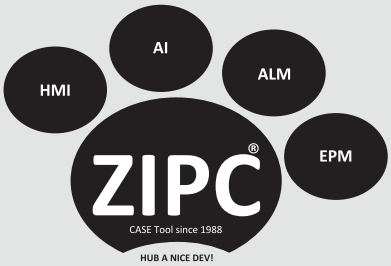
ZIPC WATCHERS

Vol.20

想像力をオン



■ インタビュー

- ●車載電子制御システムのモデルベース開発と今後の課題

■ コラム

- ●組込みシステムにハイブリッドなAIを

■ 特別寄稿

- ●コネクティッドカー／自動運転分野における ソフトウェアテクノロジーの動向と取り組み紹介
- ●製品認証における証跡管理ツールに求められる要件

■ 適用事例

- ●ZIPC TERASを活用した機能安全への取り組み
- ●車載システム開発におけるZIPC適用事例紹介
- ●衛星シミュレータへのZIPCの活用事例紹介

■ 学術研究

- ●IoT時代の組込みソフトウェアに向けた コンテキスト指向技術
- ●コードからSTMへリバース RexSTM for Cツールのこれまでとこれから

QINeS

High-Quality Innovation in the ECU-based Software

『近くにいるから出来ること』があります。
我々協業7社は、
世界中でニッポンの「ものづくり」が
高く評価され続けるために、
お客様と共に高品質なソフトウェアと
サービスを創りあげます。

—

- ビジネスキューブ・アンド・パートナーズ株式会社
- キャッツ株式会社
- イーソル株式会社
- 株式会社未来技術研究所
- 株式会社豆蔵
- SCSK株式会社
- 株式会社SRA

（アルファベット順）

SCSK

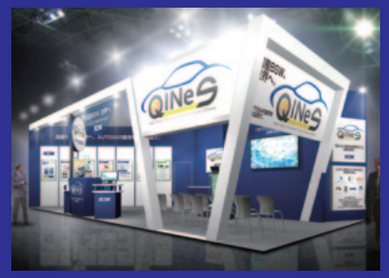
www.scsk.jp/product/ecu/ qines-info@scsk.jp

国産BSW、充実サポート

過去35年に渡り取組んできた車載システム開発の実績を活かし
お客様のAUTOSAR適用をワンストップで支援する
BSWおよび関連製品/サービス群「QINeS（クインズ）」
を提供しています

Embedded Technology 2017
組込み総合技術展

2017年11月15日（水）- 17日（金）10:00 - 17:00
パシフィコ横浜
小間番号：D-51



第10回［国際］カーエレクトロニクス技術展
～カーエレ JAPAN～

2018年1月17日（水）- 19日（金）10:00 - 18:00（最終日17:00）
東京ビッグサイト
小間番号：E52-37

ZIPC® WATCHERS

Vol.20

目　次

ZIPC WATCHERS Vol.20

人とクルマの絆をもっともっと深くする人馬一体へ挑戦し走る歓びを追求する自動車メーカー

車載電子制御システムのモデルベース開発と今後の課題

マツダ株式会社　統合制御システム開発本部
統括研究長 兼 首席研究員

今田 道宏

1. はじめに

マツダ株式会社は、“走る歓び”をテーマに人馬一体となるクルマ作りに挑戦し続けている自動車メーカーです。人の心を揺さぶる魅力的な自動車作りに取り組んでおられるマツダ株式会社 統合制御システム開発本部 統括研究長 兼 首席研究員 今田道宏様にお話を伺いました。

2. 自動車作りへの想い

自動車はただの走る道具ではないというMAZDA社の想いには、特にデザインがそれに表れていると思います。「魂動（こどう）デザインというデザインのネーミングに含まれた魂という字は、今にも動物が動き始めようとしているような躍動感を表現しようとしていて、今のマツダ車を非常に良く象徴していると思っています。それは、高級車も普及車も区別なく考えています。

また、「人馬一体」という表現をしているのですがその意味は、単なる機械を操るということではなく、人はクルマを自在に運転して走る歓びを感じ、更に自分自信の体が拡張していくイメージをテーマとしています。そのように体感してもらえる自動車作りを目指しています。

最近は、「G-ベクタリングコントロール」という技術。それは車体が曲がるときにステアリングを切りエンジンの出力を微妙に変えてあげることで、非常にクルマの姿勢がスムーズに変わっていくものなのですが、それも人とクルマの特性をとにかく突き詰めて考えていくということで、非常にこだわりを持って取り組んでいますし、それも特別な物を付けるというのではなく、技術でもともとクルマが持っている特性を活かした方法で自然な挙動になる、何か特別な装置を付けることなしに実現するような方法を採っています。

例えば、すぐ思いつくのは、曲げるのであれば左右独立したトルクをかける方法がありますが、それは極めて不自然な挙動になってしまいます。そのようなことからクルマが本来持っている特性を最大限に発揮しつつ極めてシンプルな方法で自然に姿勢を変えてあげるという考え方を大事にしています。

ユーザの方からは「リアシートに乗っていても車酔いしにくくなった」、ジャーナリスト試乗会などでは、「シートを変えましたか？」という質問が出たりします。乗っている人間の加重移動が自然になっているため、あたかも良いシートに交換したのかと思われるようです。

安全や環境性能はもちろん追及しながら、走る歓びを最も効率的に、本質を見極めて、賢いやりかたで突き詰めていくというこだわりを非常に強く持って日々取り組んでいます。

3. 本質となる一番ピンはどこに？

ボーリングの１番ピンになぞらえて、本質を追及していくことを「1番ピンはどこにある」という言い方が社内で良く使われるのですが、これは、根本は何か、本質は何か、という思考であり、本当の根っこの１番ピンを倒せば他の課題も全部カバーできるでしょうという想いからです。結果的に本質に一歩でも二歩でも近づき、最も賢いやりかた、実現手段を実行することで欲しい機能や性能が実現できる、そうあるべきだという思想で日頃使われています。これは開発だけにとどまらず、生産や事務も含め、みな一体となってそのように心がけています。

マツダは、人々にカーライフを通じて人生の輝きをお届けしますというコーポレートビジョンを持っています。それを実現しようとするなら、生産もそれ以外の事務系も販売もサービスも含めた話となりますし、そのために挑戦を続け、独創的な“道（どう）”を極め続けます、としています。そのような想いや行動が製品作りに繋がっています。

4. こだわり極める“魂動”デザイン

マツダは、デザインもこだわりを持って、誰もが乗ってみたいクルマを目指して挑戦しています。

例えば、塗装技術。アクセラを出す時に最初に使ったのが「ソウルレッドプレミアムメタリック」。赤のメタリックカラーですが、実はカラー層と反射層の塗る順番を逆転しています。こうすることで、一層目のアルミフレークに反射した光が、二層目のカラー層で鮮やかに発色し、光の反射がないときには一層目と二層目の赤が重なり、深みのある赤を表現。これらが、豊かな表情の変化を実現します。

また今年モデルチェンジしたCX-5から「ソウルレッドクリスタルメタリック」という色に変えました。これは彩度と深みを一段と増したもので、本当にこだわって作っています。ちなみに数年前から広島東洋カープのヘルメットもソウルレッド風の色に変わっています。

5. 先進安全技術への取り組み

マツダの自動運転技術の考え方に繋がりますが、まず人が運転することが中心。そうであるはずですよねという考え方があり、ADAS機能では「マツダ・プロアクティブ・セーフティ」というポリシーを安全性能について持っています。クルマは人間が運転するものであると人中心を大事にしていますので、いきなり介入するのではなく、まずは認知を支援する、それでも危ないときは回避、サポートするという段階を踏んだ支援を考えています。

更に自動運転技術に関しては「マツダ・コ・パイロット・コンセプト」という考え方を持っています。目的地をセットすれば、あとは自動で運んでくれるというのは、マツダが目指しているクルマの使われ方や自動運転の姿ではありません。

人間はミスをするところまでは、現状のADAS機能でカバーしますが、本当に不測の事態が起きた場合、特に今回「マツダ・コ・パイロット・コンセプト」で考えていることは、人間が更にミスをする予兆、例えばドライバーの体調が悪くなったら退避させるというようなことです。それが、おそらく根本のゴールではと考えています。

あくまで、主体は人であり、走る歓びを堪能し移動の自由を満喫していただく。けれども、不測の事態もしくは、更に人に何か異常が発生した場合には、必ずカバーに入るという安心をセットでお届けできるようにきちんと動くものを作っていくということです。

全てを機械頼みになるようなことは避けるというのは一つあります。しかし、機械のサポート無くしてコントロールすることができなくなった時には、万全の責任が負えるように、きちんと動くものを作るということが重要です。

お客様の特性に合わせてサポート方法も程度も変えて支援できるようにすることが先であり、相当そのことを考えて配慮しなくてはならないと思いますね。それを“人間中心開発”と言います。とにかく人の特性に合ったものを作るのだというのが根源にあるので、それは予防安全技術でも変わらず同じ考え方です。予防安全だからこれでいいとか、そのようなことは一切許されません。

クルマというのはもともと個人の自由な移動手段です。環境性能、安全性能もありますが、マツダが狙っていることが何かというと、やはり、クルマを運転するという行為そのものが人を元気にするというのがあります。

日本は高齢化社会という面で世界のトップランナーになっていますが、そういう社会に対しどこまでもフルオートにしていくということが、その社会の課題を解決する手段とは考えていません。

どうしてもできないことはカバーしなくてはなりませんが、そうならないようにするのが根っこでしょという考えがあります。クルマを運転することそのもので人は活性化するということも研究でわかってきていますので、人中心で運転することで、元気なお年寄りが増えていく、シルバー社会ではなくプラチナ社会を目指しませんか？というのがマツダの考え方です。クルマを運転して自分の意思で自分の行きたい所に行くということ自体が、人間そのものを活性化する、元気で居続けることができると私どもは考えているので、その時に必要な技術とは何かという考え方をしています。

里山経済、里山資本主義という考え方がありますが、田舎で軽トラを運転しているようなおじいちゃんは都会の方よりも比較的お元気だというデータもあります。単に寿命を伸ばすというのではなく、元気で長生きしてもらうという社会に貢献したいです。

もうひとつは、何らかの先天的な病気や、怪我などでハンデがある方にとっても、クルマで行動範囲が広がり、自分の意思で自由に移動できるということで前向きに人生を謳歌していただきたいと思うので、そういう方にとってのクルマとは？ということも考えています。そういった方にとってのクルマでも多少ミスが多いということがあった時には、必ずそこは絶対にクルマが踏ん張って安全と安心をお届けしていきたいという考えがあります。

6. ハッピーな社会を目指したクルマ作り

最近、社会貢献を考えろとよく言われます。まずお客様への提供価値を上げることを集中して考えなさい、いきなりビジネスの話をしないようにということです。一方でモノづくりそのものは、やはり究極まで突き詰めることを求められます。提供価値を最大に考えた上でモノづくりを極め、無駄なことをしなければ利益は必然的に付いてくるという考えです。これは両立できるものだと思います。コストがかかり過ぎのモノは、モノづくりが上手く行っていないからだと。挑戦と制約、そのような中でやりがいがあると思って取り組むか、制約が多いなあとネガティブに思って取り組むか、それは考え方、受け止め方次第で仕事が楽しいものになるか苦しいものになるかだと思います。だけど「本当にみんながハッピーになるにはどう考えるべきですか？」というのをマツダは常に問いかけているのです。

7. AI・機械学習の活用について

様々なユーザに対応していくという意味で、最近注目されている AI や機械学習ではありますが、その活用については、今、研究中です。エンジニアとしては、クルマを動かすのが完全に AI だけでいいのかということは疑問に思っています。まだ証明や検証をしきれていないものが残っているものを製品として出して良いのかと。けれども、その前のなんらかの状態といいますか、ドライバーの様子を探るようなことにはおそらく有効でしょうし、そのあたりは備えなければいけないだろうと思っているので、現時点では研究中です。ADAS にせよ自動運転にせよ、認知、判断、制御という流れがありますが、その中の“判断”部分でも、クルマでは安全に関わる部分、そうでない部分に分けて考える必要があると思っています。そういう面では CATS 社が研究しているルールベース AI がひとつの解になればと注目しています。そのようなわけで、今は様々な考え方を収集している状況です。

先程、魂動デザインの話でも触れましたが、一部の高級車だけが優れた安全性を持っているのではなく全ての車種に搭載し、全てのお客様に等しく安全性を提供したいと思っています。それができて初めて理想的なことができると思うのです。そうでない状態で、例えばその安全性能を積んでいない車種で、たまたま出張に行ったことで事故を起こしましたというのは、間違ってもあってはいけない話だと考えています。そして、できる限りクルマだけで自立した安全性を確立させたいです。Wi-Fi やスマホのような外部のインフラを使う場合は、あくまでより便利になるという使われ方です。電波が届かなかったから事故が起こりましたとか、電池が切れていたから事故が起こりましたという言い訳ができないので、クルマに積んでいるもので極限まで安全に乗ってもらう技術を極めた上で、更にその安全を高めるために外部インフラを使うという自立した安全性が重要と考えています。

8. モデルベース開発の取り組み

SKYACTIV-G エンジン開発の時は、All new のエンジンを開発する今回が全面的にモデルベース開発（以下、MBD）を始めるラストチャンスという位の想いで取り組みました。効果を疑問視する人もいました。なぜモデルベースなのか、それは何ができるのか、どんな効果があるのかと、なかなか必要性が伝わらない中でしたが、他の方法は無いと考えて進めていきました。結論から言うとその当時、疑問視していた人たちが最近は、MBD でと言うようになっていますし、私たちは設計であり、実験の人が疑問視していたりしたのですが、何年か経った時に実験の人の中で、「設計の連中は変わったね。俺らも頑張らないといかん。」と言ってくれる人が出てきました。

エンジン制御というのはクルマの中でも非常に長く取り組んでいる類なので、積み重ねを繰り返してきたものが多かったのに対し、ここで一新しましょうということ自体がすごくハードル高いことだったのです。MBD を進めるにあたり、特に力になってくれたのは若手でした。またベテランでも長く業務に携わっている中で問題意識を持っている人もいて、その人たちを中心に MBD への移行を進めていき、結果として従来方式をバックアップで用意するようなことをせずにやり切ることができました。

当時疑問視した方々も今では MBD を推進していますし、MBD を実施したことで設計部門が良くなったという評価にもなりました。

それまでの現場では、実験部門で実物を見ている経験者の方が知見がありまた説得力があるため、設計部門の人間が議論がしにくい状況がありましたが、近頃では実験部門が設計部門の意見に納得させられることが多くなったという声も聞こえてきます。それは、MBD の利点であるシミュレーションで設計者が説得力を持たせられるようになったことが大きいです。

また、設計部門の技術者が活き活きと元気に仕事ができるようになったという効果もありました。実際、私たちの部門は SKYACTIV-G エンジンの制御開発を行い一周終わったあたりで職場のヘルスチェックをする場がありまして、その結果の点は相当良くなっていました。

今の電子制御でも MBD は外せません。物になって動いていくところは分かりませんというのでは設計という役割は果たせません。物になり動いていくシミュレーション技術が自分たちの手の内にあることによって、きちんと考えるべきことを考えた設計書ができるということは、かなり大切なことだと思っています。

モデルという言葉が示す通り、本質とは何ぞやに繋がるのですが、全てをモデル化するのではなく本質部分をモデル化することに意義があると思います。全てモデル化してしまうと実機を作っているのと同じになってしまいますから、何でもモデル化すれば良いというものではないですね。ほどよい規模のチームのリーダーが引っ張って実施して結果を出すことが重要です。MBD は人間の考え方を癖づけるという意味でも非常に大切な手法ではないかなと思いますね。

以前は、実験部門と設計部門の間に相容れない

やりとりがありましたが、今は減っていると思います。仕事の順番として少し勘違いすると設計したものを実験するということになってしまいがちなんです。本当はそうではなくて、一番最初に実機を相手に物のからくりを見極めるという役、一番先頭は実験の方なんです。それがあった上で、次のステップとしてモデル化するということなので、実験に関わる方がモデリングするというのが一番美しい形だと思います。

そういう意味で言うと最初はなかなかそのようにいきませんでしたが、最近は実験の方も自分たちはこんなモデルベースをしたい、こんなモデルを作っていくんだと、自分達自身で出してくる人も出てきましたので、そこはすごく良くなったのではないかなと思います。

一方、設計はまた一層勉強しないといけないと思っています。つい最近まで、モデルを手にして議論し納得させていたのが、その根っこを相手に一段と見られるようになって（笑）。まあそれで切磋琢磨すれば良いのではないかなと思いますけどね。設計は設計でいろんな制約条件とか動きに値するので、それを盛り込んだ上で“実現すべき解はこうだ”とまたモデルで見せ納得してもらうようにディスカッションしていくのがすごく大事だと思います。

先進安全の方では、当然ながらそこの制御はモデルで描いたりしていますが、もっともっと、例えば普通の制御系でいうとプラントモデルですね。走行している環境や対象物が人であったりするのでセンサは非常に高度な物を使うわけです。ADAS のセンサというのは、レーダとかカメラの機能があり、しかもその中で検出したものを更に結構な処理をして、最終的にクルマはこの辺にとか人はこの辺に、というような高度な処理をしています。そのようなところをモデルベース化していきたいなと思っています。

第 22 回 ZIPC ユーザーズカンファレンス講演
（2017年9月8日（金）新横浜国際ホテルマナーハウス南館にて）

9. MBD 推進に向けツールを活用しながら目指したいこと

機能安全以前の問題として構成管理や変更管理の記録があいまいな状態というのは論外ですので、エンジン制御に携わっていた時に、初めのソフトウェア向けの構成管理ツールを導入しました。が、ツールというのは買ってきてポンと与えられて使えるものではなく、定着まで 2、3 年かかりました。

導入した ZIPC TERAS のようなトレーサビリティツールも業務で回してみたところで、想定以上の効果、実感が出てくると考えています。それには CATS 社の支援にも大いに期待しています。

導入したトレーサビリティ環境である ZIPC TERAS の活用としてですが、ADAS 周りですとシステム範囲定義書から機能定義書、ハザード分析、目標定義書辺りまではトレースを張れているところまで来ています。続けて制御仕様書からサブシステム、モデルまでのトレースを確保していくことが次の段階です。これについては、人と機械の境目になるところで勘違いが入りやすいということがあり、その勘違いが起こったところの記録も残さないといけないと考えています。本当に間違いが起こり易そうなところは良く見ているものですが、そうじゃない部分、言ってはいけない言葉ですが、つい「想定外」という言葉を発してしまいそうな部分のトレースを、なるべく楽に検証ができるようにして「想定外」という言葉を発しないようにしていきたいです。なるべく上流でのモデリングをしていきたいし、機能安全対象でない個所にも広げていきたいと思っていて、それもなるべく楽に行いたいです。それをするためにツール化やモデル化が重要だと思います。

制御ソフトウェアの開発でいうとモデルからコード自動生成をするという劇的な進化でヒューマンエラーを大幅に排除できるツール環境が登場しましたが、もう一歩進んで人間が考えていることをそのまま仕様書や制御モデルに落とすことができるとか、人と機械の境界線を更に上流に持っていきたいという思いはあります。ただ、先程の自動運転の話と同じなのですが、本質は人間が考えることなので、その辺りを勘違いせずにツールを活用していきたいです。

自動運転や安全性を開発することとツールを使うことは、人中心という点では似ているのかもしれません。

※本記事は、2017 年 8 月上旬実施しましたインタビュー取材時点の内容です。

マツダ株式会社の最新情報は、次の Web サイトよりご覧ください。

マツダ株式会社ニュースリリース
http://www2.mazda.com/ja/publicity/release/

＜インタビュアー＞
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

宮本貴之
キャッツ株式会社　プロダクト事業本部
第 2 技術 G　GM

中島美穂　今田道宏氏　宮本貴之

組込みシステムにハイブリッドな AI を

キャッツ株式会社　代表取締役社長
九州工業大学大学院 情報工学研究院 客員教授
組込みシステム技術協会　AI 技術研究委員会　委員長
博士（工学）
渡辺　政彦

AI 技術研究委員会立ち上げました

組込みシステム技術協会（JASA）AI 技術研究委員会を立ち上げました。この委員会で何をしたいのかを本稿で紹介します。小生が委員長拝命致しました。副委員長は、国立情報学研究所 情報学プリンシプル研究系の市瀬龍太郎先生と九州工業大学大学院 生命工学研究科の我妻広明先生です。市瀬先生は産業技術総合研究所 人工知能センター、我妻先生は理化学研究所 脳科学総合研究センターの肩書もお持ちです。

