

理解度モデルに基づく C言語学習支援システムの設計と実装および試用

小芦 勇介¹ 玉木 久夫¹

概要: 学習者の理解度モデルに基づいて適応的に練習問題の推薦を行う Web 上の C 言語学習支援システム MILES(Model-based Interactive Learning Support) を設計し、実装した。このシステムは学習者達がそれぞれに合ったペースで学習を進めることができるようになることを目指している。本稿では、MILES システムの概要と実装、授業での運用に向けて行った負荷試験、および授業での試用について述べる。

1. はじめに

今日の情報科学の発展により、ソフトウェア開発の機会は益々増え続けている。その中でプログラミングの能力を身につけることは不可欠なまでになり、大学等で盛んにプログラミングの授業が行われている。授業では、学習するプログラミング言語として C 言語が採用されることが多々ある。発表者らの所属する学科をはじめとする典型的な授業においては、C 言語を学ぶための教材となる問題を教師が用意し、問題を順序だててコースを組む。学生はコースに則って問題に取り組み、質問があれば巡回している教員やティーチングアシスタントに聞くことができる。学生が躓く箇所は千差万別だが、中には何故躓いているのかという原因が学生自身も分かっていない深刻なケースも少なからずある。教員側は、学生に色々な質問を投げかけることで原因を追究し理解を促して解決するが、負担はかなり大きいものとなる。

そこで、システムが学習者の学習状態を細かに把握し適切な問題を推薦することで、学習者が自身のペースに見合った学習が出来るような環境が望まれる。本研究の MILES(Model-based Interactive Learning Support) システムには以下のような特徴があり、前述した課題点の克服を目指す。

- (1) 学習者の C 言語に対する理解度をモデル化している。
- (2) 学習者に合っていると思われる問題を、モデルと、問題に設定されたパラメータに基づいて数題推薦する機能を持つ。

この特徴を実現するための原理を考案し、大学の授業の

場で運用することを前提に実装した。新しいシステムが授業中の負荷に耐えられるかが不鮮明であったため、既存のツールを用いて負荷試験を実施した。負荷試験の結果より、試用可能という判断が下りたので実際の授業の場で試用を開始し、MILES システムを現場において評価する準備が出来たことを確認した。

以下、2 章では、この MILES システムの原理の説明をする。続く 3 章において、MILES がどのような構成を取り、どのような機能が実装されているかについて述べる。次いで 4 章において、授業での試用に向けた負荷試験の方法や結果について述べる。5 章において試用試験の結果について述べ、今後の課題と関連研究を取り上げて終わりとする。

2. MILES の動作原理

MILES の特徴である (1) と (2) がどのような原理に基づいているかを述べる。この章の内容は共同研究者の山田の修士論文 [1] に詳述されている。

2.1 C 言語の理解度のモデル化

モデル化を考える過程で、先行研究の中に C 言語に対する知識を体系的に表現したものは見当たらなかった。そこで、MILES は独自のアイデアに基づいてモデル化することにした。

C 言語における知識の単位として考えられるものを成績要素とし、これを主観的に抽出した。学習者らは各成績要素に対する理解の深さの度合いを持っており、それを総合することで学習状況を把握することができる考えた。表 1 に抽出した成績要素 157 個のうちいくつかを例として示す。既に述べた通り、この成績要素は授業に合わせて抽出

¹ 明治大学
Meiji University, Tama, Kanagawa 214-8571, Japan

表 1 理解度モデルを構成する成績要素の例

for 文	while 文	繰り返し
+演算子	代入演算子	複合代入演算子
int 型	ポインタ	構造体
仮引数	実引数	関数
EOF 定数	マクロ	コメント
printf 関数	scanf 関数	配列
分岐	?:演算子	switch
入れ子	if 文	””文字列リテラル

されたものであり、一般の C 言語における全ての学習項目を抽出したのではない。

if 文や for 文といった基礎的な文法から、繰り返しや関数といった抽象的な概念が成績要素として抽出されている。ここで、表 1 内の「for 文」「while 文」「繰り返し」を見るように、互いに理解度に関連がある関係が存在する。一般的に何かを学習する際を考えても、何かの学習項目の理解の深さが他の学習項目の理解の深さに影響を与えることは自然なことだと言えるだろう。MILES は、理解度のモデルを独立した成績要素の集まりとして捉えるのではなく、成績要素同士を包含関係によって結ぶことで階層的なグラフとして捉える。これは、具体的な成績要素の理解度がそれより抽象的な成績要素の理解度に影響を与えていると考えたからである。先に挙げた 3 つの成績要素で言うならば、「繰り返し」を具体化した成績要素が「for 文」と「while 文」であり、逆に「for 文」と「while 文」の理解の深さが「繰り返し」の理解度に与していると言える。更に、ある成績要素がいくつかの具体的な成績要素と関係を持っている時、それぞれからの依存度の高さが存在すると考えた。例えば、「繰り返し」を構成する「for 文」「while 文」は同程度の影響を「繰り返し」に与えると考えられる。一方で、「分岐」は「if 文」「?:演算子」「入れ子」「switch」から構成されるとする。これらが 4 つの成績要素が「分岐」に与える影響は同じではないと考えられ、「if 文」が最も影響が大きく、その他は「if 文」よりは影響しないだろうという関係にあると考えられる。まとめると、成績要素をノード、抽象度の高い成績要素からそれを構成する具体的な成績要素への向きを持つ有向辺からなるグラフが、MILES での C 言語の理解度モデルとなる。ノードには 0~10000 の範囲の整数値のスコアが付き、理解度の深さを表す。有向辺には重みが付いており、構成の依存度の比を表す。図 1 に、理解度モデルの一部を抜粋したものを示す。

