



第4章

SpecC&組み込みCASEツール 活用チュートリアル

村上晋一郎



ここでは、SpecC言語を利用した組み込みソフトウェアの仕様決め、および設計のフローを体験していただきます。使用するツールはグラフィカルにSpecC言語を入力できる「Visual Spec」と組み込みソフトウェア開発ツール「ZIPC」です。Visual Basicで作成したバーチャル・プロトタイプを使って、不具合がないかどうかを視覚的にチェックします。さらに、 μ ITRONをターゲットとして実装し、シミュレーションを行います。サンプルとなるシステムは、本特集第3章でも利用した「車のヘッド・ライト制御」です。VisualSpecとZIPCのトライアル版、およびサンプル・プログラムは本誌付属のCD-ROMに収録されています。(編集部)

本チュートリアルではSpecC言語をビジュアルに表現するツールである「VisualSpec」と、ステートマシンを記述できるツール「ZIPC」を使用して、ソフトウェアの仕様決めから設計までのフローを体験していただきます。

仕様レベルで望まれるのは、ある程度の抽象度をもたせつつ、必要な機能動作を明確に洗い出せることです。Visual Specでは、SpecC言語のビヘイビアで機能動作を定義することにより、高い抽象度で設計を行えます。米国Microsoft社のVisual Basicで作成した仮想ターゲット(バーチャル・プロトタイプ)とリンクさせると、スパイラル・モデルの開発

フローにもとづく仕様分析を行えます。

仕様レベルで作成したビヘイビアは、ZIPCによってマトリックス形式(状態遷移表)に変換できます。状態遷移表にもとづいて、実装を意識しながら設計したり、リアルタイムOSを考慮したシミュレーションも行えます。

ここで、本チュートリアルで行う作業の概要を示します。

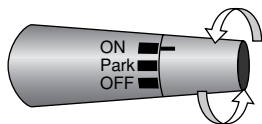
まず、今回開発するシステムの外部仕様を解説します。外部仕様をもとにVisualSpecでビヘイビアを作成して、仕様を明確にします。作成されたビヘイビアはパソコン上で実行できるので、Microsoft社のVisual Basicと組み合わせて動作させ、仕様にもれがないかを検証します。

次にVisualSpecで作成したビヘイビアをZIPCの状態遷移表に変換し、マトリックス上で仕様のもれがないかを検証します。マトリックスが完成したら、リアルタイムOS(今回のターゲットは μ ITRON)を含めたシミュレーションを行い、ターゲットに近いレベルでデバッグを行います。

1 サンプル・デザインと利用ツール

1.1 サンプル・デザインについて

開発するソフトウェアは、本特集第3章で説明した車の「ヘッド・ライト制御」です。ヘッド・ライト・スイッチがユー

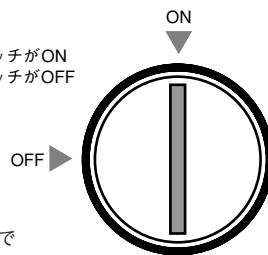


- 「ON」 : ヘッド・ランプ、パーキング・ランプともに点灯
- 「Park」 : パーキング・ランプ点灯、ヘッド・ランプ消灯
- 「OFF」 : ヘッド・ランプ、パーキング・ランプともに消灯

【図1】ヘッド・ライト・スイッチ

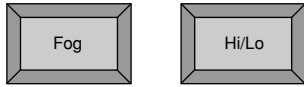
先端部分が可動式の3段階スイッチである。それぞれ「OFF」、「Park」、「ON」の状態になる。

- 「ON」 : メイン・スイッチがON
- 「OFF」 : メイン・スイッチがOFF



【図2】メイン・スイッチ

キーが差し込まれるスイッチである。ここでは「ON」、「OFF」の二つの状態が存在する。



- 「Fog」：フォグ・ランプを点灯・消灯させる
- 「Hi/Lo」：ヘッド・ランプのハイ・ビームとロー・ビームを切り替える
- 「Fog」, 「Hi/Lo」はともにトグル・ボタンとする(押されたらすぐに戻る)

