



Proposal bachelor thesis

Title: IDE for tierless web programming

Promotor: Coen De Roover

Includes preparation course: Yes / ~~No~~ (Select the correct option)

Context

The web browser is becoming the de facto platform for many internet applications, allowing applications to become more and more interactive and advanced. These applications have a rich client, which means that more and more work is done by the browser (client). Because of this, the difference between the client and the server becomes less clear.

The classic web technologies follow the three tier model where an internet application is divided into the client, server and database tier. As a consequence, each tier is implemented in a different technology, typically PHP for the server, JavaScript for the client and relational databases for the data tier. The fact that applications must be composed of these different technologies leads to a so-called *impedance mismatch*: the technologies don't integrate seamlessly, and it is up to the programmer to manually solve the issues .

As a reaction to this problem several tierless or single tier programming languages are developed. These allow the programmer to write multiple tiers of an internet application in one single programming language. Examples of tierless languages are Hop, Links, Opa, etc. Each of these languages escapes from the tyranny of the client-server composition in a different way.

Instead of developing another tierless language, we allow **tierless programs** to be written in **JavaScript** itself. Therefore we developed a tool, called Stip.js, that transforms a tierless JavaScript program into a distributed web application.

Proposal bachelor thesis

The bachelor thesis consists of the development of an IDE for tierless JavaScript code. The programmer uses a graph-based representation of a program (where each node is an expression of the program) to indicate whether certain expressions belong to the server and/or the client tier.

The IDE indicates by using e.g. colors to which tier expressions belong. It should of course be possible to change this tier information for every expression.

The Stip.js tool takes this graph as an input and makes the program it represents distributed.

On top of this, the IDE is capable of giving feedback to the user (e.g. moving this expression to the client tier invalidates previous moves, or calling this function will result in a remote procedure call to the other tier, etc.).

Preparatory course bachelor thesis

The preparatory course consists of making a literature study of program slicing and program dependency graphs. These graphs are crucial for this project, because they are a graph representation of the data and control flow of a program. Slicing algorithms always operate on a program dependency graph.

In order to develop this IDE, you need some insight in the techniques used by the Stip.js tool. More concretely, surveys about these topics will be provided such that a literature study can be performed.