組込みコンポーネント仕様（TECS）をサポートするツールの実現とその適用例の紹介

2008年11月20日

キャッツ株式会社

Communication Art Technology Systems

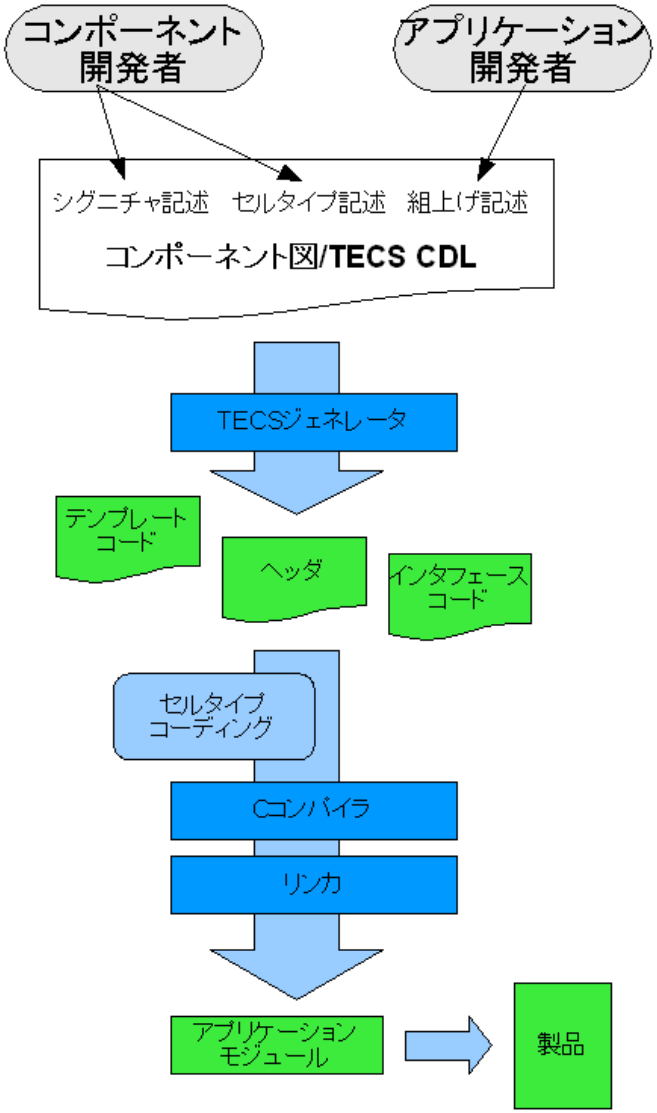
目次(全体の流れ)

- TECSとは
- TECSをサポートする ZIPC Toy!
- ZIPC AUTOSARという構造設計ツール
- AUTOSARとは?
- 構造設計の薦め
- フィーチャモデリングから始まるコンポーネントベース設計の流れ
- フィーチャモデリングと構造設計
- コンポーネントベース設計とプロトコルステートマシン
- 事例:踏み切りモデリング
  - 仕様概要
  - フィーチャモデリング
  - 構造モデリング
  - 出力されたCDL

