

プログラミングを利用したネットワーク学習の試み*

西ヶ谷 浩史[†], 紅林 秀治[‡], 兼宗 進[§]

藤枝市立青島中学校[†] 静岡大学 教育学部[‡] 一橋大学 総合情報処理センター[§]
YRR02165@nifty.com[†] eskureb@ipc.shizuoka.ac.jp[‡] kanemune@cc.hit-u.ac.jp[§]

概要

中学校におけるプログラミングを利用したネットワーク学習について報告する。情報化社会を生きる生徒にとって、電子メールや WWW に代表されるネットワークやソフトウェアの仕組みを知ることがますます重要になっている。今回は中学校技術・家庭科「情報とコンピュータ」の中で、ネットワークを使ったプログラミングを体験する授業を行った。その結果、生徒たちはネットワークに対する関心と学習の意欲を高め、ネットワークの仕組みを体験的に学ぶことができた。

1 はじめに

中学校におけるプログラミングを利用したネットワーク学習について報告する。情報化社会の中で、生徒たちはコンピュータや携帯電話の上で電子メールや WWW を利用しながら生活している。このような計算機やネットワークの利用は、生活の中で今後ますます重要度を増すと考えられる。

現在行われている情報教育では、現時点での機器やサービスの使い方についての学習が主流になっている。しかし、今後の技術革新による変化に対応できるようにするためには、将来を見据えた基本的な原理の教育が必要になる。

筆者たちはプログラミングを情報教育に取り入れることにより、「与えられたインストラクションの通りに動作する」という計算機のもっとも基本的な原理を体験的に学習するモデルを提案し、その効果を示してきた。今回はプログラミングをネットワークの学習に取り入れることにより、「ソフトウェア同士がデータを交換しながら動作する」というネットワークのもっとも基本的な原理を体験的に学習するモデルを提案し、その効果を検証する。

今回の授業では、プログラミング言語としてドリトル [7] を採用した。中学校であれば Squeak [1] や MindStorms [11] などの図形的な言語を使用し

なくても、キーボードからプログラムを入力して記述することが可能である。

2 プログラミング言語「ドリトル」

2.1 言語仕様

ドリトル [5][10] は教育用に設計されたオブジェクト指向言語である。簡潔な日本語による構文を採用しており、オブジェクトに呼び掛ける形でプログラムを記述できる。以下、授業で扱ったサンプルを用いてドリトルを解説する。

図 1 に、描いた五角形が画面上を移動するサンプルプログラムを示す。

```
カメ太 = タートル! 作る。  
五角 = 「カメ太! 100歩 歩く 72度 左回り!」 5回  
繰り返す 図形にする。  
時計 = タイマー! 作る 0.1秒 間隔 10秒 時間。  
時計! 「五角! 10 0 移動する」 実行。
```

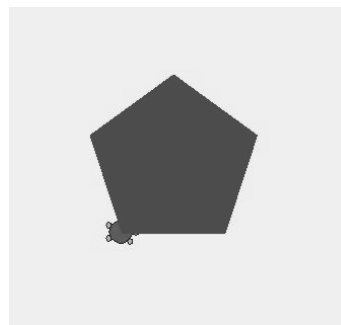


図 1: ドリトルのプログラムと実行例 (1)

*An Experience of Network Education using programming, Hirofumi Nishigaya[†], Shuji Kurebayashi[‡], Susumu Kanemune[§] (Aojima Junior High School[†], Shizuoka University[‡], Hitotsubashi University[§])

カメ太 = タートル! 作る。

タートルオブジェクトに “! 作る” を送り、画面に “カメ太” というタートルオブジェクトを作る。タートルオブジェクトを操作して画面に軌跡の線を残すことで、様々な図形を描くことができる。

五角 = 「カメ太! 100歩 歩く 72度 左回り」! 5回
繰り返す 図形にする。

“「カメ太! 100歩 歩く 72度 左回り」” はプログラムのまとめり (ブロック) を表す。ここでは “! 5回 繰り返す” を送ることでブロックに自分自身を 5 回実行させる。結果として、“カメ太! 100歩 歩く 72度 左回り” が 5 回実行される。

ドリトルのオブジェクトがメッセージ (命令) を実行すると、多くの場合は自分自身を結果として返す。ここでは最後に実行された “カメ太” がブロックの実行結果となる。それに対して “図形にする” を送ることで、“カメ太” から軌跡の線を切り離して新しい図形オブジェクトを作り変数 “五角” に代入する。

