The Journal of Systems and Software 230 (2025) 112565

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Binding of C++ and JavaScript through automated glue code generation ☆

Yijie Ou, Chenghao Su, Lin Chen *, Yanhui Li, Yuming Zhou

State Key Laboratory for Novel Software Technology, Nanjing University, No. 163 Xianlin Avenue, Nanjing, 210023, Jiangsu, China

ARTICLE INFO

Keywords:
Multilingual system
Glue code
Code generation
Rewrite system

ABSTRACT

Modern software systems frequently employ multiple programming languages to harness unique strengths of each language. One such language is JavaScript, which enjoys widespread usage, particularly in client-side web development. Nonetheless, JavaScript lacks native support for low-level operations. To address this limitation, developers often repurpose pre-existing C/C++ modules, but they need to write Glue Code to facilitate the adaptation of C/C++ modules to the JavaScript environment. Reusing C/C++ modules can significantly enhance software performance and usability, making it highly sought-after in practical applications.

In response to this demand, we present an innovative approach to generating C-to-JavaScript glue code, grounded in a rewrite system. Initially, we define the glue code generation problem with a formal multi-language system that captures interoperation semantics of JavaScript and C++. Subsequently, we abstract a framework for generating glue code and utilize a parameterized template-based rewrite system to implement this framework. Lastly, we implement our method through a tool named CJSBinder and conduct a systematic evaluation, analyzing its accuracy, efficiency, and practical applicability compared to a state-of-the-art tool. The results demonstrate that CJSBinder excels at efficiently generating accurate glue code and is compatible with a variety of real-world project contexts.

1. Introduction

In the development of modern software systems, it is becoming increasingly common to use multiple programming languages instead of relying solely on a single language (Hu et al., 2023; Mayer et al., 2017; Buro and Mastroeni, 2019). This trend is driven by the need to take advantage of the strengths and capabilities of different languages for specific tasks (Abidi et al., 2021). Numerous prominent large-scale software systems have adhered to this principle, exemplified by the simultaneous utilization of JavaScript and C/C++ in Node.js and Chrome.

JavaScript is a high-level, dynamically typed language that is widely used. While JavaScript provides a lot of functionalities for client-side web development, it does not have built-in support for many low-level operations such as I/O or system-level programming (Brown et al., 2017). Developers often turn to other languages, such as C/C++, which better support these low-level operations. By offloading computationally-intensive tasks such as image processing or cryptography to C/C++ modules, JavaScript code can focus on higher-level application logic, resulting in better performance and scalability.

Foreign Function Interface (FFI) is a common way to integrate JavaScript with C/C++, allowing JavaScript to call C/C++ modules. FFI is made possible by the APIs exposed by the underlying JavaScript engine, which are mostly written in C/C++ and bridge the JavaScript and the C/C++ code. To integrate JavaScript with C/C++ and reuse a C/C++ module, developers can choose a widely-used method — writing code that wraps the C/C++ module to adapt to the JavaScript environment and registers the module in the context of JavaScript. Such code is called *C-to-JavaScript Glue Code*. Generally, *Glue Code* is a code snippet that bridges two related components. With glue code, developers are capable of reusing C/C++ modules in JavaScript to convert values, assist memory management, propagate exceptions, etc.

Writing glue code to combine JavaScript with C/C++ is not easy, as it requires a comprehensive understanding of interoperation between JavaScript and C/C++ and skill in using interoperation APIs. Manually writing glue code is complicated and may cause bugs in many ways (Brown et al., 2017). Code generation tools may help ease the burden of writing C-to-JavaScript glue code. However, existing works may not be able to generate contextual glue code due to inflexibility

☆ Editor: Martin Pinzger.

* Corresponding author.
E-mail addresses: dz21330021@smail.nju.edu.cn (Y. Ou), chenghao_su@smail.nju.edu.cn (C. Su), lchen@nju.edu.cn (L. Chen), yanhuili@nju.edu.cn (Y. Li), zhouyuming@nju.edu.cn (Y. Zhou).
URLs: https://cs.nju.edu.cn/chenlin/index.htm (L. Chen), https://yanhuilinju.github.io/ (Y. Li), https://cs.nju.edu.cn/zhouyuming/index.htm (Y. Zhou).

https://doi.org/10.1016/j.jss.2025.112565
Received 19 October 2024; Received in revised form 27 June 2025; Accepted 6 July 2025
Available online 19 July 2025