

サイバーフィジカルな学習支援システムの開発

Development of Cyber-Physical Learning Support System

野口 孝文^{*1}

Takafumi Noguchi

^{*1} 釧路高専

National Institute of Technology, Kushiro College

In this paper, we describe the learning environment, where plural learners collaborate on tasks, which can be broken up into small tasks and integrated again. This learning environment is a system that supports learner participation in cooperative work after working on assigned tasks. It is possible to learn programming through robot making with LEGO. Furthermore, integrating LEGO robots makes it possible for learners to collaborate equally with each other by minimizing program changes.

1. はじめに

協調学習における学習者同士の知識交換は学習に有効である[三宅 02]。しかし、複数の学習者で課題に取り組むだけでは、一部の学習者が主導して十分な知識交換がないままに終了してしまうこともある。協調学習を効果的に行うためには、課題の切り分けが適切にできると、それぞれの学習者が切り分けた課題に取り組むことができなければならない。初心者のプログラミング教育を協調学習の対象に考えると、切り分けが可能な課題を用意することが難しいことが分かる。プログラミングの場合、動作の確認を切り分けた機能ごとに行う必要がある。そして再びそれを統合する必要がある。これらの操作はプログラミングの初心者にとって難しい。

そこで本研究では、複数の学習者が協同で課題に取り組みながらも、一つの課題を小さな課題に切り分けることができ、再びそれらを統合することができる学習環境を開発した。本学習環境は、学習者が割り当てられた課題に専念した後に、その学習結果を生かして協調作業に参加することを支援するシステムである。

これまで我々は、入出力やイメージ表示の機能をオブジェクト部品として与えることで、様々な能力の学習者が一人でプログラミングに取り組むことができる学習環境を提案してきた[野口 15]。また、課題には学習者の興味や努力によって、学習者間で結果に差異が現れるように、ゲーム制作やLEGOで製作したロボットを制御するといった多様な解があるものを取り入れてきた実績がある。

本提案では学習者は、LEGOを用いたロボット製作を通してプログラミングを学ぶ。さらにLEGOで製作したロボット同士の統合において、プログラムの変更を最小限にすることでロボット同士の関係を調整しながら、学習者同士が対等に協調作業することができるようにしている。コンピュータと実世界を連携したサイバーフィジカルな学習支援環境を作ることによって、上述の柔軟な連携を可能にしている。

本論文の2章では統合を可能にするシステムについて述べ、また、これまでのプログラミング教育について述べる。3章では、IoT化の技術を用いて統合を実現する方法について述べる。4章でまとめを述べる。

2. プログラミング学習環境

2.1 IntelligentPad システム

図1は、IntelligentPadを用いて開発した学習支援システムの様子である。IntelligentPadは、パッドと呼ばれるオブジェクトをダイナミックに組み合わせたり、変更したりできるシステムである。パッドは、ディスプレイ上に可視化され、マウスによる直接操作でパッドを自由に組み合わせることができる。パッド同士の結合は、標準化されたスロットの結合によって行い、様々な合成パッドを作ることができる[野口 95]。

我々は、多様な学習者に対応するために、入出力機能やイメージの表示等の機能をパッドで与えるとともに、C言語によるスクリプトの記述ができるパッド(手続きパッド)を用いたプログラミング教育を行っている。

2.2 一人一つの課題

プログラミングに関して幅広い能力差のある学習者に対応するために、我々はゲームプログラムを初心者のプログラミングの課題に用いてきた。その理由は、画面に登場するキャラクタの数や動き、表示デザイン、操作の仕方、得点などの組み合わせで、簡単なプログラムから複雑なプログラムまで様々なゲームを作ることができるからである。さらに、できあがったゲーム同士の違いが分かりやすいため、学習者間でコピーがし難いという利点もある。

学習者に提示しているシューティングゲームを図2に示す。図2の下に、パッドの貼り合わせを示す。このシューティングゲームは、「shoot」ボタンを押すと左下から「弾」が放物状に打ち出される。一方、「移動物体」が左

