

2B-8

EDSACを用いた計算機教育環境の設計と 移植性の高いシミュレータの実現

並木美太郎, 高橋延匡

(東京農工大学・工学部・数理情報工学科)

1. はじめに

当学科では、情報工学科の計算機初期教育として、世界最初のストアード型プログラミングの計算機EDSAC[1]を用い、過去10年にわたり成果をあげてきた[2,3,4]。その教育システムは、

- (1) 中型TSSシステム上に構築されたCAIシステムとソフトウェアシミュレータ[5]
- (2) 布線論理によるハードウェアシミュレータとマイクロプロセッサを用いたハードウェアシミュレータ[6]
- (3) パーソナルコンピュータPC-9801上に構築されたCAIシステムとソフトウェアシミュレータ[7]

が存在する。しかし、中型TSS上のシステムは8年、PC-9801上のものでさえ、5年が経過した。その間のハードウェアの進歩は目覚ましく、ビットマップウィンドウや日本語などユーザインターフェイスの質は格段に向上している。学生が個人の計算機を所有している例も少なくない。また、CAIの分野も人工知能の技法を取り入れた知的CAIが脚光をあびている。

本報告では、これら情報工学の進歩を考慮したEDSAC教育システムの設計とともに、種々のシステムに移植可能なEDSACソフトウェアシミュレータ核の実現について述べる。

2. 設計方針

(1) マンマシンインターフェイスの向上

使いにくいシステムは、学習者の学習意欲を低下させる。それはコースウェアなどの問題もさることながら、システムそのものも操作性も重要な要因である。次期システムでは、マルチウィンドウベース、日本語メッセージ、動画による例の提示、ヘルプメッセージを実現する。

(2) 知的CAIの実現

学習段階で、コースにそって実際に簡単なプログラムを作成させるのは、有効な教育法である。その段階で学習者の理解度を確認できる。しかし、プログラムのデバッグを支援し、「どこが間違っているか」を適格に指示できなくてはならない。そのためには、プログラムの不良箇所をシステムが認識できること、間違いの程度に応じた適切な指示を行う必要がある。知的CAIの技法が必要である。

(3) システムの移植性の向上

当学科で使用してきたシステムは、中型TSSとパソコンである。しかし、どちらのシステムもまったく別のプログラムであり、移植性がない。近年では、個人でパソコンを所有している学生が増加していること、教育環境の多様化を考慮すると、種々のマシンでの稼働が必要である。

また、システム自身の移植性もさることながら、操作性の統一性も重要である。異なるマシンで、異なる操作性を持たせたのでは、ユーザインターフェイス上問題がある。そこで、機種に依存しない統一した操作性を保証する。

3. システム構成

システム構成を図1に示す。

(1) 編集・入出力管理

本システムでは、マルチウィンドウをサポートするマルチウィンドウで、コースウェア、ヘルプ、シミュレータが同一画面に出力できれば便利であるのは言うまでもない。また、編集系の不統一を避けるため、ウィンドウシステムとエディタを統合したシステムとする。この部分には、現在我々が開発しているOS/omiconエディタを使用する[8]。この、エディタはCRTベースのシステムでマルチウィンドウをサポートしている。OS/omiconとMS-DOSで稼働しており、移植性は良好である。

(2) CAI管理

コースウェアの管理、学習の進行状況の判定、デバッグ支援などを行う。

(3) ヘルプ管理

CAI管理と関係する部分であるが、局面に応じたヘルプメッセージを管理する。なお、コースウェアやヘルプメッセージは、コースウェア・ヘルプメッセージ記述言語で作成される。テキストと同時に両面に描画する動画、各テキスト(コースやヘルプ)間の関係を内部に記述し、ユーザフレンドリーなコースウェアの作成が容易になることを目指す。

(4) EDSACシミュレータ

コースウェア中のEDSACプログラム、学習者のEDSACプログラムの実行を行う。

(5) 制御部

全体の制御を行う。

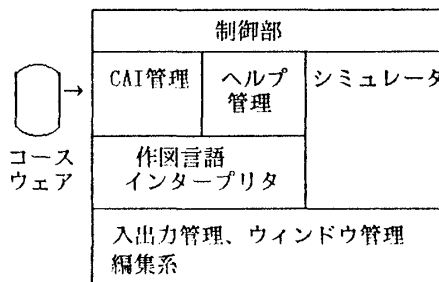


図1. システム構成

A design of EDSAC CAI and implementation of its simulator

Mitarou NAMIKI, Nobumasa TAKAHASHI

Tokyo Univ. of Agriculture and Technology, Dept. of Information Science

システム全体は、すべて言語Cで記述し高移植性を追求する。また、システム固有の機能は一切用いない。ただし、ハードウェアの機能は、カーソルアドレッシング可能なCRT端末（例えば、VT100相当の機能）、日本語の入出力が可能になることが必要である。上記の各部は、システムによっては静的リンクによって結合されるし、あるシステムではプロセス間通信で実現されるかもしれない。

