

模擬情報システム構築による情報教育と実習環境開発

青木 浩幸^{1,2}, 横山 節雄¹, 宮寺 庸造¹

¹東京学芸大学, ²相模原市大沼小学校

東京都小金井市貫井北町 4-1-1

aokih@cs.u-gakugei.ac.jp, yokoyama@cs.u-gakugei.ac.jp, miyadera@cs.u-gakugei.ac.jp

概要

高等学校の情報教育において模擬的な情報システム構築実習を授業に取り入れることを提案する。学習者は身の周りにある情報システムの仕組みを考え、プログラミングによりコンピュータに動作を指示することで模倣したシステムを構築する。このようなものづくりの活動は、高度に進化し理解が困難になった情報社会の中で学習者に情報技術の実感的理解と情報社会への主体的な態度を身につけさせることが期待でき、全ての生徒に受けさせるべき情報教育であると考え。本稿ではシステム構築実習が高等学校教科情報の中でどのような効果をあげられるかを考察し、そのような実習を実現するための環境について説明する。

1. はじめに

現代の情報社会の広がり発展にあわせ、高等学校普通教科情報（以降「教科情報」とする）は職業教育ではなく万人が学ぶ教科として2003年より実施されている。

現状では教科で教えられている内容は「情報A」にあたる情報活用の実践力の育成に重きが置かれ、ExcelやPowerPointといったアプリケーションソフトによる情報処理実習が多くの学校で行われている[1]。現代の情報社会に生まれてきた生徒たちは高度に進化し理解が困難な情報技術に囲まれて生活している。もし情報技術を「習った方法に従って利用する」経験ばかりであれば、苦手意識を生むことにもなりかねない。

情報技術に対する不安を解消するには実践的利用を進めるより仕組みを理解するような系統的知識の獲得が有効であるという報告があるように[2]、情報技術の仕組み、その成立過程をいかにして教えることができるかが情報教育の重要な課題と言えるだろう。

情報の科学的な理解のため観察実習により体

験的に学習させている取り組みもあるが[3]、一般的には教師による説話や調べ学習が中心である。それは、情報というものが実体の無いものであるということ、そして実物は高度すぎて理解困難であるという事情が原因である。

対象に対する正しい概念を学習者に構築させるためには、対象に実際に触れ試行錯誤する過程（実験や体験、ものづくりの活動）が必要である。実物が困難であれば模擬的な体験で代用すればよい。

情報技術の仕組みの理解を目的として、模擬情報システムを扱っている例が中学校の技術家庭科にある[4][5]。情報の流れを自ら観察する活動や、システムを構築することで情報技術の原理の実感的理解に効果が見られている。また「日本学生科学賞」という自由研究のコンクールではICTを用いた身の回りの問題解決を図る作品を募集している[6]。このような取り組みは情報技術の活用が生活に役立つことを実感し、情報技術に対する肯定的な態度を育てることに役立つだろう。

本研究は、教科情報において模擬情報システム構築実習を通して情報技術の仕組みを学ばせることを提案する。その授業モデルの提案とその実現のために開発した実習環境について説明する。

Introduction of Imitative Information System
Constructions to Information Study and
Development of Its Environment

Hiroyuki Aoki^{1,2}, Setsuo Yokoyama¹, Youzo Miyadera²

¹ Tokyo Gakugei University

² Ohnuma Elementary School

2 目指す授業

2.1 身に付けさせたい力

現代の情報社会に対応できる生徒を育てるために、学習の4要素のうち知識理解・関心意欲態度・思考判断の3つにまつわる、以下の3つ学力を中心的なねらいとしたい。提案する授業はこれらの習得を目指す。