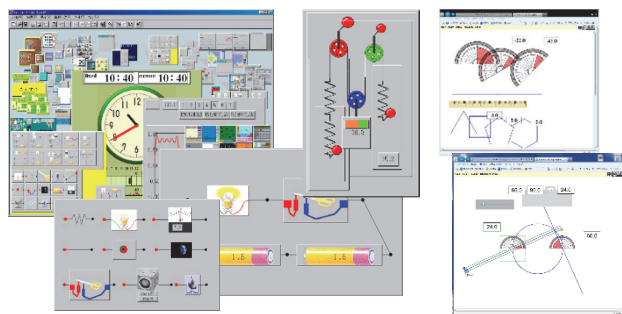


図1 IntelligentPad system

連絡先:野口孝文, 釧路高専, 〒084-0916 釧路市大楽毛西
2-32-1, 0154-57-7315, noguchi@kushiro-ct.ac.jp

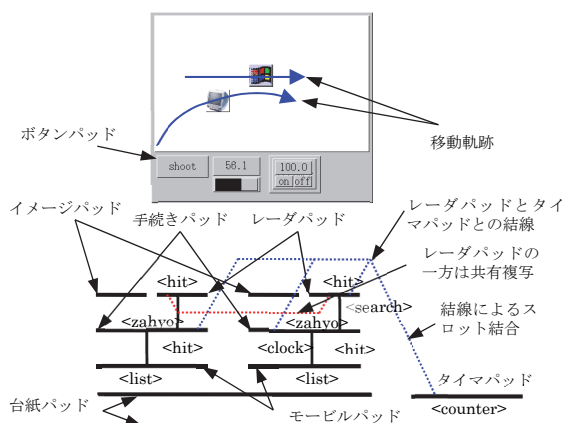


図2 シューティングゲームとパッドの貼り合わせ

端から現れる。パッド同士が当たると、下から打ち出された弾は、左下隅に消え、当てられた移動物体は、下方に落下する。絵の表示や重なり判定、移動といった機能をパッドで与えることで、学習者は、軌道の計算や衝突時の動作をプログラムに記述するだけでシューティングゲームを実現できる。

図2のシューティングゲームでは、移動するパッド（モバイルパッド）の上に手続きパッドを貼付している。複合するこれらのパッドを1つの部品（キャラクタ）として扱えるようにしている。学習者は、この複合部品（合成パッド）をコピーするだけでキャラクタを増やすことができる。またキャラクタの動作は、それぞれの手続きパッドのプログラムで行うため、他のキャラクタから独立して変更することができる。

このようなプログラムの開発方法ができるのは、各手続きパッドのプログラムがタイマパッドから送られるイベントによって駆動されているからである。次章で述べる協調学習において、複数の学習者同士が作成したプログラムを連携するときにもこの方法を用いている。

2.3 コンピュータに直接接続された機器の制御

これまで我々は、コンピュータ外部にある機器とコンピュータ上のプログラム部品とを組み合わせた学習環境を実現してきた[野口 15]。

図3は、我々の開発してきた学生実験のシステムである。LEGO (NXT) と通信する機能 (NXT コントローラ) をパッド化し、制御機能もパッドの組み合わせで作成している。このシステムでは、多様なレベルの学習者に対応するために、通信機能や表示の機能を部品として与えることで、学習者が制御プログラム作成に集中できるようにしている。

制御プログラムは、上述の手続きパッドを用いている。NXT コントローラが手続きパッドの上に貼付され、手続き

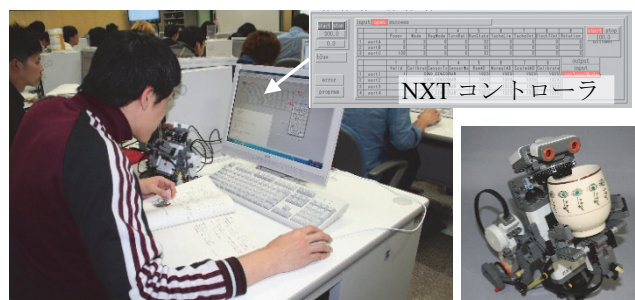


図3 LEGOで製作したロボットの制御

パッドとスロットの結合している。

図3は、お茶運びロボットとその制御プログラムの作成の様子である。NXT1 台に限定しているため、本システムを用いた実験の計画や実施は、容易である。

しかし、本システムでは、複数の NXT を含む機器を統合する場合には、NXT のポート番号を指定したり、プログラムを統合したりと操作が煩雑になるため、複数の学習者が協同して作業を進めると行った使い方に対応できていなかった。次章では、協調学習への対応について述べる。