委員会のアドバイザーには、自動車 OEM･サプライヤ、大学病院、介護ロボットメーカー、半導体メーカー・商社、SIer、産総研などの方々にお集まり頂けました。

AI 技術研究委員会でやりたいこと

AI 技術研究委員会でやりたいことは、組込みハイブリッド AI を実現することです。いきなり、「組込みハイブリッド AI」と言われても「？」ですよね。「組込みハイブリッド」のイメージをこれからお伝えします。

データ駆動型 AI・理論知識型 AI のハイブリッド

データ駆動型 AI は､ディープラーニングのようなデータに基づき統計･確率的な手法を用いた AI です。理論知識型 AI は、人が持つ知識をモデル等で表現する AI です。この 2 種類の AI を繋げることがハイブリッド AI です。こうした試みは産総研 人工知能センターで、すでに取り組まれています［1］。辻井研究センター長は、「人に寄り添うしなやかな」人工知能とは、大量データに基づく「人間を超える」人工知能と、人間の知能をモデルとする「人間に迫る」人工知能の技術を融合させることにより、人間と協働できる、人間に理解でき、人間が共同できる人工知能を実現できると述べています。

このようなハイブリッド AI を組込みシステムでどのように実現するかを研究する場が JASA AI 技術研究委員会です。人間の脳に右脳・左脳があり、脳梁が右半球と左半球を繋いでいます。データ駆動型 AI と理論知識型 AI を接続し、組込みシステムに埋め込むイメージが図 1 になります。

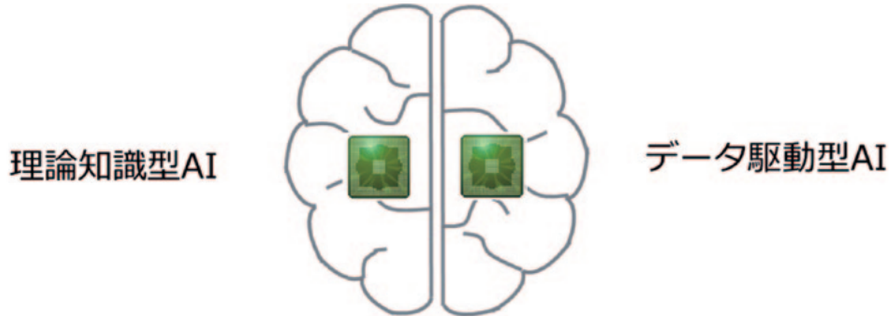


図1　組込みハイブリッドAIのイメージ

まだあるハイブリッドの種類

ハイブリッドを技術融合と定義すると、ハイブリッドの組合せはデータ駆動型 AI と理論知識型 AI だけではなく、他にもあります。図 2 に示す RASMUSSEN の熟練ヒューマンオペレータレベル SRK モデルは、スキルベース振舞（S）、ルールベース振舞（R）、知識ベース振舞（K）の 3 層に分かれています［2］。それぞれの層の間や同一層での技術融合があります。どんな組み合わせがあるかは、この後に順次紹介します。

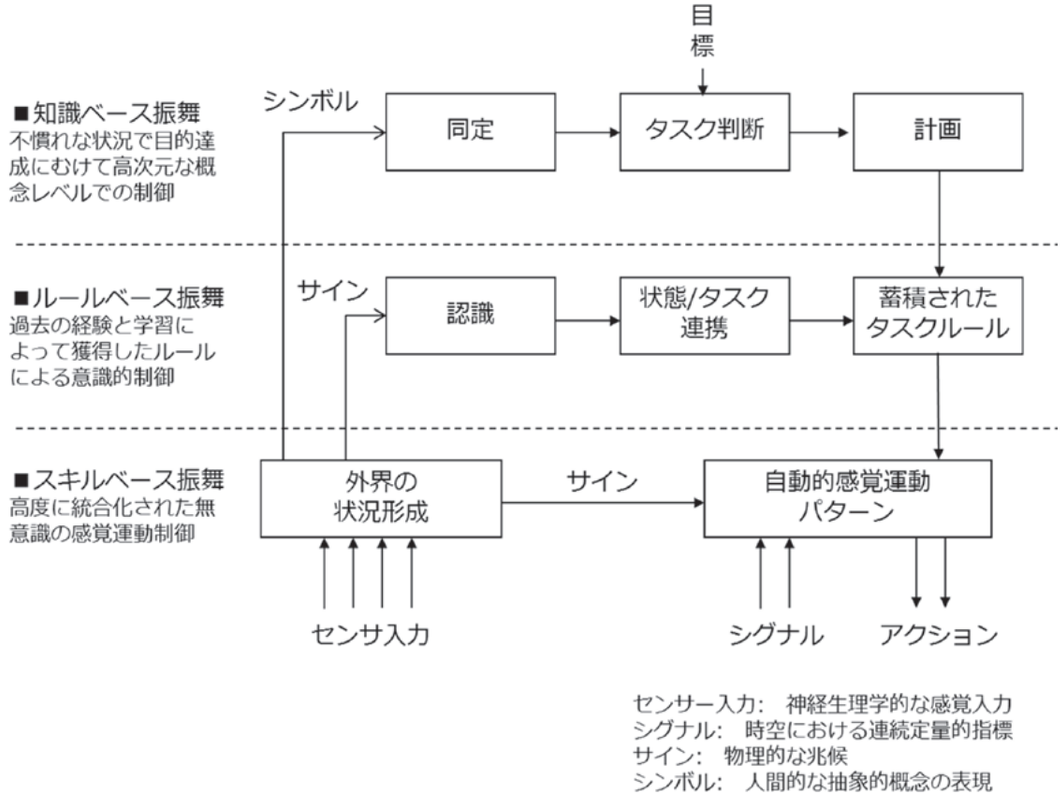


図2　SRKモデル

制御工学・データ駆動型 AI のハイブリッド

スキルベース振舞層（図２）の「外界の状況形成」にデータ駆動型 AI であるディープラーニングが用いられ、「自動的感覚運動パターン」に制御工学が用いられます。スキルベース振舞は、人間で言うところの感覚領域で、一般的に右脳の領域です。この感覚的な右脳領域の AI 化に、ディープラーニングや制御工学のように計算や理論によって開発されるのは何とも面白いですね。ところで､右脳派･左脳派といった区別は、専門家の見地では都市伝説レベルなんだそうだ。

目を手に入れた

カメラからの映像をディープラーニングで、特定の物体として認識できるようになりました［3］。物体の位置はミリ波レーダーによって分かります。これらによってコンピュータは目を手に入れました（図 3）。

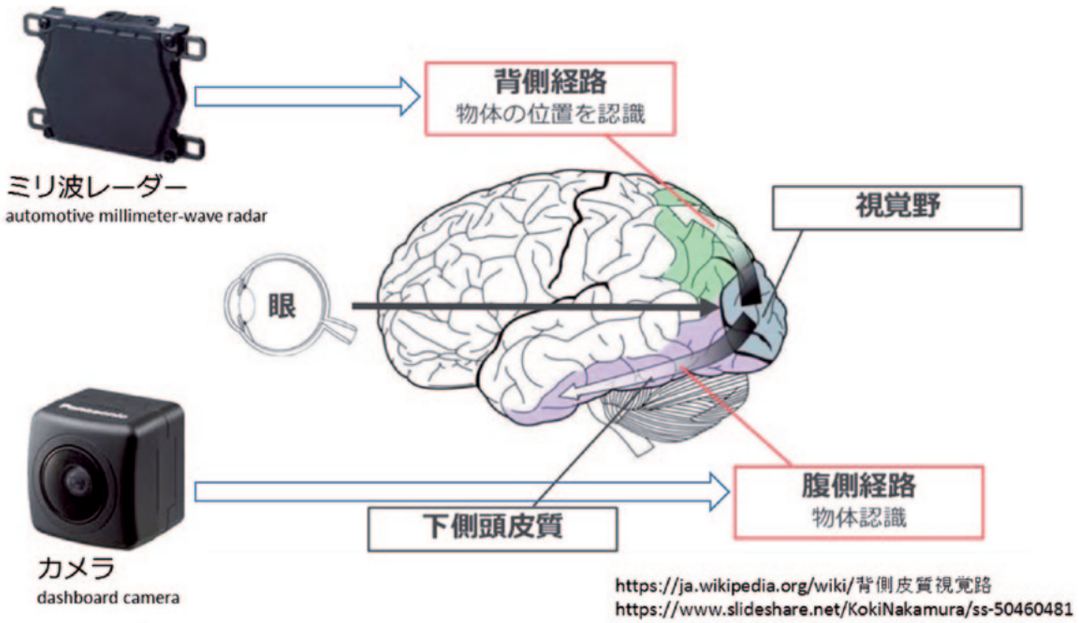


図3　物体認識と物体の位置の認識

物体認識できる

カメラの映像からそれが何かを判断することは、従来とても難しいことでした。ディープラーニングはこれをブレークスルーする技術として注目されています。図４にディープラーニングの仕組みをざっくりと示します［4］。大量のボールの画像データを与えることで、サッカーボールとしての特徴点を抽出し、サッカーボールを判定する学習モデルが生成されます。こうして手に入れた目が「外界状況形成」の一部を担っています。

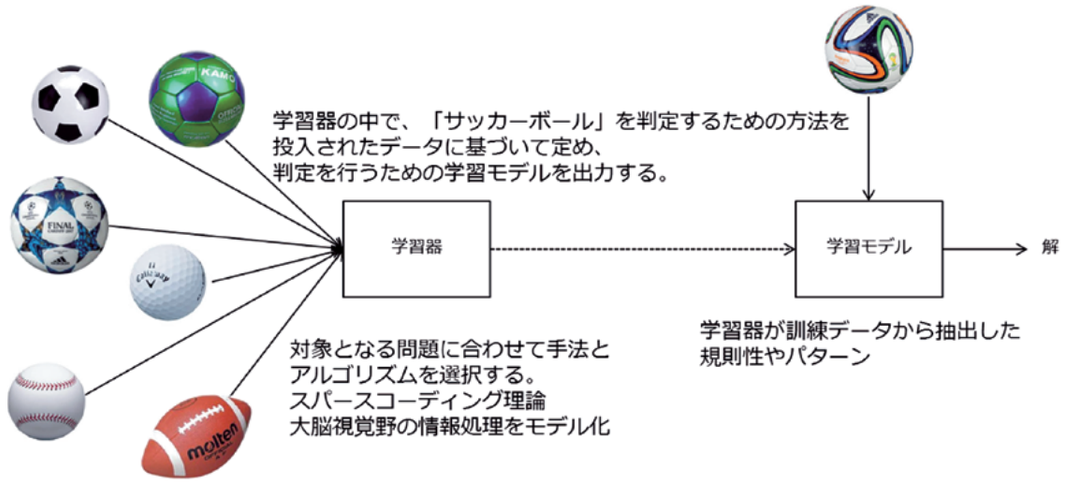


図4　サッカーボールを認識する

制御工学の成果をパッケージ化

ROS（Robot Operating System）は、ロボット・アプリケーション作成を支援するライブラリとツールを提供する OSS（オープンソースソフトウェア）です。OS ではありません。ROS が提供するライブラリとツールを使って、ソフトウェア技術者でもロボットを簡単に操作することができるようになりました。NAO（図５）やペッパーにも ROS が使われています。「自動的感覚運動パターン」をスクラッチ開発するのではなく、ROS パッケージを再利用できる時代になったんですね。

図5　ヒューマノイドロボット　NAO

ROS パッケージ

ROS パッケージには、図６に示すようなアルゴリズムが用意されています。

図6　ROSパッケージ

障害物を避けて目的地へ到着

ROS パッケージを使うと何ができるのでしょう。何と、ロボットが障害物を避けて目的地へ到着するためのアルゴリズムを再利用できちゃいます(図７)。ダイクストラ法により目的地の最短ルートをグローバル（大域的）に決めることができます。走行中にカメラがとらえた物体がディープラーニングによってサッカーボールだと認識したとします。このままだとロボットの進行にじゃまになるので、ダイナミックウインドウアプローチで、現在の速度に基づき、実行可能な複数のローカル（局所的）パス候補を生成し、コスト（距離や時間）の少ないパスを選択することができます。

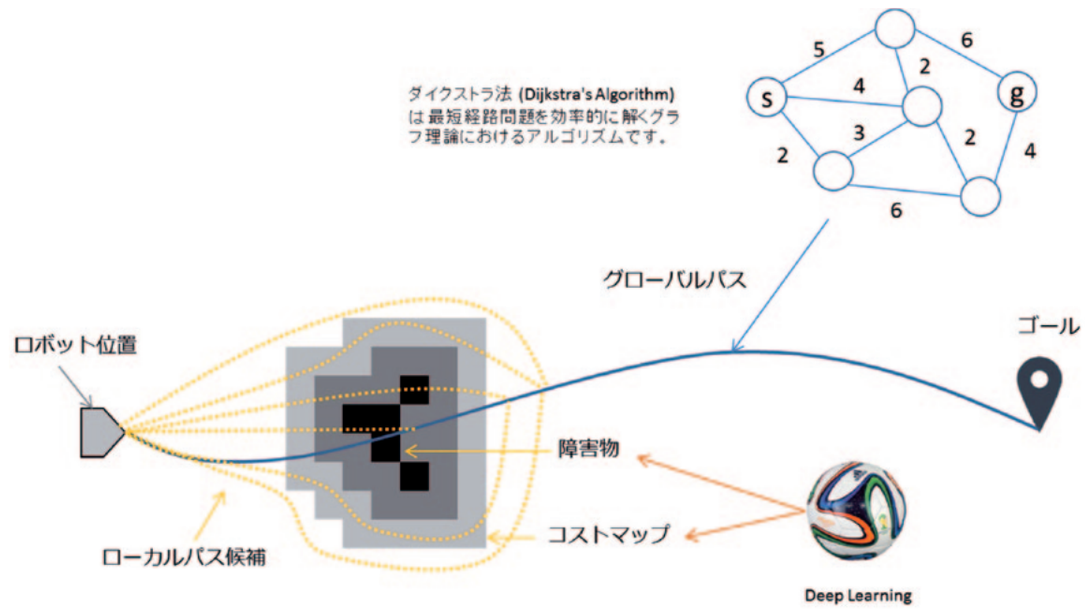


図7　ROSパッケージでできること

自動車の自律走行もできてしまう

Autoware という、Linux と ROS をベースとした自動運転システム用オープンソースソフトウェアがあります。経路計画（Path Planning）、信号機の信号灯認識（Traffic Light Recognition）、レーン検出（Lane Detection）、そして物体追跡（Object Tracking）といった自律走行するための基本機能が提供されています（図 8）。

Autoware目律走行するための基本機能

Functions
- 3D Localization
- 3D Mapping
- Path Planning
- Path Following
- Accel/Brake/Steering Control
- Data Logging
- Car/Pedestrian/Object Detection
- Traffic Signal Detection
- Traffic Light Recognition
- Lane Detection
- Object Tracking
- Sensor Calibration
- Sensor Fusion
- Cloud-oriented Maps
- Connected Automation
- Smartphone Navigation
- Software Simulation
- Virtual Reality

License
New BSD License

Spec Recommendation
Number of CPU cores: 8
RAM size: 32GB
Storage size: 30GB

Requirements
ROS indigo (Ubuntu 14.04) or
ROS jade (Ubuntu 15.04) or
ROS kinetic (Ubuntu 16.04)
OpenCV 2.4.10 or higher
Qt 5.2.1 or higher
CUDA(Optional)
FlyCapture2 (Optional)
Armadillo (Optional)

https://github.com/CPFL/Autoware/

図8　Autoware

それでも、まだ足りないもの

サッカーボールを認識して、避けることが出来ても、まだ足りません。それは人間の経験や知識からの推論です。サッカーボールが家の路地から転がってきたら、熟練ドライバであれば、ボールを避けるだけでなく、その後に、子供が飛び出てくるのではと用心します（図９）。

図9　子供の飛び出し注意

知識をどのように計算機で表現し、扱うか？

決定表モデルにより知識を表現し、計算機で取り扱うことができます。ISO5086（JIS X0125）では、「決定表とは、問題の記述において起こり得るすべての条件と、それに対して実行すべき動作とを組み合わせた表」であると定義しています。書き方はいたってシンプルで、上段に条件、下段に動作を記述します（図１０）。

条件記述部	条件指定部
動作記述部	動作指定部

図10　決定表の書式

サッカーボール子供飛び出し注意の知識をどう書くか

サッカーボール子供飛び出し注意の決定表モデルを図１１に示します。

サッカーボールは物体認識モジュールで認識します。ここでディープラーニングが使われています。物体認識モジュールは様々な認識を行い、その結果を複合イベント処理（CEP）に知らせます。CEP では、現状を判断して必要な情報（fact と呼ぶ）をルールエンジンに通知します。この例では、自動車が走行状態または走行開始状態のときにサッカーボールが検出されたことを通知します。発行された fact はルールエンジンのワーキングメモリに格納されます。

この例ではサッカーボールでも、バスケットボールでも、ラグビーボールでもとにかく何でもボールを発見したら、止まろうとするルールにしています。そこで、サッカーボールルールからボールルールを呼び出しています。

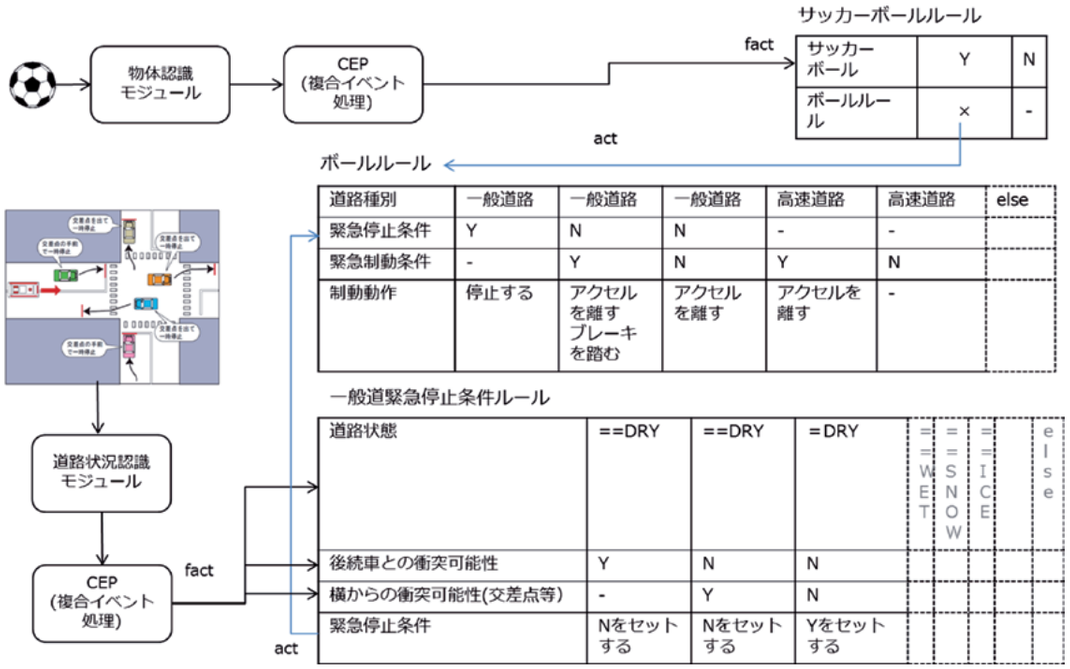


図11　サッカーボール子供飛び出し注意の決定表モデル

一般道と高速道、道路状況で挙動を変える

ボールを発見して、どのような動作をするかを道路の種別で変えています。高速道路で子供がボール遊びをしているとは考えにくく、逆に急停止は危険なので、ボールを発見しても停止しないルールにしています。ブレーキをかけても良い状況のときに、アクセルを緩める動作をすることを高速道路のルールにしています。

