

# アクティブデータベースを用いた プログラミング演習クラスの視覚化システムの実現

西川 直希<sup>†</sup> 高橋 直久<sup>†</sup>

<sup>†</sup> 名古屋工業大学大学院 工学研究科 〒466-8555 愛知県名古屋市昭和区御器所町  
E-mail: <sup>†</sup>nishikawa@moss.elcom.nitech.ac.jp, <sup>††</sup>naohisa@nitech.ac.jp

あらまし 本稿では、アクティブデータベースを用いたプログラミング演習クラスの視覚化システムを提案し、その実現法について述べる。提案システムは以下の3つの機能を持つ(1)演習システムと受講者との間のインタラクションに対応するイベント(演習イベント)を収集する機能(2)ECAルールの系列として記述されたプログラミング演習クラスの視覚化の方針(演習状況監視ポリシー)を解釈し、それぞれのポリシーごとにプログラミング演習クラスの最新状況を保持するデータ(演習状況ベースデータ)を更新する機能(3)演習状況ベースデータと座席情報を結びつけた演習クラスの状況を表す画面(演習クラスアクティビティ)を生成する機能。教師が選択した演習状況監視ポリシーに応じた演習クラスアクティビティを表示する機能。これらの機能により、教師が答案作成が遅れているなど進捗上問題のある受講者を直ちに発見し、その座席に指導に行くことが可能になる。本稿ではまた、提案システムのプロトタイプを実装し、提案システムの妥当性を評価した結果について述べる。

キーワード e-Learning, プログラミング演習, アクティブデータベース, データの可視化

## Implementation of a Programming Exercise Class Visualization System with Active Database

Naoki NISHIKAWA<sup>†</sup> and Naohisa TAKAHASHI<sup>††</sup>

<sup>†</sup> Department of Computer Science and Engineering Graduate School of Engineering,  
Nagoya Institute of Technology  
Gokiso, Showa, Nagoya, 466-8555 Japan  
E-mail: <sup>†</sup>nishikawa@moss.elcom.nitech.ac.jp, <sup>††</sup>naohisa@nitech.ac.jp

**Abstract** In this paper, we propose a programming exercise class visualization system with active database, and describe the implement method. The proposed system has the following three functions: (1) it collects exercise events that corresponds to interactions between a learner and a programming exercise system, (2) it interprets multiple visualization policies described by a teacher, and updates class status data in the way which are specified by each policy, (3) it visualizes class status data corresponding to the policy selected by the teacher. These functions enable the teacher to discover the location of learners having at once, and go to the learner's seats so as to help to help them. In this paper, we also describe the prototype of the proposal system, and clarifies the effectiveness of the propoal system.

**Key words** e-Learning, Exercise of Programming, Active Databese, Visualization of Data

### 1. はじめに

我々は、問題作成と答案評価の自動化機能により問題の提示から答案の評価までを効率的に行うプログラミング演習支援システム[1]を提案し、大学の演習授業において運用している。本稿では、このような演習システムを利用した演習形態の多人数クラスをプログラミング演習クラスと呼ぶ。プログラミング演

習クラスにおいて、担当教師やティーチングアシスタント(以下、教師とする)はプログラミング演習クラス全体や各受講者の演習状況を適切に判断し助言や指導を行う必要がある。しかし、多数の受講者のさまざまな演習行動を分析することは教師にとって大きな負荷である。また、教師がプログラミング演習クラスから答案作成が遅れているなど進捗上問題のある受講者を直ちに発見し、その座席に指導に行くのは困難である。

そこで本稿では、アクティブデータベースを用いたプログラミング演習クラスの視覚化システムの実現方式を提案する。

提案システムの特徴は次の3点にある。

特徴-1 演習システムと受講者との間のインタラクションに対応するイベント（演習イベント）を収集する機能を実現する。受講者の演習行動を演習イベントの系列によって表すことで、多数の受講者の様々な演習行動を容易に分析可能にする。また、演習イベントを用いることでアクティブデータベースにより演習状況の変化をデータベースに反映させることを可能にする。

特徴-2 複数の演習状況監視ポリシーを解釈し、それぞれの演習状況監視ポリシーごとに演習状況ベースデータを更新する機能を実現する。ここで、演習状況監視ポリシーとはプログラミング演習クラスの視覚化の方針である。教師は演習状況監視ポリシーをECA(Event-Condition-Action) ルールの系列として記述する。また、演習状況ベースデータとはプログラミング演習クラスの最新状況を保持するデータである。これにより、教師の要求に応じたデータを得ることを可能にする。

特徴-3 演習クラスアクティビティを生成する機能を実現する。ここで、演習クラスアクティビティとは演習状況ベースデータと座席情報を結びつけた演習クラスの状況を表す画面である。また、教師が選択した演習状況監視ポリシーに応じた演習クラスアクティビティを表示する機能を実現する。これにより、教師が進捗上問題のある受講者を直ちに発見し、その座席に指導に行くことを可能にする。

本稿では、2. で提案システムの実現法を示し、3. でプロトタイプシステムの実際の演習クラスへの適用について述べる。次に、4. で提案システムの評価と考察を行う。さらに、5. で関連研究について述べる。最後に、6. で結果をまとめる。

## 2. 提案システムの実現法

本章では提案システムの実現法について述べる。

### 2.1 提案システムの概要

提案システムを用いた演習モデルを図1に示す。プログラミング演習クラスの受講者は演習支援システム[1]を用いて演習を行う。演習支援システムは、受講者情報や教室の座席レイアウト、受講者がどの座席の計算機で演習しているかという座席情報を演習データベースに保存、管理している。受講者は演習支援システムから問題を要求し、閲覧する。次に、受講者はそれぞれの計算機でエディタなどを用いて答案を作成する。そして、受講者は作成した答案を演習支援システムに送信する。演習支援システムは答案を受信すると、答案の自動評価を行い受講者に評価結果を提示する。次に、演習支援システムはこのような演習履歴を演習データベースに保存する。そして、演習支援システムは演習履歴を分析し、受講者の習得度に合わせた問題を受講者に提示する。演習支援システムを用いた演習では、このような問題提示、答案受信、評価結果送信を繰り返し行う。また、受講者は答案の作成と提出以外にも、答案作成のために教材提示システムにおける教材の閲覧や、質問掲示板における質問の書き込みなどを行う。本稿では、演習システムと受講者との間のインタラクションを演習イベントと呼ぶ。演習支援シ