おおよそ20分で説明します。

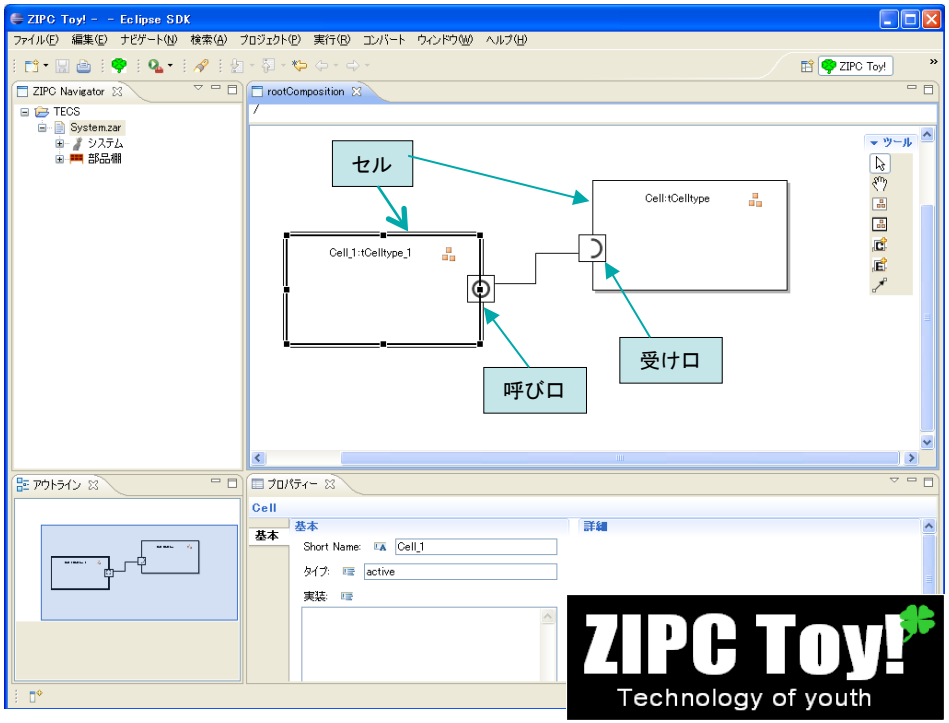
TECSとは

- TECS
  - Toppers Embedded Component System
  - 大規模化、複雑化する組込みソフトの コンポーネント化
  - CDL記述の標準化による 部品流通の促進
- TECS CDL
  - コンポーネント記述言語
    - シグネチャ
    - セル
    - 呼び口
    - 受け口



TECS開発をサポートする ZIPC Toy!

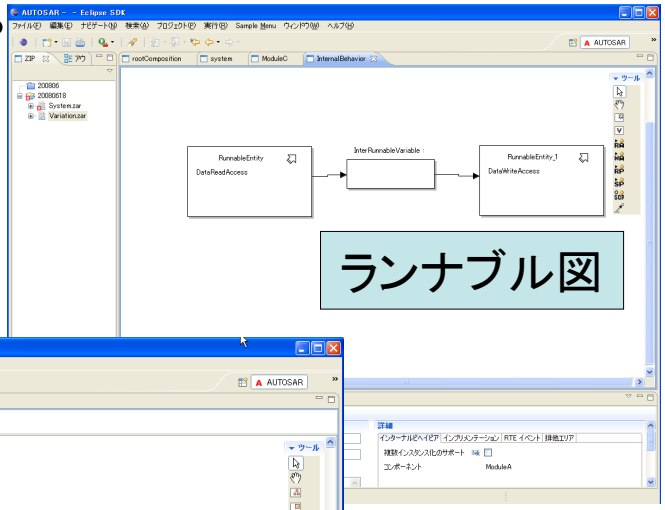
- 構造設計/コンポーネント設計ツール
  - TECSコンポーネントモデル
  - GUIによる設計
    - TECSコンポーネント図
  - 作成したモデルからCDLファイルを出力
- 設計支援機能
  - Validation機能
    - 要素のユニークさチェック
    - シグネチャが一致しているか

