

# 受講者の状況に応じた個別学習機能を有する 多言語対応プログラミング学習システムの実現

中島 秀樹<sup>†</sup> 高橋 直久<sup>††</sup> 細川 宜秀<sup>††</sup>

<sup>†</sup>名古屋工業大学電気情報工学科 〒466-8555 愛知県名古屋市昭和区御器所町  
E-mail: †hideki@moss.elcom.nitech.ac.jp, ††{naohisa,hosokawa}@elcom.nitech.ac.jp

**あらまし** 本稿では、受講者の状況に応じた個別学習を可能にするプログラミング学習システムの実現方法を提案する。提案システムは次の特徴をもつ。(1) 演習で行う問題と環境を記述するための言語 (EDML: Exercise Design Markup Language) を実現する。EDML は同じような難易度で内容の異なる複数の問題の生成、及び、受講者の理解度に応じた繰り返し学習のための答案評価環境と理解度に合わせた問題選択の指定を1つのテキストとして記述することを可能にする。(2) 既存のメールシステムを利用して学習システムの構成を単純化する方法を確立する。これより、遠隔学習システムに本質的なネットワークセキュリティの設定、一時ログの収集、過負荷時のバッファリングプログラムを新たに実装せずに遠隔学習システムの構築が可能になる。本稿ではまた、提案方式に基づき実現したプロトタイプシステムについて述べる。さらに、本システムを学部1年生のPascalプログラミングの演習に適用した結果についても述べる。

**キーワード** e-learning, パーソナライゼーション, Web とインターネット, ユーザインタフェース, プログラミング学習

## Implementation of a language-independent programming learning system with courses personalization according to each student

Hideki NAKAJIMA<sup>†</sup>, Naohisa TAKAHASHI<sup>††</sup>, and Yoshihide HOSOKAWA<sup>††</sup>

<sup>†</sup> Department of Electrical and Computer Engineering, Nagoya Institute of Technology Gokiso, Showa, Nagoya,  
466-8555 Japan

E-mail: †hideki@moss.elcom.nitech.ac.jp, ††{naohisa,hosokawa}@elcom.nitech.ac.jp

**Abstract** In this paper, we present an implementation method of an e-learning system for improving programming skills of novice learners according to their current abilities. The main feature of our method is to implement the EDML(Exercise Design Markup Language) which is used to design Q.A.cycles of a signal question. The Q.A.cycles indicate a process which novice learners solve several related questions until obtaining the knowledge from the corresponding question. Moreover, we also implement a function for creating related questions corresponding to current understandings of novice learners. By designing our Q.A.cycles previously, instructors can discover weakpoints of novice learners, and improve the Q.A.cycles according to the weakpoints. As the results, we expect that novice learners can improve their programming skills. We clarify the feasibility of our method by showing several experiments.

**Key words** E-learning, Personalization, Web and Internet, User Interface, Programming Education

### 1. はじめに

近年の情報化社会の急速な発展に伴い、社会における情報処理技術者のニーズが高まっている。特に、プログラミング言語の教育は、情報処理技術者の教育において重要な科目の一つといえる。しかしながら、情報処理関連の教師が不足しているために、十分な数の情報処理技術者を育成するには至っていない。

したがって、情報処理技術者育成プロセスを半自動化するための技術を確立することは重要である。

技術の習得の1手法として受講者に繰り返し学習を行わせる手法がある。受講者の理解度に応じて難易度の違う問題を与えながら繰り返し学習を遂行することで、技術習得効果の向上が期待できる。また、個別学習として、受講者1人1人に違う問題を与える手法もある。受講者に同じような難易度で内容の異

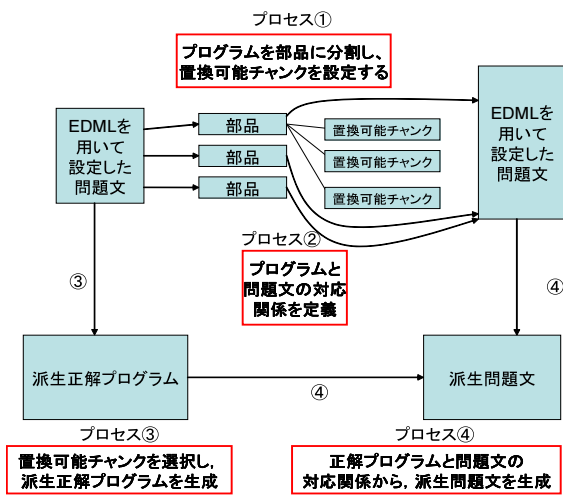


図1 同じ難易度で内容の異なる複数の問題の生成プロセス

なる問題を与えることで、受講者間の答案のコピーを防ぎ、個別に学習している環境を作り出すことができる。また、このように同じような難易度で異なる問題を与えることは、同じ受講者が繰り返し学習を行う場合にも有効である。これにより、技術習得効果の向上に貢献できると考えられる。

本稿では、受講者の理解度に応じた個別学習機能を有するプログラミング学習システムの実現方式を提案する。提案システムは、問題の難易度を受講者の理解度に応じて半自動的に変更しながら、繰り返し学習を行わせることを支援するものである。これにより、情報処理関連の教師が不足している問題を解消し、情報処理技術者育成に貢献する。ここで、繰り返し学習には2つの学習があるといえる。

