

レーザー加工機におけるCATS社製ツールの適用検討

日本車輛製造株式会社
レーザー機器部長
西脇 靖樹

1. はじめに

私がZIPCと出会ったのは、部内でのソフトウェア開発に疑問を持ったことにはじまる。会社人生活の大部分をメカ設計に携わってきた者としては、1枚の機械図面が製品に係る全てのメンバーの共通設計書となり、レビューを始め製作から品質管理に到るまで使用されることは至極当然のことである。しかし、ソフトウェアの開発現場を眺めてみると、非常に脆弱な設計書（頭の中やメモ）しかなく、レビューも満足にできない状況である。実際、私自身も若い頃には、C言語やN88Basic言語を用い、レーザーマーカの組み込みソフトの開発に携わった経験があるが、残念ながらそこには機械図面に相当する設計書はなかった。

そこで、当社の組み込みソフトウェアの開発担当部門に相談をしたところ、状態遷移表を用いた「モデルベース設計」の紹介を受けた。早速、CATS社塚田様がMONOistに寄稿されているHP「状態遷移表による設計手法」を一読させて頂いた。これを読み進めるうちに、組み込みソフトウェアの開発において、状態遷移表が共通設計書となり得ると考え、導入の検討を始めたわけ

ある。

本稿では当部の商品であるレーザー加工機へのZIPCの適用検討の事例について紹介する。

2. 導入の背景

2.1 これまでの設計手法

当部のこれまでのソフトウェア開発は、個人が簡単な状態遷移図をメモ程度ですませて、コーディング作業までをしていた。言い換えれば、個人の能力やスキルの高さに依存している、属人化である。

実際、小規模なソフトウェアの開発の場合、個人の能力が高ければ、一つの状態遷移図で全体の流れを直感的に捉え、むしろ効率的であったかもしれない。しかし、大規模化により複雑化が増すソフトウェア開発において、従来方法では、全体の流れを捉えきれず、制御の前提条件や異常ケース等の詳細部の把握も不十分となり得る。結果として品質への不安が残る。また属人化は、個人への高負荷に拍車をかけ、最終的には組織の硬直化を導く。

2.2 導入の背景

先にも述べたが、当部のソフトウェア開発における課題を列挙する。

- ソフトウェア開発の属人化
- 設計書の不備
- 仕様変更への対応が面倒
- 開発工数の増大

上記課題は全てに関連性がある。特に設計書の不備の影響は大きく、メカ側とのレビューが不十分なことによって生じる手直し、またカスタマイズ等の仕様変更に伴い発生する工数増大の要因となっている。

そこで、「モデルベース開発」を推進する第一歩として、ZIPCの導入を検討、その効果を検証した。



図1 当社レーザー加工機HC600



図2 HC600の操作盤

3. 適用検討

3.1 適用対象

今回「モデルベース設計の対象としたのは、ファイバレーザー加工機 HC600 の操作画面である。HC600 は弊社の最新機で、精密钣金に特化したレーザー加工機（図1）である。加工機正面右側には（図2）に示した操作盤を備え、これを用いて加工機始動（図3）と加工機停止（図4）の制御を行っている。特に、液晶ディスプレイ上の操作画面では、他装置との連携をおこなうためのカスタマイズボタンが追加されることもある。

今回の導入検討では、「この操作画面を再設計する」として状態遷移表の作成とコーディング作業までを行った。

3.2 状態遷移表の作成

状態遷移表を作成するにあたり、まず状態を洗い出し、現在の状態があるイベントが発生した時にどのような処理が実行され、次の状態に遷移するかを分析し設計を行った。イベントとは、遷移をもたらすための人によるSW操作やセンサからのインプット情報のことである。

