E-12

プロジェクトベースで学んだプログラミング入門の経験を振り返る

−ソフトウェア開発専攻への展開と効果の検証−

Experience of Introductory Programming Education in PBL

-Deployment to software development and verification of effectiveness-

柏　舜†　　内田 奈津子††
Shun Kashiwa　　Natsuko Uchida

1. はじめに

2020 年において、プログラミングの教育は小学校から企業研修まで様々な場所で行われている[1][2][3]。その目的は、論理的思考力の向上からエンジニアとして実践的な技術を身につけることまで様々で、目的と対象に応じて異なる言語やツール、手法を用いて学習が行われている。

現在大学の学部生としてコンピュータサイエンスを学ぶ第一著者が初めてプログラミングに触れたのは、小学三年生のときに参加したロボットコンテスト（以下コンテストと言う）であった。コンテストでの活動を通してプログラミングに興味を持った後、その他の言語やツールを独学で学び、大学でもコンピュータサイエンスを学んでいる。また大学を卒業した後も、ソフトウェアに関わる仕事がしたいと考えている。

コンテストでは、ビジュアルプログラミングを用いていた。ブロックなどの視覚的に分かりやすい表現で条件分岐や繰り返し、変数などの基本的なコンセプトを学んだ。そして、コンテストにおける開発はプロジェクトベースで行われた。与えられたテーマに沿ってチームとして制作するロボットの仕様を決定し、分担して開発を行い、最終的な成果をプレゼンテーションとして発表した。

第一著者が最初にプログラミングに触れてから 10 年が経過した。ソフトウェア開発の現場でインターンやアルバイトとして働く中で、コンテストを通して学んだことが今でも活きていると感じるようになった。

そこで、第一著者がコンテストを通してプログラミングを学んでから現在までを振り返り、どのようにソフトウェア開発のスキルを身に着けてきたか明らかにする。ビジュアルベースのツールを用いて、またチームでプロジェクトに取り組みながらプログラミングを学ぶことのメリットを、実際のソフトウェア開発と比べながら明確にする。

2. プロジェクトベースでのプログラミング入門

2.1 ロボット制作体験講習会

第一著者が初めてプログラミングに触れたのは、2009 年の小学校 3 年生の 2 月のことであった。当時、地元の公立小学校に通っていた第一著者は、緑園都市コミュニティ協会、フェリス女学院大学が協働で開催した「レゴロボット製作体験講習会」の案内を小学校で受け取った。

緑園都市コミュニティ協会は、子どもたちの科学技術への興味関心の向上と創造性や問題解決力の育成を目指し、これからの社会でグローバルに活躍する人材育成のため小中学生にプログラミングを教えたいと考えており、地域活動の一環として講習会を開いた。

漠然と科学やロボットに興味のあった第一著者は、プログラミングに触れたことこそなかったものの、講習会に参加した。

講習会では、LEGO Mindstorms NXT を用いてロボットを作り、そこにプログラムをインストールして用意されたコースの黒い線に沿って走ることが目標とされた。最初のロボットの組み立てでは、指示に従って LEGO のパーツを組み合わせた。一度ロボットを完成させた後には、パーツを付け替えて独自のロボットを作ることもできた。ロボットにインストールするプログラムの開発では、ロボットに付けられた光学センサの値に応じて適切にモーターを動かし、コースから外れないようにした。

2.2 WRO への挑戦

2.2.1 WRO について

講習会で強い興味を持った第一著者は、その延長としてロボットコンテスト”World Robot Olympiad“ に参加した。World Robot Olympiad (以下 WRO)は、「世界中の若者を集め、挑戦的で教育的なロボット競技会や活動を通して創造性、デザイン、問題解決能力を養う」[4]ことを目的として運営されているロボットコンテストである。2004 年から開催されている国際大会では、各国の代表が集い競技を行い、交流も行われる。

