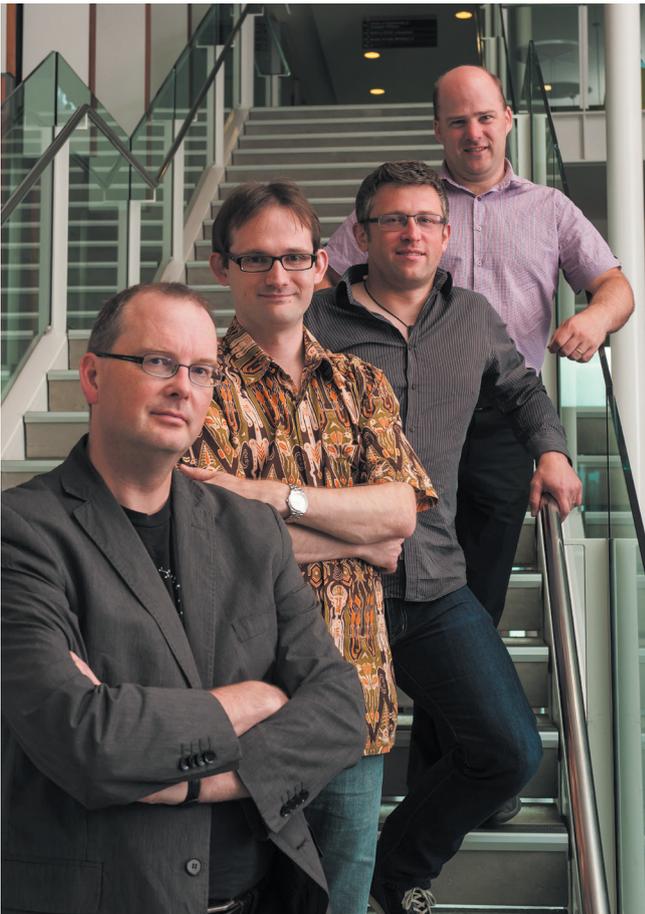


Software Engineering Research at Victoria

Welcome to Elvis, the software engineering research group in the School of Engineering and Computer Science at Victoria University of Wellington, New Zealand. Our work ranges from more technical research concerning the structure of software and tool support for development, to human-computer interaction and software development processes. We have experience and interests in a wide range of topics and applications in the object-oriented paradigm, including programming languages, visualisation, design patterns, software reuse, frameworks, and agile methods, as well as in interface design and evaluation. Below we highlight the major current projects and past and current industry collaborations in our group.



Professor James Noble, Dr Alex Potanin, Dr David Pearce and Dr Stuart Marshall.

Find out more at www.elvis.ac.nz or if you have further questions, please contact Dr Alex Potanin at alex@ecs.vuw.ac.nz who can put you in touch with the relevant members of our group.

PROGRAMMING LANGUAGES

Professor James Noble, Dr Alex Potanin, Dr David Pearce, A/Prof Thomas Kühne, and Dr Marco Servetto lead the programming languages subset of Elvis.

- **Wyvern** - aiming a secure DSL-friendly web-programming, co-led by Alex Potanin
- **Whiley** - aimed at formally-verified applications, led by David Pearce
- **Grace** - aimed at computer science educators, co-led by James Noble
- **Deep Java** - aimed at reducing accidental complexity, led by Thomas Kühne
- **L42** - aimed at library designers, led by Marco Servetto

Wyvern recently made the news with a breakthrough in enabling safer web programming without SQL injections and similar attacks. From the article:

“Common coding practice leaves behind vulnerabilities,” Dr Alex Potanin explains. “Different programming languages solve different problems—no single programming language is right for everything—so developers have to use a multitude of languages to create different functions in an application.” Dr Potanin says that while HTML works well with webpages, JavaScript helps make them responsive, and SQL is good for receiving data from a database, when developers try to use these languages together it often leads to bottlenecks and vulnerabilities. “If differing languages are not implemented well together, applications can be left open to cross-site scripting attacks, SQL injection attacks and code injection attacks—three of the most severe security threats in web applications today.” Dr Potanin says the research project—known as Wyvern— involves developing a kind of ‘programming Esperanto’ which unites all the different programming languages under a single umbrella. “We developed Wyvern as a way of stitching languages seamlessly together, so that programmers can work more productively and avoid vulnerabilities,” he said. “It’s also very easy for programmers to switch from one language to another within Wyvern, which lowers the barrier to getting the benefits of multiple languages in one programme.”

