Abstract

This research proposes a Dynamic Isomorphic IoT System Architecture to reduce the complexity of development in IoT systems, which have traditionally been considered to have a heterogeneous configuration. In conventional IoT system development, different technologies and programming languages are often used in the device layer, edge layer, and cloud layer, leading to decreased development efficiency and interoperability issues due to this heterogeneity. Moreover, the complexity of coordination and data flow between layers increases, adding to the burden on developers.

To address these challenges, this research makes the following three proposals. First, we propose and implement a data flow platform that enables integrated design and development of the entire IoT system using a single programming language. Specifically, by leveraging the Elixir language and its ecosystem, we construct a consistent development environment from the device layer to the cloud layer. This eliminates technical gaps between layers and reduces the complexity of IoT system development. Second, we propose and implement a method that allows developers to dynamically apply code changes to applications within IoT devices. Traditionally, code changes to devices required firmware updates and reboots, delaying the development cycle. The proposed method allows immediate application of code changes to running devices, enabling a rapid development cycle and improving development efficiency. Third, we propose and implement a dynamic and isomorphic IoT system architecture utilizing WebAssembly (Wasm). Development using a single language may impose constraints on incorporating new technologies, and introducing dynamic update methods without guidelines may cause confusion in system design. The proposed method uses Wasm, which has portable characteristics, to easily incorporate necessary functionalities, using common Wasm binaries across layers and making those parts dynamically updatable, thereby balancing integration and flexibility.

To verify the effectiveness of these proposed methods, we conducted implementations and performance evaluations based on actual use cases. In the first proposal, we developed a data flow platform named Pratipad and demonstrated that integrated design and implementation of data flows from the device layer to the cloud layer using the Elixir language is feasible. This enabled us to construct realistic IoT systems comprising multiple layers and provided a technical foundation that allows data flows to be described in an easily comprehensible manner, demonstrating its effectiveness on a realistic scale. In the second proposal, we proposed and implemented a method for dynamically applying code changes to IoT devices and experimentally showed that the time required for updates can be significantly reduced compared to existing firmware update methods. This accelerates the development cycle and reduces the developers' burden. In the third proposal, we evaluated the utility of an isomorphic IoT system using Wasm in use cases involving image recognition and machine learning models. By using common Wasm binaries across layers, we ensured functional consistency and reusability while enabling dynamic feature updates.

Overall, the proposed methods in this research address the challenges of heterogeneity in IoT system development, achieving both improved development efficiency and system flexibility, thereby reducing the complexity of IoT system development. This enables rapid development and deployment of IoT systems and allows for adaptation to future technological innovations, significantly contributing to the advancement of the IoT field.

Keywords: IoT system development; Dynamic isomorphic IoT systems; Elixir; Nerves; WebAssembly