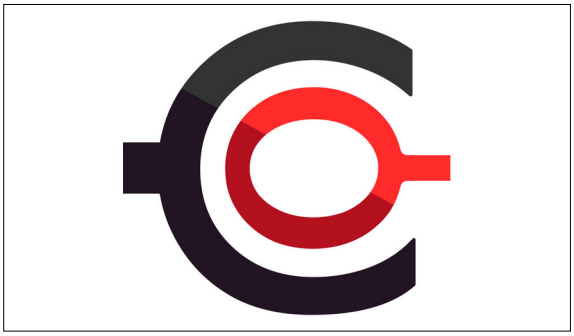
Hansruedi Patzen

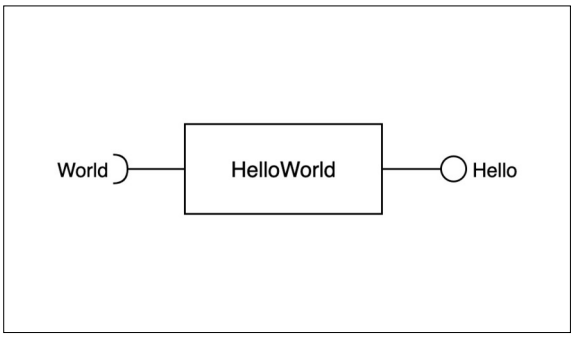
Graduate Candidate	Hansruedi Patzen
Examiner	Prof. Dr. Luc Bläser
Co-Examiner	Dr. Felix Friedrich, ETH Zürich, Zürich ETH-Zentrum, ZH
Subject Area	Software and Systems

Composita 2.0

A Modern Reinterpretation of the Language



The newly designed Composita icon
Own presentment

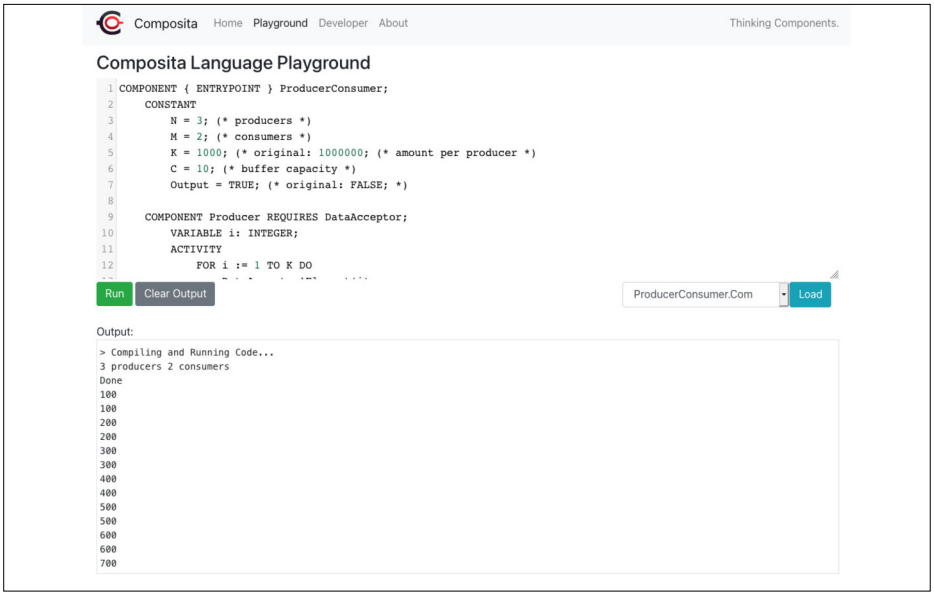


A component named "HelloWorld", offering the "Hello" and requiringthe "World" interface
Own presentment

Introduction: The Composita language and runtime system have been created by Luc Bläser during his doctorate at the ETH Zürich. It features components and interfaces as its main building blocks. Interfaces define the communication protocol by declaring an order, in which messages have to be exchanged. Each component is a self-contained unit, only able to communicate with others by sending messages to one of the targets component's interfaces. Each component may offer and require multiple interfaces. For each offered interface it either has to provide an implementation itself, or redirect it to an internal component's matching interface.

Objective: The goal of this thesis is to take the original Composita programming language and make it available to a broader audience. This is achieved by two primary means: Firstly, the current dependence on a virtual machine or x86 system is to be replaced with a new compiler and a runtime that can be deployed as a serverless web-application. This means nothing more than a modern web browser is required to write code in Composita. Secondly, changes to the Composita language are proposed to make it more appealing to developers coming from current popular languages like TypeScript, Kotlin or Swift.

Result: A Composita playground is publicly available online at https://www.composita.dev/, which is built using the React web framework and uses CodeMirror for its editor capabilities. The required Composita compiler, runtime and intermediate language representation have been designed and built using the TypeScript language. They are available as node packages in the npm registry. With some exceptions, most of the original language features have been implemented and can be tried out. Several ideas have been proposed but not yet implemented. These are syntactic and semantic changes, that would make it look and feel like a modern language.



Composita playground available under online at https://www.composita.dev
Own presentment