

# 動機付けとプログラミング能力向上を目指したコースプランの改善

## Refinement of Course Plan Aiming at Motivating Learners and Improving Their Understanding

島川博光 横田裕介 徐 剛 山口秀樹 脇田 航  
 榊原一紀 陳 延偉 泉 朋子 野口 拓 黄 宏軒  
 立命館大学情報理工学部

**Abstract:** It is always a hard task for teachers to design courses of compulsory subjects. The authors study a method to design a course plan for programming education which motivates students and improves their understanding. The method investigates student requirements with the contextual inquiry method. It confirms the effectiveness of a course plan teachers design to satisfy the requirements. It uses student learning behavior logs left on a Web-based course supporting tool. This paper explains one example designed based on the method. The example course plan consists of interesting assignments and cordial supervision. A graphical programming library is presented to students to make them engage in the former, while the latter is realized with interactive checks according to pre-announced check items. Using student learning behavior logs, this paper indicates what aspects of students are improved with the course plan, as well as illustrates how much the improvement is. It also discusses a method to encourage students to work on programming.

**Keywords:** programming course plan, motivation, contextual inquiry, learning behavior, web intelligence, interactive check

### 1. はじめに

情報教育を専門とする大学ではプログラミング教育コースは全学生が受講する必修科目であり、全員がそのコースにて必要とされる最低限の能力や技量を身につけなければならない。受講生を動機付け、理解を定着させることが教員に要求される。受講生を動機付けるために、ゲーム作成<sup>[1]</sup>やペア・プログラミング<sup>[2]</sup>など、いろいろな方法が試みられているが、大学ごとに受講者の気質や教育条件は異なるので、ある大学で成功した特定の方法がすべての大学で成功する保証はない。各々の大学に適した方法を見つけるべきである。

より多くの学生の理解を深めるために、教員は、そのコース設計に腐心する。しかしプログラミング教育にかかわらず、多数の受講生を対象とする必須科目でのコース設計は難しい。教員が学習者の潜在的な要求を理解するために、受講生へのアンケートがとられてきたが、大きく功を奏することはなかった。一般にコースに対して満足している受講生はアンケートで賛意を示さないが、コースでの勉強が辛いと感じている受講生はアンケートで酷評を示すことが多い。時間と労力を強要するプログラミングについての意見を、受講中の受講

者に問えば当然、批判的意見が大勢を占める。結局、コースを安易にするか、逆に学習者の意見を無視して厳しく指導するかのどちらかになる。

本論文では、受講者からインタビューした結果と、プログラミング演習サーバの学習者の使用ログを使って、受講者に適したコースを設計した実例を報告する。我々は、受講生の要求を獲得するために、Contextual Inquiry法<sup>[3]</sup>に基づくインタビューを受講生に実施した。報告するコースでは、受講生の興味を引く課題を作った。インタビュー結果から得られた要求に従い、手厚く指導することとは何かを分析し、それを実施した。このコースにより、受講生がどのように動機付けられ、理解を深めたかを、サーバの学習者の使用ログから導出した実データを使って示す。

### 2. ふるまいデータに基づくコースプラン

#### (1) Contextual Inquiry法

コース案 (course plan, 以下CP) への学習者の要求を獲得しようとしても、アンケートでは教員があらかじめ想定する問題への意見を調査することしかできない。我々はContextual Inquiry法<sup>[3]</sup>を用い、動機を得る、もしくは、失う事例を受講生にインタビューした。事例を分析し受講生の隠れた要求を同定する。実際に起った事例を分析するので、教員が想定していない要求の獲得をこの方法は可能とする。

Hiromitsu Shimakawa\*, Yusuke Yokota, Gang Xu, Hideki Yamaguchi, Wataru Wakita, Kazuki Sakakibara, Yen-Wei Chen, Tomoko Izumi, Taku Noguchi and Hung-hsuan Huang Ritsumeikan University  
 \*E-mail:simakawa@cs.ritsumeik.ac.jp