次にイベントリストを作成した。イベントリストを作成する際には、直接的な操作SWの名称で

表1 状態遷移図と状態遷移表の比較

	状態遷移図	状態遷移表
表現概要	状態の流れを表現	全てのイベントと状態の組み合わせを表現
メリット	概要を見やすく表現できる	制御の前提条件、異常ケースを含めて全てを網羅的に表現できる
効果	直観的に概要を把握できるため、分析効率が向上する	「モレ」「ヌケ」を発見することができるため、品質が向上する
適合プロセス	要求分析フェーズに適している	設計フェーズに適している
その他の優位性	—	修正・変更が容易に行える

出所：MONOist. 組み込み開発、状態遷移表による設計手法（1）.
http://monoist.atmarkit.co.jp/mn/articles/1204/06/news001_3.html

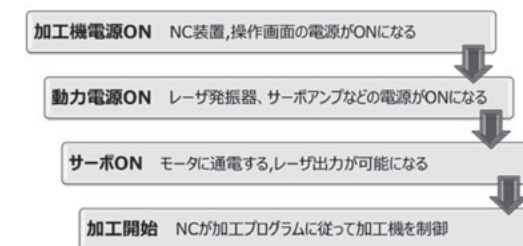


図3 加工機始動フロー

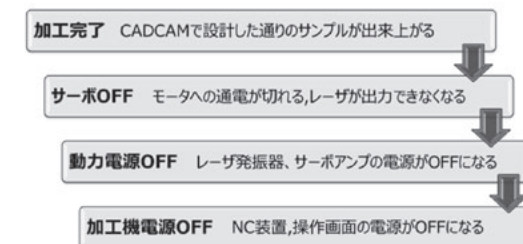


図4 加工機停止フロー

はなく要求名とした。これは、他のSW操作から既存のイベントの発行ができるからである。状態に関しては、(図3)(図4)に示す加工機の始動・停止フローから状態を抽出した。また加工機の動作上、ドアの開閉状態も機能に依存しているため状態に追記した。最後に、加工機の処理を基に、表の該当する箇所を埋めることとした。

今回の場合、液晶ボタンに関する箇所は、3.1で記載したように顧客毎にカスタマイズする可能性が高いイベントである。このイベントを階層化させ、下位の状態遷移表に移行させたことで検証範囲は最小化し、仕様変更、改造、メンテナンス性の向上が見込まれる(表3)。

3.3 階層化機能を用いた状態遷移表

(表2)のように状態遷移表を作成したが、SWの追加、削除のように仕様変更した場合、状態遷移の階層が深いため状態遷移表への影響範囲が多くなる。その結果、検証等に時間を要することになる。そこで、上位に概要、下位に詳細を記述することで、わかりやすく構成する階層化状態遷移表に修正した。

4. 成果

今回の適用検討の成果として、状態遷移表というメカ側とソフト側の両者間の共通の設計書が作成できたことで、情報の共有化=見える化が図られ、以下の相乗効果を生み出している。

- ①ソフトウェア開発の属人化の排除
- ②マトリクス構成によるモレ・ヌケ・矛盾点の早期発見
- ③異なる視点からのレビュー参加による品質向上

次に開発工数の削減が挙げられる。従前と比べてコーディングまでの開発工数は30%削減できている。これに関しては、試験的に要求仕様の一部追加したこともあり、実際には30%以上の効果があったと推察する。これもZIPCの導入成果と言える。

5. 今後の展開

今回の導入検討は、レーザ加工機HC600の操作画面の再設計であることから、ソフトウェアに対する要求仕様自体は既知のものであり、比較的容易に取り組むことができたと考える。

しかし、今後より大きなソフトウェア開発となった場合に、果たしてモレ・ヌケ・矛盾点が少なくレビューに活用できる分かり易い状態遷移表が作り込めるのか、不安点は残るがメンバーのスキルアップに期待している。

またZIPCには豊富な機能、オプションが準備されている。その中で3次元CADデータを利用し、PC上にモデル化することが可能なVPSというツールに着目している。我々機械メーカーにとって、実機が存在しない段階で機構制御ソフトウェアの動作を動的に検証できるということは、ソフトウェアばかりでなく実機の品質向上にも効果が期待できる。それに加え、開発期間の短縮にも大きく寄与できると考えるからである。