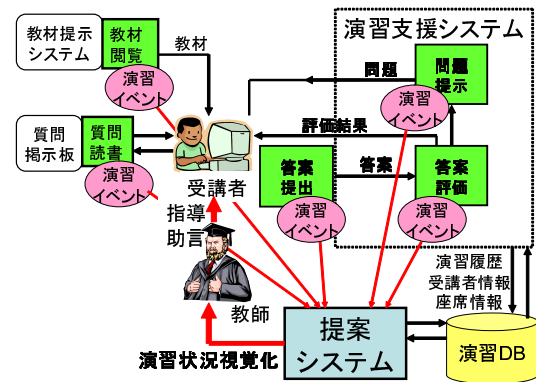


図1 提案システムを用いた演習モデルと演習イベント

テムでは、課題要求、答案送信、正解メッセージ取得、不正解メッセージ取得を演習イベントとして観測可能である。また、コンパイル機能、アセンブル機能、エディタ機能を持つアセンブリ言語シミュレータ[2]を教材提示システムとして用いる場合、コンパイル、アセンブル、エディタ操作を演習イベントとして観測可能である。

このような演習において、我々は、教師がプログラミング演習クラスから進捗上問題のある受講者を発見し、指導を行うことを可能にすることを目標とする。提案システムはプログラミング演習クラスの視覚化を行い、教師に提示することでこの目標を達成する。プログラミング演習クラスの視覚化を行うためには、受講者の演習行動を分析しなければならない。受講者の演習行動を分析するために、提案システムは演習システムで発生する演習イベントを収集し、演習イベントの系列により受講者の演習行動を表す。これにより、受講者の演習行動を容易に分析することを可能にする。しかし、多数の受講者が演習を行うプログラミング演習クラスではたくさんの演習イベントが発生する。また、受講者は取り組む問題や取り組み方、進度がそれぞれ異なる。そのため、教師が時間経過による受講者の演習行動の変化を追いかけるのは大変困難である。この問題を解決するため、提案システムはアクティブデータベース[3]を用いる。これにより、受講者の演習行動に応じた演習状況の視覚化を可能にする。

#### 2.1.1 アクティブデータベースの利用

演習中のプログラミング演習クラスでは、図1のようにさまざまな演習イベントが発生する。その演習状況を視覚化するためには、これらの演習イベントを観測して、その変化に応じた処理を行う必要がある。そこで、提案システムは環境の変化に応じて自動的に処理を行うためにアクティブデータベース[3]を用いる。アクティブデータベースに対する処理を記述するルールとして、ECAルールを用いる。提案システムにおけるそれぞれの意味は以下ようになる。

イベント：受講者と演習システムとのインタラクションにより発生した演習イベント

コンディション：ルールを実行するための制約条件

アクション：演習状況の更新動作

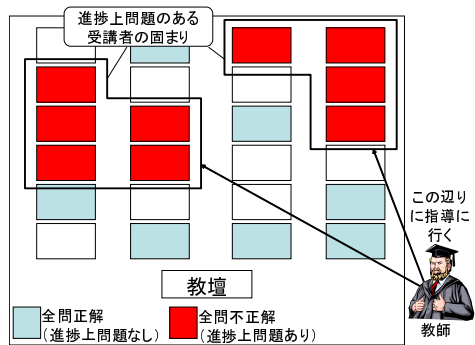


図2 プログラミング演習クラスの視覚化イメージ

アクティブデータベースを用いた演習状況の変化の反映処理は、次のとおりである。演習イベントが発生すると、ECA ルールベースから適合する ECA ルールを検索する。コンディションを満たしたルールが存在すれば、演習状況の更新を行う。これにより、演習イベントに応じた演習状況の視覚化を可能にする。

### 2.1.2 演習状況の視覚化手法

プログラミング演習において、教師からはプログラミング演習クラス全体の演習状況を見たいという要求がある。また、プログラミング演習クラス全体の中で困っている受講者や問題のある受講者を発見したいという要求がある。これらの要求を満たすためには、教師がプログラミング演習クラス全体の状況を一覧して把握する必要がある。そこで提案システムは、図2に示すようなプログラミング演習クラスの演習状況を表す画面を生成し、教師に提示する。図2の座席に対応する小さな長方形が受講者を表す。受講者の演習状況を演習室の座席表と結びつけて視覚化することにより、教師が進捗上問題のある受講者をただちに発見し、その座席に指導に行くことを可能にする。さらに、教師が受講者相互の座席の位置関係を考慮しながら受講者の進捗状況を分析することを可能にする。

ここでは、教師が求めるプログラミング演習クラス全体の状況を次のような方法で表すことを可能にする。(1) ある1つの課題に対する進捗状況 (2) 複数の課題に対する進捗状況 (3) ある課題に対し取り組んでいる時間 (4) 進捗上問題のある受講者の一覧。このとき(1)~(4)の視覚化例を図3~図6に示す。ここで、図3~図6の長方形は、図2の受講者を表す長方形一つ一つに対応している。図3はある1つの課題における、課題要求、教材閲覧、答案正解の演習イベントの発生の有無を視覚化している。これにより、教師が課題に対する進捗状況を把握可能となる。図4は複数の課題に対する進捗状況を視覚化している。これにより、複数の課題が設定された演習において教師が演習全体における進捗状況を把握可能となる。図5はある1つの課題における、課題要求、答案不正解、答案正解、教材閲覧の演習イベント発生時の時間経過を視覚化している。これにより、受講者の課題に対する進捗状況を時間と関係付けて把握可能となる。図6は図4と同様に、複数の課題に対する進捗状況を視覚化しているが、全ての課題が不正解或未解答であった場合、背景の色を変化させている。これにより、教師が進捗上問題のある受講者を容易に把握可能となる。