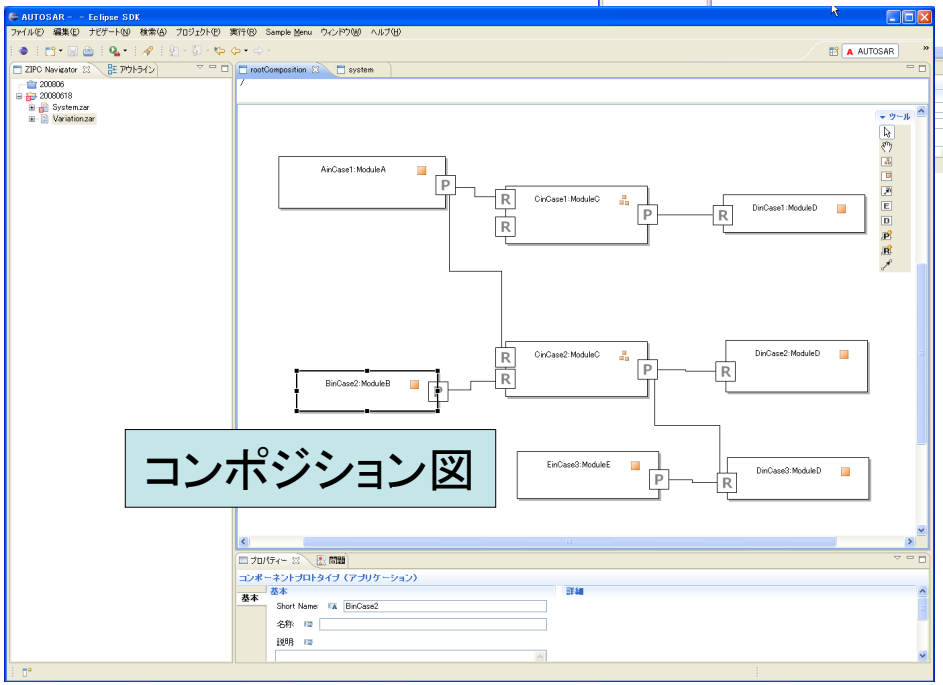


- ZIPC AUTOSARをベースに開発
  - ZIPC AUTOSAR
    ～AUTOSARをベースとした車載アプリケーション開発支援ツール

ZIPC AUTOSARという構造設計ツール（1）

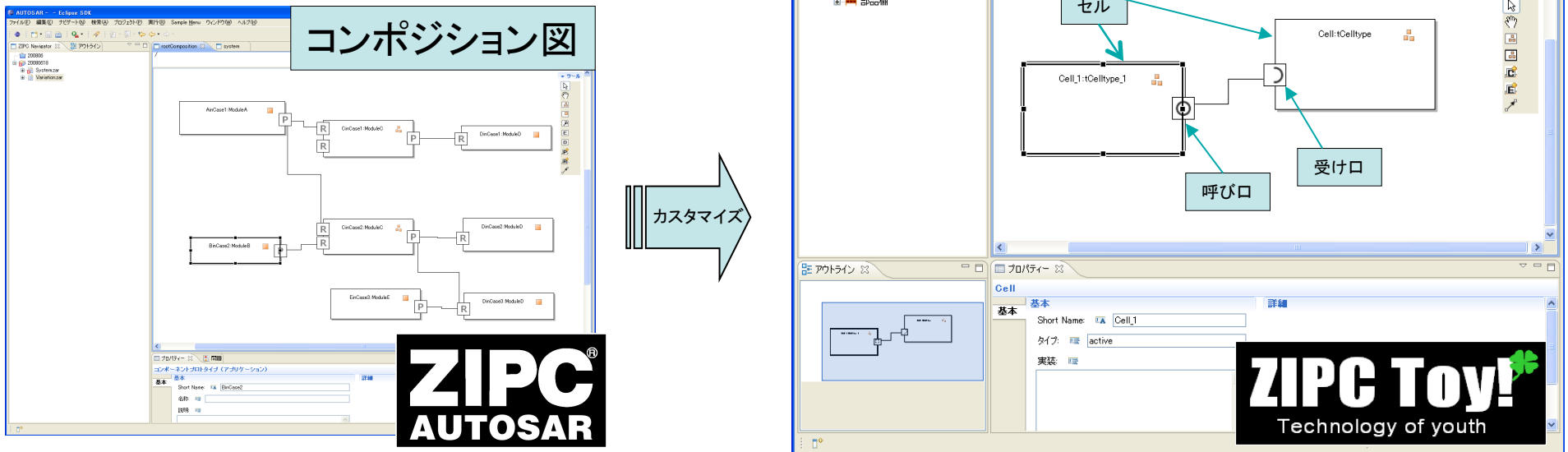
- AUTOSARプラットフォームを意識した設計支援ツール
  以下の4つの図を作成していくことで設計します。
  - コンポジション図
  - ランナブル図
  - SWC&インターフェース図
  - トポロジ＆マッピング図





ZIPC AUTOSARという構造設計ツール（2）

- コンポジション図
  - ソフトウェアコンポーネント間の関係をGUIで記述　⇒ TECS コンポーネント図と非常に似ている！
  - 記述要素も似ている
    - Pポート、Rポート(AUTOSAR)　⇒呼び口、受け口(TECS)
    - コンポーネントプロトタイプ(AUTOSAR)⇒セルタイプ(TECS)



AUTOSARとは？

- 2002年 欧州自動車メーカー等により設立
- 日本国内からも多数参加
- プラットフォーム/インターフェースの標準化
  - プラットフォームベース開発へ

- 国内団体としては、JasParがあります。
  - 非競争領域の協調開発
  - 国内各社からの意見を一本化し、世界標準への貢献
  - AUTOSAR、FlexRayコンソーシアムと協業体制

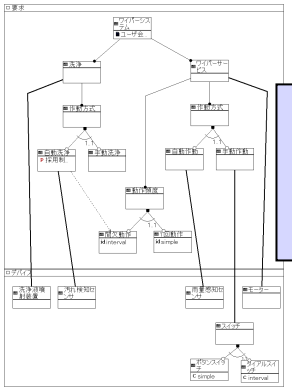
構造設計の薦め

- コンポーネントベース設計の活用
  - コンポーネントベース開発・・・・
    - 用意されたコンポーネントを組合わせることで開発していく
  - 問題点・・・
    - 用意されたコンポーネントがあまりない! ・・・・作るしかない！
      - ⇒ コンポーネントを流通させるためのIF規格で作る（TECS,AUTOSAR等）
    - うまく組合わせられない/どう組合わせる?・・・・フィーチャモデル＋構造設計
      - ⇒ 組合わせを想定して設計する
  - うまい組合わせを想定して部品開発
    - うまい組合わせ？？
      - 想定しているバリエーションを作るための組合わせ

