

# ZIPC によるシステム開発について

十日市 勉

## ■ 導入の経緯

**弊**社は、主に通信系のソフト開発を行っています。通信を応用した PC のデータベースアプリケーション開発も行っていますが、約半分はファームウェアなどの組込ソフトウェアの設計・製造です。

一昨年より、「ソフトウェア開発環境を改善しよう」という動きがあり、私はその推進を担当しています。社内で設計業務の改善を進めるための活動を行っているのですが、会社全体が忙しく、開発環境の話題を出してもほとんど掛け合ってもらえない中、あるプロジェクトのリーダーが「新しいプロジェクトをやるので何かいいツールがあればやってみてもいいですよ」と声を掛けてくれました。

そのプロジェクトは新しい電話機のファームウェアを開発するチームでした。ファームウェアであるため私の持っている開発ツールの資料の中から、組み込みソフトウェア用のものを渡し、選んでもらうことにしました。確か10種類程度はあったと思います。

次の日に「これがいいと思います」と選ばれたのが ZIPC でした。しかし、ツールは実際に使った感じがわからないと、導入するのが非常に不安なものです。でも、私は「これは、いいと思います」と即答しました。即答できた理由ですが、その前年の「ソフトウェア開発環境展」を見に行ったときに、私が一番興味を引かれたのがこのツールだった事が1つです。あの時は PC の画面上で電話機が動いており、更にその動きに合わせて状態遷移表がトレースされているデモンスト

レーションを見ました。私も組み込み系の開発担当のため、状態遷移表にはよくお世話になっています。それがシミュレーションと同期してトレースされているのを見た時には、感動以外のなものでもなかったのです。直感的に「これは使える！」と思っていたのです。その後、Interface 誌にも紹介記事が載り、高い評価を得ているも見ました。

私が一番良いと思い、Interface 誌でも高い評価を得ており、その上それを知らない担当者が、数あるツールの中から ZIPC を選んだのです。これは悪いはずがありません。私が即決できた理由はこれらの要因が重なったことでした。

購入に際して、キャッツ(株)に申し込みを行ったのですが、その時に「是非、社員の方々にツールの紹介をしたいのでお伺いしたいのですが?」と言われまして、わざわざ会社に来ていただき ZIPC の紹介セミナーをやっていただきました。首都圏の会社なら当たり前に行われている事かも知れませんが、私の会社は「青森県」にあるのです。「えっ、ここまで来ていただけるのですか?」、「はい伺わせていただきます!」これにはビックリしました(キャッツ(株)の船山さんありがとうございました)。地方の会社ではなかなかこのようなサービスを受けることは出来ないのですが、とても親切な会社である事を感じました。この原稿をキャッツさんに頼まれましたが、これだけ親切にされると断るわけにも行きません。

さて、実際の導入ですが「あまり高望みはしないで行こう!」と決めました。ZIPC はフル装備すれ

ば状態遷移表だけでなく、コード生成、シミュレーション、リバースエンジニアリングまで可能です。しかし、いきなり全部に取り組んで消化しきれないのも怖いと思い、「とりあえず状態遷移表だけでも効率よく作ればよし」として、最小限の機能である「ZIPC デザイン」を購入しました。ただし、買うのは簡単なのですが実際に使うとなると説明書だけでは不安なので、開発チームのメンバーから1人を選び、ZIPC のセミナーに参加してもらいました。セミナーは数種類あったのですが、ZIPC デザインだけではなくその先の展開も考慮しておきたいので、全ツールの内容が分かる「オペレーションセミナー」を選びました。この時にセミナーに参加した担当者は、「だいたい分かりましたよ。どうせ使うなら本気でやりましょう！」と言って協力的に動いてくれるようになりました。

