

# 医療装置システム開発へのZIPCの適用

菱光コンピュータシステム株式会社 ソフトウェア開発事業部  
技術システム第二部 マイコン応用技術課 グループリーダー  
川由 雅昭

## 1. はじめに

我々は、医療装置をはじめ、エアコン室内機やVTR等の家電製品、CD-ROMやDVD等の民生機器、そして電気鉄道分野におけるATO装置（自動運転装置）やTIS装置（車両制御情報管理装置）などの様々なマイコン組込みシステムを開発しています。今回ZIPCを導入した医療装置システムは、その目的から高い信頼性が求められています。

マイコン組込み機器のソフトウェア開発の世界では、組込みシステム自体が大規模化・複雑化・多様化しています。時代はデジタル化に向かい、従来はあまり必要のなかったユーザーインターフェースやネットワーク機能などが必須になりつつあります。また、その中で取り扱うデータも大規模かつ複雑になってきており、従来のハードウェア制御といった観点からの設計だけでは対応しきれなくなってきています。

我々は従来、自社の生産管理規定と品質管理規定に基づき、徹底したウォーターフォール型開発モデルで生産性と品質の向上を図ってきました。マイコンの高機能化を背景に、一方で開発サイクルが短くなる製品に対して、ソフトウェアの

資産化と品質はますます重要になってきており、特に品質に対しては、更に根本的なアプローチが必要になって来ています。

## 2. ZIPC 導入の背景

我々は状態遷移表を、設計に幅広く使用しています。そして、ZIPCを設計フェーズに於ける設計ドキュメントとしての状態遷移表を書くためにエディタとして使用していました。今回の医療装置システムの開発では、次の理由によりZIPCのエディタ機能だけでなくシミュレーション機能とジェネレート機能を使用することにしました。

- (1) H/Wが全面的に見直され、一新されたことによるソフトウェアの再設計
- (2) 多くの機能追加によるシステムの巨大化への対応
- (3) 増え続ける事象と状態による要求仕様の複雑化への対応
- (4) 医療装置システムの開発経験のないメンバーの投入
- (5) 高い品質の確保

この医療装置システムは、以前に開発したシステムからのフルモデルチェンジです。H/W が一新されたことによって、入出力インタフェースがすべて変更になり、ソフトウェアの基本機能設計から、ほとんど新規で開発を行う必要がありました。しかも、新しい機能が数多く追加されているにも拘わらず、開発期間は以前に開発したシステムの約半分程度しかありませんでした。

また、システムのフルモデルチェンジということで、新しく多くの機能追加要求もありました。そこで、この大規模なシステムの開発を短納期で行うために、前回の開発メンバーに加え、新たに医療装置システムの開発経験のないエンジニアを数名加えました。この経験レベルの異なる開発体制で、高い品質と効率のよい開発を行うために ZIPC の適用範囲を広げることとしました。

ほとんどの場合、設計が不十分で曖昧さを残したまま、開発作業が次工程へ進んでしまうことが、品質や生産性を低下させる原因です。この“設計の曖昧さ”を排除するための解決策として、仕様書記述の統一とレビュー品質の向上、そして多人数によるシステムチェックな開発をおこなうために ZIPC をエディタとしてだけでなく、ジェネレーションまで使うことにしました。

### 3.ZIPC 導入のねらい

#### (1) 要求仕様とプログラム処理との連動

組込みシステムの場合、どうしても「早く動かしたい」とか「どうせ仕様変更が入るから」というジレンマからある程度の設計仕様書を書いたら、頭の中で設計し、後はプログラムを書いて試験を開始する。そして、一所懸命に試験をしてバグを潰し、品質を高めていくというような開発になることがあります。その結果、「顧客の要求する仕様通りに実際のプログラム処理が設計されているか」とか「この要求仕様はプログラムのどこに書かれているのか」など、レビューを実施しても思うように設計品質は向上しません。状態遷移表設計では、要求仕様から事象と状態を抽出するため、要求仕様と設計仕様書、そしてプログラムは連続的に繋がります。また、作る（作った）ソフトウェアと仕様書の関係が担当者以外には理解できないと言うようなこともなくなります。

#### (2) 複雑かつ、巨大化する事象と状態への対応

開発するシステムが複雑化・巨大化する中、いくらわかりやすく且つ、できる限り詳しく設計仕様書を文章で書いても、「漏れ」や「抜け」を完全になくすことはできません。文章の場合、仕様書の記述方法を統一しても設計やレビューに時間をかけても、かけた時間分の成果がないのではないかと思うことがあります。