- (1) 1つの問題に対し何度も取り組む学習
- (2) 複数の問題に関して、決められた条件のコースに応じていくつかの問題を解いていく学習

本稿では、(1)の1つの問題に対し何度も取り組む学習を対象とし、その演習設計の実現方式について説明する。

提案システムの特徴は次の3点にある。

**特徴-1** 演習で行う問題と環境を記述するための言語 (EDML: Exercise Design Markup Language) を実現する。EDMLは同じような難易度で内容の異なる複数の問題の生成、及び、受講者の理解度に応じた繰り返し学習のための答案評価環境と理解度に合わせた問題選択の指定を1つのテキストとして記述することを可能にする。EDMLは、具体的に次の2つの特徴をもつ。

**特徴-1-a** EDMLを用いて個別学習のための問題生成プロセスを記述することができる。図1に、同じ難易度で内容の異なる複数の問題を生成する手法を示す。教師がまず、正解プログラムを部品 (チャンクとよぶ) に分割し、チャンクに対して置換を行ったときに難易度が変わらず、正常なプログラムとして動作するような別の内容のチャンク (置換可能なチャンクとよぶ) を設定する (プロセス1)。次に、置換可能なチャンクと置換を行った時に、正解プログラムの各チャンクに対し、問題文中で置換が行われる場所とその内容との対応関係を設定する (プロセス2)。提案システムは、教師がプロセス1, 2の設定を

EDMLで記述することで、正解プログラムの置換可能チャンクの選択を行い、違う内容の正解プログラム (派生正解プログラムとよぶ) を生成する (プロセス3)。正解プログラムが生成されることで、設定した問題文との対応関係から、違う内容の問題文 (派生問題文とよぶ) が生成される (プロセス4)。これにより、派生プログラムと派生問題の総記述量を減らすことが可能になる。

**特徴-1-b** EDMLを用いて答案プログラムの評価環境、理解度に合わせた問題の選択条件を記述することができる。具体的には、教師が、プログラムを自動評価するために必要な正解プログラムと、その正解プログラムと答案プログラムとを比較するためのデータを設定する。提案システムは、教師が設定した環境のもと答案プログラムと正解プログラムを比較することで、繰り返し学習の制御を自動化することが可能になる。また、教師が評価結果から理解度を計算する計算法と、計算結果から次に行う問題を選択する条件を定義して設定する。提案システムは、設定した計算法と問題選択の条件を用い、受講者に対し、受講者の理解度に応じた難易度の問題を提示しながら繰り返し学習を行わせることが可能になる。

これらの特徴をもつEDMLを用いて、演習の設計を1つのテキスト上で行う方法を確立する。これにより、個別学習のための問題生成プロセスと受講者の理解度に応じた繰り返し学習プロセスを半自動化することが可能になる。

**特徴-2** Webの良好なユーザインタフェースを提供しつつ、既存のメールシステムを利用して学習システムの構成を単純化する方法を確立する。これにより、遠隔学習システムに本質的なネットワークセキュリティの設定、一時ログの収集、過負荷時のバッファリングプログラムを新たに実装せずに遠隔学習システムの構築が可能になる。また、メールをベースにしてWebのユーザインタフェースを実現することで、受講者の利用しやすい環境に対応が可能になる。

**特徴-3** 問題を生成する機能と受講者の答案を評価する機能をプログラミング言語に依存しないシステムアーキテクチャにより実現する。具体的には、答案プログラムの解析、コンパイル、実行などの言語依存機能は、システムの外部機能として位置づけ、問題生成機能と答案評価機能をこれらの外部機能と呼び出して利用する機構により実現する。これにより、教育現場での様々なプログラミング言語に対応が可能になる。

本稿ではまた、提案方式に基づき実現したプロトタイプシステムについて述べる。さらに、本システムを学部1年生のPascalプログラミングの演習に適用した結果についても述べる。

## 2. 関連研究

IMSでは、演習問題・試験問題に関する問題データベースの規格として、QTI (Question and Test Interoperability) を策定している [2]。QTI規格では、演習問題の出題形式、解答形式などを定めている。QTIは、受講者の答案と模範解答との間の簡単なマッチングで正誤判定を行うことを想定している。しかし、プログラミングの演習では、プログラムの文字列の比較を行うだけで正誤判定を行うのは不可能といえる。提案システムは、

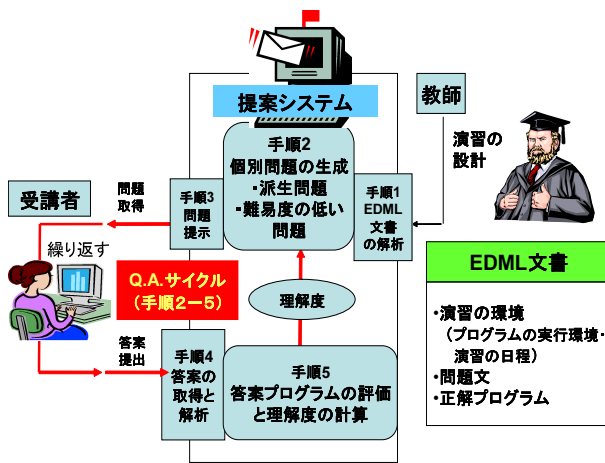


図2 提案システムによるプログラミング学習手順

受講者の答案プログラムと模範解答として用意した正解プログラムに対し、プログラムの実行時の変数の値を比較し、プログラムの正誤判定を行う。これらの評価を行うためのプログラムの実行環境や値を調べる変数の設定は、QTIでは規定されていない。それに対しEDMLは、プログラムを実行するための環境や、実行時に値を調べる変数を記述できるようにしている。

プログラミング学習における問題生成法の研究としては、PDG(Program Dependence Graph)を用いた空欄補充問題の作成法の研究が行われている[3],[4]。これらの研究では、プログラムの依存関係(データ依存関係、制御依存関係)から、問題文中の最適な空欄を見つけ、空欄補充問題を作成する手法を提案している。これにより、プログラム理解を高めるための空欄補充問題を容易に作成可能にしている。提案システムは、作成した問題をどう表現するか、または、どう繰り返し学習に利用できるようにするかに焦点をあてたものである。文献[3],[4]の方法は、提案システムの問題生成時に利用することで、問題を作成する時の手助けとなるといえる。

