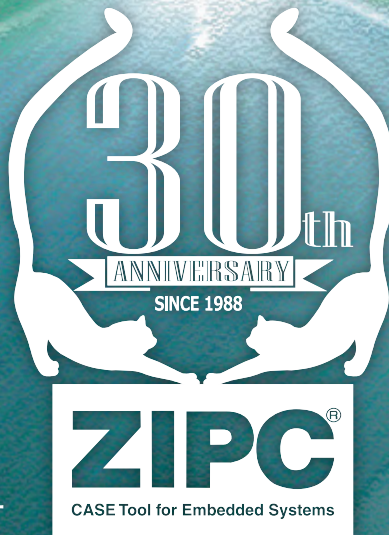


ZIPC®

W A T C H E R S

Vol. 21

2048



これまでの30年、これからの30年

■ZIPC誕生30周年記念インタビュー

- ZIPC誕生30周年に寄せて
- ZIPCとNECマイコン連携エピソード

■コラム

- ZIPC誕生30周年
～これまでの30年、これからの30年～

■特別寄稿

- MaaSがもたらす社会変革と、
その変革実現に向けた当社の取り組み
- 株式会社ネクスティ エレクトロニクス
ソフトウェア事業とこれからの取り組み

■適用事例

- 車載ソフトウェア開発への
ZIPC Tester 適用事例
- 新型HILSプラットフォーム (HELIOS) と
ZIPC Tester 連携

■学術研究コラム

- リバースエンジニアリングツール
RExSTM for Cの研究成果

■パートナー寄稿

- ソフトウェアトレーサビリティの
定着に関する提案と可視化ツール

■Tool ロードマップ

- 自動運転時代に向けた
ZIPC製品の今後の展開

■CATS SOFTWARE TOOLS

2018

1988



QINeS はお客様の車載システム開発の

生産性向上

をツールでも支援します



AUTOSARの
ビギナーでも
**BSWの
コンフィグを可能に**

Quick Configurator

QINeS-Tools

AUTOSARを
知らなくても
**MATLAB/Simulinkの
モデルをECUに自動実装**

Model Integrator

レビューや管理が
しやすいように
**トレーサビリティを
見える化**

Traceability Visualizer

ツールを展示会で体感してください!

**展示会
出展の
ご案内**

第11回 [国際] カーエレクトロニクス技術展
会期: 2019年1月16日 [水] ~ 18日 [金]
会場: 東京ビッグサイト 小間番号: E52-37

SCSK

URL : <http://www.scsk.jp/product/ecu>
E-mail : qines-info@scsk.jp

今と未来を、技術でつなぐ

Smart Life Community[®]



私たちが創造するのは、安全で快適な未来の暮らし。

コーポレートビジョンである「Smart Life Community[®]」のもと、
オートモーティブ／IoT／プロダクトフィールドを中心にサービスを展開。
モバイル端末等における組込みソフトウェア技術と
プラットフォームやクラウドを始めとするIoT技術の掛け合わせにより、
デジタル時代を牽引する新たなソリューションを提供します。



ZIPC WATCHERS Vol.21

Contents

目次

ZIPC 誕生 30 周年記念インタビュー

ZIPC 誕生 30 周年に寄せて p.8

独立行政法人 情報処理推進機構
社会基盤センター イノベーション推進部 調査役

田丸 喜一郎

ZIPC と NEC マイコン連携エピソード p.12

日本電気株式会社
組込みビジネス営業本部 エキスパート

奥村 晃子

コラム

ZIPC 誕生 30 周年 ～これまでの 30 年、これからの 30 年～ p.16

キャッツ株式会社 取締役副社長 最高技術責任者
CAL (CATS Advanced / Automotive / AI Laboratory) 所長
九州大学 スマートモビリティ研究開発センター 客員教授

博士 (工学) 渡辺 政彦

特別寄稿

MaaS がもたらす社会変革と、その変革実現に向けた当社の取り組み p.28

株式会社 NTT データ
製造 IT イノベーション本部 課長

千葉 祐

株式会社ネクスティ エレクトロニクス ソフトウェア事業とこれからの取り組み p.34

組込みソリューション本部 ソフトウェア SBU

SW ソリューション部 部長 松山 喜典

SW プロジェクト推進部 部長 赤星 博輝

適用事例

車載ソフトウェア開発への ZIPC Tester 適用事例 p.38

デンソーテクノ株式会社
DX 推進センターソフト生技部 部長

大原 孝一

新型 HILS プラットフォーム (HELIOS) と ZIPC Tester 連携 p.42

株式会社エー・アンド・デイ
XILS - PJ 技術主任

小川 篤志

学術研究コラム

リバースエンジニアリングツール RExSTM for C の研究成果 p.48

JASA
状態遷移設計研究 WG 主査

青木 奈央

パートナー寄稿

ソフトウェアトレーサビリティの定着に関する提案と可視化ツール ～取るだけでは終わらないトレーサビリティ～ p.52

SCSK 株式会社
車載システム事業本部 QINeS 製品企画部

橋沼 和子

Tool ロードマップ

自動運転時代に向けた ZIPC 製品の今後の展開 p.56

キャッツ株式会社
プロダクト事業本部 プロダクト部 部長

松枝 進介

CATS SOFTWARE TOOLS

CATS SOFTWARE TOOLS p.60

ZIPC FAMILY / 状態遷移モデリングツール / HMI モデリングツール / トレーサビリティツールプラットフォーム
パラメータ組合せテスト設計ツール / MBD テスト設計ツール / ツール導入支援サービス / ツール開発サービス

Interview

ZIPC 誕生 30 周年に寄せて

独立行政法人 情報処理推進機構
社会基盤センター イノベーション推進部

調査役 田丸 喜一郎



1. はじめに

CATS の拡張階層化状態遷移表設計支援ツール「ZIPC」をはじめ、その後、ZIPC ツールチェーンとして誕生した ZIPC Family シリーズ、組込みソフトウェア業界へのプロモートを一緒にしていく中で多くのアドバイスをくださった田丸喜一郎氏に、ZIPC と出会った当初からこれまでのエピソードを語っていただきました。

2. ZIPC との出会い

ZIPC が世の中に登場して 30 年ということで、本当におめでとうございます。また、販売されている組込みツールで 30 年間もさまざまな分野で使い続けられているものは他にないと思いますのですごいことです。

さて私が ZIPC を知ったのは 1990 年頃でした。当時 ASR という、アメリカのテクノロジーであるマイクロテックリサーチや、ウインドリバーシステム等の代理店販売をしていた会社があり、お客様やパートナー、代理店を集めた交流会を開催していました。そこに出席した時に「日本でもツールの開発販売をしている会社がありますよ」と紹介されたのが CATS でした。

その頃の ZIPC はまだ UNIX マシン版で、製品資料は、それを見ただけでは良くわからない A4 チラシ 1 枚のみ。そこで、当時の CATS の営業の方にお話を聞くと、「状態遷移表のツールです。もともとは、電力会社からの仕事をするために社内向けに開発しました。これがとても良くで

きているので、販売しはじめました」とのこと。

この製品は、どのようなことが解決できるのか、どのような効果があり、どのように良いツールなのかサッパリわかりません。ただ単に「状態遷移表のツールです」と言うばかり。

販売を始めた頃で、プレゼンも未熟な状態。

それが、ZIPC との最初の出会いでした。

3. 私が見てきた ZIPC の成長

その出会いから 2 年後くらいに当時 NEC にいらした M さんから、「ZIPC っていうツール知ってる？そのバージョン 3 ができて、CATS の渡辺さんが披露したいっていうから行かないか」とのお誘いを受け、CATS の渡辺さんの自信満々(笑)のプレゼンを聴きに行きました。しかしその段階でも、まだ電力会社向けのお仕事で、社内で使用するツールの域を出ていませんでした。当時は携帯電話や家電の組込み開発で iTRON OS が使われていたので、その OS への対応をはじめとした組込み開発向けに 10 個程度の改善提案を出しました。渡辺さんは、悔しかったらと思うのですが、そのような様子はおくびにも出さず、むしろ良いヒントをもらったという感じでその日はお開きとなりました。

その 1 年後、今度は、CATS の渡辺さんご自身から「いただいていた案をもとに改善をしてバージョン 4 が完成したから見に来て」と連絡をいただき、再度 ZIPC を見に行ったのです。その時は、前の年にコメントした事が全部反映されてい

て、説明も「使う人がこのように使うからこうなっています」とわかりやすくきちんとできるようになっていて驚きました。

1 年前は、「このような機能があるんです。すごいでしょ」と、開発者視点のプレゼンだったのですが、この時は、利用者視点でツール開発ができるようになってきたなという印象を受け、バージョン 4 では何も言う事がなく、製品もプレゼンもずいぶん頑張って成長したと感じました。

ですから、私の中ではバージョン 3 まではプロトタイプで、バージョン 4 から開発者がどのような困りごとを持っていて、どのように使えたら便利であるか、課題解決できるのかを意識して作られ、ようやく本格的な製品になったという印象です。

それまで UNIX に対応していた ZIPC は、バージョン 4 では、Windows にも対応するようになっていました。さまざまな方面から“ZIPC”という名前を聞くようになったのもこの時期でした。

当時は、ステートダイアグラム(状態遷移図)を使ってステートマシンを設計しろというのが教科書的に言われていた時代で、その手法が主流でした。それをマトリクスで表す状態遷移表にすることで、しらみつぶしにモレヌケをチェックできるという点が組込み開発に受け入れられたのだと



思います。その後、状態遷移表を軸にデバッガや上流モデリングツール等、さまざまなツールと連携して V 字プロセスをつないでいった進化の仕方も正解だったと思います。

4. 広がりを見せる ZIPC Family の 1 つ
トレーサビリティツールプラットフォーム
「ZIPC TERAS」

クラウドが流行りだしたころ、日本版クラウドの名前を考えていました。クラウド=雲だと何かぼんやりとしたイメージなのでスッキリとした明るい名前にしたい。日本ではプロジェクト名を神話から採ることが多いことからまず天照大神をイメージしました。そして陽のあたるところ、みんなが集まるところを照らすという意味で、「テラス」という言葉が思い浮かびました。「テラス」は、日本版クラウドの名前にはなりませんでしたが同時期に、設立時に CATS も携わった一般社団法人 TERAS のプロジェクトが始まりかけており、この「テラス」をプロジェクト名に採用してもらいました。後に製品化された CATS の ZIPC Family の 1 つ、ZIPC TERAS の基となったプロジェクト。この名付け親は、実は私です。



この一般社団法人 Teras は複数の会社がつながって、みんなで1つのツールを作り上げたという珍しい成功事例だと思っています。トレーサビリティはベーシックに必要な管理技術であり、機能安全やセキュリティ、自動運転の開発、検証にしてもトレーサビリティが管理できているかどうか、効率の良し悪しに影響します。

トレーサビリティ管理ができていない場合は、毎回設計ドキュメントをリニアサーチして関連するものを人力でピックアップしなければならず、非常に効率が悪くなります。

今まではドキュメントを作るときにトレーサビリティを確保する工数、コストの方が大きくなると考える人が多かったのですが、今やトレーサビリティを活用する目的、シーンがたくさん出てきました。

トレーサビリティ情報を開発の中で共有しておけば、変更時の影響範囲やボリュームが容易に判明することや、派生開発での差分管理等、さまざまな利用目的を効率的に実施できるという認識に変わりつつあります。ZIPC Teras もこれらのニーズに応えながら、拡張階層化状態遷移表設計

支援ツールの ZIPC のように長く使ってもらえるツールとして発展していくことを願っています。

5. これまでの ZIPC、これからの期待

なぜ ZIPC が 30 年もの長い年月、さまざまな現場で使われ続けているのか？

それは実際にツールを使う現場にいる開発者、ユーザ視点のバージョンアップを重ね、進化してきたこと、国立研究開発法人産業技術総合研究所の保有する技術を積極的に取り入れてきたことがあると思います。

また、CATS には、Teras のように業界内の企業を取りまとめ連携して1つのものを作り上げる力があるからだと思います。

今、CATS が他の技術との融合を研究開発中の理論知識型 AI を搭載した「ZIPC R&B」にも今後の成長に期待しています。

これからの 30 年も、これまでに培ってきたノウハウと開発者、ユーザ視点を大切に新しい技術を生み出し、開発現場で使える技術、ツールとして世の中に送り出して欲しいと願っています。

◇インタビュー◇

■渡辺 政彦 博士 (工学)

キャッツ株式会社 取締役副社長 最高技術責任者
CAL (CATS Advanced / Automotive / AI Laboratory) 所長
九州大学 スマートモビリティ研究開発センター 客員教授

■中島 美穂

キャッツ株式会社 プロダクト事業本部
営業企画・管理 G



田丸 喜一郎 氏

渡辺 政彦



第23回 ZIPCユーザーズカンファレンス懇親会では、乾杯のご発声をいただきました。
(2018年10月5日(金)新横浜国際ホテルマナーハウス南館にて)

Interview

ZIPC と NEC マイコン連携エピソード

日本電気株式会社
組込みビジネス営業本部

エキスパート 奥村 晃子



1. はじめに

1988年、UNIX版として誕生した、拡張階層化状態遷移表設計支援ツール ZIPC。

大規模制御系開発で使用されていた ZIPC は、1994年、NEC マイコンと連携し Windows 版として進化。それをきっかけとして、マイコンを搭載するあらゆる分野の組込みソフトウェア開発で活用されるようになっていきました。

CATS と NEC の連携に向けて一緒に取り組まれた奥村氏に、当時の様子を語っていただきました。

2. 当時の担当業務と

マイコンソフト開発の情勢

ZIPC と出会った頃、私はマイコンの開発環境を企画・開発する部門に所属していました。

当時 NEC には 4～32bit のマイコンがあり、マイコンを動作させるソフトウェアを開発するためのツールとして言語処理ツール、ハードウェアエミュレータとそのユーザインターフェース・ソフトウェアであるデバッガ、ソフトウェアシミュレータを提供していました。

制御用マイコンにおいてもそのソフトウェアをアセンブラでなく C 言語で開発できるようになってきており、NEC では更なる開発環境の進化に向け、上流工程を支援すべく CASE (Computer Aided Software Engineering) ツールをマイコンソフトウェア開発に導入する検討を開始しました。

その頃、CASE Fair 等 CASE に関する展示会もそれなりにあり、世の中に出ている CASE ツール

をいろいろと見ていました。CASE ツールと例えば当時は米国が進んでおり、いくつかのツールを検討しましたが、その対象は大規模ソフトウェアの複雑なロジック設計の支援が主眼であり、制御用マイコンのソフトウェアの開発にはご立派すぎてじっくりきませんでした。なぜなら、自動生成されるプログラムのサイズや、実行時に使用するワーク領域が大きくなってしまいうので、メモリ領域に制限があり、かつ TAT (Turn Around Time) が重要である組込みの制御用途には適さないからです。

エディタを使って、状態遷移図等の設計書類に全てのシナリオを書き起こすこと自体、大量複雑化するシナリオ表現を、どのように見やすく整理して書いていくかという本質から外れた課題を持つ中で、制御用マイコンの C 言語で拡張されている各種割り込み処理などを、CASE ツールから自動生成された C ソースに反映させることは、おおごとです。

3. 衝撃を受けた ZIPC との出会い

特に当時 NEC が世界 No.1 だった 8bit マイコンでは、まだ普及の緒に就いたばかりの C 言語ですら重いと敬遠される向きもある中、私としては重厚長大な CASE ツールを検討することに懐疑的な面もありました。

そのような中、上司から勧められて参加したのが、新横浜で 30 人位を集めて開催していた ZIPC のセミナー。ZIPC とは何であるかも知らな

い状態で、CATS (当時: テスコ) の渡辺さんの講義を聴きました。そこで私は、まさに天恵ともいべき衝撃を受けたのです。

ポイントは二つありました。一つは、設計手法が状態遷移表であること。状態遷移図は想定する遷移を図にしたものですが、状態遷移表は、各状態においてあらゆるイベントが生じたときにどう処理するか、いわば想定外も抜けなく検討できる手法です。制御用マイコンでバグの主原因となる想定漏れに、これは福音になるに違いありません。また、自由な平面上にバブルを配置していく状態遷移図と異なり、絵心のない日本人技術者に「表」という形式は思考にも操作にもなじむことでしょう。

二つ目のポイントは、渡辺政彦さんという人物です。それまで私が見てきた CASE ツールは、制御用マイコンのソフトウェア開発とは別世界の言葉で語られていました。しかし渡辺さんは、制御用ソフトウェア開発の現場の苦労を我がこととし、その解決策としての ZIPC を、熱く、熱く、語っていらっしやいました。制御用ソフトウェア開発を理解している人が、自ら企画開発しているツールであること、そこに情熱があること、これは NEC マイコン開発環境のパートナーとして必要不可欠、いや願ってもないことでした。

この二つの大きな衝撃を受けた、CATS の新横浜オフィスで聴いたその講義が、私にとっては、ZIPC と関わっていくすべての始まりでした。



当時のドキュメントを確認

4. ZIPC と NEC マイコン連携に向けて (プロローグ)

渡辺さんの講義を受けた後、職場に戻るなり、当時の上司に私が受けた衝撃を報告しました。制御用マイコン向け CASE には懐疑的だったけれども、ZIPC は全く違う、制御用マイコンのソフトウェア開発者が幸せになる CASE ツールだ、と。そして CATS 殿に NEC と連携してもらうにはどうすれば良いか、連携してもらいたい内容は、などの検討に着手しました。

ZIPC の良さを享受するには、多くの割り込みによる小さな処理を制御する小さなマイコンが良いと考えました。でも 4bit マイコンでは、ZIPC が自動生成する C 言語には向きません。そこで 78K0 シリーズという 8bit マイコンを対象とすることにしました。

制御用マイコンのソフトウェア開発ではコーディングよりデバッグが重要ですから、もちろんハードウェアエミュレータでのデバッグを状態遷移表上でできるところまでやりたいわけです。これには、78K0 のハードウェアエミュレータを操作する NEC のデバッガと ZIPC が連動せねばなりません。

78K0 の主な開発環境の OS は Windows でしたが、当時 ZIPC の Windows 版はエディタのみでした。私は上司や関係者を引き連れ、なんとか NEC と連携してほしい、Windows 版を開発してほしい、と CATS 殿にお願いにまいりました。すると CATS 殿からは、ぜひ Windows 版をやりたい、NEC に協力したい、との、嬉しいお返事をいただいたのです。

5. 連携取り組み中のエピソード

それから NEC マイコン 78K0 と ZIPC をつないでいく事となり、一緒に連携に向けてのトライアンドエラーが始まりました。

私は、どんな機能を実現したいか、という企画とアイデア出しに徹し、実現は CATS の技術陣にお任せしました。すると、私からは実現困難と思われるアイデアも、CATS のみなさまはすると実現してしまうのです。その技術力の高さ、

前進するパワーは輝いていました。

といっても、ハードウェアエミュレータとの連動は難しい課題でした。デバッグでは特定のイベント発生でブレークして状態を確認し、場合によってはメモリ内容を書き換えて、また実行を再開する、といった操作をします。

ZIPC の状態遷移表の記載が C 言語に落ち (スケルトン生成)、人間がその中身を埋めるのですが、その C ソースからアセンブラに落ちた機械語がマイコンで動作するわけですから、ブレークした箇所は、C ソースでは「このあたり」ということになります。

それを、ZIPC でのデバッグでは状態遷移表上で表示するわけですから、状態遷移表、C ソース、アセンブラという 3 段階それぞれでどう見せるか、そのブレークと再開をどう処理するべきか、等などを、夜が更けるのも忘れて討議して進めていきました。

CASE ツールというのは、机上のもの、雲の上のものというイメージを持つ人がとても多い中で、ターゲットボードでエミュレータをつないで、実際に動いているのを見せると、驚きと感動で、

途端にみんな目の輝きが変わりました。

そして、この実効性のある「ZIPC」という CASE ツールのお話をきちんと聴いてくれるようになりました。

ですから、難しく大変だったエミュレータ連携ですが、同時にそれが嬉しく楽しく良かったことでもあります。

何より、制御ソフトウェアの想定外イベントによるバグを設計時につぶせることで組み込み制御開発の世界に寄与できるという信念や、組み込みの世界はお客様もツールメーカーも半導体メーカーも Winwin の関係であり、お客様が成功するには開発環境がうまく動いていないといけないという、みんなが同じ想いを持って取り組んでいました。

開発するだけではなく、世の中に広める活動も一緒にしていきました。お客様に効果を見せる事ができるプロモーション用に、NEC の電話 CL5100 を ZIPC の状態遷移表エンジンを使って比較開発してもらい、その実効性を確認、情報処理学会に論文も出ささせていただきました。この CL5100 で、各処理を呼び出すという連携デモを展示会に出したりもしました (図 1 図 2)。

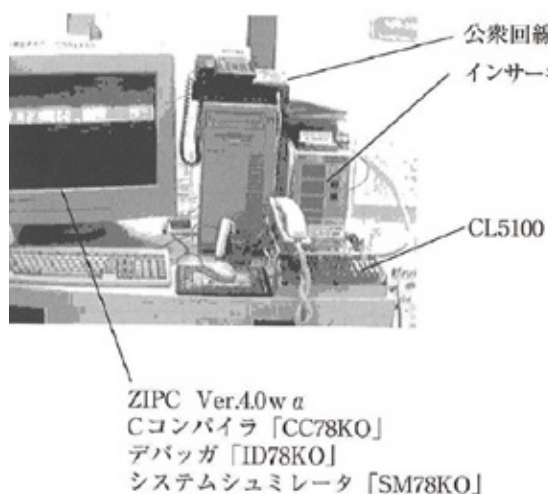


図1 CL5100とZIPC連携デモ



図2 展示会出展時の様子

徐々に ZIPC も有名になって他のマイコンメーカーからの引き合い連携のお話が入るようになりました。NEC にとっては競争相手ながら、ZIPC により組み込み開発の世界を良くしていくべく、それについては相談しながら進めていきました。