結果として、“五角” という名前の図形オブジェクトが作られる。

時計 = タイマー! 作る 0.1秒 間隔 10秒 時間。

タイマーオブジェクトはプログラムを定期的に行うオブジェクトである。ここでは “時計” という名前のタイマーを作り、実行間隔を 0.1 秒、実行時間を 10 秒に設定している。

時計! 「五角! 10 0 移動する」実行。

“時計” を使い、五角形を画面上で動かす。タイマーを利用することで、生徒たちは様々なアニメーションを作ることができる。

また、ボタンオブジェクトを利用すると、さらに、生徒達の発想は広がる。画面上に自分で作ったボタンを押して、操作することができるようになるので、生徒は非常に興味を持つ。

図 2 に、ボタンオブジェクトを利用した生徒のプログラムを示す。“//” で始まる行はコメントである。

ボタン1 = ボタン! ”回転” 作る。

“ボタン1” という名前のボタンオブジェクトを作っている。

ボタン1: 動作 = 「カメ太! 90度 右回り」。

“ボタン1” が押されたときに実行する動作を設定している。“ボタン1” が押されると

```
カメ太 = タートル! 作る。  
//カメの向きを変える  
ボタン1 = ボタン! ”回転” 作る。  
ボタン1! -200 100 位置 100 50 大きさ。  
ボタン1: 動作 = 「カメ太! 90度 右回り」。  
//カメを前進させる  
ボタン2 = ボタン! ”前進” 作る。  
ボタン2! -200 0 位置 100 50 大きさ。  
ボタン2: 動作 = 「カメ太! 50歩 歩く」。
```

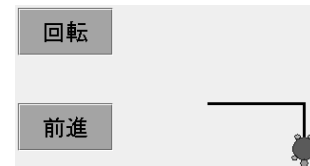


図 2: ドリトルのプログラムと実行例 (2)

“「カメ太! 90度 右回り」” が実行され、結果として “カメ太” が右に 90度回転する。

2.2 ネットワーク機能

ドリトルでは、オブジェクトを変数 (プロパティ) や配列に格納し、取り出して使う。ドリトルをネットワークに拡張した分散共有ドリトル [6] のモデルでもこの概念を継承し、オブジェクトをサーバーに格納し、取り出して使う形で、初心者がオブジェクトの転送を容易に扱えるようになっている。

分散共有ドリトルの実行時には、ネットワーク上にオブジェクトサーバーのプロセスを起動しておく。オブジェクトサーバーには、名前を付けてドリトルのオブジェクトを登録することができる。

オブジェクトの複製は、サーバからオブジェクトの複製を取り出す。取り出されたオブジェクトは通常のローカルなオブジェクトとなる。この機能はサーバを介して複数のドリトル環境 (クライアント) 間でオブジェクトや値をやりとりするために用いる。

図 3 に、オブジェクトサーバーにオブジェクトを登録・複製する様子とプログラム例を示す。

カメ太 = タートル! 作る。

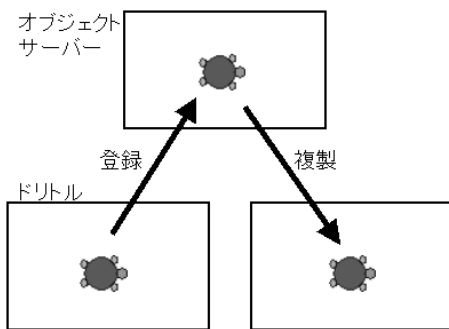
ローカルにタートルオブジェクトを生成する。

サーバー! ”sv1” 接続。

サーバー “sv1” に接続する。サーバーはホスト名または IP アドレスで指定する。

サーバー! ”kame1” (カメ太) 登録。

ローカルのオブジェクト “カメ太” を、“kame1”



カメ太 = タートル！作る。
 サーバー！"sv1" 接続。
 サーバー！"kame1" (カメ太) 登録。
 カメ吉 = サーバー！"kame1" 複製。
 時計 = タイマー！作る
 時計！「カメ太！10 歩く。カメ吉！15 歩く」実行。