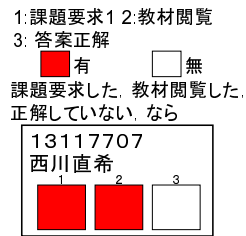


図3 演習状況視覚化の例(1)

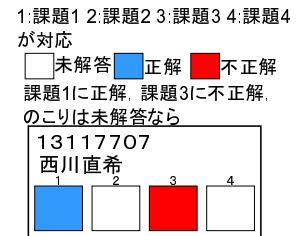


図4 演習状況視覚化の例(2)

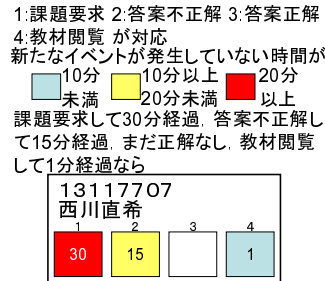


図5 演習状況視覚化の例(3)

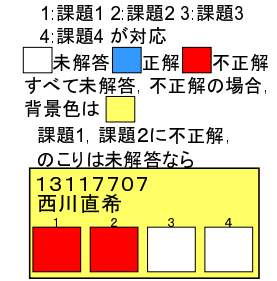


図6 演習状況視覚化の例(4)

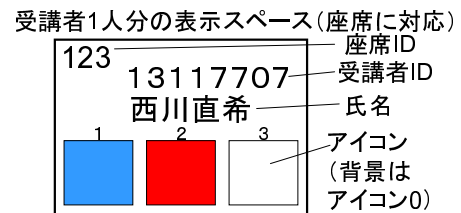


図7 受講者の演習状況表示スペース

教師の要求に合わせた多種類の演習状況の視覚化を行うためには、その要求に応じて表示の意味付けを変化させることができる必要がある。そこで、提案システムはプログラミング演習クラスの視覚化の方針を記述した演習状況監視ポリシーを用いる。これにより、教師の要求に応じて表示の意味づけを設定、変更することを可能にする。

提案システムは、決まった要素の組み合わせで表示の意味付けを変化させるために次のような表示手法を用いる。受講者の座席に対応した演習状況表示スペース(図7)にアイコンを設置する。アイコンとは、割り当てた課題の演習状況や、割り当てた演習イベントの発生状況を色や文字で表示するためのスペースである。アイコンIDの0は受講者個人表示枠の背景で、1~n以降には課題番号や演習イベントIDを割り当てる。提案システムは、アクティブデータベースを用いてアイコンの色や文字、クリック時のリンク先を変化させることにより受講者の演習状況の視覚化を可能にする。

演習状況監視ポリシーには設置するアイコンの数を記述する。また、どのような演習イベントが発生すればアイコンの表示をどう変更するかをECAルールとして記述する。これにより、複数の演習行動を関係付けて分析することを可能にする。

ここで、教師が多種類の視覚化を見るためには、演習状況監視ポリシーを選択可能である必要である。そこで、提案システムは演習状況監視ポリシーごとに生成された演習状況の視覚化を切り替えて表示する機能を実現する。これにより、教師の要

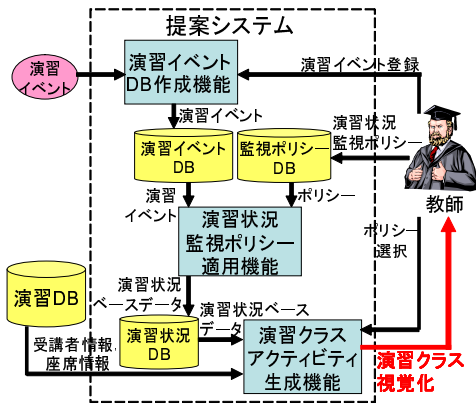


図 8 提案システムの構成図

求に合わせた演習状況の視覚化を可能にする。

### 2.1.3 提案システムの構成

提案システムの構成を図 8 に示す。提案システムは、教師が指定した演習システムと受講者との間のインタラクションに対応する演習イベントを収集する（演習イベント DB 作成機能）。また、教師は提案システムに対し演習状況監視ポリシーを登録する。提案システムは発生した演習イベントに対して演習状況監視ポリシーを適用することにより、演習状況監視ポリシーごとに演習状況ベースデータを更新する（演習状況監視ポリシー適用機能）。演習状況ベースデータはプログラミング演習クラスの最新状況を保持するデータである。具体的には、演習状況ベースデータには演習クラスの各受講者の文字や色などの描画内容を記録する。プログラミング演習クラスの状況を視覚化する場合、教師は閲覧したい演習状況監視ポリシーを選択する。提案システムは選択された演習状況監視ポリシーの演習状況ベースデータと演習 DB の受講者情報、座席情報を用いて演習クラスアクティビティを生成する（演習クラスアクティビティ生成機能）。提案システムは教師に対し、生成した演習クラスアクティビティを表示する。これにより、教師が要求した演習状況を視覚化することを可能にする。また、教師が進捗上問題のある受講者を直ちに発見し、その座席に指導に行くことを可能にする。

### 2.2 演習イベントデータベース作成機能の実現法

演習イベントデータベース作成機能は、演習システムと受講者との間のインタラクションに対応する演習イベントを収集する。

演習支援システム [1] を用いた演習における演習イベントには、課題要求、答案作成、答案送信、正解メッセージ取得、不正解メッセージ取得がある。提案システムはこのうち演習支援システムで観測可能な課題要求、答案送信、正解メッセージ取得、不正解メッセージ取得の各演習イベントを収集する。また、プログラミング演習では、アセンブル機能、コンパイル機能、エディタ機能等を持つアセンブリ言語シミュレータ [2] やプログラミング質問掲示板など、サーバ側で機能を提供する演習システムを用いる場合がある。受講者はこれら副次的な演習システムにおいても、シミュレータのエディタ操作やコンパイル、アセンブル、質問書き込み、質問閲覧などの操作を行う。これら

