

設計～テストの一貫した工程における 自動化事例QZIF（QC-ZIPC Interface）のご紹介

日本ノーベル株式会社 リサーチ&デベロップメント本部
検証システムビジネスグループ グループマネジャー

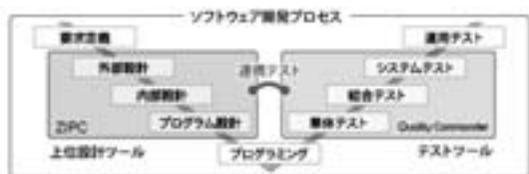
小島 大

1. はじめに

ソフトウェア開発業界には様々な自動化ツールが存在します。

組込みソフトウェアの設計ツールとして広く知られる「ZIPC」、組込み機器の自動テストツール「Quality Commander（クオリティコマンダー）」もその一つですが、ツールベンダーが異なると、折角の自動化もその工程を超えて共有することができません。

私共のユーザー様である某カーナビメーカー様では設計～シミュレーションテストにZIPCを、実機を用いたシステムテストにQuality Commanderを使用することで、バグを徹底的に排除する設計・テストを実現しています。〈図1参照〉



〈図1〉

さらに現在効率化を図るため設計からテストまでの一貫した自動化に取り組んでいます。

本稿ではZIPCの設計データから自動テストを実施するまでの事例をご紹介します。

2. 「S/W設計」と「テスト設計」 二度手間の問題

自動化は工数削減の効果をもたらしますが、ZIPCを用いた設計データの作成、Quality Commanderのテストシナリオ（テスト手順を記述したスクリプト）生成といった準備段階ではどうしても人手が必要となります。

また、両ツールの互換性がないためZIPC、Quality Commanderそれぞれの工程において工数を費やすこととなります。

ここでZIPCで作成した設計データをそのままテストシナリオに変換することができればQuality Commanderを使用する上での工数を大幅に削減することができます。〈図2参照〉



〈図2〉

3. 自動テストシステムQuality Commanderの概要

Quality Commanderは組込み機器の自動テストシステムです。

携帯電話などのボタンをロボットが操作し、画面変化をカメラで撮影して画像としてPCに取り込みます。取り込んだ静止画を期待値画像と自動比較してテスト結果を出力します。〈図3参照〉



〈図3〉

図4がQuality Commanderのテストシナリオです。ロボット操作はテストシナリオに記述したとおりに実行されます。

Quality Commanderのテストシナリオはマウスクリックだけで簡単に記述できますが、簡単であっても人手による作業が発生してしまいます。



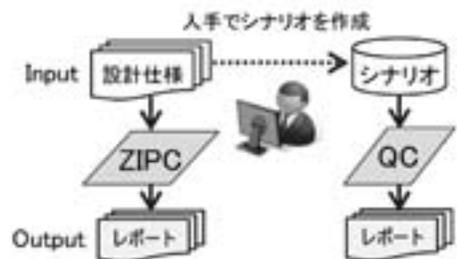
〈図4〉

4. 設計書からテストシナリオの自動生成

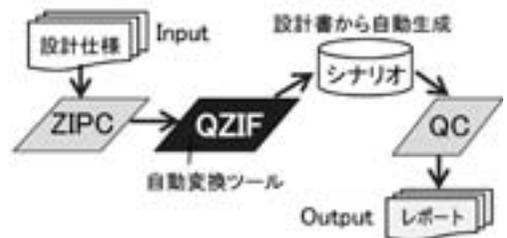
そこでZIPCの状態遷移表の概念をQuality Commanderのテストシナリオに自動変換するツール『QZIF』を開発しました。

QZIFはZIPCで作成した状態遷移設計に対してシミュレーション実行した結果をそのままQuality Commanderのテストシナリオに変換します。〈図5参照〉

(1) これまでの設計・テストの流れ



(2) QZIF 導入で設計～テストを自動化



〈図5〉

Quality Commanderでテストを実行することにより、シミュレーションだけでなく、実際の評価対象機をロボットが自動操作して動作確認することができます。

具体的なシナリオ変換の例としてランプが点灯するスイッチボックス状態遷移例を用いて説明します。(次々ページ図参照)

スイッチボックス状態遷移例は、LED_sample状態遷移表を有し、スイッチボタンをON/OFFする操作に対する各状態が宣言されています。

必要に応じて状態/イベントの開始/終了時にアクティビティやアクションを設定し、FNC 設計書の関数を呼び出すようになっていきます。

この設計書から以下に示す4種類のシナリオファイルを自動で生成します。

(1) 定義シナリオ

ZIPCの設計書をテストシナリオに変換するにあたり、様々な変数を定義する必要があります。ZIPCのプロジェクトファイルに登録されている内容などを「定義シナリオ」として変換します。

(2) 状態遷移シナリオ

評価対象機の各操作に対する状態はZIPCでいう状態遷移表として設計されます。後述する「イベント発行シナリオ」は状態遷移表に表現された遷移上をたどりまますので、先に操作上必要な遷移を全て明記する必要があります。

Quality Commanderのシナリオに変換する際は、状態遷移表で定義される全ての遷移について分岐を用いたり複数ファイルに分割してシナリオに表現します。

(3) 実行シナリオ

続いてFNC設計書を「実行シナリオ」に変換します。

遷移した先でロボットが評価対象機を操作し、自動判定コマンドを実行し、正しい結果が得られたかどうかを確認します。

例えばLEDランプが点灯するのが期待値であった場合、カメラでLEDランプを撮影し、点灯しているのをコンピュータで自動判定します。

(4) イベント発行シナリオ

最後にZIPCで行うシミュレーションの記録をログファイルとして取得します。

ログファイルを解析し、状態遷移表上のどの操作が呼び出されているかをシナリオに記述していきます。

5. 期待される効果

設計からテストまで一貫した自動化を導入することにより、以下に挙げるような効果が期待できます。

(1) 設計書からのシナリオ自動生成により最も効果があるのは、テストシナリオ作成の工数を大幅に削減できることです。

特に次機種の設計/テストにおいては、追加修正を設計書に反映するだけで、次機種のテストシナリオが自動生成できます。

(2) 遷移表のデータを全て書き出すことで自動テスト可能な範囲が広がり、これまで対応できなかった複雑な操作も自動化可能となりました。

また、テストデータも状態遷移表ファイルで管理できるため、テストの漏れがなくなることも大きなメリットと言えます。

(3) 設計とテストの連携と言ってもZIPCとQuality Commanderはそれぞれ独立した環境で使用可能です。例えばZIPCがインストールされていないPCでも、設計書から生成したテストシナリオを用いて自動テストが可能ですので、テスト機の確保もしやすくなります。

6. おわりに

ソフトウェア開発に自動化ツールを取り入れることはもはや常識となり、それだけでは差別化が難しくなってきました。

今後は自動化ツールをいかにして使いこなすかという点に注力していく必要があります。

設計～テストまで一貫した自動化がソフトウェア開発の差別化に貢献できれば幸いです。

日本ノーベル株式会社
リサーチ&デベロップメント本部
〒114-0002 東京都北区王子2-30-2
<http://www.jnovel.co.jp/>

QC-ZIPC Interface による設計～テストの自動化変換例

【LED ボタン BOX の例】



- ①電源ボタン ON で電源投入
- ②「赤 / 青ボタン」で赤い / 青いランプが点灯
- ③電源ボタン OFF でランプ消灯、電源 OFF

