and development of software systems. His work includes verification for cyber-physical systems, static analysis for C/C++, modelling and verification of wireless routing protocols, analysis of block-chain technology and more recently, developing and applying software analysis tools and methods for an educational context.

Employment

**Associate Professor (Teaching and Leadership)**
Associate Professor
School of Computing
Macquarie University
15 Nov 2021 → present

**Associate Professor**
University of Twente
NB Enschede, Netherlands
1 Feb 2016 → 14 Nov 2021

**Professor**
University of the South Pacific
Fiji
1 Feb 2012 → 31 Jan 2016

**Researcher**
National ICT Australia Limited (NICTA)
Eveleigh, Australia
1 Nov 2004 → 31 Dec 2012

Research outputs

**What's in school? – topic maps for secondary school computer science**

**Atelier Hands-On Training**
Blaas, R., Fehnker, A. & Mader, A. H., 1 Jun 2021

**Atelier – tutor moderated comments in programming education**
Automated assessment of learning objectives in programming assignments

Be lazy and don’t care: faster CTL model checking for recursive state machines

Out for coffee: with Rob

A distributed blockchain model of selfish mining

A sports day for programming

Atelier - an online platform for programming tutorials

Atelier for creative programming

Atelier for Creative Programming - Poster Video
Fehnker, A. & Mader, A. H., 2020

Atelier – Support for Creative Programming Education

Atelier voor Creatief Programmeren

Double-spending analysis of bitcoin

Preface
Tinkering in informatics as teaching method

Detecting and addressing design smells in novice Processing programs

Adaptive formal framework for WMN routing protocols

An even better approach: improving the B.A.T.M.A.N. protocol through formal modelling and analysis

The smell of Processing

Twenty percent and a few days: optimising a Bitcoin majority attack

Automated Program Analysis for Novice Programmers

Evaluating the stream control transmission protocol using Uppaal

Modelling, verification, and comparative performance analysis of the B.A.T.M.A.N. protocol

Model checking a client-side micro payment protocol

Model checking a server-side micro payment protocol

Modeling and verification of the Bitcoin protocol

Preface

Modeling and verification for the server-side Netpay protocol

Model checking driven static analysis for the real world: designing and tuning large scale bug detection

The quest for precision: a layered approach for data race detection in static analysis

Topology-based mobility models for wireless networks

Automated analysis of AODV using UPPAAL

SMT-based false positive elimination in static program analysis

Cyber security at software development time

Model checking dataflow for malicious input
Modelling and analysis of AODV in UPPAAL

Fade to grey: tuning static program analysis

Counterexample guided path reduction for static program analysis

Software metrics in static program analysis

An abstract specification language for static program analysis

Automatic bug detection in microcontroller software by static program analysis

Incremental false path elimination for static software analysis

On the impact of modelling choices for distributed information spread: a comparative study

Survey on directed model checking

Goanna: syntactic software model checking

Some assembly required: program analysis of embedded system code
Goanna - a static model checker

Model checking software at compile time

Modelling and verification of the LMAC protocol for wireless sensor networks

Hybrid system verification is not a sinecure: the electronic throttle control case study

Formal verification and simulation for performance analysis for probabilistic broadcast protocols

Refining abstractions of hybrid systems using counterexample fragments

Temporal logic model checking

Verification of a cruise control system using counterexample-guided search

Benchmarks for hybrid systems verification

Managing verification activities using SVM

Abstraction and counterexample-guided refinement in model checking of hybrid systems

Specification-guided analysis of hybrid systems using a hierarchy of validation methods
Modeling and verifying a Lego car using hybrid I/O automata

Verification of hybrid systems based on counterexample-guided abstraction refinement

Verification and optimization of a PLC control schedule

As cheap as possible: efficient cost-optimal reachability for priced timed automata

Efficient guiding towards cost-optimality in UPPAAL

Minimum-cost reachability for priced timed automata

UPPAAL - now, next, and future

Scheduling a steel plant with timed automata

Automotive control revisited: linear inequalities as approximation of reachable sets

Operational and logical semantics for polling real-time systems