

# 初心者の自己主導学習でのプログラミング学習パターン

## —ドリトルを利用した迷路脱出プログラムを作成する実験授業—

SeungWook Yoo\*, YongChul Yeum\*, 兼宗 進\*\*, WonGyu Lee\*\*\*

\*高麗大学 大学院 computer 教育学科

e-mail: {yoosw0810, yycok}@comedu.korea.ac.kr,

\*\*一橋大学 総合情報処理センター

e-mail: kanemune@cc.hit-u.ac.jp

\*\*\*高麗大学 師範大学 computer 教育学科

e-mail: lee@comedu.korea.ac.kr

### 概要

コンピュータの原理を理解するのにプログラミングはとても重要なことである。しかし小・中等学校でプログラミングを教育課程で教える場合、時間の制限と学習者のアルゴリズム的思考が発達していないため適用することは難しい。これを解決するために教育目的のプログラム言語がいろいろ作られてきた。本論文は小、中等教育で教育目的のプログラム言語の適用可能性を探すために実施した実験授業の結果を報告する。なおプログラミングをする過程で発生する認知的負荷と学習パターンを観察しその内容を整理報告する。使った言語は初歩学習者が短い時間に学ぶことができるとオブジェクト指向概念を適用することが易しい言語であるドリトルである。実験に参加した学習者は迷路を脱出するアルゴリズムを自己主導的で設計してプログラミングを完成した。最終的に学習者は認知的負荷を克服してプログラミングに必要なスキーマを形成してアルゴリズム的思考が必要なプログラムを上手に作成するようになった。一度与えられた課題を完成したあとで、学習者は新しい課題を解決する能力が向上した。

## 1. はじめに

本論文は、認知発達水準が比較的高いがプログラミング経験がない高校1年生にドリトルの文法的構造を教え、自己主導的に問題を設計し解決方法を考える中でアルゴリズム的思考で問題を解決して行く過程のパターンを観察・分析した実験授業研究報告である。

この研究の結果は初等中等教育過程のプログラミング教育課程を設計する準拠になるだけでなく、プログラミング教育で使用される課題を学習者がどのように解決して行くかに対する理解を学習者の興味を維持しながらアルゴリズム思考を伸ばす課題の開発に適用できる。なお、学習者が作成したソースコードの分析を通じて、学習者の認知負荷に関する分析も可能である。

## 2. 理論的な背景

プログラミングを初めて学ぶ学習者はいろいろな難しさを経験する。言語自体に対する難しさから、抽象化された課題を解決するためのものまで、プログラミング過程で多くの認知的負荷を感じる。

### 2.1 認知的負荷

基礎プログラミング学習者がプログラミング過程で感じる認知的負荷を、Garner は次のように分類した「1」。第一は、プログラミング言語自体またはプログラミング過程を通じて解決すべき問題自体に対する負荷である intrinsic 認知負荷である。第二は、問題をどのように表現するかに対する負荷である extraneous 認知負荷である。第三は、問題

解決過程自体に対する負荷である *germane* 認知負荷である。

*Intrinsic* 認知負荷は学習者の精神的負荷が発生するのによって決定されるもので、外国語の単語を覚える場合は少ないが、プログラミング領域では非常に大きい。プログラミング過程を通じて解決すべき問題を認知することで発生する負荷もこれに属する。これはプログラム言語の文法的構造とプログラミングテクニックにスキーマが形成されない場合発生するもので多様な *syntax error*, *semantic error* を発生する。しかし教育利用を目的した教育用プログラム言語の利用は *Intrinsic* 認知負荷を減少することに効果があるという研究がなされた。

*Extraneous* 認知負荷はプログラミング過程を通じて解決すべき問題を表現するのに対する負荷で、特にプログラミングの経験がない場合には大きい負荷となる。*Intrinsic* 認知負荷が大きい場合はもっと高い *Extraneous* 認知負荷を感じるようになるため、教師が認知負荷を下げるように *Worked examples* などを使用する場合がその例である。