6. 最後に

次世代のマイコン開発の姿を作っていくという中で、その時点ではなかなか広まらなかったとしても、それは重要なことではありませんでした。なぜなら、プログラミング言語 C が世の中に浸透するのに 20 年かかり、当時ようやくマイコンの世界にも浸透し始めたという事もあり、もう少し経てば CASE ツールによる実機デバッグを含めた開発が当たり前になるだろうと考えていたからです。

それよりも目指すものを共有できる人が居るかどうか。居ない事には、進化する方向が合わないのでは何年待っていても何ものにもなれません。

そういう意味で、高い技術と熱意ある CATS のみなさまと同じものを目指して取り組んだことで、次世代マイコン環境と ZIPC の進化や成長を実現できたのだと思います。

ZIPC の世界はこれからどのように成長していくのでしょうか。CATS のみなさまのご活躍を楽しみにしております。

◇インタビュー◇

■渡辺 政彦 博士 (工学)

キャッツ株式会社

取締役副社長 最高技術責任者

CAL (CATS Advanced / Automotive / AI Laboratory) 所長

九州大学 スマートモビリティ研究開発センター 客員教授

■中島 美穂

キャッツ株式会社 プロダクト事業本部

営業企画・管理 G

◇協力◇

■戸塚 真一

キャッツ株式会社 プロダクト事業本部

プロダクト部 第 1 技術 G シニアエンジニア

■吉野 由紀夫

キャッツ株式会社 プロダクト事業本部

プロダクト部 第 1 技術 G チーフエンジニア

■河野 陽一

キャッツ株式会社 プロダクト事業本部

プロダクト部 営業統括 G 担当課長



連携に携わったメンバー

河野 陽一 戸塚 真一 吉野 由紀夫

奥村 晃子氏 渡辺 政彦

ZIPC 誕生 30 周年 ～これまでの 30 年、これからの 30 年～

キャッツ株式会社 取締役副社長 最高技術責任者
CAL (CATS Advanced / Automotive / AI Laboratory) 所長
九州大学 スマートモビリティ研究開発センター 客員教授
博士 (工学) 渡辺 政彦

■はじめに

1988年に作ったZIPC。今年で誕生30周年を迎えます。これも多くの方のご支援によるものと感謝しています。「30年前なんて生まれてないし」という若い世代のエンジニアも多くいることでしょう。年寄りの悪い癖に付き合ってもらって、少し昔話をさせてください。

■ 30 年前にタイムスリップ

ZIPCの誕生は、三十数年前に始まった、電力制御自動化システム開発に端を発しました。当時、東京電力の配電自動化システムを日本工営の下請けでソフトウェア開発していました。まだ、MS-DOS (知ってる?) や、当然 Windows もありません。CP/M (ググりましょう) がPC用のOSとして登場していました。PCもPC-88とかPC-98 (図1) などがNECから発売されて使っていました。大型コンピュータの世界では日立がIBMに産業スパイとして訴えられた時代でもあります。CPUはZ80とかIntel8086などをアセンブラで動かしていました。



PC-98

■電力システム

交換機システムがコンピュータ制御になったように、電力制御もリレー制御からコンピュータ制御に代りました。そしてコンピュータ制御に付き物のソフトウェア、そしてソフトウェアに付き物のバグが問題になりました。電力制御システムは社会インフラですから、バグで止まるような事が有っては困ります。そこで、状態遷移表でモレヌケのないモデル (当時はモデルとは呼ばず、設計書) を作成し、コンピュータ上で実行し、Cコードを生成するCASE (Computer Aided Software Engineering) ツールを社内用ツールとして開発しました。これがZIPCです。



出典:東京電力パワーグリッド

<http://www.tepco.co.jp/pg/cost-down/it.html>

■ワークステーション

ZIPC を作ってみようと思いついたのは、仕事で使いもしないが、どうしても触りたかったビットマップディスプレイを持った UNIX マシンのワークステーションの登場でした。HP9000。Starbase というグラフィックライブラリで GUI プログラミングをすることができました。そこで作ったのが、状態遷移表のエディタです。当時はまだ Excel もない時代でしたから、同じ表なら表計算ソフトを開発した方が儲かったのか、それとも状態遷移表というニッチな市場を開拓したから 30 年間も販売し続けられているのか、どちらなのでしょうね。



https://en.wikipedia.org/wiki/HP_9000

■制御システム

ワークステーション上で電力監視制御システムのソフトウェア開発のために生まれた ZIPC は、他の制御系システムにも適用されていきました。

自律分散化防災システム事例

	音響タスク	スイッチタスク	画面タスク
STM枚数	4	65	36
事象数	88	503	640
状態数	46	466	310
関数数	20	219	937
生成方式	標準型	処理抜粋型	処理抜粋型
サイズ(step)	4890	33531	72285
自動生成率	97.5%	74.9%	55.7%

約26%の費用削減

大量ZIPCの適用によるシステム開発事例～火報システムの開発を通じて～ : ZIPC WATCHERS Vol.2
大石・大量ZIPCを適用した防災システムの開発 松下電工(株) システム開発センター システム品質開発部 : ZIPC WATCHERS Vol.3

■電子立国 日本

あらゆる機器に半導体が組み込まれ、電子化が進みました。この時代、日本をけん引した製造業は半導体産業でした。

ランク	1981年	1986年	1989年
1	TI (米)	NEC (日)	NEC (日)
2	Motorola (米)	東芝(日)	東芝(日)
3	NEC (日)	日立製作所 (日)	日立製作所 (日)
4	Philips (欧)	Motorola (米)	Motorola (米)
5	日立製作所 (日)	TI (米)	TI (米)
6	東芝(日)	NSC (米)	富士通(日)
7	NSC (米)	富士通(日)	三菱電機 (日)
8	Intel (米)	Philips (欧)	Intel (米)
9	松下電子工業 (日)	松下電子工業 (日)	松下電子工業 (日)
10	FCI (米)	三菱電機 (日)	Philips (欧)

出典:日本半導体歴史館

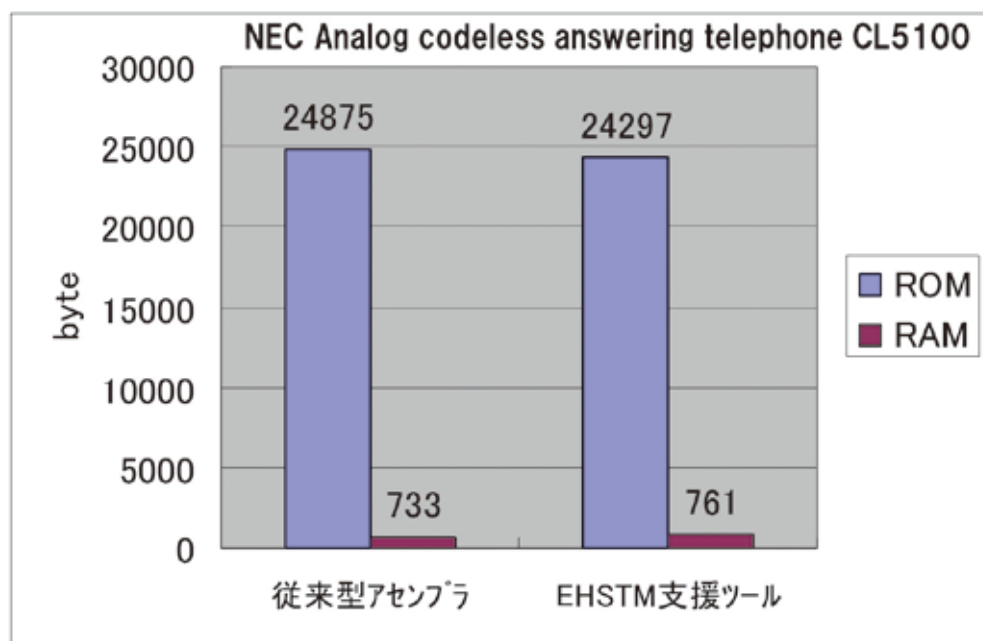
<http://www.shmj.or.jp/museum2010/exhibi065.htm>

■組み込みソフトウェア

マイコン制御プログラムと呼ばれていたソフトウェアは、いつしかアセンブラ言語からC言語となり、組み込みソフトウェアと呼ばれるようになりました。家電、OA 機器、デジタルカメラ、携帯電話、カーナビと半導体により電子化され、ソフトウェアを組み込んだ製品が世の中にあふれ出ました。DVDレコーダーは100万行、携帯電話は500万行となり、組み込みソフトウェアの開発の生産性向上が課題となりました。ここでZIPCがNEC、富士通などの開発環境と連携し、組み込みソフトウェア開発環境として大きく市場を拡大しました。

制御システムと組み込みソフトウェアでの大きな違いは、コンピュータリソースの違いでした。30年前、制御系システムではメガバイト単位のメモリ容量でしたが、組み込みシステムではキロバイト単位のメモリ容量でした。そこで取り組んだのがコードジェネレータの改良です。

高性能自動コード効率



沼田賢治・奥村典子, マイコンソフトウェア用統合CASE環境, p245-248, NEC技術Vol.50 No.3, 1997
 矢野, 渡辺, 奥村, 情報処理学会第53回(平成8年後期)全国大会, 状態遷移表を中心としたマイコン組み込みシステム用開発環境ZIPC

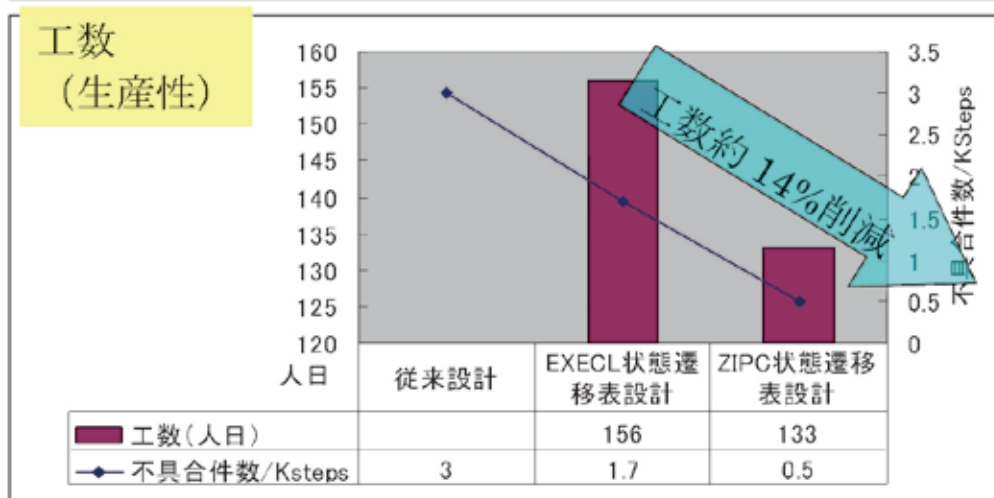
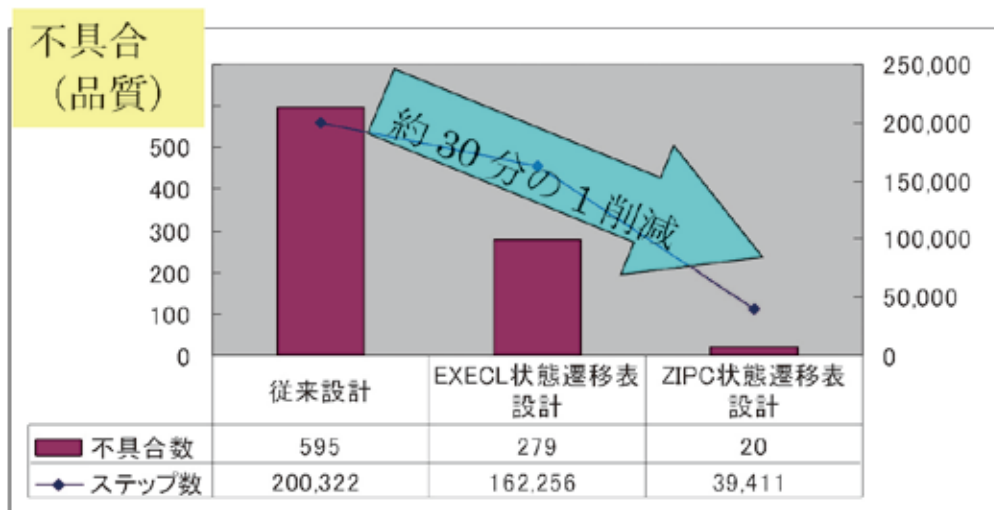
コード軽量化により車載への組み込みソフトウェアをはじめ、多くの機器に組み込まれるソフトウェアにZIPCは適用できるようになりました。

ZIPC適用事例

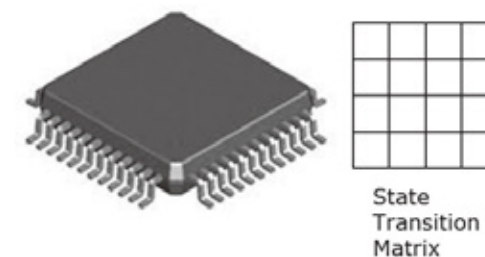
No	事例	効果
1	電力自動監視制御(テスコ)	・組合せ・統合テストの不具合件数 1/30削減 ・工数 17%削減
2	半導体処理装置自動化システム(キヤノン)	・生産性 4倍
3	複合機(コニカ)	・不具合発生件数 1/4削減
4	プリンタ(セイコーエプソン)	・不具合発生件数 1/10 削減
5	自律分散化防災システム(松下電工)	・約 26%費用削減
6	量産車載ソフト(某社)	・不具合件数 1/30削減 ・工数 14%削減
7	カーオーディオ(三菱電機コントロールソフトウェア)	・開発工数を 50%削減
8	カーステレオ(日本ビクター)	・不具合発生件数 1/5削減

詳細はwww.zipc.com

某車載ソフト（量産）の事例



1988



CASE

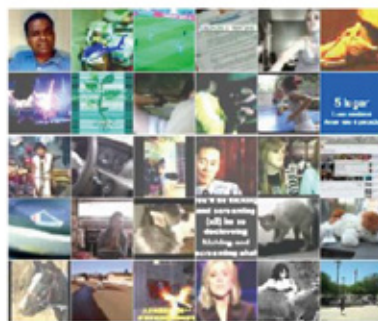
■これからの30年

さて、これからの30年を支えるキーテクノロジーの1つはAI（人工知能）だと考えています。AIは大きく2つに分けられます。1つはデータ型AI、そしてもう1つは知識型AIです。

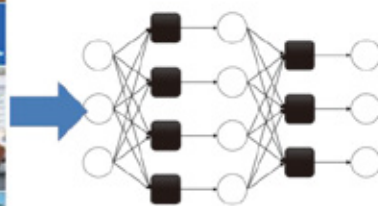
■データ型AI

囲碁でプロ棋士に勝ったり、猫を認識したりと世の中を驚かせたのが、データ型AIの機械学習や深層学習です。データ型AIは膨大なデータを処理できる時代になったので、その技術が実用に向けて開花しました。グーグル猫では、200 × 200画素の画像を1,000万個のデータセットを用意し、1,000台（16,000コア）のクラスターで3日間学習させました [1]。

200x200画素の画像を1,000万個



YouTubeトレーニング画像



従来67.2%の猫認識から
74.8%の猫認識へと
10%の相対的改善

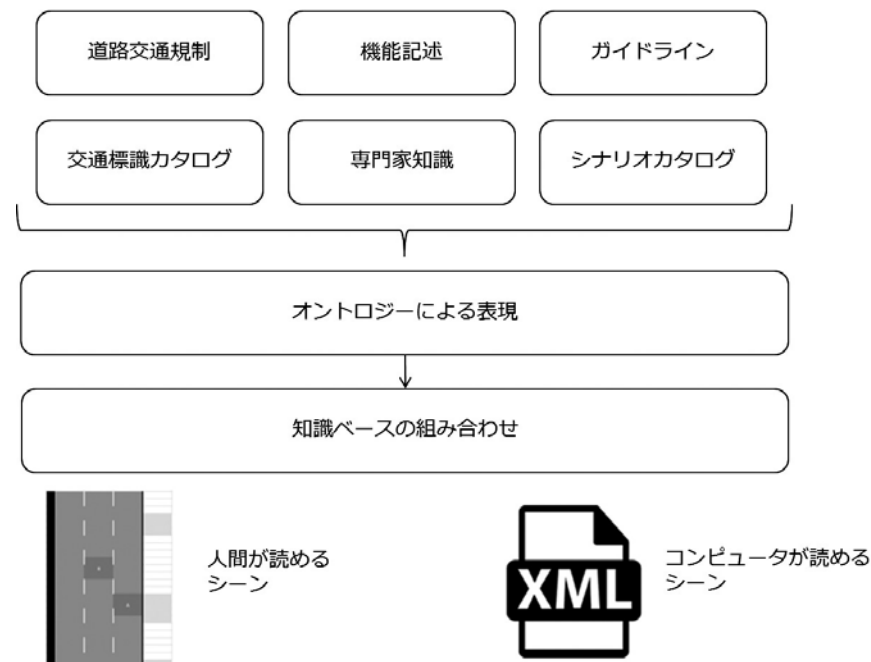
1,000台(16,000コア)の
クラスターで3日間



■ 知識型 AI

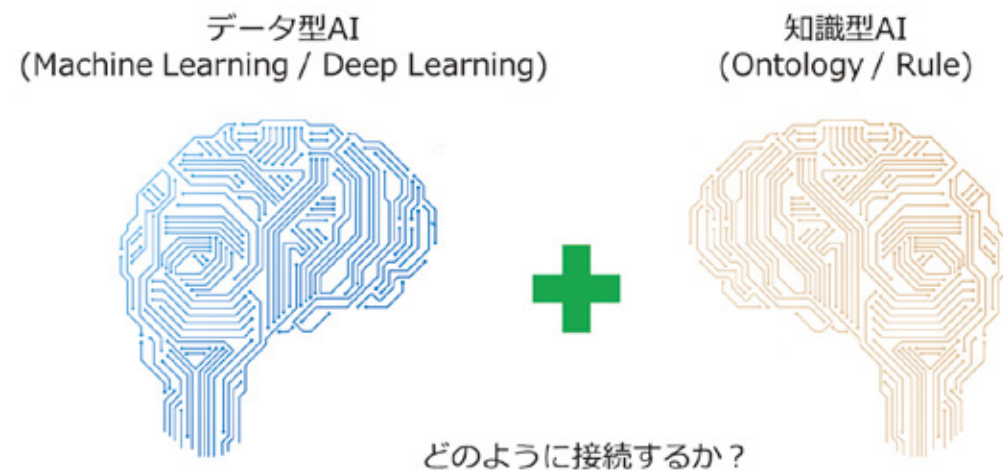
猫は1日12時間から16時間寝る、猫は爪を自由に出し入れできる、猫はツンデレだという知識をコンピュータが理解できるようにするのが、知識型AIです。データ型AIが画像から猫を認識したとして、猫というものは何であるかという知識は知識型AIの範疇です。この知識型AIをオントロジーやルールで構築することは、データ型AIのアノテーション作業より高度かつ忍耐の必要な知的作業のため、普及が遅れていました。しかしながら、最近ではドイツの自動運転プロジェクト”PEGASUS”[2]、[3]で、シナリオベースアプローチとしてオントロジーが採用されています。

Ontology-based process for scene creation



■ HybridAI

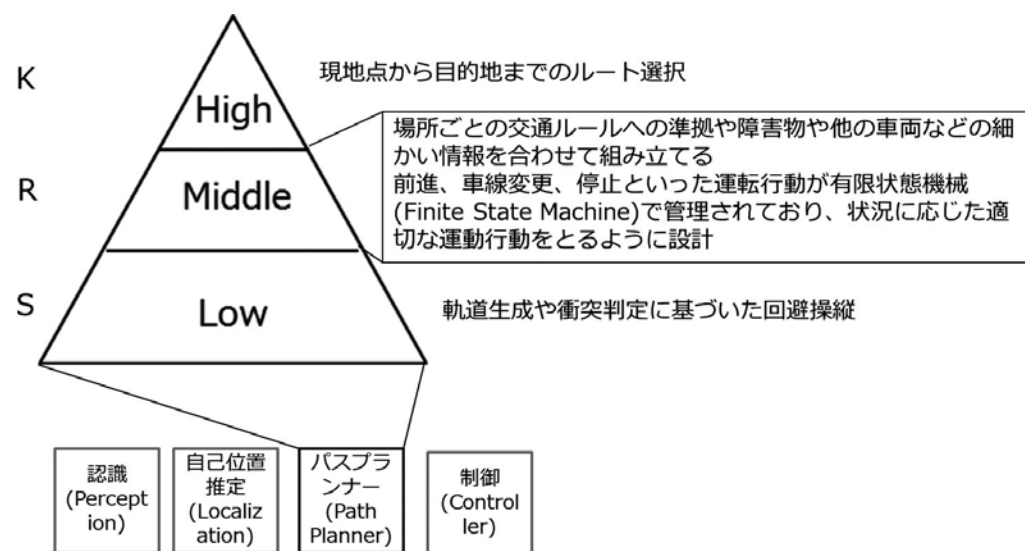
データ型AI、知識型AIどちらが良いかではなく、これらのAIをいかに融合させていくかが重要なテーマになります。



HybridAIについては、前年号(ZIPC WATCHERS Vol.20)「組込みシステムにハイブリッドなAIを」で詳しく書きましたので、そちらをご覧ください。