表2 状態遷移表の例

0	電源OFF	電源ON					
		動力電源OFF		サーボOFF		動力電源ON	
		Fアロック	Fアオープン	Fアロック	サーボON	Fアロック	Fアオープン
操作電源切	/	電源OFF処理 電源ON:動力電源OFF	電源OFF処理 電源ON:動力電源OFF	電源OFF処理 電源ON:動力電源OFF	電源OFF処理 電源ON:動力電源OFF	電源OFF処理 電源ON:動力電源OFF	電源OFF処理 電源ON:動力電源OFF
操作電源入	電源ON処理 電源ON:動力電源OFF	/	/	/	/	/	/
運転準備	/	Fアが閉まっている Fアロック処理 電源ON:動力電源ON:サーボON:Fアロック	Fアが開いている 電源ON:動力電源ON:サーボOFF:Fアオープン	/	/	/	/
非常停止	/	/	非常停止処理 電源ON:動力電源OFF	非常停止処理 電源ON:動力電源OFF	非常停止処理 電源ON:動力電源OFF	非常停止処理 電源ON:動力電源OFF	非常停止処理 電源ON:動力電源OFF
Fアオープン	/	/	/	/	Fアオープン処理 電源ON:動力電源ON:サーボOFF:Fアオープン	/	/
インターロックイベント	/	/	/	/	インターロックイベント処理 電源ON:動力電源ON:サーボON:Fアオープン	インターロックイベント処理 電源ON:動力電源ON:サーボON:Fアオープン	インターロックイベント処理 電源ON:動力電源ON:サーボOFF:Fアオープン
JOG	/	/	/	JOG処理 電源ON:動力電源ON:サーボON:Fアオープン	/	/	/
Fアロック	/	/	/	/	/	/	電源ON:動力電源ON:サーボON:Fアロック

表3 階層化した状態遷移表の例

00.1.1 main				00.1.1 sub				
00.1.1 main	電源OFF	電源ON		00.1.1 sub	サーボOFF		サーボON	
		動力電源OFF	動力電源ON		ドアオープン	ドアロック	ドアオープン	
操作電源切	/	電源OFF処理	電源OFF	電源OFF処理	電源OFF	電源OFF	電源OFF	電源OFF
操作電源入	電源ON処理 電源ON:動力電源OFF	/	/	電源ON処理 電源ON:動力電源OFF	/	/	/	/
運転準備	/	Fアが閉まっている Fアロック処理 電源ON:動力電源ON/0.1.1>1	Fアが開いている 電源ON:動力電源ON/0.1.1>0	/	/	/	/	/
非常停止	/	/	非常停止処理 電源ON:動力電源OFF	/	/	/	/	/
液晶ボタン	/	/	/	液晶ボタン インターロックイベント サーボON:ドアオープン	サーボON:ドアオープン	インターロックイベント処理 サーボOFF:ドアオープン	インターロックイベント処理 サーボON:ドアオープン	インターロックイベント処理 サーボOFF:ドアオープン
ドア開	/	/	/	JOG処理 サーボON:ドアオープン	/	/	/	/
ドア閉	/	/	/	サーボON:ドアロック	/	/	/	サーボON:ドアロック

6. おわりに

「モデルベース設計」への取り組みは、当部にとり初めての試みであったことからCATS社には、多くの初歩的な質問を投げかけた。その都度、丁寧な回答と共にアドバイスを頂いたことに心より感謝するとともに、お礼を申し上げる。

最後に、適用検討した者として、新たに導入検討をされておられる方に対して導入する際のポイントを提案しておきたい。

- ①最初は小さなシステムから
- ②貪欲で素直な若手社員から
- ③新たな発見はサポートセンターから

ZIPCはツールである。ツールへ期待するのはなく、「どのように活用していくか」明確な目的を持って導入できれば、大いに役立つものと考えられる。