### 3. 提案システムの実現方式

本節では、受講者の状況に応じた個別学習を可能にするシステムの実現方式を提案する。現在の教育現場では、受講者1人1人の理解度に応じた学習を行うのは、教師に負担がかかり、困難である。そこで、提案システムは、受講者の答案から理解度を計算し、その理解度に応じた問題を次に行わせるという手法をとる。これにより、受講者の理解度に応じた学習を可能にする。提案システムによるプログラミング学習手順は次のとおりである(図2)

#### 手順1 EDML文書の解析

EDML文書を解析し、システムの内部表現に変換する。

#### 手順2 個別問題の生成

受講者に対し、理解度に応じて難易度の異なる問題を生成し提示する。(1回目ならば理解度は問題に与えられた最高点)

#### 手順3 問題提示

受講者に問題を提示する。

#### 手順4 答案の取得と解析

受講者が提出した答案を取得し、プログラムを抜き出す。

#### 手順5 答案プログラムの評価と理解度の計算

答案からプログラム(答案プログラム)を抜き出し、プログラムを評価する。評価結果から、理解度を計算する。このとき、十分な理解度に達していると判断した場合、演習終了。十分な理解度に達していない場合は、再び手順2へ。

演習で行う問題、演習の環境を設定する言語がEDMLである。EDMLで書かれた文書をEDML文書とよぶ。また、手順2-手順5までの流れをQ.A.サイクルとよぶ。Q.A.サイクルは1課題での繰り返し学習とする。

提案システムでは、受講者の答案プログラムの実行時の変数の値の妥当性を評価し、その評価結果を理解度として表すことにする。例として、変数 $x$ (20点)、 $y$ (80点)が評価対象であった時の理解度の表し方について述べる。答案プログラムの実行時の変数 $x$ 、 $y$ の値が、正解プログラムの実行時の変数の値と同じであれば、理解度を100点として点数化して表す。この時、変数 $x$ の値が異なれば、理解度は $100-20=80$ 点で、変数 $x$ 、 $y$ 両方とも異なれば0点となる。これより、評価すべき変数の数を多くし、点数の設定を細かく行うことで、問題に対する理解度を細かに計算できるといえる。

Q.A.サイクルを繰り返すことで、受講者の理解度に応じて難易度の違う問題を与えながら繰り返し学習を行わせることを可能にしている。これにより、受講者のプログラミング技術の向上を達成できる。

#### 3.1 Q.A.サイクルの設計方式

教師は、受講者がプログラミングの学習(提案システムのQ.A.サイクル)を行えるように、設計をする必要がある。提案システムはEDML文書を用いることで、受講者の理解度に応じた学習を可能にしている。教師は演習を設計し、EDML文書を作成する。提案システムでのEDML文書によるQ.A.サイクルの設計法について述べる。

##### 3.1.1 Q.A.サイクルの設計手順

提案システムにおけるQ.A.サイクルの設計は、演習のための環境設計と、問題設計からなる。環境設計では、まず、演習の期間として答案の提出期限を設定する。また、プログラムの実行環境を設定する。

問題設計では、問題文とそれに対応する正解プログラムを設定する。正解プログラムを設定するのは、答案プログラムの自動評価と派生問題の作成を行うためである。

提案システムでは、Q.A.サイクルの設計として、受講者の理解度に応じた個別学習を行えるようにするために、次にあげる項目を設定できるようにする。ここで、提案システムでいう個別対応とは、派生問題を生成することで、受講者ごとに違う問題を生成することと、受講者の理解度に応じて、穴埋め問題的に難易度を低くした問題を生成することである。

- プログラムのチャンク化

派生問題を生成するために、正解プログラムをチャンク化し、置換可能チャンクを設定する。

- チャンクに対して置換可能なチャンクのリスト

個別問題生成を行うために、チャンクに対して置換可能な

チャンクのリストを設定する。

● チャンクの難易度

問題の難易度を変化させるために、受講者の評価結果に対してプログラムのチャンクを提示するかないかを決定する条件を設定する。

チャンクに対して置換可能なチャンクのリストを設定することで、派生問題を作成することを可能にし、個別問題を生成可能にしている。また、チャンクの難易度を設定することで、受講者の理解度に応じて受講者に提示するチャンクの数や制御することを可能にしている。つまり、プログラムとして作成するのに難しいチャンクは、理解度の低い受講者には提示する。これにより、難易度の低い部分だけを考えさせる問題に変化させて、問題の難易度を変化させることができる。

以上のように、提案システムは、教師が演習の環境と問題を設計することで、受講者の理解度に応じて難易度の違う問題を与えながら繰り返し学習を行うことを可能にしている。

3.1.2 EDML 文書による Q.A. サイクルの設計例

EDML 文書は、演習の環境 (Environment タグ)、問題文 (question タグ)、正解プログラム (answer タグ) からなる。簡単な例として、四則演算を行うプログラムを作る問題の設計について説明する。この問題は2つの STEP からなり、STEP1 で整数の読みを行い、STEP2 で読み込んだ2数に対し四則演算を行うものとする。このとき EDML 文書では、図3のように記述できる。また、結果として図4、図5、図6のようなページが受講者に提示されることとなる。

演習の環境記述用タグを表1に、問題文と正解プログラムの記述用のタグを表2に示す。

表1 演習の環境記述用タグ (Environment タグ)

