

Konrad Siek

konrad.siek@gmail.com
kondziu.github.io



I explain things
I write code, I write prose
I do sketchy things to runtimes
I solve simple problems in complicated ways
I abstract

Summary

My **core interests** are

- programming languages, especially abstractions and runtimes
- concurrency, especially transactions
- processing large data
- distributed systems

My **skills** include

- research and synthesis
- programming in numerous languages
- teaching
- technical writing

I **program** in

- (currently) Rust, C, R, Bash
- (previously) Scala, Python, Java, OCaml
- (intermittently) numerous others

I **speak**

- (proficiently) English and Polish
- (learning) Czech and French

Education

Doctorate in Computing Science
Poznań University of Technology

January
2017

Dissertation: *Distributed pessimistic transactional memory: algorithms and properties*

Advisor: Paweł T. Wojciechowski

Reviewers: Marek Tudruj, Michel Raynal

Master's in Computing Science (SE)
Poznań University of Technology

September
2009

Thesis: *A Java source code precompilation tool for static analysis and modification of programs for the Atomic RMI library*

Advisor: Paweł T. Wojciechowski

**Bachelor of Engineering in Computing
Science**

February
2008

Poznań University of Technology

Thesis: *Amebae: a group instant messenger for
developers* (co-author)

Advisor: Bartosz Walter

Bachelor of Arts in English Philology

June 2007

State University of Applied Sciences in Piła

Thesis: *Computer-assisted language learning software:
experimental study*

Advisor: Anna Szczepaniak-Kozak

Employment

Post-doctoral researcher

2017–2022

**Programming Research Lab at Czech Technical
University in Prague**

Responsibilities:

- Research
- Implementation of proof-of-concept applications
- Student mentoring and supervision
- Teaching

Topics:

- Analysis of large code repositories
- Larger than memory object abstraction for the R
language
- R runtime internals survey and dynamic analysis
- Language runtimes in general

Visiting researcher

most of
2017

**Programming Research Lab at
Northeastern University**

Responsibilities:

- Research and data analysis
- Implementation of proof-of-concept frameworks
- Student mentoring
- Teaching

Topics:

- Lazy evaluation in the R runtime
- Programming language research

Research assistant

2013–2017

**Distributed Systems Group at Poznań University of
Technology**

Responsibilities:

- Research
- Implementation of proof-of-concept frameworks
- Student mentoring
- Teaching

Topics:

- Distributed transactional memory implementation
- Transactional memory safety properties
- Static analysis and code generation

Developer at Poznań University of Technology 2009–2012

Responsibilities:

- Design and implementation of proof-of-concept applications

Topics:

- Static analysis of critical sections in distributed projects
- Code generation and Java bytecode instrumentation

Developer at PSI Poland 2008–2009

Responsibilities:

- Back-end programming
- Database management

Topics:

- Automotive factory order management system software

Apprentice English Language Teacher at 2005–2006

Elementary School No. 4 in Piła

Responsibilities:

- Teaching (under supervision)
- Mentoring

Projects

UFOs: Lazy larger-than-memory object arrays February via userfaultfd 2019

User provides an arbitrary function to populate a chunk of memory. Framework allocates an area of memory and transparently executes the population function when a chunk is read or written to. Chunks are seamlessly garbage-collected and re-generated as needed. Example implementations generate in-memory arrays from columns in CSV files, BZIP file, and formulas. Comes with C, R, and Rust bindings.

My contribution:

- Back-ends
- R bindings and utilities
- Parts of garbage collection

CodeDJ: Reproducible queries over large-scale software repositories July 2020

Infrastructure for querying GitHub and similar repositories for quantitative software engineering research (especially project selection) in large code datasets. It prioritizes reproducibility and scalability and consists of two modules. *Parasite* is an incremental downloader and persistent datastore. *Djanco* is an in-memory database and query language embedded in Rust.