## ■ 導入の状況

**Z**IPC はモデルプロジェクトで実施した訳ですが、特にそのプロジェクトだけではなく何人かに使ってみてもらいました。ソフトを専門に開発する会社とはいえ、「ちょっと使ってみてよ」と言っても新しいツールはなかなか受け入れられない事が多いのです。しかし、ZIPC は意外とすんなり受け入れられました。ほとんどの社員は状態遷移表を作成した経験があります。ZIPC は状態遷移表そのものなので、今まで設計を行った経験者には直感的に使えてしまうようです。新しいツールが受け入れられない原因に「修得する時間が取れない」と言うことがありますが、これをほとんど必要としないため受け入れられたのだと思います。更に、修得に時間がかからない上に修正作業がとても楽になります。ZIPC では状態、イベント、状態番号が全てリンクしていますから、状態やイベントの増減といったよくあ

る作業を一気に終わらせることができます。これだけのメリットがあるのですから「一度使った人間は手放さない」そんな傾向が ZIPC にはありました。

モデルプロジェクトの方ですが、こちらでも成果が上がりました。

開発の担当者の話では、今回は「ZIPC デザイン」の範囲内でしか使っていないが、4人月の機能設計期間で 100 時間ほどの節約が出来たと言うことでした。これは約 15%の節約になります。コーディングやデバックを含めた中で考えると効果はもっと少なくなります。状態遷移表をツール化しただけでこれだけの成果があるのは画期的であると思えます。

時間的にはこの程度の効果でしたが、他に得られたものが多くありました。大きな部分は、状態遷移表を主体とする事で設計方法が統一されま

- す。これにより、
- ① 他人の設計も理解しやすい
  - ② マトリクスのため設計漏れが少ない
  - ③ 理解しやすいためメンテナンスも楽である
  - ④ 他のプロジェクトで流用する場合、柔軟に対応できる(部分的、局所的な流用であっても)
- …といったメリットがありました。これらは、生産性に対して即時に効果が出るものではありませんが、将来的に有利な部分が多くなることを示しています。

設計行程後の状況ですが、コーディングの少し前にちょっと欲が出ました。これだけ徹底して使っているのであれば、自動コーディングも出来るのではないかと思ったのです。セミナーを受講したときの感じでも、それほど難しい事ではないようだったので、思い切って次の段階の「ジェネレータ」も購入しました。

早速、状態遷移表をジェネレートしてみました。

この時は試しにやってみただけなので、雛形程度のものでしたが、確かに状態遷移表がコードとなって枠組みが出来ています。あとは、中身を必要なコードで埋めて行くだけで済みます。これは、Visual Basicなどで画面を作って出来上がったオブジェクトに必要なコードだけ作り込むのに似ています。大まかな構成をプログラミングすることは、設計さえしっかりしていればそれほど難しい作業ではないとしても、手でコーディングすればそれなりに時間がかかります。それを一瞬で作ってしまうのですから、これは生産性に大きく貢献できると思えました。

その後、いくつかの表をジェネレートしているうちにまずいところが見つかりました。ある条件だけを満たすことが出来なかったのです。それは、ZIPCの癖のある部分だったのです。最初からそれを分かって状態遷移表を作っていれば何の問題もなく進めたのですが、その時点で全部を修正するのは工数に見合わず結局ジェネレートを使うのは断念してしまいました。この点について、その担当者は開発の最後まで悔やんでいましたが「もう完全に分かった。次は絶対うまくできるぞ！」と言っており、さらなる野望を燃やしているようです。

## ■ サポートについて

**わ**かりやすい ZIPC ですが、使い始めると相談したいところはあるものです。これらについては、ZIPCのサポートはとても親切に教えてくれます。

一度、状態遷移表の作成で「あれ？」と思わせる事態が発生しました。これははっきり言わせていただくと「バグ」だったようですが、それを避けるためにはほとんどの表を修正しなければならぬ状況でした。サポートに問い合わせたところ、素直に「それはバグです」として修正モジュール