(1) 情報機器の原理の理解

身の回りの情報機器が行っていることをイメージできる力。それぞれの情報機器の能力を構成する要素に分解できる。

情報機器は持てる能力を組み合わせ人間の指示通りに処理を実行する。人間が間違えば間違う、人間が指示できないことは処理できないという限界について理解する。

(2) 情報技術への能動的態度

情報機器に「やらせたい」ことを考えることができる態度。情報技術は、人類が生活を便利にするために工夫してきた成果であること、利用者は指示さえできればコンピュータにやらせたいことを行わせられることを理解し、生活の改善のために情報技術を役立てたいと考えることができる。

(3) 教科情報ならではの問題解決能力

情報機器を通した問題解決とはどのようなものか、もしくは情報社会で利用されている問題解決の適用法を学ぶ。

人間同士では「常識」や「勘」で通じてしまうところを、コンピュータ相手の問題解決により問題をより明確化し、客観的に解決方法を構成することができる。考えた解決方法に基づいて実際にコンピュータを動作させ、その正しさを検証できる。

2.2 解決すべき課題

高校生に適した題材について考える上で、高等学校教科情報の学習内容に対応し、そして高校生

の能力を踏まえながら関心意欲に応えられる題材でなくてはならない。

現状の情報教育におけるプログラミングをみると数値の並べ替え等がその題材として頻繁に取り上げられているが、入門としての高校生には身近な問題としてとらえにくい。学習者自身の問題と捉えられる題材が有効であるが、そのような題材を教えられる設備、方法論が整っていない。

プログラミングが難しいという意識は多くの教科情報担当教員に持たれているイメージのようである。しかし、中学生がプログラムをしている事例において教育用言語を利用して成功している例が示すように[5]、環境を整えさえすればプログラミングは不可能ではない。

3 模擬情報システム構築実習の提案

3.1 構築するシステムの概要

提案する実習は身の回りの情報機器を模擬的に実現するシステム構築を行う。身の回りにある情報システムであれば、その機器がどのような動作をしているのか、生徒自身がアルゴリズムを考えることができる。

模擬情報システムの典型的例として炊飯器のシステムを挙げる。電子炊飯器は身近な情報システムの代表格とも言える教材である。

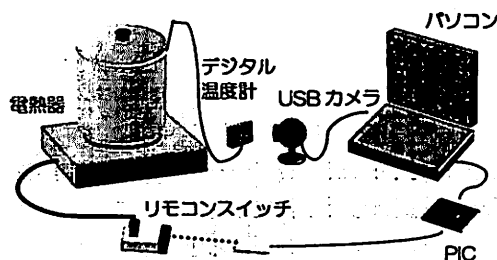


図1例：炊飯器システム

お米を炊く鍋に取り付けたデジタル温度計をUSBカメラによって撮影し、温度の値を得る。その値を基にパソコン上のプログラムで判断を行い、鍋を加熱する電熱器の電源をリモコンスイッチで制御する仕組みである。

このように今回想定している情報システムに

は以下のような特徴がある。

- ・ 映像情報を利用する。
- ・ 生活に役立つような成果の出力/制御を行う。
- ・ 生徒自身がシステム構築できるような実習用の言語を利用する。

その他の題材として、情報家電やインテリジェント自動車の分野、機械警備に用いられるような監視システム、テレビ番組でよく使われているバーチャルスタジオなどが考えられる。

3.2 実習の特徴

(1) 映像情報処理

目で見えない情報が多い中で、映像情報は目で見て実感できる情報であり、多様な処理が応用されている分野である。アナログ機器との比較の中で、デジタル化の利点を感じることができる。また、映像情報は大量の画素の情報を扱うが、この点からも大量のデータを一括処理できるコンピュータのメリットを感じさせることができる。

また、映像はテレビやコンピュータに欠かせない身近な題材であり、バーチャル映像があふれている現代にあって、その仕組みの理解はそのような作られた映像に対して冷静な対応のための有効な学習になりえる。また文字認識など社会を支える重要な技術の存在も知っていて欲しい。

