Resume: Type Systems And Programmers: A Look at Optional Typing in Dart

Mark Faldborg Troels Lisberg Nielsen

Supervisor:
Bent Thomsen

June, 2015

In this thesis we start out by categorising disciplines related to programming language research and design. Making a clear relation between programming language design and the discipline of software development. We brefily describe the history of types in programming languages and how different disciplines affect types as we know them today. The process of designing, building, and evaluating programming language (constructs) is introduced with an emphasis on the evaluation aspect. The applicability and relevance of known evaluation techniques in relation to our work with evaluating types and type systems is summarised. We arrived at the idea that user/case studies including interviews and controlled empirical experiments can inform the "static versus dynamic typing"-debate.

We then touch on some epistemological considerations and techniques used in our work:

- Distinguishing between inductive and deductive processes in science.
- Empiricism and constructivism.
- Quantitative and qualitative research.

A short description of the tools, Dart and DartPad is included together with a set of modification made to help with experimentation.

We do a walk through and analysis of quantitative data gathered in a controlled empirical experiment, we are unable to reject our null hypothesis, and find two interesting patterns in completion times for small tasks.

We also walk through eight interviews with discussions of what we find from these, the interviews give us some insight into what programmers think of types systems, specifically dynamic versus static typing.

The final parts of the thesis concludes on previous sections and gives suggestions for future research in the are of type systems. Among other things suggesting an online version of the same experiment with automated surveys.