この理解度モデルを用いれば、具体的な成績要素のいくつかを抽象化してまとめて扱うことができ、学習者へのフィードバックを考えるときの負担が減らせることが期待できる。同時に、ある成績要素の理解度が別の成績要素の理解度に影響を与える仕組みは、学習者の学習状況を精度良く表現できると期待できる。

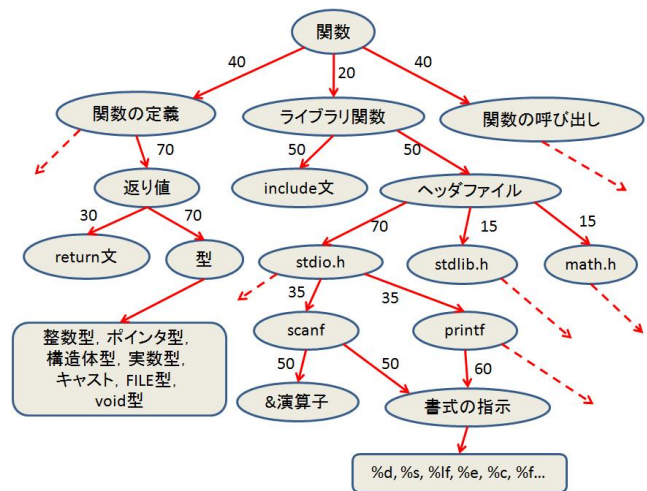


図 1 実際の理解度モデルの一部抜粋 (省略箇所は破線で示した。又簡略化した箇所がある)

2.2 理解度モデルの更新

学習者の学習状態は刻一刻と変化する。理解度モデルはそれに沿うように、モデルのノードの値が更新される必要がある。Web 上のアプリケーションであるという性質上、学習状態が変化すると検知するタイミングは、学習者が問題に解答した時と考えるのが自然である。よって、MILES は解答したタイミングでモデルの更新処理を走らせる。

モデルの更新処理を行う際に、学習者が取り組んだ問題について、それを解くと何についての知識が増えるかやどれほど増えると考えられるのかが分らなければ、理解度モデルを更新することができない。学習者が提出したソースファイルを解析したり、問題文を解析することである程度の情報が得られるかもしれないが大変な手間がかかる。更に、2.3 で述べる推薦機能においても、問題についての情報が必須になってくる。そこで、MILES は各問題に対して問題付与データと呼ばれるパラメータを設定することでこの問題に対処している。この問題付与データは、大きく 2 つの情報を含む。

- Point
- Norma

Point は、その問題を解くことで身に着く知識とその度合いを列挙したものであり、理解度モデルの更新処理の際に参照される。Norma は、この問題に取り組む学習者の想定の学習状態を記述したもので、後述する推薦機能の際の判断に主に利用される。では、問題付与データの実例を問題文と共に示す。

端末から 5 個の数を入力しその総和を出力するプログラムを書いてください。練習のために、変数は 2 つだけ使用するようにしてください。1 つは入力された数を受け取るためのもの、もう 1 つはそれまでに入力された数の和をためておくためのものです。入力は、1 行に 1 つずつ与えられるものとします。出力の書式は

表 2 問題付与データの実例

Point	
成績要素	整数値
printf 関数	5000
scanf 関数	5000
変数	5500
Norma	
成績要素	整数値
printf 関数	2000 - 10000
scanf 関数	2000 - 10000
動作最低知識	3000 - 10000
変数	2000 - 4000

自由としますが、答え以外の整数値は出力しないようにしてください。

MILESはこの問題に対して表2の問題付与データを設定している。

まず、この問題は変数を練習するための問題だと分かるので、Pointには「変数」の成績要素が記述してある。授業で扱っている他の問題と比較した結果、この問題を解くことで5500点程度の「変数」の理解の深さを確認できることが読み取れる。Normaには「変数」について2000~4000点が設定され、学習者が「変数」についてこの程度の学習状態にある時にこの問題に取り組むことを想定している意図が読み取れる。Normaには他の3つの成績要素について2000~満点を設定している。この問題を解くために、下限に設定されたスコアを超える程度の学習が済んでいることを前提にしている意図になっている。この問題は5つの変数を入力によって受け取り、答えとなる整数を出力することが出来る前提知識が求められているため、「printf関数」と「scanf関数」が表2の通りに設定されている。「動作最低知識」は「include文」と「main関数」からなる抽象化された成績要素である。問題付与データにはこのように、問題に込められた出題者の意図を記述することができる。