一般道では、緊急停止ができる状況であれば、停止します。緊急停止できない状況であれば、ブレーキをかけられる状況かを確認します。ブレーキをかけられる状況でなくとも、一般道ではボールを発見するとアクセルを緩めます。

一般道での緊急停止できるかどうかの判定は一般道緊急停止条件ルールで規定しています。路面状態が乾いていて、後続車との衝突可能性がなく、横からの衝突可能性が無い場合に、緊急停止できるルールとしています。路面が濡れている、雪が積もっている、アイスバーンなどの場合はどうするかを規定します。（注意：図１１は未完成の状態です）

道路状況や衝突可能性に関しては、道路状況認識モジュールが担当します。道路状況認識モジュールがCEPにアウトプットし、CEPが状況によりfactをルールエンジンにアウトプットします。

CEPやルールエンジンの構造、決定表の書式については、ZIPC WATCHERS Vol.19を参考にしてください［5］。

人間が体系化した知識の代表格「道路交通法」

人間が体系化した知識の代表格の法律である「道路交通法」を決定表でモデリングします。例えば、第七条です［6］。

> （信号機の信号等に従う義務）
> 第七条　道路を通行する歩行者又は車両等は、信号機の表示する信号又は警察官等の手信号等（前条第一項後段の場合においては、当該手信号等）に従わなければならない。
> （罰則　第百十九条第一項第一号の二、同条第二項、第百二十一条第一項第一号）

信号機の表示する信号の意味については、第四条４において政令で定めるとあります。

> （公安委員会の交通規制）
> 第四条
> ４　信号機の表示する信号の意味その他信号機について必要な事項は、政令で定める。

その政令は、道路交通法施行令です［7］。道路交通法施行令第二条に信号の意味が規定されています。

> （信号の意味等）
> 第二条　法第四条第四項に規定する信号機の表示する信号の種類及び意味は、次の表に掲げるとおりとし、同表の下欄に掲げる信号の意味は、それぞれ同表の上欄に掲げる信号を表示する信号機に対面する交通について表示されるものとする。

道路交通法施行令の表から自動車に関する部分を抜粋したものが表１になります。歩行者、原動機付自転車、軽車両は除いています。

表1　道路交通法施行令第二条　（信号の意味等）

信号の種類	信号の意味
青色の灯火	二　自動車、原動機付自転車（右折につき原動機付自転車が法第三十四条第五項本文の規定によることとされる交差点を通行する原動機付自転車（以下この表において「多通行帯道路等通行原動機付自転車」という。）を除く。）トロリーバス及び路面電車は、直進し、左折し、又は右折することができること。
黄色の灯火	二　車両及び路面電車（以下この表において「車両等」という。）は、停止位置をこえて進行してはならないこと。ただし、黄色の灯火の信号が表示された時において当該停止位置に近接しているため安全に停止することができない場合を除く。
赤色の灯火	二　車両等は、停止位置を越えて進行してはならないこと。 三　交差点において既に左折している車両等は、そのまま進行することができること。 四　交差点において既に右折している車両等（多通行帯道路等通行原動機付自転車及び軽車両を除く。）は、そのまま進行することができること。この場合において、当該車両等は、青色の灯火により進行することができることとされている車両等の進行妨害をしてはならない。
青色の灯火の矢印	車両は、黄色の灯火又は赤色の灯火の信号にかかわらず、矢印の方向に進行することができること。この場合において、交差点において右折する多通行帯道路等通行原動機付自転車及び軽車両は、直進する軽車両とみなす。
黄色の灯火の点滅	歩行者及び車両等は、他の交通に注意して進行することができること。
赤色の灯火の点滅	一　歩行者は、他の交通に注意して進行することができること。 二　車両等は、停止位置において一時停止しなければならないこと。

備考　この表において「停止位置」とは、次に掲げる位置（道路標識等による停止線が設けられているときは、その停止線の直前）をいう。
一　交差点（交差点の直近に横断歩道等がある場合においては、その横断歩道等の外側までの道路の部分を含む。以下この表において同じ。）の手前の場所にあつては、交差点の直前
二　交差点以外の場所で横断歩道等又は踏切がある場所にあつては、横断歩道等又は踏切の直前
三　交差点以外の場所で横断歩道、自転車横断帯及び踏切がない場所にあつては、信号機の直前

道路交通法・道路交通法施行令を決定表でモデリングする

信号灯ルール、黄色停止ルール、停止ルールの３つのルールの決定表を作成します。

信号灯ルールでは fact（条件）に「信号灯条件」、「矢印灯条件」、そして「矢印方向」を定義します。act（動作）は、各種の動作ルールを呼び出します。図１２では紙面の都合で「黄色停止ルール」と「停止ルール」の２つのルールのみ掲載しています。

信号灯ルールは８種類のルールを呼び出します。８種類のルールとは「通常通行」「黄色停止」「停止」「矢印右折通行」「通常左折」「U ターン通行」「注意通行」「一時停止通行」のルールです。

黄色停止ルールでは、安全停止条件が真、つまり成立すれば、通常停止ルールを呼び出します。偽、成立しなければ、安全に停止できない場合として、通常通行ルールを実行します。

停止ルールでは、交差点で左折中はそのまま左折を行い、右折中は直進車を注意しながら右折を行います。

信号灯ルール

信号灯条件	青色	黄色	赤色	黄色 ∨ 赤色	黄色 ∨ 赤色	黄色 ∨ 赤色	黄色 ∨ 赤色	黄色点滅	赤色点滅	else
矢印灯条件	-	N	N	青色	青色	青色	青色	-	-	
矢印方向	-	-	-	右	左	直進	Uターン	-	-	
通行動作	通常通行ルール	黄色停止ルール	停止ルール	矢印右折通行ルール	通常左折ルール（※1）	通常通行ルール	Uターン通行ルール	注意通行ルール	一時停止通行ルール	ログ（※2）

黄色停止ルール

安全停止条件	Y	N
停止動作	通常停止ルール	通常通行ルール

停止ルール

交差点通過条件	N	Y	Y
交差点通過方向	-	左折	右折
通行動作	通常停止ルール	通常左折通行ルール	注意右折通行ルール

（※1)青色の灯火の左矢印の場合はどうするかは道路交通法施行令には明記されてないが、恐らく、左折があれば通常左折で良いと判断するルールを明確にしている。
（※2)elseの判断は起こりえないが、起こった場合の状況をログする。（例えば信号機故障による青色点滅などの想定外）

図12　道路交通法･施行令決定表モデル

われわれは法令に従って社会活動を行っている

『我々の社会は、非常に多数の相互に関連した法律や法令により規定されている。それらは社会の組織や構造、目標や目的を記述すると同時に、組織における活動や手続きを定めており、われわれはその法令に従って社会活動を行っている。したがって、法令が適切に作られており、それが社会のなかで正しく運用されていることは、社会の安心性の基本である。これは、社会というシステムを考えたとき、法令がこのシステムの仕様の役割を果たしていることを意味している。したがって、その仕様が適切に書かれており、それが実際に正しく運用されていることは、我々が安心で質の高い社会生活を送る上の基本的要件ということになる。』

（COE Research Monograph Series, Vol. 2 : 法令工学の提案［8］からの抜粋）

ここまでできること

データ駆動型 AI ディープラーニングによる目、制御工学による自律制御パッケージによる足は、RASMUSSEN の熟練ヒューマンオペレータレベル SRK モデル（図２）のスキルベース振舞（S）レイヤです。理論知識型 AI 決定表モデルによるドライバの知識からサッカーボールで子供飛び出し注意や、道路交通法・道路交通法施行令を順守した走行がルールベース振舞（R）レイヤです。

データ駆動型 AI・理論知識型 AI・制御工学のハイブリッドにより、反射神経的な振舞と熟練知識や法律に基づいた振舞ができるようになります（図 13）。

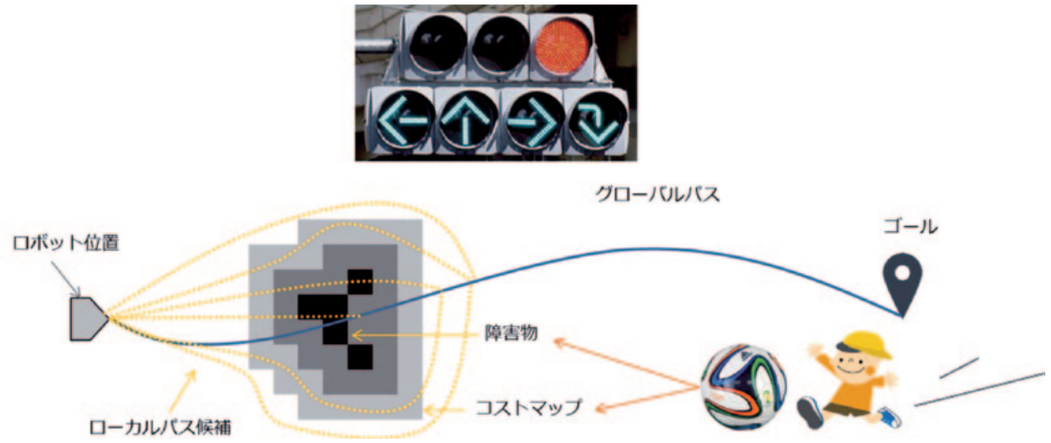


図13　データ駆動型AI･理論知識型AI･制御工学のハイブリッドでできること

知識を獲得、定義することはムズイ

「道にボールが転がってきたら子供が飛び出てくる」的な知識をどのように獲得、表現、取り扱えば良いのでしょうか。解決の鍵は「オントロジー工学」[9] と「意味ネットワーク」[10] にありそうです（図 14）。

『元来の意味でのオントロジーとは、世界に存在するものをカテゴリーに分けカテゴリー同士にどのような関係があるかを理解する哲学の一分野である。オントロジーの考察を経て、我々は基本概念や概念間の関係を取り扱う土台を得る。情報システムの構築に際しても、対象世界をモデル化するためにオントロジーの考察が必要である。かつてエキスパートシステムが盛んに研究されていた時代では、モデルの再利用性の弱さが知識ベースの問題点の一つとして挙げられていたが、適切なオントロジーはこの点を解決すると期待される。』

（「自然言語処理、オントロジーを用いた知識管理」[11] からの抜粋）

意味ネットワークは人間の記憶の一種である意味記憶の構造を表すためのモデルである。 概念の間の意味関係を表現するネットワークである。知識表現でよく利用される。概念を表す節と、概念の意味関係を表す辺からなる、有向グラフまたは無向グラフである。

（ウィキペディアからの引用）

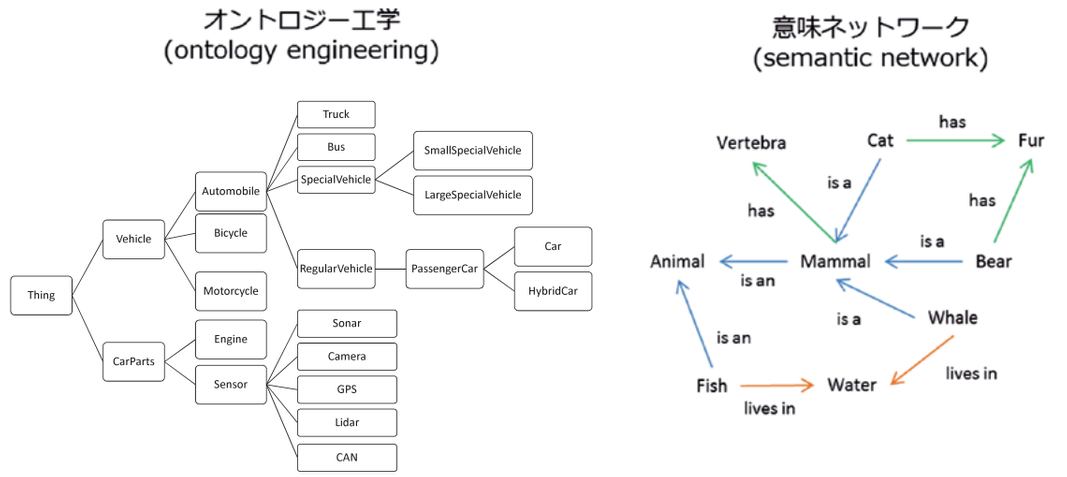


図14　オントロジーと意味ネットワーク

オントロジー、意味ネットワークで書いてみる

ボールのカテゴリー、子供が遊ぶおもちゃのカテゴリー、子供の視野に関するカテゴリー、子供特有要因のカテゴリーをオントロジーで表現し、繋げてみたものが図１５です。

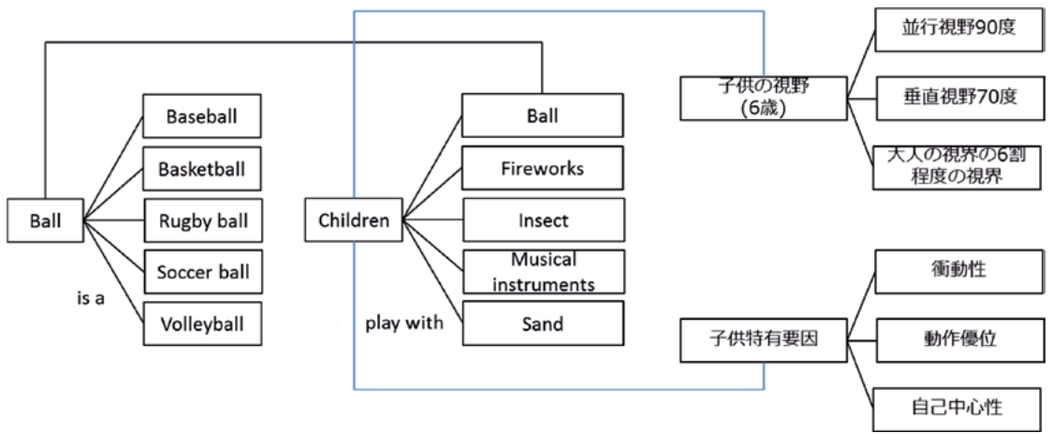


図15　サッカーボールから子供の飛び出しを予測するためのモデル

Web の世界で進む意味処理

Web の世界ではいち早く、知識の表現や処理に関する技術が実現しています（図 16）。前述の「子供の視野」［12］や「子供特有要因」［13］はネット上でググって獲得しました。こうしたネット上の情報に意味を示すタグを埋め込み、繋げる（link する）ことで新しい価値を生み出す試みが Linked Data で、それをオープンな形で公開するデータが Linked Open Data（LOD）です［14］。

現在のワールド・ワイド・ウェブ上のコンテンツは主に HTML で記述されている。HTML では文書構造を伝えることは可能だが、個々の単語の意味をはじめとする詳細な意味を伝えることはできない。これに対し、セマンティック・ウェブは XML によって記述した文書に RDF や OWL を用いてタグを付け加える。この、データの意味を記述したタグが文書の含む意味を形式化し、コンピュータによる自動的な情報の収集や分析へのアプローチが可能となると期待されている。オントロジーを扱う階層までは W3C により標準化されているが、それ以上の階層の開発は難しいため、実現と標準化には長期間掛かると予想されている。また、既存の Web サイトに対するメタデータ付与の作業が必要であるため、Web 全域への普及に関しても長期間掛かると予想されている。

（https://ja.wikipedia.org/wiki/ セマンティック・ウェブからの引用）

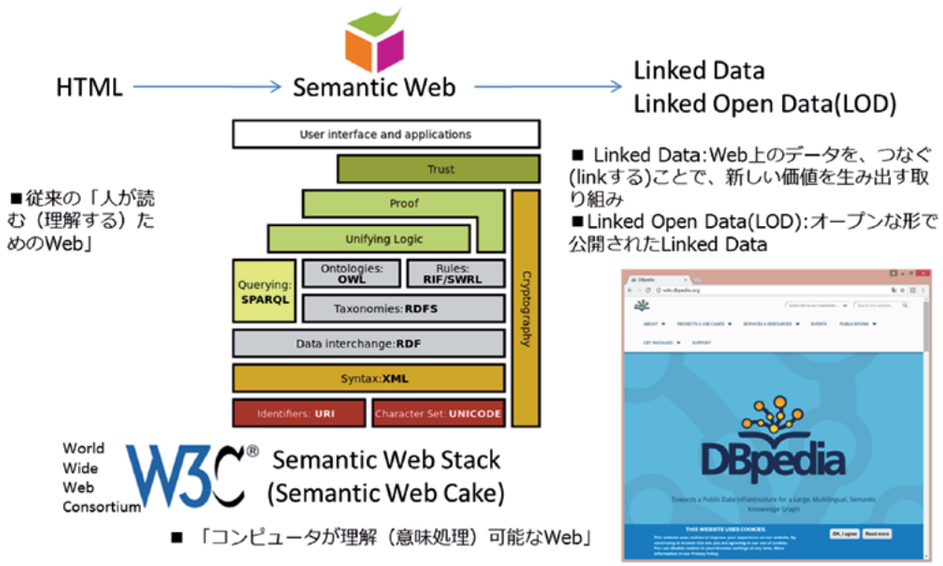


図16　HTML→SemanticWeb→LOD

自動運転にもオントロジー適用が

組込みシステム技術協会（JASA） AI 技術研究委員会　副委員長の国立情報学研究所（NII）情報学プリンシプル研究系の市瀬龍太郎先生と九州工業大学大学院 生命工学研究科の我妻広明先生は、「自動運転用危険予知防止装置へのオントロジー導入の方策と課題」を The 30th Annual Conference of the Japanese Society for Artificial Intelligence, 2016 に論文発表しました。この研究の一部は経済産業省の下、独立行政法人 新エネルギー・産業技術総合開発機構（NEDO）から委託された「次世代ロボット中核技術開発／次世代人工知能技術分野／人間と相互理解できる次世代人工知能技術の研究開発」の支援を受けています。

その課題は

オントロジーによる理論知識型 AI とデータ駆動型 AI の融合技術に関しては、推論速度と様々なドメイン分野への対応が課題です（図 17）。Web 技術である RDF による状況、知識表現と、SPARQL,SWRL による論理検索・推論は、処理が重く、まして、省リソースである組込みシステムで動作させるのはなかなかしんどそうです。

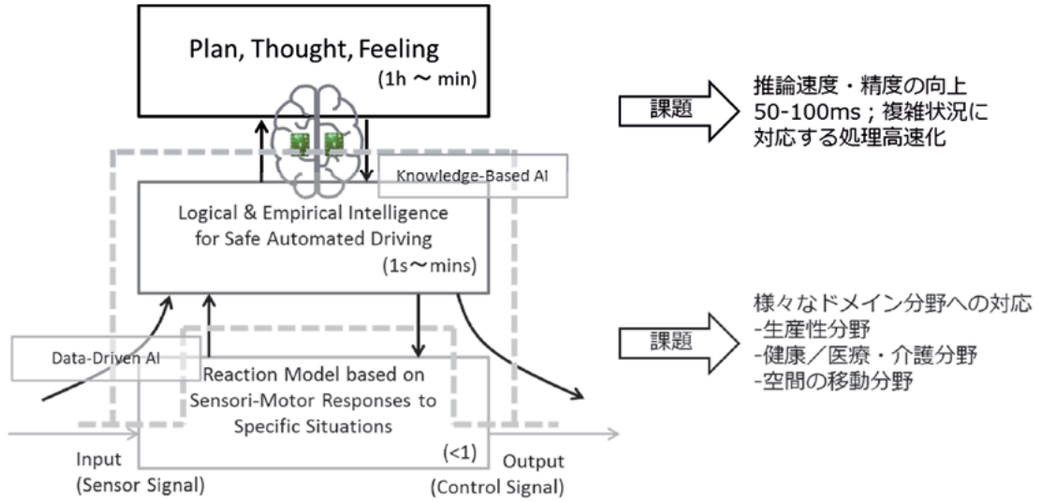


図17　オントロジーによる理論知識型AIとデータ駆動型AIの融合技術と課題

そこで、AI 技術研究委員会の登場だ！

市瀬先生（NII）と我妻先生（九工大）のオントロジーによる理論知識型 AI とデータ駆動型 AI の融合技術は、AI の説明責任を果たすことができる技術です。この技術をよりコンパクトで高速にすることで、様々な分野の組込みシステムに適用できるようにしたいと考えています（図 18）。

