



# 拡張階層化状態遷移表 CASE ツール ZIPC V10

## 状態モデル CASE ツールのスタンダード

システムの複雑な振舞いを状態モデルで設計、シミュレーション、コード自動生成まで行える  
国産 CASE ツールのデファクトスタンダードでソフトウェアの品質・生産性を向上させます！

### こんな状況はありませんか？

- ◆ 仕様と設計、成果物間で齟齬があり手戻りが多い
- ◆ 複雑な振舞いを多数のフラグや変数で制御している
- ◆ 設計書を作成せず、いきなりコーディングしている
- ◆ ソースコードのみが設計資産になっている

### 期待される品質・生産性向上の効果

- ◆ あいまいな仕様の定義を明確に表現できます
- ◆ 複雑な振舞いも状態モデル設計で整理できます
- ◆ コーディング前に振舞いの正しさをシミュレーションできます
- ◆ 設計した状態モデルがそのまま設計書として資産化できます

### 豊富な事例、実績で導入効果は確実 ～ 事例がダウンロードできます <http://www.zipc.com/> ～

ZIPC V10 は、デジタル家電、通信機器、カーエレクトロニクス、OA機器、FA機器、医療機器、防衛航空宇宙等 の様々な分野で導入実績があります。

- ・ 専用エディタを使うことでドキュメント作成時間が **1/2** に削減された
- ・ コード自動生成でコーディング工数が **1/3** に削減され、コード容量は同等システム手書きの **0.97～1.2** 倍 程度に抑えられた
- ・ 初回の実装デバッグでのバグ発生件数が **1/4～1/30** に削減された
- ・ ZIPC V10 採用箇所のコード行数あたりのバグ包含率が ZIPC V10 未使用箇所の **1/4** に低下

### ZIPC V10 を使った設計・開発の流れ ～ 分析・設計しながらモデルデバッグし品質向上 ～

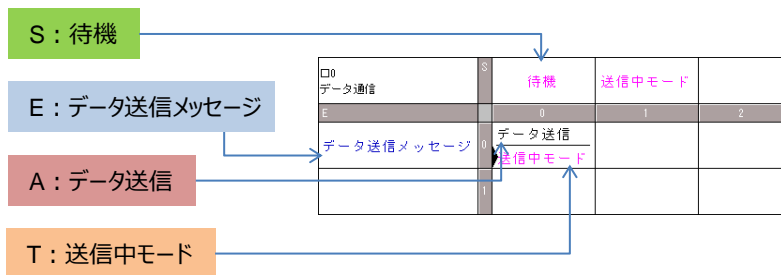
#### ① 対象システムの仕様から状態遷移の要素（SEAT）を分析、抽出する。

例：システムが待機している時にデータ送信メッセージを受信したら  
デバイスドライバにデータ送信を開始して送信中モードに入る。

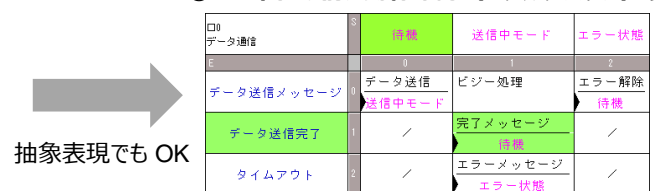
#### SEAT とは

- **S** State (状態)
- **E** Event (事象)
- **A** Action (処理)
- **T** Transition (遷移)

