

Static typing of complex presence constraints in interfaces (Artifact)

Nathalie Oostvogels¹

Vrije Universiteit Brussel, Brussels, Belgium
noostvog@vub.ac.be

Joeri De Koster

Vrije Universiteit Brussel, Brussels, Belgium
jdekoste@vub.ac.be

Wolfgang De Meuter

Vrije Universiteit Brussel, Brussels, Belgium
wdmeuter@vub.ac.be

Abstract

This artifact is based on TypeScriptIPC, a statically typed programming language with interfaces in which complex presence constraints can be defined. This enables developers to express inter-property constraints on interface properties. The need for these inter-property constraints stems from web APIs, which often impose a complex “dependency logic” between properties. For example, some prop-

erties may be mutually exclusive, or the presence of a property may depend on the presence of others, etc. TypeScriptIPC is a variant of TypeScript, in which interfaces are extended to express constraints over multiple properties, using propositional logic. This artifact contains documentation on how to build and run TypeScriptIPC, such that the code snippets from the paper can be run.

2012 ACM Subject Classification D.3.3 Language Constructs and Features, F.3.2 Semantics of Programming Languages, F.3.3 Studies of Program Constructs

Keywords and phrases type system, interfaces, dependency logic

Digital Object Identifier 10.4230/DARTS.4.3.3

Related Article Nathalie Oostvogels, Joeri De Koster, Wolfgang De Meuter, “Static typing of complex presence constraints in interfaces”, in Proceedings of the 32nd European Conference on Object-Oriented Programming (ECOOP 2018), LIPIcs, Vol. 109, pp. 14:1–14:27, 2018.

<http://dx.doi.org/10.4230/LIPIcs.ECOOP.2018.14>

Related Conference 32nd European Conference on Object-Oriented Programming (ECOOP 2018), July 19–21, 2018, Amsterdam, Netherlands

1 Scope

This artifact provides the necessary materials to run the code snippets from the accompanying paper. It contains an implementation of the programming language TypeScriptIPC, which was presented in “Static typing of complex presence constraints in interfaces”. While the paper contains a programming language that is a *sound subset* of TypeScript, the artifact extends TypeScript in its entirety. Soundness remains guaranteed as long as only language constructs from the formalisations in the paper are used.

2 Content

The artifact package includes:

¹ Funded by a PhD Fellowship of the Research Foundation - Flanders (FWO)



3:2 Static typing of complex presence constraints in interfaces (Artifact)

- 10 ■ a VirtualBox image that contains an installed version of TypeScriptIPC;
- 11 ■ a PDF (`Instructions.pdf`) that contains:
 - 12 ■ detailed instructions for running TypeScriptIPC programs, as well as shortcuts for running
 - 13 the code snippets found in the tutorial, examples from the paper and tests from the provided
 - 14 test suite;
 - 15 ■ a tutorial about programming with inter-property constraints;
 - 16 ■ a list of differences between the formalisation and the implementation;
 - 17 ■ a mapping of formalisation rules from the paper to procedures in which those rules are
 - 18 implemented.

19 **3 Getting the artifact**

20 The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the
21 Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also available
22 at: <https://github.com/noostvog/typescriptipc> (aec branch). The instructions accompa-
23 nying the artifact are also available at [http://soft.vub.ac.be/~noostvog/typescriptipc/](http://soft.vub.ac.be/~noostvog/typescriptipc/Instructions.pdf)
24 `Instructions.pdf`.

25 **4 Tested platforms**

26 The artifact can be installed on any platform running `node.js` and `npm` (package manager for
27 `node.js`: at least version 5.7.1, to support `npm ci`). The provided VirtualBox image (`.ova`)
28 requires around 1GB of free RAM to run TypeScriptIPC, and 4 GB of free RAM to built
29 TypeScriptIPC as well.

30 **5 License**

31 The artifact is available under the MIT license.

32 **6 MD5 sum of the artifact**

33 b47a983b7b1fa9fd070c36863bf83435

34 **7 Size of the artifact**

35 3.76 GiB