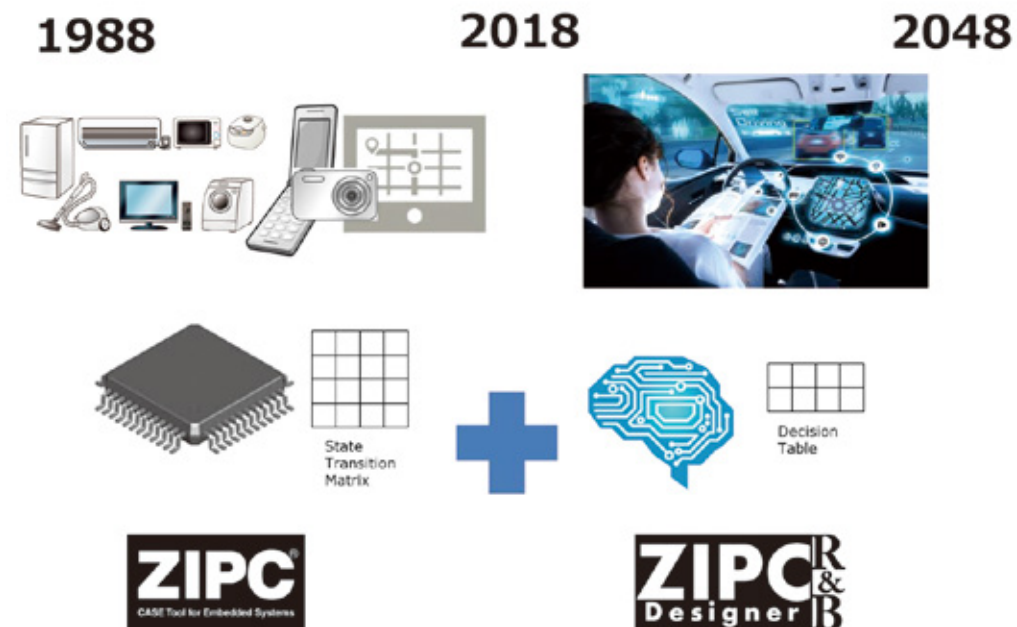
■自動運転は HybridAI

自動運転の AI は HybridAI です。認識や、ローレベルのパスプランナーはデータ型 AI で進め、ミドルレベルのパスプランナーは知識型 AI となります [4]。



■半導体、モデルベース、HybridAI

これからの 30 年、HybridAI を搭載したシステムをより効率的にする半導体やモデルベースの技術が進化し、その進化によって HybridAI がさらに成長する時代になると予想しています。



参考文献

- [1] “Building High-level Features Using Large Scale Unsupervised Learning”
Quoc V. Le ; Marc’Aurelio Ranzato ; Rajat Monga ; Matthieu Devin ; Kai Chen ; Greg S. Corrado ; Jeff Dean ; Andrew Y. Ng
Appearing in Proceedings of the 29 th International Conference on Machine Learning, Edinburgh, Scotland, UK, 2012.
<https://icml.cc/Conferences/2012/papers/73.pdf>
- [2] PEGASUS RESEARCH PROJECT
<https://www.pegasusprojekt.de/en/home>
- [3] Ontology based Scene Creation for the Development of Automated Vehicles
Gerrit Bagschik, Till Menzel and Markus Maurer Institute of Control Engineering Technische Universität at Braunschweig Braunschweig, Germany
- [4] 進化する自動運転と車載センサー；金沢大学 菅沼直樹；日経エレクトロニクス 2018.07
自動車の自動運転技術の動向と開発実例；金沢大学 菅沼直樹；電子情報通信学会誌 Vol.98 2015年1月

MaaS がもたらす社会変革と、 その変革実現に向けた当社の取り組み

株式会社 NTT データ
製造 IT イノベーション本部 課長

千葉 祐

1.MaaS とは

国土交通省によると、MaaS とは「ICT を活用して交通をクラウド化し、公共交通か否か、またその運営主体にかかわらず、マイカー以外のすべての交通手段によるモビリティ（移動）を1つのサービスとしてとらえ、シームレスにつなぐ新たな「移動」の概念である。利用者はスマートフォンのアプリを用いて、交通手段やルートを検索、利用し、運賃等の決済を行う例が多い。」と定義されています。

MaaS と一言で言っても、様々な定義があり、これもその1つと言えます。広義の意味ではかなり幅の広い抽象的な概念であると言えます。

具体的なサービスの1例として、Uber のビジネスモデルを説明すると、スマートフォンのアプリから、行きたい場所や、配車したい車のグレードなどを選ぶと、料金が決定、数分も経たないうちに車が手配され目的地まで移動、決済もスマートフォンで自動的に実行されます。世界的にはこういったライドシェアリングのサービスが当たり前になってきています。〈図-1.Uber のライドシェアリング参照〉

この MaaS を実際に実現するにあたり必要なソリューションを5つの視点で紹介させていただきます。

1つ目は「オンデマンド」、必要なタイミングでそれができるといことです。2つ目は「パーソナライズ」、より個人の嗜好に特化されているということです。3つ目は「マイクロ化」、共有するものや時間を細分化できるということです。4つ目は「交通手段の隠ぺい化」、ユーザが手段を意識する必要がないということです。そして、最後5つ目は「移動対象の拡張」、人やものだけでなく、コトや体験など移動の対象が増えるとい

うことです。

これらの視点や、これらのソリューションを実現するにあたっての課題や、必要な技術要素などをご紹介します。〈図-2.MaaS に必要な5つのソリューション参照〉

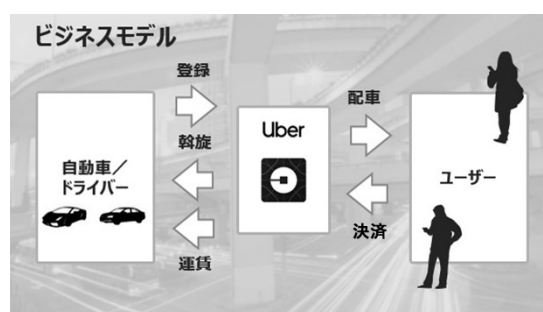


図-1.Uberのライドシェアリング



図-2.MaaSに必要な5つのソリューション

2.Mobility を取巻く社会のトレンド

2016年9月にダイムラー社が発表した中長期戦略である「CASE（ケース）」。4つのキーワードの頭文字からなるコンセプトで、電気自動車（EV）の「Electric」、自動運転の実現を目指す

「Autonomous」、それぞれ外部・相互接続性を高める「Connected」、そして、カーシェアリングなど多様なニーズに対応するための「Shared & Services」を意味しています。

MaaS を「Shared & Services」と捉えた時に、それを下支えするテクノロジーとしての、「Connected」、「Autonomous」、「EV」が実現した時、MaaS サービスは我々の生活そのものを変えていく可能性を秘めています。

その中でも、技術的なブレイクスルーとして最も期待、注目されているのは自動運転技術ではないでしょうか。

我々が移動に使用する交通手段の割合は、日本では通勤・通学、レジャーなど、どのケースにおいても自動車の割合が最も高い。これは日本だけではなく、特にアメリカは車社会ですのでその傾向は顕著と言われます。東京のように鉄道、バスを中心とした公共交通機関が整備されていると、感じにくいかもしれませんが、移動に占める自動車のインパクトは非常に大きいと言えます。〈図-3.Mobility の変革する「CASE」参照〉

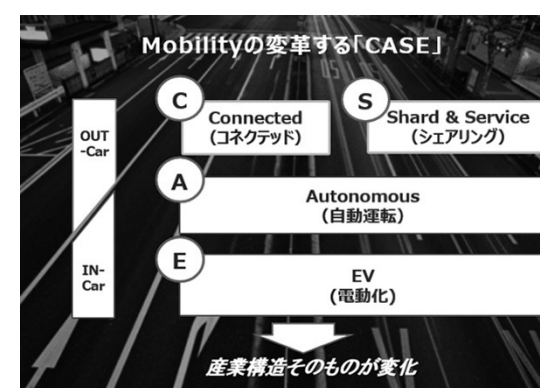


図-3.Mobilityの変革する「CASE」

現時点での自動運転技術の先行者といえば、Waymo だと答える人が多いでしょう。2016年 Google の自動運転部門が分社化した自動運転開発企業で、公開されている情報で、累計 900 万マイルを超える公道でのテスト走行、27 億マイルを超えるシミュレーション走行を実施し、2018 年以内に完全無人自動運転でのオンデマンド配車サービスを開始することを発表しています。

当初は限定したエリアでのサービス開始となりますが、そのエリアを順次拡大予定です。ただし Waymo のゴールは自動運転車の実現ではなく、その先にある自動運転車が実現した社会におけるモビリティサービスでの支配権獲得であり、それに向け動き始めたと言えます。

MaaS や、自動運転といったサービス・技術が自動車を含めた産業界に与える影響は計り知れず、各国が国策としてモビリティビジネスの拡大に取り組んでいます。日本も、トヨタ、日産、ホンダを中心とした自動車産業大国として、この新たな産業革命において国も巻き込んで取り組んでいると言えます。

国を巻き込むことで、例えば「マイクロ化」のためにもモビリティの移動率を上げること、つまり一般ドライバーの商業行為を許可することや、「交通手段の隠ぺい化」として、統合型のマルチモーダルなサービスを実現するために、鉄道やバスなどの公共交通事業者や、カーシェア、ライドウェアリング事業者が自社の事業領域の垣根を超えて連携していくこと、これを国が主導することで企業間の連携が促進されます。

各国の動きを見てみると、日本では、SIP（戦略的イノベーション創造プログラム）の1テーマで自動走行システムが採択されていますし、ドイツは、PEGASUS という名の自動運転関連プロジェクトが国主導で実行されています。また、今では一般乗用車販売の最大マーケットとなった中国で、百度の自動運転プロジェクトである「アポロ計画」が、国家開発プロジェクトに認証されています。このプロジェクトには米フォードや、独ダイムラーなどの OEM だけでなく、独コンチネンタル、独ポッシュのようなグローバル Tier1、米インテル、米エヌビディアなどのチップメーカーも参画し、中国マーケットにおける自動運転の中心的な取り組みとなっています。

3.MaaS がもたらすバリューチェーンの変革

世界市場における自動車販売台数の予測では、中国や新興国を中心に自動車需要は伸びますが、2020 年以降、シェアリングサービス等の台頭に

より、自動車販売台数全体で見ても飽和状態となり、むしろ個人への乗用車販売台数は減少すると予想されています。それに変わり、シェアリングサービスを前提とした車両の販売台数は急成長するでしょう。

この急成長の中で、バリューチェーンの変革をもたらすキープレイヤーの有力候補はWaymoです。Waymoの自動運転開発の今後の予測をする上で、2つ参考になるものがあります。1つは、PC業界のMicrosoftです。MicrosoftはOSの領域をWindowsで制覇し、それを搭載したPCを各ハードメーカーが作らざるを得ない状態にしました。

同様の戦略は、スマートフォンの世界でも起きました。スマートフォンのOSであるAndroidをGoogleが開発、それをオープン化することで各ハードメーカーがiOSの対抗としてAndroidを搭載したスマートフォンを開発しました。その結果、当時Appleに独占され兼ねなかったスマートフォンの市場で、今やiOSを大きく上回る市場シェアを獲得することができています。(図-4.自動運転のプラットフォーム参照)

Waymoが、自動車メーカーに自動運転システムを売り込み、自動運転システムの市場シェアを拡大していくことになれば、PCやスマートフォンと同様に、既存のOEMメーカーはハードウェアの製造会社となり、競争力を失います。かつ、顧客の接点や、アプリケーション領域の主導権を奪われ、急速にその支配力が低下していく可能性もあります。



図-4.自動運転のプラットフォーム

次に、完全自動運転の実現によるライドシェア事業者の利益の面を考察していきたいと思えます。広義の意味では、タクシーがライドシェアの先駆けということが出来ます。タクシーが日本で始めて創業したのが1912年、このビジネスモデルを破壊しようとしているのが、Uberを代表としたライドシェアリングです。加えて、それ以外にも、非営利型の相乗りマッチングサービスや、様々なモビリティのシェアリングサービスが普及してきています。さらに、今後レベル4以上の自動運転が実現することになれば、完全無人のロボタクシーが更にライドシェアリングの業界を破壊することになるでしょう。そして、それは、決して遠い未来のことではないのです。(図-5.ライドシェアリングの歩み参照)

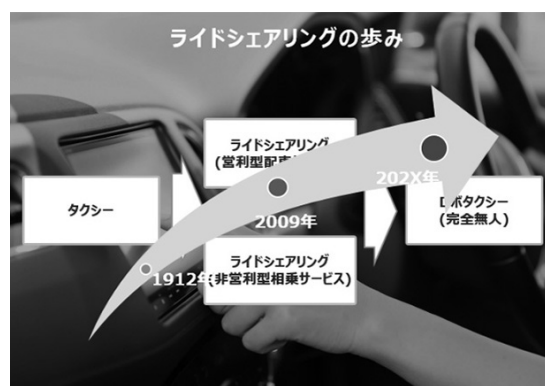


図-5.ライドシェアリングの歩み

4.MaaSの実現する世界

これらの技術革新によって、様々なMaaSサービスが成長し、生まれます。例えば、フィンランドのMaaSグローバルのWhimに代表されるマルチモーダル型の統合モビリティサービスが、対象となるモビリティを広げることで更にサービス拡張していくことが予想されます。公共交通や自動車はもちろんのこと、小型モビリティ、自転車、ローラーボードなどのラストワンマイルを想定した機動性の高いモビリティや、空飛ぶタクシーなども実現が期待されます。空飛ぶタクシーは、ドイツのポロコプター社が昨年度からドバイで実証実験を始めていますし、Uberも実証実験を開始

すると発表したばかりです。Uberの実証実験の候補地には東京も含まれています。更に、これらのサービスが、移動の目的地と結びついて、複合的なサービスを実現するような世界が来るはずで。 (図-6.MaaSソリューションの将来像参照)

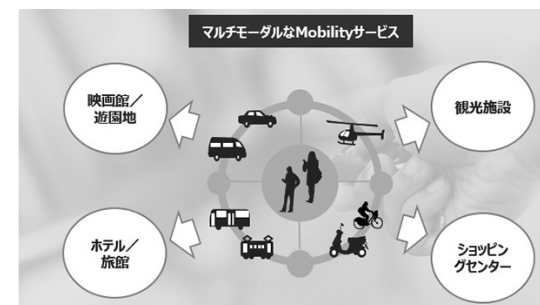


図-6.MaaSソリューションの将来像

それ以外にも、無人移動店舗の拡大が考えられます。「移動対象の拡張」は、その拡張の幅に限りがありません。既に検討が進んでいるものでは食料品やコンビニの無人移動店舗化があります。

その他にも、専門の店舗に行くことなく、オンデマンドでサービスを受けられれば便利なものは無数に存在し、それらが無人移動店舗で買えるようになる日も、遠くはありません。

5. 当社の取り組み

これまでご紹介してきたMaaSを取り巻く業界の変動の中で、当社も新たな取り組みを始めています。

1つ目は、トヨタとNTTグループでのコネクテッドカー基盤の実現に向けた協業活動です。当社は、大量データの処理基盤、分析基盤の開発で培ったノウハウを活用し、コネクテッドカーでの技術検証を行っています。車両とセンターがつながることで、将来的には何百万台という莫大な車両データから、利用者個人の嗜好や、集団の傾向を分析し、その結果がパーソナライズされたサービスや、これまでに見えなかったモビリティサービスの発掘につながる可能性があると考えています。(図-7.トヨタ自動車×NTTグループ協業参照)



図-7.トヨタ自動車×NTTグループ協業

また、MaaSにおける「オンデマンド」、「マイクロ化」の観点においては、2018年9月に、当社本社のある江東区豊洲エリアにて、3台の自動運転車を活用したオンデマンド移動サービスの実証実験を実施しました。豊洲のマンションにお住まいの方にご協力いただき、スマートフォンのアプリから好きなタイミングで配車し、目的地まで自動運転車で移動します。裏では、これらの自動運転車両が安全な運行が行われるよう、運行管理システム側から監視する仕組み作りに取り組んでいます。

6. 今後の課題

MaaSに必要な5つのソリューションの観点と、それらがより効果的に実現されるために必要となる自動運転技術という観点から、MaaSを取り巻く現在の競争環境と、今後の予測について、当社としての見解を示してきました。ただし、これらの5つのソリューションが更に利用者にとって利便性が高まり、なくてはならないものとなっていくためには、課題も多く存在します。

「オンデマンド」においては、複数の交通手段が繋がった場合に、どれか一つでもオンデマンドを実現していなければ、サービス全体で見るとオンデマンドを実現できないことが挙げられます。

「パーソナライズ」においては、個人情報の活用が細かく制限されていく中で、個人の権利を守りながら、よりパーソナライズされたサービスを実現しないといけないことが、「マイクロ化」においては、複数人の利用者でものやコトをシェア

するモデルにおいて、利用者間でのトラブルや、利用者同士に不利益にならないようにサービスが提供されなければならないことが挙げられます。「交通手段の隠ぺい化」においては、各事業者が自社の利益を優先する中で、事業者間で協調、連携をしなければならないことが、「移動対象の拡張」においては、無限大に考え得るビジネスにおいて、本当に必要でビジネスとして成立し得るものを見つけていくことが必要となります。

7. おわりに

日本企業が日本のマーケット、さらにはグローバルマーケットにおいて競争力を発揮していくためには、Google だけでなく、Apple や Amazon といった IT ジャイアントやプラットフォームと競争するための戦略を整えていくことが必要です。Apple の具体的な戦略については不透明ですが、カリフォルニア州で自動運転の公道テストは実施していて、その台数も現在では 70 台と今年 1 年で急拡大しています。Amazon も、自動運転事業に取り組んでおり、彼らの主力である物流事業の強化につながる有力な手段となることでしょうし、音声認識 AI である「Alexa」を車載システムに搭載することで様々なビジネスの拡大を検討しているはずで

このような IT ジャイアントと呼ばれるグローバル企業と戦わない、協業するという選択肢も考えないといけません。例えば、トヨタは、2018 年 1 月の CES で発表した「e-Palette Concept」のなかで、Amazon との協業を発表しています。これは、「移動対象の拡張」という視点で、トヨタが IT ジャイアントと協業する道を選択していると言えます。

このような動きの中で、これまでにあった自動車業界という垣根は、今後なくなっていくでしょう。モビリティという名のもと、サービスとしてこれまでのビジネスも再編されていくでしょう。我々日本企業も、既存のビジネスモデルにとらわれず、業界を跨いで新たな先行者優位となる競争領域を見つけ出すことが必要になってきます。そのためには、日本だけに視点を留めず、グローバルマ

ケットの変化を見逃さないようにしなければなりません。

もう 1 つ重要なことは、MaaS や自動運転という技術は、まだ法的にも、社会的にも受容されていない側面を持つということです。日本で一般ドライバーによるライドシェアリングが法的に認められるには、乗客の安全性が確保されるなど、日本の法規制を変えるほどの環境の変化が必要となります。自動運転のもともとの思想は、人間のドライバーによるヒューマンエラーが交通事故の割合の大多数を占めている中、より安心安全な交通を実現する手段だったはずで

す。自動運転が社会に受け入れられるためには、人間のドライバーより安全であることが定量的にも定性的にも証明され、一般の方にも理解されていく必要があります。日本の製造業は、その高品質、安全な製品を武器に海外のマーケットで評価を受け成長してきました。激変するモビリティ業界において、安心安全で、誰もが幸せになるためモビリティサービスの実現に向け、絶えず努力していくことが、日本企業が日本における MaaS サービスを拡大してい

参考文献

- [1] 国土交通省「土交通政策研究所報第 69 号 2018 年夏季」
- [2] 日経 BP 社「EV・自動運転を超えて“日本流”で勝つ」
- [3] 日経 BP 社「Q&A 形式でスッキリわかる完全理解 自動運転」
- [4] Waymo 公式 HP <https://waymo.com/>
- [5] 日経 XTECH「シリコンバレー Next レポート自動運転技術はダントツ、ウェイモの開発力を数字で読み解く」
- [6] 「自動運転技術と国際標準化について (ITS 標準化委員会 WG14)」(内閣府)
- [7] 日経 XTECH「2035 年の自動車業界、「利益の 4 割は新規事業から」

- [8] 日経新聞「ライドシェア陣取りに IT や車大手「自動運転」見据え先行投資」平成 29 年 9 月 23 日
- [9] 日経 BP 社「AI と自動運転車が街を見守る」
- [10] Whim 公式 HP <https://whimapp.com/>

株式会社ネクスティ エレクトロニクス ソフトウェア事業とこれからの取り組み

組込みソリューション本部 ソフトウェア SBU
SW ソリューション部 部長 松山 喜典
SW プロジェクト推進部 部長 赤星 博輝

1. はじめに

株式会社ネクスティ エレクトロニクスは、前身である株式会社豊通エレクトロニクス時代の2004年 JASPAR 設立を機に、2005年タイ、2008年には中国に、それぞれ車載ソフトウェア開発会社を設立して活動を広げ、2017年には海外オフショアに500人以上の組込みソフトウェア・エンジニアが活動出来るようになりました。しかし、近年の車載開発では、我々の想定を超えたエンジニアリソースと要素技術が必要とされており、それに対して新しい取り組みが必要になっていきます。

2. 車載制御ソフトウェア開発の課題

長年、日本ではC言語を中心とした車載制御ソフトウェアの開発が行われてきましたが、近年急激にSysMLやMATLAB/Simulinkなどのモデルを使ったシミュレーションベースの開発に移行しています。多くの開発現場では、モデルベース開発のメリットを享受するどころか、課題や問題などで苦労することが多いのが実情です。

人員のリソース不足とスキルセット不足が課題の一つです。現実的にプロジェクトに必要な人員を集めることが出来ないケースも増えていきますし、リソースが不足するためにスキルセット不足の人員を投入するという悪循環も発生しています。