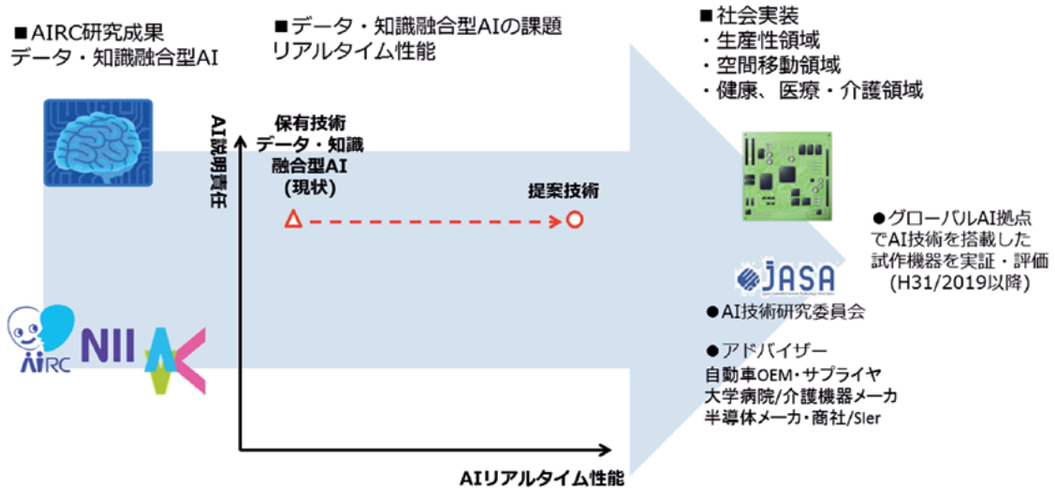


図18　AIリアルタイム性能

オントロジーを決定表に変換すれば、速くなるんじゃね（若者風イントネーション）

オントロジーを決定表に変換し、決定表からC言語コードで実装することで、よりコンパクトで高速な理論知識型AIを実現できるのではないかと考えました。また、オントロジーと決定表モデルを組込みシステムで共通にオープンに使える知識プラットフォームLODEHAI（Linked Open Data Embedded Hybrid AI）の実現に向けて、JASA AI技術研究委員会で検討したいと思います（図19）。

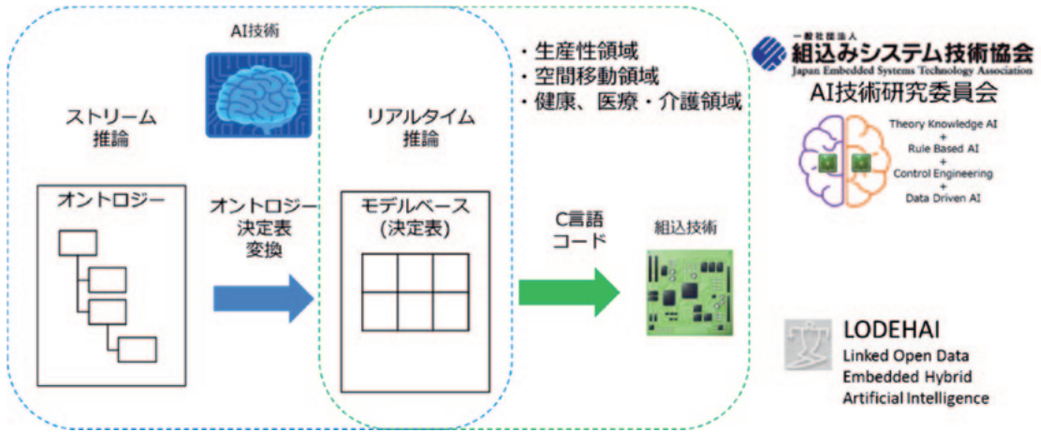


図19　AI技術研究委員会とLODEHAI

お！幸先いいね！

NEDO事業「次世代人工知能・ロボット中核技術開発／次世代人工知能技術分野」（先導研究）に「オントロジー推論のリアルタイム処理を実現する組み込み技術の実現と安全・安心分野への応用」で採択されました［15］。

皆さん、組込みハイブリッドAIを
一緒にやりましょう！

一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association
AI技術研究委員会

★彡参加メンバー募集中彡★
＊詳細はこちら
* 問合せ先⇒ nedo@jasa.or.jp　（JASA 事務局・AI 技術研究委員会担当）

☆ JASA AI 技術研究委員会のホームページ
http://www.jasa.or.jp/TOP/members/ai_technology/

参考文献

［1］「ビッグデータ、人工知能と知識の有効な活用」産業技術総合研究所 人工知能センター
研究センター長　辻井潤一 , Japlo YEAR BOOK 2015
［2］ JENS RASMUSSEN：Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models：
IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. smc-13, No.3, MAY/1983
［3］「人工知能研究のための視覚情報処理」慶応義塾大学 理工学部 情報工学科 中村 晃貴 ,
全脳アーキテクチャ 若手の会 勉強会
［4］「初めてのディープラーニング」武井宏将 , リックテレコム : 2016
［5］「エッジインテリジェントを実現するルールベースシステム
～自動運転システムの場合～」渡辺政彦 , ZIPC WATCHERS Vol.19 : 2016
［6］ http://law.e-gov.go.jp/htmldata/S35/S35HO105.html
［7］ http://www.houko.com/00/02/S35/270.HTM
［8］「COE Research Monograph Series, Vol. 2 : 法令工学の提案」
片山卓也 ; 島津明 ; 東条敏 ; 二木厚吉 ; 緒方和博 ; 有本泰仁 ; 落水浩一郎 ; 早坂良 :
JAIST Press URL
http://www.jaist.ac.jp/library/jaist-press, COEResearch Monograph Series, Vol. 2 :
法令工学の提案 , 片山卓也（編）, 2007
［9］「ストリーム推論」市瀬龍太郎 , 人工知能 30 巻 5 号 (2015)
［10］https://ja.wikipedia.org/wiki/ 意味ネットワーク
［11］「自然言語処理、オントロジーを用いた知識管理」辻井潤一、西村悟史、中田亨 ,
電子情報通信学会誌 Vol.100, No8, 2017
［12］http://www.kurashikihokenya.com/wp-content/uploads/2012/04/120423.pdf
［13］https://www.itarda.or.jp/ws/pdf/h23/14_06kodomo.pdf
［14］「LOD 技術の概要と LinkData.org を用いた LOD 公開」大阪大学産業科学研究所　吉崎晃司
https://www.slideshare.net/KoujiKozaki/20140823kozaki-v2
［15］http://www.nedo.go.jp/news/press/AA5_100814.html

ZIPC WATCHERS Vol.20

コネクティッドカー／自動運転分野における ソフトウェアテクノロジーの動向と取り組み紹介

株式会社 NTT データ　ビジネスソリューション事業本部
次世代技術戦略室　次世代オートモーティブ技術担当　シニア・スペシャリスト
古賀　篤

1. 自動車業界の潮流

近年の自動車業界における主要なキートレンドには、「コネクティッドカー」「電気自動車（EV）」「自動運転」の 3 点が挙げられる。本稿では、まず初めにこれらのトレンドを軸として、自動車業界でいま何が起きているのかに関する大きな潮流を読み解くこととしたい。

1.1 コネクティッドカー

コネクティッドカーの概念は未だ抽象的であり、現状様々なサービスが提唱されているが、基本的な概念としては、「車を起点としてあらゆるデータが複合的につながり、相互に会話することで、車の安全性・快適性・利便性が飛躍的に向上すること」であると言える。

具体的には、車車間通信（V2V）・路車間通信（V2I）・対個人（V2P）・対企業（V2E）・対社会サービス（V2S）など、車を起点につながる情報のバリエーションが多岐に渡り、多様なサービスを実現することが可能になると考えられている。

世界市場におけるコネクティッドカーの販売台数は、現在 2017 年時点で 2,025 万台（29.1％）に対して、2030 年に 8,890 万台（85.5％）、2035 年に 9,990 万台（93.4％）と飛躍的に増加すると予測されている。［1］

また、コネクティッドカー関連のマーケット規模（安全技術・自動運転・コネクティッドといった各軸合計）は、現在 2017 年時点で 525 億ドルに対して、2022 年に 1,559 億ドルと年間平均成長率 24.3％もの伸長率での市場拡大が予想されており、今後数年で市場規模が 3-4 倍に急成長していくとの予測が出ている。［2］

自動車業界の主要キートレンドがもたらす 100 年に 1 度の産業革命を眼前に控え、OEM メーカ各社では IT 企業との異業種連携を積極的に実施し、自社単独での車両開発から IT 企業と連携してのコネクティッドカー開発を指向しつつある。（表 1）

IT 企業としては、これらコネクティッドカーの販売台数・市場規模急増によるスケールインパクトについて、IT インフラ・サービスの開発・提供・運用といった視点で影響を熟慮していく必要がある。

コネクティッドカーにより実現される具体的なサービスは、以下に挙げるように車の走行情報や趣味・趣向に応じた高度な車両サービスが実現されていくと予見されている。［3］

- ドライバーの感情・嗜好性に応じた提案
- 家庭内機器との連動・制御
- 運転特性に応じた自動車保険
- リモート診断・メンテナンスサービス
- 行先提案エージェントサービス

表1　OEM･IT企業間連携

トヨタ・MS	車両データ収集解析に向け「Toyota Connected Inc」を設立（2016年4月）
トヨタ・KDDI	コネクテッドカーのグローバル通信プラットフォームを日米展開（2016年6月）
トヨタ・NTT	「コネクティッドカー」向けICT基盤の研究開発に関する協業を開始（2017年3月）
ホンダ・SB	人工知能(AI)を使った自動車の運転支援システムを共同開発（2016年7月）
日産・MS	コネクテッドカーの開発について提携し、MS社のクラウドサービスを採用（2016年9月）

1.2 電気自動車（EV）

電気自動車（以下 EV）に関する動向把握としてまず触れておきたいのが、昨今の急激な EV シフトに関する風潮についてである。

北米・欧州・中国等の大規模市場における環境対策や規制強化、VW のディーゼル車排ガス不正問題の影響等を背景として、各国政策や OEM メーカの開発・販売戦略において EV シフトが鮮明となっている。

北米では、2018 年に ZEV 規制をカリフォルニア他 10 州で施行し、EV/PHV 開発に拍車がかかる見込みであり、中国も 2019 年の NEV 規制でこれに追随して EV/PHV 普及に向け OEM 規制を実施予定である。また、英仏では 2040 年を目標にガソリン･ディーゼル車販売を禁止、ノルウェーでは 2025 年迄に EV/PHV 完全シフトを宣言している。

OEM 各社の動向としても、欧州メーカを中心に EV 開発へのシフトが加速しており、VW では 2025 年までに EV50 車種以上 2-300 万台、2030 年までに 200 億 € を投資するとしている。ダイムラーでは 2022 年までに EV10 車種を市場投入、BMW も同様に 2025 年までに EV12 車種を投入という状況にある。

EV には航続距離・生産コスト・充電インフラ・中古者価格・給電含むトータルコストといったネガティブ要素も存在する。また、バッテリー技術革新、環境問題対応等を踏まえても、インフラ整備などの未解決課題も多い。このため、全面・一気呵成でのピュア EV 化は進まず、現実的には、ピュア EV はハイエンド車種又は商用などに二極化し、普及車は PHV・レンジエクステンダー等が占めるなど、既存のガソリン・ディーゼルエンジンといった内燃機関も含めて適材適所でのパワートレイン多様化が進むと考えられる。

しかしながら、一度火がついた急激な EV 普及加速シフトは、既に世界的な潮流となっており、現在時点から 2035 年までの約 20 年間において HV・PHV も含めた EV 全体台数は 10 倍から 20 倍にまで拡大していくと予想される。［4］

次に、EV の特徴的な点として、車を構成するハードウェアコンポーネント構成が大きく変化する点が挙げられる。ピュア EV を例にとると、従来の内燃機関で必要とされたエンジン・ガソリンタンク・マフラー・燃料噴射装置等は不要となり、新たにバッテリー･モーター･インバーターといった部品群が必要となる。［4］

このハードウェア構成の変化は、EV 普及に伴う車両コンポーネントのコモディティ化であるとも言える。2015 年の全世界の EV/PHEV 年間販売台数ランキングによると、ピュア EV の発売・普及開始からたったの数年間で Tesla・BYD といった新興 EV メーカの車両販売台数が EV 市場で Top10 入りしており、コンポーネントの簡素化･コモディティ化により自動車業界への参入障壁が大幅に低下した結果であると見て取れる。［5］

1.3 自動運転

自動運転に関しては、2017 年より自動運転レベルの定義が 4 段階から 5 段階に細分化されており、現在時点ではレベル 2、つまり複数の運転操作に関する自動制御機能を実現した車が OEM 各社より市場投入されている状況にある。［6］（表 2）

表2　自動運転レベル定義

レベル		概要	安全運転対応主体
LV0	自動運転化無し	運転者が全ての運転タスクを実施	運転者
LV1	運転支援	システムが前後・左右のいずれかの車両制御に係る運転タスクのサブタスクを実施	運転者
LV2	部分運転自動化	システムが前後・左右の両方の車両制御に係る運転タスクのサブタスクを実施	運転者
LV3	条件付運転自動化	システムが全ての運転タスクを実施（限定領域内） 作動継続が困難な場合の運転者は、システムの介入要求等に対して、適切に応答することが期待される	システム （作動継続が困難な場合は運転者）
LV4	高度運転自動化	システムが全ての運転タスクを実施（限定領域内） 作動継続が困難な場合、利用者が応答することは期待されない	システム
LV5	完全自動運転自動化	システムが全ての運転タスクを実施（限定領域内ではない） 作動継続が困難な場合、利用者が応答することは期待されない	システム

ZIPC WATCHERS Vol.20

また、今後はレベル 4・5 といった完全自動運転に向けた技術開発が加速する見込みである。また、2020 年前後に OEM 各社ともレベルは様々ではあるが自動運転車を市場投入するという発表を行っている。

完全自動運転車の本格的な市場投入時期について様々な発表・予測がある中、世界・販売台数ベースの市場規模については、2025 年に部分自動運転車が 1,390 万台（12%）、完全自動運転車が 60 万台（0.5%）、自動運転車の全体合計で 1,450 万台（13%）という現実的な数値が予測されている。

2035 年時点では部分・完全の比率が均衡化し、部分自動運転車が 1,840 万台（15%）、完全自動運転車が 1,200 万台（10%）、自動運転車の全体合計で 3,040 万台（25%）と予測されており、新車販売の 1 割が完全自動運転車となり順次拡大する世界が想定されている。［7］

一方、NTT データに於いては、群馬大学との共同研究として、レベル 4 相当の自動運転技術（限定された道路での自動運転：域内無人バス・無人トラック・無人タクシー等）について共同研究を推進中である。

群馬大学と NTT データは、次世代モビリティの社会実装研究として、AI 技術やビッグデータ処理技術等の完全自動運転社会に求められる技術要素について共同で研究するとともに、今後、群馬県内の自治体を中心に実証実験を開始し、完全自動運転車の社会実装化を目指していく予定である。

2. コネクティッドカー時代のテクノロジー動向

続いて、コネクティッドカー時代のキーテクノロジーに関して、2017 年度にガートナーが発表している IT トレンド分析ハイプ・サイクル［8］より、期待のピークにある「IoT プラットフォーム」と「AI・機械学習・Deep Learning」、また、これに「ソフトウェア工学」を加えて動向を述べていく。

2.1 IoT プラットフォーム

昨今の IoT ビジネスの活発化により、多様なセンサーデバイスからシステム基盤へのデータ連携、リアル・バッチ両処理でのビッグデータ蓄積、ETL ツール等を介したデータ分析など、つなぐ・ためる・使うといった主要 3 機能を実現するビッグデータに対応した IoT プラットフォームシステム基盤の重要性が増している。

しかし、コネクティッドカー（自動車 IoT）におけるプラットフォームシステム基盤の機能要件を考察すると、一般的な IoT 用途のそれとは異なり、様々な観点での課題対応が必要となってくる。（図 1）

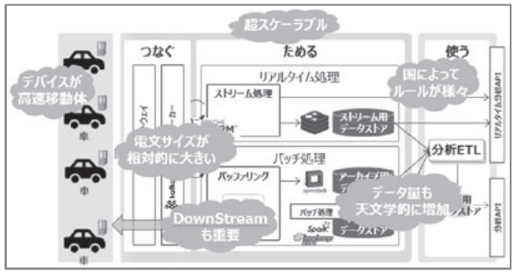


図1　自動車IoTプラットフォーム

具体的な課題を数点ご紹介すると、まずデバイスが高速な移動体であることが挙げられる。車は高速道路では時速約 100km、一般道でも時速 40km-60km といった速度で走行するが、これがコネクティッドカーとなった場合、複数のモバイル基地局の管轄範囲を高速で跨ぎながら移動するため、ハンドオーバー（基地局切替）が頻繁に発生することとなる。これの対応として、瞬間的な通信遮断時においても継続的なアプリケーション動作を保証していく、といったような技術課題対応が必要になる。

次に、コネクティッドカー 1 台単位の通信データ量が多いことが挙げられる。他の IoT においては、固定機器のデバイス状態管理等が目的となるため、1 件あたりの通信データサイズは数K byte 程度と想定されるが、コネクティッドカーの場合は走行履歴管理や ECU 状態管理のため、1 件あたりの通信データサイズが数 10Kbyte ～ 1Mbyte に及ぶことが想定される。

今後自動運転車の普及を視野に入れると、カメラや各種センサーのデータ収集が必要になると考えられ、1 件あたりデータ量が上記の 10 倍～ 100 倍レベルのデータサイズとなりえる。また、2020 年前後より想定されるコネクティッドカーの急増により、そのデータを受けるクラウド側にも非常に高いスケーラビリティが要求されることとなる。コネクティッドカーの IT インフラに求められるスケーラビリティについては、相当な規模かつ柔軟なスケーラビリティを持つ IT インフラが要求されることが理解できるだろう。（図 2）

更には、一般データ保護規制（2018 年に EU で GDPR：General Data Protection Regulation を施行予定）への対応や、国別の通信規制への準拠なども、同時進行で実現していく必要があることも忘れてはならない。

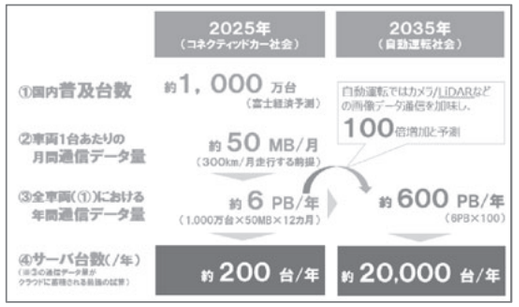


図2　自動車IoTプラットフォームの将来インフラ規模

2.2　人工知能 (AI)・機械学習・Deep Learning

次に、人工知能（以後 AI）に関してだが、現在第三次 AI ブームが到来していると言われている。AI 自体は 1956 年～ 1960 年代の第一次ブームにて原型が生まれ、その後 1980 年～ 1990 年代の第二次ブームを経て、2010 年頃より Deep Learning という機械学習の深層学習という学習手法が一気にブームに火を付けて現在に至っている。

ISLVRC（ImageNet Large Scale Visual Recognition Challenge）という画像認識精度を競うコンテストでは、2012 年頃を境に Deep Learning による画像認識精度が大幅に向上し、2015 年にはエラー率 4-5% と人間の認識率を上回るレベルに到達している。

昨今は Deep Learning を利用可能なネットワークやアプリケーション開発用フレームワーク等が公開され、直近では Deep Learning 専用のチップ開発も進み、自動車分野においても自動運転ソフトウェアやエージェント機能への適用が進み始めている。