*Germane* 認知負荷は問題を解決して行く過程で発生する認知負荷で *intrinsic* 認知負荷がそれほど高くない教育用プログラミング言語の場合 *extraneous* 認知負荷を減ることができるなら学習者はプログラミング過程で問題解決に対する認識過程が促進されてスキーマ形成を助ける。この過程で発生するのが *germane* 認知負荷である。

## 2.2 作動記憶

基礎プログラム学習者はプログラミング過程に対するスキーマが形成なされていないので、課題が付与されるとプログラミング言語と問題自体に難しさを感じて、これを解決するために *Working Memory* を動作させる。仮に課題をプログラミング言語で表現することに関する難しさまでを合わせると、*Working Memory* はより強く動作するようになるだろうし、これは学習者の学習放棄

を誘発する可能性がある。しかし学習者のスキーマに適合しやすい言語の場合、*Working Memory* の負担は軽くなり、*Long-Term Memory* を動作させながら新しいスキーマを形成するようになる。このような過程を通じて、*Germane* 認知負荷も *Working Memory* で動作するようになる。

認知的負荷を少なく感じる学習者、すなわち *intrinsic*, *extraneous*, *germane* 認知負荷を克服した学習者はプログラミング過程の中で発生する問題解決過程を *long-term memory* のスキーマを通じてなされることになる。仮に新しい文法的構造を学習する場合、*long-term memory* のスキーマが形成されていない場合には *working memory* を動作させて新しいスキーマを形成し *long-term memory* に包含させる。

## 2.3 教育用言語ドリトル

プログラムを初めに学ぶことは、*long-term memory* にスキーマが形成されていない学習者は言語自体と言語で表現する問題、そして解決方法に対する認知的負荷が大きい。これを小さくするために、色々な教育用言語が開発されて認知的負荷を減らしている。

ドリトルはオブジェクト指向言語で初等中等教育のため開発された教育用プログラミング言語である。学習者の認知負荷を減らすのため文法構造と興味を引くための活動者 (*actor*) であるカメを画面に配置することは以前の *LOGO* と同様である。しかし、次の主な特徴から、本研究の言語として選択した。

- テキストで記述する言語である。プログラムで動作する計算機の原理を直感的に理解させる長所だけでなく、学習者にプログラムコードを直接見せることは大切である。また、学習者のソースコードを収集することも可能である。
- オブジェクト指向言語である。学習者がオブジェクトの概念を認知するパターンを

把握し、問題をオブジェクト概念で解決する過程を分析するのに適している。

- 文法構造が簡潔であり、短い時間にスキーマ形成が可能である。これは他の言語にない特徴であり、以前は不可能だった研究を進めることが可能になる。

この中で一番重要な要因は、認知的負荷が小さい点と短い時間で long-term memory にスキーマが形成されることである「2」。

### 3. 実験授業の設計および実施

本研究の結果を得るための実験授業の総時間と課題を解決するための学習者が学ぶべき学習ポイントを見積もることは容易ではなかった。生徒に要求したのはカメが迷路を脱出するアルゴリズムを作成するものだが、これも学習者が自ら設計し挑戦した課題であった。

学習者は自身の学習パターンを決定し、教師は必要に応じて助言するなど学習者の要求によって知識を教える方式で実験授業を進めた。全体的に自己主導学習で行った。時間的な要素を把握するため、学習者の個人的な練習はさせなかった。よって、他の場所でドリトルを使ってプログラミングをする機会はなかった。

#### 3.1 学習者の分析

実験授業に参加した学習者は韓国の高校1年生である。1次授業には4名が参加した。中学校の成績が11.04%, 11.98%, 46.74%, 66.23%の優秀な学生であった。1人を除き、3人はプログラミング経験がなかった。

2次授業にはドリトルを学ぼうとする15名が参加した。1次授業と比較するため、成績は学生全体の中で中上位圏で構成した。プログラミングの経験は1次授業の結果を参考にして調査しなかった。

#### 3.2 実験授業の実施

1次授業は2005年9月から10月までの約2ヶ月に行い、一週間に4-5回で合

計23時間実施した。2次授業は2006年5月中に行い、5日間で毎日3時間合計15時間実施した2次の場合課題の設計および例題プログラムの説明に所要された5時間を除いたまま短時間の集中授業で進行されたので、1,2次実験授業の間には時間差はないのであると判断する。研究のため制限された課題とプログラミング環境が適用されたあとで学習者のプログラミング全過程のソースコードを保存した。

