

## 第四世代言語によるシステム設計教育

5X-7

新田雅道

日本電子専門学校

## 1. はじめに

専門学校教育の主たる目的は、即戦力となる人材を育成することであろう。このことからいえば、プログラマ教育に重点を置けばよいことになるが、企業が本当に求めているのはSEであり、学生もそれを目指している。したがって、学生時代にSEの仕事の一部を経験させることは、自分の将来を多少知ることができ、社会に出る上での心構えもできるので有意義なことであろう。

さて、SEの基本的な仕事はシステム設計である。よって、SEを育てるにはシステム設計教育を中心としたカリキュラムを組めばよいことになるが、システム設計の教育は、次項であげるような問題点があり、これらを解決するには新しい教育環境が必要になってくる。最近、CASE (Computer Added Software Engineering) や第四世代言語 (4 Generation Language: 4GL) といった、新しい開発環境が注目を集めている。本稿では、このうち第四世代言語を取り上げ、SE教育に活用する方法を述べてみる。

## 2. システム設計教育の現状

学内におけるシステム設計教育は、つぎの点において難しい。

## ①. システム設計の教育は抽象的になりがちである。

システム設計のさまざまな技法を理解させるために具体的な例題を用いるが、その例題は千差万別の実際の適応業務の事象を表わしているものではない。したがって、技法を実務でいかに生かすかまでは教えきれない。

## ②. 学生が業務の仕組みや社会の構造を知らないので、いきなりシステム設計の教育に入ることはできない。

授業用の例題としてシンプルなものを用いているが、業務の仕組みを知らないため、業務内容の説明から始めなければならない。このためシステム設計の本題に入る前の準備に時間がかかる。

## ③. 学生が演習として行うシステム設計の成否は、運用まで至らないと確認できない。

システム設計は机上設計が中心なので、おおまかな論理の正しさは検証できるが、詳細は実際にシステムを動かしてみないと検証するのは難しい。したがって、

開発、運用まで至らないと、自分で作ったシステムの評価ができない。

## ④. 技量の向上はフィードバックによって大きく前進すると考えるが、そこまで十分な時間をさくことはカリキュラムの都合上難しい。

これらのことを考慮した上で、システム設計教育およびシステムの構築実習を行わせるが、図1に示す通り、期間内に全作業を終了するのは困難である。

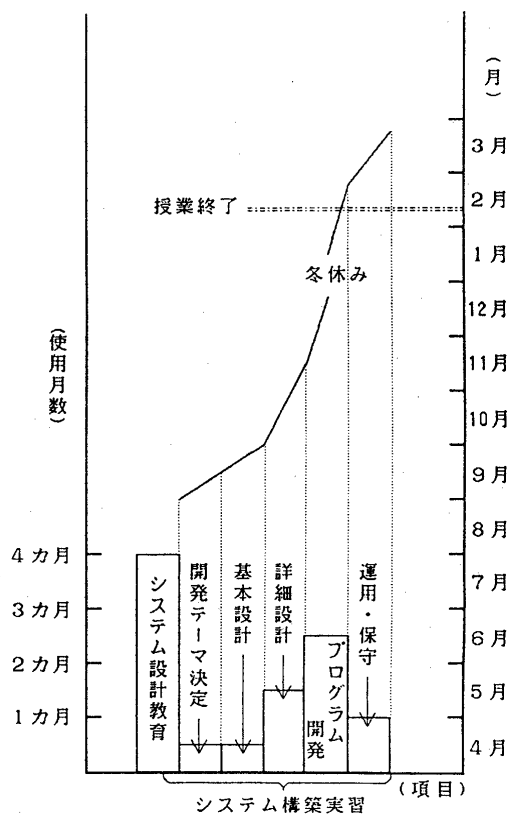


図1 システム設計教育の消費時間と累積月数

System design education by 4GL

Masamichi NITTA

Nippon Electronics College

図中のシステム構築実習は、本校の平均的な学生の作業時間である。

この図では、運用・保守も含めて1カ月強ほど時間が足りない。そこで、期限内にシステムを完成させるには、

- ①. 授業時間を増やす。
- ②. もっと優しいテーマを与える。
- ③. 開発効率を上げるシカケを考える。

などの工夫が必要である。しかし、授業時間を増やすのは他科目とのバランスがあり難しく、②も実際にやってみないとテーマの難易度がわからない。よってここでは、③の開発効率を上げる方法を検討してみる。