        ↓
    - あらかじめ、バリエーションを考慮した構造設計が必要
      - バリエーション考慮⇒フィーチャモデリングによる変動性分析

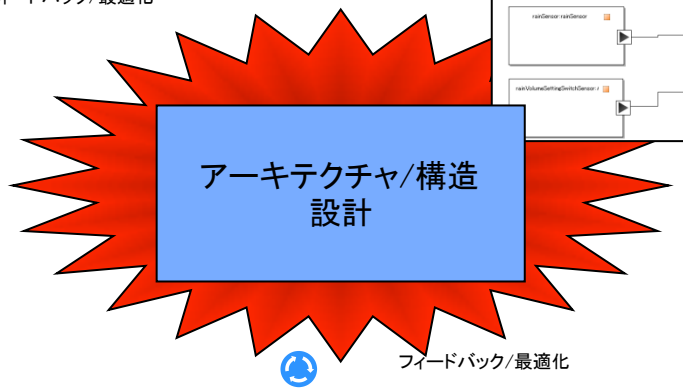
フィーチャモデリングから始まるコンポーネントベース設計の流れ

フィーチャモデル



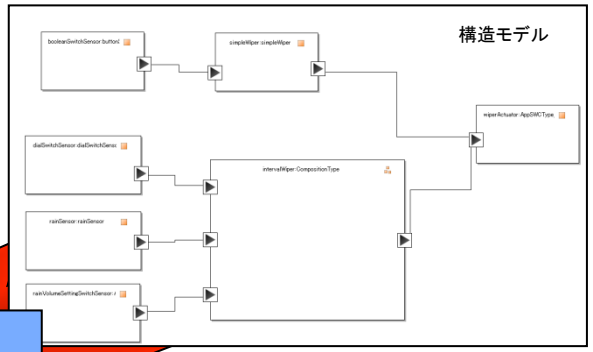
要求/バリエーション
設計

フィードバック/最適化



フィードバック/最適化

構造モデル



振舞い
設計

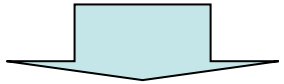
フィードバック/最適化

振舞いモデル

			ﾜｲﾊﾟｰ動作ﾀｲﾐﾝｸﾞ決定	停止中 Entry:ﾀｲﾏ停止	動作中	
					短い Entry:短ﾀｲﾏ起動	長い Entry:長ﾀｲﾏ起動
E				0	1	2	3
間欠時間	停止	0	／	=>停止	／	／
	短く	1	=>短い	／	／	=>短い
	長く	2	=>長い	／	=>長い	／
ﾀｲﾑｱｳﾄ		3	／	ｲﾍﾞﾝﾄ(ﾜｲﾊﾟｰ操作指示)	／	／

フィーチャモデリングと構造設計（1）

- フィーチャモデリングの目的
  - ソフトウェアの変動する要因を設計する
    - どこが変わるのか/変わらないのか
  - バリエーション
    - どのような組合わせは存在するのか/存在しないのか



  - コンポーネントの組合わせ可能性を列挙

- コンポーネントベース設計でのバリエーション
  - コンポーネントを置き換えることで、バリエーションを実現

  - どのような組合わせがあるか要検討

  - フィーチャモデリングと相性が良い

構造設計

- 構造モデル
  - どのような構造の上に機能を実装するのか
  - 要求/バリエーションに対してどのように構造を変化させて対応するのか
- 何を考えるのか
  - どのような機能分割をしてコンポーネントとするか
  - コンポーネント間の関係をどうするか
  - 今後の機能拡張時の拡張方針
- 構造モデルを使用して考えることで以下を考える
  - 再利用性
  - メンテナンス性
  - 拡張性
- 発生するトレードオフ(エンジニアリングデシジョン)を検討する
  - バランスが大事

フィーチャモデリングと構造設計（2）

- フィーチャモデリングをベースとした構造設計
  - 構造とコンポーネント化を意識した機能分割を検討
  - コストのかかる組合わせ/比較的低コストで実現できる組合わせ

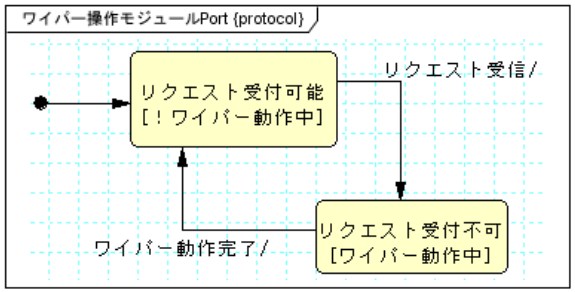
- どのような機能の組み合わせが効率が良いか
  構造設計視点でフィードバックする
  - XXとYYの組合わせは高コストだけど作る？
  - XXとZZの組合わせは要求されていないけど低コストですよ

