フィジカルデバイスで体感するプログラミング教育

Physical Device for Visible Programming Education

椋田 實[*1]，片山 茂友[*1]
Minoru MUKUDA[*1], Shigetomo KATAYAMA[*1]
[*1] 日本工業大学情報工学科
[*1]Computer and Information Engineering, Nippon Institute of Technology
Email: mukuda@nit.ac.jp

あらまし:プログラムと物理的な動きを関連させることで手続きの論理を考えさせる. このような視点で、指訓練装置とパソコンによる体感型プログラミング教材を開発した．この教育方法は，指訓練装置による「物理的な動き」やディスプレイによる「絵の動き」を体感することで、プログラムの問題点を発見し、その解決方法を考え，具体化することを学ぶ．この教材を使用した授業の実施経過について報告する．

キーワード：**プログラミング**教育，フィジカルデバイス，プログラミング教材，体感的プログラミング

1. はじめに

プログラミングに物理的な動きを関連させることで「手続きの論理的な正しさを考えさせる」ために，学習者自身が体感できる環境を提供する．このような視点で，指訓練装置とパソコンによる体感型プログラミング教材を開発した[1].

この教材を使用した授業の対象者として、少し遅れている学習者に適用した．目標は自力で問題解決が出来るようにすることである．この教育方法は，指訓練装置による「物理的な動き」やディスプレイによる「絵の動き」を体感することで、プログラムの問題点を発見し、その解決方法を考え，具体化することを学ぶ．この教材を使用した授業の実施経過について報告する．なお, 実施期間は 2010 春学期から 2012 年春学期である．

2. 体感型プログラミング教育

プログラミング教育は論理的な内容であり、頭に中での思考が中心となる．プログラムの正しさの検証と確認は、あるインプットに対して想定されたアウトプットが生成されるかを判定することで検証する．我々はインプットとアウトプットをフィジカルデバイスにすることで、学習者が体感できる環境にした．本来のプログラムはフィジカルデバイスを操作することでアプリケーションの目的を達成させるが、この教育ではフィジカルデバイスの動きからプログラムの不備を知り、プログラムを修正する．

また、実世界はアナログ的であることが普通であり、判定には絶対的ではなく相対的な基準が要求される．ハードウェアを含むシステムのプログラミングにおいて，ある判定に判断範囲や相対的な閾値，時間による変動などが伴うことがある．たとえば，圧力の AD 変換と速度変化の測定において，経験の差やサンプリング間隔，機器の個体差（電子部品のバラツキ）を考慮することが必要になる．このような問題を想定したプログラム設計やプログラミングが要求される．この特徴をプログラミングの学習に応用することでプログラミング教育を効果的に進めることを考えた．これらの能力や目標達成状態が学習者自身で把握できることを目指した学習教材を考える．

この体感型プログラミング教育は学習者自らが問題点を見いだし，解決をすることを目指している．学習者（疑似訓練者）は図 1 のように指訓練システムの実行結果（プログラミングの結果）を自身で体感する．

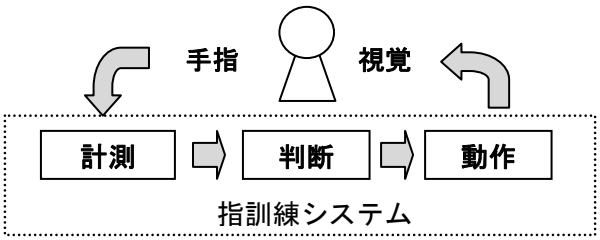

図 1 学習者とプログラムの試行

◎指訓練システムの開発

この学習方法ではハードウェア機器を通してプログラミングの問題点を知り, 問題解決の方法を学ぶ. ハードウェアを含むシステムの開発は次のような特徴がある.