My contribution:

- Djanco DSL
- Djanco database and surrounding tooling

FML: A small runtime for teaching runtimes February
Toy bytecode compiler and interpreter 2021
designed as a model for student implementations in a
runtimes class Runs a vaguely ML-like toy dynamic
language with objects, inheritance, dynamic dispatch and
garbage collection but not much else. The compiler
generates slightly extended Feeny bytecode (another
teaching language) consisting of 17 ops and 7 internal
objects.

Rust-delegate: Method delegation generator July 2021
macro for Rust
A Rust macro that generates method delegation to inner
fields within structs.

My contribution:

- Syntax for injecting arbitrary expressions as arguments to delegated functions

TinyTracer: A minimalist tracer for analyzing July 2018
the composition of R objects
R 3.5 runtime variant instrumented to analyze objects at
garbage collection. The tracer records the types of each
object, and the types object in all the slots slots in each
object. Used to find rare and anomalous object
construction.

R-dyntrace: A dynamic tracer for R 2017–2018
An infrastructure for programmable probes into the R
runtime. Used to perform custom dynamic analysis of R
code. Subsequently used in research into lazy evaluation
in R programs

My contribution:

- Overall design and implementation of probes in the previous version

GHGrabber: Small Git scraper 2019–2020
A small multi-process bash program that gathers basic
information about a Git repositories from a list of URLs
and outputs CSV files. Used to collect large datasets for
teaching and software engineering research.

AtomicRMI: Pessimistic distributed 2010–2016
transactional memory system over Java RMI
Implementation of pessimistic transactional concurrency
control for Java RMI RMI objects are instrumented to The
algorithm assigns versions to shared objects and uses
them to guide how transactions lock and release them It
uses upper bounds on the number of accesses of an
object within transactions to release locks early, if this is
safe. It also uses local buffers to defer the need to
synchronize transactions in specific situations.

My contribution:

- Optimizations to the original algorithms
- Design and implementation of most the framework

GrittyScripts A blog collecting miscellaneous scripts 2008-2016

A blog presenting small useful scripts in Bash, Python, AWK, OCaml, and other languages with some analysis of the problem and attempts at clever solutions and dirty hacks.

Teaching

Runtime systems (NI-RUN B202) at **Czech Technical University in Prague** 2019-2021

Course on process virtual machines for programming languages teaching basic concepts, taxonomy of VMs, architecture of runtimes, bytecode compilation and interpretation, memory management, and just in time compilation.

Designed and taught independently

Object-oriented programming (BIE-OOP B191) at **Czech Technical University in Prague** 2018

Course on the principles of object oriented programming and design in Scala with an emphasis on practical techniques for developing complex software and software engineering skills: testing, error handling, refactoring, design pattern.

My contribution: leading a lab group and lectures on object-oriented design

Led by: Filip Křikava

Expeditions in Data Science (DS6050) at **Northeastern University** 2017

Course on practical problems of data science projects, teaching importing, tidying and transformation of large data, statistical modeling, visualization, repeatability and reproducibility of results.

My contribution: assignment design and lectures on databases

Led by: Jan Vitek

Parallel data processing in MapReduce (DS6240) at **Northeastern University** 2017

Course on distributed processing of large data involving Hadoop and Spark, H2O, and Tensorflow.

My contribution: assignment design and lectures on pipeline processing, Spark and Hadoop

Led by: Jan Vitek

Safe programming methods (MBP) at **Poznań University of Technology** 2014–2016

Course on safe programming methods and languages in the context of concurrent and distributed systems: memory models, monitors, transactional memory, message passing, map-reduce. Labs involved a complete

functional programming course in Scala explaining techniques like currying, lazy evaluation, and trampolining, as well as programming with actors.

My contribution: designed and implementation of the functional programming sub-course as teaching associate
Led by: Paweł T. Wojciechowski

Network Programming (SK2) at Poznań University of Technology 2016

Network programming and advanced network concept course. C systems programming involving sockets and threads. Ad-hoc wireless networks, load balancing, DNS, and VLAN.