構造設計とプロトコルステートマシン

- 構造を検討することで、コンポーネントに期待する機能を検討
  - コンポーネントの責務は何か
  - コンポーネントが受け取る情報は何か
  - コンポーネントが出力する情報は何か

  - 責務を分割し、構造を構成するコンポーネントへの割振を検討

- プロトコルステートマシンでポートの振舞いを検討
  - コンポーネントのポートに対しての制約を設計/記述
    - （外部から見て）どのような状態を持たなければいけないか
    - （外部から見て）どのような状態遷移をしなければいけないか
  - コンポーネントはプロトコルステートマシンで指定された振舞いを満たさなければいけない

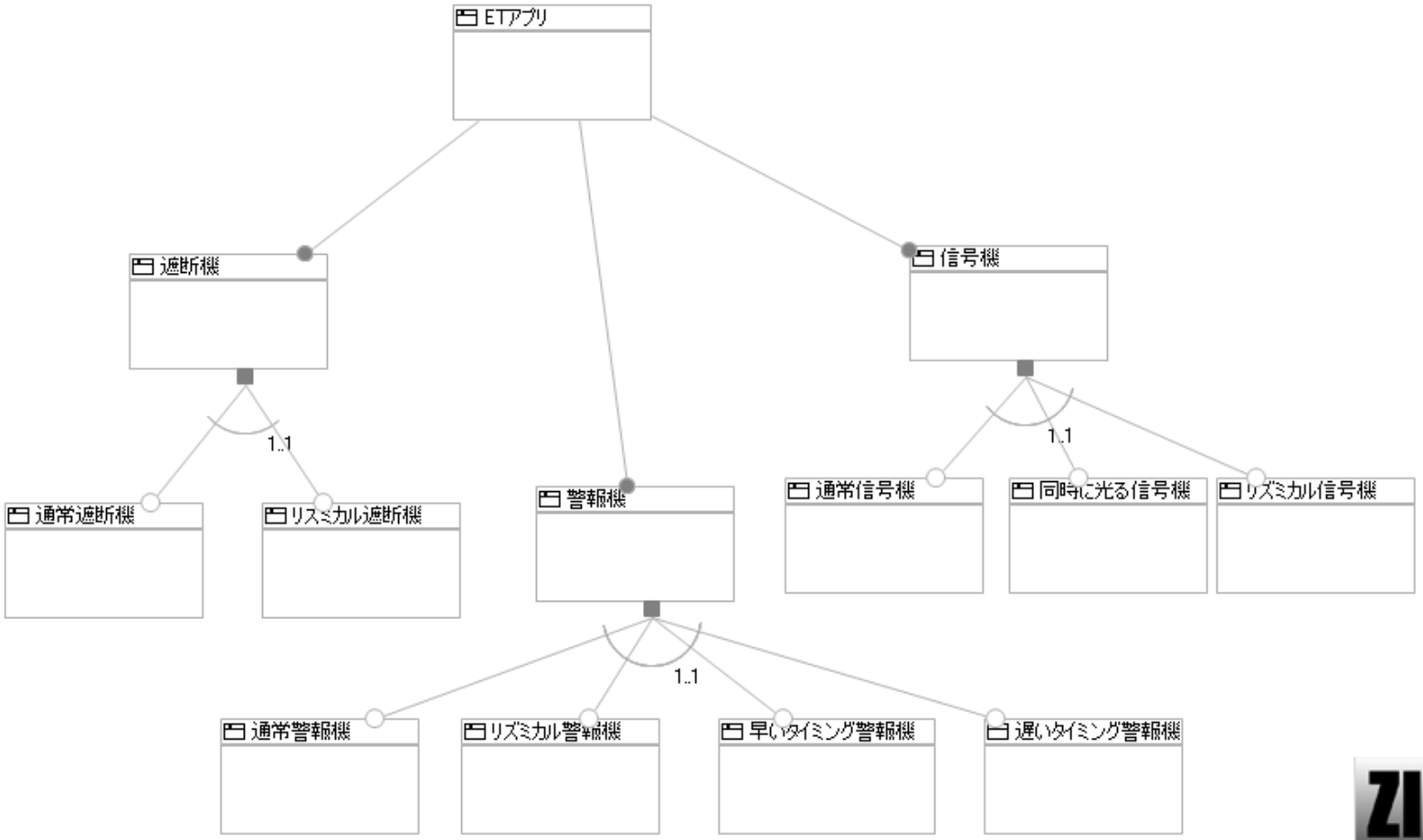


事例：踏み切りモデリング（仕様概要）

- 機能概要説明
  - 外部センサーの検知状況に対応して、警報機・信号機・遮断機を制御する
  - 警報機は、警告音を発生/停止する
  - 信号機は、2つのLEDを点灯/消灯する
    - 警告音とあわせた動作が必要
  - 遮断機は、遮断機を上げ/下げする。
- 以下のようなバリエーションを作成する
  - リズミカル遮断機
    - 337拍子で遮断機を上げ下げする
  - リズミカル警報機
    - 337拍子で警告音を発生する
  - タイミングの早い警報機
    - 通常より早いタイミングで警告音を発生する
  - タイミングの遅い警報機
    - 通常より遅いタイミングで警告音を発生する
  - 同時に光る信号機
    - 左右のLEDを同時に点灯/消灯する
  - リズミカルな信号機
    - 337拍子で信号機を点灯/消灯する

