

テキスト・プログラミング混在型学習環境の開発と実践

Development and Application of mixed textbook and programming learning environment

坂田 圭司^{*1}

Keiji SAKATA^{*1}

^{*1}東海大学情報教育センター

^{*1}ICT Education Center, Tokai University

Email: kgsakata@tsc.u-tokai.ac.jp

あらまし：本研究では、Web ベースのプログラミング学習環境を開発して、学校内での柔軟な開発環境構築と運用を可能し、さらに授業において学習内容に注力しやすいシステムの実現を目的としている。本システムではテキスト内に埋め込まれたテキストエディタとコンパイル・実行環境を用いて、シームレスな学習を行うことができる。テキストの記述には Markdown を採用して、授業中での編集を簡便に行えるため、臨場感のある授業展開が可能になる。

キーワード：プログラミング学習、開発環境、電子テキスト、Markdown

1. はじめに

プログラミング基礎レベルの学習において、複雑なソフトウェア開発環境の操作に慣れるために費やす労力と時間は、アルゴリズム習得等のプログラミング学習内容への注力を低下させる一因になっている。

本研究では、授業での活用に適したソフトウェア開発環境を作成して、授業での限られた時間内で、学習者が本来のプログラミングに注力できる環境を提供することを第一の目標とした。

第二の目標として、電子テキストとプログラミング環境の混在できるようにして、解説と実習をシームレスに繋ぐ環境を目指した。オンライン文書中にソースコード編集および実行環境を埋め込み、この文書記述に Markdown⁽¹⁾ を採用することで、電子テキストの編集を容易にして、さらに授業中におけるライブコーディングに対応できるようにした。システムの基本設計については、第41回教育システム情報学会全国大会で発表を行った⁽²⁾。今回はシステム実装の詳細と実践事例について報告する。

2. プログラミング環境の検討

大学のプログラミング授業での開発環境では、統合開発環境 (IDE)、またはテキストエディタとコマンドライン環境 (CLI) が多く利用されている。これらの環境は、プログラミング初学者にとって、表 I に示す問題点を生じやすい。

表 1 従来の IDE 利用時の主な欠点

機能	欠点
プロジェクト管理	基礎学習には不要な手順
コード編集	過剰な入力サポート
ビルド・実行環境	複雑なデバッグ環境
ファイル管理	保存場所の指定ミス

IDE の操作を習熟することは、研究や業務においては必須であるが、基礎レベルの学習では、アルゴリズムの理解や言語記述へ注力したい場合に、阻害要因となりうる。

本研究では、上記の問題点を解決して、授業利用に適したプログラミング環境を構築するために、次の機能を持たせることにした。

(1) Web サーバでのシステム運用

学内で授業用システムを新規に用意するには多くの制約を伴う。本システムは技術面での対策としてオンプレミス、クラウド両対応させるために、Web サーバを用いたシステムの管理を採用した。

また、Git (分散型バージョン管理システム)⁽³⁾ を用いて、マルチユーザでのソースコード保管・履歴管理を可能にした。

(2) Web ベースのクライアント環境

学習者が用いるクライアント環境も Web ベースにして、各端末へのインストールを不要にした。公開サーバの場合は、学習者が自宅からでも利用可能なため、反転学習にも使える。

また、ソースコード編集は、Atom エディタをベースにしたインライン環境にすることで、編集しやすい環境を整えた。

(3) 幅広いプログラミング言語への対応

ソースコード編集環境と、ビルド・実行環境を独立させて Web サーバ上でプログラム処理を担当させることで、サーバ上に入れたプログラミング言語を学習者のクライアント環境で利用可能にした。

これらの機能によって授業担当者は、Web サーバ上にシステムを設置して、学習者に対して URL を伝えてユーザー登録させるだけで利用できるようになる。

3. 電子テキストの記述とソースコード提示

プログラミング授業において、従来はテキストを紙または電子的に提供して、それを見ながら解説を聞いてノートを取り、開発環境で実習を進めるスタイルがとられてきた (図1の(a)).

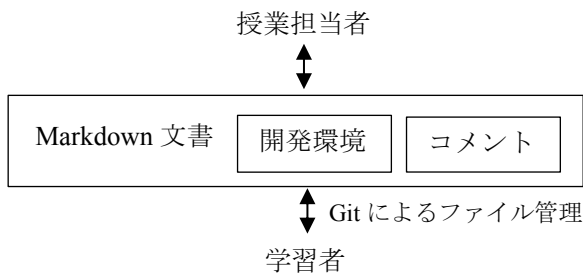
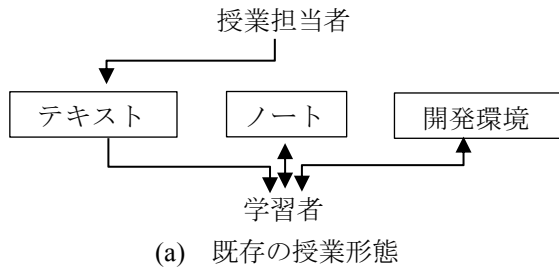


図1 授業形態

従来のスタイルでは、実際に作成したプログラムと、テキストおよびノートの内容は離れて存在しており、学習者と授業担当者のやりとりは、LMSなど別の手段を必要とする。

本システムでは、電子テキストの記述に、軽量マークアップ言語 Markdown を用いることで、HTMLよりも簡単な文法で書ける上に、開発環境の画面を統合しているので、解説と実習を密な関係で結んで学習を進めることが可能になる (図1の(b)).

また準リアルタイムでレンダリングした文書を表示可能になるので、授業中の内容変更にも対応できる (図2および図3).

```

- 文字は、英数字モードで入力して下さい。
- 入力完了したら、*保存*、*コンパイル*、*実行*の順番で作業を進めましょう。

```cpp
#include <iostream>
using namespace std;

int main()
{
 cout << "Hello!" << endl;

 return 0;
}
```

```

図2 Markdown ソースコード

```

• 文字は、英数字モードで入力して下さい。
• 入力完了したら、保存、コンパイル、実行の順番で作業を進めましょう。

#include <iostream>
using namespace std;

int main()
{
    cout << "Hello!" << endl;

    return 0;
}

```

図2 Markdown レンダリング出力

さらにコメント機能により、教材単位での教員と学習者の双方向性を持たせている。

4. 期待される効果

授業担当者の立場から見ると、Markdown によるシンプルな文書記述を用いることで、教材の共有と変更がしやすくなる。さらに準リアルタイムでのレンダリングが可能なることから、解説しながら説明文やソースコードサンプルの追加を行い、ライブ感のある授業進行ができる。

学習者の立場から見ると、テキストとソースコードが一体化していて、その中でビルドと実行できるので、解説と実習のつながりを実感しやすい。また、Markdown の文法は簡単なので、自ら文書を改編してノートとしての利用も可能である。

両者共通の利点としては、授業毎に Web サイトを用意した場合、ファイル管理が容易で、実習結果やコメントの共有がしやすい点がある。

これらの特徴から、本システムをプログラミング基礎レベルの授業で利用することで、アルゴリズムやソースコードの理解に注力した学習を期待できる。

参考文献

- (1) Official Markdown project at Daring Fireball:
<http://daringfireball.net/projects/markdown/> (2016)
- (2) 坂田圭司:" 授業用プログラミング環境の開発",第 41 回教育システム情報学会全国大会 (2016)
- (3) git website
<https://git-scm.com>