二つ目の課題は、業務を依頼する側がリソースを調達する業務の負荷が上がり、本来の業務に集中することが出来ない状態です。本来は開発に集中したいところが、人集めに奔走したり、中・大規模の開発では複数社によるチーム編成をした場合に管理工数が増大するといったものです。

三つ目は、ソフトウェア企業は中小企業が多く、

資金繰りや計画的な人員計画が出来ないなどの課題があります。

3.MBDセンター

弊社では2017年からMBDセンターの前身となる活動を実施しており、2018年下期からは実運用を開始しています。MBDセンターでは弊社パートナーと連携し人的なリソースやノウハウなどの知見をまとめ、さらにプロジェクトマネジメント (PM) 機能も併せて顧客に提供します。(図1参照)

これ以降、MBDセンターでの実際の取り組み／活動についてご紹介をいたします。

4. リソース確保

近年の人的リソース不足に対して、MBDセンターでは2つの取り組みを行っています。1つは、顧客と中長期の計画をふまえて弊社パートナーとの業務の調整です。パートナーとしても短期間の仕事ではなく、人員の育成計画も考慮した受注が可能となります。2つ目は、弊社海外拠点(タイ、中国)のエンジニアによる、オフショアサービスの提供です。

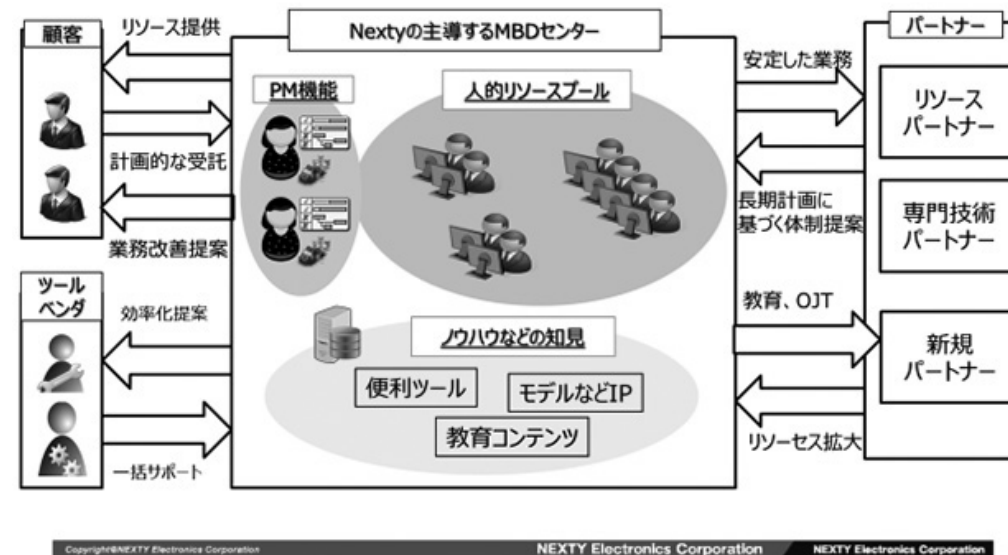
5.PM 機能の提供

近年のプロジェクトの大規模化とリソース不足により、複数社で開発メンバーをそろえると、複数社をまとめて管理する必要がありますが、顧客側でこのような管理業務を実施すると、本来やるべき開発業務に注力できないという問題が発生しています。そこでMBDセンターでは、複数社を跨いだプロジェクトマネジメント (PM) 機能を弊社が担う事で、こうした問題を解消しています。(図2参照)。

MBDセンターとその狙い



今後より拡大・複雑化していくモデルベース開発を支える1つの回答
→ MBDセンター



<図1 MBDセンターとその狙い>

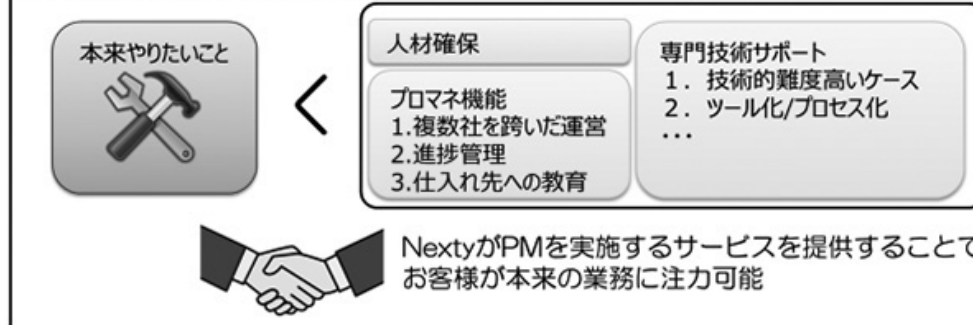
プロジェクトマネジメント(PM)機能の提供



● 1社で人材確保が難しく、複数社でのチーム構成が増加



この体制の課題：人材確保後も、本来のやりたい業務に集中できない。



<図2 PM機能の提供>

6. チーム化と教育支援

MBDセンターではPM機能を提供しています。これにより1社では受注出来ない業務を他社と連携して受注する事が可能となります。たとえば、車両知見が少ない会社でも他のパートナーと連携するようなケースです。

さらに弊社による技術サポートおよび教育のサービスも提供しています。特に、日本ではC言語のソフトウェア開発者が多いこともあり、そうしたメンバーにMATLAB/Simulinkの教育を実施し、実際のプロジェクトに参加して頂いています。また、教育コンテンツも実際のプロジェクトの経験から絶対必要な項目を中心に教育しています。シミュレータ構築などの高い技術力が必要な部分のサポートも実施しています。

7.QCD改善

ネクスティのPM機能では複数社を束ねた中でプロセスの標準化やツールなどの導入をすることで品質・コスト・納期(QCD)の改善もサービスとして実施しています。特にモデル開発ではプ

ロセスを定義し、ツール化することで生産性が50%上がった実績もあります。

8. ソフトウェア産業の課題

日本ではソフトウェア産業の多くは中小のソフトウェア会社が支えています。近年の開発契約では3か月単位という事も多く、資金繰りや人員を計画的に増員することが難しいという現実があります。MBDセンターによる中長期的な業務を得る事で、このような点が改善可能となります。

9. モデルやツール流通

MBDセンターではモデルやツールなども流通させますので、すべてのモデルを自前で作ることなく、シミュレーション環境(MILS)構築が可能となります。良いモデルやツールを流通させる事で、モデル開発を効率よく実施することが出来るようになります。また、販売力がないソフトウェア会社でも多くの顧客にアピールできるなどのメリットがあります。

10. おわりに

MBDセンターの取り組みにご興味がありましたら、以下までご連絡をお願いいたします。

株式会社ネクスティ エレクトロニクス

組込みソリューション本部 ソフトウェアSBU

■ SWソリューション部 部長 松山 喜典

e-mail : yoshinori_matsuyama@nexty-ele.com

■ SWプロジェクト推進部 部長 赤星 博輝

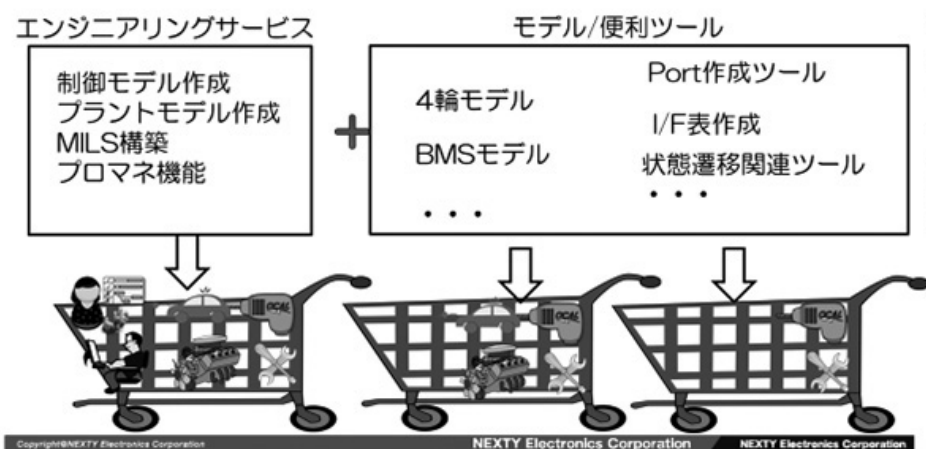
e-mail : hiroki_akaboshi@nexty-ele.com

モデルや便利ツールの流通



● 非競争領域のモデルやツールなどをMBDセンターで流通

- すべてのモデルを自前で作らなくても、MILS構築可能
- MILS構築自体の作成サービスも提供



車載ソフトウェア開発への ZIPC Tester 適用事例

デンソーテクノ株式会社
DX 推進センターソフト生技部 部長
大原 孝一

1. はじめに

デンソーテクノ株式会社は 1984 年設立以来、株式会社デンソーが担当する車載製品を主体として、ソフト・ハード・機器設計分野で仕様検討から量産対応までの開発・設計を担ってきた。

現在自動車業界は、CASE（コネクテッド、自動運転、シェアリング、電動化）のトレンドに代表されるように、100 年に一度のパラダイムシフトの時代を迎えている。

車の中では、走行制御と HMI 制御がより密に連携するようになり、各種コンポーネント・制御コンピュータの機能配置最適化、制御の高度化が進んでいる。またビッグデータ活用、インフラ・車々間連携を実現するために、車の外との連携も必要となっている。

つまり、従来の技術に加え自動車を利用したサービスが今後台頭してくる（図 1）。

車載システムの現状と動向



図1. 車載システムの動向

このサービス実現のために、携帯電話網を使用して常時車の中と外をつなぐ車載通信機が DCM (Data Communication Module) である（図 2）。トヨタ自動車は車載通信機 DCM をグローバルで共通化して、2020 年までに日米ほぼ全ての乗用車に標準搭載する予定である。



図2. DCM概要

DCM はサービス要求とつながるデバイスの増加に伴い、年々仕様が複雑化している。全てのシーンを網羅し仕様の抜け漏れを排除するために、弊社では DCM のソフトウェア設計に「状態遷移設計」を導入しプロセス改善の活動を推進してきた。これにより品質向上を達成するとともに、シナリオテスト支援ツール ZIPC Tester を導入することで効率化も達成できたのでその事例を紹介する。

2. 活動の背景

DCM は「非常時につながる」アラーム通報や緊急通報機能からスタートし、「必要なときにつながる」ナビ連携やリモートサービスが追加され、現在は「いつもつながる」ことでビッグデータ活用等スマートセンターとの密なやりとりを可能としている。これに伴い機能数は 2008 年以前の 5 から現在は 30 に増加し、仕様の全てのシーンを網羅して設計することが困難な状況となってきた。

この状況に対応すべく弊社では 2014 年から状態遷移設計を開発プロセスに導入して、プロセス改善を進めてきた。

要求仕様書の振舞いをイベントと状態で抽出して設計することにより、仕様書に現れないシーンを抽出することが可能となり、仕様の全てのシーンを網羅することが可能となった。

図 3 に状態遷移設計の例を示す。緊急通報サービスのシステムシーケンス仕様から対象である DCM に対してのイベントと状態を抽出して状態遷移表を作成している。

ここで状態遷移表を作成すると動作が仕様上定義できていない領域が存在した。このような動作不定領域を仕様作成元にフィードバックして動作を決定していくことが可能となる。たとえ通常では起こりえないパターンだとしても、もしその状態に陥った場合に車両動作が不定になることは絶対に避けなければならない。

この例のように仕様の抜け漏れを排除するうえで、状態遷移設計が品質の肝となると考える。

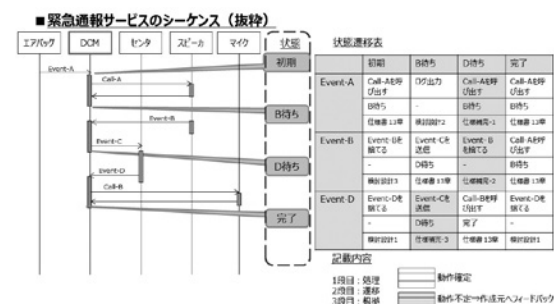


図3. 状態遷移設計例

3. 開発プロセス

DCM のソフトウェア開発プロセスは V 字開発プロセスで、弊社では仕様書をインプットとして構造設計からシステム検査までを担当している（図 4）。下流工程のソフト部品レベルの設計・検査については、製品横串でソフト工場と称して標準化を推進している。またシステム検査領域では、弊社で自動検査ベンチ（HILS）を構築し展開することで効率化効果を上げている。

今回の活動領域である、仕様書をインプットとしたソフトウェア構造設計領域では、まず Step1 として 2016 年にキャッツ様にコンサルをお願いし、状態遷移設計の厳密なルール化を実施した。

これにより品質は向上したが、検査工程の工数がネックとなっていたので、Step2 として状態遷移検査の自動化に昨年より取り組むこととした。

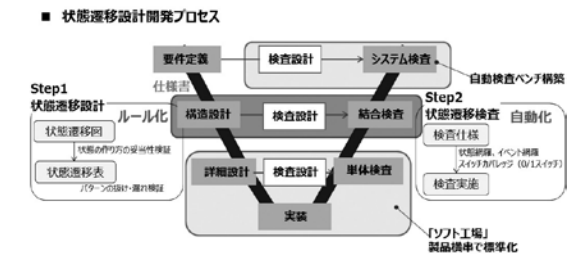


図4. 状態遷移設計開発プロセス

4. 検査自動化の取り組み

今回の取り組みは、状態遷移表をインプットとして、検査仕様書をアウトプットする工程を自動化することを目標とした。

従来の工程は、状態遷移表からイベントと状態の組み合わせで検査項目を作成し、検査データ・手順を実装し、期待値を決定する（図 5）。

このイベントと状態の組み合わせが DCM 機能の増加により増大しており、全体の検査工数を圧迫し、全ての検査が実施できない事態も危惧される状況にあった。

また検査パターンとして、A → B → A というような状態連続遷移（1 スイッチカバレッジ）を実施すると更に検査工数が膨大となるために、A → B、B → A の単一遷移（0 スイッチカバレッジ）のみしか検査できていない状況にあった（図 5）。これでは、状態遷移の組み合わせの検証網羅性が不十分である。

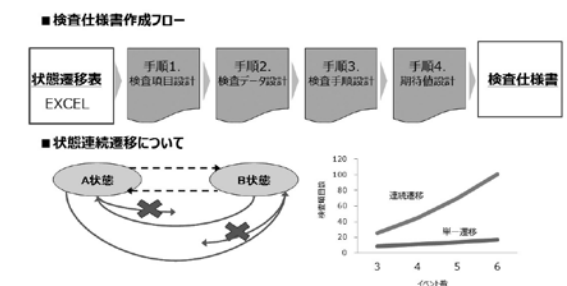


図5. 検査仕様書作成フロー
状態連続遷移

そこで検査仕様書作成の自動化を実現することで、工数を大幅に低減して状態連続遷移パターン検査の実現を目指した。

図5で示した従来の手作業プロセスの自動化を目指し、まず自動化ツールとしてキャッツ様から紹介いただいたZIPC Testerを試行した。

検査データ設計、検査手順設計、期待値設計に関しては想定通り適用できたが、肝心の検査項目設計が我々の要求と合わなかった。弊社では既にエクセルによる状態遷移表を設計で使用しており、今後も派生開発で使用していくために、これをZIPC Testerの形式に変換する必要があった。

キャッツ様と何度も打合せを実施させていただき、エクセルからZIPC形式にフォーマット変換するツールをカスタム対応していただけることで合意し開発していただくことができた。これにより弊社フォーマットの状態遷移表から検査仕様書作成の自動化が可能となった。

またZIPC Testerにはスイッチカバレッジ生成機能があり、1スイッチカバレッジをツール上で選択するだけで状態連続遷移パターンの生成が可能となる。2スイッチは更にもう1つ遷移が増え検証レベルは向上するが、現状では1スイッチカバレッジで実施すれば品質上十分であると判断している。

図6の例では、0スイッチカバレッジでは4つの検査パターンであったものが、1スイッチカバレッジでは8つの検査パターンに増加している。これは状態・イベントが増えると組み合わせで増大するので、手作業でのパターン網羅は非常に困難である。今回ZIPC Testerを使用することで、状態連続遷移の検査が可能となった。

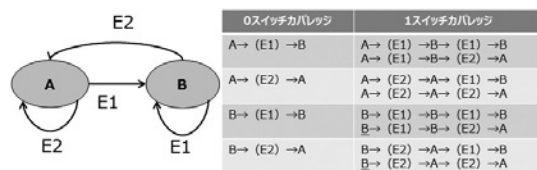


図6. スイッチカバレッジ

5.ZIPC Tester について

ここで、今回検査仕様書作成プロセスへZIPC Testerを適用した詳細について説明する(図7)。

ZIPC Testerには、シナリオ設計/状態遷移テスト/テスト管理/Simulink連携/State flow連携の5つの機能がある。今回の検査仕様書作成自動化には、状態遷移テスト機能を使用した。

今回状態モデル設計については、エクセルの状態遷移表を読み込むことで代行している。以降、検査項目設計→検査データ設計→検査仕様書生成については全てZIPC Testerで実現できる。ツール上ではスイッチカバレッジ数を設定し、イベント入力データを定義することで、エクセルの状態遷移表から自動で検査仕様書を作成することができる。



図7. ZIPC Tester適用

6. 活動成果

今回の状態遷移設計の活動について全体の成果についてまとめる(図8)。

Step1(2016年):状態遷移設計を導入し、状態遷移図と遷移表のフォーマット、作成・検証方法をルール化した。これによりV字プロセス左側の設計工程内不具合が70%近く削減できた。これにはこの活動の中で状態遷移設計の専門家を育成して、全レビューに参加させたことも大きく寄与している。

Step2(2017~2018年):ZIPC Testerを導入し、状態遷移表から検査仕様書作成までの工程を自動化した。これにより従来の単一遷移(0スイッチカバレッジ)でも検査工数が破綻しかけていたところが、状態連続遷移(1スイッチカバレッジ)でも検査工数を大きく削減することができ、今後の機能増大にも対応できる目途がついた。検査仕様書作成時間は75%削減でき、現在まで状態遷移関係の流出不具合0を継続中である。

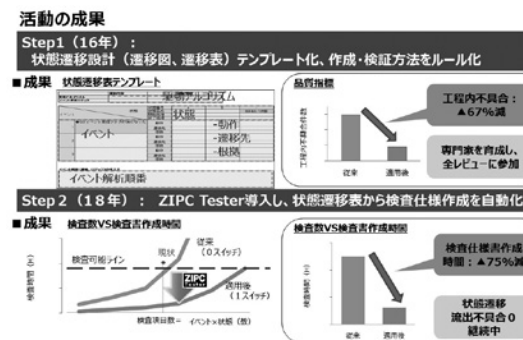


図8. 活動の成果

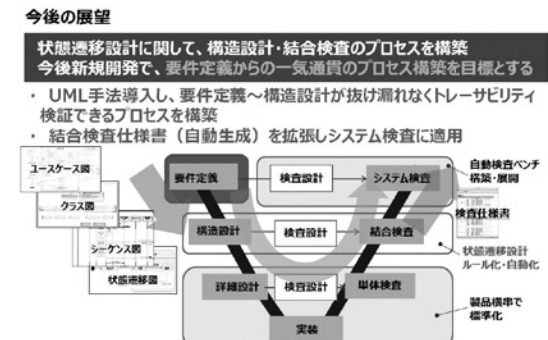


図9. 今後の展望

7. 今後の展望

最後に今後の展望についてプロセスの観点で触れておく(図9)。

今回状態遷移設計について構造設計から結合検査までのプロセスを構築することで、実装工程も含めた全体の状態遷移設計のプロセスが完成した。

今回は主に派生開発に対応したプロセスを構築したが、今後新規開発対応として要件定義にUML手法を取り入れて、上流から下流まで抜け漏れなくトレーサビリティ検証できる一貫通貫のプロセス構築を目標とする。つまり、ユースケース図、クラス図、シーケンス図と今回作成した状態遷移図のトレーサビリティをとることで実現させる。またこの構築には、今回適用したZIPC Testerをはじめとした各ツール間の連携も非常に重要になってくると考える。

もう一つ今回自動生成した結合検査領域での検査書をシステム検査領域へ拡張することを検討する。この拡張には設計プロセスでのトレーサビリティが必要と考えている。また車載システムは実機でのリアルタイム検証が求められるが、構造検査等バーチャル(SILS, MILS)検証が可能な部分もある。結合検査とシステム検査のすみ分けも効率化を進めるうえで重要と考える。

8. むすび

現在まさに自動車業界はパラダイムシフトの真っただ中である。今後、CASE(コネクテッド、自動運転、シェアリング、電動化)のトレンドが本格化するにつれて、状態遷移設計の考え方が益々重要となってくる。

今回は一製品での活動であったが、この考え方を多くの製品へ展開して、クルマ全体の開発効率化と品質向上に貢献していきたい。

最後に今回の活動に導入から深くかかわっていただき、ツール面からも多大な支援をいただきましたキャッツ様に深く感謝申し上げますとともに、今後の活動に際しても継続して支援いただきたいと思います。

新型 HILS プラットフォーム (HELIOS) と ZIPC Tester 連携

株式会社エー・アンド・デイ
XILS - PJ 技術主任

小川 篤志

1. はじめに

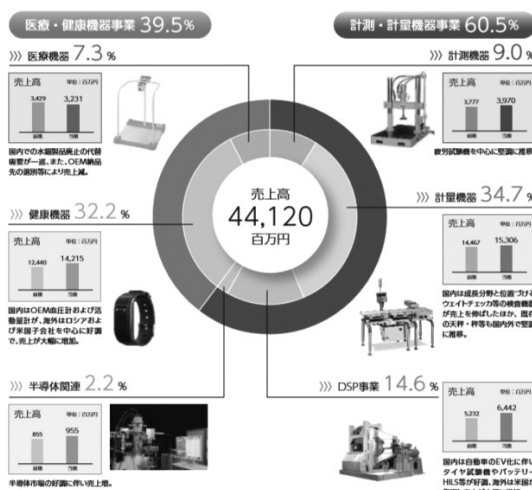
1977年の創業以来、エー・アンド・デイ（以下、A&D）は「はかる」ことを通じ、産業と社会の発展に貢献してまいりました。今日、私たちの計量器や計測器は、さまざまな産業の現場で、そして家庭で幅広くお使いいただいております。