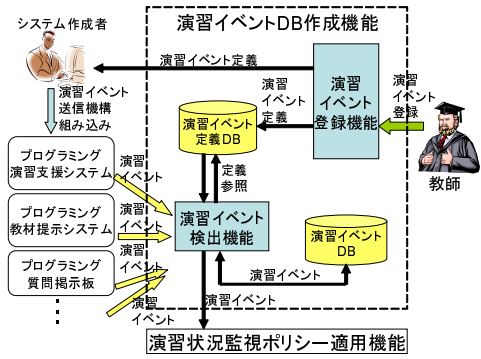


図 9 演習イベントデータベース作成機能の構成図

表 1 演習イベント定義 DB の構造

| 演習イベント ID | 演習イベント名 |
|-----------|---------|
| 3         | 課題要求    |

表 2 演習イベント DB の構造

| 受講者 ID   | 演習イベント ID | 演習イベント値 | 発生日時                |
|----------|-----------|---------|---------------------|
| 13117707 | 3         | 課題 1    | 2006-01-01 14:21:24 |

の操作を行うインタラクションを演習イベントとして送信する機構を組み込むことにより提案システムによって観測することができる。

演習イベントデータベース作成機能の構成を図 9 に示す。新たに演習イベントを定義、収集する場合、演習イベントデータベース作成機能は、演習イベント登録機能により教師が入力した演習イベント定義情報 { 演習イベント ID, 演習イベント名, 取得条件 } を取得し、演習イベント定義データベースに保存する。演習イベント定義データベースの構造を表 1 に示す。つぎに、観測対象の演習システム作成者が、演習イベント定義データベースから演習イベント定義を参照し、{ 受講者 ID, 登録された演習イベント ID, 発生日時, 演習イベント値 } を提案システムに対し送信する演習イベント送信機構を、演習システムの演習履歴観測部に組み込む。演習イベントデータベース作成機能は、演習イベント検出機能により演習システムから送信された演習イベントを収集し、演習イベントデータベースに保存する。演習イベントデータベースの構造を表 2 に示す。演習行動を演習イベントの系列により表すことにより、多数の受講者のさまざまな演習行動を容易に分析可能にする。また、アクティブデータベースにより演習状況の変化をデータベースに反映させることを可能にする。

### 2.3 演習状況監視ポリシー適用機能の実現法

提案システムでは、プログラミング演習クラスの視覚化の方針（演習状況監視ポリシー）を ECA ルールの系列として記述する。教師がポリシーを記述し提案システムに与えることで、提案システムはポリシーに従った演習状況の視覚化を行う。

演習状況監視ポリシーは、アイコン定義部と ECA ルール部からなる。アイコン定義部の構造を表 3 に示す。アイコン定義部には、設置するアイコンとアイコンに対して表示を割り当てる課題、デフォルトのアイコン色、アイコン内文字、クリック



表 3 演習状況監視ポリシー DB(アイコン定義部) の構造

| ポリシー ID | アイコン ID | 割当課題 | 色     | 文字   | リンク先    |
|---------|---------|------|-------|------|---------|
| 1       | 0       |      | white |      | ./個人の状況 |
| 1       | 1       | 課題 1 | white | 課題 1 | ./答案詳細  |

表 4 演習状況監視ポリシー DB(ECA ルール部) の構造

| ポリシー ID | ルール ID | イベント |    | コンディショ   | アクション |     |
|---------|--------|------|----|----------|-------|-----|
|         |        | イベント | 課題 |          | アイコン  | 値   |
| 1       | 1      | 3    | 1  | not blue | 1     | red |

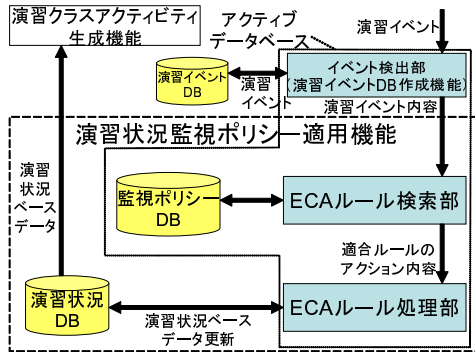


図 10 演習状況監視ポリシー機能の構成図

表 5 演習状況ベースデータの構造

| 受講者 ID   | アイコン ID | 色     | 文字  | リンク     |
|----------|---------|-------|-----|---------|
| 13117707 | 0       | white |     | ./個人の状況 |
| 13117707 | 1       | red   | 不正解 | ./答案詳細  |
| 13117800 | 0       | blue  |     | ./個人の状況 |
| 13117800 | 1       | blue  | 正解  | ./個人の状況 |

時のリンク先を記述する。ECA ルール部の構造を表 4 に示す。ECA ルール部には、演習イベント発生に対するアイコンの表示の変化のさせ方を ECA ルールで記述する。演習状況監視ポリシーにより、プログラミング演習クラスの視覚化の方針を記述することを可能にする。

演習状況監視ポリシー適用機能の構成を図 10 に示す。演習状況監視ポリシー適用機能は、演習イベントを検出するとポリシー DB 内の ECA ルールベースから演習状況監視ポリシーに記述された ECA ルールを検索する。検索した ECA ルールに従い、演習状況監視ポリシーごとに演習状況ベースデータを更新する。演習状況ベースデータには、表 5 に示すように演習クラスの各受講者のアイコンの文字や色などの描画内容を記録する。演習状況ベースデータは、演習状況監視ポリシーの登録時に監視ポリシーのアイコン定義部と演習 DB の受講者名簿を参照して生成する。教師が指定した演習状況監視ポリシーの演習状況ベースデータを用いて演習状況を描画することにより、教師の要求に合わせた演習状況の視覚化を可能にする。

#### 2.4 演習クラスアクティビティ生成機能の実現法

演習クラスアクティビティ生成機能の構成を図 11 に示す。演習クラスアクティビティ生成機能は、演習クラスアクティビティ選択機能により教師が選択した演習状況監視ポリシーの演習状