## (2) プログラミングコース案の洗練

受講者を動機づけ、理解力を向上させるコースを、限られた教育資源の中でどのように実現するかは、教員の力量に委ねられている。それぞれの大学の教育条件にあったコースを設計する一般的な方法が望まれる。

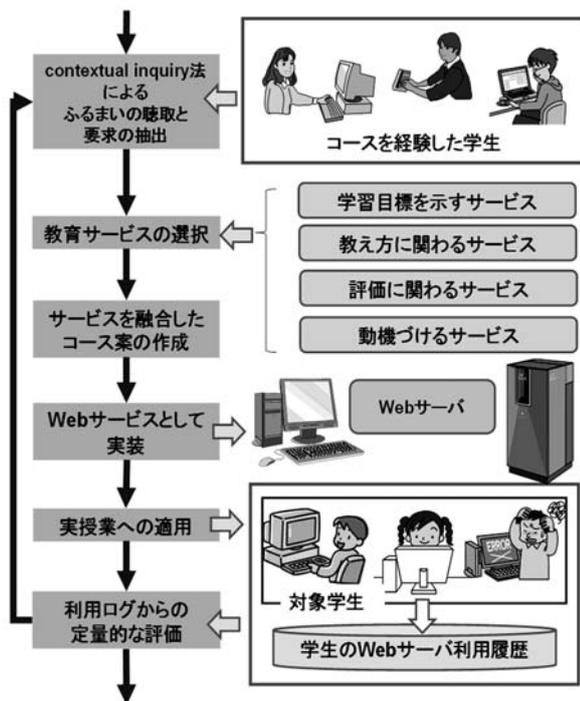


図1 コース案の洗練

文献<sup>1)</sup>が示すように、すでに多くの教育サービスが提案されている。我々は教育サービスを組み合わせ、図1に示すようなCPを洗練しながら構築する手法<sup>1)</sup>を別に研究している。この手法では、インタビューを使って学生を理解し、彼らの指導に適したサービスを選択しWebコースウェアとしてCPを実現する。CPを実授業に適用し、学習者のWebサービスの利用ログからCPが期待通りに働いているかを評価する。この過程を繰り返し、CPを洗練する。たとえば個別の教え方に関わるサービスでも、指導者が受講生に向きあう指導や、提出物に指導者がコメントを追記するサービスがある。前者は臨機応変な指導が可能であるが指導コストがかかる。後者は柔軟性にかけるが、学生への評価返却は速くなり、コストも抑えられる。サービス選択時にはこのような得失の平衡を考慮する。ログに記録された、授業以外でのサーバ利用時間から自習時間が推定できる。指導者の利用時間から指導にかかるコストがわかる。選択されたサービスの組が正しい平衡を作っているかを自習時間や指導コストより検証する。

## (3) 動機付けの要因

我々は、立命館大学情報理工学部でのプログラミング演習を終えて半年程度経過した学習者約80名を二人一組とし、プログラミングにやる気が出るときとはどういうときか、また、そのための演習環境の改善点について互いにインタビューさせた。コースを修了して時間をおいた受講生を対象としたのは、受講中の学生への調査結果では、批判的意見が支配的になると考えたからである。インタビュー結果を分析すると、成績の上位、下位にかかわらず、以前に学習したロボット制御プログラミング課題を受講生は称賛していた。この課題は、自動車をかたどった、モータ付き模型をC言語プログラムで制御して、決められた道程の走行時間を競うもので、難易度はかなり高い厳しい課題である。しかし、いかなる理解レベルの受講者も、このように興味がわく課題に取り組みたいと述べていた。

単に厳しい試練を与えるだけでは勉学は長続きしない。試練を克服するための支援を与えることは不可欠である。インタビュー結果を調べたところ、受講者はすべて、疑問が湧いたときはいつでも指導者に質問できることや、提出した課題をすぐに採点してくれることを望んでいた。逆に、何を質問すべきかわからないときや、指導者の準備不足が感じられると、受講生はやる気を大きく減退させる。指導者からの手厚い指導を受講生は求めていると我々は捉えた。指導者が受講者の提出物を迅速に評価してフィードバックし、受講者からの疑問にはいつでも答え、受講者が行き詰まっているときには指導者側から手助けするような体制を作る必要がある。