進展著しい AI や Deep Learning であるが、これら技術を自動運転に活用していくにあたっては、以下に挙げるような社会課題への対応が必要になる。

① 不完全知覚問題：
完全な環境認識は不可能
② 切り替え問題：
人の運転に切り替える際の対応遅れ
③ 混在問題：
異質な自動運転と人の運転の不整合
④ 賠償責任問題：
事故補償、過失賠償責任が未定
⑤ トロッコ問題：
コンピュータロジックと人の倫理観が相容れない

よく話題となる「⑤トロッコ問題」について具体例を挙げると、自動運転車の前方に人が歩いており避けられない状態の時に、ハンドルを切って自ら壁に激突するか、歩行者を轢いてしまうのか、といった判断を AI に任せなければならない時に、機能としてどうあるべきかという問題であり、社会倫理が絡んだ難題であると言える。

また、Deep Learning の精度が飛躍的に向上する一方で、実用化のためには、Deep Learning が出力する確信度（確率値）の妥当性判断や、精度を維持した状態でのモデル圧縮、精度向上に向けたチューニングの試行錯誤、といった技術課題も発生する。

車載用コンピュータというコンパクト性が求められる領域に対して AI・Deep Learning といった技術を適用していくために、これらの技術課題の解決検討が日進月歩で進められている。

2.3 ソフトウェア開発方法論・管理手法

今後、コネクティッドカー・自動運転車の普及段階を迎えるにあたり、車両構成要素となるハードウェア・ソフトウェアの技術成熟度を鑑みると、ハードウェアよりもソフトウェアの伸びしろが非常に大きく、ソフトウェアの開発ボリュームが飛躍的に増えていくと予想されている。［4］

車載ソフトウェアに関する課題は、その開発ボ

ZIPC WATCHERS Vol.20

リューム増に伴う開発の複雑さに他ならず、その開発方法や管理手法については、更なる高度化が求められるようになる。

このため、NTT データでは、ミッションクリティカルシステムや大規模エンタープライズシステムの開発で培ったプロジェクト管理手法やソフトウェアライフサイクル管理手法、アジャイル開発方法論等を車載ソフトウェアの組込み開発領域に応用・融合させ、開発メソドロジの高度化に取り組み始めている。(図 3)

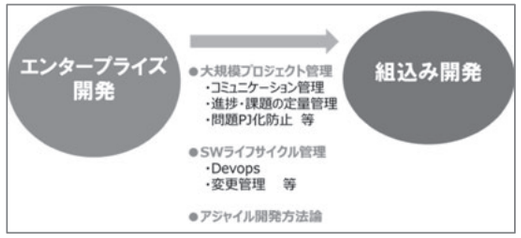


図3　車載ソフトウェア開発･管理手法の高度化

3. CATS 社との先進的な取り組みの紹介

本稿の最後に、NTT データにおけるキャッツ社との先進的な取り組みを以下に 3 点ご紹介する。

3.1 ZIPC TERAS をベースとした ALMSuite・DevOps 対応

1 点目は OEM への導入を積極的に展開中である ZIPC TERAS について、トレーサビリティ観点のみでなくアプリケーションライフサイクルマネジメント（ALM）全体の観点から、更なる機能拡充に取り組んでいる。

機能拡充の着眼点としては、OSS のプロジェクト管理ソフトウェアである REDMINE や課題・プロジェクト追跡ソフトウェアの JIRA など、ALM の要求機能を実現する現存ツール類とのインターフェース連携を取る、という事ではなく、要求・デザイン・コーディング・テスト等の個々の機能の充実・高度化を図っていく方向性で取り組みを進めている。

また、開発領域のみならず、製品企画段階や運用フェーズへの機能拡充・幅だしについても検討を実施中である。(図 4)

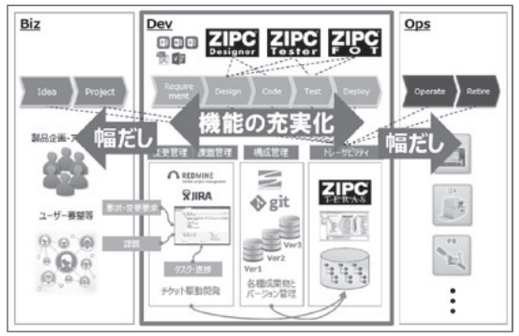


図4　ZIPC TERASをベースとしたALMSuiteとDevOpsへの対応

3.2 グラフ DB を活用した次世代 EPM の研究

次に、定量的なプロジェクト管理手法の EPM（Enterprise Project Management）に関する取り組みとして、EPM ツールへのグラフ DB 活用を検討している。

グラフ DB は、データの高い視認性や柔軟なデータ集計が可能であるため、これを活用してプロジェクト管理における各種要素群の関係性や中心的要素（問題真因）の発見に役立てるべく、実プロジェクトを通じて実用性の検証を進めていく予定である。

3.3 ハイブリッド AI の研究

2.2 項において前述したように、Deep Learning には判断過程がブラックボックス化してしまう点や、レアケースの学習不足といった技術課題が存在する。

これらの課題に対応するため、ルールベースの AI エンジンによって Deep Learning をサポートする「ハイブリッド AI」について NTT データ・キャッツ両社で技術検討・研究を開始している。

参考文献

［1］ 乗用車を中心に普及が進むコネクテッドカーの世界市場を調査 - 富士経済
［2］ Strategy& PwC コネクテッドカーレポート 2016
［3］ 2017 年 7 月 13 日 総務省 Connected Car 社会の実現に向けた研究会「Connected Car 社会の実現に向けて」(概要案)
［4］ 2017 年 3 月 29 日 経済産業省 シリコンバレー D-Lab プロジェクトレポート
［5］ 兵庫三菱自動車販売グループ　世界での EV・PHEV 販売台数データ［2015 年 年間ランキング］
［6］ 2017 年 5 月 30 日 内閣官房 IT 総合戦略室「官民 ITS 構想・ロードマップ 2017（案）」抜粋・追記
［7］ ボストンコンサルティンググループ「自動運転車市場の将来予測」
［8］ ガートナー「日本におけるテクノロジのハイプ・サイクル :2017 年」

ZIPC WATCHERS Vol.20

製品認証における証跡管理ツールに求められる要件

一般財団法人　日本品質保証機構
認証制度開発普及室　主幹　博士（工学）
櫛引　豪

1. はじめに

一般財団法人日本品質保証機構（以下、JQA）は、認証・試験・検査等を実施する日本の第三者機関であり、2017年10月に創業60周年を迎える。製品やサービスの認証事業を通じて、安全な社会を実現する一助となるべく、事業展開を行っている。

近年、IoT（Internet of things）を中心とした技術革新が大幅に進んでおり、様々な分野で新しい製品やサービスが次々に発表されている。ISOやIECに代表される国際規格についても、これまでの経験に基づいた要求事項に加え、新しいタイプの製品の安全性も事前に担保できるような規格作りが求められている。

第三者機関においても、時流に即した社会貢献が求められており、新しい国際規格に基づく認証サービスの開発や技術支援の準備は待ったなしと言える。このような状況を踏まえ、JQAは、2009年から2014年までNEDO「生活支援ロボット実用化プロジェクト」[1]に参画した。JQAでは、このプロジェクトを通じ生活支援ロボットの国際規格ISO 13482の認証スキームを、国立研究開発法人産業技術総合研究所、独立行政法人労働者健康安全機構労働安全衛生総合研究所、一般財団法人日本自動車研究所、国立大学法人名古屋大学等の各専門家のご協力を頂きながら開発した。この認証スキームの下、ISO 13482が発行された2014年からおよそ3年間で、12件の生活支援ロボットの認証を実施［2］している。

本稿では、生活支援ロボットを念頭に置き、製品認証における証跡管理について述べる。また、証跡管理ツールについて、使用者の視点ではなく、むしろ当該ツールの出力を受け取る側の視点から、証跡管理ツールに求められる要件を整理する。

以下では、ISO 13482の概要を紹介し、ロボット開発に連動した認証活動を紹介する。次いで、ロボット開発における証跡とその管理について述べ、証跡管理ツールに求められる要件を、第三者機関の視点から述べる。

2. ISO 13482について

ISO 13482［3］は、2014年に発行された国際規格で、当時ISO/TC 184（現ISO/TC 299）で開発された。この規格は、機械安全の基本規格であるISO 12100［4］の流れを汲むと共に、機械分野の安全関連制御部を対象としたISO 13849-1［5］やIEC 62061［6］が参照されている。

なお、ISO 13482（JIS B 8445［7］）の用語の定義として、“ロボット”は「2軸以上がプログラム可能で、一定の自律性をもち、環境内を移動して所期のタスクを実行する作動メカニズム」、“自律性”は「人が介入することなく、現在の状態及びセンサ計測に基づいて、意図したタスクを実行する能力」である。また、“生活支援ロボット”とは「医療用を除く、人の生活の質の改善に直接寄与する行為を実施するサービスロボット」と定義されている。一方、当該規格の中では、以下のロボットが適用外とされている。

- 20 km/h を超える速度で移動する生活支援ロボット
- ロボット玩具
- 水中ロボット及び飛行ロボット
- 産業用ロボット
- 医療機器としてのロボット
- 軍用又は公権力に資するためのロボット

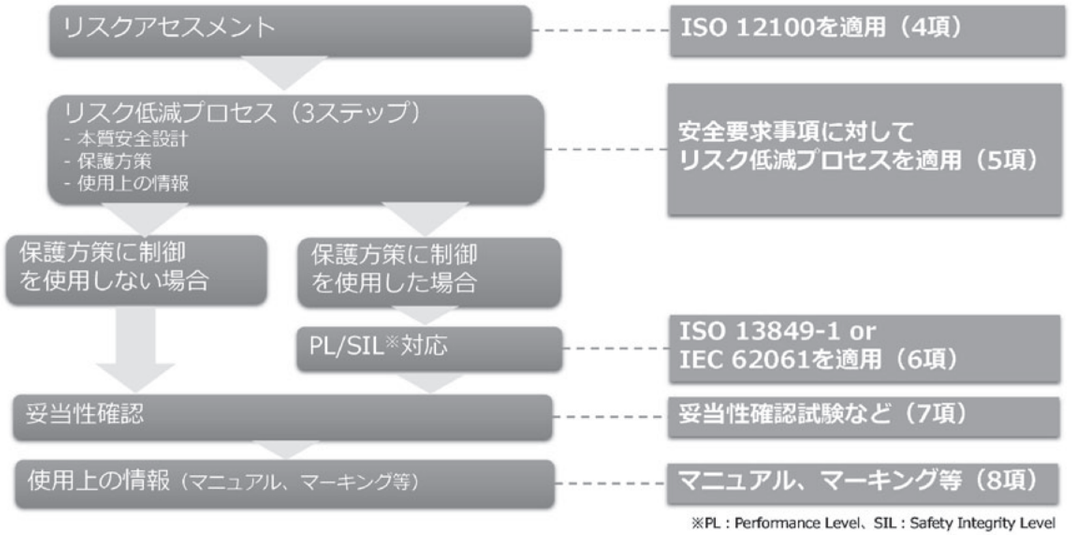


図1　ISO 13482の構成イメージ

当該規格の中では、以下の3タイプのロボットがあげられている。

－移動作業型ロボット：図2（a）
－身体アシストロボット：図2（b）
－搭乗型ロボット：図2（c）

ただし、このタイプのロボットだけに限定しているわけではない。

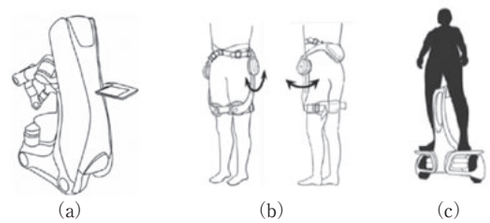


図2　ISO 13482における
3つのロボットのタイプ例[3]

この規格の構造を図1に示す。同図が示すとおり、“リスクアセスメント”を端緒に“リスク低減の3ステップ”を経て、必要に応じて“制御システムによる保護方策”を講じて対象ロボットにおけるリスク軽減を求めている。次いで、“妥当性確認”、“使用上の情報の確認”という造りである。認証取得に向けた活動の初期においては、リスクアセスメントを進める上での前提となる「ロボットの意図した使用を明確にすること」が重要である。誰が、いつ、どこで、何のために、どのように使うのかを明らかにすることによって、対象とするロボットの具体的な危険源の同定が可能となるためである。開発するロボットにおける危険源の同定を漏らしてしまうと、出来上がったロボットが、ある危険源に対応できない未熟さを有する可能性を否定できない。そのため、危険源の同定には細心の注意を払うとともに、この活動の足跡を記録しておくことも重要である。なお、ISO 13482の附属書Aには、表1に示す通りロボットで考えるべき危険源リストがあげられており、ロボットメーカは、これを参照しながら危険源の同定を進めることができる。

ISO 13482の安全要求には、リスクアセスメントの実施に加え、「機械、電気、電磁両立性（EMC）、肉体的または精神的なストレス、機能安全」と多岐にわたる観点が含まれる。ロボットメーカは、認証取得のためにこれらの要求への適合性を示すことが求められ、そのための証跡を予め用意することが必要となる。

次に、JQAにおける生活支援ロボットの認証活動の流れを図3に示す。同図に示したように、JQAでは、開発の初期段階である“概念検討”

表1　ISO 13482　附属書Aにおける危険源の種別の例

危険源の種別	
バッテリ充電の危険源	有害な電離放射線
エネルギーの貯蔵及び供給の危険源	EMI/EMC の危険源
ロボットの起動による危険源	ストレス，姿勢及び使用の危険源
ロボット形状による危険源	ロボットの動きによる危険源
騒音による危険源	安全関連物体との衝突
認識不足による危険源	人とロボットの相互作用中の危険な身体接触
危険な振動	耐久性不足
危険な物質及び流体	危険な自律動作
危険な環境条件	可動部との危険な接触
極限温度	位置確認及びナビゲーションエラーによる危険源
有害な非電離放射線	その他の危険源の種別

から“リスクアセスメント”を経て上市に至るまでの各ステップをロボットメーカと並走しながら、効率的に ISO 13482 の認証に向けた評価活動を進めて行く。また、品質マネジメントの側面から、設計工程（図中のフェーズ 1）と製造工程（図中のフェーズ 2）において、それぞれの品質管理体制を評価する。

3. 認証取得活動の効率化

認証は、図 4 に示すように、第三者が製品に係る規格の要求事項を満足していることを証明（attestation）した結果である。そのため、認証取得により期待される最大の効果は、客観的に製品の安全性を示せることである。これにより、メーカ視点では製品の訴求点になり、ユーザ視点では安心して導入、使っていく拠り所となる。図 3 に示したように、生活支援ロボットの認証活動では、技術的な側面と設計・製造の過程における品質管理の側面を有している。ロボットメーカにとっては、通常の開発活動に加え、認証取得のための活動として、例えば、適用する規格要求事項の理解、要求事項に対応するための設計や検証、付随する文書や記録の作成・管理、これに加え認証費用も必要となる。したがって、認証取得では、客観的に安全性を示せる反面、手間、時間、コストが掛かる、言わばトレードオフの関係となる。

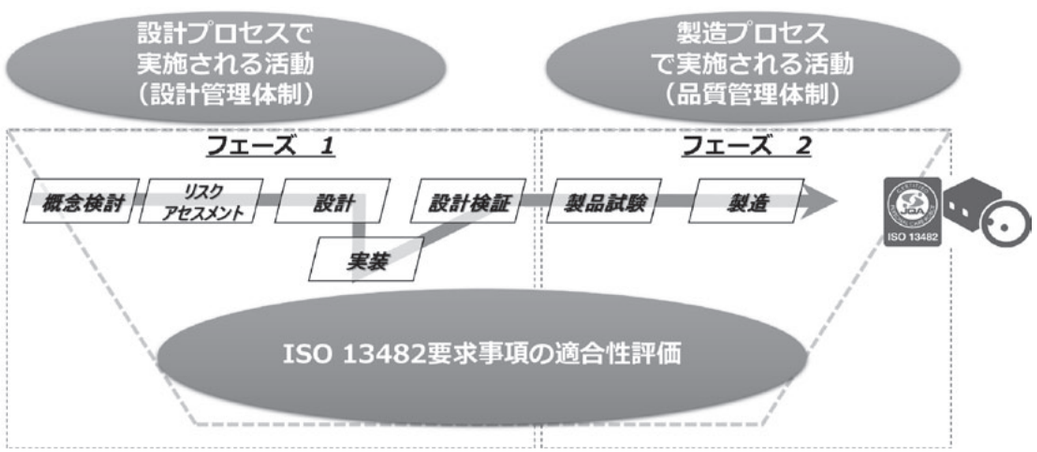


図3　JQAの生活支援ロボットの認証の流れ（イメージ）

図4　第三者評価の概念

認証取得のための活動のうち、多くを占めるのが、文書・記録の作成・管理と見られる。一方で、第三者機関でも、要求事項を満足しているか評価を進める中で、関連文書や記録類をレビューするために多くの時間を費やす傾向にある。評価の際、ロボットメーカとの文書・記録のやり取りが五月雨式の場合や、文書・記録間の紐づけが複雑な場合では、この傾向が強くなる。

したがって、文書・記録の作成・管理およびレビューに関する活動を効率的に実施できれば、認証取得のマイナス面が軽減できると考えられる。

4. ロボット開発における証跡とその管理

証跡とは、痕跡や証拠の意味を持つ。製品開発においては、例えば、下記が該当する。

－プロジェクトの契約関連文書
－開発プロセスに関する規程文書類
－開発中に生成される設計文書・記録類
－テスト計画・結果に関する文書・記録類
－開発環境に関連する文書類
－参考情報（規制類、ノウハウ集等）

これらの証跡は、製品の開発期間を通じ、様々なタイミングで生成、改訂される。例えば、開発フェーズの進展、製品試作の繰り返し、内外環境の変化、サプライチェーン間の文書の往来である。さらに、派生文書や下位文書の生成、文書が廃止され管理対象外となる等、各文書・記録の寿命は様々である。このように、証跡の種類が多く各々が複雑に絡み合うため、その管理は容易ではない。

生活支援ロボットの認証活動では、ロボットメーカには ISO 13482 への適合性について証跡間の関係を用いた説明が求められる。すなわち、ロボットメーカが、第三者機関に対し、リスクアセスメントからの一連の開発の流れの中で、ヌケモレや前後のズレがないように開発活動を実施したことを系統的に示すためである。そのために、開発の流れに沿った証跡間の繋がりと、開発の流れを遡った証跡間の繋がりを説明できるようにすることが重要である。

そして、これらを説明していくために、必要な時に関連する文書や記録がすぐに確認できるように管理される必要がある。品質マネジメント規格である ISO 9001:2015 [8] では、“管理された状態”について定義されている。以下に、その抜粋を示す。

> **8.5.1 製造及びサービス提供の管理（抜粋）**
> 組織は、製造およびサービス提供を、管理された状態で実行しなければならない。・・・
>
> a. 文書化された情報を利用できるようにする
> c. …適切な段階で監視及び測定活動を実施する
> d. プロセスの運用のための適切なインフラストラクチャ及び環境を使用する
> （以下省略）

したがって、ロボットメーカは、このような点に配慮しながら証跡管理を行うことが必要である。次に、その方法と求められる要件について検討する。

5. 証跡管理ツールに求められる要件

証跡が“管理された状態”を実現するためには、開発プロセスに証跡を管理する仕組みを織り込むことが考えられる。端的に言うと、ルールを設け、それに従って運用することである。例えば、管理対象の範囲、証跡として登録するタイミング、各文書や記録の識別方法及び保存方法、証跡の変更や廃止の手続きを予め規定しておく。

これらのルールの運用を開発組織内に定着させる手段として、ツールの導入が有効と見られる。ただし、例えば、管理項目数やツールのユーザ数から、図 5 のように帳票・台帳による証跡管理でも対応できる場合もあると見られる。一方で、組織によっては、多くの製品ラインナップを有する場合や、多数の部署とサプライヤと連携して開発