タグ	意味
< program >	プログラム言語の種類
< time >	提出期限
< testcase >	プログラム評価時の入力データセット
< mail >	メールでの演習用

表2 問題文 (question タグ) と正解プログラム (answer タグ) 記述用タグ

タグ	意味
< qchunk >	問題文中のチャンクリスト
< anschunk >	正解プログラム中のチャンクリスト
< chunk >	置換可能なチャンクのリスト
< nextquestion >	次に提示する問題と条件
< trace >	評価対象となる変数

EDML では、教師が問題文としてコメントと問題を解くための STEP (処理) の2つを設定するものとする。STEP を設定するのは、STEP 毎に指定した変数の値を調べることで、処理毎にプログラムの動作を確認することを可能にするためである。図3の例では、STEP1 (STEP タグの number=1) として2数を読み込み、STEP2 (STEP タグの number=2) として四則演算を行うという2つの処理にわけて設定することになる。これにより、図4

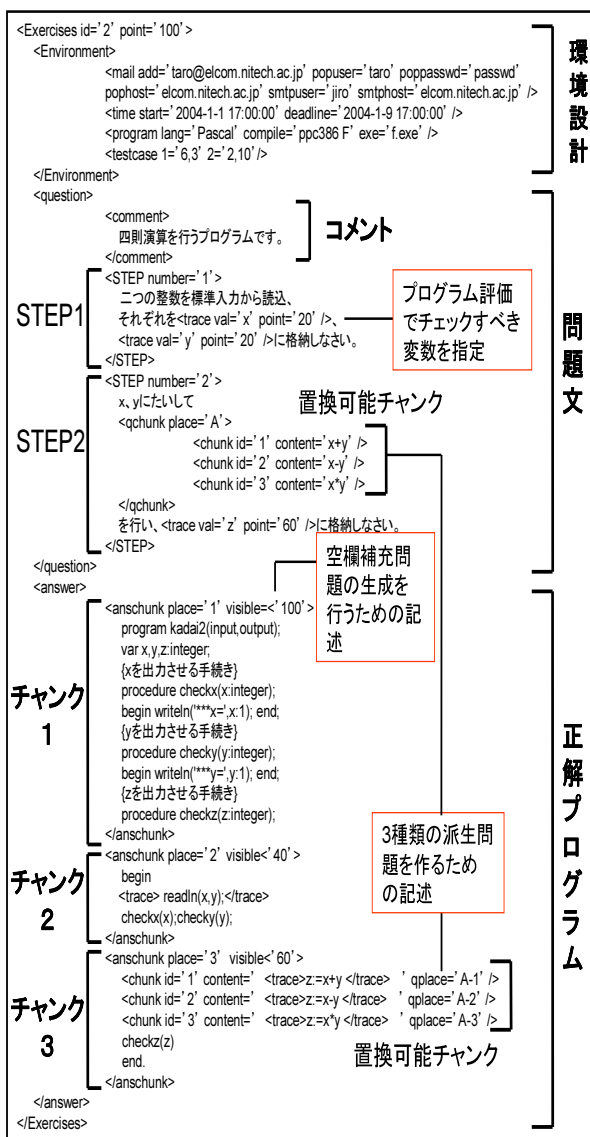


図3 EDML 文書の例

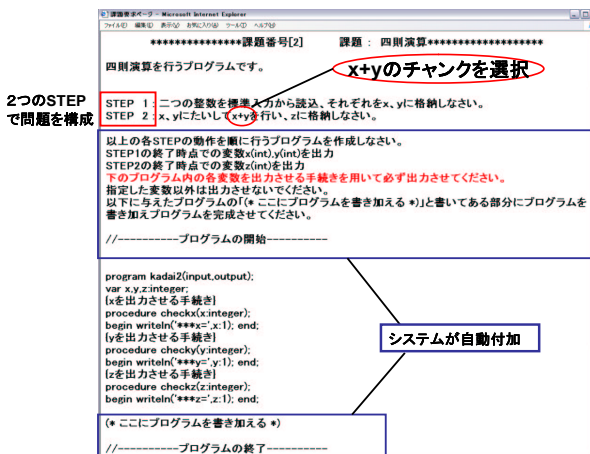


図4 受講者に提示される問題のページの一つ (理解度 100 で加算の問題)

例のように、提案システムが2つのSTEPを行う問題として受講者に提示する。正解プログラム (answer タグ) は、問題文でSTEPにわけたことを考えながらチャンクに分割して設計す