(2) 計測制御

近年ではアシモや AIBO が注目され、いつの時代もロボットが人気を博すように、人間の意図通りに機械を動かすことは人類の夢である。実際に機械が自動的に動作する様子は面白く、物を作りだしたり移動させたり実際的な成果を得る働きをさせることができる。

近年、証券取引所の誤発注事件やエレベータの誤動作などコンピュータにまつわる事故が発生し、「間違わない」はずのコンピュータになぜこのようなことが起こるのか、不安感が高まっている。その点、計測制御は人間が誤った指示を出せばコンピュータはその通りに間違った動作をするということが実感を持って体験できる題材であり、これらの事件の本質を理解することにつな

がる。コンピュータ画面上の数字の操作だけでは学べない、物理情報のあいまい処理の難しさやその結果の重大さといった情報の奥深い側面を見せてくれる。

(3) プログラミング

プログラミングはその習得が目的ではなく、システムの挙動を記述する手段として捉えている。生徒自身が考えた機能を実現できるように利用できる機能をカプセル化して提供し、アルゴリズムをできるだけ素直に表現できるように、認知的負荷が少なくなることを目指した実習用言語を用いている。

3.3 教材の基本構成

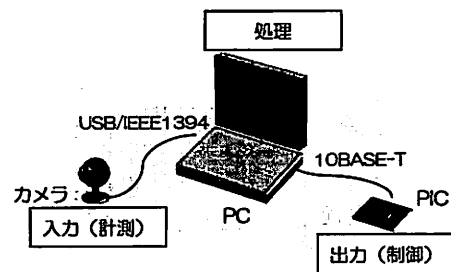


図2 基本構成

映像情報を入力し、機器を制御することができる要件を満たすため、ハードウェアは図2に示したような入力と処理、出力からなる基本構成を基とした。これに生徒それぞれが発想した測定対象と制御対象を組み合わせることで、生徒独自のシステムが実現される。

コンピュータの入力はUSBカメラなどの映像により行う。計測値を表示するセンサーを撮影することで値を取り込むことができ、その他の物理情報である長さや位置・色についても取得することができる。直接ADコンバータなどで値を取り込むのではなく映像として統一的に取り込むことで、様々なものが測定対象になりシステムの自由度が増すことと、情報の流れを意図的に明示することが期待される。

制御用の出力としてイーサネット接続のPIC（周辺機器インタフェースコントローラ）である

PICNIC[7]を用いている。これによりモータードライバを制御して物を動かしたり、家電製品の場合はリモコンスイッチを介すことでコントロールしたりできる。リモコンスイッチを介するのはコンピュータと対象を電氣的に分離して安全性と汎用性を確保するためである。

3.4 授業の流れ

情報技術の世界では技術的な内容を効率的に学ぶ方法として模倣による学習がよく用いられる。これを高等学校の学習でも応用する。図3が授業の流れを模式的に表したものである。

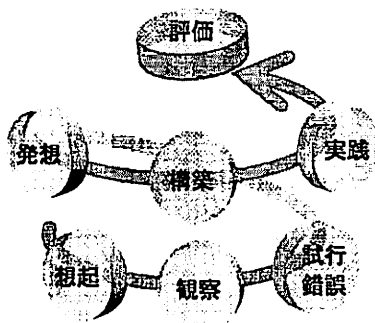


図3 授業の展開

まず、教師が例題となる情報システムについて生徒に想起させ、そのシステムのアルゴリズムについて考えさせる。その後、教師側がサンプルとして実現したシステムを提示し、そのプログラムと動作を観察して、自分たちのアルゴリズムと比較する。プログラムを試行錯誤しながらプログラムの仕組みについて学んでいく。

次の段階として生徒はグループとなり、これまでの活動を模倣しながら、自分達で考えたシステムを実現するアルゴリズムを考える。話し合いの中で生徒同士が伝えあうことから、次の段階のコンピュータに指示することにステップアップする。機器をそろえ、プログラミングをしてシステムを構築し、実際に動作させて期待通りの結果を得られるように調整する。