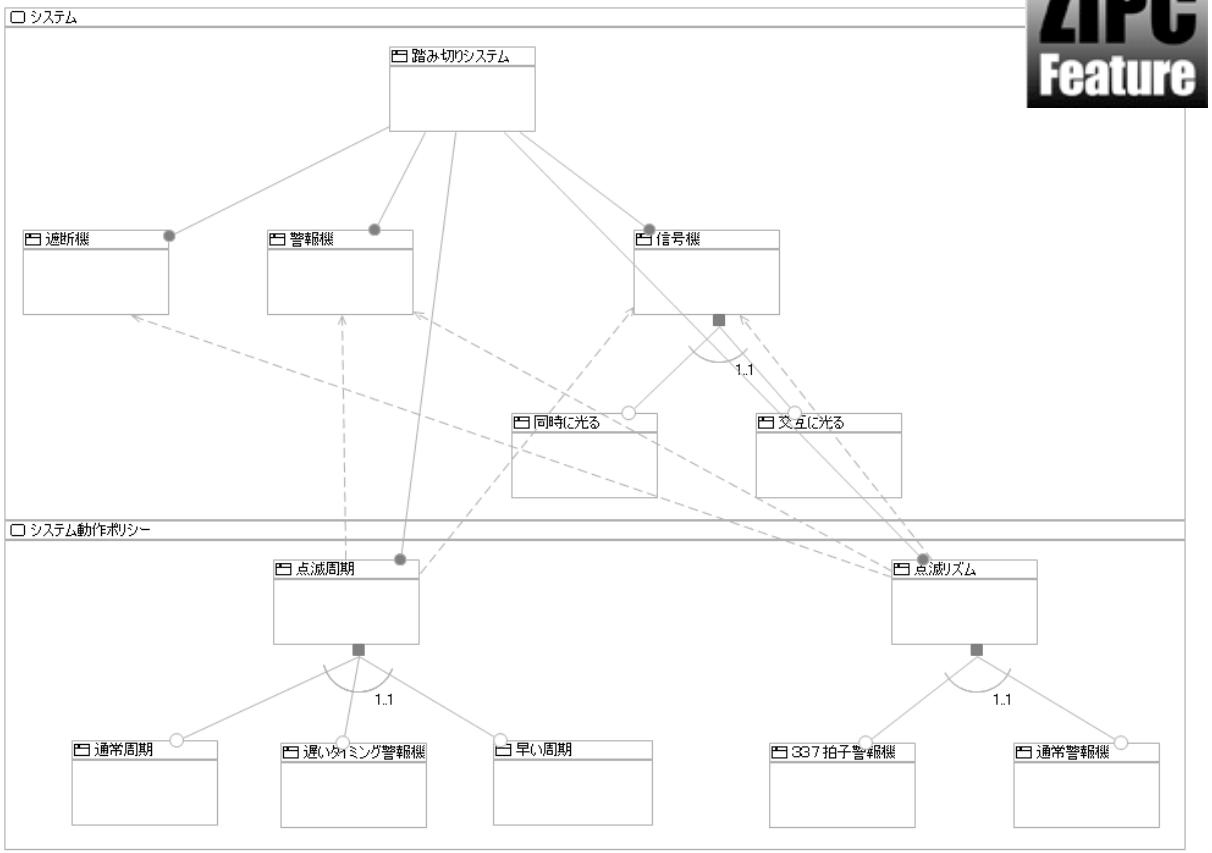
事例：踏み切りモデリング（フィーチャモデリング１）

- フィーチャモデル



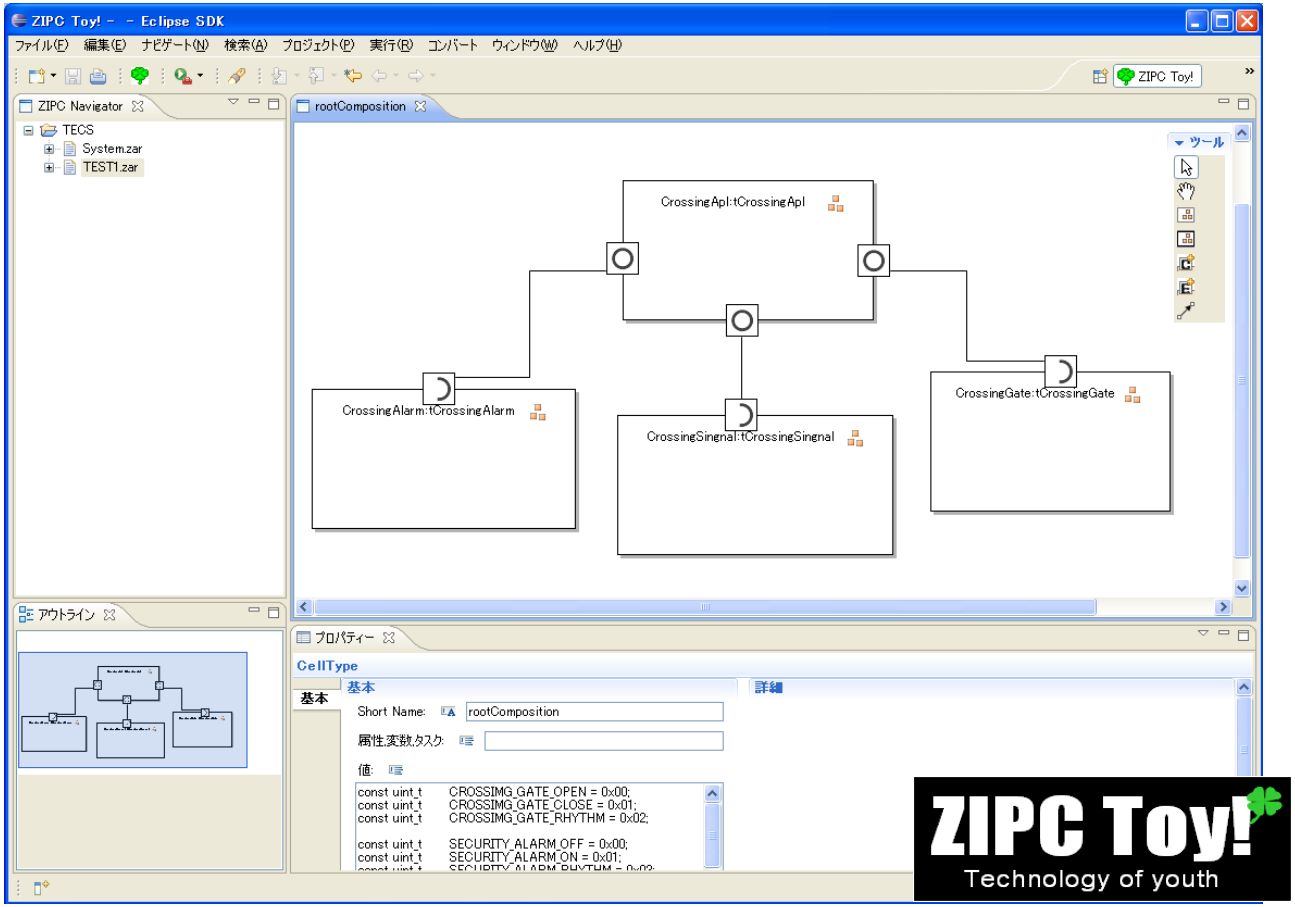
事例：踏み切りモデリング（フィーチャモデリング２）

- 改訂 フィーチャモデル
  - システムと
    システム動作ポリシーの分離
    - 個々の要素の動作ではなく、
      全体としての動作
  - 暗黙の仕様の明示化
    - 同時に光る信号機