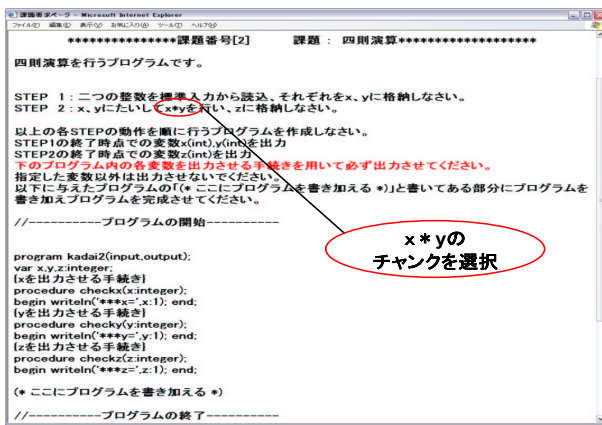


図5 受講者に提示される問題のページの一つ（理解度 100 で乗算の問題）

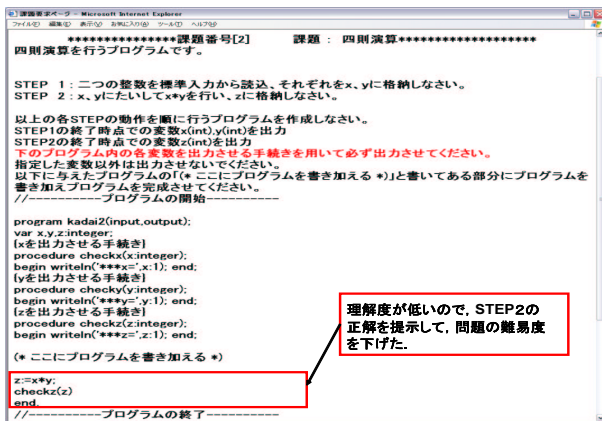


図6 受講者に提示される問題のページの一つ（理解度 40 で乗算の問題）

る (anschunk). 図3の例では、変数の定義、STEP1とSTEP2の3つにチャンクを分けて設計している。正解プログラムのanschunkタグには置換可能チャンク(chunk)を設定できる。この時、対応して問題文中の変化する部分(qchunk)にも置換可能チャンクを設定し、正解プログラムと問題文との間の対応関係(qplace)を設定する必要がある。図3の例では、anschunk place=3の部分(四則演算を行うSTEP)に置換可能チャンクを設定することで、問題文中のqchunk place=Aの部分に対応する置換可能チャンクを設定している。この置換可能チャンクを設定することで、派生問題を生成し、受講者毎に異なる問題を提示することが可能になる。つまり、図4が、chunk id=1(x+y)のチャンクがシステムによって選ばれたときのページで、図5が、chunk id=3(x\*y)のチャンクがシステムによって選ばれたときのページとなる。正解プログラムの各チャンクには、チャンクの難易度を点数化して設定する(visible)。これより、難易度が100以下のチャンク(visible=<100)を理解度が100以下の受講者に提示するといった設定が可能になる。つまり、受講者の理解度に応じて、穴埋め問題的に問題の難易度を変化させる設計が可能になる。図3の例では、anschunk place=1(変数の定義)が難易度100以下の理解度、anschunk place=2(STEP1)が難易度40より小さい理解度、anschunk place=3(STEP2)が

難易度60より小さい理解度の受講者に提示するように設計している。この設計により、理解度が40だったとき、図6のページのように、STEP2(四則演算)の解を提示することになる。つまり、STEP1を行うだけの問題となり、難易度の低い問題を生成できるといえる。

EDMLでは、教師が答案プログラムを評価するときの指標となる変数(トレース変数とよぶ)を設定するものとする(trace)。変数を指定することで、その変数のプログラム実行時の値の妥当性を評価することが可能になる。また、変数に点数をつけることで、妥当性を評価した結果を用い、理解度を点数化することが可能になる。具体的なプログラムの評価方法と理解度の計算方法は3.4で述べる。図3の例では、STEP1として、xとyの値を調べ、STEP2でzの値を調べるように設計している。この結果、図4の自動提示部分のように、どこでどの変数を出力すべきかといったコメントをシステムが自動で付加して提示することになる。また、出力形式を指定する手続きを正解プログラムに書くことで、評価を行うときに指定した出力形式のみを比較の対象(トレースデータとよぶ)とすることができる。図3の例では、checkx等という手続きを出力形式として指定していることになる。

このようにして、EDMLを用いてQ.A.サイクルの設計を行うことができる。EDMLを用いてQ.A.サイクルの設計を行うことで、答案の評価、個別問題の生成、繰り返し学習において理解度に応じた難易度の問題の提示を半自動化することが可能になる。また、プログラミング言語に依存するようなデータを、EDMLを用いて教師が設定することにより、提案システムの内部機能をプログラミング言語に依存しない形で実現することが可能になる。

### 3.2 EDML 文書の解析

ここでは、手順1(EDML文書の解析)の実現方式について述べる。提案システムは、EDML文書の内容から次にあげる項目の内容を取得する。

- 問題番号とその問題の最高点
- 問題文(置換可能チャンクとチャンクID付)
- 問題文中の置換可能チャンクに対応する内容(それぞれの内容にID(項目ID)がふられている)とその数
- 正解プログラム(置換可能チャンクとチャンクID付)
- 正解プログラム中の置換可能チャンクに対応する内容(それぞれの内容にID(項目ID)がふられている)とその数
- トレース変数とその出力方法
- テストデータセット(入力データセット)
- 問題要求期限・問題提出期限
- 正解プログラムのチャンクの難易度(点数)
- コンパイル・実行コマンド
- 講義のメールアドレス、送・受信メールサーバ情報

EDML文書の内容を解析し、上記の項目に分けてデータベースに保存することで、提案システムの内部表現に変換できる。

### 3.3 個別問題の生成

次に、提案システムがEDML文書を用いた個別問題の生成法(手順2)を例(図3)とともに説明する。

**手順 2-1** 派生問題を生成するために、まず受講者がシステムに登録した時に与えられる番号（ユーザ ID とよぶ）をもとに、図 3 に書かれた正解プログラム文中の置換可能チャンク（answer タグの内容）の中からチャンクを選択し、派生正解プログラムを作成する。例えば、ユーザ ID が 1000 のとき、chunk id=1 を選択したとすると、正解プログラムのチャンクとして、“z:=x+y;” を選択し、一つの派生正解プログラムを生成する。chunk id=3 を選択した場合は、正解プログラムのチャンクとして、“z:=x\*y;” を選択し、chunk id=1 の時とは異なる派生正解プログラムを生成する。