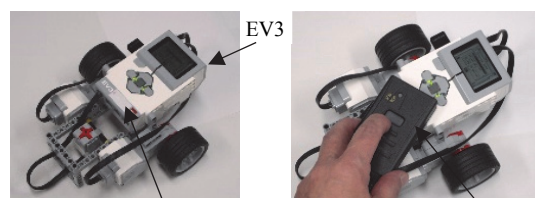
3. 個別学習から協調学習へ

3.1 IoT 化された機器とパッドの連携

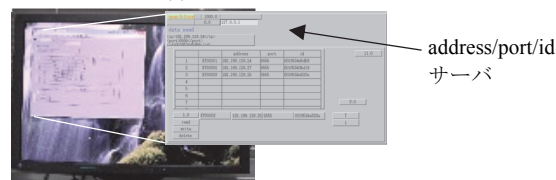
NXT を用いた実験では、PC と NXT との接続が、ケーブル接続であるばかりでなく USB 接続であったため、学習者が製作した作品同士の連携は簡単ではなかった。しかし LEGO システムが NXT から EV3 になったことにより、Wi-Fi の利用が可能になった。

図4はRFIDによってIPアドレスを管理し、即座にパッドとEV3の組み合わせができる様子を示している。図4の(a)から(c)のデータの流れを説明する。(a)のEV3にはあらかじめRFIDタグが貼付されている。(c)のRFIDコントローラのボタンをマウスクリックし、つづけて(a)のRFIDリーダーのボタンを押すと、RFIDコントローラにRFIDリーダーが読み込んだIDが取り込まれる。その後RFIDコントローラは、(b)のサーバにタグデータを送り、IPアドレス、ポート番号、EV3のIDの3つのデータを受け取る。そして、それをEV3コントローラ(クライアント)にそれらを設定する。このようにEV3は、IoT機器としてネットワークに接続されているため、(a)と(c)の操作で自由に選択することができる。

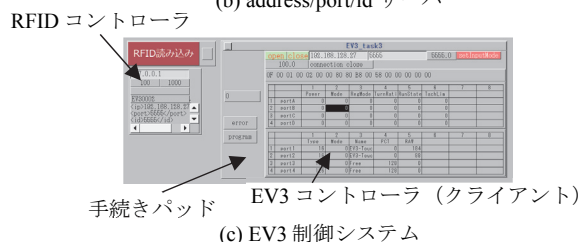
またEV3コントローラの直下には、EV3を制御するプログラムを記述した手続きパッドがある。学習者は、即座にこのプログラムをEV3に関連づけて制御することができる。



(a) RFID と ID の読み取り



(b) address/port/id サーバ



(c) EV3 制御システム

図4 RFIDを用いたEV3と制御システムとの連携

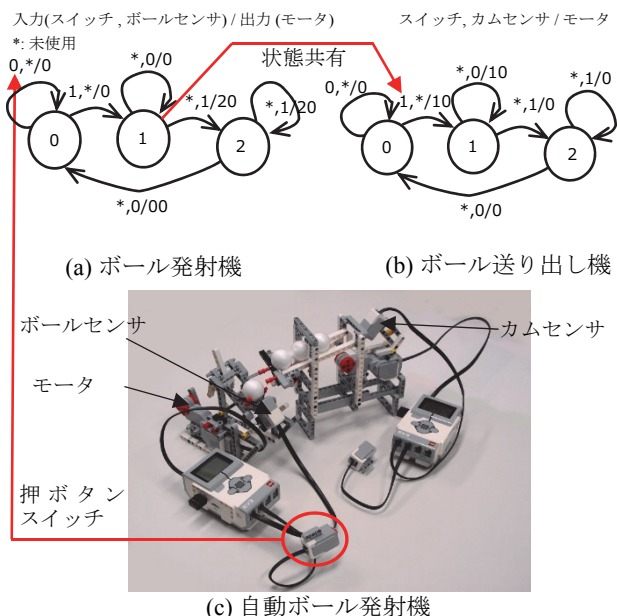


図5 状態の共有による装置の連携

プログラムの記述の仕方は、前章の2.3節で述べた方法を用いている。

3.2 個別学習と機器同士の連携

Wi-Fi 接続が可能になったEV3のシステムは、2章で示したNXTを用いたシステムと同様に、学習者が一人で取り組む課題を用意することができる。これまでと同様にLEGOでロボットを組み立て、手続きパッドに記述したプログラムを試行錯誤で動作確認しながら完成させることができる。

