

ソフトウェア設計の基本(その2)

4 J-1

—現場担当者主導のシステム開発—

今井 恒雄、荻田 直史、増田 秋俊(富士通) 平木 しげ子(東京家政学院筑波短大)
荒木 雄豪(京産大) 古谷 千恵(東大生研)

1. はじめに

オープン化、ダウンサイジングなどの言葉で代表される最近の動向の中で、ハードウェアの性能が向上し、価格が安くなるとシステムの開発もより現場主導型になり、ソフトウェアの開発の仕組みや生産性がこれまで以上に大きな課題になることは明らかであろう。

我々はこれまで、システム開発のあり方とその教育方法について、大学や高校における事務処理システム(入試、学籍、成績、図書など)の開発や、言語教育を通して、研究と実践を進めてきているが、今回は現場主導でシステム開発を進めるためにはソフトウェア設計はいかにあるべきかについて述べる。

2. システム開発の考え方

いまの一般的な考え方では、現場(エンドユーザ)は現状の調査分析と画面入出力・帳票出力の設計に参加したあとは、テストの段階までは何もしない。つまり、アプリケーションプログラムを実際に作成することはない。

我々は、現場の人達がシステム開発の最初から最後まで主導的な役割を持つことを目指している。

このため、我々のアプローチは『大部分のプログラムがやさしい処理になるような設計にする』ことにある。

『オンラインは難しい。バッチ帳票出力はやさしい。』という感覚ではなく、例えば『帳票出力を本当の意味で簡単にする』に設計のポイントをおいている。

現場担当者は業務についてはよく分かっており、そのシステム化も自分達で行うのであるから、一般的なシステム開発方法と同じ方法をとる必要はない。つまり、『開発の各フェーズを順番に行うのではなく、できるところから並行して行い、何度も初めにもどって繰り返すこと』にある。

次の発表(学生による図書館業務のOA化)はこの方法で実現したものである。

3. 90%のプログラムをやさしくするための

ソフトウェア設計のポイント

システムを、やさしい部分(90%)と難しい部分(10%)になるように設計する。

(1)やさしい部分

- ・帳票出力
- ・データエントリ
- ・非定型業務

(2)難しい部分

- ・入力データのチェック
- ・ファイルのマッチング処理
- ・ロジックで難しい処理：例えば、図書目録の複数行にわたる場合の単語のつなぎの処理

(1)やさしい部分に対して

- ・データエントリプログラムの自動作成*1

画面入出力プログラム言い換えればデータエントリプログラムを、ファイルの各入力項目に関する情報などを与えてデータエントリのプログラムを自動的に作り出す。

- ・エディタの利用

フルスクリーン・エディタをデータエントリにも使う。

- ・主ファイル

システムの大部分をやさしい部分にするファイル設計技術であり、業務のあるまとまりに対して、処理(特に各種帳票出力)に必要なデータを一つの順編成ファイルにまとめたもの。

これによりプログラムは、主ファイルを入力してソートを行えば、簡単に必要な情報をレコード単位に順次取り出し、目的の帳票を作成することができる。

同じ帳票を出力するにも、この主ファイルを使った処理は複数ファイルをマッチングして行う処理に対して格段に簡単になる。

この主ファイルの設計がプログラムの組みやすさに対してとりわけ重要である。

- ・雛形プログラムの準備

ファイル処理プログラムの典型的なパターン約10種類を各ファイルごとに用意しておき、現場担当者はこの雛形のどれかに処理の部分を追加して帳票出力プログラム等を完成させる。

例えば、非定型業務の場合もこれを応用すれば簡単に対処できる。

A Basis of Software Design (No.2)

Tuneo IMAI¹, Naofumi OGITA¹, Akitosi MASUDA¹, Sigeko HIRAKI², Hugo ARAKI³, Tie HURUTANI⁴

