

グローリー株式会社の通貨処理機 ZIPC 適用事例紹介

グローリー株式会社
開発本部 システム開発統括部 設計一部
小家 善朗

1. はじめに

グローリー株式会社では、紙幣や硬貨などを高速・正確に認識して集積したり、金種や各国通貨の選別、紙幣の裏表をそろえて束ねるなどの機能を持った通貨処理機、情報処理機及び通貨端末機器、自動販売機、自動サービス機器などの開発・製造・販売・メンテナンスを行っている。



図1 グローリーの製品群

ZIPC との関係で言えば、弊社では、通貨処理機のソフトウェア開発に古くから社内独自の状態遷移マトリクス設計 CASE ツールを使用していたが、1999 年頃から一部の通貨処理機では ZIPC を採用しはじめており、現在では 15 年以上の実績がある信頼の厚いツールともいえる。

本稿では、グローリー株式会社の ZIPC 適用状況を基に、通貨処理機制御のソフトウェア開発の特長を紹介する。

2. 事例紹介

2.1 開発部門・通貨処理機の特長

弊社の開発部門ではその販売市場別に海外/金融/流通/遊技市場向けの通貨処理機を開発している。

各部門内で開発される多くの通貨処理機は、複数の光学式センサーやモーターなどのアクチュエーターを有し、紙幣や硬貨などを高速（硬貨：30 枚/秒以上、紙幣：12 枚/秒以上）に識別、搬送集積する。駆動系の制御はミリ秒オーダーでの、検出・制御を次々に行っている。

処理機の制御機構はリアルタイム OS を搭載したワンチップマイコンで制御され、開発言語としては、C 言語や C++ 等でプログラミングされていることが多い。

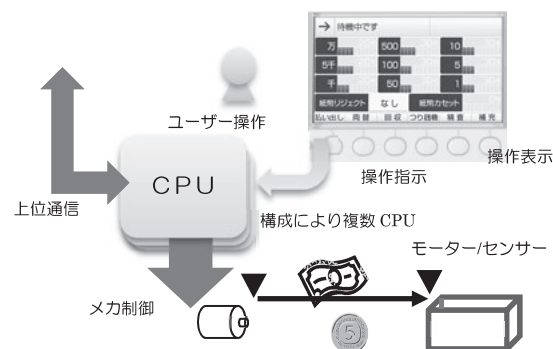


図2 通貨処理機構成イメージ図

開発体制で言えば、市場ニーズから、年々多機種・高品質・短納期開発が求められ、新製品の開発においては、通貨処理機のメカ機構設計、アクチュエーターや CPU の制御基板設計とほぼ同時にソフトウェア開発を行うことが必須となっている。したがって、開発中の制御仕様変更の発生は、前提条件として考えざるを得ない状況である。

通貨処理機開発の特長

- 複数入力複雑に関連したメカ制御が必要
- 高いリアルタイム性が必要
- ワンチップマイコン上で動く環境
- 多機種・高品質・短納期開発
- 変更に強いソフトである

通貨処理機制御を高品質・短納期で開発する生産性向上のためには、状態遷移マトリクスからプログラム実装が可能になる CASE ツールの採用が欠かせない施策の一つとなっている。

2.2 グローリーの適用状況

(1) CASE ツールの採用状況

状態遷移マトリクスからプログラムを自動実装したいという要求は社内でも昔から上がっており、社内の適用事例としては、通貨処理機の機能規模、流元プログラムなどの背景から以下の 3 種類に大別される。

- ZIPC の採用
- 自社ツールの採用
- その他、ツール使用なし

(2) ZIPC 採用経緯

状態遷移マトリクスからプログラムを自動実装する CASE ツールとしては、自社製のツールを従前利用していたが、1999 年頃に ZIPC を知り、アクションの日本語表記を含めた、ツールの操作性や、アクションの分割記述などの機能が充実していること、また、問い合わせ回答など、保守体制も含めて、ZIPC を採用するプロジェクトが増えてきた。