状態遷移表を使用することで、「いつ」「どこで」「何をする」ということが明確になるために、設計やレビュー時に「漏れ」や「抜け」が発見し易くなります。ZIPC を用いて設計フェーズに重点を置くことで、設計品質が向上します。設計品質が向上すれば、下流の試験工程での負担が軽減できることとなります。

### (3) 仕様変更や障害改修における影響範囲の検討の徹底

開発工程が進むにつれて、仕様変更や仕様追加が頻繁に発生します。この仕様変更や仕様追加の際に、上流工程に戻り、影響範囲をしっかりと見極め、再設計を行わないと障害のモグラ叩き状態に陥ることとなります。(これは、精神的にかなり辛いです)

これに対し、状態遷移表設計手法を用いた場合は、状態遷移表から変更・追加作業をすることになるため、状態遷移表から再設計を行うことになり、事象と状態のすべての組合せを、この設計時に強制的に検討させられることとなります。これにより、プログラム変更による障害波及の発生を低減できます。

### (4) 設計ドキュメントとソースコードの完全一致

最初はどのような開発でも何らかの仕様書はあるはずですが、途中で仕様変更や仕様追加が入るのが常です。設計製作 評価と開発を進めていくと、納期に追われて仕様変更や仕様追加を仕様書に反映する時間がとれなくなって、ソ

ースプログラムのみが先行し、最後にはソースプログラムだけが正しいということになりがちです。

その点、ZIPCを使用してソースコードをジェネレートすれば、設計ドキュメントとソースコードは、いつでも完全に一致することとなります。これで、開発途中で問題が発生しても、ソースプログラムと完全に一致している設計ドキュメント(状態遷移表)を照合することにより問題解決のためのレビューが実施できます。

## 4. システムの特徴

今回 ZIPC を適用した医療装置システムについて、その概要を紹介します。

CPU 300MHz (メモリ 32MB)

リアルタイム OS QNX4 を搭載\*1

- ・制御系、データ処理系、表示系を含め、18 タスクで構成
- ・PCMCIA などの、3 つのドライバを搭載 (すべて、自社開発)
- ・リアルタイム描画のための、自社開発の GUI を搭載

数多くの情報をリアルタイムに制御し表示

(\*1) QNX4 は、QNX Software Systems が開発した、POSIX 仕様準拠のマイクロカーネルアーキテクチャを採用している RealtimeOS です。

表 1 に今回の開発で作成した STM のサイズを示します。

表1 作成したSTMのサイズ

タスク機能	STM枚数	イベント数	状態数	関数数	ステップ数
制御系	64	268	140	324	7595
データ処理系	47	176	93	128	4557
表示系	105	222	2095	2277	9548

(ステップ数の単位:Kstep)

## 5. 開発スタイルの変化

今回の医療装置システム開発に ZIPC を導入し、我々はリスクドリブン型のインクリメンタル開発を目指しました。(図1参照)

インクリメンタル開発は、従来から知られているウォーターフォール型開発とスパイラル型開発を融合させた、繰り返し型の開発方式です。

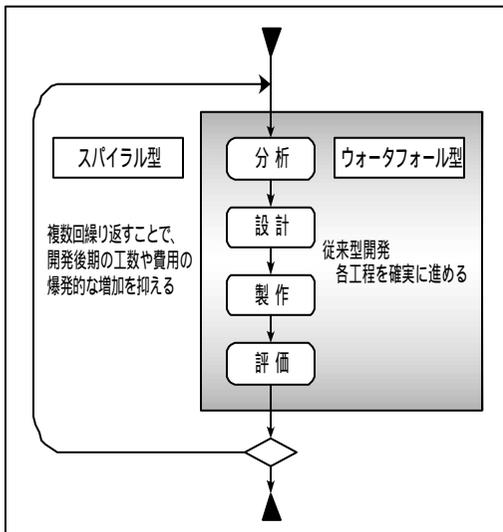


図1 開発プロセス

近頃、リスクドリブンという言葉をよく聞きます(ちょっと流行みたいです)。リスクドリブンとは、「重要な部分を先

行して確認していく」ということで、従来のようなウォーターフォール型の開発では、開発期間の後半で起こると大きな問題になるような“致命的”な問題を開発の初期に確認し、その開発におけるリスク(開発後期での仕様変更や障害による開発工数の爆発的増大など)を減らすことを目的としています。

例えば、RTOSの動作や入出力インタフェース、通信機能などの基本部分を、まず始めにそれらの機能を組み込み確認する。次にそのシステムの核になる部分についての機能を組み込み確認する。タイミングが厳しい部分やソフトウェアとして懸念される部分をできる限り早期に実装し、確認してしまうということです。