最後にお互いの作品を見せ合い自らの振り返

りと他者の考え・良さの理解を図る。

4 処理系

4.1 機能

プログラミング環境及び処理系が融合した統合環境を開発した。作成したプログラムがインタプリタにより解釈されることにより、構築されたシステムが動作する。

処理系は、オブジェクト指向の考え方によるカプセル化によってさまざまな機能を提供する。コンピュータに接続されたデバイスからの入力にはコンピュータの画面上に可視化された「オブジェクト」として表示され、備えている処理を容易に扱うことができるようになっている。

カメラからの入力である映像情報は「カメラオブジェクト」から3次元配列(2次元平面×色)として受け取ることができる。映像情報を利用者が簡単に扱うことができるために3次元配列に対する演算(切り出し、差分、上書き、平均値、合計値、重心等)を備える。また、カメラに写った値としての物理的情報を取り込むための数値映像を値に変換する、数字認識モジュールも提供する。

制御用出力はPICとLANを介してTCP/IPもしくはUDPにより通信を行うオブジェクトが用意されている。制御する対象により種類があり、入切の2値の制御に関しては「スイッチオブジェクト」、Fwd(正回転)とBack(逆回転)、Stop(開放)、Break(短絡)という4つの値による制御は「モーターオブジェクト」にメッセージを送ることにより行われる。

4.2 プログラム言語の仕様

情報システムの挙動をやさしく記述できるためのプログラム言語を開発した。すべての文法を記述すると煩雑になるため、特徴がわかる一部を拡張バックス記法で表したものをリスト1に示す。

系列内ではどのトークン間で改行を行ってもよいが、インデントレベルは各系列の階層構造を表現するのに利用されている。なお、同一行にあ

る系列は同じ階層である。

```
〈プログラム〉 ::= 〈宣言部〉〈ステージ〉
〈ステージ〉 ::= [①]〈ステージ名〉[改行]〈系列〉
〈系列〉 ::= [ | ]〈文〉 [ [ . ] 〈文〉 ]
〈文〉 ::= 〈オブジェクト文〉〈制御構文〉〈ステージ〉
オブジェクト文 ::=
  〈オブジェクト名〉〈メソッド名〉〈パラメータ〉
制御構文 ::= 〈For 節〉 | 〈If 節〉 | 〈Event 節〉 | 〈Loop 節〉
  |
  〈Redo 節〉 | 〈Exit 節〉
[... ]は終端文字を表す
```

リスト 1 言語の文法 (一部)

実際の例として炊飯器の動作を実現するプログラムをリスト 2 に示す。

```
Cam1 = videoinput "devicename"
SW1 = picnicswitch "URL"
温度 = Cam1 → Trim(Pos)52,36 (Size)120,40
      → デジタル数値変換

①炊く
  | SW1 On
  | For 温度 > 118 ; 水が無くなるまで
②熬らす
  | Event 温度 < 104 · SW1 On
  | Event 温度 > 106 · SW1 Off
  | For 10min
③保温
  | Event 温度 < 72 · SW1 On
  | Event 温度 > 74 · SW1 Off
  | For End
```

リスト 2 炊飯器のプログラム

初めの 4 行が宣言部であるが、このプログラム中でどのようなオブジェクトが利用できるかを表している。この部分は利用者が書く必要はなく、統合開発環境側で機能を選択すると自動的に生成されるものである。

予約語や処理系側で提供しているオブジェクトは基本的にアルファベットの単語を採用した。これは語尾活用の混乱が無いこと、間違いや手間を省くためにタイプ数を減らすためである。普段慣れ親しんで様々な連想をしてしまう日本語と違って一意に意味を定義しやすいことも利点である。BASIC が命令を覚えることが苦にならなかったのは簡潔な単語を用いていたからであり、本言語も省略形を用い、比較的日本人にも一般的