A&Dの特徴は、開発重視の研究開発型企業であることです。研究、開発、設計、品質管理・生産管理・販売を行う一方、生産は国内外の関係会社に委託、両者の緊密な連携に基づいて事業展開しています。もう一つの特徴はグローバルな事業展開です。現在は、世界8カ国に開発・製造・販売拠点を設けています。

日本メーカーならではの迅速なサポートが強みで、国内外ともに展開しております。

A&Dは以下の事業領域としましては、医療機器、健康機器、半導体関連、計測機器、計量機器、DSP事業があります。

<図1 参照>



<図1 事業領域>

DSP事業では、計測・制御・シミュレーション用のプラットフォームを開発しております。今回、ご紹介するのはHILS用に専用設計した新型HILS「HELIOS (ヘリオス)」とそれを使ったXILS連携についてご説明します。

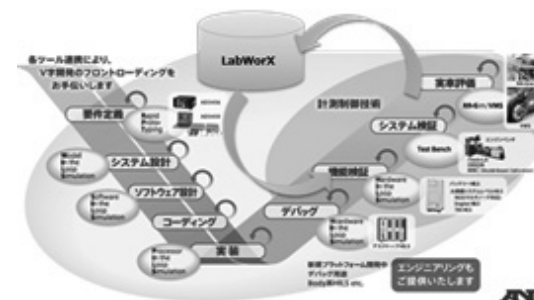
2. XILSの取り組み

XILS (エックスアイエルエス) とは、「X In The loop Simulation」の略で以下<図2>の総称。(他の名称もありますがA&Dでの定義を記載しています)

		定義
MILS	Model in the Loop Simulation	車両本体や、車両コントローラなどの全てを、MATLAB/Simulinkのモデルのみで実行するシミュレーション
RPT	Rapid Prototyping	汎用試作ECU
SILS	Software in the Loop Simulation	MILSに対して、車両コントローラをモデルではなく、C言語などのソフトウェアで実行するシミュレーション (RAM植込み)
	Software in the Loop Simulation	MILSに対して、車両コントローラをモデルではなく、C言語などのソフトウェアで実行するシミュレーション (フラッシュ植込み)
SPILS	Simulator based Processor In the Loop Simulation (ISS: Instruction Set Simulation)	車両コントローラに「マイコンシミュレータ」を使用して実行するシミュレーション
PILS	Processor in the Loop Simulation	実際の組込みマイコン (Processor, MCU) で動作させるシミュレーション
HILS	Hardware in the Loop Simulation	車両コントローラに、実際のECUホックス (ハードウェア) を使用するシミュレーション

<図2 XILS>

A&Dは、開発プロセスで言うと検証領域<図3参照>側のツールを提供しています。実車検証では、センサ (6分力計) による車体のデータ収集のためのVMSがあり、システム検証は台上試験による性能や耐久、VRSを行うベンチシステム、そして、ECU検証ではHILSを提供しています。



<図3 開発プロセス>

A&Dは、検証領域での更なる開発効率化を目指して、新型HILS「HELIOS」をリリースしました。HELIOSのシミュレーションソフトを、ベンチソフトのプラットフォームと統合することにより、検証資産の相互活用ができるようになりました。また、ベンチシステムで実績のあるデータ管理ツール「LabWorX」を使用することで、ベンチとHILSを一元管理することが可能になります。

「LabWorX」とは、Webベースでベンチを監視・管理することができるソフトウェアで、テストスケジュールの管理やデータ管理が可能です。

<図4 参照>



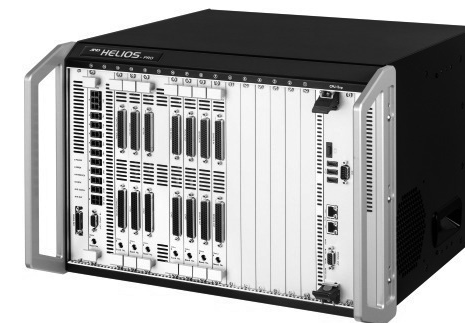
<図4 テストスケジュール管理>

開発プロセスの更なる効率化のためには設計領域との連携が必要であり、A&Dは設計領域で使用されるさまざまなツールと連携し、ソリューションを提供していこうと考えております。

モデルベース開発のテスト設計を強力に支援する「ZIPC Tester」と連携することにより更なる検証効率化が図れると考えております。

3. 「HELIOS」の紹介

今年2月にリリースした新型HILSプラットフォーム「HELIOS-PRO」<図5と6参照>のコンセプトは、「拡張性があり、大規模・精密モデルのシミュレーションが可能」というところにあります。



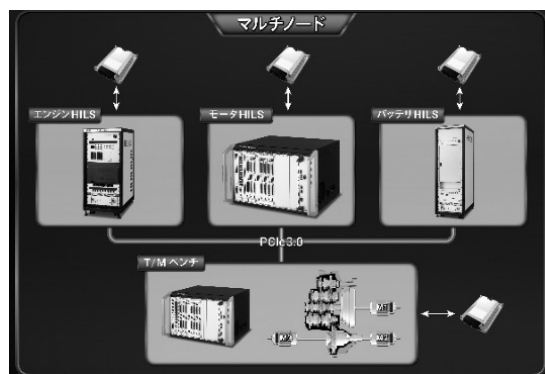
<図5 サブラックタイプ>



<図6 ラックタイプ>

「HELIOS」の大きな特徴として、

- ①マルチノード<図7参照>による大規模・詳細モデルが可能
- ②ソフトウェア統合によるベンチ連携
- ③電流計測や自己診断など HILS 専用設計されたハードウェアがあります。

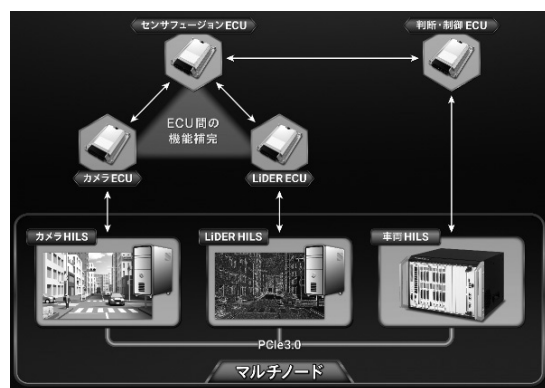


<図7 マルチノード統合イメージ>

A&D の提案するマルチノード HILS は高速のノード間通信 (PCIe3.0) と同期信号を有しており大規模や精度の高いシミュレーションをリアルタイムに実行できます。

自動運転開発で、環境やセンサー、車体などを再現させるには統合 HILS での検証が必要になってくると考えます。外部環境に応じたカメラや LiDAR 等の複数のセンサをリアルタイムに模擬

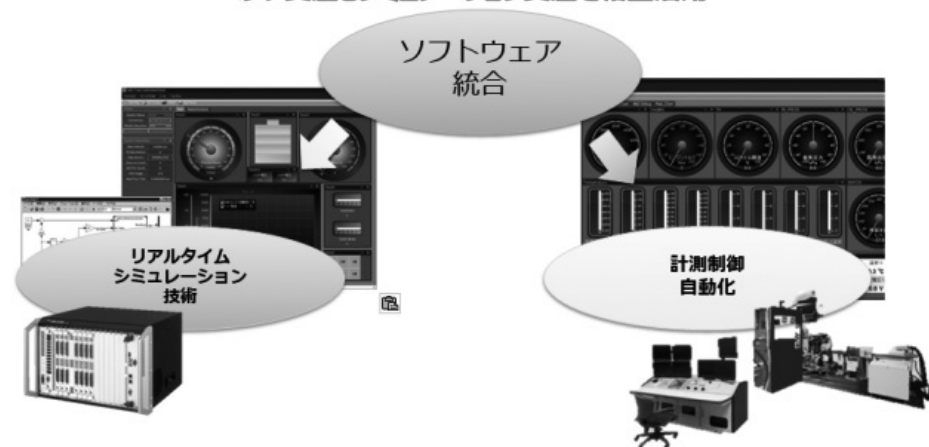
するためには、大規模なシミュレーションを高速で実現する必要があるため、広帯域の通信ができるマルチノード HILS が必要になると考えています。<図8参照>



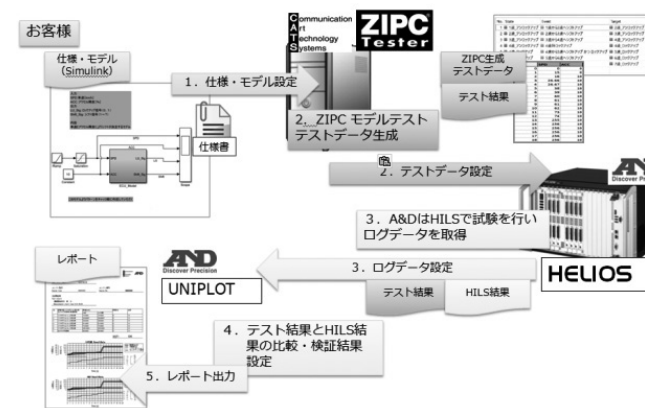
<図8 マルチノード自動運転イメージ>

ソフトウェアの統合とは、A&D のベンチと HILS で同じソフトウェアプラットフォームを使用することです。A&D のベンチシステムでは、以前より「iTest」という、ソフトウェアプラットフォームを使用しておりました。HILS システムにおいても「iTest」をベースにした「iTest-VSA」を使用することにより、ベンチと HILS の資産の共有を可能にしました。資産とは、プロジェクトにおける GUI、テストパターン、モデル、データ等を指します。<図9参照>

ベンチ資産とシミュレーション資産を相互活用



<図9 ソフトウェア統合イメージ>



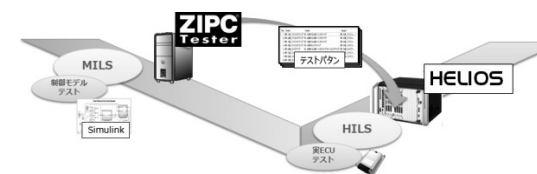
<図10 ZIPC Tester連携フロー>

4.ZIPC Tester 連携に関して

今回の取り組みは、A&D の HILS と ZIPC Tester を連携させました。ZIPC Tester 連携の目標としては、モデルと仕様書により ZIPC Tester が出力したテスト結果とテストパターン / テストデータを HILS に入力すると自動で HILS テストを実行します。その後、HILS でのテスト結果と ZIPC Tester のテスト結果を比較し、同等の試験結果であるかを判定します。最後にレポートを出力する流れになっています。

<図10参照>

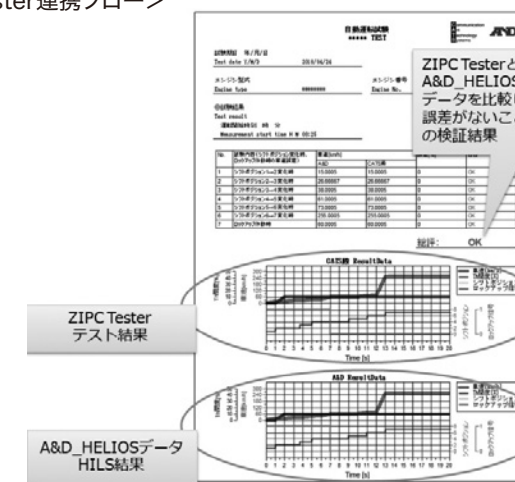
ZIPC Tester の検証資産を HILS にそのまま利用して検証することで MILS と HILS の連携を行いました。<図11参照>



<図11 MILS-HILS連携>

今回の成果として、一部手動での設定等がありますが、ZIPC Tester で生成されたテストパターン / テストデータを読み込み、自動で HILS テストを行い、ZIPC Tester のテスト結果と HILS 結果をレポートにまとめました。

<図12参照>



<図12 HILSテスト結果レポート>

今回の連携で以下に示す効率化が図れると考えます。

1. HILS で ECU チェックをする際のテスト生成を ZIPC Tester で生成したテストパターン / テストデータを使用することで HILS のテストパターン / テストデータを作成する工数の削減をします。

2. HILS で行った HILS のテスト結果と ZIPC Tester のテスト結果を比較し、自動で帳票化することで、帳票作成する工数の削減をします。

ZIPC Tester と HILS を連携することで MILS と HILS の B2B テストを行うことが可能と考えます。

5.ZIPC Tester 連携の課題

今回の取り組みのHELIOS側での課題として、手動で行っている「テストデータ設定」「レポート出力」等の設定部分の自動化が必要です。

更なる工数の削減のために

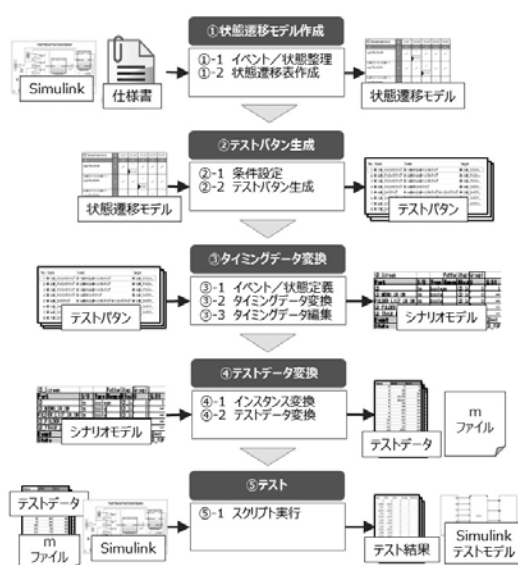
- HELIOS の出力信号とテストデータのマッピングを自動化
- HILS 内のモデル作成の自動化を検討しています。

キャッツ社に、ZIPC Tester 側の改善も検討していただいております。

連携時の ZIPC Tester のフローの項目 / 生成物は、<図 13 参照>

- ①状態モデル作成 / 状態モデル
- ②テストパターン生成 / テストパターン
- ③タイミングデータの変換 / シナリオモデル
- ④テストデータ変換 / テストデータ
- ⑤テスト / テスト結果

となっております。項目ごとに設定があり多くの工数が発生してしまいます。

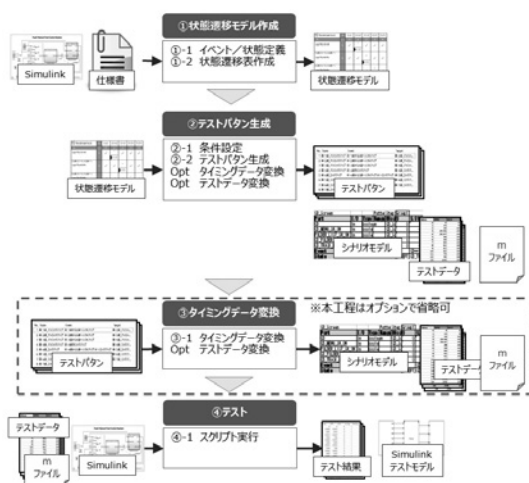


<図13 連携時のZIPC Testerフロー>

そこで下記の自動化を検討しています。改善案のフローの項目 / 生成物は、

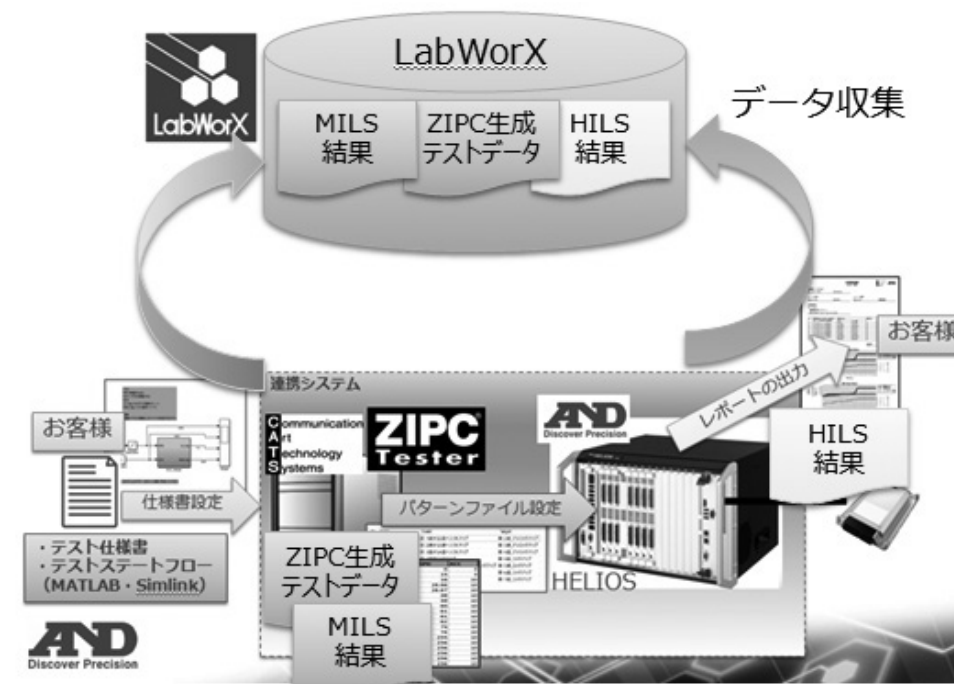
- ①状態モデル作成 / 状態モデル
- ②テストパターン生成 / テストパターン
(オプション：シナリオモデル・テストデータ)
- ③タイミングデータの変換 / シナリオモデル
- ④テスト / テスト結果

となります。



<図14 改善案フロー>

改善案<図 14 参照>では、オプションの機能として、項目②でテストパターン / テストデータを作成でき、工数の削減が可能となります。



<図15 LabWorXを使用したイメージ>

6. 連携の今後の展開

連携のさらなる自動化と LabWorX を使用して MILS、HILS でシームレスに繋げるようなソリューション<図 15 参照>の検討を行いたいと考えています。

7. 謝辞

今回の ZIPC Tester 連携を一緒に検討していただいたキャッツ株式会社の皆様に深く感謝を申し上げます。

リバースエンジニアリングツール RExSTM for C の研究成果

JASA
状態遷移設計研究 WG 主査
青木 奈央

ZIPC WATCHERS では、過去 2 年間にわたり、JASA の状態遷移設計研究 WG で議論してきた RExSTM for C について書いてきました。ここでは、過去 2 回分の ZIPC WATCHERS を振り返って、状態遷移設計研究 WG ではどのような研究がされてきたのか、今後どのような活動をするのかについて紹介していこうと思います。

まずは、JASA ですが、組込みシステム技術協会の略称で、状態遷移設計研究 WG は長年にわたりこの JASA の活動の一環として色々な課題について議論してきました。例えば、過去には設計手法普及調査を実施したり、状態遷移表の表記法を提示したりしてきました。そして、2013 年度からは、レガシーコードの活用方法に着目し、どのようにしたら再利用や、再利用不可の判断ができるかについて議論してきました。

組込みソフトウェアの開発の傾向として、派生開発による短納期・高品質の要望が強くなってきていて、レガシーコードの肥大化・複雑化やメンテナンス性の低下の声が WG 内で聞こえてきました。ところが、この問題を解決しようとしても、設計資料がなかったり担当者がすでに退職している等で解決方法がなかったり、更には、新しい開発の効率化のための手法を適応したくても、レガシーコードが問題となって、適応できなかつたりと大変でした。

そこで、WG 内では、レガシーコードについて議論を行い、「状態遷移設計には普遍的なモデルである」、「レガシーコードにも状態遷移は必ずある。例えばフラグのあるところには状態がある」というような仮説を立てて、研究テーマを「状態遷移表のリバースモデリングへの適用」としました。(図 1)

- ・ 組込みソフトウェア開発の傾向
 - ・ 派生開発による短納期・高品質の要望
 - ・ レガシーコードの肥大化・複雑化→メンテナンス性低下
→ 設計資料なし、担当者もすでにいない、修正したら予想外の問題が出る...

