

ZIPC を使用した開発

松田光浩

■ ZIPCとの出会い

部としてあるいは会社として生産性の向上・品質の向上と叫ばれるのは、いつの時代でも変わらないでしょう。当然のごとく当部でも何らかの措置をとるべき必要がありました。今の流行としては、CASEツールが特に注目されていますが、世の中には数多くのCASEツールが存在します。しかしどれも現状の開発スタイルとあまりにもかけ離れているので、いまいち踏み切れない所がありました。そんな中、新しい手法が現状の開発スタイルを壊すことなく使えるということから、ZIPCとのつきあいが始まりました。

これからの解説の中に実際のシステム名称やロジックを紹介できないのが残念ですが、ZIPCをどの工程で使用し、どのようなサイクルを経て1つのシステムが完成したかを紹介します。

■ なぜZIPCなのか

ZIPCの謳い文句である状態遷移表の考え方は、当部門では古くからありました。「全ての状態で全てのイベントを考えなければいけない」というスタンスがあったためです。

今までは、状態遷移表を書くツールなどは無く、ワープロで作成するのが現状でした。普通はこれで問題ないのですが、開発するソフトウェアにはバグが付き物です。これらの修正を重ねていくうちに、元の状態遷移表とはかけ離れたものが最終的に出来上がってしまうということは良くある事です。これを補正するのは、開発スケジュールの厳しさからどうしても開発終了後に行っていました。

プログラムソースは「なまもの」です。バグなどにより修正が次々と入ると、もうそのドキュメントは過去のものであり参考程度にしかありません。この状況は悲しいことになかなか防げないのです。そのドキュメントに信頼性があれば、どんなに見やすい事でしょう。

しかし、ZIPCはそれを解決してくれました。ZIPCをどの様に使用するかはユーザー側の考え方で変わってきますが、我々はZIPCでの生産物を100%リンクさせて、修正などはソースには一切行わずに、必ずZIPCのドキュメント上で行う事にしました。

■ 生産性と品質

私自身の考える生産性と品質の向上とは、「どれだけ短い期間にバグの無いプログラムを作成し、システム試験あるいは運用試験をスムーズに完了させられるか」という事だと考えます。

話は少しそれますが、私自身の性格はかなりの面倒くさがり屋で、手を抜ける所は手を抜きたい主義です。開発過程にて数多くのドキュメントを作成しますが、「信頼性が低くなる恐れのあるドキュメント」より「開発担当者に必要なものだけあればいい」と考えます。しかし一方、会社側からすると1つのソフトを完成させるまでの途中の生産物を必要としてきます。これがくせもので、開発中の仕様変更やバグによる修正などにより、これらの生産物の信頼性は月日が経つごとに薄れていくものです。こうなると一番信頼がおけるのは、いつもソースコードになってしまうのです。あまり良い事ではありませんが、これが現実で

す。もちろん、見やすく書いてあるドキュメントが信用できるなら、それを見るのが一番です。

バグの少ないソフトを作成するには検討・設計工程におけるバグを押さえておくべきです。ZIPCでは、ZIPCシミュレータがこれをサポートしてくれました。状態遷移表を作成し最低限の仕様を記述しておけば、大筋が試験できてしまうのです。ソースも何も書いていないのに動いているのです。これには驚きました(私の考える高度な試験が出来ないのが不満ですが・・・)。

このZIPCシミュレータを使用しながら修正をZIPCのドキュメントに加え、膨大なプログラムが完成してゆけば、ドキュメントの信頼性は上がるしプログラムの品質も上がるので、最高のものが出来るでしょう。さすがに全部はZIPCで出来ませんが、それは使い次第でどうにでもなるでしょう。

■ ZIPC を使用した開発

はじめてZIPCを使用した開発をするので、万一失敗してもやり直しができる程度の小規模プログラムを開発しました。規模としては2Kステップ弱、イベント数が5、状態数が3といったものです。

1) 開発工程

ZIPCの状態遷移表やFNCなどの設計書が基本となる設計方法をとりました。基本設計、機能設計などでZIPCの様々な設計書を作成し、出来た時点でシミュレートします。バグがあれば設計書を直し、またシミュレートします。これが完了した時点でソースに落とすべく細かい設計書を作成し、一気にジェネレートをかけCソースコードを作りあげます。この後、コンパイルエラーや様々な試験でのバグが発生した場合、また頭からもどりZIPCの設計書を修正し、ジェネレートする。つまり、設計書とソースコードは常に一体なのです。(図1参照)
試験工程ではソースを見ながらデバッグするのが普通ですが、今回はいつもZIPCの設計書を見ていました。見やすいからです。