### 3. 第四世代言語の活用

第四世代言語は、システム開発の生産性、信頼性、保守性などの向上を目的とした言語で、現在の情報処理ニーズにフィットした機能を持っているのが特徴である。開発効率がよい反面、言語体系はメーカに依存する部分が多いので、学校内でのメインとなる言語にはなりにくい。これらのことを踏まえた上で活用方法を検討してみた。

本校のメインフレームはIBMのES/9221なので、第四世代言語は同社が提供するCSP(Cross System Product)を用いる。CSPは、画面制御、データベース操作などが手軽にできる言語で、その言語体系はかなりCOBOLに近い。データベース操作言語はSQL(Structured Query Language)である。

CSPには専用のエディタが用意されており、画面設計やプログラミングはこのエディタを使って行う。また、機械語への翻訳にはインタープリタとコンパイラが用意されているので、プログラムのテスト時は、インタープリタを使ってステップ実行を行い、完成したらコンパイラによってロードモジュールを生成するということもできる。データベースとのリンクについても、データ定義のモジュールが用意されている。このように統合環境になっているので、習熟すれば確かに開発効率は上がるだろう。

メーカの資料によると、CSPを使って開発すれば内容にもよるが、従来 of 言語を使うよりも30%の効率アップが期待できるそうである。このことを、図1の『プログラム設計』の作業時間で計算すると、今までプログラム開発に2.5カ月かかっていたのが、1.75カ月(約2カ月)で済むことになる。そうなれば、プログラム開発は期間内に終われるようになる。あとは『運用・保守』をどうするのだが、これについては次項でとりあげる。

とこのように、計算上はうまくいくが、いくらCSPがCOBOLライクな言語とはいえ、新しい言語には違いないので、言語教育にもある程度の時間をさかなければならないし、構造化設計技法やファイル操作法といった基本的なことも理解していなければならない。なぜなら、CSPの言語体系は完全な構造化プログラミングしか許していないので、徹底した構造化設計の知識が必要になるし、データベースを使うなら、SQLの構文も知っていなければならないからだ。

### 4. CSPの習熟と『運用・保守』について

CSPの言語教育は、短期間で終わるためにサンプルプログラムを中心に行う。このサンプルプログラムはリストを机上で説明するだけでなく、実際にオペレーションを行えるようにしておく。しかも、サンプルプログラムは単体ではなく、テーマをもったサンプルシステムとして学生に提供する。これは、次のようないくつかの効用がある。

- ①. まねることから入った方が、理解が速い。
- ②. 実際に運用することにより、人、物、金の動きをシミュレーションすることができる。
- ③. 他人のシステムの評価ができるようになる。

また、『運用・保守』についてはぜひやっておきたい。なぜなら、運用することにより評価を出すことができ、フィードバックがかけられ、よりよいシステムにすることができるからである。しかし、図1のままで時間切れでできない。そこで、『運用・保守』は、プログラム開発が始まる前にサンプルシステムを使ってやってしまう。もちろん自分が作ったシステムではないので、自分のシステムに対する評価はできないが、システムを評価し、フィードバックをかけ、保守を行うという一連の作業はできる。既存システムの保守は、新入社員が最初に体験する作業でもあるので、ある意味では実務的である。サンプルシステムは、より多くの評価を得るために、わざと操作性の悪いシステムにしておくのがよいだろう。

### 5. おわりに

以上、システム設計の教育に第四世代言語を取り入れたときの効果を中心に、簡単にカリキュラム運用について触れてみた。

学生にシステムを一から構築させると、遅々として進まず、中途半端な状態で終わってしまっていることに、担当教員はいつもジレンマに陥っている。やはり、どんなに簡単なシステムでもいいから完成させ、達成感を味わわせてやりたい。そんな気持ちから、第四世代言語を取り入れてみたが、その運用については別の形で現在進行中なので、今回は報告はできない。

第四世代言語は最近あまり脚光を浴びていないので、強気に押し進めることはできないが、今後の動向を見ながらさらにシステム設計教育について検討していきたい。