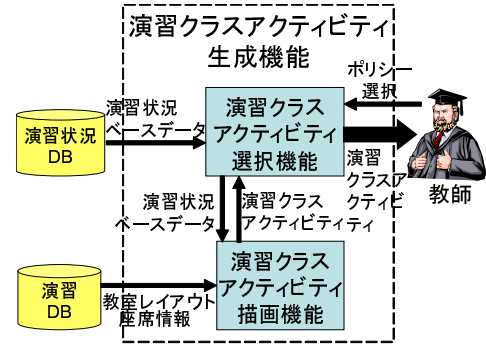


図 11 演習クラスアクティビティ生成機能の構成図

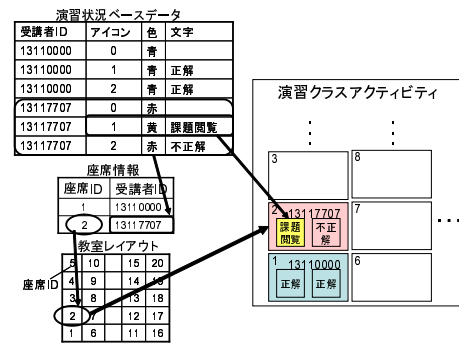


図 12 演習クラスアクティビティ描画機能の実現法

況データベースを取得し、演習クラスアクティビティ描画機能に渡す。演習クラスアクティビティ描画機能は、演習 DB から教室レイアウト、座席情報を取得する。取得した演習状況ベースデータと教室レイアウト、座席情報を用いて、演習クラスの状況を表す画面（演習クラスアクティビティ）を生成する。演習クラスアクティビティ描画機能の実現法を図 12 に示す。演習クラスアクティビティ描画機能は、教室レイアウトに従い演習状況表示スペース（図 7）の外枠を描画する。次に、座席情報を参照し、受講者が存在する演習状況表示スペースに対して、受講者 ID と受講者名を描画する。さらに、演習状況ベースデータを参照し、色と文字、クリック時のリンク先表示アクションを持つアイコンオブジェクトを描画する。以上の手順で演習クラスアクティビティを生成する。演習クラスアクティビティを教師に提示することにより、教師が進捗上問題のある受講者を直ちに発見し、その座席に指導に行くことを可能にする。

### 3. プロトタイプシステム

本章では、提案システムのプロトタイプの実装と実際のプログラミング演習への適用について述べる。

#### 3.1 実装

本プロトタイプシステムでは演習イベントデータベース作成機能、演習状況監視ポリシー適用機能、演習クラスアクティビティ機能を実装した。演習イベントデータベース作成機能の演習イベント定義機能は教師が入力しやすいインターフェースとして Web ページからの登録方法を用意した。観測可能な演習イベントは、演習支援システムにおける、ログイン、課題要求、答案送信、正解メッセージ取得、不正解メッセージ取得の 5 つ

である。演習状況監視ポリシー登録処理は教師が入力しやすいインタフェースとして Web ページからの登録方法を用意した。プロトタイプシステムの演習状況監視ポリシー設定ページを図 13 に示す。

演習状況監視ポリシーの記述方法として、次のような複数の課題に対する進捗状況を視覚化する演習状況監視ポリシーをプロトタイプシステムに与える場合を例に挙げて述べる。

(1) アイコン 1 からアイコン 4 に対してそれぞれ課題 1 から課題 4 を割り当てる。

(2) アイコン色は課題閲覧イベント発生で黄色、不正解イベント発生で赤、正解イベント発生で青にする。

(3) イベント発生無し、または課題閲覧イベントのみならば背景を赤、課題に 1 問でも不正解したら背景黄、課題に 1 問以上正解したら背景を白にする。

(4) 必須課題 (課題 1, 2, 3) に全問正解したら背景を青にする。

(1) に基づいて、図 13-(a) の演習状況監視ポリシーのアイコン定義部にアイコンの色、アイコンに割り当てる課題またはイベント、アイコン内の文字、アイコンをクリックした場合のリンク先を設定する。図 13-(b) の条件設定部には、ECA ルールで用いるコンディションを設定する。設定できる条件は、演習イベントの発生の有無とアイコンの色である (2) (3) (4) に基づいて、図 13-(c) の ECA ルール部には、発生した演習イベントに対するアクションとしてアイコンの色と文字とリンク先の変化を設定する。コンディションには、条件定義部で設定したコンディションの論理積または論理和で設定する。

上記の演習状況監視ポリシーに基づき生成された演習クラスアクティビティを図 14 に示す。図 14 では (1) の指定に従い各課題に対する対応したアイコンが表示されている。また (2) の指定に従い各課題に対応するアイコンが不正解をした場合は赤、正解した場合は青に変化している。このように受講者の各課題に対する取り組み状況がアイコンの色の变化として表示されるので、教師はアイコンの色の变化に注目することにより課題に対して進捗が遅れている受講者を発見することが可能になる。また (3) の指定に従い課題に1問も答案を送信していない受講者の背景は赤色に変化している。また (4) の指定に従い必須課題に全て正解した受講者の背景は青色に変化している。このように受講者の複数の課題に対する進捗状況が背景の色の变化として表示されるので、教師は背景の色の变化に注目することにより複数の課題に対して進捗が遅れている受講者を発見することが可能になる。以上により、プロトタイプシステムが教師が入力した演習状況監視ポリシーどおりに演習状況を視覚化可能であることを確認した。

本プロトタイプシステムは Java 言語を用いて実装した。また、Web ページを作成するために Tomcat を利用した。演習イベント定義データベースと演習イベントデータベース、演習状況監視ポリシーデータベース、演習状況データベースをリレーショナルデータベース管理システム MySQL を用いて実現した。

### 3.2 プログラミング演習への適用

本プロトタイプシステムを名古屋工業大学の 2006 年度 C プ

#### 演習状況監視ポリシー詳細登録

(a) 1. 背景(アイコンID)の設定

アイコン名:初期色 初期文字 リンク先

red http://hidekiss.eicom.nitech 設定

2. アイコンの設定

| 削除                       | アイコンID | アイコン名 | 割当情報                          | 割当情報ID | 初期色   | 初期文字 | リンク先                          |    |
|--------------------------|--------|-------|-------------------------------|--------|-------|------|-------------------------------|----|
| <input type="checkbox"/> | 1      | 課題1   | <input type="radio"/> イベントの課題 | 1      | white |      | http://cyabin.eicom.nitech.jp | 設定 |
| <input type="checkbox"/> | 2      | 課題2   | <input type="radio"/> イベントの課題 | 2      | white |      | http://cyabin.eicom.nitech.jp | 設定 |
| <input type="checkbox"/> | 3      | 課題3   | <input type="radio"/> イベントの課題 | 3      | white |      | http://cyabin.eicom.nitech.jp | 設定 |
| <input type="checkbox"/> | 4      | 課題4   | <input type="radio"/> イベントの課題 | 4      | white |      | http://cyabin.eicom.nitech.jp | 設定 |