各システムで保証される可搬性は次の通りである。

- (1) 入力・編集などの操作系に対する互換性
- (2) 作成されたコースウェアに対する移植性
- (3) システムのプログラムに対する、ほとんどすべてに対する移植性（システム依存ルーチンを除く）

なお、当研究室では、プログラミング言語に対するCAIの研究も行っているが、本システムはコースウェアの内容とCAI管理を差し替えれば、そちらにも使える設計を目指している。つまり、CAI研究のための、CAIシステム実現の“台”として設計を進めている。

4. EDSACシミュレータ核の実現について

以上がシステム全体の設計方針である。システム全体の實現に先立ち、最も利用頻度の高いEDSACシミュレータを實現した。EDSACは図2に示すようなアーキテクチャとなっている。シミュレータのコーディングで注意した点は次の通りである。

- (1) 8ビットや16ビット長のデータの宣言には、言語Cの基本型を直接書かず、マクロ定義とtypedefを用いた

例えば、EDSACのメモリ(17ビット1024語)の定義は次のように記述した。

```
/* OS/omicron言語Cコンパイラの場合 */
#define WORD int
#define BYTE short
WORD edsc_mem[1024]; /* 下位16ビット */
BYTE edsc_lbit[1024/sizeof(BYTE)]; /* 最上位1ビット*/
BYTE edsc_swd[512/sizeof(BYTE)]; /* サンドイッチ*/
/* デジット*/
```

これらの配列に対するアクセスもマクロ定義とし、システム移植時に変更しやすいものとした。

- (2) 割り付けに依存しないコーディングとする

例えば、バイトスワップを起こすシステムとそうでないものがある。どちらに対しても無関係に作動するコーディングとした。

- (3) 文字コードに依存しない

一般に知られている英数字のコードには、ASCII、EBCDICがある。日本語の場合はシフトJIS、EUCなど様々である我々の實現したOS/omicronでは、すべての文字コードをフル2バイトコードとして扱っている。本シミュレータは、内部コードと外部コードを分離し、内部コードをフル2バイトコードとしている。

- (4) 入出力などのライブラリを直接記述しない

ANSIの規格では、ライブラリなどの仕様も規定されているが、日本語を扱う処理は処理系定義が多い。そこで、ライブラリ関数の使用はすべて一か所に集中させた。

以上の点を考慮して、EDSACシミュレータを實現した。言語Cで約1500行程度である。現在、OS/omicronとMS-DOS上で稼働している。

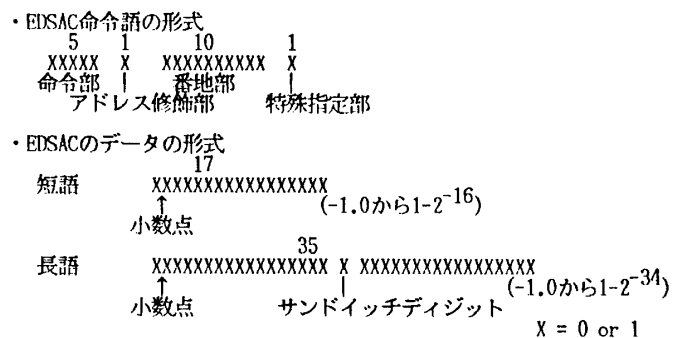


図2. EDSACの命令語とデータ形式

5. おわりに

EDSAC CAIシステムの設計と、移植性の高いEDSACシミュレータの實現について述べた。今後は、

- (1) 編集・入出力部の改造
 - (2) コースウェア・ヘルプメッセージ作成言語の設計と實現
 - (3) CAI部の設計と實現
- を行う予定である。

参考文献

- [1] M.V. Wilkes et al., The Preparation of programs for an electronic digital computer, Addison-Wesley company, INC.
- [2] 高橋延匡, EDSACによるプログラミング教育の方針, 情処学第21回全大5L-1, pp.1221-1222, 1980.
- [3] 清水他, EDSACによるプログラミング教育の実際, 情処学第21回全大5L-2, pp.1222-1223, 1980.
- [4] 清水他, 情報工学系学科の計算機初期教育におけるEDSAC活用の試みと効果, 情処学論Vol.24, No.3, pp.281-292, 1983.
- [5] 岩沢他, EDSACの演習補助用システムの基本設計, 情処学第22回全大5G-4, pp.1075-1076, 1981.
- [6] 井上他, 教育用EDSACハードウェアシミュレータ, 情処学第21回全大5L-3, pp.1225-1226, 1980.
- [7] 清水他, 情報工学科の計算機初期教育のための環境について, 信学論Vol.J71-A, No.9, pp.1734-1741, 1988.
- [8] 高橋延匡, 研究プロジェクト総説: OS/omicronの開発, 情処学OS研資39-5, 1988.