つまり、実現上の重要な課題はすべて開発の初期段階で確認し、後半は余裕をもって作業を進めていけることとなります。すなわち、「後はパワーだけ」ということです。ZIPCの一貫した環境は、このインクリメンタル開発に向いていると思います。今回のシステム開発では以下の4つの開発プロセスに分割して取り組みました。(図2参照)

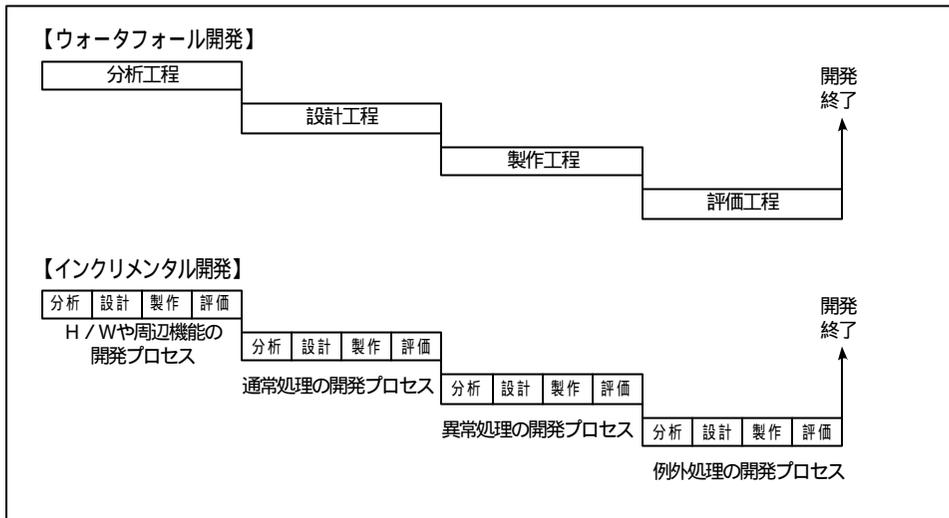


図2 インクリメンタル開発

- (1) H/W や周辺機能に依存する部分を  
開発するプロセス
- (2) 各機能の通常処理を開発する  
プロセス
- (3) 各機能の異常処理を開発する  
プロセス
- (4) システムの例外処理を開発する  
プロセス

## 6 . ZIPC 導入の効果

ここまで、ご紹介したとおり、我々は医療装置システムの開発にあたり、ZIPCを適用しました。ここでは、ZIPCを適用した効果を紹介します。ここで使用するデータは、医療装置システムをフルモデルチェンジする前に開発したシステム

(旧システム)と今回 ZIPC を導入してフルモデルチェンジした医療装置システム(新システム)のソフトウェア開発に関するデータです。

- (1) 設計品質の向上により総合生産性と品質が向上

状態遷移表を使用して設計を重視したことで、外部設計と内部設計を合わせた設計フェーズの割合は大幅に増加しました。しかし、試験フェーズの割合は大幅に減少し、トータルの生産性は20%程度向上しました。これは、ZIPC で状態遷移表を設計し、論理シミュレーションでの検証が効果的であったことを示しています。開発フェーズの割合と生産性を表 2 に示します。

表2 開発フェーズの割合と生産性

	外部設計	内部設計	製作	評価	生産性
旧システム	13%	16%	30%	41%	15.7Step/Hr
新システム	18%	34%	25%	23%	19.4Step/Hr

## (2) ZIPCのジェネレーション機能の効果

フルモデルチェンジ前のシステムとフルモデルチェンジ後のシステムのコードサイズとステップ数に関するデータを図3に示します。

### a) ジェネレータにより生産効率は向上する

フルモデルチェンジによる新規機能の追加により、コードサイズは約2.5MB増加しました。しかし、ZIPCのジェネレータ機能を使用したことによって、我々がコーディングしたステップ数は旧システムと同じぐらいのコーディング量でした。今回開発したシステムにおける自動生成率は、約42%という結果になりました。

### b) コード生成効率は予想以上に良い

自動生成では、ある程度の冗長なコード生成は仕方がないと思っていましたが、従来機能部分のコードサイズと比較しても、ほぼ同等の結果を得られています。

## 7. おわりに

ツールを導入する場合、その理由は「ソフトウェアを可視化する」とか「上流工程からモデリングすることで品質を改善する」といった定性的な場合がほとんどです。しかし、本質的に現場では、特にマイコン組み込みシステムの世界では、ソフトウェア開発を「職人」の仕事から「工業化(システムチック)」することが、ツールを導入するひとつの大きな理由でもあり、目的の一つであると考えています。また、ZIPCは、ソフトウェア開発のノウハウを蓄積できる環境を提供するツールであり、従来型の開発ではな