MILESは、問題に解答した時点の理解度モデル、問題付与データのPoint、解答の正否等の情報を材料に更新処理を行う。しかし、モデルの更新処理アルゴリズムは様々なものが考えられ、その上アルゴリズムの評価が難しく優劣の判断が付かない。このような理由で、利用する更新処理アルゴリズムを1つに決められない。そこで、考案したアルゴリズムをいくつか蓄えておき、容易に切り替えられるようにした。アルゴリズムを使ってみて評価をする、という手順を踏みやすくするためである。実際にいくつかのアルゴリズムが山田[1]によって考案・実装されている。1つ1つを説明することはできないので、そのどれもが採用しているノードのスコアの計算方針を述べるだけに留めることにする。

前述した通り、理解度モデルは抽象度の高さとそれを構成する具象的な成績要素に基づいて階層化されている。抽

象度の高い成績要素のスコアは、それを構成する成績要素のスコアと依存度の高さに基づいて算出することになっている。よって、まず更新が起こるのは階層の最も低い成績要素（それを根とした有向辺が伸びていないもの）のスコアである。問題付与データのPointデータを利用して、これらのスコアを更新した後に、抽象度の低い順に成績要素のスコアが計算される仕組みになっている。山田のアルゴリズムは、階層の底のスコアの計算方法や、一時的に有向辺の重みを書き換える工夫をする等の点において異なる。異なるアルゴリズムの詳細は、[1]を参照されたい。

2.3 学習者への推薦機能

既に述べたが、理解度モデルに基づき、膨大な問題の中から学習者に合った問題を何題か推薦し提示するのもMILESの特徴である。学習状況を反映した推薦問題を常に提示することで、学習者にとって学びやすく効率的な学習環境を目指したいからである。学習者に合っている問題というものかどうか判断するのも色々な基準が存在するため、推薦アルゴリズムが複数考案できる。理解度モデルの更新処理アルゴリズムの時と同様に、推薦アルゴリズムの優劣を付けるための評価が難しく、利用するアルゴリズムをすぐに決められない。やはり同じ理由で、推薦アルゴリズムをいくつか考案しても容易に切り替えられるようにMILESシステムを設計した。

MILESの推薦アルゴリズムは、推薦時の学習者の理解度モデル、問題付与データ等を判断材料にして動作する。尚、問題付与データが設定されていない問題は推薦対象にならない。現在、山田[1]によっていくつかの推薦アルゴリズムが考案・実装されている。どのアルゴリズムも、理解度モデルと全ての問題付与データを付き合わせ、Normaデータと理解度モデルを比較し、一定の基準を満たした問題を推薦する問題リストに追加するようになっている。推薦対象となる条件に、理解度モデルとNormaが完全一致しているかで判定する方法や、理解度モデルとNormaの差の二乗和を取ることで判定する方法等の違いがあり、アルゴリズムの違いは基準の違いと言える。各推薦アルゴリズムの詳細は[1]を参照されたい。

3. MILESの構成と機能

第2章において、MILESがどのような原理で動作するかを述べた。この章ではMILESがSakai[2]のツールとして実装されている等の構成についてや、実装されている機能について述べる。

3.1 Sakai上での連携

Sakaiとは、統合eラーニング環境と呼ばれるeラーニングシステムを構築するための環境である。Sakai上の様々な機能はツールという単位で扱われ、独自にツールを開発

して Sakai に載せることができたり、Sakai 上のツール同士が容易に連携できるという特徴がある。MILES は Sakai のツールとして実装されており、他のツールと連携を取って構成されている。

MILES が連携を取っているツールは藤井 [3], [4] により開発された Wasabi である。Wasabi は問題資源を管理するツールである。Wasabi の特徴は、様々な形態の問題を作成し管理することができ、これを他のツールへ提供することで、異なるツール間で問題を同じように扱うことができることである。Wasabi の提供する問題には解答チェック機能も付いている。MILES が試用環境として計画している授業では、藤井 [3], [4] により開発された MAX/C という Sakai 上のツールが利用されている。MAX/C は Wasabi を用いて問題の管理を行っているため、Wasabi には MAX/C の問題が登録されている。このことから、MILES は Wasabi と連携を取ることで、試用環境と同じ問題を Wasabi を介して利用することができ、更に解答チェックの部分も作りこまなくて済むという大きな恩恵を受けることができる。

3.2 実装されている機能

実装済みの機能は以下の通りである。

- (1) モデル更新処理インタフェースと他ツールへ API 公開
- (2) 推薦機能インタフェースと他ツールへ API 公開
- (3) 各種ログの取得
- (4) 推薦結果に関するアンケート
- (5) 成績要素の管理と階層化
- (6) 問題付与データの管理

以下、各項目について述べる。

(1)、モデルの更新処理のインタフェースを用意することで、更新処理アルゴリズムの切り替えを容易にした。実装したアルゴリズムの中でどれを利用するかを XML ファイルに記述することで、Spring フレームワーク [5] の DI によって切り替える。更新処理を行う機能を MILES は API として他の Sakai ツールへ公開している。

(2)、推薦機能のインタフェースを用意し、推薦アルゴリズムの切り替えを容易にした。仕組みは (1) と同じである。推薦機能も他の Sakai ツールへ公開している。