(3) 適用状況

1997 年～現在までの間に出荷された（又は予定している）主な通貨処理機に対して、全社でツールの適用状況は次の通りである。

- ・ ZIPC 約 59.4%
- ・ 自社ツール 約 13.3%
- ・ その他 約 27.3%

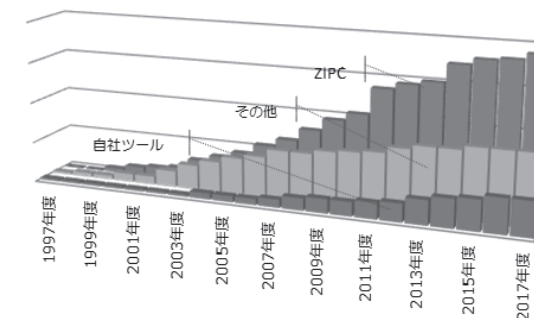


図3 出荷年度別ツール採用数累積

近年はますます、通貨処理機に対する高機能化が求められてきており、それに伴い ZIPC の採用件数が高まっている。

2.3 通貨処理機適用事例

グローリーの代表的な通貨処理機における ZIPC の適用事例を紹介する。



図4 通貨処理機例

代表的な通貨処理機は、大まかに以下のような機能部から構成されている。

- オペレータとのやり取りを行う操作表示部
- 上位端末とのやり取りを行う通信制御部
- 通貨処理機の動作シナリオ制御部
- 紙幣や硬貨の整理搬送を行うメカ制御部
- 通貨処理機を構成するメカブロック制御部

旧来の通貨処理機は、紙幣や硬貨を“識別して数える”ということ自体が通貨処理機の目的であったため、そのプログラム構成はメカ制御部の機能が主体であったが、近年は通貨処理機自体に高機能化が求められており、数えるだけの機械から、“売り上げを管理する”、“他システムと連携して店内現金を管理する”、といったような付加機能が通貨処理機自体に搭載されている。

通貨処理機内のプログラムにおいても、各機能分担をもつ機能部がそれぞれに状態を持って協調

制御するような構成となっており、その状態制御について、ZIPCの状態遷移表（STM）を用いた設計を行い、各機能部に対応した設計モデルを等価にプログラムへ置き換えるといった、開発構成となっているものが多い。