### 4. 学習者の学習段階観察

実験授業を段階別で区分するのは難しいが、今回は1,2次実験授業で学習者が自己主導的に学習を進行する過程でプログラムを完成させるのためのパターンを3段階に分類することができた。

1段階は図形を作成する過程を通じてドリトルの使用法を学んで挑戦すべき課題を設計した後、初めの試みをする段階で図2の①, ②にあたる。2次実験授業では課題の設計過程は省略された。2段階は設定された課題を解決するために問題を表現する方法を含めてプログラム言語に対する深化学習がなされる段階で二番目の試みがなされる。図2.で③, ④が該当し③-①, ③-②は学習者によって選択的に適用がなされた。比較的長いプログラムを学習する過程を通じて、プログラムの作り方と継承およびオブジェクトの概念に対する反復学習がなされた後、課題を解決する長い時間があり図2.で⑤, ⑥が該当し⑥-①, ⑥-②は選択的に適用された。これを通じて学習者は自己主導的に課題を完成することができ、強いスキーマが形成されたことが⑦で確認された。これを3段階学習に分類してこれに対して詳しい内容を説明する。

#### 4.1 1段階学習

実験授業の初めはドリトルの使用法および文法を学ぶことから始めた。学習者はドリトルを初めて学んだが図形を作成した後、動きを与える過程で大きな難しさはなく、短い

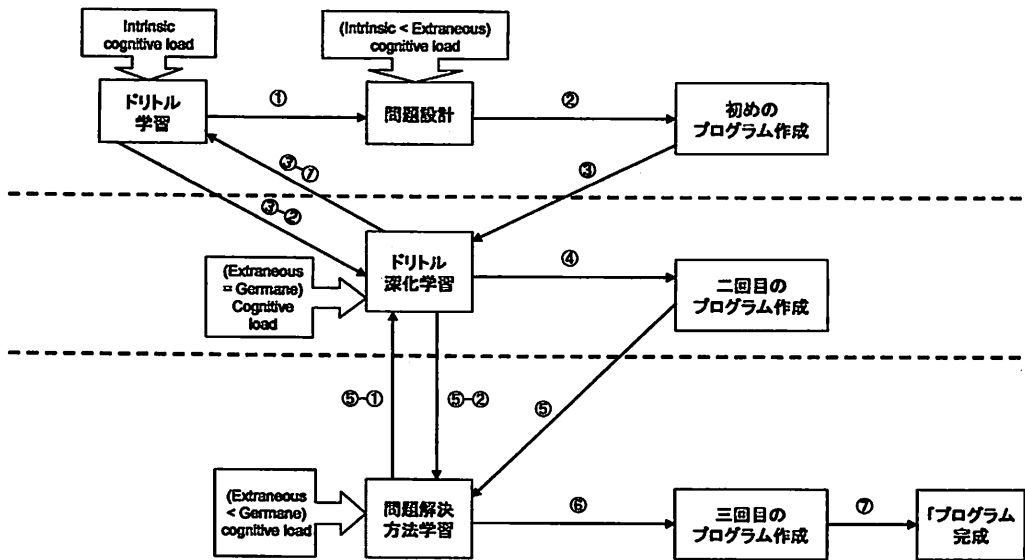


図 2. 段階別学習進行

時間で理解した。この過程でオブジェクトの生成、タイマーと衝突メソッドなどを学習しながら簡単な例題問題を解いた。総所要時間は約 3 時間であった。

生徒は Intrinsic 認知負荷を感じたが、大きい難しさのないまま解決されたのは以前の研究結果と同様である「2」「4」。学習者はドリトルの文法を学んだ後、自分で課題を設定し、学習を進めた。画面を設計し、プログラムの設計および作成、デバッグする過程をしたが、学生は全員がプログラミング経験がなかったにも関わらず、学習者自身が知っている以上の難しい内容で問題を設計した。オブジェクトの生成とメッセージ送信に関する概念を正確に把握しない状態で、迷路を脱出するカメのアイデアが現れた。