(3)、稼働中に様々な情報をログとして記録する。MILES の評価のために使ったり、モデル更新アルゴリズムや推薦アルゴリズムにおいて参照されることを想定している。記録されるログは 4 種類ある。本稿では、各ログの詳述は控える。

- 学習者の提出記録
- 推薦結果
- 学習者のページ遷移記録
- 学習者の各問題の正答数、誤答数

(4)、推薦機能が学習者にとって有用かどうかを評価したい時、学習者に直接尋ねることを 1 つの手段と考えて用意

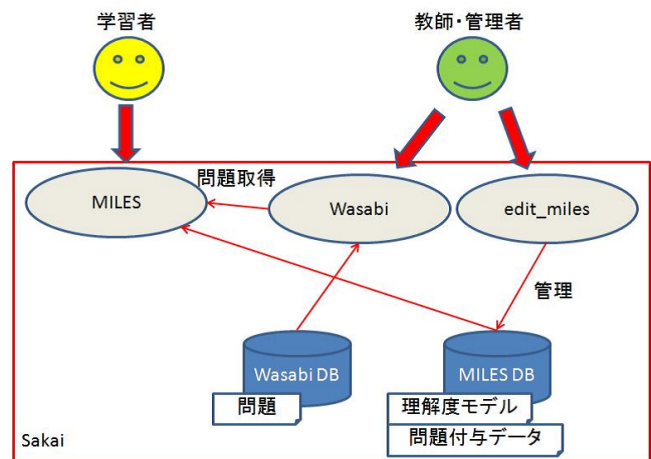


図 2 Sakai 上のシステム構成図

した。アンケートは、推薦機能を使って開いた問題からのみ答えることができる。推薦機能の評価方法についてはまだ手探りの状態である。アンケートの項目は暫定的に決めて、次のことを推薦機能利用者に尋ねる。

- 問題の難しさはどうか
 - 簡単すぎる
 - やや簡単
 - ちょうどよい
 - やや難しい
 - 難しすぎる
- 推薦機能について、当てはまるものを選んでください（複数選択可）
 - 時間があつたので利用してみた
 - 難しい問題に取り組みたくて利用している
 - 易しい問題に取り組みたくて利用している
 - 推薦の精度が悪いと感じている
 - 推薦の精度は悪くないと感じている

- 推薦機能に関する自由記述欄

推薦の精度について直接的な感想を求める内容になっている。

(5)(6)、理解度モデルの成績要素や階層化データを管理、問題付与データの管理を Web ページから行える。実際には、これらの編集機能は MILES から切り離され、edit_miles という Sakai ツールとして実装されている。

3.3 システム構成図

システムの構成図を図 2 に示す。

MILES は Sakai ツールであると既に述べた。Sakai ツールは Java で実装されるため、MILES も Java によって実装されている。edit_miles も同様に Java で実装されている。Sakai は Apache Tomcat 5.5.28[6] 上に配備して起動する。Sakai, MILES, Wasabi ツールはデータベースに MySQL 5.1.46[7] を利用している。授業はサーバ 1 台で運用される。サーバ OS は Solaris 10, CPU は UltraSPARC

T2 Plus(6Core/48Threads)[1.2GHz] × 2, メモリは 32GB である。

学習者は MILES にアクセスして推薦問題を取得し解答するという流れを繰り返して学習を進める。Wasabi, edit_miles の管理ツールへのアクセス権限を持たない。教師側は, Wasabi にアクセスして問題を作成する。edit_miles を用いて, 理解度モデルや問題付与データの編集を行う。Wasabi で新たな問題を作成した場合, 問題付与データを作成しないと推薦対象にならない。

尚, 図 2 の構成は MILES システムを運用する際の最終的な構成である。MILES の試用試験は授業の場を借りて行われる関係で, この構成が少し変えられている。詳しくは次の 4 章で述べる。

4. 試用に向けた負荷試験

4.1 節にて試用試験環境についての説明や試用の目的, 試用に向けて準備したことについて述べる。4.2 節で負荷試験の必要性, 4.3 節で負荷試験の手順と結果の考察について述べる。

なお, 負荷試験において設定したパラメータに誤りがあったことが, 5 章で述べる試用試験再開後に発覚した。4.3 節では, パラメータの誤りを含んだ負荷試験の結果について述べている。試用試験再開の判断は, 4.3 節で述べる負荷試験の結果に基づいて下されている。誤りの内容の説明と, パラメータを正しく設定し直して実施した負荷試験のやり直しの結果については 4.4 節で補足する。

4.1 試用試験環境と準備

MILES システムについての評価を行うために試用試験を行うことにし, その環境について説明する。試用試験の環境として, MAX/C が利用されている筆者の大学の授業を選んだ。授業は 120~130 人程度の学生が受講しており, その大半が C 言語の初学者である。これだけ多くの C 言語初学者が揃っている環境は, MILES にとって理想的である。第一発表者もティーチングアシスタントとして授業に参加するので, 学生達から直接意見が聞けたり, MILES の稼働状況が目に見える点も, この授業を試用環境にするメリットである。しかし, 授業で使うシステムを MAX/C から MILES に切り替えるというわけにはいかない。MILES の提供する学習環境の効果は未評価の状態, 授業で利用できるシステムとはとても言えないからである。そこで, 学生はあくまで MAX/C を用いて演習を行う。MILES は授業中にいつでも利用可能なサイドツールという立場を取り, 学生に利用を強制することはしない。MILES を利用することによって生じるいかなる事象も授業の成績に何ら影響を与えないことを学生に通知する。