- ✓ 実世界と仮想世界の差異（非論理的）
- ✓ 機器の個体差と対処方法（相対量）

✓ アナログ量のデジタル化（サンプリング）
✓ 指訓練装置（図１）とインターフェース
✓ 実時間処理（インターバル・タイマ）
✓ センサーと制御（測定とフィードバックループ）
✓ 人のインターフェース（感覚は非線形）
✓ 実験と結果（分析，ルールの選出，抽象化）

指訓練装置は図 2 のようにバルーン（10bits×2），回転円盤（2bits×2），押しボタン（1bit×8），LED ランプ（1bit×8）がある．バルーンは圧力を AD 変換してデジタル値で入力する．回転円盤は回転方向で bit の ON/OFF となる順が逆になる．

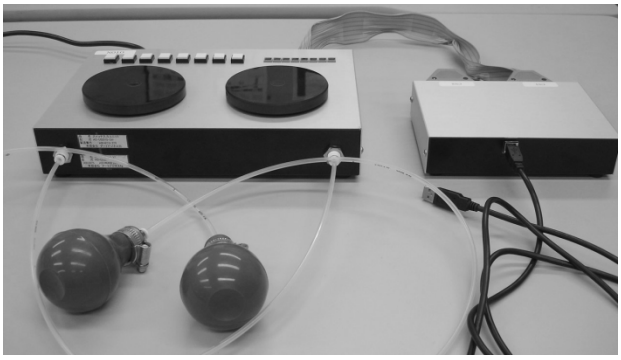

図２ プログラム演習用の指訓練装置

機器の制御はマイコンボード USBM3069-HS(L)を使用している[3]．学習者には，①時間的な要素，②アナログ的な変化量，③ハードウェアの許容範囲などを実体験で把握させる．

3. 教材の構成と学び方

ソフトウェア設計開発演習はプログラミングの基礎を学んだ人を対象として，設計から開発までを演習形式で学ぶ授業である．本テーマの授業は春・秋の２期から構成されている．このテーマの春学期は基礎的なリアルタイム・プログラミングを学びながら各部品に対応する技術的な技法，Windows アプリケーションの基礎を学ぶ．アプリケーションは C++ と MFC、ダイアログベースで開発するが、オブジェクト指向的な機能（クラス継承やオブジェクトの生成）は使わない様に構成した．

秋学期のソフトウェア設計開発演習はハードウェアの操作と「問題を把握して，解決方法を考え，具体化し，作成し，目標を基準として評価する」を学習する．特に，頭で論理的に構成した設計図と実際のアプリケーションシステム（例：円盤の回転方向を判断することやバルーンの圧力を測る機能など）の違いの体験を通して学ぶ．

4. 授業の実施と結果

ソフトウェア設計開発演習には４つのテーマがある．2010 年度の受講生は 76 名で，本稿のテーマを選択したのは 22 名，秋学期は 12 名した．2011 年度春学期の受講生は 17 名、2012 年度春学期は 29 名でした．2010 年､2011 年は学習進度と教材の難易度調整，理解状況の把握を行った．改善点として、2012 年度は春学期にハードを使わないアプリケーションの開発課題を用意した．なお、2012 年度からは大別して２つのテーマ（本教材での開発と PBL による実務システムの開発）で演習を実施している．

◎ 課題の難易度を調査

教材の理解と妥当性を調べるためにアンケート調査を 2011 年度春学期に行いました．最初の質問は「難しい課題を複数選択で選びなさい」で，回答を得た後,「難しいと感じた学習項目は何か」を調べた．なお，アンケートは記述式で，これらの文章を集計した．

全般的に難しいのがタイマーイベントの使い方である．また､イベントによるプログラム構造である．大きな原因はタイマーイベントの意味が理解されていない事である．次のカベは図形操作と三角関数である．また，図形を動かす方法で,「少しづつ移動した絵を描画する」ことで実現するアニメーションが解らないと感じている．これは，単に絵を描きかえるのであるが，図形を描く場合，背景を書き変えることに気づかないこともある．

具体的な内容では､「秒針を秒単位に動かす」､「多角形で葉を描く」､「構造体で魚等を管理する」､「デバイスコンテキストを作成する」が難しいと感じている．想定外は「math.h ファイルの機能が解らない」であった．

5. おわりに

学習者が「自身で解決できる」ことを目標にしたプログラミングの教育方法で使用する指訓練装置とパソコンによる体感型プログラミング教材を開発した．2 年 6 カ月の運用により,「学習者自身で解決できる状況」が見えてきたが，基本的な実時間処理や一般知識の不足が見え隠れする．この点を考慮した教材と授業の進め方をどの程度まで実施するかが課題である．

参考文献

[1] 椋田，片山：ハードウェアのリアル性を活用したプログラミング教育，IEICE，教育工学研究会 ET222, 2011.
[2] USBM3069-HS/USBM3069-HSL ユーザーズマニュアル，テクノウェーブ株式会社，2009