

英国の公立小学校における教科「Computing」における プログラミング教育の内容

石塚丈晴^{†1} 堀田龍也^{†2}

概要：2014年9月より英国（イングランド）の公立小学校の全学年にこれまでの教科「ICT」に代わり、新教科「Computing」が導入された。この「Computing」に関しては、national curriculum では児童が教わるべき内容は、非常に簡潔に記載にとどまり、具体的なカリキュラムについては小学校に任された。本報告書では、現在最も多くの小学校で採用されている教材集である「SWITCHED ON Computing」から、プログラミング教育に関する内容を抽出して、その内容の概略を紹介し、3回に渡って継続的に調査したロンドン市内の小学校での実施状況も含めて報告する。

キーワード：英国，イングランド，小学校，Computing，プログラミング教育

Content on Programing Education in “Computing” in England

TAKEHARU ISHIZUKA^{†1} TATSUYA HORITA^{†2}

Abstract: A new subject named “Computing” was started in September, 2014 for all the local education authority maintained schools in England, UK. In this report, content on programing education in the educational materials published as “SWITCHED ON Computing” for “Computing” is shown.

Keywords: England, primary school, Computing, programing

1. はじめに

英国ではデジタル産業界からの要請などを受け、英国（イングランド）の公立小学校1年生（=5歳）から、新教科「Computing」の導入が決定され、2014年9月から実施されている。2014年度以前には、1995年に定められて導入された教科「ICT」が全学年で実施されていた。「ICT」では、ICTの活用が主目的の授業であったが、「Computing」ではComputer Scienceの内容が大幅に取り入れられた。尚、従来の「ICT」は「Computing」の導入に伴い廃止された。

2. 本研究の意義

我が国の現在の学習指導要領において、プログラミングに関する記述を探すと、中学校の「技術・家庭」での「情報に関する技術」で「情報処理の手順を考え、簡単なプログラムが作成できること。」という記述が見られる[1]。また高等学校では主として専門学科において開設される各教科を除くと「情報の科学」での「問題の解決と処理手順の自動化」及び「モデル化とシミュレーション」で「学校や生徒の実態に応じて、適切なアプリケーションソフトウェアやプログラム言語を選択すること。」という記述がある[2]。しかし、小学校には一切記述が無い[3]。

しかし、次期学習指導要領の方向を検討している中央教

育審議会による論点整理（案）では、中学校の技術分野で「プログラミング」などの「情報活用能力の育成等について、更なる充実が求められる」、また、高等学校の情報で「小・中・高等学校の各教科等を通じた情報活用能力の育成について、三つの柱に沿って明確化し、学校外の多様な教育活動も連携しつつ、プログラミングや情報モラルなどに関する学習活動の充実を発達段階に応じて図る」と記述されている[4]。

これは、我が国のこれからの学校教育で、発達段階に応じたプログラミング教育の導入を検討している事を示しており、先進的にプログラミング教育が導入されている国の情報を調査・報告することは意義があると考えられる。

3. 「Computing」のカリキュラム

イングランドの公立小学校における教育内容については、教育省(Department for Education)による national curriculum に従う必要があり、定期的に Ofsted(Office for Standards in Education, Children's Services and Skills)による査察を受ける。

「Computing」の national curriculum [5]では、「教科の目的」の冒頭で、「児童が computational thinking や創造性を用いることによって、世界を理解し変革していく」と述べている。また、「Computing」の内容は3つの要素（Computer Science、

*†1 福岡工業大学短期大学部
Fukuoka Institute of Technology, Junior College
†2 東北大学
Tohoku University

Information Technology, Digital Literacy) に分けられると記されている。

続いて Key Stage と呼ばれる学年区分毎に、児童が教わるべき内容が記されている。尚、小学校の学年区分は、Key Stage 1 が 1~2 年生で、Key Stage 2 が 3~6 年生となる。それぞれの内容は久野らにより表 1 の通り訳されている[6]。

表 1 児童が教わるべき内容 (文献[6]より転載)

Table 1 Content which pupils should be taught (from Ref.[6]).