試用試験の目的を述べる。基本的には MILES の評価を行うためだが, MILES の提供する学習環境が学生に取って

適切であったかをデータの的に評価するのは難しい。理解度モデルの構築, モデルの更新処理, 推薦アルゴリズム, 問題付与データのパラメータ等, 筆者らの主観に基づいて設定された項目が多く複雑に作用し合っているためである。(山田 [1] のシミュレーションを用いた方法によって, 理解度モデルや推薦の精度の一定の改良が確認されている。) まずは, MILES を利用してもらい, どのような理解度モデルが得られ, どのような推薦がなされるのかを知ることが必要であると考えた。試用中は 3.2 節で挙げた様々な情報を記録するログを取り, 理解度モデルの変化の様子, 推薦結果, 学生のページ遷移の履歴等が後から再現できるようなデータを取っておくことで評価の準備をした。つまり, 学生に MILES に興味を持って利用してもらい, 生のデータを収集することが試用試験の目的である。

試用試験の実施に向けて行った準備は以下の通りである。

- (1) MAX/C で利用される全問題に問題付与データを設定
- (2) Wasabi を用いてオリジナル問題を作成, 問題付与データの設定
- (3) MILES の推薦機能が使えるページを MAX/C へ埋め込み

(1), 授業で取り組まれる問題が MILES の推薦対象になるために問題付与データを設定した。問題の形式は様々あるが, およそ 400 題が推薦対象となった。

(2), 推薦機能の魅力を上げ, 学生の MILES 利用率を上げるための工夫である。前述した通り, 学生は主に MAX/C を使って学習を進める。MILES が推薦する問題が MAX/C 上に登場する問題だけでは, 新鮮味が無く興味が湧きにくいだろう。推薦機能によってのみ登場する問題があれば, その問題目当てに利用率が上がることを期待できる。問題の難易度は, 易しめのものと発展的なものがおよそ半々の割合である。一度に作り切ってしまうのではなく, 授業の進行と共に継続的に問題を追加した。現在, 30 題程度のオリジナル問題がある。

(3), MILES 利用率を上げるための工夫である。MAX/C と MILES は別ツールであるため, 学生が推薦機能を利用するには MILES のページにアクセスしなければならない。すると, 推薦機能を利用するページに辿りつくために何度かマウスクリックをする必要があり, これは推薦機能を使うのを躊躇うほどの負担となる。MAX/C で推薦機能を使うことができるようにページを埋め込み, この負担を軽減させた。3.2 節で述べたように, MILES は推薦機能とモデル更新処理を外部に公開しているため, この埋め込みが実現できる。結局, 試用中の構成は図 2 から図 3 のように変わる。

4.2 負荷試験の必要性

授業の開始回とともに MILES の試用を開始させたがサーバがダウンした。急遽, MILES のログ取り機能だけ

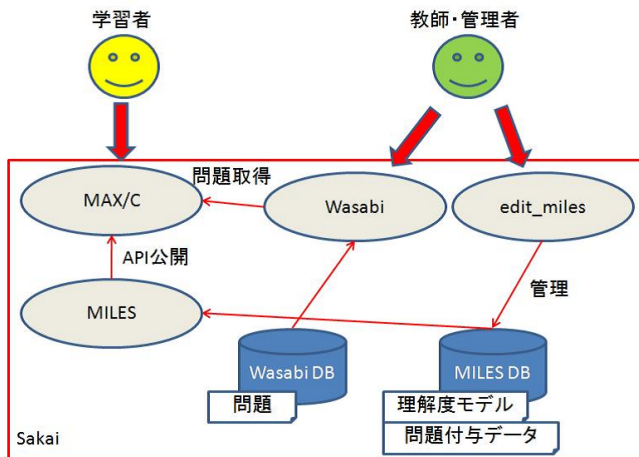


図 3 試用試験におけるシステム構成図

を残し推薦機能をオフにして授業を続行させた。後の調査で、推薦アルゴリズム内のデータベースアクセスの処理が非常に重いことが分かり、その部分を改善した。このことから、ローカル環境ではなく、現場の環境で初めて気づくことができる問題点が存在することを認識し、負荷試験の実施が必須となった。

4.3 負荷試験の実施と結果

負荷試験には、テストを自動化して容易に繰り返し試験できる点からツールを用いることにし、Apache JMeter 2.5.1[8], [9] (以下、JMeter) を使った。

負荷試験の手順は次のように踏んだ。

- (1) 2種類の MILES を用意
- (2) 2種類のテストシナリオを作成
- (3) シナリオを実行・結果・考察

(1), 処理の内容が異なる MILES を用意し、以下 A, B と呼ぶことにする。A は 4.2 節にてデータベースアクセスの改善を行ったもの、B は A に加えドリル形式問題の誤答時にモデル更新処理を行わないようにしたものである。B を用意したのは、ドリル形式問題でかなり大きな負荷がかかっていることを偶然突き止めたからである。ドリル形式問題は、プログラムを書く必要が無く、文章の選択肢を選んだりプログラム片のトレースが求められる基本事項確認のための問題形式である。学生は途中経過を確認するために解答を途中で送信することが多い。これにより学生の解答数が増大し、理解度モデルの更新処理が頻繁に呼ばれるため、負荷が高くなる。更に、途中経過の解答送信に対してモデルの不正解更新処理を走らせるのは、不要に学生の学習状況を低く見積もってしまう恐れがあるため、B を用意した。