第一著者は、2009 年から 2010 年にかけて、WRO の「オープンカテゴリ」に出場していた。「オープンカテゴリ」では，毎年異なるテーマが与えられる。2 メートル四方のブース内において LEGO Mindstorms を用いてテーマにマッチした自律型ロボットを制作し、持ち時間の中でプレゼンテーションとデモンストレーションを行った。審査では、「テーマとの合致」「表現の創造性」「制作技術的内容」「プレゼンテーション」の観点から、ルーブリックを用いて点数化され、総合点をもって評価される[5]。

2.2.2 活動内容

2009 年に開催された WRO 2009 でのテーマは「アーティストロボット」であり、科学技術の集合体であるロボットと芸術を組み合わせることが焦点となった。

† The University of California, Santa Cruz
†† フェリス女学院大学　情報センター

[1] 緑園都市コミュニティ協会(横浜市泉区)は，アメリカの住宅地にあるホーム・オーナーズ・アソシエーションを参考として地権者、物件購入者、開発業者が一体となって旧来の自治会組織とは別に、日本では始めて住民主体のまちづくり組織として設立された。 http://www.ryokuen.gr.jp/external/rca/rca/what_rca.html

第一著者のチームでは、日本の伝統文化である茶道に着目した。作法に従ってお茶を点てる「茶会」とそこで飲まれるお茶に芸術性を見出し、お茶をふるまうロボットを制作した。

ロボットの設計にあたっては、茶道に関することを学ぶことからスタートした。実際に茶道教室にて茶会を体験し、その一連の流れやお作法、お茶の点て方について学んだ。それを元にブースで再現するストーリーを作り、ロボットを作成した。

「ロボット茶会」では、まずロボットがお菓子をゲストのもとへ運ぶ。ゲストがお菓子を食べ終わるタイミングで別のロボットがお茶を点て、茶道にしきたりにそってゲストの手元まで運ぶ。「お湯を注ぐ」「抹茶の粉を入れる」といったことは技術的に難しいことからロボットが行う作業からは外した。

図 1 に、作成したロボットを示す。

図 1　WRO2009：お茶会ロボット

ソフトウェア・ハードウェアともに難しかったのがお茶を点てる部分の制作であった。人間なら自然に行える茶筅を動かす動作をロボットで再現するために、お茶を点てる動作を詳しく観察した。茶道の先生にもアドバイスをいただきながら茶筅の描く軌道、動かす速度と時間を調整していき、人が点てたものと同様の泡を再現できるようになった。図２に、先生の点てたお茶とロボットが点てたお茶を比較した写真を示す。

図 2　お茶の点て方

LEGO Mindstorms NXT には、複数の本体を Bluetooth で接続し、データの受送信やタイミングの同期を取ることができる機能が備わっている。茶会をスムーズに進行するため、この機能を使うことに挑戦した。実際には、まず 2 台の本体を連携させる実験を行い、機能の使い方や制限を確認した上で、茶会で用いるプログラムに応用させていった。

ロボットを制作する過程では、チーム内での決定事項や開発の過程で問題となったことをメモとして残すことで、メンバー間の情報共有に役立ったほか、その後の修正にも活かされた。

国際大会への出場は果たせたが、残念ながら入賞することはできなかった。そのため、翌年再チャレンジをして入賞を果たしたいという気持ちが高まった。

翌年に開催された WRO 2010 は、「ロボットで旅行の楽しさを伝えようーあなたの国や文化遺産をロボットで紹介」というテーマが与えられた。

日本には，国際的に有名な観光スポットも多数あり、どのようなオーディエンスに向けて何を伝えるかがロボットのデザインの鍵となった。著者らは、約二ヶ月かけてテーマにマッチするアイデアを出し、具体的な設計に落とし込んでいった。アイデアを絞る過程では、日本にいる海外からの留学生にもアンケートを行い、日本についての理解を深めた。

その上で、一般的に「プリクラ」と呼ばれるような、ゲームセンターにあるフォトブースを模したロボットを制作することとした。「友達と写真を撮影する」という日本の最先端のエンターテインメントと、富士山や金閣寺といった日本の観光地を背景として組み合わせることで、日本の観光の楽しさを伝えようとした(図 3 参照)。