する場合もあり、図5のように、基幹システムに証跡管理の機能を持たせる方法もあると考えられる。いずれにしても、開発規模や組織の状況にマッチした証跡管理の手段を講ずることが肝要である。

これまでの検討を踏まえ、証跡管理ツールに求められる要件をあげてみる。まず、基本的な機能としては、例えば、次の通りである。

- 各証跡を識別し、各々紐づけて管理できる
- いつでも必要な状態を識別できる

次に、第三者機関の視点を踏まえ、効率的に説明責任を全うする上で便利と思われる項目を列挙する。

- 各種の管理機能がある（要件管理、変更管理、構成管理等）
- 要件レベルでの双方向トレーサビリティが確保できる
- 証跡の変化（削除、変更、追加）に追従し、かつ証跡間の参照関係が維持される（参照関係が壊れた場合にはアラートされる）
- 複数のアクセスからの情報の整合性の確保や不正改ざんが防止できる
- 様々なデータ形式を扱うことができる
- アラート機能（変更箇所やヌケ・モレ・ダブりの検出等）がある
- 承認パスの実装およびメール等のメッセージ機能がある
- GUIが充実している
- 証跡管理ツールの妥当性が確認されている

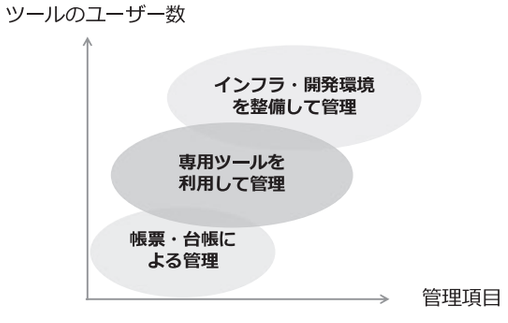


図5　証跡の管理方法の分類例

開発の対象製品の安全機能に係る証跡を扱う場合には、最後にあげた「証跡管理ツール自体の妥当性が確認されていること」が重要である。

機能安全の基本規格IEC 61508シリーズのソフトウェア要求事項であるPart3［9］では、開発支援ツールの要件があげられている。

- オンライン支援ツール（稼働中に安全関連系に直接影響を及ぼす）
- オフライン支援ツール（稼働中には影響を及ぼさない）

オンライン支援ツールは、安全関連ソフトウェアと同様の開発が求められる。オフライン支援ツールについては、表2の分類が示されている。T1よりT3の方が安全機能への影響度が高く、要求事項が多くなる傾向がある。

車載用電気電子（E/E）システムに関する機能安全規格であるISO 26262シリーズのうち、Part8［10］においても開発ツールに関する要求事項がある。まず、ツールのユースケースから、安全機能への影響度（TI：Tool Impact）を分析し、ツールのエラー回避・検出対策（TD：Tool error Detection）の有効性を判定して、TCL（Tool Confidence Level）を決定する。次に、表3に示す通り、TCLとASIL（Automotive Safety Integrity Level）の関係に応じて評価手法を決め、TCLが満たされていることを確認する。

このように、第三者機関の視点としては、証跡の管理状態と合わせて、信頼の置ける管理ツールが利用されていることも、効率的に説明責任を全うするための重要なポイントである。

表2　IEC 61508-3におけるオフライン支援ツールの分類

分類	説明	ツール例
T1	安全関連系の実行可能コードに寄与する出力を生成しない	テキストエディタ、構成管理ツール
T2	ツール内にエラーがあっても欠陥をあきらかにすることはできないが、実行可能ソフトウェア内に直接エラーを生成することはなく、設計又は実行可能コードのテストまたは適合確認を支援する	カバレッジ測定ツール、静的解析ツール
T3	安全関連系の実行可能コードに直接または間接的に寄与できる出力を生成する	コンパイラ

表3　ツール認定におけるTCLとASILの関係に応じた評価手法（ISO 26262-8）

	手法	TCL2				TCL3			
		ASIL				ASIL			
		A	B	C	D	A	B	C	D
1a	使用による信頼性向上	++	++	++	+	++	++	+	+
1b	開発プロセスの評価	++	++	++	+	++	++	+	+
1c	ソフトウェアツールの妥当性確認	+	+	+	++	+	+	++	++
1d	安全規格に沿った開発	+	+	+	++	+	+	++	++

++：高い推奨
+　：推奨

6. おわりに

本稿では、生活支援ロボットを念頭に置き、開発における証跡とその管理について述べ、第三者機関の視点を踏まえ証跡管理ツールに求められる要件を述べた。

本稿が、証跡管理ツールを活用することによって製品の安全性に関する効率的な説明責任を果たすことの一助になれば幸いである。

参考文献

［1］ 生活支援ロボット実用化プロジェクト、http://www.nedo.go.jp/activities/EP_00270.html
［2］ JQA　認証取得者リスト、http://www.jqa.jp/service_list/fs/action/clientele/index.html
［3］ ISO 13482:2014　Robots and robotic devices -- Safety requirements for personal care robots
［4］ ISO 12100:2010　Safety of machinery -- General principles for design -- Risk assessment and risk reduction
［5］ ISO 13849-1　Safety of machinery - - Safety-related parts of control systems -- Part 1: General principles for design
［6］ IEC 62061:2012　Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems
［7］ JIS B 8445:2016 ロボット及びロボティックデバイス－生活支援ロボットの安全要求事項
［8］ JIS Q 9001:2015 品質マネジメントシステム－要求事項
［9］ IEC 61508-3:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 3: Software requirements
［10］ ISO 26262-8:2011 Road vehicles -- Functional safety -- Part 8: Supporting processes

ZIPC WATCHERS Vol.20

ZIPC TERAS を活用した機能安全への取り組み

古河 AS 株式会社
技術本部 RA 統括部 技術開発 3 部 マネージャー
平井　秀明

1. はじめに

古河 AS 株式会社では、車載用の周辺監視レーダの開発を行っており、昨年から 24GHz 帯レーダの量産を開始した。本レーダは、BSD（Blind Spot Detection、死角検知）、LCA（Lane Change Assist、車線変更検知）、RCTA（Rear Cross Traffic Alert、後退時接近物検知）の機能を搭載しており、車両後方からの接近物へ警報を行うものである。

一方、周辺監視レーダなどの先進運転支援システムでは自動運転に向けた取り組みなどもあり機能安全としての要求が高まっている。機能安全規格の ISO26262 の中ではトレーサビリティの必要性、追跡可能でなければならないことが記載されており、またツールの使用が推奨されていることもあり、ますますトレーサビリティの重要性が増している状況にある。

2. 導入の背景

先の説明でもふれたが、機能安全としての必要性が導入の第一の背景となる。機能安全の説明として本質安全を比較として説明することがある。本質安全は、潜在危険そのものを除去するというもので、例えば電圧を下げて危険を除去する、スピードが出ないようにするなどして潜在的な危険を除去するというものである。現在の自動車のシステムは機能の複雑化、ECU 搭載個数の増大、それに伴うソフトウェア量の増大により、本質安全だけでは解決できない課題が増加している。それに伴い、自動車の電気電子システムの機能不全の振る舞いに限定したものとして規定されているのが ISO26262 であり、本規格へ対応することが周辺監視レーダとしても必要となっている（図 1）。

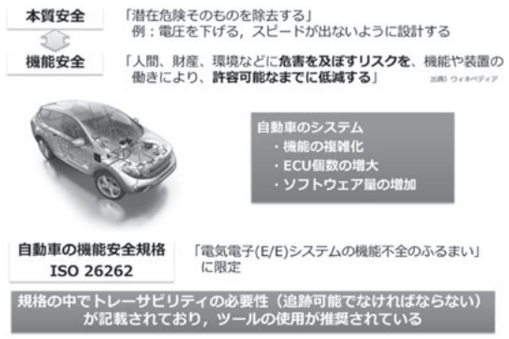


図1　本質安全と機能安全

第二の背景として、現在開発の中でプロジェクト管理ツールとして Redmine、構成管理ツールとして Subversion を利用しており、これら 2 つのツールと連携させて変更管理を行い、効率的な運用を行う「3 ツール連携構想」があり、それを実現させるためのツール導入も背景となっている（図 2）。

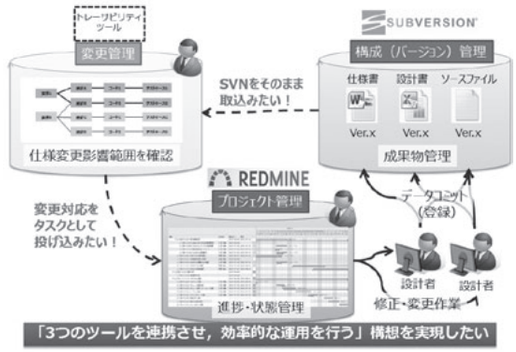


図2　3ツール連携構想

3. ZIPC TERAS 選定理由

ツールの導入計画に関しては、社内では別のツール利用などもあるが、今回レーダ開発としてツールの再選定を行った。レーダ開発そのものは昨年 11 月より量産が開始されており、量産開発とは別軸での検討が必要である。トレーサビリティ対象物としては量産開発品のドキュメントを用いるが、別軸でツール選定、環境構築、実モデルへの適用、効果確認を行った（図 3）。

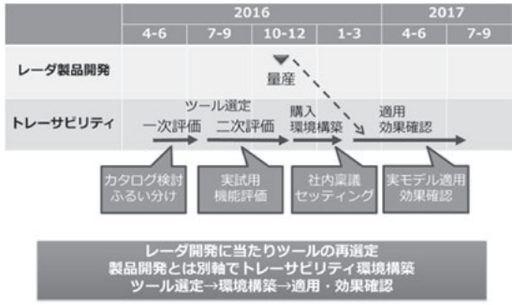


図3　導入活動計画

選定に向けた計画としては、一次評価、二次評価を経てツールを選定し、その後環境構築を行い、実モデルへの適用、効果確認を行うこととした。

まず一次評価に関しては、現状トレーサビリティが実施可能なツールに対してカタログ調査とベンダーへのヒアリングを行い、その結果をもとに比較表を作成した。今回は先にふれた「3 ツール連携構想」が可能であるということを重視して連携機能が備わっているトレーサビリティ専用ツールを一次選定品とした。

続いて二次評価だが、選定した 3 つのツールの評価版を一定期間借用し、実際のトレーサビリティを行い機能の評価を行い、最終的には点数付けして選定を行った。評価した項目としては、ドキュメントの取込み形態やカバレッジ確認方法、影響範囲分析の表示方法などの機能確認、変化点の確認方法、ツール連携機能、サポート要望対応力、価格となる。最終的には、これらの評価項目の点数値の高さ、今後の展望を考慮して ZIPC TERAS を採用した（図 4）。

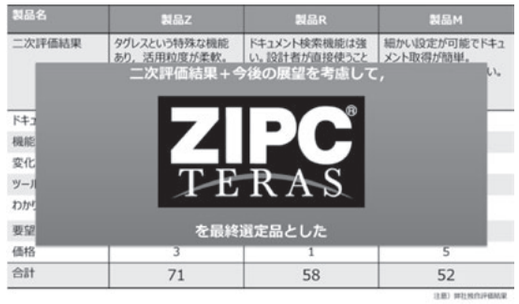


図4　二次評価結果

4. ZIPC TERAS 適用範囲

今回のトレーサビリティの対象範囲としては、図 5 の Automotive SPICE プロセスのソフトウェア要件分析からソフトウェア適格性確認テストまでを範囲としている。また、対象としたドキュメントはエクセル文書、ワード文書、テキスト（ソースコード）となっている。

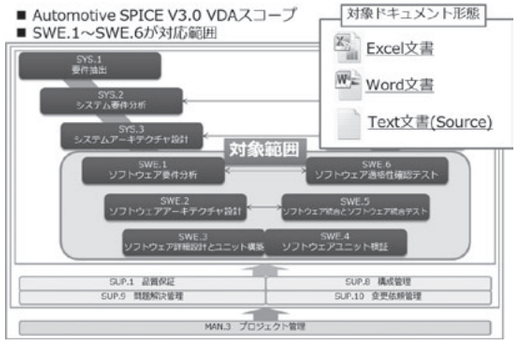


図5　ZIPC TERAS適用範囲

5. ZIPC TERAS タグレス機能に関して

ZIPC TERAS には特徴的な機能として、タグレス機能があり、今回の適用では本機能を使用した。通常のタグベースのトレーサビリティでは自タグ ID とリンク対象となる参照 ID を付与し、それらの ID 間の関連性をもとにリンクを自動生成する。よって ID が主体となり、ドキュメントへ ID を如何につけるのかドキュメントが主体でトレースが行われる。

一方、タグレスでは参照 ID を付与せずにドキュメントの要素をトレース対象とする。よって ID ではなく、要素が主体となり、リンクはツール上で行うためツールが主体でトレースが行われる。

タグレスはZIPC TERASの特徴的な機能であり、開発過渡期などの設計変更が多く行われる状態では非常に有効な手段となる（図6）。

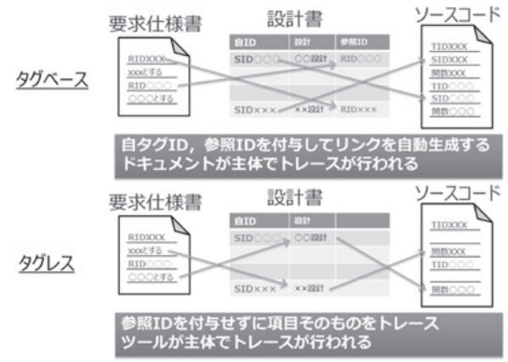


図6　タグレス機能

6. ZIPC TERAS 適用事例

ここよりタグレスのトレース作業の流れを実際のドキュメントを使用して説明する。基本的な流れとしては、ルールファイルの設定、ドキュメント登録、ドキュメント間のリンク付け、カバレッジ計測出力、影響範囲確認（変更時）となる（図7）。

図7　ZIPC TERASトレーサビリティの流れ

まず、ドキュメントを取込むためのルールを設定する。このルールは1つのドキュメントごとに設定が行われ、ツール上ではxmlファイルに取込む手法を記載する形となる。今回新たにバージョン3がリリースされた。このバージョン3によって、階層取込み設定や取込みスピードの高速化といったエクセルの取込み機能が強化された（図8）。

図8　ルールファイルの設定

次にドキュメントの取込みだがまずはエクセル文書の例を示す。今回の適用では、行方向へ全行走査とし、階層取込み定義を行って、列の取込み範囲の指定と階層化を行った（図9）。

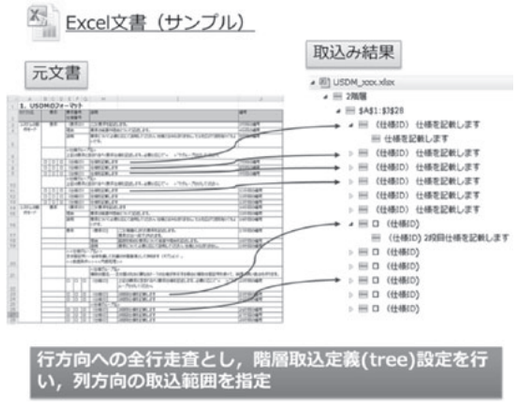


図9　エクセル文書の取込み

ワード文書の取込みは、ツールのデフォルト設定にて、ワード文書の見出しの取込みが可能となる。また、部分的に取込みたい要素がある場合は、手動で選択して要素化することで取込みが可能となる（図10）。

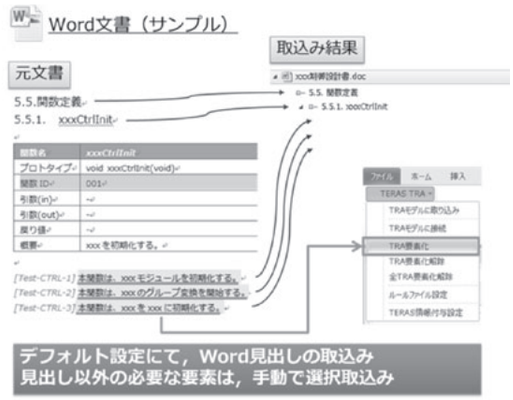


図10　ワード文書の取込み

ソースコード等のテキストファイルに関しては、ルールファイル内で正規表現にて取込み定義を行うことで取込みが可能となる（図11）。

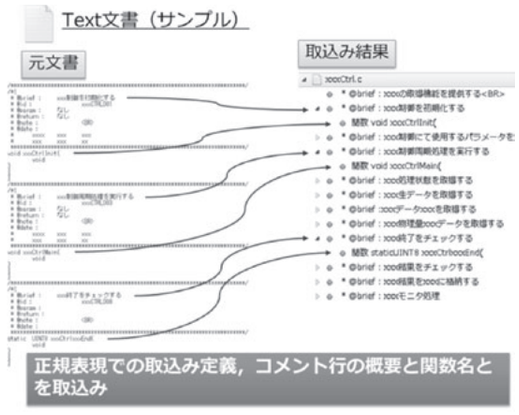


図11　テキスト文書の取込み

取込んだ文書の関連性を付与するためのリンク付け作業に関しては、タグレスではツールの画面上で行う。実際に要素を見ながら選択しリンク付けが行える。これによりドキュメントへの作業が不要となり、ドキュメント作業者との分業や客観性の担保が可能になる（図12）。

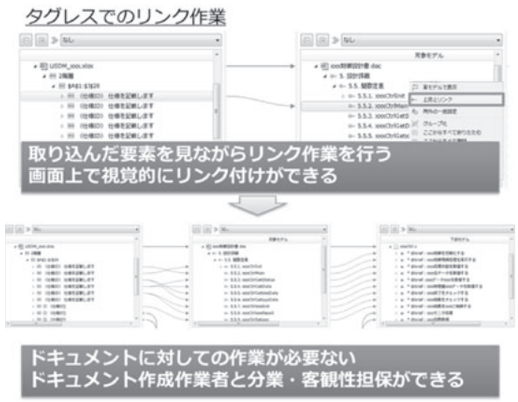


図12　タグレスでのリンク付け

リンク付けを行った後のカバレッジ確認方法は、まずどの要素同士をカバレッジ計測対象とするのかを設定するため、計測対象としない要素として例外設定を行う。これにより、ツール上でリンク有無の確認やカバレッジ率の確認が行える。また、結果をCSVファイルで出力して内容を確認することもできる。タグレスではあるが、内部的にIDが付与されていることが出力内容から確認することができる（図13）。

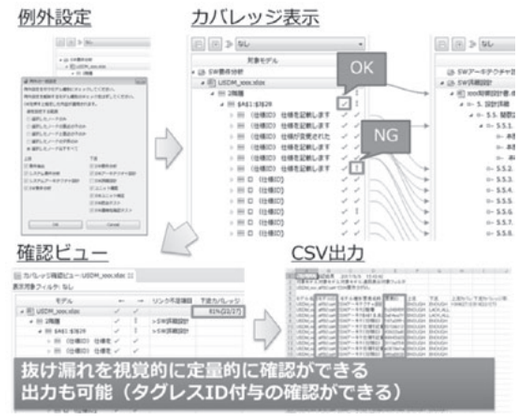


図13　カバレッジ確認

ドキュメントが変更された場合の影響範囲の確認に関しては、ドキュメントの変更箇所の再取込みを行い、ツール上で更新確認を行うことで変更箇所をハイライト表示することが可能となる。これにより、変更箇所の特定が容易に行える（図14）。

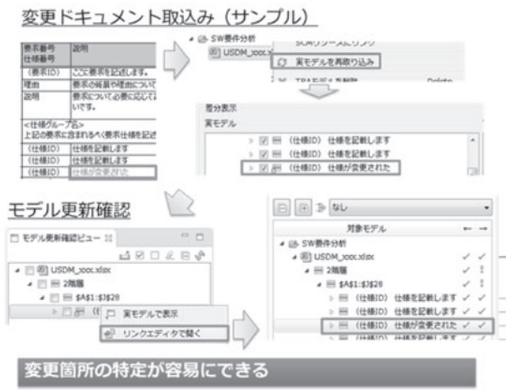


図14　変更時の影響範囲確認

影響範囲を確認した後は、関連ドキュメントやソースコードの修正が必要となるが、その際に有効となるのがRedmineへのチケット発行機能である。影響範囲を確認後、本機能を用いることでRedmineへ影響範囲対応に関するタスクとしてのチケットをZIPC TERAS上から発行することができる。これにより、当初の目的であった、プロジェクト管理と変更管理の連携が可能となった（図15）。