(2), サーバに与えるテストのシナリオを作成した。ユーザが同時にドリル形式問題を開き解答を途中送信しつつ最後に完答する、という状態の下で最大負荷相当の負荷をかける。授業中のドリル形式問題の提出数と正答率を 10 分

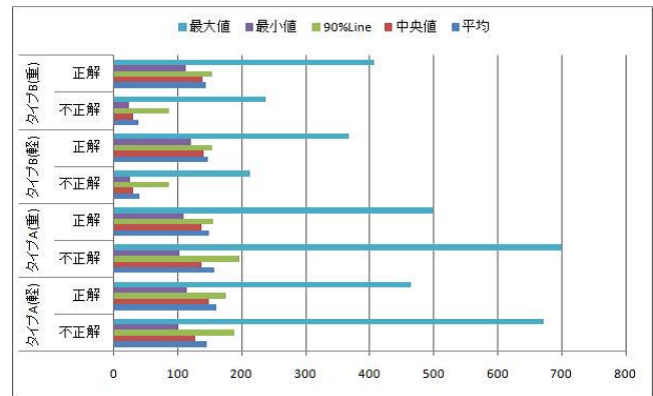


図 4 各タイプの不正解時正解時における応答時間 (ms)

間隔毎に集計した所、最大提出数 1833 件、正答率は 10%~40%程度が多いと分かった。この程度の負荷がかかるように、テストシナリオのリクエスト間のタイマー等のパラメータを調整した。しかし、上述した通りの負荷を完全に再現するようにパラメータを調整するのは難しかったので、最大負荷の近傍のやや軽いテストシナリオとやや重いテストシナリオと 2 種類用意した。

実際には上述の負荷通りのパラメータ調整が出来ていなかったが、それに気付かず負荷試験を進めてしまった。この誤りについては 4.4 節で述べる。

(3), A, B にテストシナリオ 2 種類をそれぞれ適用した。結果にノイズを抑えるため各 3 回ずつ適用し、応答時間について平均を取った (図 4)。

B は全体的に A に比べて応答時間が良かった。最悪応答時間に注目すると、A では 700ms なのに対し、B では 400ms 程度で抑えられている。どちらも 1 秒以内にレスポンスを返すことができていた。90%Line は、応答時間を昇順にソートした時に 90%の位置に来る値を示すもので、A よりも B の方が値が低くなった。不正解の時 A は 200ms 程度なのに対し、B は 100ms もかかっている。B では不正解の時にモデル更新処理を行わないようにしたからである。正解の時 A は 150~180ms 程度、B は 150ms 程度で、若干 B の方が速い。他の、平均値、中央値、最小値の値についても、不正解時の応答は B が著しく速く、正解時は A と B がほとんど同じになった。

図 5, 図 6 の緑の線は重いシナリオを実行した時の A, B のスループット曲線である。

スループット曲線は、単位時間にサーバが処理することができたリクエスト量を示す。曲線はテストが始まってから上がり始めて一定量の処理を続けている。これはテストスレッドが徐々に立ちあがっていき、リクエスト量が増えているためである。サーバの処理の限界を迎えるとスループット曲線は下がるが、A も B もスループット曲線が下がっている様子は見えなかった。

以上より、スループット曲線が下がらない結果から A,

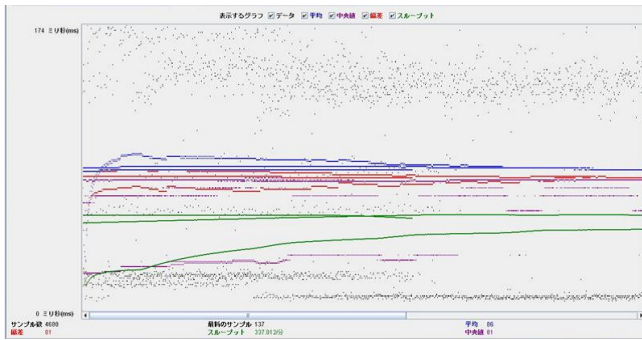


図 5 タイプ A のスループット曲線 (緑)

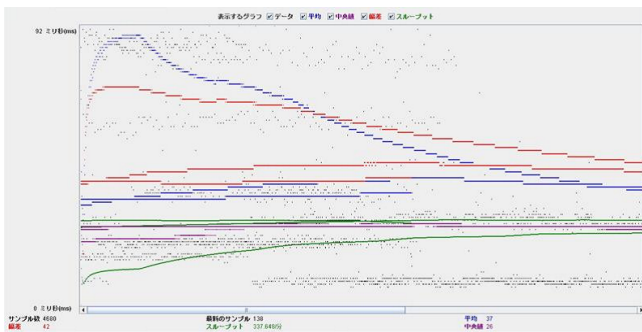


図 6 タイプ B のスループット曲線 (緑)

