

発表概要

ライブプログラミングのためのデータ構造の可視化と対話機能

岡 明央^{1,a)} 増原 英彦^{2,b)} 青谷 知幸^{2,c)}

2017年3月4日発表

ライブプログラミング環境はプログラムを編集すると即座にプログラムを実行し、その結果を出力する。これまで、画像や音を出力するプログラムに対して特に有効だと考えられていた。一方、データ構造の定義や操作を行う際、プログラマはオブジェクト間の参照関係を思い浮かべながらプログラムを書くことが多い。そこで我々は、データ構造のためのライブプログラミング環境 Kanon を提案する。具体的には、プログラムが編集されるたびに、プログラムを実行し、その途中で生成されたオブジェクトおよび参照関係（以下、グラフ構造と呼ぶ）を図表現として可視化する。プログラマによる試行錯誤を容易にするため、以下のような機能を設計・実現した。(1) カーソル位置に連動して実行途中のグラフ構造やその変化を表示する機能、特に複数回実行された命令上にカーソルがある場合は、個別の状態を表示したり、すべての回で起きた変化をいっせいに表示したりする機能、(2) 図表現上のオブジェクトや参照関係をクリックすることで、それらが作られたときのグラフ構造を表示し、コード位置を教えてくれる機能、(3) プログラムの編集によってグラフ構造が変化した際にも、図表現の概形を保つ機能である。JavaScript 言語を対象に、Kanon の処理系を Khan Academy の live-editor を拡張して作成した。

Visualization and Interactive Features for Supporting Data Structures in Live Programming

AKIOU OKA^{1,a)} HIDEHIKO MASUHARA^{2,b)} TOMOYUKI AOTANI^{2,c)}

Presented: March 4, 2017

Live programming environments are tools that, when a piece of a program is edited, instantly execute the program and display its output. They have been recognized as useful to programs that output images and sounds. When the programmers define data structures and their operations, they often write program with references among objects in their mind. We propose a live programming environment, called Kanon, for assisting data structure programming. When a program is edited, it executes the program, collects created object and references among them (hereafter referred to as the graph structure), and displays the graph structure as a node-link diagram. In order to make trial-and-error processes by the programmers easier, we designed and implemented the following mechanisms: (1) a mechanism that displays an intermediate state and changes of the graph structure during the execution in synchronous with the cursor position, including the features for displaying one particular execution at the cursor position or all executions at once, (2) a mechanism, when a node or an edge of the node-link diagram is clicked, that displays the graph structure at the time of creation of the corresponding object or reference and navigates the cursor to the position of the respective code, (3) a mechanism that mostly preserves the shape of the graphical representation even if an edit of the program changes the graph structure. We constructed an implementation of Kanon for JavaScript by extending the Khan Academy's live-editor.

¹ 東京工業大学理学部情報科学科
Department of Information Science, Tokyo Institute of Technology, Meguro, Tokyo 152-8552, Japan

² 東京工業大学情報理工学院数理・計算科学系
Department of Mathematical and Computing Science, Tokyo Institute of Technology, Meguro, Tokyo 152-8552, Japan

a) oka.a.ab@m.titech.ac.jp
b) masuhara@acm.org
c) aotani@is.titech.ac.jp