②は intrinsic 認知負荷を少し経験しながら extraneous 認知負荷が初めに発生される。学習者は認知負荷があるにもかかわらず、自身の主導的な学習方法に満足した。

迷路を脱出するカメのプログラミングをする試みは、学習した内容では図形を構成する以上は進展しなかった。短い時間試みをしてから再びドリトルの文法を学習者自ら学ぶように決定した。

#### 4.2 段階学習

プログラムを作成して課題を解決するために色々な認知的負荷を感じる段階である。オブジェクトの概念とメソッドの使用に対してスキーマが形成されないために多くのミスが発生しているし、これは extraneous 認知負荷に intrinsic 認知負荷に含まれる様相を見せたが intrinsic 認知負荷は相対的に多くなかった。

多様な学習行為がなされた。代表的なものを次に整理する。

- 配列オブジェクトを使った同期および非同期的な動作指定
- 継承と同じ概念の必要性認識およびプログラム作成
- 条件文を利用した簡単なプログラム作成

長時間迷路脱出プログラムを作成しようとする作業が続いた。この過程はほとんど intrinsic 認知負荷がない状態で extraneous 認知負荷が大部分であり、germane 認知負荷が現れはじめる段階である。

生徒の感想では、タイマーの概念と継承の

概念を確かに理解したと話しており、反復文と条件文の使用が始まった。しかし条件文は文法的に理解していたが実行時はエラーが多く、問題解決過程自体の難しさに対する感想が多かった。5-6回の試行錯誤では解決しない生徒が多かったが、教師の助けのない状態で自力での解決を希望した。

生徒はときどき、③-①, ③-②過程を反復したが、認知負荷というより long-term memory に syntax のスキーマがないことが確認される程度であった。

### 4.3.3 段階学習

⑤番は問題を解決する方法に対して学習者が深化学習の必要性を感じる過程である。オブジェクトの概念およびオブジェクト指向言語で提供する継承を適用しようとする試みがなされた。この時点はドリトルの文法、アルゴリズム的思考の表現方法に対してある程度自信をもっていることが確認されたが、問題解決方法に関するスキーマが形成されなかった。したがって germane 認知負荷を extraneous 認知負荷よりも感じる姿が見られた。

これを解決するため、比較的長い例題プログラムを使ってプログラムの作成および表現方法に対して自己主導学習がなされるように誘導した。この段階で変数の使用方法(大域変数、局所変数)、条件文の必要性および使用方法に関する認識がなされた。

自己主導的にプログラムを直す行為が続いて迷路脱出プログラムに適用しようとする努力があった。これは germane 認知負荷の適切な調節がプログラミングの学習に助けることを確認させるものである。

迷路脱出プログラムに対する挑戦がつづいているうちに数多くの無限ループを経験しながらカメが迷路に衝突する回数と距離を利用してアルゴリズム的に解決しようとする試みが⑤-①, ⑤-②過程にあった。問題を解決する過程でアルゴリズムで問題を解決するスキーマが形成されるのを学習者自身が認知して

いった。

1次実験授業で4名の中で3名がお互いに違うアルゴリズムを開発して迷路を脱出するプログラムを完成した。2次 実験授業では15名の中で5名がプログラムを完成した。完成されたプログラムの画面を図3に示す。

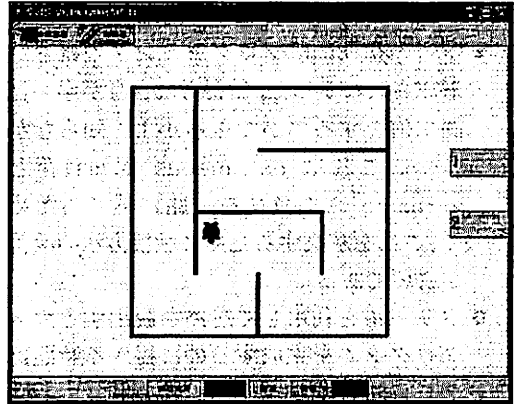


図 3. 完成された作品

⑦は working memory で学習者が認知していた認知負荷を克服して long-term memory に強いスキーマを形成する過程で現れるもので、以後アルゴリズム的思考で類似の問題を解決するときに long-term memory のスキーマが動作することを意味する。