B 共に授業中の負荷に耐えることができると判断した。不正解時のモデル更新の意義や応答時間の比較から、試用試験には A よりも応答時間が速い B を用いることにした。

4.4 負荷試験のやり直し

章の始めに述べたが、4.3 の負荷試験内容には設定したパラメータに誤りが含まれていたことが後に分かった。誤りにより、4.3 節で行った負荷試験では最大負荷通りの負荷がかかっておらず、130 人の学生が同時にアクセスし 8 分程度かけて最大負荷より軽い負荷をかける、という負荷試験になっていた。

パラメータを設定し直して、4.3 節で述べた想定最大負荷の通りのテストシナリオを作成した。忠実な負荷規模の再現は難しいので、やはり、軽いテストシナリオと重いテストシナリオに分けた。2 種類のテストシナリオを A, B それぞれに 3 回ずつ適用して結果のノイズを抑えた。各試行の応答時間の平均を取ったものを図 7 に、スループット曲線をそれぞれ図 8 と図 9 に示す。

2 つの負荷試験の結果における違いをまとめる。90%Line, 最小値, 中央値, 平均値に違いはほとんど見られなかった。A の重いテストシナリオの正解時の最悪応答時間は 498ms から 652ms と増加した。また、A の軽いテストシナリオの不正解時における最悪応答時間は 670ms から 526ms に減少した。このように、最悪応答時間については若干の差が生じたが、試用中の負荷に耐えられると判断できる範疇の誤差であった。スループット曲線についても、サーバが処理限界を迎えている様子は見られなかった。

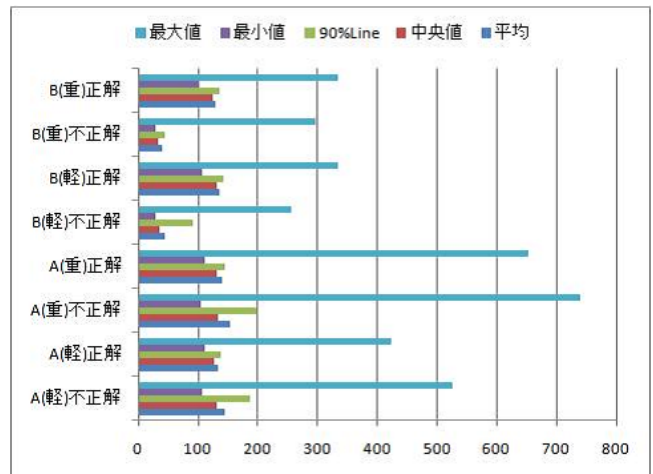


図 7 やり直した負荷試験における各タイプの不正解時正解時における応答時間 (ms)

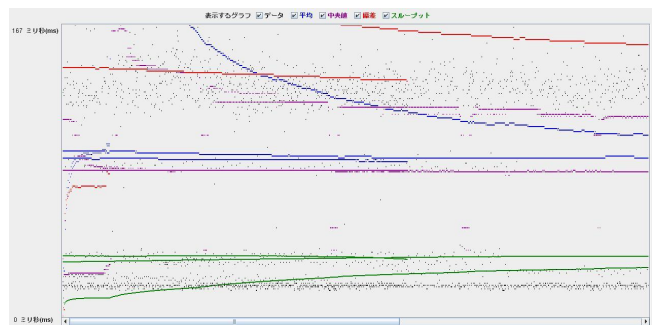


図 8 やり直した負荷試験におけるタイプ A のスループット曲線 (緑)

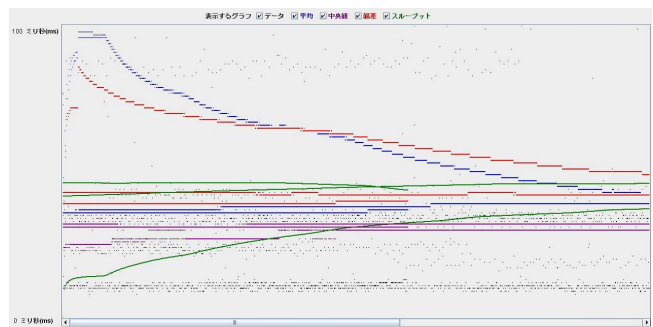


図 9 やり直した負荷試験におけるタイプ B のスループット曲線 (緑)

5. 試用試験の再開

授業中の試用を再開し、最後の 5 回分の授業において MILES を運用した。授業の途中からの試用となったので変更点が生じた。

1 つ目は、理解度モデルの初期状態についてである。当初は、学生が C 言語について何も知らない状態であったのでスコアが全て 0 という状態が初期状態であった。再開時に、学生はある程度の C 言語の学習が進んでしまっているのでこれを反映させる必要があった。MILES のログ取り機能は生きてままであったため、授業開始から再開直前の