2) 開発期間

検討	1週間
設計書(ZIPC)	1週間
シミュレート	1週間
試験(通常的环境)	1ヶ月

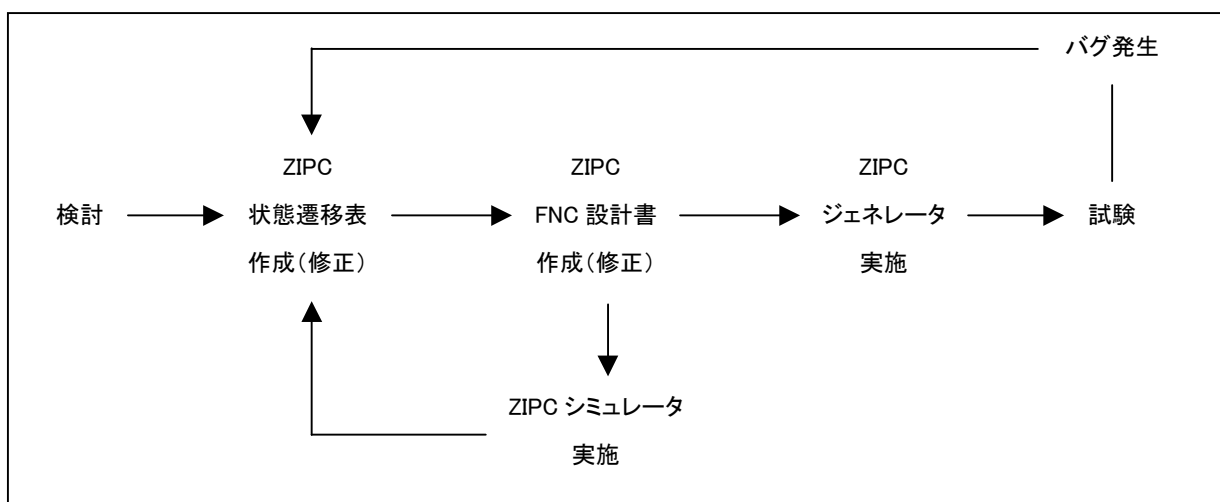


図1 ZIPC を使った設計書 ↔ Cソースコード生成の開発サイクル

3) 設計の内訳

- ・ZIPC未使用のソース:ZIPC使用のソース… 29:71
- ・ZIPC使用のソース中、自分のコーディング無しにて完全に自動生成されたもの、あるいは少し意識して設計し半自動生成されたソース … 24%

5. 評価

納期を2ヶ月後に控えた小規模のプログラム(大規模システムの1パネル)を作成したわけですが、どうにか無事に終わることができました。数字的判断をすると、ZIPCを使用した71%の内の24パーセント、つまり全体の約17%は全くコーディング(設計)を行わずに済みました。これをパーセントで判断して良いかは不明ですが、まぎれもなく工数を短縮しています。また、この部分に関しては試験も行っていないが、バグは無いはずです。もしあったとしたら、それはZIPCジェネレータのバグだからです。そこまで疑ったら、どんなCASEツールでも同じ事が言えるでしょう。もちろん良い事だけではありませんでした。本来、このシステムは回線毎に状態を設け独立した動作をさせたくて、当初は全部で11の状態遷移表を考えていました。自分で手書きの設計をするなら、引数で回線番号を渡し配列で各テーブルやフラグを操作すれば済むところですが、ZIPCには状態遷移のセルに引数を使用することが出来ず、似た様なものを数多く作成しなければいけないのでやめました。結果としてその部分はFNC

にて処理を記述する事にしました。

では、良い点、悪い点を以下に箇条書きで挙げてみます。

[良い点]

- ・概要を書く場合は書式に制限が無いので自由に表現できる。(図2参照)
- ・ドキュメントとソースがリンクしているのでドキュメントを修正すればソースに反映される。