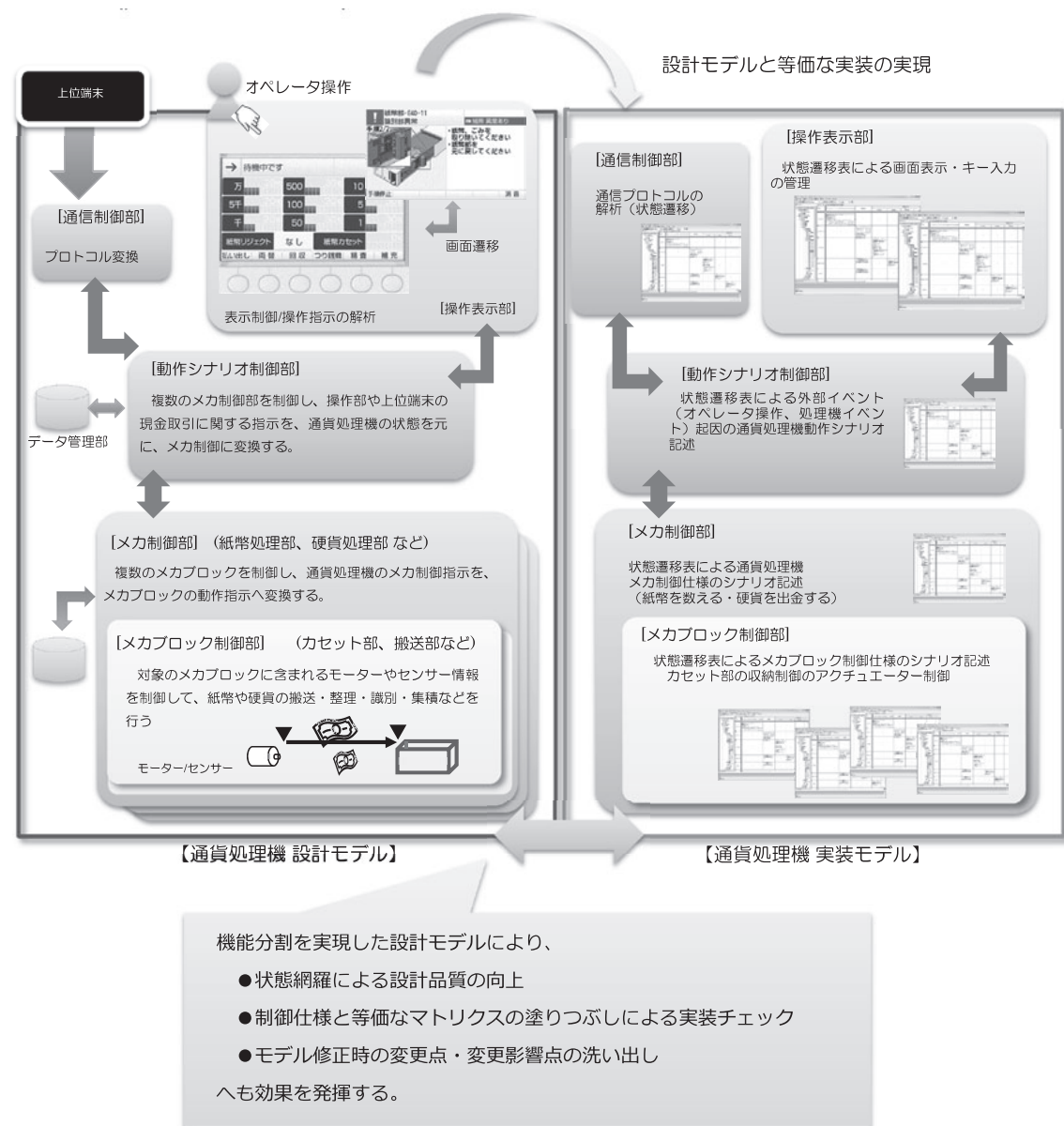


図5 設計モデルと等価な実装の実現

こういったSTMの構成は、自社ツールでも可能ではあったが、ZIPCの優れている面として、

- アクションセルを日本語などの自然言語で記述できる
- アクションセル内のif/switch分岐記述が可能と言った特長から、各振る舞いの記述を整理して高い可読性を保つようにしている。



図6 マトリクス例

可読性の高い整理されたマトリクスは

- 塗りつぶしによる制御仕様の実装チェック
 - 修正時の変更点レビュー
- に対しても効果を発揮できている。

通貨処理機の複雑さによるが、操作表示部、動作シナリオ制御部、メカ制御部、メカブロック制御部などの機能部は比較的複雑になりがちで、ZIPCを用いられることが多く、それ以外の機能部でも、ZIPCによるマトリクス化であったり自社ツールを組み合わせることで実現されたりすることが多い。

2.5 近年の傾向

近年はますます通貨処理機に対する高機能化が求められてきており、それに伴い、マトリクス自体の機能分割も進んできている。10年前であれば、通貨処理機内の一つか二つ程度のタスクにしかSTMを適用していなかったが、通貨処理機の高機能化に伴い、そのタスク毎にメカ制御を行うドライバ部と上位端末からの要求を組み合わせるメカ制御のシナリオへ変換するシナリオ制御部に分けて状態遷移を持ち、それぞれにSTMを設計して実装するスタイルへと変遷していった。

STMをシナリオ系、ドライバ系に大別してリリース年度別に累積すると、その増加は顕著である。

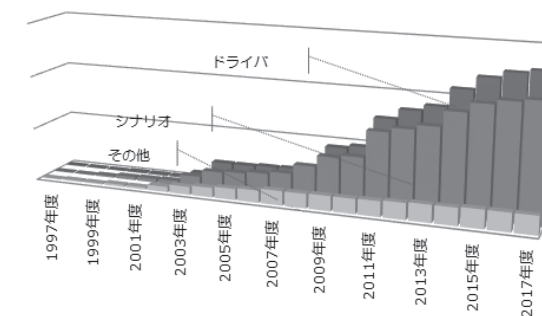


図7 リリース(予定)年度別マトリクス種類累積数

このようなタスク毎の機能分割においても、ZIPCを用いることで、スムーズに対応できている。

たくさんの機能を持つZIPCは、設計者によってモデリングに差異が出やすい傾向があり、差異に起因する不具合を抑制する目的で、各部門で標準の運用を決めるなどの工夫を凝らしている。

3. おわりに

高機能化が求められる通貨処理機の開発において、ZIPCを用いた状態遷移マトリクスの実現はグローリーのソフトウェア開発になくはないCASEツールになっている。

古い通貨処理機にもZIPCが採用されており、スクラッチで開発するテーマが少なくなってきた現在では、旧バージョンのZIPC適用機もまだまだ多く、CATS殿へは旧バージョンのサポート含めて今後ともご協力よろしくお願いたします。

また、本稿の作成にあたって、社内のZIPC採用状況調査や情報共有会を開催する機会ができ、なかなか共有する機会が無かった中、貴重な情報交換の場が作り出せたこと、関係者各位へ感謝しております。