学生の提出記録を時系列順に並べ、順番にモデル更新処理を行って初期状態とした。Bを用いているので、ドリル形式問題の誤答に関してはモデル更新を行わなかった。

2つ目は、推薦対象の問題についてである。当初は、推薦対象は全問題であったが、オリジナル問題だけとした。授業で使われている問題を推薦対象に含めしまうと、慣れてきたMAX/Cシステム以外の場所に授業の成績に関係する問題が出てくることになるため、混乱が起きることを懸念した変更である。

試用中の負荷は、授業がサーバダウンによって中断されることも、サーバが重くなる様子も見られず、概ね負荷試験の結果通りとなった。ログが生成されること、推薦アルゴリズムが動いていること、モデル更新処理が行われていることも確認でき、実装した機能に問題が見られない事を確認出来た。

5.1 利用状況についてのまとめ

アンケートログへの解答は0件であった。

アンケートログとは別に授業の最後で行ったアンケートの結果について述べる。アンケートには124名から解答があった。推薦機能を利用したことがあるかという問いにYESと答えた学生は17名であった。利用状況についての問いに関して17名の学生は、「問題を開かなかった」が8名、「問題を開いたが解答はしなかった」が6名、「問題に解答したことがある」が3名であった。これらの学生の中に「難しかった」「難しそうだった」と答えたのが4名おり、解答に辿りつけなかった原因の1つに「推薦された問題が難しい」が上げられそうである。また、推薦機能を利用をしたことがない学生から様々な意見が上がっている。「授業の本来の問題を解くことで手いっぱい」が23名。授業の終盤に試用が始まったことが影響したようである。「使おうと思わなかった」が10名。推薦機能に興味が無い学生も少なくなかった。「推薦機能の存在を知らなかった」が30名。周知が足りなかったようである。「使い方が分からなかった」が8名。推薦機能のページには使い方を記述していたが、それでは不十分だったようである。

6. おわりに

本稿では、C言語の学習者の理解度をモデル化し、適応的に推薦問題を選出することで学習者の実力に見合った学習ができることを目指すMILESシステムについて述べた。MILESの動作原理と実装した機能の説明をし、負荷試験と試用試験を行った結果を述べ、来年度からのMILESシステムの運用が出来る状態であることを確認した。

プログラミング学習を支援する研究はこれまでに様々な進められている。[10]では、過去の学習者の演習内容から強調フィルタリングを用いて課題の達成度を算出したり、学習意欲をもとに問題を選出する仕組みを提案している。授

業での運用結果から学習継続率が向上していることが報告されている。MILESはC言語の知識を細かく分けてスコアを持たせて理解度を表現し、問題に付与した出題者の意図と付き合わせて問題の選出を行っており、学習状況の表現方法や問題の選出方法が異なる。[11]では、問題の解答に細かなアドバイスを付与することで、学生自身に間違いに気付いてもらい解決させる方法を用いて、プログラミング学習を支援している。これにより、間違いに自力で気付かなかった学生が間違いに気づけるようになったと報告されている。

今後は、本格的に授業においてMILESを運用し、得られるデータや学生の声をもとに推薦の精度やモデル更新アルゴリズムの評価をする体制を整えたい。同時に、モデルの状態を学生に提示することで、学習状況を明確にし学習のモチベーションを高めようとするプロジェクトも動き出している。また、教師側にモデルの状態を見やすく提示することで、成績処理の一助にすることも考えられる。

謝辞 MILESの研究を昨年度まで共同で行った山田悠に感謝する。彼は主に、成績要素の階層化、推薦アルゴリズムの考案と実装、モデル更新アルゴリズムの考案と実装およびシミュレートによる改善を行った。

参考文献

- [1] 山田 悠：C言語学習システムにおける理解度モデルの構築とシミュレーションによる改良, Master's thesis, 明治大学大学院理工学研究科基礎理工学専攻情報系 (2011年度).
- [2] Sakai Foundation. Sakai CLE. <http://www.sakaiproject.org/>.
- [3] 藤井 聡一郎：統合eラーニング環境Sakaiを用いたプログラミング教育システムの構築, Master's thesis, 明治大学大学院理工学研究科基礎理工学専攻情報系 (2010年度).
- [4] 藤井 聡一郎, 大久保 和則, 玉木 久夫：MAX/C on Sakai - A Web-based C-Programming Course. In Proceedings of the 2nd International Conference on Computer Supported Education (to appear)(2010).
- [5] SpringSource. Spring framework. <http://www.springframework.org/>
- [6] The Apache Software Foundation. Apache Tomcat. <http://tomcat.apache.org/>.
- [7] ORACLE. MySQL. <http://www.jp.mysql.com/>.
- [8] The Apache Software Foundation. JMeter. <http://jmeter.apache.org/>.
- [9] Emily H. Halili: *Apache JMeter: A Practical Beginner's Guide to Automated Testing and Performance Measurement for Your Websites*, Packt Publishing (2008).
- [10] 田口 浩, 糸賀 裕弥, 毛利 公一, 山本 哲男, 島川 博光：個々の学習者の理解状況と学習意欲に合わせたプログラミング教育支援, 情報処理学会論文誌, Vol.48, No.2, pp.958-968 (2007).
- [11] 南山 幸紀：初心者のためのアドバイスの提供機能を持つプログラミング演習支援システムに関する研究, Master's thesis, 三重大学大学院工学研究科電気電子工学専攻 (2008年度).