**手順 2-2** 受講者に提示する空欄補充問題用のプログラムを生成する。受講者が前に同じ問題を取り組んでいたら、前回の結果を用いて（取り組んでいなかったら問題の最高点とする）、受講者に提示する正解プログラムのチャンクを決定する。これにより、理解度の低い受講者には正解プログラムの一部を見せることで、問題としての難易度を低くする。例えば、前回の理解度（点数）が 40 点だった場合、図 3 では、anschunk place=1 が visible<100, anshunk place=2 が visible<40, anshunk place=3 が visible<60 であるので、40 以上の条件に当てはまる anshunk place=1 と anshunk place=3 を提示することが決定する。これにより、STEP2 の解を提示することになり、STEP1 を解くだけの空欄補充問題となる。

**手順 2-3** 正解プログラムのチャンクを決定したら、それに対応する問題文中のチャンクの内容を決定する。図 3 の場合では、正解プログラムの anshunk place=3 に対し chunk id=1 を選択すると、それに対応する問題文のチャンクとして qchunk place=A の問題文の chunk id=1 を選択する。つまり、問題文として、“x+y” を選択する。これより、受講者に問題として図 4 のページを提示することとなる。また、正解プログラムの anshunk place=3 に対し chunk id=3 を選択した場合は、問題文のチャンクとして qchunk place=A の問題文の chunk id=3 となり、問題文として、“x\*y” を選択する。これより、受講者に問題として図 5 のページを提示することとなる。ただし、図 4 と図 5 では理解度 100 の場合のページである。理解度が 40 であった場合は、図 6 となり、提示する正解プログラムが多くなる。

以上の処理を行うことで、受講者毎に問題文と正解プログラムを生成し、問題を提示する。

### 3.4 答案プログラムの評価と理解度の計算

ここでは、手順 5（答案プログラムの評価と理解度の計算）の実現方式について述べる。

Q.A. サイクルは理解度に応じた学習を行うための演習手順である。Q.A. サイクルにおいて、答案プログラムを評価し、理解度を計算する方法について説明する。

#### 3.4.1 答案プログラムの評価

多種類のプログラミング言語に対応する形で答案プログラムの自動評価を可能にする方法として、言語に依存しない比較対象を決め、それに対して評価する方法がある。そこで、提案システムでは、プログラム中使用される変数のトレース結果（トレースデータとよぶ）を比較対象とした。トレースデータは、プログラミング言語に依存しないものである。また、プロ

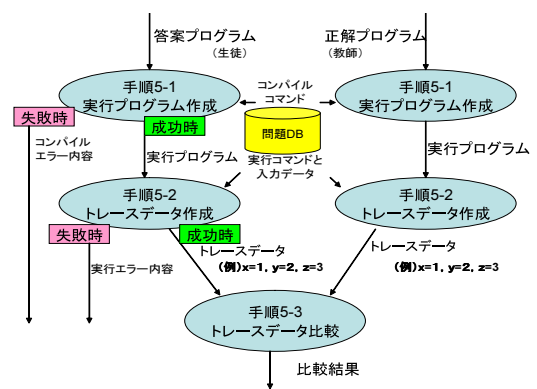


図 7 答案プログラムの評価の流れ

グラムの評価を行うためのテストデータは EDML 文書により設定しているので、プログラミング言語に依存しない形で比較機能を実現可能である。答案プログラムの評価の流れを図 7 に示す。次に、各処理について説明する。

#### 手順 5-1 実行プログラム作成処理

EDML 文書で設定したコンパイルコマンドを用いてプログラムのコンパイルを行う。コンパイルが成功すれば、実行プログラムが作成される。コンパイル失敗時は、コンパイルエラー内容を結果として返す。図 3 の例では、コンパイルコマンド（program タグの compile）として“ppc386 ファイル名”を用い、プログラムをコンパイルする。

**手順 5-2** EDML 文書で設定したテストデータと実行コマンドを用いて、プログラムを実行する。正常に実行できたら、EDML で設定したトレース変数のトレースデータを取得する。実行エラー時はエラー内容を結果として返す。図 3 の例では、実行コマンド（program タグの exe）として“ファイル名.exe”と、入力データ（testcase タグ）として“6, 3”を用い、プログラムを実行する。そして、変数 x, y, z の実行時の値をトレースデータとして取得する。

**手順 5-3** 受講者の答案と正解プログラムのそれぞれのトレースデータを比較する。EDML で設定したトレース変数に対し、比較結果が一致したら、結果を正解として返し、間違いなら、どの変数が間違いであったかを結果として返す。

以上の処理を行い、受講者の答案プログラムの評価結果を決定する。また、トレースデータ作成処理（手順 5-2）とトレースデータ比較処理（手順 5-3）は、比較結果が正解であれば、EDML で設定したテストデータのセット数だけ繰り返される。図 3 の例では、testcase タグの内容として 2 種類あるので、比較結果が正解であれば、2 回行われることになる。

#### 3.4.2 理解度の計算

評価結果に応じて理解度を計算する。理解度を計算することにより、手順 2 で受講者の理解度に応じた難易度を持つ問題を提示することができる。次に、評価結果をもとにユーザの理解度を計算（手順 5-4）する方法について説明する。

評価結果が全ての変数で正しかった場合、その問題の理解度は、その問題に設定された最高点となる（図 3 の例では 100 点）。間違えた変数があれば、その変数に設定した点数を最高点

から引いた結果が理解度となる。例えば、図3の例で変数zの値が間違っているという評価結果となった時は、変数zは60点 (trace val='z' point='60') であるので、理解度は100-60=40点となる。

### 3.5 ユーザインタフェースの実現方式

ここでは、手順3(問題提示)と手順4(答案の取得と解析)の実現方式について述べる。手順3は、手順2(個別問題の生成)によって作成した問題の内容をhtmlにしてWebページとして表示することで実現する。次に手順4の実現方式について述べる。