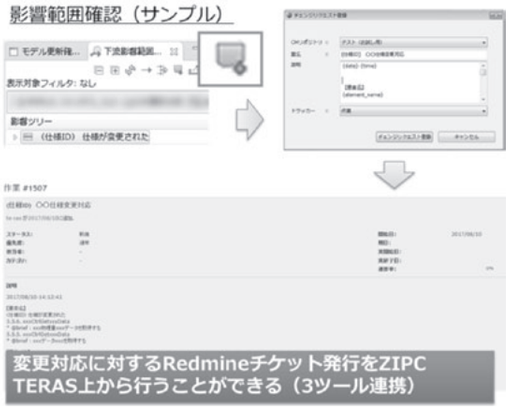


図15　ZIPC TERASからのRedmineチケット発行

7. ZIPC TERAS 適用結果

導入の背景にて「3ツール連携構想」についてふれた。今回、変更管理ツールとしてZIPC TERASを利用することとした。これにより、Subversion管理下のドキュメントがそのまま使用できるようになり、また変更対応がZIPC TERAS上からRedmineへチケット登録にてできるようになった。これらにより、当初達成したいと考えていた「3ツール連携構想」の実施が可能となり、抜け漏れのない対応を行う環境の構築をすることができた（図16）。

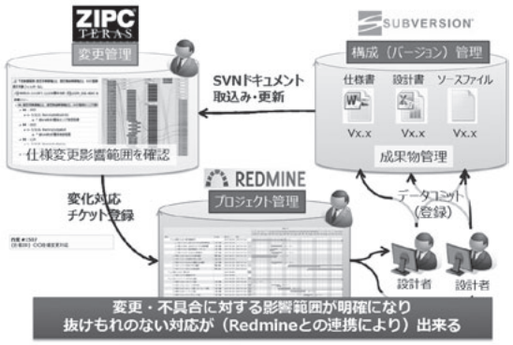


図16　3ツール連携構想の達成

これまでのツールを使わない状態でのトレーサビリティは、エクセルで行っていた（図17）。その場合上流からの順方向トレーサビリティと下流からの逆方向トレーサビリティと両方向からの表を作成しての確認作業が必要となっていた。これにより、多くの確認工数が必要となっていたが、ZIPC TERASのリンクビューアーを用いることで順方向、逆方向の確認が1つのビューアー上で簡単に行うことができるようになった（図18）。

図17　エクセルベーストレーサビリティ

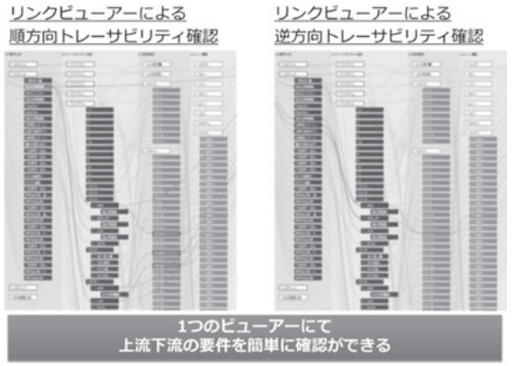


図18　順/逆方向トレーサビリティ

また、エクセルではトレーサビリティとしての設計書の質まで俯瞰視しての確認はしづらいが、ZIPC TERASを使用して要素を取込むことで設計書記載の粒度、精度を確認することができた。

図19の左側はエクセルベースのトレーサビリティで作成したものをそのまま取込んだもので、途中で要素がシュリンクしてしまう形となることがわかった。今回、ZIPC TERASを使用した結果、図19の右側のように記載の粒度を細分化する修正を行うことができ、設計品質向上につなげることが可能となった。

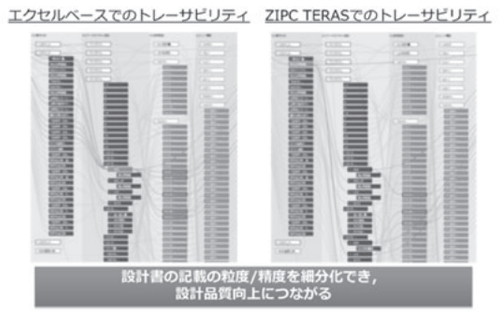


図19　設計書の質

8. 今後の展開

今回、ZIPC TERASを利用することでドキュメント記載粒度、記載手法などに関しての新たな知見が得られ、ルールファイルの設定手法に関しての知見が得られた。これらを以降の開発に向けて規定を行っていきたいと考えている。また、MATLAB/Simulinkモデル等の関連ツールの取込みも行っていきたいと考えている。

9. むすび

ZIPC TERASと知り合ってから本日まで1年超。ZIPC TERAS導入に当たって、初期検討から評価版借用評価、環境構築、導入時のトラブル、導入後のサポート、不具合解明、修正に対して、長きにわたり営業の方、技術の方含め誠心誠意対応していただいた。新しいツールということもあり、まだ足りない部分もあるが、今回の導入初期からの手厚い支援が最終的な導入の決め手になったと考えている。その点、キャッツ株式会社の皆様の対応に深く感謝申し上げるとともに、ZIPC TERASの更なる進化に向けた活動を継続していただきたいと考えている。

ZIPC WATCHERS Vol.20

車載システム開発における ZIPC 適用事例紹介

株式会社コマス
第一事業部 事業部長
大村　一将

1. はじめに

当社の社歴は古く設立は 1970 年、今期で 48 期目、2019 年には 50 周年を迎える、この業界でも老舗の会社です。

当社の商号であるコマスの名前の由来は、Computer の COM、Application の A、Service の S を組み合わせ“COMAS”でコマスとなります。

当社の事業内容はソフトウェア開発ですが、主に「社会インフラを支えるクリティカルソフトウェアの開発」です。24 時間、365 日確実に動き続けることを要求される「安心・安全なソフトウェア」を開発する技術力を有しています。

図1　事業内容

クリティカルソフトウェアの中でも、車載関連事業は注力分野で、当社の開発体制の約 4 割強が車載関連のソフトウェア開発に従事しています。今回の事例紹介の対象である、車載機器や、エンジン制御 ECU、HMI プラットフォーム、車載 IoT 関連等、多岐に渡る開発実績があります。

2. 事例紹介

2.1 システム概要

今回 ZIPC を適用し開発を行ったシステムの概要を図 2 にて説明する。

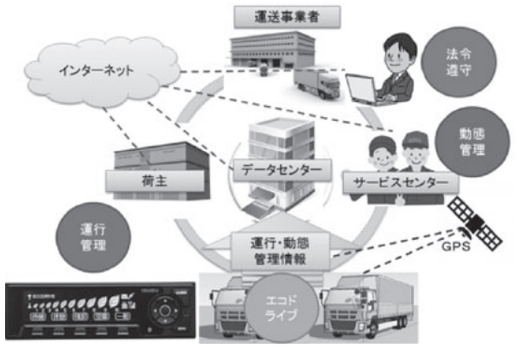


図2　システム概要

当社が開発に従事したのは、図 2 左下にある車載機器のソフトウェアである。この車載機器は商用車（トラック）の運転席ダッシュボードに組み込まれる機器で、車両に搭載された各種センサー、GPS モジュール等のデバイス、ETC、デジタコ等の他の車載機器と、CAN、UART 等のインターフェースで通信している。

この車載機器を用いて、商用車向けのテレマティクス（テレコミュニケーション＋インフォマティクス）サービスとして以下の機能を実現している。

＜エコドライブ＞

燃費抑制や安全確認といった、エコドライブのガイド機能、トレーニング機能を提供する。

＜運行管理＞

運行状況確認など、インターネット経由でリアルタイムでの運行管理を幅広くサポートする。

＜動態管理＞

安全確保に必要な情報サービスを、リアルタイムで幅広くサポートする。

＜法令順守＞

データセンターに蓄積されたデータを基に、運転日報、労務管理等のサービスを提供する。

2.2 開発プロセス

開発プロセスは図 3 にて示すように、一般的なウォーターフォール型であり、当社の受託範囲は基本設計工程からシステム試験である。

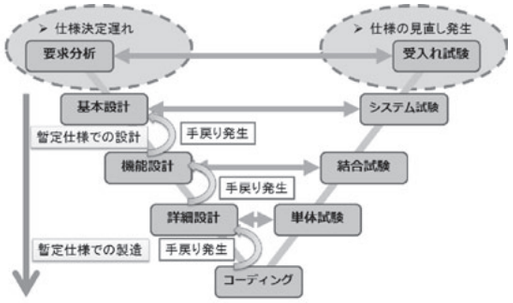


図3　開発プロセス

車載テレマティクスサービスを提供する自動車メーカー様、通信サービスを提供する通信キャリア様、車載機器を供給するサプライヤー様とステークホルダーが多く、仕様決定が遅れがちである。更に、受け入れ試験中や、製品リリース後も、法規対応や、実際にサービスを利用するエンドユーザー様に近い販社様からの要望等で、最終仕様の見直し等が発生する。

2.3 車載システム開発の特徴

- レガシーシステムをベースとした流用・改造が中心の開発手法
- 新規ハードの適用や、部品収束に伴うシステム開発も発生する
- 仕様決定に係るステークホルダーが多く、仕様決定が遅れる傾向
- 暫定仕様にて設計・製造を進める結果、仕様確定後に手戻り作業が発生

年々多機能・高品質・短納期開発が求められ、製品ハード設計と同時進行となることが前提である。したがって、ソフトウェア開発体制においては、設計、実装、評価の一貫性と、品質、効率が非常に重要な要素である。

2.4 ZIPC 導入の背景

ZIPC 導入の背景には以下の要素があった。

① レガシーシステムは状態遷移表設計手法で設計
② 長期間の開発でマトリクスが肥大化
③ 暫定仕様での開発や継続的なエンハンス開発が前提
④ 10 年振りに新規ハードでの大規模システム開発が発生

これらにより、開発プロセスとソフトウェア構造の改善が必要と判断し、ZIPC を画面部タスクの開発へ導入を決定した。

2.5 ZIPC 導入に期待する効果

■ソフトウェア構造と開発プロセスと品質マネジメントの改善

ソフトウェア構造が変われば開発プロセスも変わる。開発プロセスが変われば品質マネジメント手法も変わる。つまりは、ソフトウェア構造、開発プロセス、品質マネジメントは密接に関連している。

過去に私が従事した某システム開発にて、ZIPC 適用を廃止する判断がされたが、その原因は、ZIPC を適用した開発手法と、ソフトウェア構造のミスマッチであった。更に年 2 回の大規模機能リリースと、継続した仕様変更、保守対応により、開発当初のポリシーが守られず、ZIPC のイレギュラーな使われ方によってソフトウェアの改悪になっていた。ZIPC は素晴らしいツールであるが、使い方を間違うとリスクになってしまう場合もある。

今回は今後 10 年エンハンスするシステムの初回開発につき、ソフトウェア構造、開発プロセス、品質マネジメントの再考の機会と捉え、ZIPC 導入を決定した。

■マトリクス規模の削減

画面部タスクのリファクタリングによるマトリクス構造の最適化を行う。

レガシーシステムの画面部タスクには、Excel ファイルで作成された図 4 にて示すような、

265（状態）× 21（イベント）の巨大な状態遷移表が存在していた。従来構造のままでは、更なるマトリクスの肥大化と、生産性・品質の両面でリスクとなり得る為、図 5 にて示すような状態遷移表の階層化を適用することでマトリクス規模の削減を期待した。

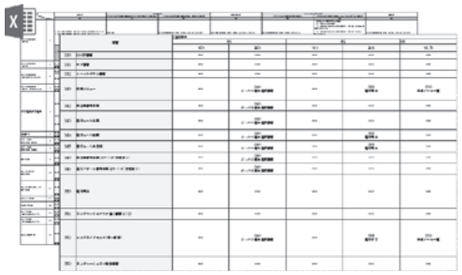


図4　既存状態遷移表

図5　階層化イメージ

2.6 ZIPC 活用事例の紹介

■チェッカ機能の活用

各設計工程で作成したドキュメントに対して、構文や関連ドキュメントとの整合性を静的検証する ZIPC のチェッカ機能を活用した。設計工程のレビュー時にチェッカ機能を活用することで、設計品質の向上が図れた。

図6　チェッカ機能

■マトリクス構造の改善

レガシーシステムの画面部タスクは、全画面に対する画面遷移処理を最下層の状態遷移表にて処理していたため、巨大な状態遷移表となっていた。この構造では、仕様変更等でイベントが追加された場合、265 状態全てに対して、設計、製造、評価の影響確認が必要である。

今回の開発にて、画面単位の機能と画面遷移をパターン化することで、階層化設計と状態遷移表間の疎結合化を適用した。これにより仕様変更等でイベントが追加されても、設計、製造、評価に対する影響を限定することができた。

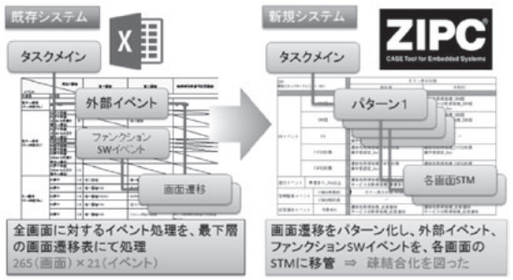


図7　階層化設計

2.7 ZIPC 導入の効果

本開発にて、画面部タスクに ZIPC を適用したことで、システム規模の 34% に対して ZIPC を活用することができた。

図 8 にて ZIPC 導入による生産性面での効果を示す。当初計画段階では、ZIPC 導入により初回開発で約 24% の効率化を期待していたが、結果は 16% の効率化であった。

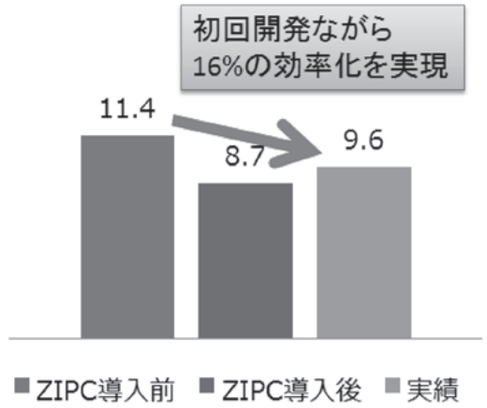


図8　ZIPC導入効果

効率化の予実差異について、表 1 に示すように、設計・製造・評価のプロセス毎で振り返る。

表1　見積りと実績

	見積り		実績工数
	ZIPC 導入前	ZIPC 導入後	
設計	3.7	1.7	4.6
製造	2.0	1.7	2.5
評価	5.7	5.3	2.5
小計	11.4	8.7	9.6

＜設計＞

従来の開発と異なる設計方法による不慣れさによる生産性低下と、画面部タスクに係る全てを ZIPC 適用予定も、仕様確定の遅れに伴う機能の段階リリースの都合から、一部の ZIPC 適用を断念したことで手戻り発生。

＜製造＞

ZIPC 適用を断念した部分の従来と同様な実装方法と、置換ルールの定義製造の不慣れさから想定以上の工数が掛かった。

＜評価＞

シミュレート機能の完全導入ならずも、チェッカ機能及び、ジェネレータ機能により、処理の抜け漏れの早期検出の結果、評価工程における軽微なバグ数を大幅に削減できた。

3. 今後の展望

3.1 シミュレート機能導入

ZIPC 適用範囲を更に拡大し、シミュレート機能を活用することで、設計段階または、仕様検討段階で画面（システム外観図）の動作イメージを早期に確認することができる。図 9 参照。

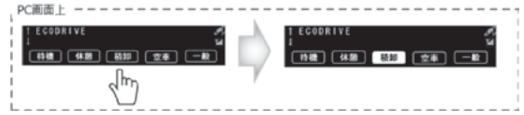


図9　シミュレート

システムの外観部を Visual Basic 等で作成することで動く画面仕様書が作成できる。最終的に出力する C ソースをコンパイルしてシミュレートを行うため、シミュレート通りの動作が保障される。

3.2 まとめ

ZIPC を導入し、チェッカ機能、ジェネレータ機能を活用することで、仕様変更や機能追加が多い画面部タスクの生産性、品質向上が実現できた。ZIPC 適用部、非適用部が混在すると、適用効果が限定的となるため、今後の開発において、適用範囲の拡大を推進する。これによりシミュレート機能の完全導入を実現する。

本開発のようなレガシーシステムのリファクタリング目的で ZIPC 適用の際の注意点を示す。見積り計画段階においては、開発プロセスと工程生産物に対するトライアル結果を踏まえた、見積り、計画が重要である。設計、製造段階においては、レガシーシステムの過去の工程生産物との関連性を可視化など、中間生産物を作成する導入準備が必要である。

最後に、本稿の作成にあたりご協力いただいた関係者各位へ感謝を申し上げる。

衛星シミュレータへのZIPCの活用事例紹介

日本電気株式会社
宇宙システム事業部 主任
佐久間　彬

1. 目的および背景

近年の衛星システム開発は、ミッション要求の高度化に伴い、複雑および大規模となってきている。そのため、上流工程における設計検証を充実化させ、ミッションの成立性を早期に確認する必要性が高まっている。

弊社では、高度なミッション要求を実現する衛星運用機能を検証するため、ZIPCを用いた衛星シミュレータを開発した。本論文では、その活用事例を紹介する。

2. 衛星システムの概要

衛星システムは、宇宙環境から衛星本体を保護・維持するバスサブシステムや、撮像などの目的を実現するミッションサブシステムで構成される大規模なコンカレントシステムである。

図2-1に標準的な衛星システム構成を、表2-1に各サブシステム名称と役割を示す。

衛星は、ロケットなどの輸送機によって打上げられる際に、激しい振動や衝撃に耐えなければならない。また、宇宙空間において、宇宙放射線や広い温度変化にさらされる。一方で、搭載できる大きさや重量はあらかじめ決められているため、衛星システムはこれらの制約を考慮したうえで開発を進める必要がある。

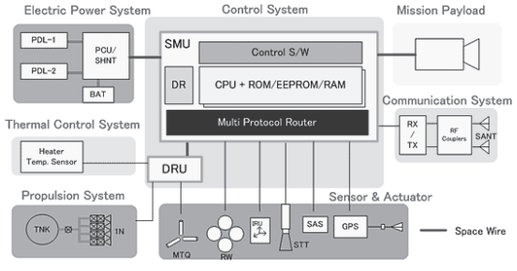


図2-1　標準的な衛星システム構成[1]

表2-1　各サブシステム名称と役割

名称	役割
電源系 Electric Power System	太陽電池パドルによって電力を発電し、各系に供給する
熱制御系 Thermal Control System	衛星搭載機器の温度を制御する
推進系 Propulsion System	スラスタにより推力を発生させる
姿勢軌道制御系 Sensor & Actuator	衛星の姿勢や位置を計測し制御する
通信系 Communication System	地上のアンテナと通信し、衛星データの送受信を行う
マネジメント系 Control System	各系に対して指令を通知、状態を監視する
ミッション系 Mission Payload	撮像カメラなど、衛星の目的（ミッション）に応じて搭載する

3. 衛星システム開発プロセスの概要

衛星システム開発のプロセスは、V字型モデルである。分割のプロセスでは、総合システム設計、衛星システム設計、衛星サブシステム設計およびソフトウェア設計という流れで機能が分割される。一方、統合のプロセスでは、ソフトウェア試験、衛星サブシステム試験、衛星システム試験およびシステム運用という流れで統合されていく。

図3-1に、衛星システム開発プロセスを示す。

近年は、搭載機器の標準化が進んでおり、衛星システム開発における課題は、どのように高度なミッションを低コスト、短納期で実現するかという点にある。そのため、設計検証の手法を衛星実機の試験から、検証ツールを用いたシミュレーションに移行し、上流工程での検証内容を充実させることが必要である。上流工程において、様々な設計検証を実施することができれば、後工程における後戻りリスクの低減が期待できる。（フロントローディングの実現）

