Curriculum vitae

Danil Annenkov
danil.v.annenkov@gmail.com
+4581900892
GitHub: https://github.com/annenkov/
Personal page: https://dannenkov.me

PERSONAL INFORMATION

Birthdate: 16.11.1981.
Marital status: married, wife: Anna Annenkova.
Children: daughter Arina Annenkova.
Citizenship: Russia.
Current location: Aarhus, Denmark.

PROFILE

I am an experienced software developer working comfortably with multiple programming paradigms. I am especially interested in applications of functional programming, domain-specific languages, and proof assistants to make better software.

My recent focus at Concordium is the Rust/Wasm stack for smart contracts and integration of cryptographic libraries from the Rust ecosystem (e.g. dalek cryptography and artworks) into the Concordium infrastructure. As part of the internal science team, I also participated in specifying a zero-knowledge query language used in the Concordium ID layer and the verifiable credentials framework.

I have research and development experience related to the financial domain: I studied stochastic models of financial instruments during my master's and later, during my Phd studies, I accompanied this knowledge with a domain-specific language for financial contracts. My recent experience is related to another form of contract language: smart contracts for blockchains.

EMPLOYMENT

Senior Scientist (current employment) **from September 2022**
At Concordium, Aarhus, Denmark
R&D in smart contract technology and formal methods. Software development in Rust.
Specifications for Concordium's verifiable credentials framework. Integration of Rust cryptographic libraries into the Concordium's infrastructure.

Postdoc Researcher **December, 2018 - August 2022**
At COBRA, Aarhus University, Aarhus, Denmark

Postdoc Researcher **February, 2018 - November, 2018**
At INRIA, Nantes, France

PhD Fellow/Research Assistant **November, 2014 - January, 2018**
At The University of Copenhagen, Copenhagen, Denmark

Founder of a Software Development Company **2009 - 2014**
Web-based enterprise solutions
(CRM, Service Department Information System, e-Commerce)
Python/Javascript Developer **September 2013 - January 2014**
At Exploriana Inc., remote.
Member of the guidemore.com development team.
Part-time teacher **2010 - 2012**
At Irkutsk State Technical University, Russia
Programmer **2004 - 2005**
At Railroad Clinical Hospital, Irkutsk, Russia
Deployment of a medical information system
DevOps/Programmer
At Demetra, Irkutsk, Russia **2002 - 2004**
Windows/Novell Netware administration
Software Development: Delphi, InterBase RDBMS
Research Assistant **2002 - 2006**
At Irkutsk State Technical University, Russia

EDUCATION

PhD, University of Copenhagen, DIKU **2014 - 2017**

Graduated with distinction from Irkutsk State Technical University,
Faculty of Cybernetics, Department of Automated Systems (5 of 5 years)
(equivalent to a master's degree) **July, 2002**

SKILLS

10+ years of programming experience (both industry and academic).
Programming languages
Functional programming languages: Haskell, Scala, Racket (Lisp).
Smart contract languages: CameLIGO (Tezos), Liquidity (Dune), Solidity (Ethereum), Rust (Concordium)
Other: Java, Python, JavaScript, Rust, Matlab, Object Pascal, Borland Delphi.
Proof assistants
Coq, Lean, Agda
Version control systems
Svn, Git, Mercurial
Team Leadership
Was a team leader for a small group of developers (3-4 persons).
Misc
Have experience with agile development techniques, testing, and continuous integration.

ACADEMIC PUBLICATION SUMMARY

Google Scholar statistics: https://scholar.google.com/citations?user=JjtxzfkAAAAJ&hl=en

Publications	20
Citations	203
h-index	7
I10-index	5

Publications in top journals and conferences on formal verification and functional programming:

- Journal of Functional Programming (JFP)
- International Conference on Functional Programming (ICFP)
- Conference on Certified Programs and Proofs (CPP)
- Principles and Practice of Declarative Programming (PPDP)

SOFTWARE

1. **Concordium's smart contract framework**
   - Integrating a property-based testing library
   - Developing new smart contracts for the Concordium ecosystem.
   - Improving documentation.
   - Collecting and documenting best practices for developing smart contracts for the Concordium blockchain.
   - Smart contract code review, including external partners.
   - Integrating new cryptographic libraries into the Concordium infrastructure.

   Technology: Rust, property-based testing, Cryptographic libraries.

   Project repositories:
   - https://github.com/Concordium/concordium-rust-smart-contracts
   - https://github.com/Concordium/concordium-base
2. **ConCert**

   (one of the main contributors)

   A framework for smart contract verification in and code extraction from the Coq proof assistant.
   - A verified embedding for a functional language into Coq.
   - An execution model formalisation for functional smart contract languages.
   - Code extraction to functional smart contract languages (Dune - Liquidity, Tezos - CameLIGO, Concordium - Rust) and general-purpose languages (Rust, Elm).

   Project repository: https://github.com/AU-COBRA/ConCert

   Some stats: ~50 KLOC, 6 contributors, 3 master's projects, 4 papers (3 co-authored), industrial collaboration with Concordium and Tezos, academic collaboration with MetaCoq team (INRIA) and Futhark team (DIKU).

CONTRIBUTIONS TO OPEN-SOURCE PROJECTS

1. **Contract DSL**

   (contributed as part of my PhD)

   Extensions of the domain-specific language for financial contracts by Patrick Bahr, Jost Berthold and Martin Elsman. “Certified Symbolic Management of Financial Multi-Party Contracts”. ICFP’2015. Compilation of the DSL to the intermediate Payoff language implemented in the Coq proof assistant.

   Technology: the Coq proof assistant, Haskell.

   Project repository: https://github.com/annenkov/contracts

2. **The HIPERFIT Portfolio Management Prototype**

   (developed as part of my PhD, used for bachelor’s projects)

   A web-based system that integrates the Contract DSL and the HIPERFIT parallel pricing engine (FINPAR). The system is written in Haskell and features automatic web-form generation on the base of Haskell data types along with OpenCL code generation for payoff functions.

   Technology: Haskell, OpenCL.

   Project repository: https://github.com/HIPERFIT/prototype

3. **MetaCoq**

   (contributed as part of the ConCert project development)

   Meta-programming in Coq and formalisation of the meta-theory of Coq in Coq. Contributed with the support for definitionally proof-irrelevant propositions (SProp) in MetaCoq, after SProp was added to Coq.

   Technology: the Coq proof assistant.

   Project repository: https://github.com/MetaCoq/metacoq/

LANGUAGES

Russian - native

English - fluent (IELTS score 7.0 in 2014)

Danish - completed DU3 (PD3 exam), corresponds to the B2 level of CEFR (according to uim.dk)