1 Fujitsu Ltd. 2 Tokyo Kasei Gakuin Tukuba Junior College 3 Kyoto Sangyo Univ.

4 Institute of Industrial Science, University of Tokyo.

(2)難しい部分に対して

- a. 本質的に難しくないものを、気がつかずに難しくしている場合
 - ・難しいコーディング (OCCURSを使用している) の場合は、やさしいコーディング (OCCURSを使用しない) に変える。
 - ・帳票の形式が原因で処理が難しくなっている場合は、話し合いで形式を変えれば、やさしい処理に変えられることも多い。
- b. 本来の難しい処理に対する対処
 - 入力されたデータに対する様々なチェック、マスタファイルの更新プログラムやロジックの難しい部分のサブルーチン化は、リーダーが担当する。既存のサブルーチンライブラリがあれば、それを利用する。

4. 開発の進め方のポイント

- ・現在使っている帳票をもとに業務の調査・分析をし、それと並行してファイル設計を進める。
- ・リーダーは現場担当者が作業できる環境をまず整える。
 - 主ファイルについては設計と同時に雛形プログラムとテストデータを作成する。
 - 難しい部分を出来るだけ早い時期に作り終えておく。
- ・帳票出力プログラムは、自分達が納得するまで何度も繰り返して作成する。
- ・ドキュメントはできるだけ作らない。
 - ファイル設計書、業務フローチャート、難しい部分の説明書は作成するが、「90%のやさしい個々のプログラム」に対しては、操作法と注意事項を載せた一枚程度の簡単な説明書を作る。

5. 現場担当者主導のための訓練

担当者がコンピュータとはどういうもので、どういうことができ、どういうことは難しいかということを、実際に体験しながら、自分達のシステムを自分達で、自分達の納得のいくものにする。

我々は、これを実現できるような訓練を目的に以下の5冊の訓練書を出版している。

- ①「COBOLによる業務プログラムの作成 (I)」*2
- ②「COBOLによる業務プログラムの作成 (II)」*3
- ③「ファイル処理入門」*4
- ④「PFD入門」*5
- ⑤「システム設計 (誰にでもできるOA化)」(未刊)

この訓練書は、トレーニングの途中で実際の業務プログラムの作成を始めるように書いてあるので、それをよく意識してスケジュールを組むと効果が極めて大きい。

6. リーダーの育成

我々は長年の経験から、ソフトウェア設計をどうすべきかについて一つの考え方、方法論を作り上げたが、リーダーがキーを握っていることは明らかである。リーダーはやはり実際のシステム開発の中で育成するしかないが、前述の「システム設計 (誰にでもできるOA化)」はリーダー育成に大いに役立つと考えている。

7. おわりに

我々の考え方は、例えば中小企業の現場担当者が自分達で開発するケースなどに最適であろう。しかし、現在世間一般での方法とは相入れないところが多いと考えている。それは、仕事の進め方が根本的に違っているからであろう。

- ・システムを納得のいくまで何度でも作り直す。
- ・やさしい部分の割合をできるだけ大きくする。
- ・コンピュータの経験のない人の方が、システム化の担当者としては適任である。

このような仕事の進め方は先程述べたようにシステム開発を外部に委託する場合には馴染まないことは明らかである。

しかし、我々の考え方の中で、例えば「やさしい部分の割合を大きくする」という考え方は現状のシステム開発のやり方でも充分取り入れることができるものであり、大きな効果 (生産性向上) が期待できると考えている。

参考文献

- *1 今井他「データエントリプログラムの自動作成について」情報処理学会 第34回全国大会 2T-6 (1987)
- *2 出版社:恒星社厚生閣 著者:平木茂子
- *3 出版社:恒星社厚生閣 著者:平木茂子
- *4 出版社:恒星社厚生閣 著者:平木茂子, 荒木雄豪, 今井恒雄
- *5 出版社:恒星社厚生閣 著者:今井恒雄, 荒木雄豪, 平木茂子