以上のようにインタビュー結果を分析し、我々は、立命館大学情報理工学部での受講者を動機付ける要因は、興味深い課題と手厚い指導であると判断した。必須科目ゆえに配備された多くのTA( Teaching Assistant)が標準的な指導を提供できる仕組みを整え、これと興味深い課題を連携させて、受講者を動機づけ、理解力を向上させるCPを設計した。

## (4) コースツール

上述のCPの有効性を検証するため、実際のプログラミング演習の授業にCPを適用し、その際の学習経過を観察した結果を分析する。分析のために、我々はWebサーバ上にコースツールを開発した。Webに基づくコースツールを採用したのは以下の理由による。

- 授業の時間以外にもプログラミングに取り組

め、課題の提出ができる環境を受講生に提供する。

- 課題に対し提出されたソースコードを、指導者がいつでも評価できるようにし、受講生に早くフィードバックを返すことを可能にする。
- 多くの受講者を指導するために、複数の教員とTAが必要となる。指導者の間で提出物の採点基準が異なることは避けるべきである。Webサーバには、要求された事項が実現できているかどうかをYes/Noの二者択一で採点できるチェック項目を掲載する。これにより、最低限の指導基準の標準化を図る。
- 受講生の学習上のふるまいをサーバの利用履歴として収集する。提出物への採点結果だけでなく、どの問題から解こうとしたか、どのくらい長くサーバを使用していたかなどの、学習の進め方を示す、受講生のふるまいを収集する。

最後の項目は、Webコースウェアを実授業に適用し、その成果を学習者のサーバ利用ログから評価するうえで重要である。アンケートでのCP評価は受講者の恣意に依存するが、サーバ利用ログは、受講者のふるまいをそのまま表し、正当な評価が可能となる。

### 3. 興味深い課題と手厚い指導の連携

#### (1) 図形描画課題

我々はプログラミング初心者の受講生の興味をそそる課題として図形描画課題を用意した。あらかじめ用意されたライブラリ関数をプログラムから呼び出すと、直線を描画するペンを操作することができる。操作の組合せにより線画が描画できる。例として二重ループを使って同心六角形を描く課題を考える。六角形を一つ描画し次の六角形を描画するには、図2に示すように、幅0の線を描き、ペン先を一つ外側の六角形の開始位置まで移動させなければならない。図形描画はこのようにペンを細かく制御する難しい問題であるが、実行結果が目に見えるので、正しくプログラムが動作したときの喜びが大きく受講生の意欲を得やすい。

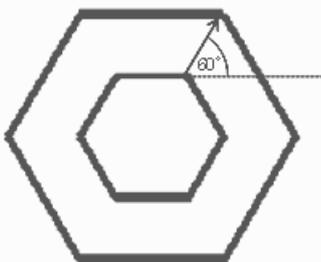


図2 同心六角形描画でのペン先の移動

#### (2) インタラクティブ・チェック

インタビュー結果の分析により、わからないときには受講者は何を質問すべきかもわからないことが判明した。何の指導も受けられない受講生は、置き去りにさ

れた感覚に陥り脱落していくことも明らかになった。

我々は、このような事態に陥ることを防ぐサービスとして、インタラクティブ・チェックを導入した。このサービスでは、表1のような公開されたチェック項目に対し、教員もしくはTAからなる指導者がソースコードを学習者と1対1で検査する。指導者が指定した機能を実現するコードを受講者が示し実現できる理由を説明する。このサービスは、何を質問すべきかわからない学習者が指導者に質問するきっかけを与える。表1における質問は指導者と受講者の双方に公開されるが、期待される回答は指導者のみに知らされる。質問3で、六角形を描き終え、その外側の六角形に移動するときの処理が問われている。この課題の題意は、多重ループで1回の内側ループの処理の後、次の内側ループに入る前に実施すべきことを受講者に理解させることである。これを共通のチェック項目で全員に示すので、もれなく重要事項を指導する、教え方に関わるサービスが実現される。