ソースコードのブラックボックス化が進行中！

- ・ 効率化のための手法導入が進まない
 - ・ 派生開発・レガシーコードのせいで従来のものを踏襲せざるを得ない...
- ・ リバースエンジニアリング
 - ・ レガシーコードを解析することで仕様を明らかにする

レガシーコードの存在が足かせになっている！

- ・ 仮説
 - ・ 状態遷移設計は普遍的なモデル
 - ・ レガシーコードにも状態遷移は必ずあるはず！

フラグのあるところに、状態がある！

- ・ 研究テーマ
 - ・ 「状態遷移表のリバースモデリングへの適用」
 - ・ レガシーコードから状態遷移表をリバースモデリングする

状態遷移表でレガシーコードを蘇生！

図1 レガシーコードについて

議論を重ねていくうちにリバースモデリングの作業手順を確立し、名古屋大学の協力を得て、ツール化することに成功しました。C のソースコードは非常に汎用性が高く、どのソースコードでも状態遷移表にするというのはとても困難なので、今回は、対応可能な範囲を決め、ツールを開発しました。(図 2)

リバースモデリングの作業手順

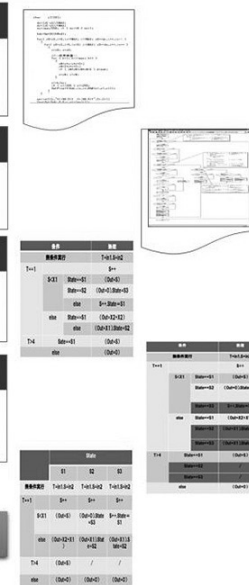
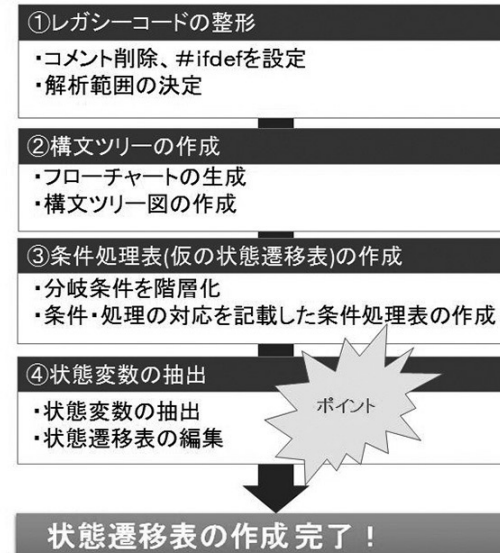
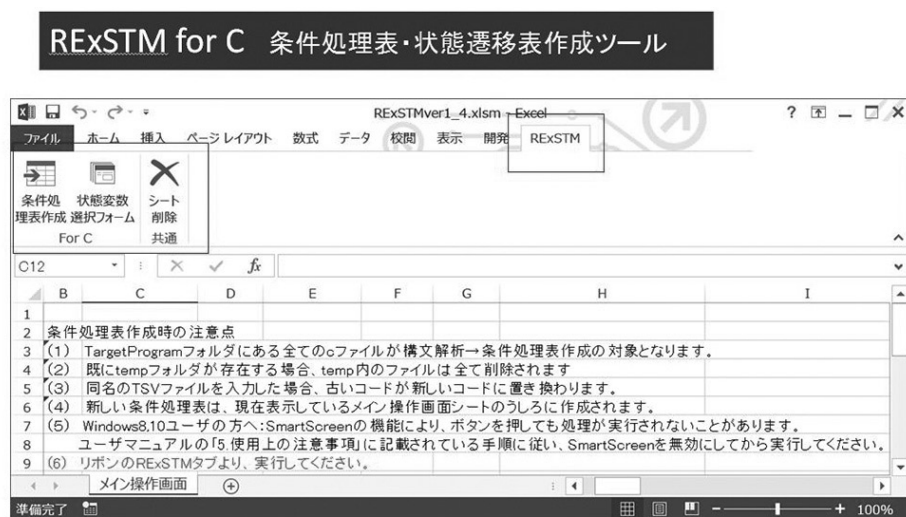


図2 リバースモデリングの作業手順

今回リリースをしたツールRExSTM for Cは、特に状態変数の候補抽出をすることに着目しました。残念ながら、全ての工程を自動化することは難しいので、「候補を抽出して、手動で選択をしてもらい、状態遷移表にする。」という工程を踏みました。(図3)



ソースコードの整形・解析を行い、状態遷移表生成のための情報整理を実施

図3 状態変数の候補抽出

このRExSTM for Cを実際に名古屋大学の学生さんに使用してもらい、手作業との比較実験も行いました。その結果、手動でレガシーコードから状態遷移表にする場合とツールを利用する場合は、時間を1/6も短縮することができました。(図4)

実験: 状態遷移表のリバース 手動 VS 自動

名古屋大学の学生4名によるソースコードから状態遷移表を手動vs支援ツールで作成する実験を実施した。
 実験対象ソースコード: 自動車の動作の一部を表すプログラムLOC110 (プラットフォームや開発環境向けに使用することが可能なコード)
 電子レンジの動作の一部を表すプログラムLOC190 (RTOS(TOPPERS/ASP)のアプリケーションを意識したコード)

被験者	修正回数	合計秒数	被験者	修正回数	合計秒数
学生1	0回	19分	学生1	1回	91分
学生2	0回	12分	学生2	2回	108分
学生3	0回	10分	学生3	4回	85分
学生4	0回	20分	学生4	4回	122分

支援ツールを利用する場合

手動で行う場合

- 手動の場合は、間違いや見落とし等も発生しやすく修正回数が発生し、ツールを利用する場合と比べて合計秒数も5倍から6倍程度かかる
- ソースコードと状態遷移表の対応がとれていない
- 関数が不足している
- 表現するべきではない箇所についても状態遷移表を作成している

図4 状態遷移表のリバース 手動 VS 自動の比較

現在は、JASAのWEBページからJASA会員限定で公開をさせてもらっています。利用期間は30日程度にしてもらい、使い勝手や、コメントなどをもらい、来年度は、このコメントをベースに更に改良を重ね最終的にはオープンソース化し、JASA会員だけではなく、幅広く多くの方々に使ってもらえるようなツールを目指して研究をしていきたいと思っています。

次号のZIPC WATCHERSでは、ツールを使っていた方々からの意見やコメントをベースに、どのようにしてオープンソース化をしていくかについて書かせていただこうと思っていますので、次号のZIPC WATCHERSを楽しみにしていただけましたら幸いです。

青木奈央

1998年 American University, Washington D.C. (BS in Mathematics) を卒業。
 国内外の企業数社を経て2001年からキャッツ株式会社へ入社、2004年からIPA/SECの立ち上げに従事し、2011年から北陸先端科学技術大学院大学にてソフトウェア検証センター研究員を務める。現在、同大学院理論計算機センター研究員、キャッツ株式会社プロダクト事業部、JASA状態遷移設計研究会主査。C言語から状態遷移表へのリバースエンジニアリングの研究に従事。

JASA 状態遷移設計 WG

<http://www.jasa.or.jp/TOP/activity/technology/state/>

参考文献

- [1] ZIPC WATCHERS Vol.19 「コードから STM ヘリバース RExSTM for C ツールについて」
- [2] ZIPC WATCHERS Vol.20 「コードから STM ヘリバース RExSTM for C ツールのこれまでとこれから」

ソフトウェアトレーサビリティの定着に関する提案と可視化ツール

～取るだけでは終わらないトレーサビリティ～

SCSK 株式会社
車載システム事業本部 QINeS 製品企画部

橋沼 和子

1. はじめに

車載システム開発において、機能安全規格への対応や品質向上などの目的で、すでに多くの人々がトレーサビリティに取り組んでいる。ここで言う「トレーサビリティ」とは、ソフトウェア要件のトレーサビリティであり、要件がソフトウェアのどこで実現され、検証されているか追跡可能にすることである。

しかし、いざ取り組みを始めると、次第にトレーサビリティを「取る」ことがゴールにすり替わってしまう傾向が見受けられる。その原因は、トレーサビリティに関する方針が定まらないまま、手探りで運用を進めていることにあると考える。

目的に沿った視点、範囲、粒度と運用ルールを方針として定めれば、運用は現場に定着し、トレーサビリティに取り組む本来のメリットをより実感できるはずである。本稿では、上記を実現する具体的な方法を提案する。

2. 背景

車載システム開発において、機能安全規格への対応や品質向上などの目的で、すでに多くの人々が、トレーサビリティの重要性や意義を認識している。

弊社もその認識はあったが、取り組みを始めた当初は、さまざまな問題に直面していた。例えば、現場に浸透しない、ツールを使いこなせない、工数が掛かり過ぎるなどの問題が挙げられる。

そこで、2018年1月に開催された「カーエレクトロニクス技術展」において、弊社ブースの来場者を対象にアンケート調査を実施した(n=29)。「トレーサビリティにおける課題は何か(複数選択可)」という質問に対し、「現場への浸透」、「ツ

ルの機能や使い勝手」、「工数増大」との回答が全体の7割を占めた。この結果から、他の企業も同様の問題を抱えていることが明らかになった。

前述の問題について、弊社で取り組んだ事例をもとに以下のように原因を分析した。

- ・現場に浸透しなかったのは、トレーサビリティの作業に掛けた労力に対して、メリットが小さいと感じたから
- ・ツールの機能や使い勝手に課題を感じたのは、自分たちの使い方とツールが想定している使い方間にあるギャップを埋められていなかったから
- ・多大な工数を要したのは、手探りで運用をスタートしてしまったから

これらは、トレーサビリティの取り組みに関する方針を定めきらないまま、ツールを導入することから進め、運用を場当たり的に行っていたために起きた。そして、方針として次の2点が定まっていなかったことに原因がある。

- ・視点、範囲、粒度
- ・運用ルール

1点目に、視点、範囲、粒度が定まっていなくて、表1のように情報は複雑化し、作業量は増加する。

2点目に、運用ルールが定まっていなくて、担当者の負担が一層大きくなり、「取る」ことがゴールになる現象に陥りやすい。

以上のように、現場が直面している問題の根本には、「手探り状態」で運用をスタートしている状況がある。これを解決するために、取り組みに関する方針を明確にすることが必要である。

表1 視点、範囲、粒度の定義と影響

視点	定義	影響
視点	どこに着目してトレーサビリティを取るのか、何の流れを見えるようにしたいのか	視点が定まっていなくて、範囲や粒度も明確にできないため、情報の取捨選択がされず情報が複雑化する
範囲	開発成果物全般から、何をトレーサビリティの対象とするのか	範囲が網羅されていないと不完全な情報になり、逆に余計なものまで範囲にしていると情報が複雑化する
粒度	トレーサビリティを取る単位のこと	範囲が網羅されていても、粒度がまばらになっていたり、粗すぎたり・細かすぎたりしていると、何の流れを見えるようにしたいのかが伝わりづらくなる

3. 解決策の提案

前述した問題に対し、具体的な方針として定めるべきことを提案する。

なお、取り組み方針として定めた内容は明文化し、プロジェクト内で共有するのがよい。

参考までに、弊社では、この解決策を3つの開発プロジェクトで実践している。その結果、トレーサビリティに関する対応工数が、以前に比べ減少するという効果が出てきている。

3.1. 目的を再確認する

ソフトウェアのトレーサビリティに取り組む目的は、ソフトウェアの品質の維持・向上に寄与することである。より具体的に言えば、検証不足や設計ミスなどの人的要因のリスクを最小化するために、開発成果物間を追跡可能にするのである。

3.2. 視点、範囲、粒度を定める

視点、範囲、粒度を定めることで、トレーサビリティの全体像が明らかになる。

ソフトウェア開発を行うとき、成果物内で、機能IDやテストケースIDなど、何らかの識別子をすでに付与していることと思う。しかし、それらのあらゆる識別子を関連付ければ、上記の目的を達成するようなトレーサビリティを実現できるかと言えば否である。これを実施すると、情報量が多く複雑で、作り上げるためには多大な労力を要するが、関連するものが増えすぎて追跡可能性が下がってしまう。

そこで、ポイントとなるのが前述した視点、範囲、粒度である。

視点は、機能に着目し、要求から、要件、設計、実装、テストの流れを可視化すると定める。

範囲は、開発における成果物全般を対象にするが、視点から判断し、本筋から外れるような補助資料などは対象外にする。

粒度は、各工程の最小の要素とする。例えば設計工程では、工程内で設計が全体から部分へと詳細化されるが、その詳細化した最小の要素を次工程の要素と関連付ける。

これらについて、プロジェクト内で検討していく際には、図1のように、工程や成果物の関係性を図示していくことが、皆の認識を合わせることに非常に役立つ。また、折に触れてこの図に立ち返ることで、進むべき方向を見定めることができるため、取り組みに不可欠なツールと言える。

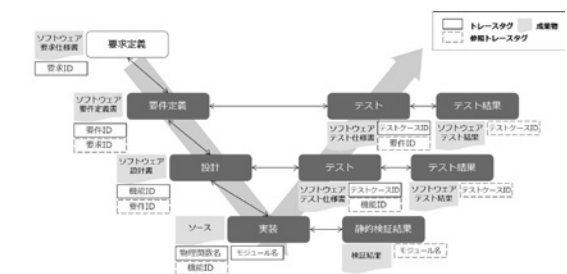


図1 成果物関連付けモデル

3.3. 運用ルールを定める

運用ルールとして定めるべき主なことを示す。この運用ルールは、Automotive SPICE 3.0[1]をよりどころとし、これまで弊社でトレーサビリティに取り組んだ経験をもとに作成した。

また、トレーサビリティツールは、キャッツ社の「ZIPC TERAS」を利用することを前提とする。

①関連付けはタグベースで行う

成果物に関連付けの起点となるタグを書き込むことで、成果物間を関連付ける。それらの成果物を ZIPC TERAS に登録すると、リンクを自動作成することが可能である。ZIPC TERAS は、クライアント画面から、手動でリンクを作成することもできるが、その場合、成果物と ZIPC TERAS 上のリンクの 2 箇所をメンテナンスすることになる。タグベースであれば成果物のみのメンテナンスで済む。

②作業の完了基準を定義する

トレーサビリティの作業は終わりが分かりにくい面がある。適切に作業の完了を判断するために、完了基準は表 2 の通りとする。

表2 定量的基準・定性的基準

定量的基準	<ul style="list-style-type: none"> ✓ 対象として規定した成果物が全てトレーサビリティツールに登録済みであること ✓ 登録された成果物は、全てタグによるリンクが作成されていること ✓ 登録された成果物は、カバレッジ率が100%であること
定性的基準	<ul style="list-style-type: none"> ✓ 要求から要件、設計、実装、テストまでを一貫して抜け漏れなくトレーサビリティ情報が取得できていること ✓ 要件から実装までは、各工程の最小要素にタグが付与されてつながること ✓ テスト仕様と対応する設計工程、テスト仕様とテスト実施結果との双方向性が確立されていること

③カバレッジ基準を定義する

カバレッジとは、対象成果物が上流/下流の成果物とどれくらいリンクしているかの度合いを指す。各工程の成果物は、カバレッジ率が 100% であることを基準とする。

もし、カバレッジ率が 100% 未満になる場合は、理由を明確にし、ZIPC TERAS の Properties のコメント欄に理由を記録する。

ZIPC TERAS には、カバレッジの計測対象外とする「例外設定」という機能があるが、成果物の中で、タグが付与されている箇所に例外設定は行わない。

④タグを埋め込むことを前提にしたフォーマットにする

あらかじめ、トレーサビリティのタグを記入する欄をフォーマットに作成しておく。タグを記入

する箇所が固定されれば、タグの記入漏れを防止しやすく、ZIPC TERAS に登録する際に必要となる解析ルールファイルの再利用が可能になるため、作業が効率化する。

⑤タグとなる ID の採番ルール定める

ID の採番ルールは以下の通りとする。

- ・ ID は、工程を移行するときに、適宜振り替えながら、上流/下流へのトレースを可能とする
- ・ ID に連番を含める場合、番号が途中で抜けることを許容する
- ・ ID は、製品やプロジェクトを越えて一意にする
- ・ ID は、半角英数字を用いる。フォーマットは以下の通りとし、関数名そのものをタグにする場合など、一部でフォーマットから外れることを許容する。

[製品識別子]-[工程識別子]-[機能識別子]-
[番号識別子]-[枝番]
【例】TV-RD-DIS-001-01

⑥トレーサビリティを成果物レビューの対象とする
成果物レビューで、設計書などの内容を確認する際に、トレーサビリティについてもレビューアが確認を行う。レビューイは、事前に ZIPC TERAS への登録・確認を済ませておき、レビューでは、上位成果物とのリンクに過不足ないことを説明する。トレーサビリティの作業結果を成果物レビューの確認対象とすることで、開発プロセスの中で、自然にトレーサビリティの取得を進めていくことが可能になる。

以上のように、トレーサビリティにどのように取り組むかを考えることは、そのまま自分たちがどのようにして開発を行ってきたかを考えることに等しい。自分たちの開発と向き合うよい機会となる。

4. 取得を促進する施策

解決策に加えて、トレーサビリティの取得を促進する施策として、トレーサビリティ情報を俯瞰で見られることを提案する。

一般論として、ものごとを俯瞰で見るとは、全体像を理解すること、抜け漏れや論理の飛躍に気づくことにつながる。トレーサビリティ情報、つまりリンクされた要素の関係性を俯瞰で見るとも同様のことが言える。

前述の通り方針を定めてから取り組むと、膨大な開発情報の中から、トレーサビリティ情報を作り上げることができる。その情報は、マトリクスや ZIPC TERAS の画面から見ることができるが、巨大なマトリクスや何重にも重なった線という複雑な姿をしている。そのため、詳細に確認することは可能であるが、トレーサビリティ情報の全体感をつかむことは難しい。

ここで、当社が開発した「QINeS-Tools Traceability Visualizer」(以降、本ツール)を紹介する。本ツールは、トレーサビリティツールに登録された情報をよりシンプルな形にして可視化することで、トレーサビリティの状況を一目で把握することを可能にしている。(図 2)

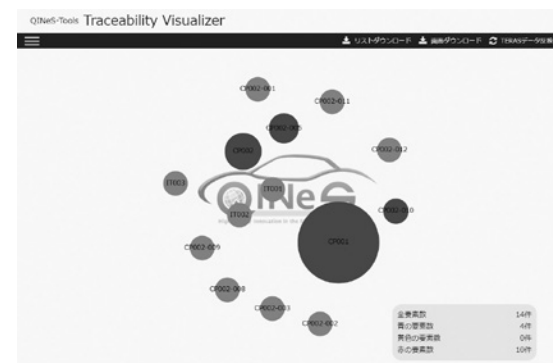


図2 Traceability Visualizer画面イメージ

本ツールは、視点として、下流工程へのリンクに着目しており、上流から下流への工程順のリンクと、開発工程と検証工程のように対になる工程間のリンクの両方を可視化する。さらに、可視化する範囲を自在に変えながら、粒度の大小を評価することを可能にしている。

開発プロセスの中で、開発者のセルフチェックや成果物レビューに本ツールを適用することができる。上位成果物とトレーサビリティが取れているかを、可視化した情報を見て簡単に把握できる

ため、取得・確認作業が促進される。

また、ある程度取得が進んだときには、可視化した情報を見て、要件の実現や設計の機能分割の妥当性を議論・確認することができる。

本ツールは、図 2 にあるようにトレーサビリティ情報を円で表示する。成果物の中でトレースタグを付与した箇所が、円で表示する対象となる。そのため、ZIPC TERAS で例外設定をせずとも下記のような可視化が可能になる。

円の大きさは、タグの下流工程へのリンク数を表し、リンク数が多いほど大きく表示する。円の色は、下流工程へのリンクのカバレッジを表している。カバレッジの状態によって、青(カバレッジ十分)、黄(カバレッジ不足)、赤(カバレッジ無し)の 3 色で表示する。円と円の距離は、下流工程において共通する対象へのリンクが多い場合、近くに表示する。

このように円で表現することで、直感的な理解を可能としている。他にも、トレーサビリティのセルフチェックやレビューをサポートする機能がある。

運用の定着と情報の活用によって、トレーサビリティに取り組むメリットをより実感できることを、本ツールを通じて実現していきたい。

5. おわりに

本稿では、トレーサビリティの運用が現場に定着し、そのメリットをより実感するための方法を提案した。本稿が、取り組みをより有意義なものとする一助となれば幸いである。

参考文献

[1] Automotive SPICE 3.0,
<http://www.automotivespice.com/>

お問い合わせ窓口
SCSK 株式会社
QINeS 製品企画部
トレーサビリティ製品担当
E-Mail : qines-info@scsk.jp

自動運転時代に向けた ZIPC 製品の今後の展開

キャッツ株式会社
 プロダクト事業本部 プロダクト部 部長
松枝 進介

1. 自動運転実現に向けたロードマップと課題

自動運転実現に向けた世界的な競争が本格化している。Google 等旧来の自動車業界以外のプレイヤーもこうした競争に参画してきており、技術開発の競争は激しさを増している。本稿ではまず、自動運転実現に向けた現時点でのロードマップと技術的な課題について分析を行いたい。

1.1. 実現に向けたロードマップ

平成 30 年 3 月 30 日に経済産業省より公開されているレポートを基に、自動運転実現に向けたロードマップを振り返ってみたい。SAE の自動運転実現レベルの定義に即した実現度と実現時期は次の通りとされている。

自家用車に関しては、2020 年までに高速道路に限定したレベル 2 の自動運転の実現、2025 年までには高速道路においてレベル 3 の自動運転が実現され、レベル 2 の自動運転については国道等の主要幹線道路までその対象が拡大されていく見通しである。また、MaaS に代表される事業用の自動車については、社会的ニーズの大きい地域には限定されるものの、2025 年までにレベル 4 の自動運転を搭載した事業用自動車が移動サービスや無人宅配等のサービスを実現するとされている。

本稿執筆時点（2018 年 10 月）より起算すると 2 年後には自動運転車が走り始め、7 年後には MaaS が実際のサービスとして本格的に開始される「自動運転社会」が実現することになる。こうした未来予測を背景に、自動運転技術の確立が、今まさに急がれている現状だと考えられる。

1.2. 自動運転技術実現に向けた課題と ZIPC 製品の今後の方向性

次に自動運転技術確立に向けた現状の課題について分析を行いたい。先ほどの経済産業省より公開されているレポートに戻ると、レベル 2 以上の自動走行実現に向けた技術課題が 10 項目として示されている。（表 1）

表1 自動走行実現に向けた重要10分野

分野
1. 地図
2. 通信インフラ
3. 認知技術
4. 判断技術
5. 人間工学
6. セーフティ
7. サイバーセキュリティ
8. ソフトウェア人材
9. 社会受容性
10. 安全性評価

自動運転を実現するための技術課題は表 1 の通り多岐にわたり、継続的な新技術の開発が求められる状況である。ZIPC 製品としては、これまでのキャッツ社で積み重ねてきた研究開発成果を活用できる領域として、上記の中から認知技術・判断技術の領域に対応した製品開発を推進する考えである。

認知・判断技術に関しては、これまでセンシングデータの収集技術等が開発され、確立されつつある状況となっており、今後は収集したデータを自動運転技術開発へ活用し、開發生産性の向上や新技術の実現をするかが、検討すべき点となっていると考えている。収集したデータの活用方法と