事例：踏み切りモデリング（構造モデリング１）

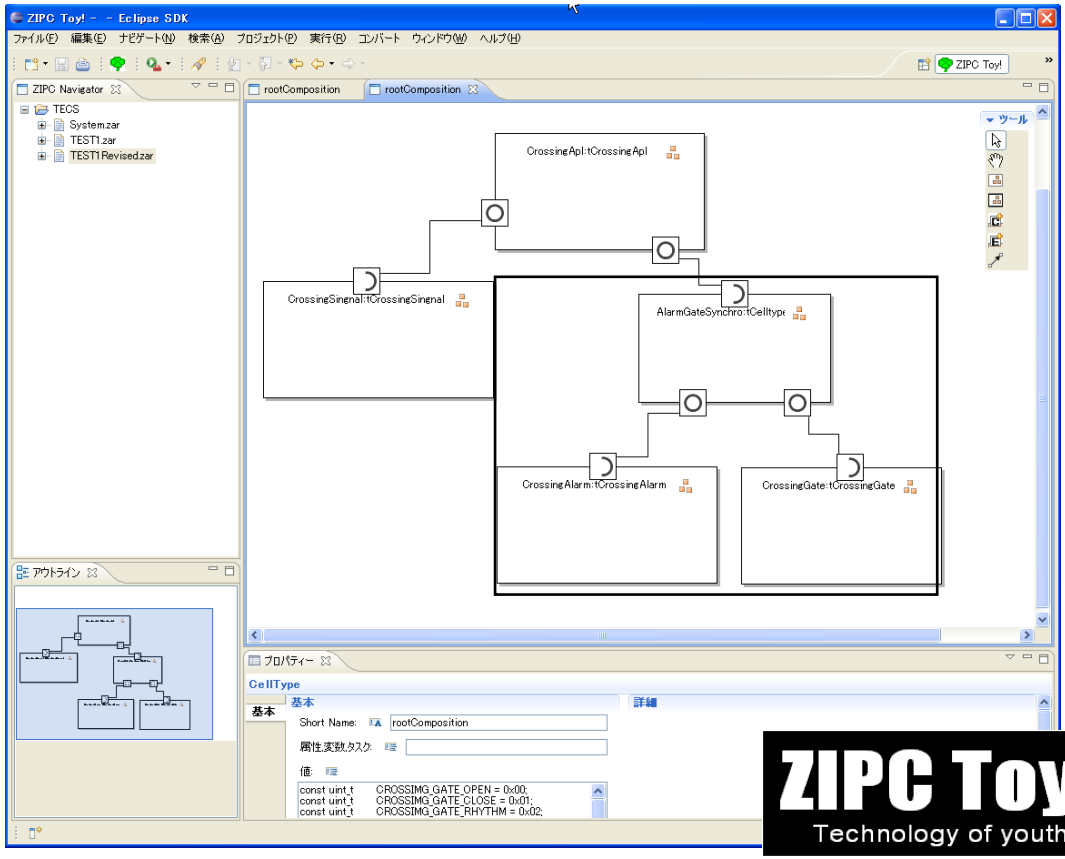
- 構造モデル
  - 信号機・警報機・遮断機
    - 全体のポリシーを受け取る
    - 個別に動作する



ZIPC Toy!
Technology of youth

事例：踏み切りモデリング（構造モデリング２）

- 改訂 構造モデル(未実装)
  - 同期させる要素の明示化
    - 警報機と遮断機の連動
    - 常にセットで使用する要素を考慮



ZIPC Toy!
Technology of youth

事例：踏み切りモデリング（出力されたCDL）

- 後工程
  - TECSgenでコード生成⇒機能実装⇒・・・
  - コンポーネント内部を実装

- ブース(F-30)で
  踏み切り動作中！

TEST1_conv.cdl - ワードパッド

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ヘルプ(H)

```
celltype tCrossingSingnal {
  entry sCrossingSignal eCrossingSignal;
};

celltype tCrossingGate {
  entry sCrossingGate eCrossingGate;
};

cell tCrossingAlarm CrossingAlarm;
cell tCrossingApl CrossingApl;
cell tCrossingSingnal CrossingSingnal;
cell tCrossingGate CrossingGate;

cell tCrossingGate CrossingGate {

};

cell tCrossingSingnal CrossingSingnal {

};

cell tCrossingAlarm CrossingAlarm {

};

cell tCrossingApl CrossingApl {
  cCrossingAlarm = CrossingAlarm.eCrossingAlarm;
  cCrossingSignal = CrossingSingnal.eCrossingSignal;
  cCrossingGate = CrossingGate.eCrossingGate;

  cCrossingAlarm = CrossingAlarm.eCrossingAlarm;
  cCrossingGate = CrossingGate.eCrossingGate;
  cCrossingSignal = CrossingSignal.eCrossingSignal;
  priority_sysini = 6;
  stack_size_sysini = 1024;
  priority_syster = 6;
  stack_size_syster = 1024;
  priority_ctlact = 7;
  stack_size_ctlact = 1024;
  priority_ctlter = 7;
  stack_size_ctlter = 1024;
  cycle = 10;
```

ヘルプを表示するには、F1 キーを押してください。 NUM