〔図3〕 フォグ・ランプ・スイッチおよびハイ/ロー・ビーム切り替えスイッチ

両方ともトグル形式のボタンである。押されるたびに状態が切り替わる。

ザの入力によりONになったらヘッド・ランプを点灯するなどの制御を行います。仕様は第3章の図4(p.47)の通りで、入力I/Oについては第3章の表1(p.48)を、出力I/Oについては第3章の表2(p.48)を参照してください。

入力の概要は以下ようになります。

- ヘッド・ライトはヘッド・ランプのON/OFFおよび、パーキング・ランプを操作できる(図1)。
- メイン・スイッチはキーを差し込んで、ONとOFFを切り替えることができる(図2)。
- フォグ・ランプ、およびヘッド・ライトのハイ・ビームとロー・ビームはトグル・スイッチで操作する(図3)。フォグ・ランプはONのときに押されるとOFFになり、OFFのときに押されるとONになる。ハイ・ビームとロー・ビームはハイ・ビームのときに押されるとロー・ビームになり、ロー・ビームのときに押されるとハイ・ビームになる。
- ドアの開閉は、スイッチが自動的に検出するようになっている(図4)。

1.2 利用ツールについて

本チュートリアルで使用するツールは、以下のとおりです。

(1) VisualSpec

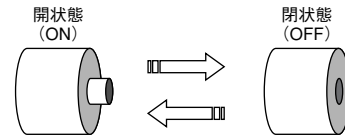
SpecC言語をサポートするシステム仕様を作成するツールです。トライアル版が本誌付属のCD-ROMに収録されています。以下の「制限機能」があることをご理解のうえ、使用してください。

使用期限は2000年9月30日まで。この日以降は、VisualSpecは起動できなくなります。トライアル版では50個のビヘイビアまでしか作成することができません。VisualSpecのいくつかのオプション・ツールについては、本トライアル版では利用できません(詳しくはインストール・マニュアルを参照)。

VisualSpecの「動作環境」は以下のとおりです。

OSは、Microsoft Windows NT 4.0 日本語版(Service Pack 4 以上), あるいはMicrosoft Windows 2000 日本語版, 必要メモリ容量は、34Mバイト(推奨64Mバイト)以上。

(2) ZIPC Ver. 6.0a



ドアには上記のようなスイッチが付いており、開いた状態では突起が飛び出し、閉まった状態では押し込まれた状態となる

〔図4〕 ドア開閉スイッチ

構造はソフトウェアの仕様とは関係ない。ドアが開いたら「ON」になり、閉じられると「OFF」になる。

拡張階層化状態遷移表設計手法をサポートしたソフトウェアの設計、テスト、コード生成、実機エミュレーションを行うツールです。トライアル版が本誌付属のCD-ROMに収録されています。以下の「制限機能」があることをご理解のうえ、使用してください。

使用期限は2000年9月30日まで。この日以降はZIPCは起動できなくなります。

「動作環境」は以下のとおりです。

OSは、Microsoft Windows 98 日本語版, あるいはMicrosoft Windows NT 4.0 日本語版. 必要メモリ容量は、34Mバイト(推奨64Mバイト)以上。


(3) Microsoft Visual Basic

VisualSpec, およびZIPCを使ってバーチャル・プロトタイプを作成する際に必要です。

(4) Microsoft Visual C++ 6.0

VisualSpecで作成したビヘイビアをコンパイルして、アプリケーションを作成する際に必要です。

2 VisualSpecによるビヘイビアの作成

VisualSpecを起動します。作成したドキュメントを初めて保存する場合は、メニュー・バーの「ファイル(F)」→「名前を付けて保存(A)」を選択します。一度保存されたドキュメントは、「ファイル(F)」→「上書き保存(S)」を選択するか、 ボタンをクリックします。

ここで紹介するサンプル記述はCD-ROMに収録されています。「HeadLightCtrl.xml」がVisualSpecで読み込み可能なドキュメントです。

2.1 ビヘイビアの登録

最上位のビヘイビアには「ヘッド・ライトOFF」, 「パーキング・ランプON」, 「ヘッド・ライトON」, 「自動ヘッド・ライトOFF」が存在するので、手順にしたがって登録します。

「Main」コンポーネントを選択して、ポップ・アップ・メニューを表示します(ポップ・アップ・メニューはマウスの右