図 3　WRO2010 で作成したロボット

前年と同じく、ブースを訪れたゲストに体験してもらうストーリーを考えつつ、具体的なロボットの設計を行った。フォトブースで一般的な「背景を選ぶ」操作については、日本の観光地の写真と説明が書かれたカードを用意し、ロボットに差し込むことで実現した(図 4,5 参照)。

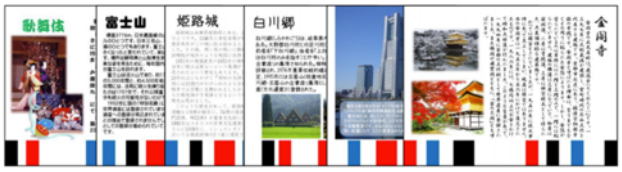

図 4　バーコードとカード

図 5 バーコードを読み取る機構

カード下部には２次元コードを設置し、ロボットのセンサーで判別を可能にした。背景に関しては、画像が印刷されたスクリーンを上下させることで変更を可能にした。背景が用意された後、インスタントカメラを用いて自動で写真が撮影されるようにした。

2.3 NXT ソフトウェアを用いたプログラミング

WRO に用いたプログラムは、全て LEGO Education によって開発されている NXT ソフトウェアを用いて開発した。NXT ソフトウェアは、教育目的で開発されている LEGO Mindstorms のソフトウェア開発ツールであり、コードを使わないビジュアルを用いたプログラミングが可能である。

NXT ソフトウェアでは、ロボットの機能であるセンサーの値の読み取り、モーターの制御、液晶画面の表示などが「ブロック」として提供されている。条件分岐やループといった制御構造は、中に他の「ブロック」を持てる特殊な「ブロック」として表現されており、センサーの状況に応じた制御を行うことも可能である（図 6 参照）。「ブロック」の中にはデータを扱うことができるものがあり、データの流れは「ワイヤー」として表現されている（図 7 参照）。扱えるデータ型は数値、真偽値、文字列の 3 種類であり、それぞれのデータを扱う「ワイヤー」には異なる色が用いられる。プログラムの見通しをよくするため、複数の「ブロック」を組み合わせて独自の「ブロック」を作ることができる「マイブロック」という機能も提供されていた。

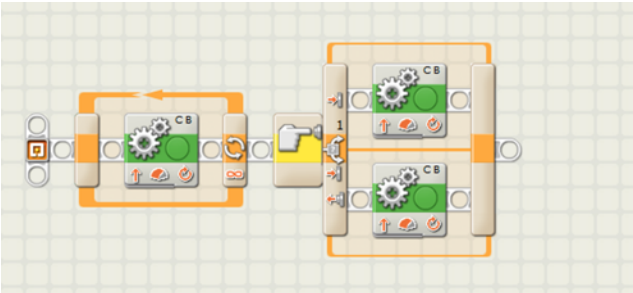

図 6 NXT ソフトウェアにおける条件分岐と繰り返し

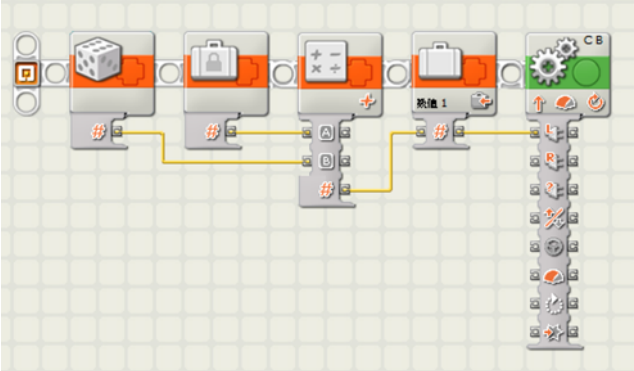

図 7 NXT ソフトウェアにおけるデータの受け渡し

3. ビジュアルプログラミング及びロボットを用いてプログラミングを学ぶメリット

3.1 前提知識や環境構築の必要がないこと

Java や Python といった汎用プログラミング言語を用いる場合、コンパイラをはじめとする実行環境を用意する必要がある。実行環境のセットアップや実際にプログラムを動作させるにあたってはコマンドライン上での操作が求められる場合も多く、環境変数といった初学者には分かりづらく、かつプログラミングと直接関係ない事項を扱わないといけない。