表1 チェック項目

番号	質問	期待される回答
1	RitsPen.hはどのディレクトリにあるか	ソースコードと同じディレクトリ。このファイルを#include "RitsPen.h"としてインクルードしているか
2	RitsPenをつかうのに最初に実行すべき処理は	PEN型変数penを定義し、pen = CreatePen("課題名")を呼ぶ
3	ひとつの六角形を書き終えたあと、透明の線を引くことによって、penの移動が実現できているか	透明の線を引いた後、penの向きを水平に戻しておくことが大切
:	:	:

#### (3) 連携の効果

図形描画問題とインタラクティブ・チェックを、Webサーバ上で受講者と指導者が共有するチェック項目により連携させることにより、以下の効果が期待できる。

- 出題者の意図を、多数の指導者と受講者に広く伝えることができる。これにより、重要事項をすべての受講者に抜けなく教育できる。
- インタラクティブ・チェックは、受講生に、丁寧な指導を実感させる。この感覚は受講者を動機付ける。しかも、指導基準の標準化により、
- 多数のTAがいても同一の指導が実施される。指導者に準備不足があると、受講者はそのやる気を大きく減退させるが、インタラクティブ・チェックの枠組みは、TAに指導の準備をさせるので、受講者の満足が得やすい。

### 4. 実践による改善効果

興味深い課題、手厚い指導を融合したCPの有効性を検証するため、1年生約500名向けに2010年前期に開講されたプログラミング演習1にこれ

を適用し、その際の学習経過を記録した。

2010年では、第1週と第2週はUNIXコマンドを学習した。第3週から第8週までは課題を満たすプログラムを作成する演習を実施した。2010年は、プログラミングにおける初期ハードルを受講生が乗り越えることを期待してインタラクティブ・チェックはここで実施した。第9週は中間テストで、第10週から第14週までの演習ではインタラクティブ・チェックを実施していない。第15週は期末テストであり、最終週までに提出されないソースコードは評価の対象より外した。

### (1) 図形描画課題

単位取得のために、ほとんどの受講生は最終的に全課題を解いてソースコードを提出しているが、受講生は興味ある課題から閲覧を始める。Webサーバ利用ログより、受講生がどの問題から閲覧しているかを分析した。図3はインタラクティブ・チェックを実施した第3週から第8週の6週間において、セメスタ終了後の成績別に受講生を分別したときのグループを横軸に、図形描画課題から先に閲覧している受講者の割合を縦軸に示している。成績は130点を満点としている。6週のうち3週以上、図形描画課題から閲覧している受講者が、ほとんどのレベルで半数を占めている。多くが図形描画課題に興味をもっているといえる。

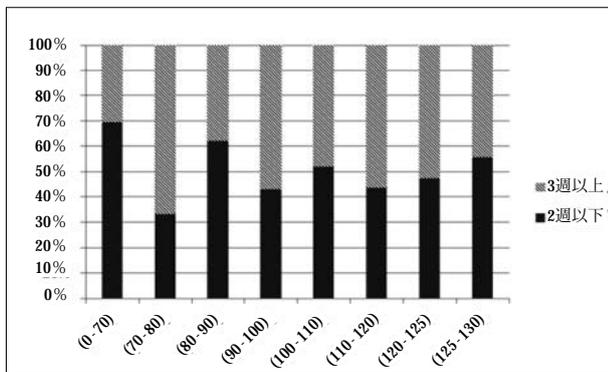


図3 はじめに閲覧する課題

### (2) インタラクティブ・チェック

セメスタ終了後の成績で受講者を分類し、それぞれの週の7日間でサーバを利用していた時間の平均値を調べた。1週間の中で強制的にサーバを利用するのは、演習の授業を実施している90分間であるから、それ以外は受講者が自習のためにサーバを利用したと言える。自主利用時間を計算した結果を図4に示す。横軸は週を、縦軸は自主利用時間を示す。第12週から第13週にかけて、自主利用時間が急激に増大しているのは、課題提出の締切りによるものと考えられる。第9週は中間テ