アイコン新規登録

アイコンID: アイコン名: 割当情報: 割当情報ID: 初期色: 初期文字: リンク先:

イベントの課題 設定

(b) 3. 条件の設定

| 削除                       | 条件ID | 属性                            | イベント(アイコン)ID | 条件の値  |    |
|--------------------------|------|-------------------------------|--------------|-------|----|
| <input type="checkbox"/> | 1    | <input type="radio"/> イベントの存在 | アイコン色 1      | blue  | 設定 |
| <input type="checkbox"/> | 2    | <input type="radio"/> イベントの存在 | アイコン色 2      | blue  | 設定 |
| <input type="checkbox"/> | 3    | <input type="radio"/> イベントの存在 | アイコン色 0      | blue  | 設定 |
| <input type="checkbox"/> | 4    | <input type="radio"/> イベントの存在 | アイコン色 4      | blue  | 設定 |
| <input type="checkbox"/> | 11   | <input type="radio"/> イベントの存在 | アイコン色 1      | red   | 設定 |
| <input type="checkbox"/> | 12   | <input type="radio"/> イベントの存在 | アイコン色 2      | red   | 設定 |
| <input type="checkbox"/> | 13   | <input type="radio"/> イベントの存在 | アイコン色 0      | red   | 設定 |
| <input type="checkbox"/> | 14   | <input type="radio"/> イベントの存在 | アイコン色 4      | red   | 設定 |
| <input type="checkbox"/> | 21   | <input type="radio"/> イベントの存在 | アイコン色 0      | blue  | 設定 |
| <input type="checkbox"/> | 22   | <input type="radio"/> イベントの存在 | アイコン色 0      | red   | 設定 |
| <input type="checkbox"/> | 23   | <input type="radio"/> イベントの存在 | アイコン色 0      | white | 設定 |

(c) 4.ECAルール設定

| 削除                       | ルールID | ルール説明                  | 外部イベント                         | 外部イベント属性                       | 外部イベントID | 条件        | アクション                     | 対象アイコンID | 種類  | 値      |    |
|--------------------------|-------|------------------------|--------------------------------|--------------------------------|----------|-----------|---------------------------|----------|-----|--------|----|
| <input type="checkbox"/> | 1     | 正解..アイコン1青             | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 4        | 1         | <input type="radio"/> 色変化 | 1        | 色変化 | blue   | 設定 |
| <input type="checkbox"/> | 2     | 正解..アイコン1青             | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 4        | 2         | <input type="radio"/> 色変化 | 2        | 色変化 | blue   | 設定 |
| <input type="checkbox"/> | 3     | 正解..アイコン1青             | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 4        | 3         | <input type="radio"/> 色変化 | 3        | 色変化 | blue   | 設定 |
| <input type="checkbox"/> | 4     | 正解..アイコン4青             | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 4        | 4         | <input type="radio"/> 色変化 | 4        | 色変化 | blue   | 設定 |
| <input type="checkbox"/> | 5     | 不正解..青でなければ..アイコン1赤    | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 5        | 11        | <input type="radio"/> 色変化 | 1        | 色変化 | red    | 設定 |
| <input type="checkbox"/> | 6     | 不正解..青でなければ..アイコン2赤    | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 5        | 12        | <input type="radio"/> 色変化 | 2        | 色変化 | red    | 設定 |
| <input type="checkbox"/> | 7     | 不正解..青でなければ..アイコン3赤    | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 5        | 13        | <input type="radio"/> 色変化 | 3        | 色変化 | red    | 設定 |
| <input type="checkbox"/> | 8     | 不正解..青でなければ..アイコン4赤    | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 5        | 14        | <input type="radio"/> 色変化 | 4        | 色変化 | red    | 設定 |
| <input type="checkbox"/> | 13    | 正解..背景黄でなければ..背景白      | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 4        | 121       | <input type="radio"/> 色変化 | 0        | 色変化 | white  | 設定 |
| <input type="checkbox"/> | 19    | 答案送信..解答が赤ならば..背景黄色    | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 3        | 22        | <input type="radio"/> 色変化 | 0        | 色変化 | yellow | 設定 |
| <input type="checkbox"/> | 21    | 課題解答..黄と青でなければ..アイコン1黄 | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 2        | 11 11 111 | <input type="radio"/> 色変化 | 1        | 色変化 | yellow | 設定 |
| <input type="checkbox"/> | 22    | 課題解答..黄と青でなければ..アイコン2黄 | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 2        | 12 11 112 | <input type="radio"/> 色変化 | 2        | 色変化 | yellow | 設定 |
| <input type="checkbox"/> | 23    | 課題解答..黄と青でなければ..アイコン3黄 | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 2        | 13 11 113 | <input type="radio"/> 色変化 | 3        | 色変化 | yellow | 設定 |
| <input type="checkbox"/> |       | 課題解答..黄と青でなければ..アイコン4黄 | <input type="radio"/> 演習イベント発生 | <input type="radio"/> アイコンクリック | 2        | 14 11 114 | <input type="radio"/> 色変化 |          | 色変化 | yellow | 設定 |

図 13 プロトタイプシステムによる演習状況監視ポリシーの設定

プログラミング演習 (4 クラス, 受講者数計 239 名) に約 1ヶ月間適用し、負荷集中時にも正常に動作していることを確認した。本プロトタイプシステムを利用した教師, TA は 14 名である。