また、環境を構築してプログラムを実行するにあたっても、標準入出力やファイルの読み書き、プログラミング言語の細かい文法などが必要とされる。言語によって難易度に差はあるものの、セミコロンの付け忘れや閉じカッコの不足は初学者には気づきにくいものである。

初めてプログラミングを触ったとき、自らのプログラムによって目の前のロボットが動くことが嬉しく、その次のステップに進むモチベーションとなった。前提知識や環境構築を必要とせず、プログラミング自体の学習に集中できるということは、ビジュアルプログラミングを用いる一つの大きなメリットである。

3.2 制御構造やデータの流れが分かりやすく表現される

プログラミングを初めて学ぶ際に扱うコンセプトとして、条件分岐や繰り返し、変数、データ型などが挙げられる。2.3 章で示したように、ビジュアルプログラミングではこれらがブロックやワイヤーといった視覚的に分かりやすい形で表現され、コンセプトの把握に役立つ。

3.3 動作の確認が平易であること

ロボットのプログラミングでは、ロボットを意図通りに動かすことがゴールとなる。結果がターミナルや画面に表示されるプログラムに比べ、意図した通りに動作しているか簡単に確認することができる。

色や距離など、センサーからの入力を元に動作するプログラムでは、ロボットの位置やセンサーの前の色などを変更することで動作テストをすることができ、どの入力に対して想定された動作をするのか確認することも直感的であった。

一部分のみの動作の確認をすることも、ロボットのプログラミングは優れている。テキストベースのプログラムを部分実行する場合、その前段でセットされる変数などの状態を用意する必要がある。一方、ロボットを用いる場合、ハードウェアが状態を持つため、ロボットを適切な位置に

配置することでプログラムを途中から動かすことも平易である。NXT ソフトウェアには，プログラムを部分実行できる機能も標準搭載されており、プログラムが意図通りの動作をしない場合に問題を特定する方法を自然に学ぶことができた。

3.4 汎用プログラミング言語への応用

ビジュアルプログラミングで実現できることには制限があり、より多くのことを実現するためにはテキストベースの汎用プログラミング言語を学ぶことが必要になる。ビジュアルプログラミングで用いる考え方の多くは汎用プログラミング言語にも応用することができ、ビジュアルプログラミングでの経験が役に立つ。

NXT ソフトウェアにおける「ブロック」はメソッド、「ワイヤー」は値のやり取りに対応する。条件分岐や繰り返しに関しても、考え方は全く同じであるため、プログラミング言語の文法を覚えることでNXTソフトウェアを用いて学んだことをそのまま応用することができる。

ビジュアルプログラミングでプログラミングの概念を先に学ぶことで、その後に汎用プログラミング言語を学ぶハードルを下げることが出来る。

4. プロジェクトベースでプログラミングを学ぶメリット

4.1 ソフトウェア開発に求められる能力

プロジェクトを通してプログラミングを学ぶことのメリットとして、同時にソフトウェア開発の現場で必要となるスキルを身につけることがある。ソフトウェア開発には、実際にコードを書く能力だけでなく、何を作るのか具体化する要件定義の力、工数を見積もりチーム内でタスクを分割する力、成果物が実際の使用に耐えうるかテストする力などが求められる。実際のソフトウェア開発と似た工程をプロジェクトベースの学びの中で辿ることで、そうしたスキルを身につけることができる。

4.2 要件定義および設計

WRO では、著者らは，与えられたテーマからブース内で表現するストーリーを考え、それを実現するために必要なロボットを設計していった。また、必要に応じてストーリーを実現するためのメカニズムを勉強した。

2010 年に制作したフォトブースを例にすると、「ゲストに背景を選んでもらう」というストーリーを実現するために「カードを判別する」機能が必要とされ、２次元コードの仕組みについて調査をし、それに適したロボットを制作した。２次元コードで選択肢を表現する仕組みについては文章にまとめ、チーム内で共有した。