図 3: オブジェクトの登録と複製

という名前でサーバーに登録する。この結果、サーバーには "kame1" という名前で "カメ太" の複製が登録される。

カメ吉 = サーバー！"kame1" 複製。

サーバーから "kame1" を複製し、ローカルにカメ吉という名前のオブジェクトを作る。

時計 = タイマー！作る

タイマーオブジェクトを生成する。

時計！「カメ太！10 歩く。カメ吉！15 歩く」実行。

ローカルのオブジェクト "カメ太" とサーバーから複製したオブジェクト "カメ吉" を同時に動かす。サーバーから複製したオブジェクトは、ローカルのオブジェクトと区別することなく操作することが可能である。

3 ネットワークに関する授業実践

3.1 学習の流れ

平成 15 年度から平成 16 年度にかけて、2,3 年生の技術・家庭科の中で授業を行った。カリキュラムを表 1 に示す。

2 年生では 11 月から 3 月までに週 2 時間を使い、ドリトルのプログラミングと自走ロボットの製作を扱った。3 年生では 4 月から 3 月までに隔週で 1 時間を使い、ドリトルを用いた自走ロボットの制御、単独で動くゲームプログラミングと、それをネットワークに拡張した共同作成プログラミングを扱った。

このカリキュラムにより、技術・家庭科の技術領域の中で、「自走ロボットの製作」というもの作りの要素と、「制御、プログラミング、ネットワーク通信」という情報基礎の要素をバランスよく実現することができた。

授業では 1 人 1 台のコンピュータを使用した。仕様を示す。

- CPU: Celeron400MHz
- メモリ: 64MB
- OS: Windows98

今回の題材はゲームプログラムであるため、動作速度が重要になる。しかし、コンピュータのスペックが十分でない環境であり、特にメモリが少なかつたため、メモリの使用量を増やさないよう簡潔なプログラムを指導した。また、教室内の LAN は 10BASE-T のダムハブに 30 台以上が接続された帯域を確保できない環境であるため、サイズの小さい数値オブジェクトを使い通信量を減らす工夫をした。

以下では、3 年生で扱った 2 種類のプログラミングの授業を解説する。

3.2 個人ごとのプログラミング

3 年生の授業では、制御プログラミングの後で 2 種類のプログラミングに取り組んだ。

最初はネットワークを学習する前段階として、一人でやるピンポンゲームを作成した。このプログラムでは、ドリトルの衝突命令とボタンオブジェクトを利用して、パドルを動かしながらゲームを行う。図 4 にピンポンゲームの作品例を示す。

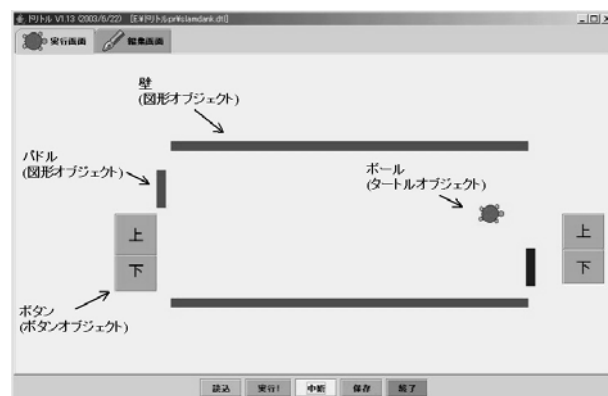


図 4: ピンポンゲーム

このプログラムでは、タートルがボールの役割をし、タイマーによるアニメーションにより画面上を移動する。ボールが壁にぶつかった場合には、

表 1: 授業カリキュラム

【2年生】		
単元	内容	時間数
ドリトルでプログラミング	オブジェクトの変身からタイマーまで	6
	個人作品制作	4
ロボットの製作	2軸制御ロボットの製作	15
【3年生】 (今回の紹介範囲)		
単元	内容	時間数
ロボットの制御	制御プログラムの基本と転送方法	1
	簡単な課題をクリアーする	1
	迷路1と迷路2に挑戦	4
ドリトルでプログラミング	ボタンを作り画面上のオブジェクトを動かそう	1
	ボタンオブジェクトを使ってピンポンゲームを作ろう	2
ドリトルで共同プログラミング	友達とネットワークを利用したゲームを作ろう	5
	自分たちの作ったプログラムを発表しよう	1

壁の役割をする図形オブジェクトからタートルに“衝突”命令が送られ、タートルはぶつかった壁に応じて自分の向きを変更する。

タートルの移動する方向は、X方向とY方向の向きを符号(+/-)で管理している。初期設定は「X方向=1、Y方向=1」となっているが、上下の壁にぶつかったときはY方向の向きを逆転し、「X方向=1、Y方向=-1」とすることで、縦の跳ね返りを実現する。

