Proposal to Eliminate the Complexity of Design and Implementation in The Bidirectional Dataflow of IoT Systems

2030006 Kentaro Kuribayashi

In this paper, we propose a method for describing the data flow and processing of bi-directional and diverse data flow patterns in IoT systems using a single language and communication protocol in a comprehensive manner, with the aim of reducing the complexity in the design and implementation of IoT systems.

IoT systems, which bridge devices such as sensors and actuators in physical space with computational processes in cyberspace, have been applied in various domains. Such IoT systems are designed and implemented based on an architectural model consisting of three layers: device layer, edge layer, and cloud layer. The three-layer structure allows data to be given meaning in the middle layer, ensures security by avoiding direct access by the device layer, and can take advanteges of edge computing.

On the other hand, the adoption of the three-layer architectural model poses a challenge in terms of structural complexity in system design and implementation. This is due to three factors: (1) the variety of programming languages and communication protocol options, (2) the variety of data acquisition methods and bidirectional data flow, and (3) the poor visibility of data flow throughout the IoT system. While related research can solve each of the challenges individually, this research aims to solve all the challenges mentioned above.

The proposal that addresses the first issue is a method that can design and implement the three layers in an integrated manner using the same programming language and communication protocol. We propose a method to design and implement the three layers in an integrated manner using the Elixir programming language as the core. In the proposed method, we use Elixir as a programming language and Erlang/OTP distributed network protocol as a communication protocol. In addition, we use a secure communication protocol and an authentication method for security measures among the three layers.

Our second proposal is an infrastructure that supports push, pull, and demand methods and can be used in different ways. We propose a method to solve the second problem by using Pratipad, a library developed by the author, on the distributed Erlang/OTP network infrastructure. In the proposed method, the user can specify which data acquisition method is used to acquire data from a device. Then, by defining the type of message for each mode, the proposed method can handle any data acquisition method while using the same infrastructure.

As the solution for the third problem, we propose a notation that can declaratively describe the data flow of an IoT system consisting of three layers, and then separate the data flow from the processing. The proposed notation can describe and execute the data flow as Elixir code, including the types of modes and bidirectionality described above, and the data processing in the edge layer. By using these notations, we can grasp the whole data flow under one view.

In order to evaluate the proposed method, for each of the proposals, we evaluate (1) that the proposed method can be used to design and implement a three-layer IoT system in an integrated manner, (2) that the proposed method can easily handle any of the data acquisition methods, and (3) that the proposed notation can sufficiently represent the entire data flow. As a result, we believe that the proposed method, which simultaneously solves all of the problems that related research has attempted to solve individually, has achieved an academic contribution in that it solves the problems of IoT systems and shows a method that can be designed and implemented more effectively with concrete implementation.