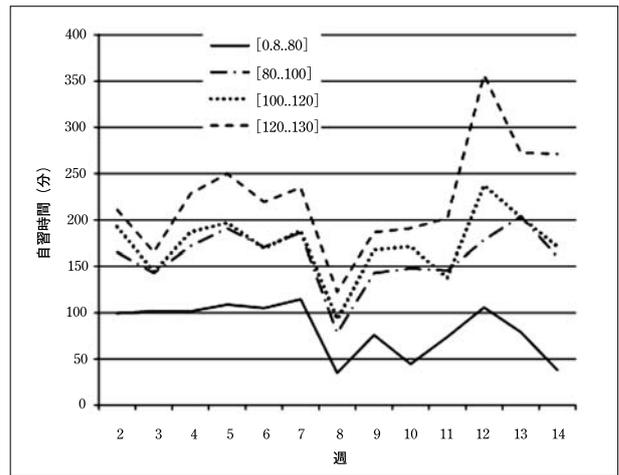


図4 サーバ自主利用時間

スト、第15週は期末テストであった。これらの週を除外すると、インタラクティブ・チェックを実施した第3週から第8週までの週では、インタラクティブ・チェックを実施していない第10週から第14週までの週に比べ、すべてのグループで自主利用時間が長いことがわかる。インタラクティブ・チェック実施期間中は受講者はモラルを保ってサーバを自主利用していたが、これがなくなるとサーバを自主利用していない。インタラクティブ・チェックは受講者の動機付けに正の影響を与えることがわかる、しかし、同時に自主学習を習慣づけるまでの影響力はないこともわかる。自習を習慣づけるには、4.(3)に示す厳しいノルマを課す方法が有効であった。

インタラクティブ・チェックの実施期間と非実施期間で、提出されたソースコードの評価値を調べた。セメスタ終了時点でのすべての週のソースコードの平均点で受講者を分類して、各週で出題された課題に対するソースコードの評価値を図5に示す。横軸は週、縦軸はソースコード評価値を示す。ソースコード評価の満点は130点である。本コースの課題は週を増すごとに難解になり、評

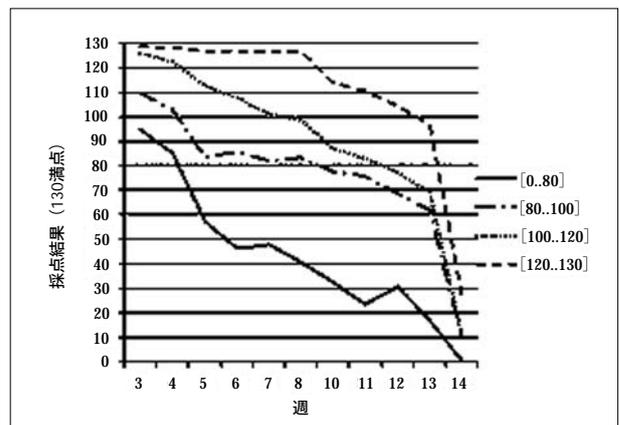


図5 ソースコードの質

価値は減少する傾向にある。インタラクティブ・チェックを実施した第3週から第8週までは、多くの受講者グループの評価値の減少度は小さいが、実施しなかった第10週以降は評価値の減少度が大きい。インタラクティブ・チェックが受講者の理解を促進する上で有効であったことがわかる。

### (3) 厳しいノルマの追加

多くが落第する必須科目では、受講者からの不満が大きい。必須科目はその後の学習の礎となるので、教員も全員の合格を願う。このため、ややもすると、必須科目の難易度は下がってしまう。しかし一方で、単位を落とすことを受講者が極度に嫌うのならば、教員は厳しいノルマを課して、受講者を勉学に励ますこともできる。2010年には2週間の締切りをつけた課題を各週に2題ずつ出題し、他の課題は締め切りなしとした。図6で、横軸は提出から経過した週、縦軸は締切りをもつ課題の提出数を示す。一方、締切りのない課題はコース終了時に集中して提出された。締切りのようなノルマも、受講者を学習させる動機となる。