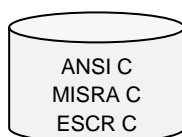
#### ② SEAT を状態遷移表に配置して行く。



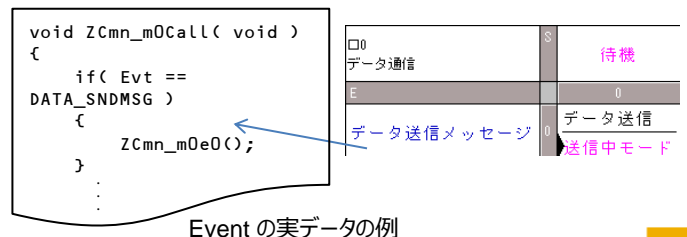
#### ③ コード化の前に動作確認（モデルデバッグ）する。



#### ⑤ モデル+実データから ANSI C コードを自動生成する。 MISRA 準拠、ESCR 準拠のコード生成も可能。



#### ④ 状態遷移、処理条件の正しさが確認されたモデルに Event、Action の抽象表現の実データを定義する。既存の関数等も利用可能。





# ZIPC V10 の機能紹介

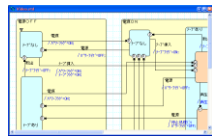
## 状態遷移表専用のエディタでモデリングがスムーズ、効率UP

	10OFF	自己診断中	通常
ライトOFF	ライト無効	ライト有効	
自己診断開始	自己診断開始	自己診断中	自己診断完了
電源OFF	自己診断中	自己診断中	自己診断完了
自己診断完了			自己診断完了
ライトチェック	ライトOFF	ライト有効	ライト有効
アクセルチェック	エンジン停止	エンジン停止	エンジン停止
異常チェック			異常検出

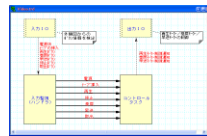
STM



StmDiff (差分検出機能)



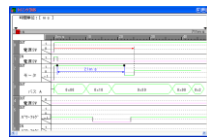
STD



TRD



MSC



TC

### ◆ 5つの設計書でシステムのモデリングを支援

- **STM (State Transition Matrix)**
  - ZIPC V10 の中核になる状態遷移表モデル
  - 階層化に対応
- **STD (State Transition Diagram)**
  - 状態、正常ケースを表現する状態遷移図モデル
  - STMとの互換性があります
- **TRD (Task Relationship Diagram)**
  - システム間、タスク間、資源の関連性を表現
- **MSC (Message Sequence Chart)**
  - システム間、タスク間、資源のメッセージ送受信を表現
  - 自動シミュレーションの入力データにも使用可能
- **TC (Timing Chart)**
  - システムの変数、I/O の変化タイミングを表現
  - 自動シミュレーションの入力データにも使用可能

### ◆ 状態遷移表の差分検出機能により、派生開発や流用開発時の効率がUP

### ◆ ドキュメントチェック機能により、静的な記述チェックをツールで自動化し、人手によるケアレスミスを発見/防止

## 多彩なシミュレーション機能でモデルベース開発を強力に支援

### ◆ モデルベース開発を支援する多彩なシミュレーション機能

- 日本語記述に対応した抽象的なレベルでもシミュレーション可能
- シミュレーションシナリオ作成機能
- Event、State、Action 等のセル単位でブレイク実行が可能
- 変数ウォッチ機能
- ステップ実行機能
- MSC、TC を入力とした自動シミュレーション、検証機能
- マルチタスクシステムに必要な RTOS の動作もシミュレーション可能
- バーチャルプロトタイプと組み合わせてのシミュレーション機能
- STM カバレッジ算出機能

	10OFF	自己診断中	通常
ライトOFF	ライト無効	ライト有効	
自己診断開始	自己診断開始	自己診断中	自己診断完了
電源OFF	自己診断中	自己診断中	自己診断完了
自己診断完了			自己診断完了
ライトチェック	ライトOFF	ライト有効	ライト有効
アクセルチェック	エンジン停止	エンジン停止	エンジン停止
異常チェック			異常検出

バーチャルプロトタイプ

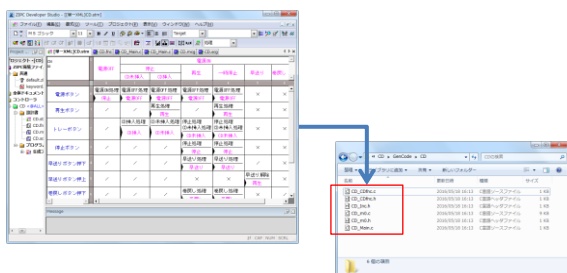


MSC



TC

## モデルからソースコードを自動生成するのでコーディングミスを排除



### ◆ シミュレーションで正しさを確認したモデルから設計通りにコーディング

- ANSI C コード自動生成 : ターゲットを選ばない実装可能コードを生成
- MISRA C コード自動生成 : より安全性を求められるシステムにも適用可能
- ESCR C コード自動生成 : IPA(独立行政法人 情報処理推進機構)が策定した可読性、メンテナンス性が高いコード規約に対応

### ◆ コードフォーマットの統一、ドキュメントとコードが常に一致