さらに本システムは、ロボットの制御プログラム開発において、状態遷移図を用いている。そのため、2つのロボット同士を連携させるときにも、プログラムの変更は最小限で連携を行うことができる。またEV3を制御するシステムは、IntelligentPadでできているため、LEGOコントローラを同じウインドウに共存させることができる。そして、スロットデータを介して、パッド同士を連携させることもできる。それらの様子を図5に示す。

図5は、ボールの供給装置とボール発射装置を個別に製作し統合したものである。これらの装置の動作の仕組みを説明する。それぞれの装置は、EV3とモータ、押ボタンスイッチ等からできている。プログラムは、前述の手続きパッドを用い、(a)と(b)に示した状態遷移図に基づいて作成している。それぞれの装置は、押ボタンスイッチを押すとボールの供給や、ボールの発射を行うことができる。(c)は2つの装置を連携したところを示している。2つの装置を連携したプログラムは、個別に製作した2つのプログラムを並べ、発射装置の状態をボール供給装置の制御プログラムにスロットを介して送り、その値を入力「押ボタンスイッチのON」信号の代わりに用いることで、プログラムをほとんど変更することなく実現することができる。そのことについては、3.4節で述べる。

3.3 EV3コントローラと制御プログラム

前節で述べたプログラムの連携が容易な理由を、本節ではEV3の制御方法から、次節ではプログラム同士の連携方法から説明する。

EV3の制御は、3.1節で示したEV3コントローラを介して

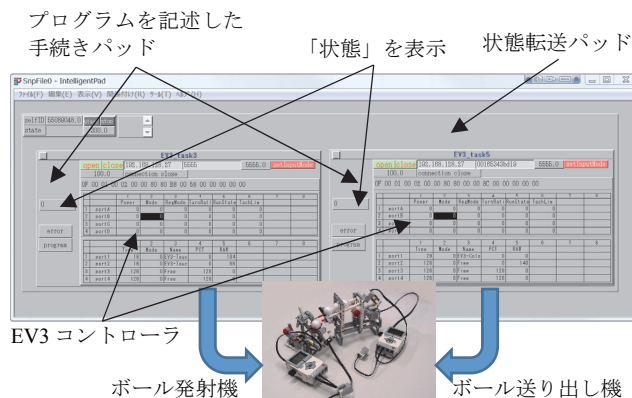


図6 コンピュータ上の連携

行っている。EV3に接続されたセンサからのデータの取り込み方法とモータやランプへのデータの出力方法は以下の通りである。

図4の(c)に示したEV3コントローラでは、入力の状態と出力の状態をパッドの表面に表形式で表示している。EV3に接続された各種センサ入力は、一定時間間隔でEV3コントローラに取り込まれ表に表示するとともに、スロット「input」のデータを更新するようにしている。取り込み間隔は、あらかじめ100msecにしているが変更することもできる。このスロット「input」は、下方の手続きパッドに結合されている。手続きパッドでは、一定時間ごとにスロット「input」に対応したアクション関数が（イベントによって）起動されることになる。

一方、EV3コントローラへの出力は、スロット「output」に値を書き込むと即座にEV3に送られるようにしている。したがって、EV3を制御するプログラムは、図5の(a)や(b)に示した状態遷移図に従ってプログラムを記述することで実現できる。EV3から送られてくる入力イベントを処理するアクション関数において、状態遷移図の状態に応じた出力値を作りEV3に送ることで簡単に制御できる。

切り分けられた個別の学習では、学習者は上述のように状態遷移図に基づき、ロボットの組み立てと調整をしながら学習を進めることができる。次節では、このようにして作られた複数のプログラムの統合について述べる。

3.4 プログラムの統合

図5に示した2つの装置について、コンピュータ上でのプログラムの統合の様子を図6に示す。3.2節でも述べたように2つのプログラムの統合は、状態遷移図に対応した一方の状態を他方に伝えることで実現できる。本システムでは、状態転送パッドを作りこれを実現した。

状態転送パッドは直上に貼付した複数のEV3コントローラの状態を取り出し、あらかじめ指定（状態転送パッドにスロット名を記述）されたEV3コントローラのスロット（後に示すプログラムでは「states」）に設定する。このときのデータは、それぞれのEV3コントローラのラベル名をタグにして付けることで、区別できる様になっている。それぞれのEV3コントローラは独立に動作しているが、他のEV3コントローラの状態を受け取ることによって、自身の状態を変更したりEV3の出力ポートに値を出力したりする。