Key Stage 1 (5歳-7歳)
--アルゴリズムとそのプログラムによる実装、デジタル機器、命令によるプログラムの実行などの理解 -簡単なプログラムの作成とデバッグ -簡単なプログラムの論理的推論による動作予測 -デジタルコンテンツの作成、構成、格納、操作、取り出し -技術の安全かつ他人を尊重した利用、個人情報、ネット上の素材の利用、誰に助けを求めらるか -学校外での一般的な情報技術の利用
Key Stage 2 (7歳-11歳)
--機器制御やシミュレーション等の特定の目的を達成するプログラムの設計、作成、デバッグ;問題を小さい部分に分けて解く -接続、選択、反復の使用;変数や様々な入出力の使用 -単純なアルゴリズムが動作する様子の論理推論;アルゴリズムやプログラムの誤り発見と修正 -ネットワークの理解;WWWなど複数のサービスの存在とそれらがコミュニケーション、協調作業を可能にすることの理解 -検索の活用と結果の選別・ランクづけの理解;コンテンツの評価 -多様なデバイスで動くソフトウェアの選択・利用・組み合わせによるプログラム構築;データや情報の収集・分析・評価・発信 -技術の安全かつ責任を持った利用、適切/不適切なふるまいの理解、問題の発見やその対処

表 1 に示されるように、教わるべき内容については非常に簡潔に示されており、実際 Key Stage 1, 2 の「Computing」の national curriculum はわずか 2 ページの構成である。他の伝統的な教科である英語では 88 ページ、算数では 47 ページの構成と比べると極めて少ない分量である。そこで、「Computing」を推進するために設立された機関である CAS (Computing At School) が小中学校教員向けのガイドを作成し、カリキュラムの解説などを行っている[7]。

4. 教科書及び教材

教科書や教材、カリキュラムについては特に定められておらず、学校の裁量で自由に決定することができる。各種の団体によって教材が提供されているが、最も多くの小学校で採用されている教材は、RISING STARS 社が発行している「SWITCHED ON Computing」である[8]。この教材は Havering educational services (ロンドン市 Havering 地区教育委員会) が CAS や Naace (The National Association for all those interested in technology in Education) らと共同で開発したものである。Havering 地区教育委員会は、教科「ICT」の時に、小学校現場からの要請に応じて、RISING STARS 社から「SWITCHED ON ICT」を開発しており、そのノウハウを

もとに短期間で制作された。2015 年 6 月に関係者に問い合わせたところ、6000 校以上で使用されているとのことである。

5. 本研究報告の目的

本研究報告では、前節で紹介した SWITCHED ON Computing で扱っている学習内容の内、特にプログラミングに関係したカリキュラムや内容を抽出して紹介すると共に、この教材を基に、カリキュラム設計し実践を行ったロンドン市内の小学校の事例を紹介する。

6. 教材の構成

英国の公立小学校では 3 学期制を採用しており、各学期は更に 2 分割されている。「SWITCHED ON Computing」では、各学年に対して 3 学期×2 = 6 単元で構成されており、それぞれ 1.1~6.6 の 36 単元に分けられている。単元欄の左側の数字は対象学年を示している。各学年の単元は「Programming」「Computational thinking」「Creativity」「Computer networks」「Communication/Collaboration」「Productivity」の 6 つのカテゴリーの学習教材が提供されている。これらの中から「Programming」に関する単元を抽出したものが、表 2 である。尚、表 2 の単元 6.5 に関しては、6 年生では年間を通してスマートフォンのアプリ開発のカリキュラムが組まれており、6 年生のカリキュラムの全体は表 4 で示す。

表 2 「Programming」カテゴリーの単元
Table 2 Units of “Programming.”