タートルがパドルにぶつかったときは、向きを変えると同時に得点を加算することで点数をカウントすることができる。

生徒たちは思い思いのデザインでパドルの形や位置を設定しながらプログラムを作っていた。

3.3 共同プログラミング

次に「自分たちのピンポンゲームをもっとおもしろくするにはどうしたらよいか」を考えさせた。当初、生徒たちはボールのスピードやパドルの大きさに着目したが、教師が2台のパソコンを並べてタートルオブジェクトが隣の端末に移動するデモを見せたところ、「すごい」という驚きの声上がり、「そんなことができるのか」と強い興味を示した。

そこで、次のようにネットワークについて説明した。

- ネットワークには郵便の住所に相当する番地が振られ、IPアドレスと呼ばれている。
- 教室のネットワークでコンピュータ同士が通信するときは、相手のIPアドレスによって通信相手を指定できる。
- ネットワークはインターネットという形で世

界中とつながっている。

- 世の中ではWWW、電子メールなど多くのことがサービスを提供するコンピュータ(サーバー)によって提供され、そこを起点として複数のコンピュータが通信している。

続けて、「このネットワークを利用して、友達と共同で一つのゲームを作ってみよう」という学習課題を提示した。最初に、生徒たちは自分のIPアドレスを調べた。そして、友だちとペアを作り、1人がサーバーを起動した。各自のプログラムの接続先をサーバーを起動した端末のIPアドレスに変更して実行することで、ペアを組んだ生徒同士の端末が通信できることを確認した。図5に授業の様子を示す。



図 5: 授業の様子

授業は3つのステップで進めた。

1. パドルが動くようにボタンオブジェクトをつける。
2. ボールが画面からはみ出て消えないように、上の壁、下の壁をつける。
3. 得点が表示できるようにする。

図 6 に生徒の作品例を示す。生徒たちは単独で動くピンポンゲームを拡張し、ネットワークで通信しながら動作するプログラムを完成させた。生徒たちはネットワークに対応するために以下の修正を行った。

サーバー！”横位置” 100 登録。

サーバーを介してボールの座標と向きを共有するようにした。サーバー上には“横位置”、“縦位置”、“向き横”、“向き縦”などの値を置く。上はサーバー上の“横位置”にボールの X 座標の初期値 100 を登録するコードのサンプルである。

時計=タイマー!作る 0.1 間隔「ボール!移動」実行。

ボール：移動 = 「x = サーバー！”横位置”複製」。

サーバーから最新の座標を得て、定期的に画面の適切な位置にボールを表示するようにした。上は 0.1 秒ごとにボールに“移動”命令を送るタイマーオブジェクトと、ボールの“移動”メソッドの定義である。

ボール：移動 = 「サーバー！”横位置” (x + 10 * xdir) 登録」。

一方のプログラムから、定期的にボールの位置を移動させるようにした。この処理では、サーバーから最新の座標と向きを取得し、計算した新しい座標をサーバーに登録する。上はボールの“移動”メソッドの定義である。X 座標 (x) を移動距離 (10) と向き (xdir) から計算し、サーバーの“横位置”に代入している。

パドル：衝突 = 「サーバー！”向き横” 1 登録」。

壁やパドルに衝突した場合には、新しい向きをサーバーに登録するようにした。上はパドルの“衝突”メソッドの定義である。ボールの向きを右向き (1) に設定している。

4 考察

生徒は毎回の授業で感想を記録した。また、授業の終了時には、理解度と難易度の 2 種類のアンケートを行った。これらの結果から、中学生が今回の授業を通して「プログラミングを通して体験的にネットワークを学ぶ」ことが可能であることを確認した。

4.1 理解度の分析

表 2 に理解度のアンケート結果を示す。

表 2: 授業アンケート (理解度)

質問	回答 (人)
ネットワークに興味を持てた	185 (93%)
ネットワークの仕組みを理解できた	150 (75%)
単独プログラミングは難しい	118 (59%)
単独プログラミングは楽しい	185 (93%)
共同プログラミングは難しい	151 (76%)
共同プログラミングは楽しい	186 (93%)

表 2 と生徒の感想から、生徒たちは次の概念を理解したことを確認した。

(1) ネットワークへの興味を持てた

表 2 から、93%の生徒がネットワークに興味を持てたと回答した。生徒の感想でも、「インターネットは調べるだけのものと思っていたが、いろいろなことができ、自分でもプログラムを作れるのだとわかった」という感想や、「生活が便利になる」「音楽配信が行われると聞いてすごく便利だと思いました」といった生活と結び付けた感想が多く見られた。