図7に制御プログラムの内、プログラム同士の連携のために変更した一部分を示す。図7のプログラムの7行目から始まる関数は、EV3からの入力イベントによって起動するアクション関数である。9行目から12行目は、EV3に接

```

1 slot Dictionary input, output, states;
2 slot String state;
3 Void state_0(Number);
4 Void state_1(Number);
5 Void state_2(Number);
6 Void outData(Number);
7 action input()
8 {
9     Dictionary port_in;
10    Number data0,data1;
11    port_in=getDictionary(input,"port1");
12    data0=getNumber(port_in,"PCT");
13    if(getString(states,"EV3_task5")=="1") data1=1;
14    else data1=0;
15    if(state=="0") state_0(data1);
16    else if(state=="1") state_1(data0);
17    else if(state=="2") state_2(data1);
18 }

```

図7 プログラム同士の連携

続されたセンサの値を取り出すプログラムである。取り出した値は、スロット「data0」に入力され、単独の装置を制御しているときには、15行目から17行目に示すように、状態に応じた処理を行う関数に値を渡している。なお、スロット「state」は、状態遷移図に対応した自身の状態を保持している。また、スロット「states」は、自身の状態も含め他のプログラムの状態を保持している。この値は、状態転送パッドによって、タグ付きデータとして保存されたものである。

図7のプログラムは、複数のプログラムと連携しているときに、プログラムの13行目でタグが「EV3_task5」と付けられた装置の状態を取り出している。さらにその値が1のときは、スロット「data1」に値1を、それ以外ときには0を設定している。15行目と17行目ではこれを、自身のEV3に接続されたセンサの値の代わりに用いている。プログラムの3行目から6行目は、関数のプロトタイプ宣言で、15行目から17行目でそれらの関数を呼び出している。これらの関数は、状態遷移図の状態に対応していて、EV3および他のプログラムの状態によって、EV3に値を出力したり状態を遷移させたりする機能が定義されている。

本システムでは、このように数行のプログラムの変更で、複数のプログラム同士の連携を実現している

3.5 個別学習から協調学習へ

1章でも述べたように、我々は初心者プログラミング教育であることを考慮して、学習者の興味を引く課題として、LEGOを用いたロボットの制御を取り入れてきた。またプログラムとロボットの動作の関係を分かりやすくするため、プログラムを変更すると即座に反映する様にNXTとPCをケーブルで接続している。EV3では、ケーブルの代わりにWi-Fiを用いている

そして学習の課題をロボットの製作と制御にすることによって、これまで個別の学習で行ってきた課題を統合した課題を作ることができるようになった。3.2節でも示したように、複数の機能を組みあせた装置の製作を課題として与えることができるようになったことで、学習者同士で作業を分担し再び統合することが可能になった。

学習者は、それぞれの小さな課題を実現することによって、プログラムや装置のことが理解できているため、統合時にはそれぞれの経験を基に協調学習ができると期待できる。このことは、初めから協同で作業を行う従来の学習と

は、異なっている。

4. おわりに

本学習支援システムは、従来から行ってきた個別学習を変更することなく協調学習へシームレスに移行できる。従来の我々の個別学習は、個々の学習者ができるだけ他の学習者に頼らずに課題を解決することを目的に行ってきた。いっぽう、協調学習には1章でも述べたように、課題に対する問題解決に至るまで、より多様な視点から見た試行錯誤が行われると期待できるという利点がある。

本提案は、学習方法がシームレスに移行できるという点ばかりでなく、個別の課題解決で基本的な知識が育成されている点にも従来にない利点があると我々は考えている。

また、提案したシステムは実世界にあるIoT化された機器とコンピュータ上の教材を学習者自身が組み合わせたり編集したりできるサイバーフィジカルな学習環境である。

謝辞

本研究の一部は、科学研究費基盤研究(C)(一般)(16K01150)を受け推進している。

参考文献

- [三宅 02] 三宅なほみ, 三宅芳雄, 白水始: 学習科学と認知科学, *Cognitive Studies*, Vol. 9, No. 3, pp. 328-337, 2002.
- [野口 95] 野口孝文, 田中譲: コンストラクションセットを持つマイクロワールド, *情報処理学会論文誌*, Vol. 36, No. 1, pp. 152-166, 1995.
- [野口 15] 野口孝文, 千田和範, 稲守栄: 初心者から上級者までシームレスにプログラミングを学ぶことができる持続可能な学習環境の構築, *教育システム情報学会論文誌*, vol. 32, No.1, pp.59-70, 2015.