単元	タイトル	内容	使用ソフト/ ハード
1.1	We are treasure hunters	Using programmable toys	Programmable toys
2.1	We are astronauts	Programming on screen	Scratch/ Kodu/ Scratch Jnr
3.1	We are programmers	Programming an animation	Scratch/ PowerPoint
4.1	We are software developers	Developing a simple educational game	Scratch/ Snap!
5.1	We are game developers	Developing an interactive game	Scratch/ Kodu
6.5*	We are app developers	Developing a simple mobile phone app	App Inventor

また、表 2 以外では、「Computational thinking」のカテゴリーでもプログラミングに関する単元があり、表 3 に示す。

表 2 及び表 3 より、小学校 1 年生から 6 年生までの全ての学年でプログラミングの単元が配置されていることが分かる。

表 3 「Computational thinking」 カテゴリーでプログラミングに関する単元

単元	タイトル	内容	使用ソフト/ ハード
2.2	We are game testers	Exploring how computer games work	Scratch, Screencast-o-matic
3.2	We are bug fixers	Finding and correcting bugs in programs	Scratch/PowerPoint
4.2	We are toy designers	Prototyping an interactive toy	Scratch

表 4 は、6 年生のカリキュラムを示している。6 年生では年間を通して、スマートフォンのアプリを開発するための活動が組まれている。

表 4 6 年生の単元一覧

単元	タイトル	内容	カテゴリー
6.1	We are app planners	Planning the creation of a mobile app	Computer networks
6.2	We are project managers	Developing project management skills	Computational thinking
6.3	We are market researchers	Researching the app market	Productivity
6.4	We are interface designers	Designing an interface for an app	Communication/ Collaboration
6.5	We are app developers	Developing a simple mobile phone app	Programming
6.6	We are marketers	Creating video and web copy for a mobile phone app	Creativity

表 2 及び表 3 の各単元の概略を以下に示す。

(1) 「1.1 We are treasure hunters」

この単元では、マス目上に置かれた宝物を、Bee Bot などの programmable toy や児童が「前に進む」や「左を向く」などの指示に従って取りに行く活動をおこなう。この単元を通して、algorithm, debug, instructions, predict, programming, robot などの言葉についても学習する。

(2) 「2.1 We are astronauts」

この単元では、単元 1.1 を受けて、Scratch の拡張機能を使用してあらかじめ作られている、「前に進む」「左を向く」などのブロックを並べることで、ロケットを目的地まで移動させる活動を行う。

(3) 「2.2 We are game testers」

この単元では、あらかじめ Scratch で作成された、いくつかのゲームを通して、コンピュータ・ゲームがプログラムで出来ていることを理解させる。

(4) 「3.1 We are programmers」

この単元では Scratch を用いて、児童が作成したアニメキャラクターをスクリプトで移動させることで、アニメーションを作成しその原理を理解させる。この単元では新た

に animation, script などの言葉についても学習する。

(5) 「3.2 We are bug fixers」

この単元では、児童はあらかじめ Scratch で作成されたサンプルを用いて、スクリプトの働きを理解し、問題点の修正や改善を行う。この単元では bugs, debug などの言葉についても学習する。

(6) 「4.1 We are software developers」

この単元では、児童は最初に教育用ゲーム（単元 3.1 で使用したものなど）を題材として、その特徴や設計思想などを学ぶ。次に、教育用ゲームを設計し、Scratch で制作する。この単元では、interface, prototype, repetition, variable などの言葉についても学習する。

(7) 「4.2 We are toy designers」

この単元では、児童は何人かで簡単なデジタルおもちゃ（例えば、画面のある部分をクリックすると状態が変わる）を設計して、Scratch で作成する。この単元では、interactive, simulation などの言葉についても学習する。

(8) 「5.1 We are game developers」

この単元では、児童は簡単なゲームを設計し、Scratch で制作する。この単元では、ゲームの背景画や効果音なども加え、出来たゲームの評価をフィードバックしてもらう。また、ゲームの説明文なども作成する。