また、「友だちのパソコンにファイルを送れるのは便利だけど、個人情報が出たりデータを覗き見されることがあるので注意が必要になる。ネットワークを買い物や交流に利用するときも用心深く利用する必要があると思う」という感想から、生徒たちはネットワークの原理を理解したことにより、ネットワークの利便性とリスクについても類推して理解することができた。

(2) ネットワークの仕組みを理解できた

表 2 から、75%の生徒がネットワークの仕組みを理解できたと回答した。生徒の感想でも、「IP アドレスに相手のアドレスを入れた」というように IP アドレスによって通信の相手を指定できることや、「無線 LAN の暗号化の技術はすごいと思った」のようにセキュリティ技術が使われていること、そして「メールは直接相手に送られると思っていたが、実際にはサーバーを経由して情報が送られている」というサーバーの仕組みなどについて理解していることを確認した。

(3) 共同プログラミングの楽しさを理解できた

表 2 を見ると、単独プログラミングは 59%が難しいと回答し、93%が楽しいと回答している。一方、ネットワークを使った共同プログラミングは



図 6: 対戦型ピンポンゲーム

単独より多い76%が難しいと回答しているが、単独と同数の93%が楽しいと回答した。

単独のプログラミングでは、「自分で工夫したり、やりたいようにプログラミングできて楽しかった」という感想にあるように、自分の判断でプログラムを作っていける楽しさを感じていた生徒が多かった。

一方、ネットワークを使った共同プログラミングでは、2人の生徒が通信し合うプログラムを作成する。そのため、相手のプログラムを含めたデバッグ作業が必要になり、プログラミングの難易度が高まった。しかし、楽しさの度合いは変化しておらず、これは難しいにも関わらず、楽しさを感じていたことを示している。

「難しいがおもしろい」という傾向はロボット制御を取り入れたプログラミング学習 [8][9]でも観察されたが、今回のプログラミング学習では、それに加えて共同でプログラムを作ることに興味を感じていた点が異なっている。これは生徒の感想の中に「共同で作ったときはお互いの考えを出し合い組み合わせるゲームを作っていました」とあるように、お互いに自分の考えを出し合いながら学習できるところにおもしろさを感じていることがわかる。

今回の授業では、通信し合う形のプログラムを扱うことで、生徒たちは互いに仕様を相談したり、デバッグや動作確認までの作業を会話をしながら進めていった。ネットワークを利用したプログラミング学習を行うことで、個別学習になりがちなコンピュータ学習を、生徒同士がコミュニケーションしながら進める共同学習へと転換することができる。今後の授業でも、さまざまな形でコンピュータを利用した共同学習を取り入れていきたい。

4.2 難易度の分析

授業後の生徒のアンケートから、今回の授業で扱った内容についての難易度を分析した。表3にアンケート結果を示す。

表 3: 授業アンケート (難易度)

授業	作業内容	達成者 (人)
単独作業	ボタン・パドル	185 (95%)
	ボールを跳ね返す	170 (88%)
共同作業	ボタン・パドル	181 (93%)
	接続・友達のボール	152 (79%)
	上の壁	142 (74%)
	下の壁	125 (65%)
	得点の表示	66 (35%)
	ボールのスピード	71 (37%)

この結果から、「ボタン・パドルの操作」「ボールの跳ね返り」といった基本部分は、多くの生徒が問題なく扱っていたことがわかる。ネットワークを使ったプログラムの学習を進める上で、ピンポンゲームの基本部分を中心に授業を組み立てたことは生徒にとって有用であった。

一方、「得点の表示」と「ボールスピードを変える」ことは生徒にとって難易度が高かった。これらはネットワークの学習にとって本質的な部分ではないため、今後のカリキュラムでは応用問題としての位置付けで扱うことを考えたい。

5 関連研究

ネットワークを利用したプログラミングは Imagine[2][4]でも実現されている。Imagineは Logo 言語 [3]であり、数値などの基本データとともにタートルなどのオブジェクトを扱うことができる。

ネットワークで通信を行う場合には、1台の端末がサーバーになり、他の端末はクライアントに

なる。ユーザーからは P2P(Peer to Peer) に見えるが、1 台のクライアントがサーバーを兼ねている形であり、ドリトルのネットワーク方式と本質的な違いはない。