を送ってきてくれました。

これには驚きました。色々なメーカーのサポートを利用しますが、バグの回避方法を教えてくれるサポートはあっても、個別のユーザが困っている部分のために修正版を送付してもらえるなんて滅多にありません。なんと言っても「純国産ツール」ですから、どこかのサポートのように「それは英語版からの仕様でして…」と言うのはありません。

サポートは電話を受ければよいというものではありません。大手メーカーにも見習ってほしいものです。

## ■ 今後の ZIPC の扱いについて

**モ**デルプロジェクトを通して、ZIPCが組み込み系開発に最適なツールである事がわかりました。画期的な使い方をするには慣れが必要としても、修得に時間がかからないのは大きな魅力です。

世の中にはソフトウェア開発ツールとして、オブジェクト指向を土台とした CASE ツールが多く出回っています。しかし、これらを修得するのは並大抵のことではありません。ZIPCはそれらの開発ツールの中で「組み込み系開発のかゆいところに手が届くツール」と思えます。

今後は、このモデルプロジェクトで得られたノウハウを整理し、社内の標準ツールとしての展開を行っていきたいと思います。

## ■ 要望事項

**欲**しいところをうまく構成したツールですが、マンマシンがそろそろ古くなってきている感じがします。贅沢だと言われるかも知れませんが、MS-Word、MS-Excelの操作性が取り込めれば良いと思います。また、状態遷移表とメッセージシーケンスチャートがワープロの文書に

気軽に張り付けられるようになれば、もっと利用範囲が広がります。その他、Windows NT で動けば文句なしです。

ZIPC Ver5.0 ではかなりのマンマシンの改善を計画しているようなので、それに期待したいと思います。

ZIPC 開発、販売、サポートに携わっている皆さん、これからも良いツール開発、良いメーカーであることを続けて行って下さい。ありがとうございました。

[ とおかいち つとむ ]

## ZIPC 関連イベントのお知らせ (2)

### ITRON オープンセミナー ~ 拡大する標準化とその成果 ~ について

- 日時 : 1998 年 7 月 15 日 10:00 ~ 18:30
- 場所 : アルカディア市ヶ谷(私学会館)・・・JR 市ヶ谷駅より徒歩 2 分  
東京都千代田区九段北 4-2-25
- 内容 : ITRON プロジェクトの最新動向  
田丸喜一郎 (ITRON 専門委員会 / (株) 東芝)  
μ ITRON4.0 仕様の概要  
高田広章 (ITRON 専門委員会 / 豊橋技術大学)  
パネルセッション「ITRON のデバッグインターフェース標準化に向けて」  
(パネラー未定)
- ITRON プロジェクトの最新の成果
- (1)「ITRON TCP/IP API 仕様の概要」  
児玉剛 (Embedded TCP/IP 技術委員会 / アルパイン情報システム(株))
  - (2)「ITRON (Ver.2) 仕様の概要」  
八谷祥一 (Java Technology on ITRON-specification OS 技術委員会 / (株) アプリックス)
- ITRON 関連製品・応用事例の紹介
- (1)「JTRON 仕様 OS Jblend のご紹介」  
矢内健治 ((株) アプリックス)
  - (2)「RTOS Resource-Aware Cygnus GDB Debugger for μ ITRON」  
David Henkel-Wallace (Cygnus Solutions)
  - (3)「ITRON 仕様 OS に対応した ZIPC」  
渡辺政彦 (キヤッツ(株))
  - (4)「産業用ネットワークコンピュータ NetJacs」  
中本幸一 (日本電気(株))
- 特別講演「HFDS の実現に向けて Part2」  
坂村健 (東京大学)

### ESC 1998 について

- 展示会名 : Embedded Systems Conference 1998
- 開催期間 : 11 月 3 日 (火) ~ 11 月 5 日 (木)
- 場所 : 米国 SanJose Convention Center  
(サンノゼコンベンションセンター)
- 内容 : ZIPC Ver.5.0 English Version 発表 !