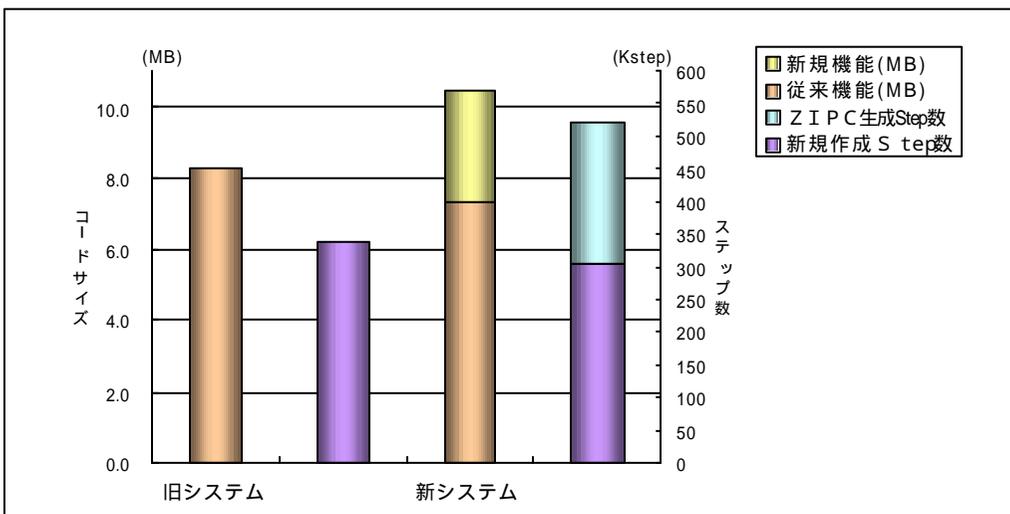


図3 コードサイズとステップ数の比較

かなうまくなかった「ノウハウの共有」といったことを容易に実現してくれる可能性を持ったツールでしょう。ZIPC は、ソフトウェアを開発するレシピを整備し、誰もが同じように高品質なソフトウェアを開発する環境を提供してくれるのではないかと思います。

そして、新しいツールを導入し、実用レベルにまで持ち上げることは、簡単なことではありません。初期の導入コストやツールに関する教育や環境整備など、クリアしなければならない課題も多くあるのが現実です。また、状態遷移表設計手法を使いこなすには、状態と処理を分析して区別する訓練や教育が必要になります。ZIPC を用いて状態遷移表設計手法で開発をやり抜けば、必ず成果が得られるものと確信しております。

ここまで色々と書いてきましたが、我々はまだZIPCの持つポテンシャルの、ほんの一部しか使っていません。今回ZIPCのエディタ機能 シミュレータ機能 ジェネレータ機能を使用しましたが、ZIPCには、ATV機能による自動試験機能や上流設計におけるUMLとの連携など、まだまだ多くの可能性があることがわかりました。今後は、ZIPCのポテンシャルを更に引き出し、生産性と品質を向上させていくことが、我々の課題です。

最後に、21世紀のZIPCが更なる進化を遂げ、我々組込みシステムのエンジニアに明るい未来を与えてくれることを期待しています。

## 第4回組込みシステム開発技術展に出展!

展示会場名	第4回組込みシステム開発技術展 (通称:ESEC)
開催期間	2001年6月27日(水)~2001年6月29日(金)
場所	東京ビッグサイト
出展製品	ZIPC2001 VisualSpec ASN.1 ツール オブジェクト指向CASE ツール
併催企画	ワークショップ開催! 日時: 6月29日(金) 13:30~14:30 「組込み開発環境紹介 - ZIPC2001, VisualSpec, オブジェクト指向CASE ツール」

当日は弊社ブースにおきまして、「ZIPC2001」をはじめとするキャッツ(株)の新製品と、技術コンサルティングのご相談を承っております。

- リアルタイム制御系CASE ツールZIPC 最新バージョン「ZIPC2001」の発表!
- システムレベルの設計をサポートするSpecC 言語をビジュアルに作成する「Visual1Spec」の最新バージョン!
- プロトコル構文規定言語ASN.1 ツール」の最新バージョン!
- リアルタイム制御系オブジェクト指向CASE ツールの出展!

また、当日は弊社ブースにて各技術担当者・コンサルティング担当者によるマンツーマン形式のデモンストレーションを行います。技術的なご相談を承りますので、是非この絶好の機会をご利用ください。