Thomas Kühne started the design of Deep Java based on the observation that the traditional object-oriented paradigm limits modelers and programmers to only two-levels (classes and objects) and hence forces them to introduce accidental complexity in their models and programs. DeepJava adds an unbounded number of classification levels on top of Java objects and Java classes. One of the consequences is that classes and their (potentially new) features may be created at runtime just like objects. DeepJava is unique in the way it makes such flexibility possible without comprising static type safety.

Other work included ownership types (as used in Rust), performance analysis (together with Mozilla Corporation), security (funded by government agencies), advanced program analysis (together with a NZ-based start-up, Innaworks) and many others.

We are interested in further collaborations around modern object-oriented programming languages, security, capabilities, program analysis, and advanced type systems.

SOFTWARE VERIFICATION

A/Prof Lindsay Groves, Dr David Streader, Dr Alex Potanin, Dr David Pearce work in the areas of software verification and formal methods. Major industrial projects include collaborations with Oracle, Transpower and other clients as well as research and publications around algorithm verification, machine assisted proofs (using PVS and COQ), and extended static analysis. One project

currently being undertaken is the development of a programming language suitable for software verification. This language is called Whyley and aims to automatically eliminate a large range of software errors. This would be particularly useful in the area of safety-critical systems, where software errors can endanger human life, modern cars being one such example.

INFORMATION VISUALISATION

Dr Stuart Marshall, Professor James Noble and others are also working in the field of information visualisation. We are particularly interested in designing and evaluating interactive visual representations of large data sets. Often these large data sets do not have a default spatial layout (unlike a land map for example) and may have many dimensions or aspects that need to be represented. An interactive, explorable visualisation of the data can reveal information that is otherwise hidden in the text-based

mass of raw data. Staff and students in the software engineering research group have previously created visualisations that allow users to first generate an overview, before interactively exploring and filtering the information to best identify trends, patterns and answers in the data. We are also interested in exploring how users can collaboratively work together to explore large data sets, both in a work setting and in an educational setting.

AGILE SOFTWARE DEVELOPMENT

Professor James Noble and other members of our group are involved in Agile Development studies including ground-breaking research on the customer's role in Agile projects, the impact on software architecture in the Agile setting and running a top-rated Agile Methods course at Victoria. We have engaged with industry to identify common practice of how Agile teams develop software. This research has enabled us to collate undocumented and low-

level Agile practices across the industry in New Zealand and around the world.

The software engineering research group is interested in exploring the consequences, pitfalls and benefits of various Agile practices, and the barriers that naturally arise when trying to practice the Agile philosophy. We welcome industry collaboration.

DATABASES AND MODELLING

Dr Hui Ma, A/Prof Thomas Kühne and other members of our group work in the modelling and databases research areas, including

a recent collaboration on the data quality in the transportation industry where safety concerns are paramount.

META PROGRAMMING

L42, designed by Dr. Marco Servetto, aims to allow seamless integration of multiple libraries in the same software product using a form of safe meta-programming to automatically compose code. L42 is an object-oriented, nominally typed language that should feel familiar to Java programmers. The meta-programming also

allows to automatically generate most of the boilerplate code, in a similar style as what is now offered by "project lombok", but in a much more robust, reliable and efficient way. The strong nominal type system in L42 allows one to express mutability and aliasing constraints, allowing optimizer to perform automatic parallelization.