My contribution: leading the labs as teaching associate, preparing course materials and assignments
Led by: Michał Sajkowski

Networks (SK1) at Poznań University of Technology 2012–2013

Introductory course to computer networks: IP addressing, subnets, structural cabling, OSI model with details on physical, network, and transport layers, static and dynamic routing.

Contribution: leading the labs as teaching associate, preparing course materials and assignments
Led by: Michał Kalewski

Operating systems (SOP) at Poznań University of Technology 2009–2016

Introductory course to operating systems: OS definition, processor scheduling, virtual memory, I/O, and file systems. The laboratory portion involved an introduction to Linux and Bash

Contribution: leading the labs as teaching associate, preparing course materials and assignments
Led by: Dariusz Wawrzyniak, Anna Kobusińska, and Michał Sajkowski

Basic IT (PIN labs) at Poznań University of Technology 2009–2017

Introductory course on basic concepts, usage, and tools of informational technology for students of Poznań University of Medical Sciences.

Contribution: leading labs as teaching associate
Led by: by Izabela Szczęch

Papers

CodeDJ: Reproducible queries over large-scale software repositories July 2021

Petr Maj, Konrad Siek, Alexander Kovalenko, and Jan Vitek
In Proceedings of ECOOP'21: European Conference on Object-Oriented Programming
DOI: 10.4230/LIPIcs.ECOOP.2021.6 and artifact: [🔗](#)

Last-use Opacity: A Strong Safety Property for Transactional Memory with Prerelease Support Submitted December 2019

Konrad Siek, Paweł T. Wojciechowski
Distributed Computing. *To appear.* Accepted December 2021

Helenos: A realistic benchmark for distributed transactional memory March 2018

Paweł Kobyliński, Konrad Siek, Jan Baranowski, and Paweł T. Wojciechowski
Journal of Software: Practice and Experience, vol. 48, iss. 3
DOI: 10.1002/spe.2548

Proving opacity of transactional memory with early release December 2015

Konrad Siek and Paweł T. Wojciechowski
Foundations of Computing and Decision Sciences, vol 40, issue 4
DOI: 10.1515/fcds-2015-0018

Atomic RMI: a distributed transactional memory framework April 2015

Konrad Siek and Paweł T. Wojciechowski
In Proceedings of HLPP 2014: Symposium on High-level Parallel Programming and Applications
International Journal of Parallel Programming, vol, 44, issue 3
DOI: 10.1007/s10766-015-0361-x

A formal design of a tool for static analysis of upper bounds on object calls in Java August 2012

Konrad Siek and Paweł T. Wojciechowski
In Proceedings of FMICS 2012: Workshop on Formal Methods for Industrial Critical Systems
Lecture Notes in Computer Science, vol. 7437
DOI: 10.1007/978-3-642-32469-7_13

Barcode scanning from mobile phone camera photos delivered via MMS: Case study October 2008

Adam Wojciechowski and Konrad Siek
In Proceedings ER 2008: International Conference on Conceptual Modeling
Lecture Notes in Computer Science, vol. 5232
DOI: 10.1007/978-3-540-87991-6_27

Userfault objects: transparent July 2021

programmable memory

Konrad Siek and Colette Kerr

In Proceedings of ICIOOLPS'21: Workshop on Implementation, Compilation, Optimization of Object-oriented Languages, Programs and Systems

Larger-than-memory R July 2020

Konrad Siek and Colette Kerr

In Proceedings of useR'20: The R User Conference

Atomic RMI 2: distributed transactions October
for Java 2016

Paweł T. Wojciechowski and Konrad Siek

In Proceedings of AGERE'16: Workshop on Programming Based on Actors, Agents, and Decentralized Control