演習を行う上で、学習環境は受講者毎に違うものである。学習環境とは、学習を行う日程・時間、場所や使用するアプリケーションなどである。提案システムは、Webとメールの2つのユーザインタフェースを提供することで、受講者に対し、ネットワークに繋がる環境があれば、いつでも、どこでも学習可能な環境を提供する。

いつでも学習を行えるようにするために、システムが過負荷によりダウンすることをできるだけ防ぐ必要がある。Q.A. サイクルの中で、答案の評価を行う処理は他の処理に比べて時間がかかる処理であり、重い処理といえる。この処理のインタフェースとしてWebを用いると、Webは同期的な処理をするものであり、負荷がかかりやすくその後に影響を与える。そこで、提案システムでは、答案の評価処理において、メールベースで処理を行う手法をとる。メールベースで処理を行うことで、非同期的に処理を行うことが可能になる。これにより、システムにかかる負荷を軽減することが可能になる。メールベースでの処理とは、Webからの答案提出に対し、システム内でメールに変換してから解析を行うことを意味する。また、提案システムとして既存のメールシステムを利用することで、システムの構成を単純化でき、問題発生時のバグの早期発見にもつながる。メールシステムを利用することで、次の利点がある。

**利点1** メールシステムの場合、ファイアウォールの設定によりメールの送受信のポートが開いている可能性が高く、環境の設定を余計にする必要がなくなる。

**利点2** メールシステムで一時ログをとる場合、Webとは違い既存のシステムの機能として取得することが可能で、ログを残しやすく、本学習システムに一時ログを取得する機能を拡張する必要がなくなる。

**利点3** メールシステムを使用することで、ログをメールシステムに残しておくことができ、本学習システムのダウン時において本学習システムが行わなければならない処理を後回しにできる。これにより、システムダウン時の簡単な解決策として、利用することができる。

以上の機能をメールシステムで行うことができるため、提案システムを単純化できる。

## 4. 提案システムの評価と考察

提案システムの有効性を確認する目的で、プロトタイプシステムを実装し、その評価を行った。実装したプロトタイプシステムの構成図を図8に示す。プロトタイプシステムは、問題生

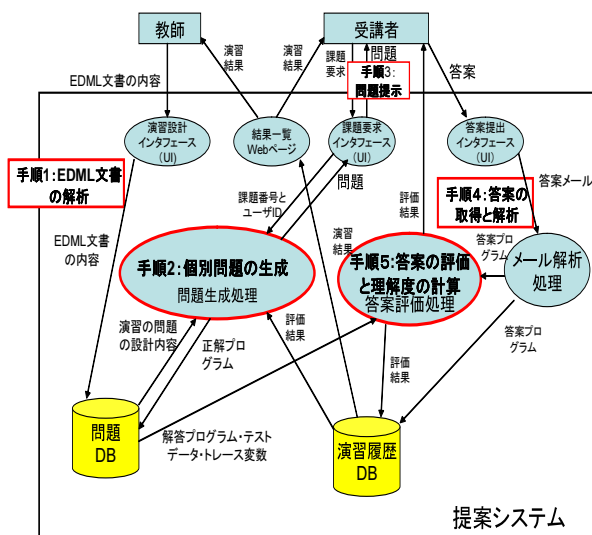


図8 プロトタイプシステムの構成図

成処理により手順2:個別問題の生成処理を行う。答案評価処理では、手順5:答案プログラムの評価と理解度の計算処理を行う。メールの解析処理は、提出された答案メールから、答案プログラム、ユーザID、課題番号、提出時間の4つの情報を取得する機能として実現した。ユーザインタフェースは、軽い処理にはWebページを用い、答案の評価処理に繋がるユーザインタフェースでは、メールベースとして実装した。メールベースでの処理は、Webページとして用意したフォームに答案プログラムを入力させ、フォームの内容を取得して、メールの本文に付加し、プロトタイプシステムに送信する形で実装した。

プロトタイプシステムを、名古屋工業大学電気情報工学科の情報工学 Pascal 講義の演習に適用した。その結果と受講者のアンケートを基に、個別学習が行えたかどうかの評価・考察を行った。受講者は学部1年生(171人、前期にPascal及びプログラムの基礎を学習済み)と学部2年生以上(再履修36人、2年生時にC言語を学習済み)であった。評価するデータは、10月17日から、12月9日までの演習結果を対象とした。また、演習は2週に1回課題が出され、隔週で演習を行わなければならない指定の時間が用意されている。

### 4.1 EDMLを用いた演習設計の効果

#### 4.1.1 受講者の理解度に応じた繰り返し学習によるプログラミング技術の向上効果

提案システムは、Q.A. サイクルを繰り返し行わせることで、受講者のプログラミング技術の向上を達成することを目的としている。Q.A. サイクルでは、不正解だった受講者に、理解度に応じた難易度の問題を提示することができる。図9に難易度の高い問題で間違え、難易度の低い問題に取り組んだ受講者の割合を示す。図9より、難易度の高い問題(課題D, E)では正解できなかった受講者の半分近くが、難易度の低い問題(課題D-2, E-2)では正解している。これにより、EDML文書を登録するだけで、受講者に対し、繰り返し学習を行う過程で、受講者の理解度に応じた難易度の問題を行わせることができた。これにより、繰り返し学習と受講者の理解度に応じた難易度も