複数のフラグの状態を管理する処理を状態遷移表にすることができるのでフラグが減った。

→ フラグの状態不一致などのバグは無かった。

- ・試験工程でドキュメントが非常に頼りになった。

[悪い点]

- ・タスクのメインルーチンが自由にカスタマイズできない。技を要する。
- ・生成されたCソースが非常に見にくい。
- ・シミュレータ作業・ジェネレート作業が100%結びつかない。
- ・ジェネレートに時間がかかる。かなりのCPUスベックを要すると思われる。
- ・ひな型が無い設計書を作成するのは、慣れないとかなり大変。
- ・

6. おわりに

現バージョンのZIPCは、全ての開発をZIPCでまかなえるものではありません。ZIPCの制限を

	概要設計		Cソースコード
例1	if(送信可能な状態か)	→	if(SendFlag ==OK)
例2	送信可能な状態なら送信する	→	if(SendFlag==OK) TxData(Count,Buffer);

図 2 ZIPC による概念設計と生成結果



考慮しながらの開発は、少し苦労したのは事実 により便利なツールが出るとの噂を聞いています
です。しかし、次バージョンでは柔軟性に富んだ ので、今から待ち遠しい次第です。

[まつだ みつひろ]

ZIPC 各種セミナーのご案内

弊社では、ZIPC Ver.5.0 に関しまして以下のサービスを提供させていただいております。

- (1) ZIPC Ver.5.0 無料セミナー
- (2) ZIPC Ver.5.0 有料セミナー
- (3) ZIPC Ver.5.0 使用受託業務

ZIPC Ver.5.0 開催セミナーについて

No.	セミナー名	内容	対象者	金額
1	ZIPC Ver.5.0 入門セミナー	ZIPC Ver.5.0 の全機能を理解することができます。	エンジニア全般	無料
2	ZIPC Ver.5.0 実習セミナー	ZIPC Ver.5.0 の全機能を一通り操作できるようになります。	エンジニア全般	無料
3	EHSTM Ver.2.0 手法セミナー	拡張階層化状態遷移表設計手法の文法・考え方を理解することができます。	エンジニア全般	2万円
4	ZIPC Ver.5.0 デザインセミナー	状態遷移表・タイミングチャート・メッセージシーケンスチャート等のドキュメンテーションと、その構文・整合性チェックを実習しながら理解する事ができます。	ZIPC Ver.5.0 で設計を行いたい方 上記セミナー1~3の知識を前提として行われます。	2万円
5	ZIPC Ver.5.0 シミュレーションセミナー	ビジュアルプロトタイピング / 自動試験 / 検証を実習しながら理解する事ができます。	ZIPC Ver.5.0 でシミュレーションを行いたい方 上記セミナー1~3の知識を前提として行われます。	2万円
6	ZIPC Ver.5.0 コードセミナー	C プログラムコードの自動生成を実習しながら理解する事ができます。	ZIPC Ver.5.0 でコードを自動生成したい方 上記セミナー！~3の知識を前提として行われます。	2万円
7	ZIPC Ver.5.0 エミュレーションセミナー	ドキュメントレベル実機デバッグを実習しながら理解する事ができます。	ZIPC Ver.5.0 で実機デバッグを行いたい方 上記セミナー1~3の知識を前提として行われます。	2万円

ZIPC Ver.5.0

入門セミナー スケジュール

月	日
8	7
9	4 11 18
10	2 9 16
11	5 13
12	4 11

セミナーについて

- 1) 全てのコースに ZIPC Ver.5.0 試用版 CD-ROM と入門書を、セミナー終了後お持ち帰りいただけますので、自習・復習ができます。予習をご希望の方は、セミナー申し込み用紙にチェックいただければ、セミナー前に教材を送付いたします。
- 2) 全てのセミナーについて出向 / カスタマイズできますので、キャッツ(株)にお問い合わせください。
- 3) 全セミナーの開催日数は1日(10:30~17:00)です。
- 4) 入門セミナー以外のセミナースケジュールにつきましては、キャッツ(株)にお問い合わせください。

ZIPC Ver.5.0 使用受託開発

ZIPC Ver.5.0 を使用して御社の業務の一部ないし全てを受注いたします。実際の業務を ZIPC の専門技術者に発注する事で、ZIPC のノウハウと業務の完了の両方を得る事ができます。詳しくはキャッツ(株)までお問い合わせください。