

Dr. Gergő Érdi — Curriculum Vitæ

Personal information

Name: Dr. Érdi, Gergő
Location: Singapore
Telephone: +65 9186-8336
E-mail: gergo@erdi.hu
Home page: <http://gergo.erdi.hu/>



Education

2003–2011 M.Sc. in Computer Science, Eötvös Loránd University, Hungary
1996–2006 M.D., Semmelweis University of Medicine, Hungary

Previous work

2011– Software developer, [Standard Chartered Bank](#). Rapid development of end-user GUI applications for traders and financial structurers, including building the infrastructure for delivering applications over the web
2008–2010 Software developer, Emacs integration of [RefactorErl](#), a refactoring tool for Erlang: Emacs Lisp project, Eötvös Loránd University
2005–2011 Software developer, [Intentional Software Corporation](#), Domain Workbench/Structural Editor. Research and implementation of a DSL workbench for language-oriented development
1999–2005 Software developer, various Free Software projects including the [GNOME](#) desktop environment, [Evolution](#) groupware suite, [Guikachu](#) GUI development tool and the [GTKmm](#) C/C++ language interoperability bridging solution

Functional and formal programming methods

Functional languages at SCB At Standard Chartered Bank, we use a software stack written in part in Haskell, and exposed to developers of end-user applications as a scripting language that is an in-house, strict dialect of Haskell.
Dependent types Recently started using Agda. Contributed to the Agda standard library. [Modular arithmetics library](#) (work in progress).
Alef Type checker, interpreter and compiler for a lazy functional language with Hindley-Milner type system; written in Common Lisp
Tandoori Compositional type checker for Haskell 98; written in Haskell. M.Sc. thesis at Eötvös Loránd University.

Desktop applications

<i>Financial applications at SCB</i>	Development of interactive GUI applications (Windows and web front-ends) for traders and financial structurers. I developed a DSL for high-level description of structured product editors.
<i>Intentional Domain Workbench</i>	Structural editor tool to create schema, editable projections, a type system, semantic validators, and compilers for domain-specific languages (DSLs). Large-scale .NET project written in C# and in-house languages (some of them functional).
<i>Guikachu</i>	High-level graphical editor for the RCP resource description language of the PalmOS handheld platform. Large-scale C++ project.
<i>GNOME desktop</i>	Contributions to open source projects including the Evolution groupware suite and the Gnumeric spreadsheet.

Compiling and bridging

<i>Haskell to Javascript</i>	To turn the existing and growing codebase of in-house GUI applications into web applications, I wrote a new backend for the compiler of our Haskell dialect that emits Javascript. With some hand-written Javascript runtime for building and manipulating DOM trees, we managed to compile the same code base to native Windows applications and web applications with the editing logic running on the client side and expensive calculations (like pricing of trades) on the server side; runtime written in Javascript, compiler written in Haskell
<i>MetaFun</i>	Compiler for a Haskell-like functional language into C++ compile-time template metaprograms; written in Haskell
<i>GTKmm</i>	C++ wrapper around the C API of the GTK+ GUI toolkit. GTKmm presents the developer with a native C++ interface, using features of C++ such as class inheritance and templates to maximize productivity, and using compile-time type safety to help applications become more robust.

Languages

- Hungarian: Mother tongue
- English: Fluent
- German: Rusty