な単語を用いることで抵抗感を少なくしている。一方、自分でつける変数名等の名前については意味を理解しやすくするために日本語を推奨している。

4.3 プログラム言語の特徴

(1) 2 次元の分割

プログラムは分割して小さい単位で考えればやさしくなる。プログラムを適切に分割する手法としてプログラムの進行段階である「ステージ」と、構成要素別の記述や並行処理の単位である「系列」による 2 次元の分割を提供している。

「ステージ」は処理単位 (コードブロック) であり、プログラムの進行段階つまり時系列による処理の分割である。ステージは頭に「①」の付いたステージ名で始まり、インデントレベルにより範囲が決まる。ステージは入れ子にでき、並列処理や反復処理という構造の中ではステージは一つという制約があるため、複雑な処理はステージの入れ子により実現する。

変数やオブジェクト、イベントハンドラのスコープ (有効範囲・動作期間) はステージ毎である。このことでイベント駆動の記述ができながら、プログラムの進行によってイベントハンドラを切り替えることができる。

また、システムの進行状況を可視化するためにステージが役に立っている。ステージにはステージ名をつけるようになっているが、プログラム実行中にステージ名をステータスバーに表示してプログラムの進行をトレースすることができる。

「系列」はステージ内における処理の並列化であり、例えばシステムの構成要素についてそれぞれプログラムを記述するような用途で用いられる。ステージ中、「|」で始まる部分が一つの列として並列的に処理される。それに対して順次的な処理は「・」が先頭につけられている。

本実習で扱うシステムのような処理では並列に行うのか順次に行うのかは重要な違いになる場合がある。本処理系では異なる系列において並列して同じ機器にメッセージを送る場合、処理系側でメッセージをまとめ、動作の時間差をなくす

ことが可能である。

各並列処理が協調する仕組みとして脱出がある。脱出は処理が属しているステージの終了を意味し、合わせて並行している処理もそれぞれ終了して、次のステージに移行する。

(2) 直感的な制御構造

これまでの一般的なプログラム言語の制御構造は人間の素直な思考と相容れないところがあり、思考の翻訳が求められた。それがプログラム経験の間われるところで、初心者にとっては大きな障害となっている。

学習者の発想に合わせ、システムを記述するのに適した制御構造を以下の8つに整理した。

- ・ 並列: 処理の分岐や同時に行う処理 (系列)
- ・ 順次: 順番通りに処理する (文の連続)
- ・ 継続: 条件に達するまで現在の処理を続ける (For 節)
- ・ 条件通過: 条件によって処理を行う (If 節)
- ・ イベント: きっかけ (イベント) がおきる度に処理を行う (Event 節)
- ・ 反復: 一連の処理を繰り返す (Loop 節)
- ・ 再実行: 処理が失敗したとき等に一連の処理を再度行う (Redo 節)
- ・ 脱出: 一連の処理を終了させ、次のステージに進む (Exit 節)

(3) 安全機構

各オブジェクトは動作期間が終了したときに自動的に実行される「終了処理」のメソッドを安全機構として備えている。システムは例外の発生により進行不可能になったりユーザーにより停止させられたりすることがある。そのようなときプログラムが終了しているのにモーターが動き続けたり電熱器が加熱し続けたりしては危険である。終了処理に安全な停止方法を記述しておけばそのようなことを防ぐことができる。このことで安心してシステムを止めながら試行錯誤することができる。

並列処理では脱出によっても突如処理が終了

させられるということがある。該当ステージで使用が開始されたオブジェクトはステージ終了により動作期間が終了するため終了処理が呼ばれ問題が起こることは無い。

それでもカバーしきれない問題には、ステージもオブジェクトとして扱うことができることを利用し、ステージ自体に終了処理を設定して対応することができる。

5. 応用例

本実習の中心は生徒自身が模擬情報システムを考案し設計、構築することである。多様な種類のシステムが実現可能であることを示すために、この実習環境を応用したシステムをいくつか紹介する。