Relaxing opacity in pessimistic October
transactional memory 2014

Konrad Siek and Paweł T. Wojciechowski

In Proceedings of DISC'14: Symposium on Distributed Computing

Zen and the art of concurrency control: July 2014
An exploration of TM safety property space with early release in mind

Konrad Siek and Paweł T. Wojciechowski

In Proceedings of WTTM'14: the 6th Workshop on the Theory of Transactional Memory

Having Your Cake and Eating it Too: April 2014
Combining Strong and Eventual Consistency

Paweł T. Wojciechowski and Konrad Siek

In Proceedings of PaPEC 2014: Workshop on the Principles and Practice of Eventual Consistency

Towards a Fully-Articulated Pessimistic July 2013
Distributed Transactional Memory

Konrad Siek and Paweł T. Wojciechowski

In Proceedings of SPAA 2013: the 25th ACM Symposium on Parallelism in Algorithms and Architectures

Rollbacks in Pessimistic Distributed TM June 2013

Paweł T. Wojciechowski and Konrad Siek

SRDC'13: TRANSFORM Summer School on Research Directions in Distributed Computing

Transaction Concurrency Control via April 2012
Dynamic Scheduling Based on Static Analysis

Paweł T. Wojciechowski and Konrad Siek In Proceedings of WTM 2012: Euro-TM Workshop on Transactional Memory

Statically Computing Upper Bounds on July 2010
Object Calls for Pessimistic Concurrency Control

Konrad Siek and Paweł T. Wojciechowski

In Proceedings of EC² 2010: Workshop on Exploiting Concurrency Efficiently and Correctly

Supervised theses

- Nilay Baranwal:** Structured printing framework ongoing
Bachelor thesis at Czech Technical University in Prague
As supervisor
- Jan Jindráček:** Usability improvements to JavaScript/ECMAScript 2021
Master thesis at Czech Technical University in Prague
As supervisor
- Jakub Cieślak and Kamil Kozubal:** 2016
Implementation of distributed transactional memory
Engineering thesis at Poznań University of Technology
As assistant supervisor under Paweł T. Wojciechowski
- Jan Baranowski:** Benchmarks for evaluating distributed transactional memory 2015
Master thesis at Poznań University of Technology
As assistant supervisor under Paweł T. Wojciechowski
- Martin Witczak:** Atomic Café—A distributed multimedia playback system 2015
Master thesis at Poznań University of Technology
As assistant supervisor under Paweł T. Wojciechowski

Volunteer and organizational work

- ETAPS'19:** European Joint Conferences on Theory and Practice of Software 2019
Role: poster chair
- SPLASH'18:** Systems, Programming, Languages, and Applications: Software for Humanity 2018
Role: poster chair
- COST Action IC1001:** Transactional Memories: Foundations, Algorithms, Tools, and Applications 2015
Role: management committee substitute member for Poland
- PIWO:** Poznańska Impreza Wolnego Oprogramowania (Poznań Free Software Meetup) 2014–2016
Role: lead organizer and resurrector
- KN SKISR:** Koło Naukowe Sieci Komputerowych i Systemów Rozproszonych (Networks and Distributed Systems Student Club) 2012–2016
Role: faculty organizer
- District Municipal Public Library in Piła CMS system** 2007
Role: development and deployment for a content management system for the local library website

Popular science talks and programming courses:

- *Remedial Scala* 2019
- *Transactional safety primer (aka Konrad does Transactions)* 2018
- *System Aktorów (Actor Systems)* 2016
- *Podstawy Programowania w Pythonie (Basics of Python Programming)* 2015
- *Programowanie Funkcyjne w Pythonie (Functional Programming in Python)* 2013
- *Warsztat Python (Python Workshop)* 2013
- *Bezbolesne Programowanie Współbieżne (Painless Concurrent Programming)* 2012, 2014

Hobbies

Inking (like drawing, but with ink)
Taking overexposed photos
Uncool musical instruments
Bad sci-fi audiobooks
Explaining how language works
Kendo for one summer and then never again