提供したい機能を元に必要となる画面や技術をリストアップし、必要に応じて実現方法を模索する一連の流れは、実際のソフトウェア開発でも頻繁に登場する。

手引きに従うような学習では、特定の技術を使って何かを作るような課題設定が行われることがある。こうした課題は、特定の技術の習熟度を評価しやすいといったメリットの反面、技術を前提として設計をすることはオーバーエンジニアリングに繋がりかねない。入門の段階から適切な技術選定を行う機会を与えられていたことは、その後の技術との向き合い方に影響を与えた。

4.3 見積もりと分担

コンテストに取り組んだことで、時間的な制約が与えられることとなり、その中で成果を出すことが求められた。ロボット本体の制作、ソフトウェアの開発、提出するレポートの執筆などを、チームメンバーの得意不得意を考慮しつつ分担していった。最終的には一つのプレゼンテーションとして発表するため、別々に開発を行うというよりも、それぞれのタスクに異なるメンバーがリーダーシップを持つような形となった。振り返ると、自然と締め切りまでの大まかなロードマップを作成し、１週間程度の単位でマイルストーンを設定していた。必要に応じてスケジュールや担当者に見直しを加えながら、作業を進めていった。

こうした進め方は、多くのソフトウェア開発で用いられているアジャイル開発と類似点が多い。

著者らは、知識からではなく経験から学び、成長していく過程で専門知識に繋げてスキルを高めてきたように思う。

4.4 動作確認と不具合の修正

WRO の準備期間の後半には、プレゼンテーション本番でロボットが期待通り動くよう、動作の確認を徹底的に行った。あらゆるケースで正しく動作することを確認し、発見された不具合には修正を加えていった。

最終的には，複数台のロボットを通信で連携させることを目標としていたが、動作の確認にあたっては、まずそれぞれのロボットが期待された動きをしていることを確認した。次に、通信部分のプログラムを実行し、正しくデータが受送信されていることを確認した。最後に、複数台のロボットを同時に実行し、全ての機能が問題なく動作することを確認した。

1 台のロボットの動作を確認することは単体テスト、データ受送信の確認は疎通テスト、全機能の確認は結合テストと考えることができる。プレゼンテーション本番で失敗できないことから、一度動作して満足するのではなく、テストの手法を用いてロボットの信頼性を向上させる努力をした。

こうした経験から、単体テストの重要性だけでなく、多角的に物を見る視点を学んだ。

4.5 チーム開発

現代のソフトウェア開発において、ソフトウェアエンジニアはデザイナーやマネージャー、ライターをはじめ多くの関係者と協働する必要がある。しかし、大学でコンピュータサイエンスを学んでいても、授業として非エンジニアと一緒に開発を行うことはほとんどない。また、チームとして開発を行う授業は、基礎的なプログラミングの能力を前提とするものばかりである。

WRO においては、非エンジニアを含むチームでの開発において必要とされる最低限のスキルを身につけることができた。話し合いの内容を議事録として残すことやお互いの作業の進捗状況を確認することは、10 年が経過した今でも欠かさず行っている。

WRO のようなコンテストには専門的なエンジニアリングの能力は要求されないため、エンジニアを目指していない人でも参加することができる。ソフトウェア開発の大まかな流れを体験しておくことは、就職後非エンジニアとしてソフトウェア開発に関わる際、役立つと考えられる。

また、作業内容を明確にすることやチーム内でタスクを分割すること、成果物の品質を確認することは、ソフトウェアやロボット開発以外の領域にも応用できるはずである。

5. ソフトウェアエンジニアの育成

5.1 ソフトウェア開発スキルの習得

第一著者は、講習会や WRO への参加を通じてソフトウェア開発に興味を持ち、中学に進学した後、独学で C, Python, JavaScript, PHP といったテキストベースのプログラミング言語を学ぶようになった。

3.4 章で述べたとおり、ビジュアルプログラミングで学んだ内容の多くを活用することができた。また、プロジェクトベースの活動を行う中で身につけた技術を必要に応じて身につける力で、少しずつ異なる言語や技術を習得してきた。