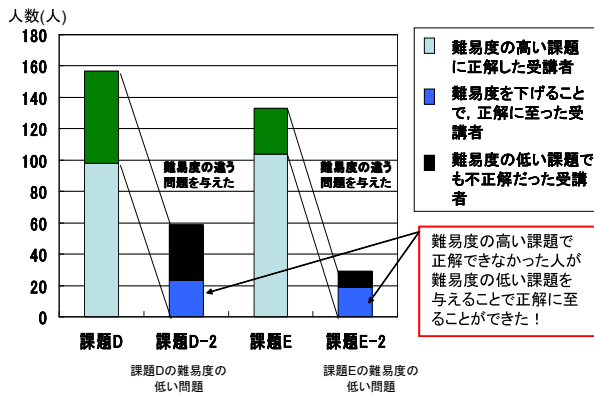


図9 難易度の違う問題により正解に至った受講者

つ問題の提示機能の半自動化できたことが確認できた。

#### 4.1.2 多種類のプログラミング言語への対応

提案システムは、多種類のプログラミング言語（コンパイル・実行を行って動作するプログラミング言語）に対応することを目的としている。プロトタイプシステムは、内部機能を言語に依存しない形で実装した。プロトタイプシステムをPascalの講義に適用した結果、正常に動作した。これにより、Pascalで動作することが確認できた。確認のために、コンパイル・実行の二つのコマンドで動作する言語であるJavaで動作するか試した。結果は、正常に動作した。これにより、提案システムは多種類のプログラミング言語（コンパイル・実行を行って動作するプログラミング言語）に対応可能であることが確認できた。

#### 4.1.3 派生問題の作成

提案システムは、複数の問題を生成することを目的としている。演習で実際に問題を作成した結果、一つの問題に対しチャプクの内容を複数登録すれば、複数の問題に変換することができた。これにより、提案システムはEDML文書を登録すれば、複数の問題を作成可能であることが確認できた。

### 4.2 提案システムの利便性の評価

#### 4.2.1 メールシステムを利用したことによる効果

提案システムは、メールベースで重い処理を行うことで、重い処理には非同期的処理を行いシステムの安定性を高めることを目的としている。システムの安定性に対し、演習に適用した結果をもとに評価する。演習は一度に約80人の受講者がPCを利用できる環境で行った。現在まで負荷の集中によるシステムダウンはおきていない。集中時では、1分間に最大12通もの答案を解析した。10分間では40通以上の答案を正常に解析した。その場合にも、システムは正常に稼動することを確認した。また、メールシステムを組み込むことで、比較的短期間でこのような、安定運用可能なシステムを実現することができた。

#### 4.2.2 受講者の学習時間と場所

演習の履歴から答案の提出時間を調べた結果、大学の講義時間以外にも様々な時間で答案の提出を行っている受講者がいた。答案の提出時間は学習時間と考えることができるので、提案システムはいつでも学習可能な環境をサポートできた。

学習場所と提出方法については、演習の履歴からの調査結果

表3 答案の提出場所

場所	人数	提出方法
計算機室（大学内）	67	メール
計算機室（大学内）	133	Web
家庭	4	メール
家庭	5	Web

(表3) から、少なくともあるが家庭で学習している受講者がいることを確認できた。また、提出方法はWebページを用いる受講者の方が多いが、メールでの提出も多く、提出方法は人によって使いやすい方法が分かれることが確認できた。これにより、提案システムは、受講者に対し、どこでも（ネットワークに繋がる環境であれば）学習可能な環境を提供することができた。

## 5. まとめ

本稿では、実際の講義において、個別に対応した問題を生成することを考え、教師になるべく負担にならないように受講者の状況に応じた難易度の問題を作成できることと、受講者が学習を行いやすくすることを目的とした、プログラミング学習システムを提案した。得られた成果をまとめると次のようになる。

- 提案システムにより、受講者の理解度に応じた問題を自動生成することが可能になった。
- 提案システムを用いることで、多種類のプログラミング言語の演習に適用できるようになった。

実装したプロトタイプシステムでは、提案システムの全てを実現してはいない。プロトタイプシステムでは、EDML文書の登録として、EDML文書の内容を入力フォームから入手することで実装した。つまり、EDML文書のテキストを解析しない形で実装した。この場合でもシステム上問題ないが、テキストファイルで書いたほうが一括して扱いやすく、また、テキストファイルをメールで送るといったやり方のほうがやりやすい場合もある。このため、EDML文書の解析機能を実現する予定である。また、正解プログラムのチャプクが置換可能かどうか判断する助けとなる機能を実現する予定である。これは、プログラム解析が必要であるため、外部機能として実現する。

また、派生問題に対し定量的な評価と、難易度の違う問題に対し、受講者のプログラミング技能の向上に役立っているのかといった評価を行っていききたい。

## 文 献

- [1] 伊藤健二：e-learning とは何か、情報処理学会，Vol.43 No.4, pp.394-400(2002)
- [2] IMS(The Instructional Management Systems)：IMS Question & Test Interoperability Information Model Specification v1.01，<http://www.imsproject.org/question/qinfo101.html>，(2000)，ALIC(先進学習基盤協議会)：IMS Question & Test Interoperability Information Model Specification v1.01 要約，<http://www.alic.gr.jp/activity/2000/iop/wg5.htm>，(2001)
- [3] 柏原昭博，久米井邦貴，梅野浩司，豊田順一：プログラム空欄補充問題の作成とその評価，人工知能学会論文誌 16 巻 4 号 C(2001)
- [4] 小野祐昭：プログラム評価のための問題の自動生成，[http://softeng-www.cs.shinshu-u.ac.jp/Thesis/2002/Undergraduate/cono/cono\\_final.pdf](http://softeng-www.cs.shinshu-u.ac.jp/Thesis/2002/Undergraduate/cono/cono_final.pdf) 信州大学工学部情報工学科 海尻・海谷研究室 卒業論文 (2003)