通信は、データ(またはオブジェクト)の転送によって行われる。送り側は相手を指定してデータを送信する。受け側では、データが届いたときにイベントが発生する。データの種類に応じた処理を定義しておくことで、特定の手続きを実行することができる。

「データをサーバーに登録し、サーバーからデータを取得する」というドリトルのサーバー経由のモデルと比較すると、クライアント間でデータが直接送り付けられる形をとるため、受け側ではデータに応じたイベント処理をあらかじめ記述しておく必要があり、生徒にとって難易度が高い。

また、すべてがオブジェクトであるドリトルと比較して、Imagine では旧来の Logo から存在したデータがオブジェクトとは独立して存在する。そのため、ネットワークを扱う際に送受信命令とデータ受信時のイベント定義においてデータとオブジェクトを区別して記述する必要があり、生徒の負担が大きくなることが考えられる。

6 今後の課題

今回の授業では、生徒のプログラムの間で数値オブジェクトを交換した。今後の授業では、メソッドという形でプログラムを埋め込んだタートルオブジェクトを交換して再利用するなど、共同プログラミングの可能性を追求したい。自作のオブジェクトを登録しダウンロードして再利用することで、教室内だけでなく、インターネットを通じて世界中の学校と共同プログラミングを行うことが可能であると考えている。

自作オブジェクトの交換は、情報化社会に参画する態度の育成にも利用できる。現在行われているモラル教育では、生徒を利用者に限定する形で著作物の利用を制限する方向に指導することが多い。一方、自作オブジェクトを交換する授業では、生徒は生産者と利用者の両方の立場を経験することができるため、自分たちの権利を守るものであるという意識を持ちながら学習することが可能である。ネットワークやプログラミングの学習にとどまらず、今回の体験を広くさまざまな学習を行うためのモデルに発展させていきたい。

7 結論

ドリトルのネットワーク機能を利用して、プログラミングをしながら体験的にネットワークを学習する授業を行った。中学校技術・家庭科の授業の中で、生徒たちはネットワークについて興味を持ち、社会における利用の可能性やリスクを理解した。また、プログラミングを通して IP アドレスやサーバーの意味や役割を理解しながら、「ソフトウェア同士がデータを交換しながら動作する」というネットワークの基本的な仕組みを学ぶことができた。

謝辞

この研究の一部は、日産科学振興財団、上月スポーツ・教育財団および経済産業省「IT クラフトマンシップ・プロジェクト」の助成を受けて行われました。筑波大学ビジネス科学研究科久野靖教授をはじめ、アドバイスをいただいた方々に感謝いたします。

参考文献

- [1] Alan Kay. Etoys and simstories in Squeak. <http://www.squeakland.org/author/etoys.html>.
- [2] Logotron. Imagine logo. <http://www.logo.com/cat/browse/logo.html>.
- [3] Seymour Papert. *Mindstorms : children, computers, and powerful ideas*. Basic Books, 1980.
- [4] L'ubomir Salanci. Networking in logo. In *EuroLogo2001*, 2001.
- [5] 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), pp. 78–90, 2001.
- [6] 兼宗進, 中谷多哉子, 御手洗理英, 福井眞吾, 久野靖. 端末を飛び出したオブジェクト: 分散プログラミングを活用した情報教育の提案. 情報処理学会 情報教育シンポジウム (SSS2003), pp. 91–98, 2003.
- [7] 兼宗進. プログラミング言語「ドリトル」. <http://kanemune.cc.hit-u.ac.jp/dolittle/>.

- [8] 紅林秀治, 兼宗進, 岡田雅美, 佐藤和浩, 久野靖. 画面を飛び出したオブジェクト: 自立型ロボットを活用した情報教育の提案. 情報処理学会 情報教育シンポジウム (SSS2002), 2002.
- [9] 紅林秀治, 兼宗進. プログラミング学習についての一考察: ロボット制御のプログラミング学習とソフトウェア作りのプログラミング学習を比較して. 情報処理学会 情報教育シンポジウム (SSS2004), 2004.
- [10] 中谷多哉子, 兼宗進, 御手洗理英, 福井眞吾, 久野靖. オブジェクトストーム: オブジェクト指向言語による初中等プログラミング教育の提案. 情報処理学会論文誌, Vol. 43, No. 6, pp. 1610–1624, 2002.
- [11] 増井俊之. インターフェースの街角 (13) LEGO の MindStorms. *UNIX Magazine*, No. 12, pp. 164–169, 1998.