演習室レイアウトは名古屋工業大学の 3ヶ所の演習室 (定員はそれぞれ約 60 名) に対応した。演習室の計算機は、対称型マルチプロセッサ型のホスト計算機に 207 台の Linux 端末が接続された構成の共通ワークステーションである。

## 4. 提案システムの評価と考察

本章では、前節のプロトタイプシステムを用いて提案システムの妥当性を検証した結果について述べる。

### 4.1 演習状況監視ポリシーの効果

演習状況監視ポリシーを用いることで、教師が要求した演習状況の視覚化を容易に設定可能であることを示す。また、異なる視覚化への変更を容易に行うことが可能であることを示す。

評価-1 複数の課題に対しての進捗状況を表す視覚化を行う場合、以下の 2 つを比較する。



図 14 プロトタイプシステムにより生成した演習クラスアクティビティ

比較対象 1-1 プログラムで直接書いた場合の行数  
 比較対象 1-2 監視ポリシーを用いた場合の ECA ルール数の数  
 評価-2 複数の課題に対しての進捗状況を表す視覚化から、ある 1 つの課題に対する進捗状況を表す視覚化に変更するとき、以下の 2 つを比較する。

比較対象 2-1 プログラムで直接書いた場合の変更する行数  
 比較対象 2-2 監視ポリシーを用いた場合に変更する ECA ルール数の数

演習状況の視覚化には次の 2 機能が必要である。  
 機能-1 演習支援システムの演習履歴を解析し描画に必要なデータを取得する機能  
 機能-2 描画データに基づいて演習クラスアクティビティを描画する機能

評価-1 の演習状況視覚化に必要なプログラム行数と ECA ルール数を表 6 に示す。表 6 より、演習状況監視ポリシーを用いることで約 25 分の 1 の記述量で視覚化を行うことが可能であることがわかる。評価-2 の演習状況視覚化に必要なプログラム行数と ECA ルール数を表 7 に示す。機能-1 では、異なる演習履歴データを用いるためにプログラムの変更が必要であった。機能-2 では、描画の色や表示する文字、リンク先などが異なるためプログラムの変更が必要であった。表 7 より、必要な機能全体では演習状況監視ポリシーを用いることで約 20 分の 1 の記述量で視覚化の変更が可能であることがわかる。また、評価-2 の視覚化から機能-1 の描画データを変更せずに機能-2 の描画内容のみを変更する場合には、変更する ECA ルール数は 16 と変わらないが、ECA ルール内での変更箇所は描画内容を設定するアクション部のみで視覚化可能であった。このように、演習状況監視ポリシーを用いて視覚化の変更を行う場合には ECA ルールの変更が局所化されているため、記述量の差以上に変更が容易であるといえる。以上により、演習状況監視ポリシーを用いることで教師の要求する演習状況の視覚化を容易に設定、変更可能であることを明らかにした。

#### 4.2 演習クラスアクティビティ生成機能の妥当性

演習状況と座席位置を結び付けて表すことにより、教師が受

表 6 演習状況視覚化に必要なプログラム行数と ECA ルール数

| 必要機能 | プログラム行数 | ECA ルール数 |
|------|---------|----------|
| 機能-1 | 233     | 26       |
| 機能-2 | 425     |          |
| 機能合計 | 648     |          |

表 7 異なる演習状況視覚化に変更する場合のプログラム変更行数と ECA ルール変更数

| 必要機能 | プログラム変更行数 | ECA ルール変更数 |
|------|-----------|------------|
| 機能-1 | 108       | 16         |
| 機能-2 | 221       |            |
| 機能合計 | 329       |            |

講者相互の座席の位置関係を考慮しながら受講者の進捗状況を分析することを容易にすることを示す。

評価のためのモデルケースとして、教師が正解していない受講者が多く固まっている座席空間に指導に行くまでの手順を考える。ここで手順とは、システムの画面を切り替える動作とする。この場合教師は以下の手順-1、手順-2 を受講者数  $i$  人分繰り返す。そして正解していない受講者が多く固まっている受講者の固まりを特定する。

手順-1 受講者 1 人分の座席を調査する

手順-2 受講者 1 人分の履歴を調査する

以上のモデルケースにおいて、指導に行くべき正解していない受講者の固まりを決定するまでに要した手順数を以下の 3 つの手法で比較する。

手法-1 演習支援システムの履歴表示のみ (座席情報無し)

手法-2 演習支援システムの履歴表示及び、座席情報

手法-3 提案システムの演習クラスアクティビティ

手順数  $S$  を数式表すと以下のようなになる。 $i$  は受講者数である。ここで、手法-1、手法-2 では受講者の座席と履歴を調査した後、その 2 つの情報をメモを取るなどして結びつける必要があるが、ここではその作業については考慮していない。

手法-1 の手順数  $S = 2i$

手法-2 の手順数  $S = i$

手法-3 の手順数  $S = 1$

各手法に対する受講者数と手順数の関係を図 15 に示す。図 15 より、手法-1、手法-2 は、演習クラスの受講者数に比例して手順数が増加していることがわかる。そのため、受講者が多くなるほど教師が正解していない受講者の固まりを調査することは大きな負荷になると言える。一方、手法-3 の提案システムを用いた場合には、受講者の数に関わらず 1 度で指導に行くべき受講者の固まりを決定できる。これにより、教師が提案システムを用いることで教師が受講者の演習状況と受講者相互の座席の位置関係を容易に把握することが可能であることを明らかにした。

#### 4.3 提案システムの効果と妥当性

提案システムの効果と提案手法の妥当性を調査するため、実際の演習の教師と TA に対し 5 段階項目 (1. 非常によい, 2. よい, 3. どちらでもない, 4. 悪い, 5. 非常に悪い) 選択式アンケートを行う。アンケート項目を以下に示す。(1) 提案システ