応用例(1) ロボティクス: ライントレーサ

ライントレーサはフィードバック制御とロボットの基本として有名な題材である。これまでのライントレーサは光センサーを必要な数だけ物理的に配置しなければならなかったが、カメラをセンサーとして利用することで、画面上の自由な位置に仮想的なセンサーを配置することができる。ただのライントレーサなら中学校の題材であるが、たとえば前の車と接近したら減速する、赤信号で止まる、2つ目の交差点で曲がるなど、自動車の自動運転の理解につながるような高度な処理に発展できるところが異なっている。

システムのハードウェア構成は図4の通りである。ロボットカーには PICNIC と DC モータードライバ、ネットワークカメラ、LAN のハブが搭載され、これが LAN ケーブルによってホス

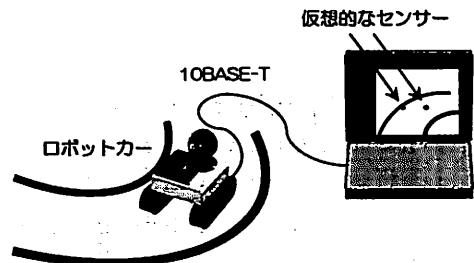


図4 ライントレーサ

トであるパソコンにつながっている。これまでのライトレーサはロボットカーに搭載した制御用ワンチップマイコンの性能上の限界があったが、本構成では処理はパソコン上で行うと割り切ることによって高度な処理を可能にしている。プログラム例をリスト3に示す。

```

Cam1 = videoinput "devicename"
Motor1 = picnicswitch "URL"
Motor2 = picnicswitch "URL"
左点 = Cam1→Trim (Pos)52,36 (Size)10,10
      → 明るさ → {平均<100:黒 平均>200:白}
右点 = Cam1→Trim (Pos)52,36 (Size)10,10
      → 明るさ → {平均<100:黒 平均>200:白}
@発進
| 右点監視 | 左点監視
@右点監視
| Event 左点 = 白 · Motor1 Stop
| Event 右点 = 黒 · Motor1 Fwd
| For End
@左点監視
| Event 左点 = 白 · Motor2 Stop
| Event 左点 = 黒 · Motor2 Fwd
| For End

```

リスト3 ライトレースプログラム

応用例(2) 映像情報処理：映像合成

最近のテレビでは合成映像を使った仮想的スタジオが良く見られる。その仕組みを理解することで、いかにテレビがCGにより作られた映像にあふれているかを客観視できる力を育てるとともに、面白い利用方法を発想させられる題材でもある。昔から有名なクロマキーから発展して、背景差分法のような映像情報処理が簡単に実現できる。

システムのハードウェア構成は図5の通りで

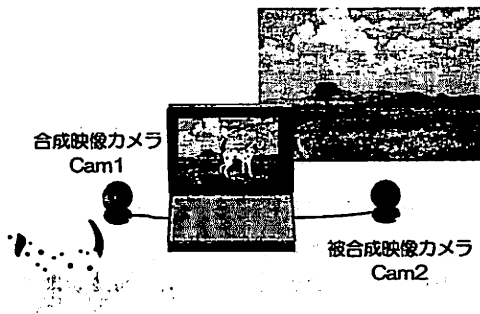


図5 映像合成

ある。USBカメラを複数用意することで合成する映像を撮るカメラ Cam1, 合成される映像を撮るカメラ Cam2の入力を取り扱うことができる。開始時に Cam1 の映像を背景として保存する。その後 Cam1 の前に合成対象が現れると保存した背景と新しい Cam1 の差分を求めることで合成映像が得られ, Cam2 の映像に上書きして描画することで2つの映像を合成することができる。結果は View というコンピュータ画面上の仮想デバイスによって行われる。

プログラム例をリスト4に示す。

```

Cam1 = videoinput "devicename"
Cam2 = videoinput "devicename"
View1 = videopreview (Size)640,480
@背景設定
· Cam1 → 背景
· @合成表示
| Cam2→ Draw
  (Pos) Cam1 Not Like 背景
  (Val) Cam1
  → View1
| For End

