

要旨

プログラムの構造を短く指定することによるコード補完手法 の提案

米田 裕司

ソフトウェア開発の生産性を向上させる方法の一つとして、コーディングを効率化するためのコード補完がある。コード補完は、コードの不足部分を補完する技術である。この技術はコードの入力やライブラリ参照の労力を減らすだけでなく、構文エラーの元となるミスタイプを防ぐことができる。しかし、統合開発環境やテキストエディタに組み込まれているコード補完機能は主にトークン単位の単純なものを対象とし、コードの構造を補完することができないという問題がある。

一方、HTML/CSS を編集するエディタの一部には ZenCoding という省略した記法でコードを短く指定して展開するプラグインがある。

そこで本研究では、ZenCoding のコードを短く指定して展開する機能をプログラムに応用し、プログラムに対しても構造を補完するシステムを構築した。既存のコード補完機能を備えたテキストエディタと、そのテキストエディタに提案手法を組み込んだものそれぞれの環境で、ソースコードを入力し打鍵数を比較した。その結果、提案手法を組み込むことで大幅に打鍵数を削減できることが分かった。

キーワード コード補完, 統合開発環境, 構文解析

Abstract

A code completion method by briefly specifying the structure of a program

Yuji YONEDA

Code completion improves efficiency of coding and productivity of software development. Code completion is a technique to complement a part of code based on already typed code. The technique reduces the efforts of typing codes and searching for library functions. Moreover, it prevents mistyping that causes syntax errors. However, the function of code completion in integrated development environments (IDEs) and text editors mainly targets at the completion of tokens and cannot complement the structure of the code.

In contrast, some HTML/CSS editors have plug-ins that can generate a part of code by briefly specifying the structure of the code in the notation called ZenCoding.

In this thesis, the function of ZenCoding for briefly specifying and generating the structure is applied to programs, and a system to complement the structure of a program is developed. We compare the number of keying for inputting source codes on a text editor with proposed code completion to that on a text editor with existing code completion. As a result, the number of keying is reduced by embedding the proposed method.

key words code completion, integrated development environment, parsing