しては、ディープラーニングによる学習データ生成等、自動運転開発の推進に資する、様々なユースケースが想定されると考えているが、この中で ZIPC 製品の親和性やキャッツ社で蓄積した技術等考慮し、直近の製品開発テーマとして「自動運転設計・検証シナリオの自動生成」及び「自動運転検証のモデルベース開発化」について製品開発を進めていく考えである。

2. 自動運転設計・検証シナリオ生成の具体的な取り組みについて

自動生成化による開發生産性向上に向けた取り組みは現時点では途上であるものの、自動運転設計・検証シナリオの生成に関する具体的な取り組みについて紹介をする。

自動運転においては、ドライバーをシステム化する必要があり、要求定義の段階及び検証においては、自動車を取り巻く多様で複雑な外界の状況

をモデリングし、シナリオ化する必要がある。モデリング対象となる外界の状況としては、例として道路構造や障害物、車両や信号の状態、交通量モデルや道路交通法に至るまで、分析すべき対象とその組み合わせは複雑多岐にわたることから、この状態を抜け・漏れ無く抽出することが、開発上の解決すべき重要な課題となっている。

こうした課題を解決するため、キャッツ社としては、USDМ、要求仕様整理表、状態遷移表を用いた分析手法を推奨している。USDМを活用することにより、要求を階層化し、範囲を絞ることで仕様の抜け・漏れを防止する。次に USDМにて抽出した仕様をインプットとし、要求仕様整理表を作成する。要求仕様整理表では、自然言語で記載される USDМの仕様を形式化することにより、仕様の曖昧性を排除する。その後、状態遷移表を用い、状態と事象を表形式化することで、処理と遷移の抜け・漏れを防止することを実現している。（図 1）

ドライバーのシステム化 モデリング手法

3つの道具で段階的に「要求仕様」の網羅性を高める。状態遷移表からテストシナリオを抽出。

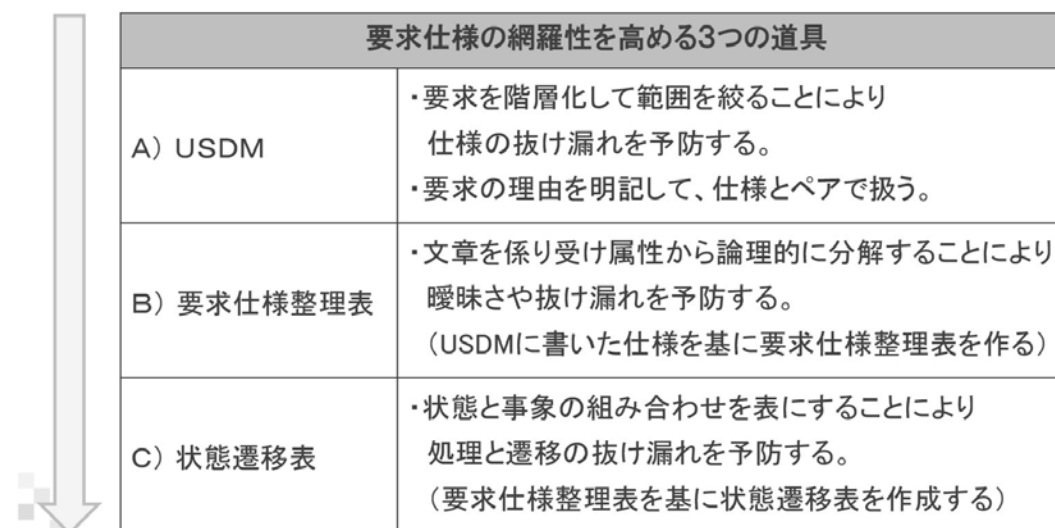


図1 シナリオモデリング手法

こうしたモデリング手法を活用し、シナリオ生成の分析を進めつつも、分析プロセスの中でZIPC製品を活用することで、生産性向上を実現することを推進している。

例えば、状態遷移表化したシナリオをZIPC Testerを用い検証するテストシナリオを自動生成することや、バリエーションが複雑化するシナリオをPair Wise法を活用し、品質劣化を発生させず、シナリオ数を合理的に削減するため、ZIPC FOTの活用をすることを推進する考えである。(図2)

現状では自動生成化の領域が部分的に留まっているが、将来的には自動化を実現する領域を拡大し、更なる生産性向上を実現することを考えている。

3. おわりに

本稿では自動運転開発に向けたZIPC製品の今後の展開について記載を行ったが、これまで活用してきた領域についても引き続き対応の強化を推進することには変わりがない。

一方、CASE時代の到来に代表されるように製品開発及び制御開発の分野も、時代の進展に伴い、新しい技術開発が展開されていくことになる。ZIPC製品としては、自動運転開発に留まらず、今後生まれるであろう新しい技術開発領域に対し、技術課題の解決及び生産性や品質の向上に資する製品を継続的に開発し、製品開発現場の改善に貢献を続ける考えである。

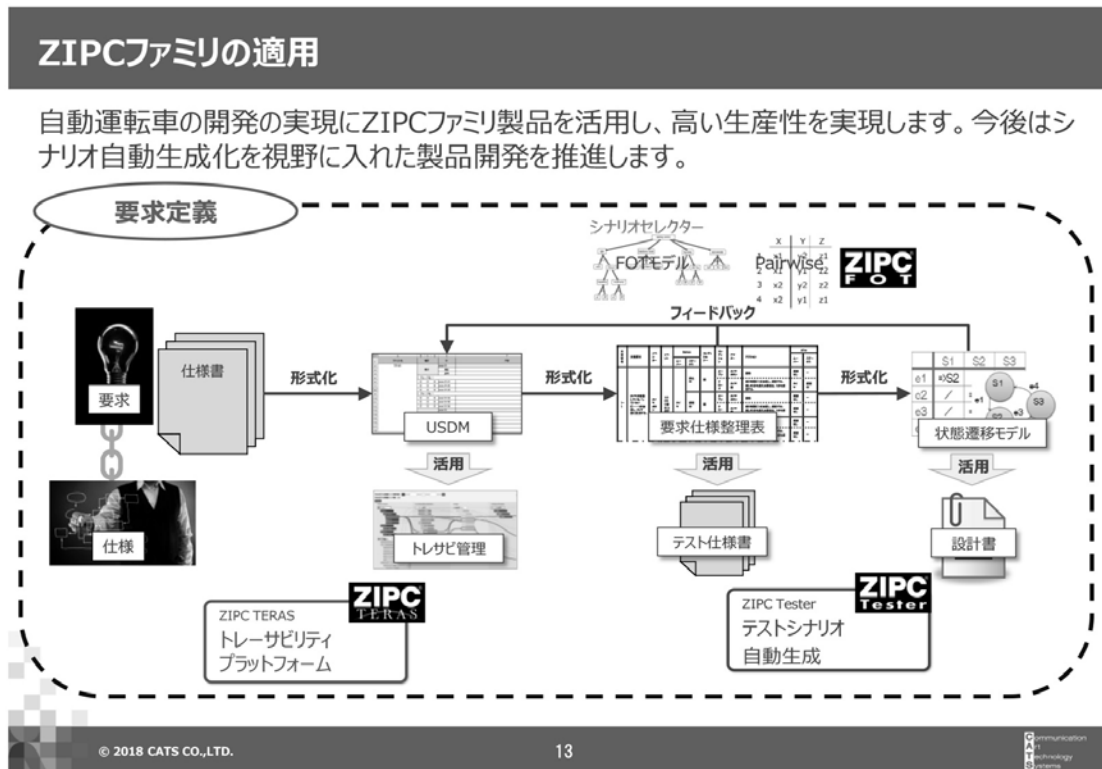


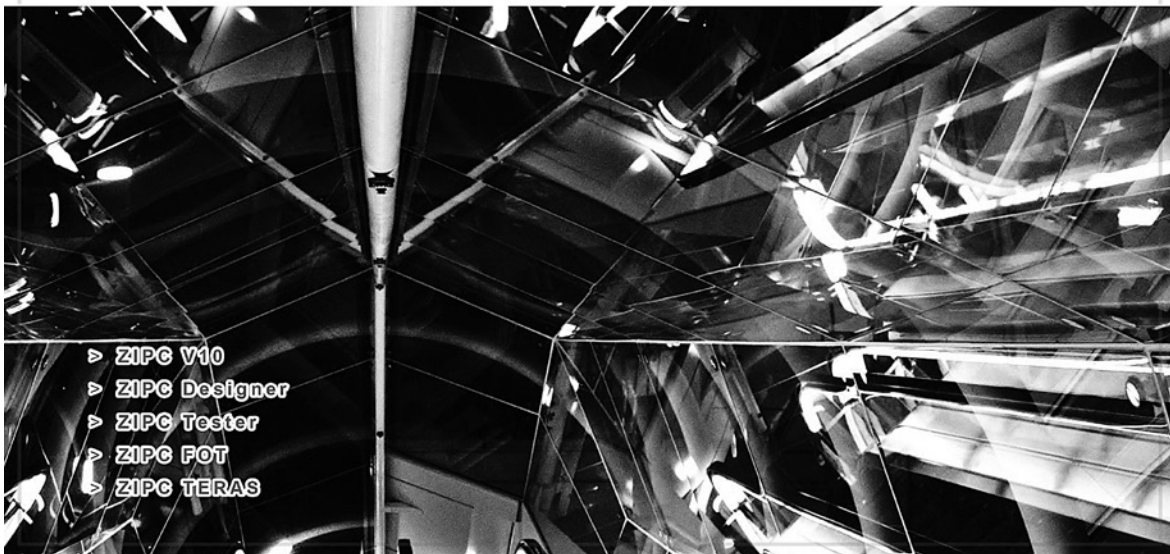
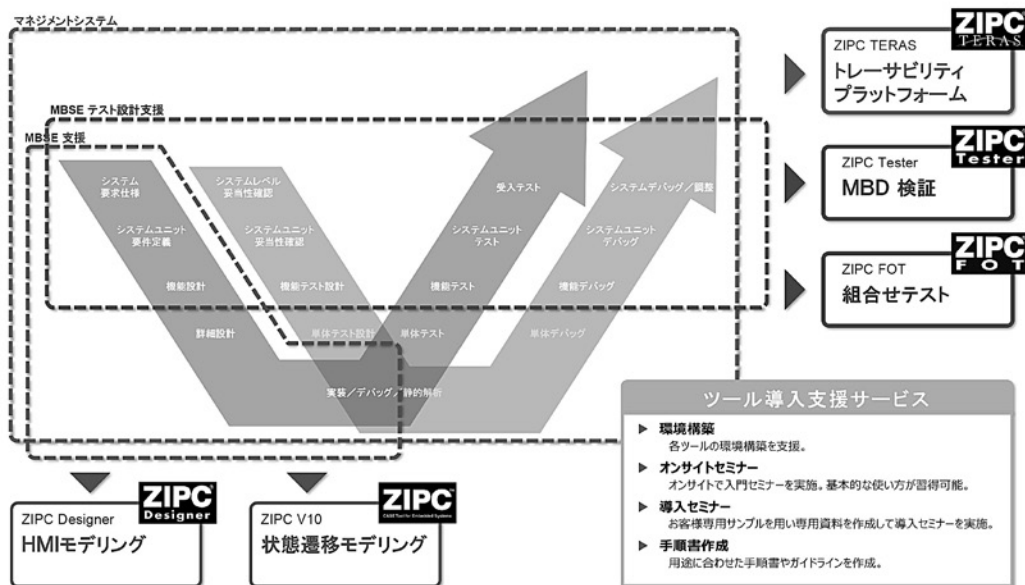
図2 シナリオ生成におけるZIPC製品活用例

Communication
Art
Technology
Systems

CATS の先端技術が実現したソフトウェア開発支援ツール群

ZIPC FAMILY

ZIPC FAMILY は、CATS にて開発・販売・サポート・導入支援等を行っているソフトウェア開発支援ツール群の総称です。システム要求仕様からシステムデバッグまでの各フェーズを支援するツールをご用意しております。



- ▶ ZIPC V10
- ▶ ZIPC Designer
- ▶ ZIPC Tester
- ▶ ZIPC FOT
- ▶ ZIPC TERAS

Communication
Art
Technology
Systems

ZIPC V10

状態遷移モデリングツール



状態モデル CASE ツールのスタンダード

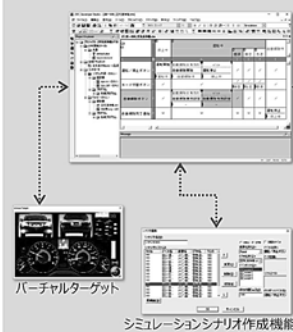
ZIPC V10 は、システムの複雑な振舞いを状態遷移モデルで設計、シミュレーションで動作検証、コードを自動生成し、ソフトウェアの品質・生産性の向上を支援するモデルベース開発ツールです。拡張階層化状態遷移表(EHSTM)設計手法を採用しており、状態遷移モデルの課題(巨大化、複雑化、etc)の解決を支援します。

ツールの機能概要

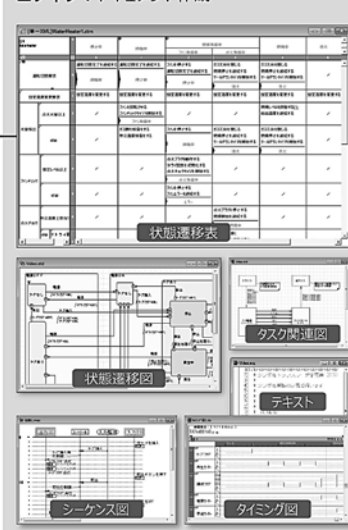
開発作業のフェーズに合わせて、デザイン・フェーズ、シミュレーション・フェーズ、ターゲット・フェーズと用意しており、各フェーズでエディタ、チェッカ、シミュレータ、ジェネレータ等の機能を使用して開発を進めます。

シミュレータ：振舞い検証

- ・抽象度の高いモデルを使い、簡単にシミュレーションの実行が可能。
- ・コード化の前に、状態遷移の振る舞いの正しさを検証できるため、コードレベルの不具合が激減。
- ・シミュレーション実行結果から、状態遷移表のカバレッジを確認可能。
- ・実機の完成前に、バーチャルターゲットを使いPC上で振る舞いの確認が可能。
- ・多彩なシミュレーションデバッグ機能を搭載。



エディタ：ドキュメント作成



チェッカ：ドキュメントチェック

- ・作成したドキュメントに対して静的なチェックを行い、人手によるクアリティチェックを省略し防止する。
- ・各セルの未記入チェック、遷移先セルに記述された状態の存在チェック、ある状態から他の状態に遷移しない場合(未遷移状態)チェック等。

ジェネレータ：Cコード自動生成

- ・状態管理部分を全て自動生成
- ・ANSI C 準拠、MISRA C 準拠のコードを自動生成
- ・フォーマットの統一
- ・ドキュメントとコードが常に一致

期待される効果

- ・あいまいな仕様の定義を明確に表現でき、仕様と設計、成果物間での齟齬による手戻りを減らせます。
- ・複雑な振舞いの制御も状態モデル設計で整理できるので、多数のフラグや変数を減らせます。
- ・コーディング前に振舞いの正しさを確認できます。
- ・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

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。

ツールの適用効果

- ✓ 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 1/30 ~ 1/4 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 に低下した
- ✓ 従来手法による開発費用と比較して 26% の費用が削減された etc

Trusted Global Innovator
NTT DATA Group

ZIPC Designer

HMI モデリングツール



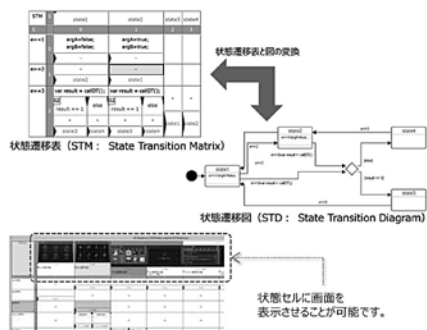
HMI の振舞い設計は、状態遷移/画面遷移が重要

ZIPC Designer は、HMI の「振舞い」「画面遷移」を、画面(状態)遷移モデル、決定表で記述し、PC 上のシミュレーションで動作確認後、C コード生成まで支援するモデルベース開発ツールです。また、主要なオーサリングツールとの連携も可能です。

ツールの機能概要

特長① 画面(状態)遷移設計

あいまいな仕様を明確に表現できます。複雑な振舞いも状態と事象の組合せで網羅します。状態モデルがそのまま設計書として資産化できます。



特長② 決定表

ISO 準拠の決定エディタです。複雑な条件の組合せを整理、定義できます。

条件部	動作部	条件記述部		条件指定部	
		条件	動作	条件	動作
1	argA	OFF	ON	OFF	ON
2	argB	OFF	OFF	ON	ON
1	SetMessage	setMsg("1")	setMsg("2")	setMsg("3")	setMsg("4")
2	SetEvent	e=1;	e=2;	e=3;	
3	CallSTH	stm_eventCall();stm_eventCall();	stm_eventCall();	stm_eventCall();	

特長③ 様々なオーサリングツールとの連携

主要なオーサリングツールと連携し、実機での表示に近い環境で HMI の動作を確認できます。

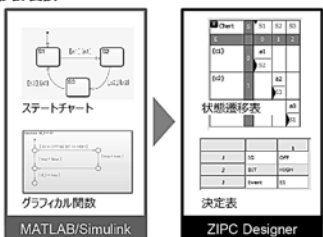


Automotive Option

ZIPC Designer に、オプション機能の「Automotive Option」を組込むことにより、Simulink/Stateflow モデルと状態遷移表を相互に変換可能となります。

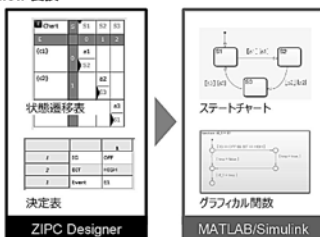
Stateflow ⇒ 状態遷移表 変換

Stateflow モデルを、状態遷移表モデルへ変換します。遷移条件と状態の組合せ等を確認でき、Stateflow モデルの静的検証を支援します。



状態遷移表 ⇒ Stateflow 変換

状態遷移表モデルを、Stateflow モデルへ変換します。状態遷移表でできつつ設計したモデルを Stateflow へ自動変換でき、状態遷移設計の品質を向上させます。



期待される効果

- HMI の「グラフィックデザイン」と「振舞い、画面遷移」のモデリングを並行で進められる。
- 動く仕様書により明確な振舞いを早期に定義、確定できます。
- 振舞いのモデルからソースコードを生成し手戻りが減ります。
- 状態遷移/画面遷移モデル、決定表により、複雑な制御も可能です。

ZIPC TERAS

トレーサビリティツールプラットフォーム



既存の資産/環境を変えずに成果物間のトレーサビリティを確保

ZIPC TERAS は、システム開発に関わるすべての成果物間のトレーサビリティを確保するオープンなトレーサビリティツールプラットフォームを提供します。様々なモデリングツール、構成管理ツール、変更管理ツール、プロジェクト管理ツールと連携した ALM 環境を構築できます。

ツールの機能概要

システム および ソフトウェア開発に関わる全ての成果物間のトレーサビリティの確保に特化したツールプラットフォームです。

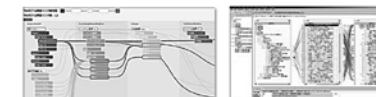
特長① 既存ファイル・文書形式の継承

既存の Microsoft Word、Excel、PowerPoint、PDF、テキスト文書等のドキュメントを項目毎に取り込むことが可能です。



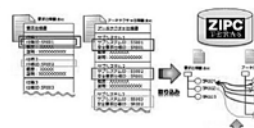
特長② リンクエディタでの直感的操作

リンクエディタで直感的な操作により、トレース作業を行い、リンクビューで全体的なドキュメント構造を俯瞰(確認)することが可能です。



特長③ タグベースリンクによる自動リンク

リンク対象のドキュメントに、予め記述されている「トレースタグ」を解析し、ドキュメント間で対応する項目を自動でリンクすることが可能です。



特長④ カバレッジ確認、影響範囲検索

上流と下流の対応関係のモレヌケを確認・判定できます。また、変更・不具合に対する影響範囲を明確にできます。



作業の流れ



Microsoft Office 等の普段利用しているツールで作成された文書(既存資産)を登録できます。

文書を ZIPC TERAS に登録します。ローカルにある文書や構成管理ツール(Subversion, Git 等)で管理された文書も登録できます。

文書内に ID が記述されている場合には、ツールが自動でトレース情報を作成します。文書内に ID が無い場合には、手動でトレース情報を作成します。

作成したトレース情報を利用して、カバレッジの確認や影響範囲の検索等の機能を使い、品質確保に役立ちます。

期待される効果

- 「担当者でないと分からない」こと(属人性)がなくなる。
- 要件がすべて設計、実装、試験されているかを検証できる。
- 成果物をレビューする際に、上流の要件を簡単に確認できる。
- 変更に対する影響範囲を論理的に把握できる。
- 不具合の影響範囲が明確になり、モレヌケなく対応できる。
- ドキュメントのギャップ診断として活用できる。

各種 OSS との連携

開発現場にて多く活用されている OSS と連携して、品質を確保した開発プロセスの運用が実現可能です。

変更管理、課題管理、プロジェクト管理

- Trac、Redmine、JIRA (※1) と連携可能。
- 影響範囲分析やカバレッジ確認結果を基に、ZIPC TERAS からチェック起票を支援します。
- 登録済のチケットを ZIPC TERAS に取込むことも可能です。

構成管理

- Subversion、Git と連携可能。
- ローカル環境のファイルだけでなく、Subversion、Git のリポジトリ上のファイルでトレーサビリティ管理も可能です。

※1: JIRA は、アトラシアン社製の管理ツール(有償)です。

ZIPC FOT

パラメータ組合せテスト設計ツール



テストケース生成に CTM を活用

ZIPC FOT は、CTM(Classification Tree Method)を活用して、ブラックボックステストのテストケース作成を支援するツールです。CTM でテスト設計して、N-Wise 網羅度のテストケースを自動生成します。