例えば、中学 3 年生のときには、学校の友人と共にスマートフォン向けのアプリケーションを作成し、ストアにて公開した。プロジェクトを始めた段階では、スマートフォンのアプリケーションを開発するための知識はほとんどなかったが、インターネットで情報を集めながら、技術要件と自らの知識を総合的に判断して使用する言語とフレームワークを選定した。

高校進学後は、業務委託やインターンなどとして主に Web アプリケーションの開発に携わった。新規にサービスを開発することも多く、WRO で行っていたことを発展させ、顧客へのヒアリングやユーザーエクスペリエンスの設計、チーム内でのコミュニケーションなどに努めてきた。

大学の進路としても、プロダクト開発に必要なエンジニアリングとビジネスのスキルを身につけることができる専攻を選択した。

5.2 ソフトウェアエンジニアを志すきっかけとして

前述の通り、第一著者が初めて講習会や WRO に参加するまで、プログラミングの経験は一切なかった。プログラミングという単語についても、「コンピュータへの指示」のような漠然としたイメージしかなく、また複雑なコードを書くものだと思いこんでいた。講習会を通して、プログラミングやアルゴリズムについて学んだことで、自らのプログラムによってコンピュータが動作することに感動した。さらに、コンテストでは自らの設計した体験をソフトウェアによって人々に届けることができると知り、社会を便利にするようなソフトウェアの開発がしたいと思うようになった。

我が国においても義務教育段階でのプログラミングが必修となり、第一著者が小学生だった頃と比べて子どもたちがプログラミングに触れる機会は増えている。教育目的のプログラミングツールも Google[6]や Apple[7]といった大手 IT 企業から公開され、アクセスもしやすくなっている。

その一方、第一著者が経験しその後の活動に影響を与えた、ソフトウェアを用いて設計を具体化する体験及びその方法を学ぶことは簡単ではない。WRO のように、ただプログラミングを学ぶだけでなく、それによって何が可能になるのか、チームでプロダクトを開発するにはどのようなスキルが必要なのか同時に学べる課題設定によって、より多くの人がソフトウェア開発に興味を持ち、また将来的にソフトウェア開発に携わるときの助けになるのではないかと考える。

6. おわりに

第一著者を例として、ビジュアルプログラミングを用いた複数人でのプロジェクトを通してプログラミングに入門することの特徴を示した。プロジェクト形式で開発を行うことで、プログラミングはもちろん、要件定義や設計、動作検証、チーム内での連携など、ソフトウェア開発に求められる様々な能力を自然と身に付けることが出来る。ビジュアルプログラミングを活用することでプロジェクトに参加するのに必要な前提知識を取り払うことができ、それを用いて学んだ知識は他のプログラミング言語にも応用することができる。そうした観点から、プロジェクトをベースにプログラミングに入門することには価値があると考える。

参考文献

[1] 久野靖, 和田勉, and 中山泰一, “初等中等段階を通した情報教育の必要性と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（TCE）*, vol. 1, no. 3, pp. 48–61, 2015.

[2] A. Pears *et al.*, “A survey of literature on the teaching of introductory programming,” *ACM SIGCSE Bull.*, vol. 39, no. 4, pp. 204–223, 2007.

[3] M. Shiozaki, S. Tsuruga, H. Nakagawa, Y. Furutani, and N. Yoshida, “企業の新入社員研修におけるプログラミング教育の実践例,” vol. 2020, no. 8, pp. 1–9, 2020.

[4] World Robot Olympiad Association, “Introduction.” [Online]. Available: https://wro-association.org/association/introduction. [Accessed: 22-Jul-2020].

[5] WRO Japan 実行委員会, “WRO Japan 2009 オープン部門ルール,” 2009. [Online]. Available: https://www.wroj.org/wp-content/uploads/2019/04/wroj2009_O_rule.pdf. [Accessed: 22-Jul-2020].

[6] “Blockly,” *Google Developers*. [Online]. Available: https://developers.google.com/blockly. [Accessed: 22-Jul-2020].

[7] Apple, “Swift Playgrounds.” [Online]. Available: https://www.apple.com/swift/playgrounds/. [Accessed: 22-Jul-2020].