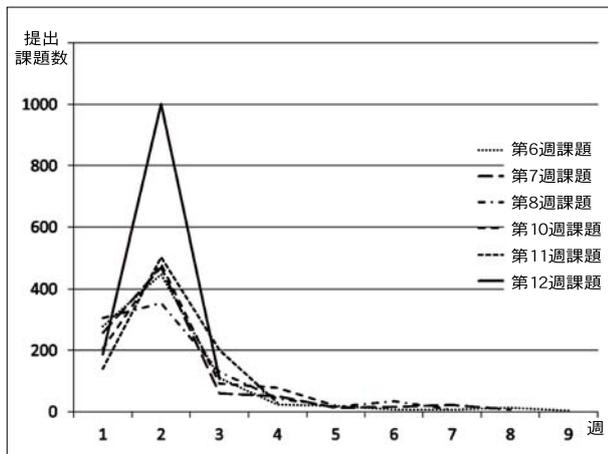


図6 締切りの効果

図形描画課題とインタラクティブ・チェックの双方を取り入れた2010年度に加え、2011年度には提出期限を厳格にするなどの厳しいノルマを課す一方、毎週の理解度を小テストで確認し、理解が定着していない学生にのみインタラクティブ・チェックを実施した。結果として、受講者の動機、理解度が向上した。たとえば、中間テストでの100を満点とするプログラム読解問題の平均点が2010年度では473名に対し70.74であったが、2011年度には550名に対し80.50で9.76ポイント改善した。さらに、サーバ利用履歴を分析したところ、自習時間が増大し、課題に早くから取り組む傾向が見られた。

## 5. おわりに

本論文では、学習者のふるまいデータよりプログラミング学習におけるコース案を洗練した実例を示した。その例では、興味深い課題として図形描画課題を、手厚い指導としてインタラクティブ・チェックを導入し、これらを、Webサーバ上で公開されたチェック項目を使って連携させた。紹介したコース案を実際のプログラミング演習授業に適用し、その効果をサーバ利用履歴から導出した学習上のふるまいデータで示した。

興味深い課題を与えても、受講者がそれに取り組まなければ意味が無い。課題提出の締切りや頻繁に試験を実施するといった厳しいノルマは、受講者を勉学させるのに有効である。勉学する対象が興味深いもので、かつ、それを理解できていれば、受講者はそれ以降、自主的に学習に取り組むようである。2010年から2011年にかけての改善は、この事象の現れであると我々は考えている。しかし、どの程度まで厳しいノルマを課せるかは、今後、コース案を策定して、その適用結果を分析し、検討する必要がある。

## 謝辞

本研究に貢献ある、立命館大学理工学研究科博士後期課程のDinh Thi Dong Phuong女史に謝意を表します。

## 参考文献

- [1] Michael Köling, Poul Henriksen: Game Programming in Introductory Courses With Direct State Manipulation. *ITiCSE'05*, Monte de Caparica, Portugal, 2005.
- [2] Laurie Williams, Kai Yang, Eric Wiebe, Miriam Ferzli, Carol Miller: Pair Programming in an Introductory Computer Science Course: Initial Results and Recommendations. *OOPSLA2002*, Seattle, WA, 2002.
- [3] Hugh Beyer, Karen Holtzblatt: Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann, 1998.
- [4] Dinh Thi Dong Phuong, Fumiko Harada, Hiromitsu Shimakawa: Refinement of Programming Course Plan with Inquiry and Behavioral Data. *IADIS e-Learning2011*, Rome, Italy, 2011 (to be published)
- [5] Robert E. Slavin: Educational Psychology: Theory and Practice. 10<sup>th</sup> ed., Pearson, 2011.