ツールの機能概要

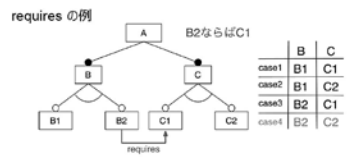
特長① CTM を用いたテスト設計

システムに対するパラメータとパラメータ値をツリー状に設計します。ツリー状に設計するため、テスト設計のレビューが容易になります。



特長② テスト制約記述

テスト設計で考慮すべきテスト制約を記述できます。無効値を含むテスト設計も可能です。



特長③ N-wise 網羅のテストケース生成

テスト戦略に合わせて、テストケースを絞り込み生成できます。様々な網羅基準で、絞り込み可能です。

網羅基準	生成されたテストケース数
3-Wise (全網羅)	24 件
2-Wise (Pairwise)	13 件
1-Wise (Pairwise)	4 件

特長④ テスト仕様書生成

生成したテストケースをテスト仕様書として Excel 形式で出力可能です。Excel 形式で出力することにより、多くのツールとの連携が可能です。

ツール利用方法



導入事例

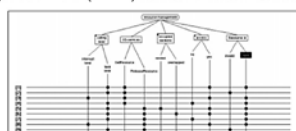
オムロンソーシアルソリューションズ株式会社様にて、「IC カード千円チャージ機」に ZIPC FOT を適用して頂きました。

- ・テストツリーの作成は容易
- ・テストツリーのレビューが図で容易
- ・テストケース数を絞り込みによりコントロール可能 などが利点
- ・テストケースの網羅性を保証可能 なども利点



CTM (Classification Tree Method) とは

CTM は、1993年に開発されたブラックボックステストを支援するテスト手法で、システムのパラメータ(Aspects)とパラメータ値(Class)を分類木に設計して、テストケースを生成する手法です。ソフトウェアテストの国際標準である ISO/IEC 29119 で仕様ベーステスト技法として挙げられています。



出典：OSEK/VDX Conformance Testing Methodology Version2.0, April 16th 1999

ZIPC Tester

MBD テスト設計ツール



モデルベース開発におけるシナリオテストを総合支援

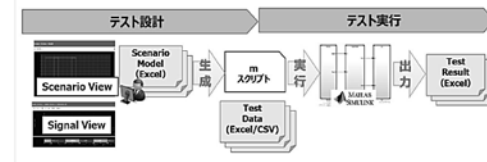
ZIPC Tester は、シナリオベースのテスト設計、テスト結果判定、テスト管理を支援するツールです。「テスト設計品質の担当者依存 軽減」「既存のカバレッジテストツールでは困難な仕様の検証」を支援します。

ツールの機能概要

状態遷移表、タイミングチャートの専用エディタにより、効率的なテスト設計を支援します。また、テストケース、テスト結果を一元管理し、自動判定機能により、テスト判定を支援します。MATLAB/Simulink との連携も可能です。

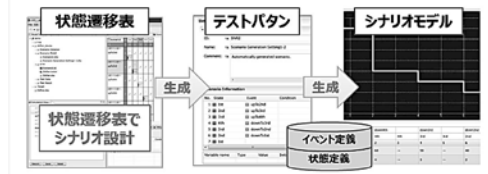
特長① シナリオ設計支援機能

テストシナリオのExcelデータをブラウザで直感的に確認できます。テストシナリオをSimulinkが認識できる形式のTest Dataに変換してスクリプトを実行し、テストモデル作成からシミュレーション実行まで自動化できます。



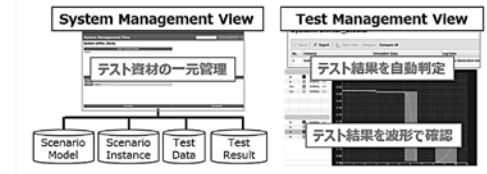
特長② 状態遷移表からのシナリオ生成、テストシナリオからのリバース

状態遷移表によりテストシナリオを設計し、テストボタンを自動生成することができます。表に基づく設計により、シナリオのモレヌケを防止することができます。また、イベント・スタート定義により、シナリオモデルを自動生成できます。



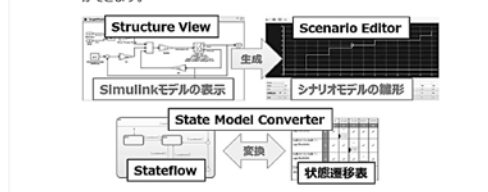
特長③ テスト管理機能

テスト管理機能により、シナリオモデル、シナリオインスタンス、テストデータ等のテスト資材や、テスト結果を一元管理できます。また、テスト結果の自動判定ができます。

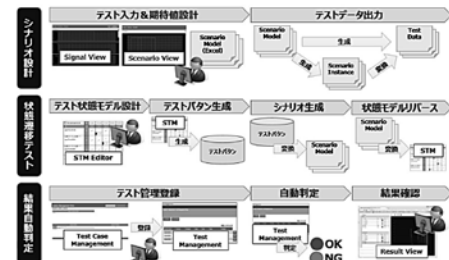


特長④ Simulink 連携

Simulink モデルを読み込み、テストシナリオの雛形やテストデータ、テスト実行のスク립トの生成が可能です。また、Stateflow と状態遷移表の相互変換ができます。



利用シーン



適用領域

	ホワイトボックス		ブラックボックス	
	カバレッジ(MCDC等)	組合せ	シナリオ	状態遷移
テスト設計	MATLAB/Simulink	ZIPC Tester	ZIPC Tester	ZIPC Tester
テスト実行	MATLAB/Simulink	MATLAB/Simulink		
テスト確認	MATLAB/Simulink	ZIPC Tester	ZIPC Tester	ZIPC Tester

貴社ドメインに合わせたご支援により理解が深まります

ツール導入支援サービス

ツールを効果的に使いこなせるようになります

CATS では、ツールの導入支援から導入後のサポートまで、お客様のニーズに合わせてお手伝いしております。ぜひ、ご相談ください。

<p>ツールを 導入したい</p>	<ul style="list-style-type: none"> ✓ ツールを使いこなすのに時間がかかりそう・・・ ✓ ツールは自社の開発環境に適しているのかな・・・ ✓ ツールを使ってイメージ通りに設計できるかな・・・ ✓ 既存の開発環境のノウハウをムダにしたい・・・ ✓ 既存資産をムダにしたい・・・ etc 	<ul style="list-style-type: none"> ✓ 手法トレーニング ✓ ツール・トレーニング ✓ ファーストステップ導入支援 ✓ 設計コンサルティング ✓ 受託開発
<p>ツールを さらに 使いこなしたい</p>	<ul style="list-style-type: none"> ✓ ツールの機能を十分に使いこなせていない・・・ ✓ ツールの使い方を調べるのに時間がかかりそう・・・ ✓ もっとツールの効果的な使い方はないのか・・・ ✓ ツールを社内に広げたいので新人や協力会社へ教育したいが、リソース/時間がない・・・ etc 	<ul style="list-style-type: none"> ✓ 既存資産のモデル置き換え ✓ エキスパート派遣 ✓ 特別運用サポート etc



開発支援ツールのことならお任せください

ツール開発サービス

ツールベンダとしての開発実績と技術力でご要望に応えます

ツール開発のノウハウを活用して、専用ツールの開発や、内製ツール、市販ツールに対する支援ツール/機能を開発致します。要求仕様の作成からツールの保守まで、ご要望に合わせた対応が可能です。設計・検証ツールからコンバータなどの支援機能、領域特有のツールなど、多くの開発実績がございます。

ツール開発サービス	ツール保守運用サービス																				
<ul style="list-style-type: none"> ▶ ツール開発サービス ユーザ企業様の要望に合わせた新規ツールの開発 ▶ ツールカスタマイズサービス 内製ツールや ZIPC FAMILY の機能追加や機能カスタマイズ ▶ 連携ツール開発サービス 市販ツール、内製ツールと連携する機能の開発 	<ul style="list-style-type: none"> ▶ QA対応 ツールに対する使い方や要望に対する QA 対応 ▶ 不具合改修 ツール不具合・ドキュメントの改修 ▶ リビジョンアップ版の提供 リビジョンアップ版を提供 																				
COC や NTTデータグループと連携した柔軟な開発 & 保守体制																					
<p>ユーザー ↔ CATS Technology Systems (ツールの開発 & 保守)</p>	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="background-color: #cccccc;">ツール開発フェーズ</th> <th style="background-color: #cccccc;">保守運用フェーズ</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">要求仕様作成</td> <td style="text-align: center;">QA 対応</td> </tr> <tr> <td style="text-align: center;">↓</td> <td style="text-align: center;">↓</td> </tr> <tr> <td style="text-align: center;">ツール設計</td> <td style="text-align: center;">不具合改修</td> </tr> <tr> <td style="text-align: center;">↓</td> <td style="text-align: center;">↓</td> </tr> <tr> <td style="text-align: center;">ツール実装</td> <td style="text-align: center;">ドキュメント改修</td> </tr> <tr> <td style="text-align: center;">↓</td> <td style="text-align: center;">↓</td> </tr> <tr> <td style="text-align: center;">ツールテスト</td> <td style="text-align: center;">リビジョンアップ</td> </tr> <tr> <td style="text-align: center;">↓</td> <td style="text-align: center;">↓</td> </tr> <tr> <td style="text-align: center;">マニュアル類作成</td> <td style="text-align: center;">※ 多言語対応、機能安全対応も可能です</td> </tr> </tbody> </table>	ツール開発フェーズ	保守運用フェーズ	要求仕様作成	QA 対応	↓	↓	ツール設計	不具合改修	↓	↓	ツール実装	ドキュメント改修	↓	↓	ツールテスト	リビジョンアップ	↓	↓	マニュアル類作成	※ 多言語対応、機能安全対応も可能です
ツール開発フェーズ	保守運用フェーズ																				
要求仕様作成	QA 対応																				
↓	↓																				
ツール設計	不具合改修																				
↓	↓																				
ツール実装	ドキュメント改修																				
↓	↓																				
ツールテスト	リビジョンアップ																				
↓	↓																				
マニュアル類作成	※ 多言語対応、機能安全対応も可能です																				



CATS出展
小間番号: **C-67**
(展示ホールD)



ブースイメージ

ブース 出展内容

- ◇次世代オートモーティブ社会のCASE
「自動運転ソフトウェア検証プラットフォーム」
- ◇次世代モビリティ時代のZIPCファミリの活用
「自動運転開発支援」
「HMI高度化へのZIPC Designer/V10の活用」

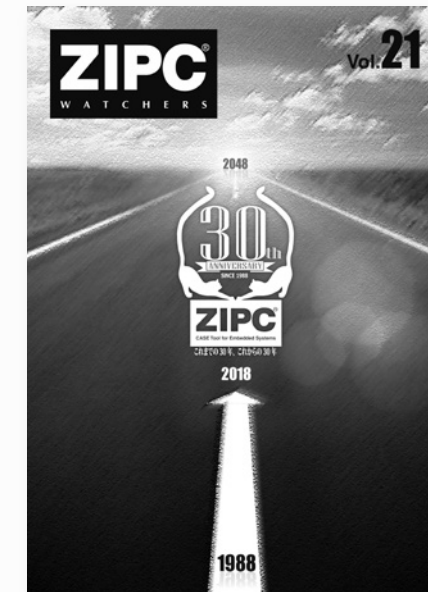
カンファレンスプログラム (無料/事前登録制)

- ◇日時:2018/11/15(木) 11:30-12:15
- ◇会場:展示会場内 セミナー会場C
- ◇セミナNo./カテゴリ/テーマ
JM-2 次世代モビリティトラック
「自動運転ソフトウェアテスト
～シナリオベースアプローチ PEGASUS～」

詳細はこちら

<http://www.zipc.com/event/et/2018/index.html>

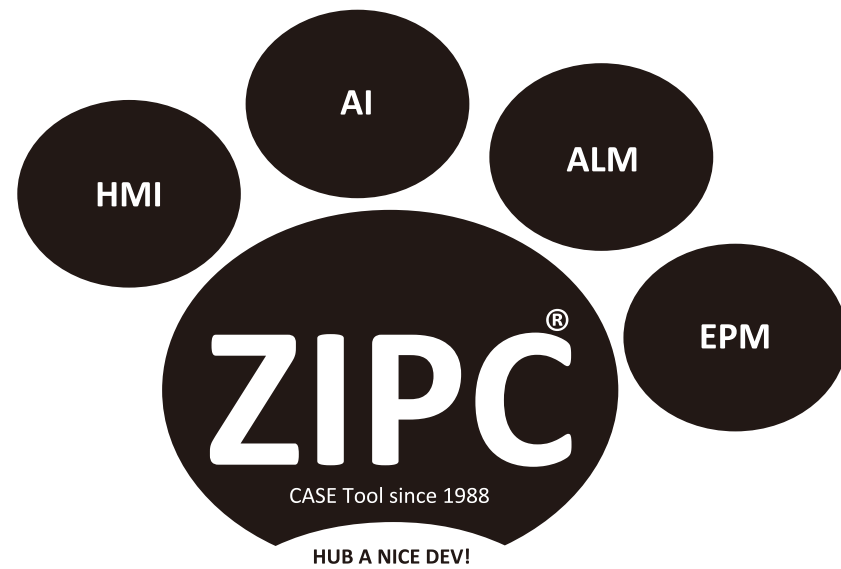
COVER IMAGE



これまでの30年 これからの30年

30年前、世界で初めて「拡張階層化状態遷移表」を設計手法として誕生したCASE(Computer Aided Software Engineering)ツール“ZIPC”は、さまざまな業種のソフトウェア開発のQCD向上に貢献してきました。

今後は、いま自動車産業が業界の垣根を越えて、次世代モビリティ社会の実現に向けて取り組むCASE([C]onnectivity [A]utonomous [S]hared [E]lectric)に対しても、これまで培われたノウハウと豊かな発想により新しい技術を創出し、世の中の発展とみなさまの明るい未来のため前進していく様子をイメージしています。



CATSのWebサイトでは、ソフトウェア開発支援・検証ツール、ソリューション技術、活動イベント、開催セミナー等の最新情報をご案内しています。ぜひご覧ください。

URL. <http://www.zipc.com/>



製品・サービス・現場の課題解決のご相談等、お気軽にお問合せください。

E-mail. info@zipc.com

ZIPC WATCHERS Vol.21

2018年11月14日 初版第1刷発行

発行 キャッツ株式会社
〒222-0033 神奈川県横浜市港北区新横浜3-1-9 アリーナタワー
TEL:045-473-2816 <http://www.zipc.com/> E-mail:info@zipc.com

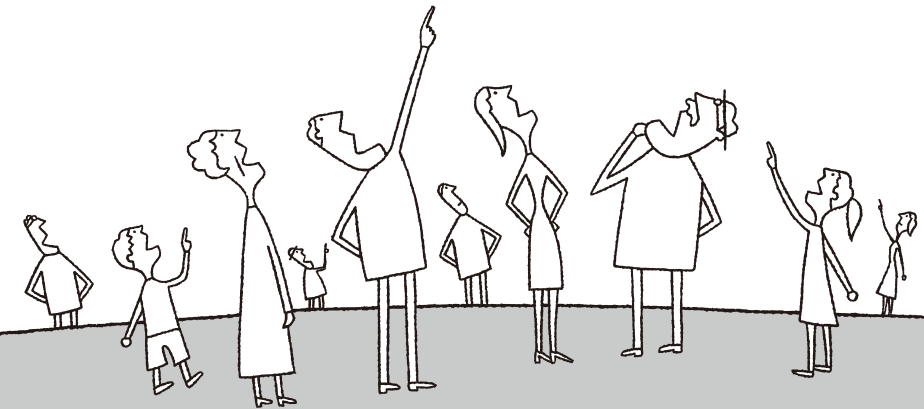
本誌に記載されている各製品・会社名は各開発／販売元会社の商標または登録商標です。本文中では、TM/®マークは明記していません。本誌掲載記事を、キャッツ株式会社の承諾無しに転載・翻訳複写・その他の複製及びデータベース・磁気媒体・工学媒体などへの入力を禁じます。無断で行いますと、損害賠償・著作権法の罰則の対象となることがあります。

UD書体

ユニバーサルデザイン書体

分かりやすく、読みやすく、間違えにくい。

車載・組み込み機器にもモリサワの「UD書体」



●書体見本(一例)

UD新ゴ R 車載のシーンで活躍する ABCDEFGHlabdefghi0123456789&!?
UD新丸ゴ R 車載のシーンで活躍する ABCDEFGHlabdefghi0123456789&!?
B 車載のシーンで活躍する ABCDEFGHlabdefghi0123456789&!?

車載・組み込みフォントについての詳しい情報は

www.morisawa.co.jp/products/fonts/embedding

車載・組み込みフォントに関するお問合せは
株式会社モリサワ 東京本社 フォントソリューション課
〒162-0822 東京都新宿区下宮比町2-27 Tel:03-3267-1305
E-mail:fontsolution@morisawa.co.jp

●モリサワUD書体は株式会社モリサワの登録商標です。



Trusted Global Innovator

信頼の歴史が、新たな時代を描く。



NTTデータは、2018年で創立30周年。
源流となる日本電信電話公社 データ通信本部の発足から50年。

〈お問い合わせ先〉 株式会社NTTデータ 広報部
〒135-6033 東京都江東区豊洲3-3-3 豊洲センタービル
<http://www.nttdata.com/jp/>

株式会社ネクスティエレクトロニクス
自社製品紹介 : 自動運転およびEV検討用シミュレーター

ezVDM

自動運転/EV分野に注目した車両運動モデルおよびバッテリーモデルを組み合わせたシミュレーション環境を提案します。お客様のニーズに合わせてご選択いただけます。

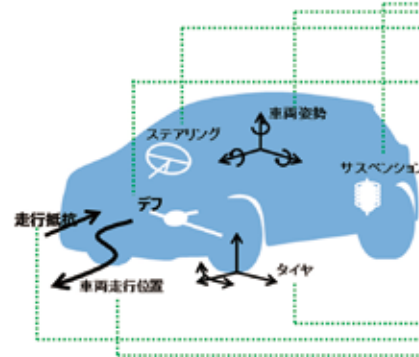
自動運転/ADAS検討に適した車両運動モデルパッケージ

複雑化する自動運転/ADASの制御開発をサポートします。

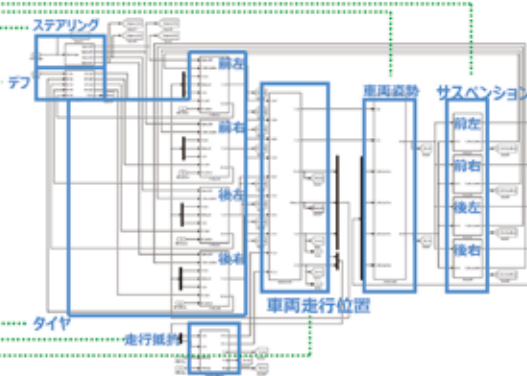
■特徴

- ・自動運転/ADASの制御検討をターゲットにした車両運動モデル
- ・適が必要なパラメータが少ないので、シミュレーションのインシタルタイムを削減可能
- ・制御に使用する入出力信号の追加や要素モデルの変更が柔軟に対応可能
- ・MILS、HILSに適用可能

■車両運動モデルのイメージ図



■車両運動モデル (Simulink)



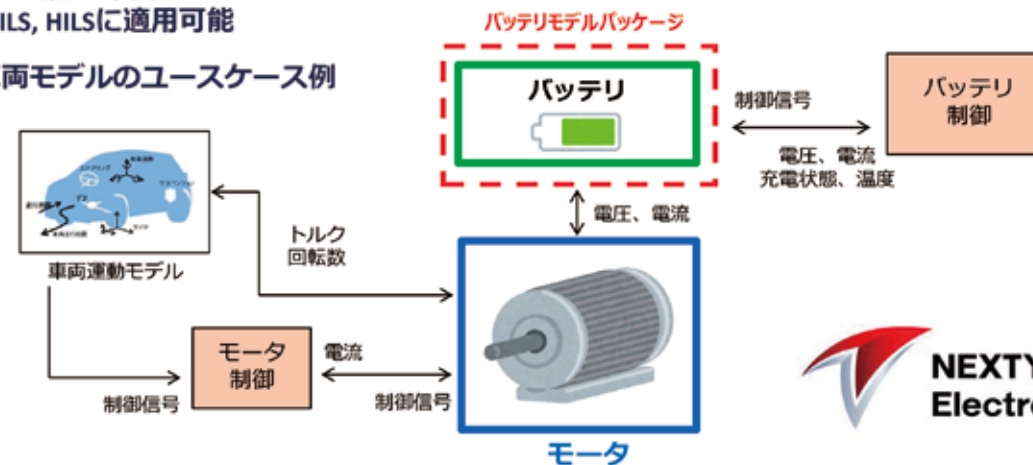
EV検討に適したバッテリーモデルパッケージ

車両運動モデルとバッテリーモデルの組み合わせにより、EV開発をサポートします。

■特徴

- ・物理モデルを数式で抽象化しており計算負荷が軽い
- ・適が必要なパラメータが少ないので、シミュレーションのインシタルタイムを削減可能
- ・標準的な特性を網羅したバッテリーモデル (材質特性、充放電特性、温度特性、熱損失特性)
- ・FMI規格に準拠
- ・MILS、HILSに適用可能

■EV車両モデルのユースケース例





これまでの30年、これからの30年

Communication
Art
Technology
Systems

www.ZIPC.com

キャッツ株式会社

〒222-0033 神奈川県横浜市港北区新横浜 3-1-9 アリーナタワー
TEL: 045-473-2816 FAX: 045-473-2673
<http://www.zipc.com/> E-mail: info@zipc.com