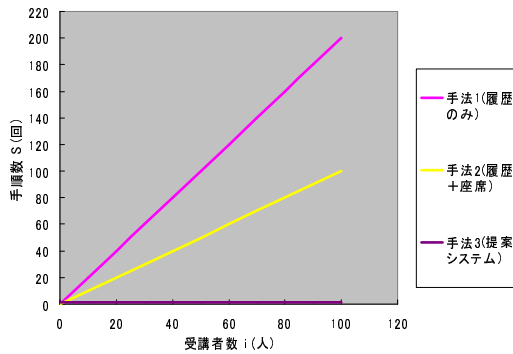


図 15 受講者の固まり発見の各手法に対する受講者数と手順数の関係

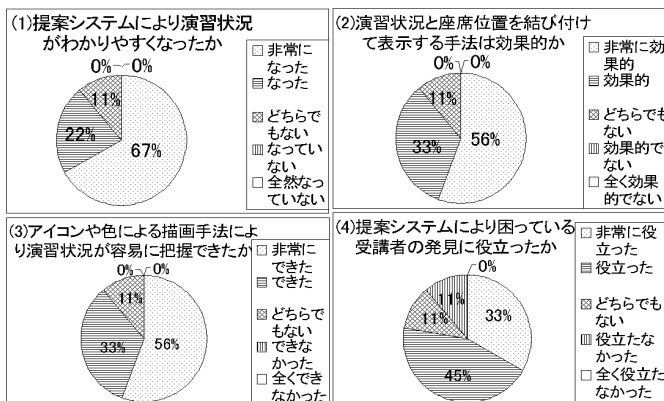


図 16 提案システムの効果と妥当性に関するアンケートの集計結果

ムにより演習状況がわかりやすくなったか。(2) 演習状況と座席位置を結び付けて表示する手法は効果的か。(3) アイコンや色による描画手法により演習状況が容易に把握できたか。(4) 提案システムにより困っている受講者の発見に役立ったか。

提案システムのプロトタイプを適用した名古屋工業大学の4つの演習クラスの教師, TA 合計 14 名に対しアンケートを要請し, うち 9 名から回答を得た。アンケートの集計結果を図 16 に示す。(1) ~ (3) の集計結果より, 8 割以上の教師, TA から提案システムにより演習状況がわかりやすくなった, 演習状況を結び付けて表示する手法は効果的である, アイコンや色による描画手法により演習状況が容易に把握できたという回答を得た。また, (4) の集計結果より, 7 割以上の教師, TA から提案システムにより困っている受講者の発見に役立ったという回答を得た。以上により, 実際の演習における提案システムの効果と妥当性を明らかにした。

## 5. 関連研究

文献 [4] は, アクティブデータベースを e ラーニングシステムに応用している。文献 [4] の提案するシステムは, ECA ルールを利用することにより正誤パタンの把握とリアルタイムな教材の提示順を制御 (シーケンシング) を実現している。提案システムは, アクティブデータベースを用いることにより教師が時間経過による受講者の演習行動の変化を追いかけ, 受講者の演習行動に応じた演習状況の視覚化を可能にしている。

文献 [5], [6] は, 演習形式の講義における学習状況把握のため

のシステムについて述べている。文献 [5] は, WebCT の CALL 教材システムにおける, 学習者の学習進捗率, 学習時間, 学習者間の学習履歴の相関関係をグラフ化して視覚的に把握できるシステムを提案している。文献 [5] は, 学習進捗率の視覚化の手法としてグラフを用いている。しかし, 演習形態の授業において教師が受講者の指導や助言を行う場合, 受講者の進捗状況に加えて受講者の位置を把握する必要がある。提案システムは, 受講者の進捗状況と座席位置を結びつけることにより直ちに進捗上問題のある受講者を発見し, その位置に指導に行くことを可能にする。文献 [6] は, 教師はプログラムの動作理解時とコーディング時において, それぞれ発生している学習者が抱える問題の傾向をリアルタイムに把握することを可能とするシステムを提案している。ここで文献 [6] は, プログラムアニメーションシステムに操作履歴収集機能を組み込むことで受講者の問題の傾向の分析を行っている。提案システムは, 演習システムに演習イベントを送信する機構を組み込むことで, 受講者の演習状況の分析を可能にする。また, 文献 [6] は共通問題を抱える学習者の推定を行っているが, 指導の為の学習者の位置特定までは行っていない。提案システムは, プログラミング演習クラスの状態を座席位置と結びつけて視覚化することで, 教師が受講者の演習状況と受講者相互の座席の位置関係を把握することを可能にする。

## 6. おわりに

本稿では, アクティブデータベースを用いたプログラミング演習クラスの視覚化システムの実現法を提案した。また, 提案システムのプロトタイプを実装し, 実際の講義に適用した。そして, 提案システムの妥当性を示した。現在, タイマイベントによる時間経過に関する演習状況の視覚化の設計を進めている。これにより, 教師が長時間アクションを行っていない受講者を発見することを可能にする。また, 他システムとの定量的な比較, 評価を行う予定である。

## 文献

- [1] 中島秀樹, 高橋直久, 細川宜秀, プログラミング学習のための QA サイクル-受講者の習得度に応じた問題自動提示メカニズム-, 電子情報通信学会論文誌, VOL.J88-D-I, NO.2, pp.439-450, 2005.
- [2] 宮地恵佑, 高橋直久, 構造誤り検出機能を有するアセンブラプログラミング演習支援システム, DICOMO2006 シンポジウム論文集, pp.521-524, 2006.
- [3] 石川博, アクティブデータベース, 情報処理学会誌, 35-2, pp.120-129, 1994.
- [4] 延原 哲也, 庄司 成臣, 劉 渤江, 横田 一正, ECA ルールを活用した e ラーニングシステムにおけるシーケンシング制御の改善, 情報処理学会研究報告, 2005-DBS-137(2)-(80), Vol.2005, No.68(20050714) pp. 601-606, 2005.
- [5] 安浪 誠祐, 高橋 幸, 河津 秀利, LMS を活用した CALL 教材学習支援, <http://www.webct.jp/>, 日本 WebCT ユーザー会第 3 回 WebCT 研究会, 2005.
- [6] 倉澤邦美, 鈴木恵介, 飯島正也, 横山節雄, 宮寺庸造, プログラミング演習における一斉指導のための学習状況把握支援システムの開発, 電子情報通信学会技術研究報告, Vol.104, No.703, pp.19-24, 2004.