(9) 「6.5 We are app developers」

この単元では、児童は単元 6.4 までで設計したスマートフォンのアプリを Scratch に似たインターフェイスを持った App Inventor などのツールキットを使用して作成する。実際の作成の前には擬似コードも用いたりする。

7. 導入事例

7.1 年間計画

表 5 は、筆者らが 2014 年 9 月から 2015 年 3 月の各学期に調査を行ったロンドン市内の小学校の SWITCHED ON Computing をベースとしたカリキュラムで、プログラミングの授業の実施計画である。「Computing」の授業は週 1 回行われる。表中の秋 1 は秋学期（前半）を示している。表 5 より必ずしも 6 年生以外は 1.1 から順番に実施するわけではなく、学校（担当教員）毎に異なっている。

表 5 プログラミングの実施計画(2014-2015 年)

学年	秋 1	秋 2	春 1	春 2	夏 1	夏 2
1				1.1		
2			2.1	2.2		
3			3.2		3.1	
4	4.2			4.1		
5					5.1	
6	6.1	6.2	6.3	6.4	6.5	6.6

7.2 実施状況

導入直後の2014年9月の秋学期(前半)には、表5の学校以外にも2校訪問したが、特に低学年の児童のクラスでは担当教員の授業計画どおりに行かない部分も目に付いた。

しかし、2015年3月の春学期(後半)に2回目の訪問を行った際には、ほぼ全ての学年で計画通りに実施されていた。尚、この時点で6年生は全ての単元を修了し、コンピューター・アンプラグド・サイエンスの内容の学習を行っていた。

2015年7月に3度目の訪問を行った際にも、ほぼ全ての学年で計画通りに実施されていた。また、6年生は中学校への進学に備えて、多くの中学校のプログラミング教育で使用されているPythonを用いたプログラミングの授業を行っていた。

7.3 担当教員による評価

児童の学習成果(2015年7月)について担当教員にインタビューしたところ、2013年度の「ICT」の評価では、到達目標に達した児童の割合は担当教員(=クラス担任)に依存し、全体的に見ても50%には達していなかった。2014年度の「Computing」では教科担当の教員が全学年の授業を受け持ち、到達目標に達した児童は50%強となった。50%強という数字は決して良い数字とは言えないが、導入初年度ということが大きく影響しており、このカリキュラムを続けて行くことで、徐々に改善されていくことが予想されるとのことであった。

8. まとめ

2014年9月に導入された英国(イングランド)の教科「Computing」で、多くの小学校が採用している教材から、プログラミングに関する学習内容を調査した。その結果、1年生から6年生までの全ての学年で、プログラミングに関する授業が行われていた。

1年生は、実際にはプログラミングは学習せず、アルゴリズムの概念などを学習する。2~3年生は、Scratchを使用してプログラミングを学習する。4年生以上では、プログラミングだけではなく、作成するソフトウェアの設計なども学習する。

導入1年後に授業について振り返ると、開始当初は特に低学年でやや混乱が見られたものの、その後は、ほぼ順調に計画通りに進んでいた。また、担当教員による評価では、到達目標に到達している生徒は50%を超えており、導入初年度としては妥当な数字であり、今後を期待したい。

参考文献

1) 文部科学省：中学校学習指導要領(2008)

- 2) 文部科学省：高等学校学習指導要領(2009)
- 3) 文部科学省：小学校学習指導要領(2008)
- 4) 中央教育審議会初等中等教育分科会教育課程部会教育課程企画特別部会：論点整理(案)(2015)
- 5) Department for Education: National curriculum in England: computing programmes of study (2013)
- 6) 久野靖, 和田勉, 中山泰一: 初等中等段階を通じた情報教育の必要性和カリキュラム体系の提案, 情報処理学会論文誌 教育とコンピュータ, Vol.1, No.3, pp.48-61 (2015).
- 7) Computing At School
<http://www.computingschool.org.uk/>
- 8) RISING STARS: SWITCHED ON Computing Year 1 – Year 6 (全6巻) (2014)