最後に、アルゴリズム思考の伸展を通じた問題解決能力を測定するために類似問題を教師が提示した。本研究では3個以上の別々に動作するストップウォッチをプログラミングさせるようにした。自己主導的学習を通じて実験授業が成功した場合、短い時間に全員が課題を解決することができ、 long-term memory に強いスキーマが形成されることを確認した。

## 5. 整理と結論

本研究では、プログラミングを初めて学ぶ学生が言語の文法を学習した後、どんなパターンでアルゴリズム思考を利用して問題を構成し解決するかを検討した。そして、実験授業を行い、以下の結論を得た。

- 自己主導的にプログラミングを学習する過程で、一定なパターンを持つ段階別学習がなされた。特に1次実験授業に参加した学習者は自身が設定した課題を解決するため3種類の試みがあったし、これは別の特徴を見せてこれを3段階の学習で分類した。
- 学習の初期にはプログラミングの概念が形成されなかった。学習者は自身が学ぶ言語で何をすることができるかよく知らなかった。これは extraneous 認知負荷と germane 認知負荷を一緒に感じるため、これを解決するために教授方法の改善が必要である。
- ドリトルを利用して迷路を脱出するカメプログラムを完成するのに掛かる時間は20時間前後であった。段階別には、図4の各段階はそれぞれ30%、40%、30%程度であったが、絶対的な数値ではない。各段階内で文法の学習に所要された時間は全体的に1/3程度で、ほかの時間は問題の設計、表現と実際プログラムを作成するのに所要された。これは初等中等教育で教育用プログラミングの適用可能性を現実的に確認することができた。

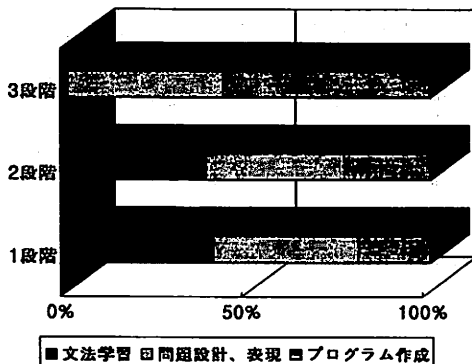


図 4. 1次実験授業での段階別背所要時間

- 各段階の間の学習転移は認知的負荷を感じてそれを解決しようとする努力がある場合に発生した。即ち、循環学習が発生しているし、各段階の中での学習は手続き的

な流れを見せた。

## 6. おわりに

初等中等教育課程でプログラミング教育が独立教科で運営されていない現状では、情報教育の一部として認知負荷を最小化しながら計算機の原理を教えることは重要である。そのためには、まず学習者の学習パターンを分析することが重要であり、本研究ではこの点に貢献することができた。

今後の課題は、確認された学習パターンによってプログラミング言語自体が持っている認知的負荷を最小化する研究と同時に、問題の表現方法、問題解決過程自体に対する学習者の負荷を最小化する教授方法の研究である。さらに詳しい学習者パターンについては、本研究で収集されたソースコードを分析することで明らかにしたいと考えている。

## 参考文献

- 1) Garner, S. "Connitive Load Reduction in Problem Solving Domains", International Conference in Computer Education, ICCE2001 Seoul, South Korea, 2001.
- 2) 呂修京、ドリトルを使った‘デジタル論理回路’教授設計および教材開発と学業成就度分析。高麗大学。2005。
- 3) 兼宗進、御手洗 理英、中谷 多哉子、福井 眞吾、久野 靖。学校教育用 オブジェクト指向言語「ドリトル」の設計と実装。情報処理学会論文誌, Vol.42, No.SIG11(PRO12), pp78-90, 2001。
- 4) HyeMin Gil. "Application of Object-Oriented EPL Dolittle in Secondary Education", Korea University, 2004。
- 5) 兼宗進。プログラミング言語ドリトル」。 <http://dolittle.eplang.jp/>