```

リスト4 映像合成プログラム

6. 考察と展望

本稿校了時には本実習環境の開発は完了しておらず、残念ながら実践を行うことができなかった。大学内でも試験を重ねるとともに、協力していただける学校を募り高校生の関心度合いと学習効果について検証していきたい。本研究は生徒自身が情報を活用して身の回りの問題を解決するユニークなシステムを考え実装してくれることを目指している。高校生にどこまでシステム開発が可能なのか関心のあるところである。

処理系の性能面では画像情報処理などはコンピュータが高性能になったとはいえ、本環境のようなハードウェアアクセラレータに頼らない処理は時間がかかり、断続的な動画になってしまう。しかし、教育用途という特性上実用的なスピードは必ずしも不可欠ではないし、逆に処理が遅いことから効率的な処理の工夫を考えるきっかけを与えることにもなると考えている。

情報の入力についてはUSBカメラのおかげで

簡単に取り扱えるようになったが、出力である制御機器については一般的製品が市販されておらず、授業者による手作りとなっている。本研究のような実習が認知されるようになって、キットメーカーにより既製品が販売されるようになるのを期待する。

また、この実習環境の機能を利用して物理運動の解析といった他の教科の学習や部活動でのスポーツ解析、アンケートの集計のツールとして利用することもできる。教科情報の授業の枠を超えた実用的な利用も期待している。

7. おわりに

情報機器の仕組みを理解するために教科情報においてどのような実習を行ったらよいか、その目的と生徒の関心を考慮して必要要件をまとめ、授業モデルを提案した。現在はその実習の試行段階である。

人間がコンピュータと正しく付き合うために必要なのは、人間とコンピュータの意志の疎通を図ることだといえる。そのためにはコンピュータの方も積極的に人間の意図をくみ取る仕組みを取り入れなければならないし、人間も積極的にコンピュータを問題解決の道具として活用していくべきだろう。そのようなとき、プログラミングの経験がきっと役に立つだろう。

学習者の成長段階に適した題材を与えることは大切なことであり、高校生には身の周りの問題の解決に取り組みさせることで「世の中を便利にしていくにはどうしたらいいか」という視点を持ってもらいたい。それが既に快適になりきってしまった現状に甘んじがちな今の時代に必要なことだと考える。

「プログラミングは難しい」と避けてばかりいでは本当に教えるべきことを教えることはできない。まだ教科情報は始まったばかりである。しっかり教科の目的は何なのかを見据えて、教材研究や環境の整備を行い、実績を積み重ねていきたい。

参考文献

- [1]中野由章,「近畿圏の高校における現状と課題」,情報処理学会情報処理教育委員会シンポジウム 高等学校「情報」の現状と将来, pp.19-25, 2005.
- [2]新藤茂,「学習者にとっての情報技術習得の構造と情報技術に対する不安との関係」,日本教育工学会論文誌, Vol.27 (Supple.), pp.173-176,2003.
- [3]天良和男,「教科『情報』における科学的理解のための教材の開発」,日本教育工学会研究報告集, No.5, pp.63-70, 2005.
- [4]田村俊之, 竹野英敏「中学校技術科『情報とコンピュータ』の導入段階に POS システムの仮想体験を取り入れた学習指導に関する研究」日本教育工学会論文誌 Vol.28, No.3, pp.227-236, 2004.
- [5]西ヶ谷浩史他,「IT クラフトマンシッププロジェクト～中学生によるネットワークプログラミング～」,情報処理学会研究報告 CE(83) pp.173-180, 2006.
- [6]全日本科学教育振興委員会, 読売新聞社, 独立行政法人科学技術振興機構,「日本学生科学賞」<http://www.jssa.com/>, 2006.
- [7]TriState 株式会社「PICNIC Version 2.0 製作マニュアル」<http://www.tristate.ne.jp/>, 2000.