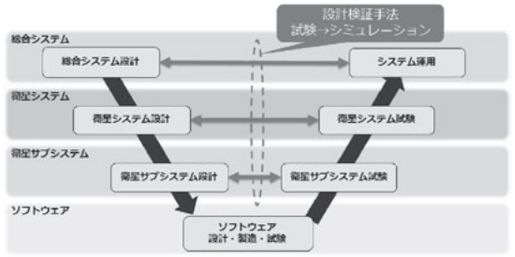


図3-1　衛星システム開発プロセス

4. 衛星シミュレータの概要

衛星シミュレータは、設計検証、運用手順検証および衛星運用者に対する訓練のためのツールという位置づけである。衛星の打上げ前において、安全に運用できることを確認するため衛星シミュレータを用いた「運用手順の検証」および「衛星運用者に対する訓練」を行う。

図4-1に衛星シミュレータの機能ブロックおよびハードウェア構成を、図4-2に衛星シミュレータの利用イメージ（訓練時）を示す。

衛星管制模擬は、衛星シミュレータに対して、コマンド（遷移トリガ）送信およびテレメトリ（状態）確認など監視制御を行う機能である。

衛星シミュレータは、衛星管制模擬機能よりコマンドを受信し、インタフェース変換機能により状態遷移モデル操作API、または外部シミュレータに分配する。各機能は受信したコマンドに応じて状態を遷移させ、テレメトリとして衛星管制模擬機能に通知する。ここでいう外部シミュレータとは、エミュレータ（組込み用FW SILS[注1]）やハードウェアシミュレータ（HILS[注2]）といったシミュレータを接続する。状態遷移モデルは、状態遷移モデル管理機能で作成し、状態遷移モデル操作APIによって操作することで状態遷移シミュレーションを実現する。このような複数の機能により、衛星システムのシミュレーションを実現している。

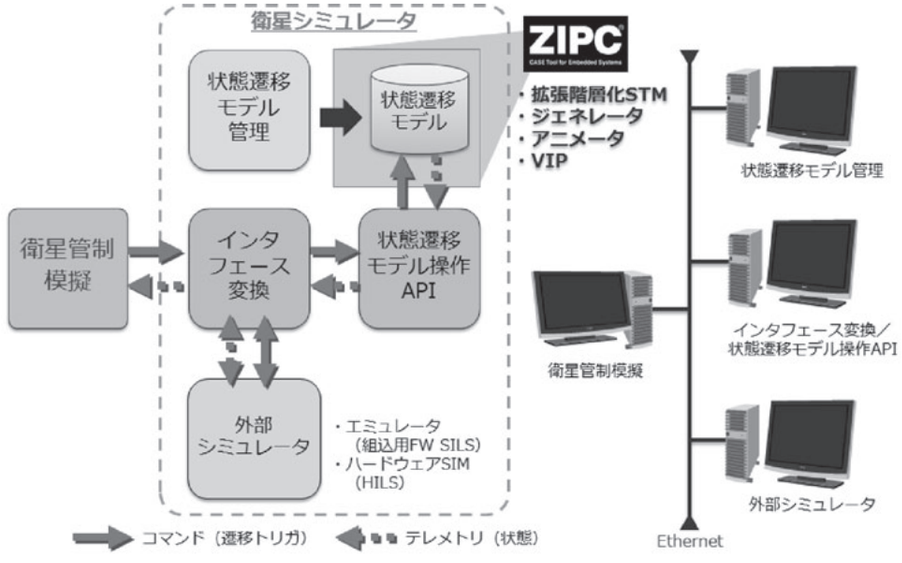


図4-1　衛星シミュレータの機能ブロックおよびハードウェア構成

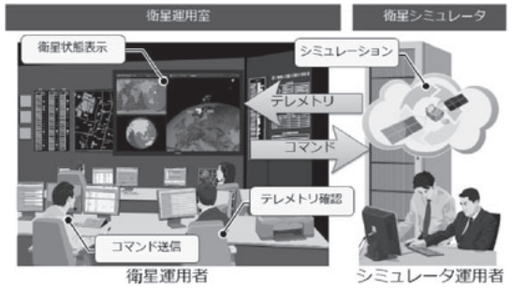


図4-2　衛星シミュレータの利用イメージ
（訓練時）

弊社で開発した衛星シミュレータでは、以下の4つのZIPC機能を活用している。

① 拡張階層化状態遷移表
② ジェネレータ
③ アニメータ
④ VIP（Visual Interface Prototype）

5. ZIPCの導入

本章では、ZIPC導入の目的および期待する効果について述べる。

5.1 なぜZIPCの導入が必要か

衛星シミュレータを開発するにあたって、以下に示す3つの大きな課題があった。

① 効率的な衛星設計情報の取り込み
② 複雑な衛星動作シミュレーションの実現
③ 複数のシミュレータとの連携、高速シミュレーションの実現

衛星システムは、大規模なシステムであるため、膨大な設計情報を衛星シミュレータ開発に効率よく取り込む必要がある。また、各衛星サブシステムは連携して並行動作するため、複雑な衛星動作のシミュレーションを実現する必要がある。一方、衛星シミュレータの利用時においては、設計検証の効率化のため、外部シミュレータと連携し、高速でシミュレーションを行う必要がある。

これらの課題を解決するため、ZIPCの導入を行った。次項では、それぞれの課題についてZIPC導入に期待する効果を述べる。

5.2 効率的な衛星設計情報の取り込み

まず、1つ目の課題である「効率的な衛星設計情報取り込み」について、ZIPC導入に期待する効果を述べる。

衛星設計情報は、コマンド・テレメトリ情報および動作モード設計など多くの文書データとして存在しており、膨大な衛星設計情報をもとに1からマニュアルで状態遷移モデルを構築するのは現実的ではない。したがって、衛星シミュレータに効率的に衛星設計情報を取り込む方法が必要である。この課題を解決するために、衛星設計情報を共用DB化するとともに、ZIPCを用いて状態遷移モデルを自動生成することができれば、作業の効率化が期待できる。

図5-1に、効率的な衛星設計情報の取り込みイメージを示す。

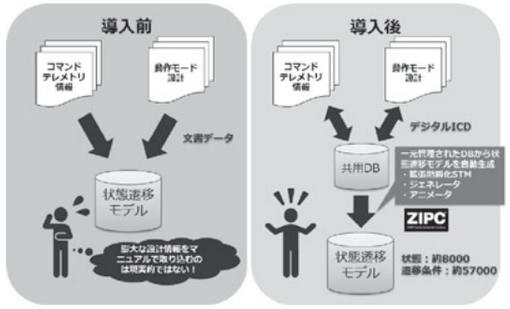


図5-1　効率的な衛星設計情報の取り込み

5.3 複雑な衛星動作シミュレーションの実現

次に、2つ目の課題である「複雑な衛星動作シミュレーションの実現」について、ZIPC導入に期待する効果を述べる。

衛星の一部搭載機器は、上流機器の電源をONしていなければ、コマンドやテレメトリを伝送できない。また、異常時は地上からコマンドを送信しなくても、各搭載機器が連携し、自律的に安全な状態に遷移するなど複雑な動作を行う。衛星シミュレータは、上記のように複雑な動作をシミュレーションできる状態遷移モデルが必要である。

ZIPCでは、拡張階層型状態遷移表により、形式的に表現することができるため、複雑な衛星動作であってもシミュレーションの実現が期待できる。

図5-2に、複雑な衛星シミュレーションの実現イメージを示す。

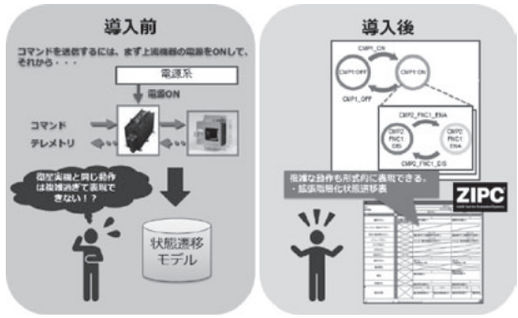


図5-2　複雑なシミュレーションの実現

5.4 複数のシミュレータとの連携、高速シミュレーションの実現

最後に、3つ目の課題である「複数のシミュレータとの連携、高速シミュレーションの実現」について、ZIPC導入に期待する効果を述べる。

衛星システム開発の設計検証において、衛星実機を用いた試験を行う場合、大型かつ複数の試験装置を使用し、クリーンルームで試験を行うため、大掛かりな試験環境が必要である。また、衛星実機および試験装置を動作させるため、試験手順書・試験データの整備を行う多数の人員リソースが必要となる。試験実施時には、衛星実機を用いているため、リアルタイムでしか動作させることができず、昼夜をかけてシフト体制を組み、長時間かけて試験を行う必要がある。

設計検証の手法として、ZIPCを活用した状態遷移シミュレータを導入することができれば、ソフトウェアによる状態遷移シミュレーションを可能とし、かつVIP機能を用いることで、さまざまな外部シミュレータと連携することが可能である。衛星システム開発では、設計検証ツールとして多くの外部シミュレータが整備されるため、これらと連携することができれば、検証環境の省力化が期待できる。また、全てを汎用計算機上のソフトウェアで実現することができれば、高速でシミュレーションを行うことも可能となり、検証期間の短縮および検証内容の充実化が期待できる。

図5-3に、外部シミュレータ連携および高速シミュレーションの実現イメージを示す。

図5-3　外部シミュレータ連携および
高速シミュレーションの実現

6. ZIPC活用事例の紹介

本章では、ZIPCの活用事例を紹介する。

6.1 デジタルICDに基づくSTMの共用DB化

5.2項で述べたとおり、衛星システムの状態遷移モデル（STM）を1からマニュアルで整備することは現実的ではないため、搭載機器のコマンド・テレメトリ情報、動作モード（状態遷移図）、発熱量、消費電力および形状について、電子的に蓄積されたデジタルインタフェース管理図面（ICD）からSTMを生成できるようにした。これにより、製品データベースから構成品を組み合わせることで、衛星ごとのSTMを効率的に自動生成できるようになる。また、過去に作成したSTMも利用できる。

図6-1に、デジタルICDに基づくSTMの自動生成イメージを示す。

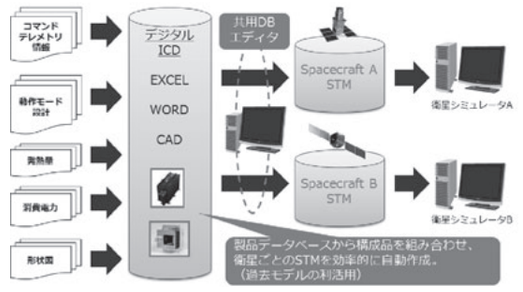


図6-1　デジタルICDに基づくSTMの自動生成

6.2 運用手順の早期検証

ZIPC のアニメータ・オプション機能を使用し、状態遷移シミュレーションを行うことで、上流工程で設計検証を実現できるようにした。これにより、製品製造前の抽象度が高い設計段階においても早期に運用手順の検証ができるため、後工程における後戻り発生リスクを低減することができる。

図 6-2 に、早期運用手順の検証イメージを示す。

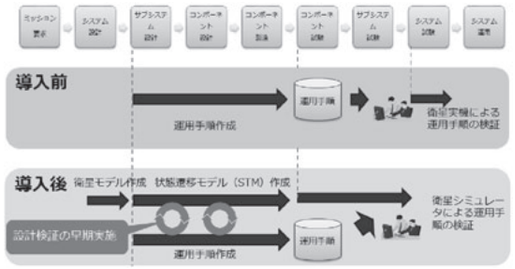


図6-2　早期運用手順の検証

6.3 外部シミュレータとの連携動作

ZIPC には、VIP(Visual Interface Prototype) 機能が備わっており、さまざまな外部シミュレータと連携することができる。

例えば、姿勢・軌道制御系の搭載 FW を活用した SILS と連携することで、太陽電池パドル展開時の姿勢変動をシミュレーションすることができる。これにより、衛星運用者に対して実運用を想定したリアリティの高い訓練が実施できる。

図 6-3 に太陽電池パドル展開時の姿勢変動シミュレーションイメージを示す。

また、上流機器の電源 ON/OFF 状態の関係および異常時の連携動作など複雑な衛星動作をシミュレーションするために、外部シミュレータを経由して再度 STM に入力し自動で遷移できるようにした。

図 6-4 に上流機器電源 ON/OFF 時のシミュレーションイメージを示す。

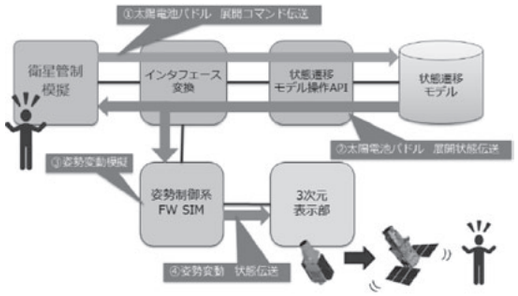


図6-3　太陽電池パドル展開時の姿勢変動シミュレーション

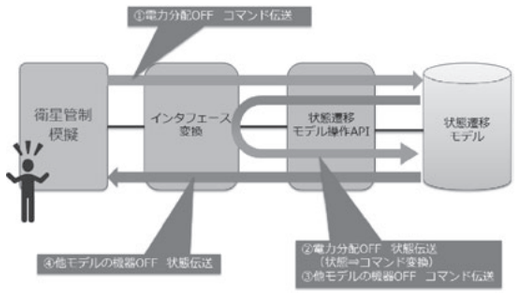


図6-4　上流機器電源ON/OFF時のシミュレーション

7. ZIPC 導入の効果

本章では、ZIPC 導入による効果を述べる。

衛星シミュレータ開発において、ZIPC を導入した効果は大きく 2 つある。

1 つ目は「モデルによる設計作業の共通化」ができることである。状態遷移モデル（STM）を中心に、衛星システム開発と並行して開発することが可能であり、効率のよいものづくりを行うことができる。また、モデルから成果物を生成することが可能であり、過去モデルの利活用もできる。

2 つ目は「検証作業の充実化」である。ZIPC を用いた動的検証は、抽象度の高い開発初期から設計検証に活用することができ、後工程での後戻りリスク低減に有効である。また、衛星システム開発において整備する多くの外部シミュレータ（SILS、HILS）と連携可能であり、同じような機能を持った設計検証ツールの乱立を抑制することができる。

8. まとめ

本論文では、衛星シミュレータへの ZIPC の活用事例紹介として、衛星シミュレータの開発から利用時までの ZIPC 活用事例を紹介した。

一般的に難しいとされる衛星システム開発を含め、高度化・複雑化し続けるシステム開発の QCD を改善するために、ZIPC を活用してモデルベース開発を行うことは有効な手段である。とりわけ、ZIPC はモデルベース開発に必要な多数の機能を備えており、CASE[注3] ツールとして多くの実績があり信頼性が高い。

今後の衛星システム開発では、さらにモデルを活用したシステムエンジニアリングの導入を検討していく必要があるため、ZIPC に期待し活用していきたいと考えている。

9. 謝辞

本論文の作成にあたって、衛星シミュレータの ZIPC 活用範囲の調査にご支援頂いた CATS 殿、またご協力頂いた関係各位に感謝の意を表する。

参考文献

[1] 標準衛星システム NEXTAR
http://jpn.nec.com/solution/space/technology/bus/nextar.html
[2] 三好弘晃、三好寛：人工衛星システム開発への ZIPC の適用、ZIPC WATCHERS Vol.6、pp.9-20

注 1 Software In the Loop Simulation
注 2 Hardware In the Loop Simulation
注 3 Computer Aided Software Engineering

ZIPC WATCHERS Vol.20

IoT 時代の組込みソフトウェアに向けたコンテキスト指向技術

東海大学　情報通信学部　教授

渡辺　晴美

1. はじめに

非正常系や幅広いシリーズ・バージョンへの対応は、組込みソフトウェアを複雑にし、理解性を損ね、保守やテストを難しくする問題として知られている。

IoT、Industry4.0、スマートロボット、自動運転等の近未来型組込みシステムにおいて、従来の非正常系等の問題は、より深刻化する。近未来型システムの特徴に、沢山の機器とのつながり、様々な環境に応じたサービスの提供がある。環境への適応は、従来の組込みシステムにおける非機能要件の発展型とみなすことができ、たくさんつながることで、個々のシステムが判断すべき入力情報も増える。従って、近未来型システムは、従来よりも、複雑で多入力な非正常系を扱うことになる。こうした問題を解決するために、環境に応じたサービスの適応、すなわち非正常型の問題を緩和する方法の一つとして、コンテキスト指向技術と呼ぶ技術がある。

本稿では、非正常系と環境適応の問題について再考し、コンテキスト指向技術の紹介をする。また、本技術と関連した振る舞いモデリングとテストについて述べる。

2. 非正常系と環境適応

非正常系の問題について考えてみる。何か一つの異常信号を受信した場合、振る舞いの変化は受信したモジュールのみに留まらず、他のモジュールに影響を及ぼす場合が多々ある。図 1 にその様子を示す。例えば、自動走行しているロボットが、カメラと距離計の両方で障害物を探知している際に、カメラの信号が途絶えたとする。この場合、カメラのモジュールの処理のみが変わるのではなく、障害物の探知方法も変わり、走行スピードを遅くすることになる。図 1 では、Monitor() と Analyze() に影響を及ぼしている。すなわち、一つの信号が、横断的に複数のモジュールに影響を与えることになる。このような事柄は、横断的関心事と呼ばれ、ソフトウェアを複雑にする事柄として知られている。また図 3 上側のプログラム例に示すように、従来のプログラムの多くは、非正常系のための多数の分岐を有し、これらが横断的関心事となることが多々ある。

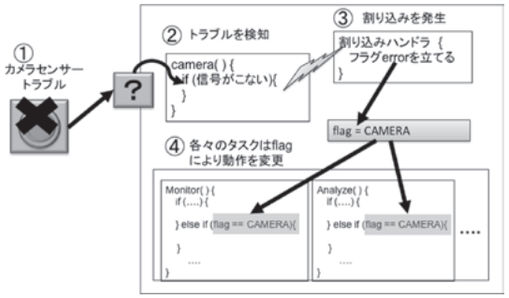


図1　非正常系処理

さらに、前述したとおり、未来型組込みシステムでは、スマートフォンのように、環境に応じた複数サービスの提供が望まれている。このことが、非正常系問題をさらに深刻にする。機器を動かすシステムをオブジェクト指向開発した場合、骨格となるクラスは、監視→解析→計画→実行［5］のような、システムを動かすのに必要な動作の基本要素となることが自然に思える。その場合、図 2 の Monitor、Analyze はサービスごとの分岐が加わり、各サービスがさらに非正常系を持つことになる。逆に、動作の基本要素ではなく、サービスを中心に考えても、基本要素の方が、横断的関心事になり、本質的な解決にはならない。

この非正常系と横断的関心事の問題は、多数の分岐による理解性の低下の問題に加え、テスト・保守の抜け漏れの問題を引き起こす。例えば、カメラのトラブルが影響を与えるモジュールを全てテストすることは容易であろうか？設計時に把握していなければ難しく、テスト漏れになる可能性がある。この問題は網羅率で防ぐことはできない。例えば、カメラのトラブル以外のテストケースが、全てのモジュールを実行してしまえば、網羅率は 100% になるためである。影響範囲を意識し、テスト項目が設定されていたとしても、コードに明示的に表示されていなければ、トレーサビリティが低下する。

保守を行う場合も、同様に影響するモジュールを全て洗い出すのは容易ではない。加えて、分岐の多い理解性の低いモジュールでは、すでにあるコードを冗長に加えてしまうかもしれない。

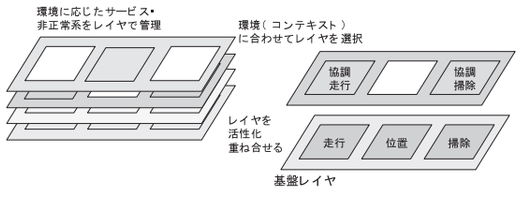


図2　コンテキスト指向技術の概要

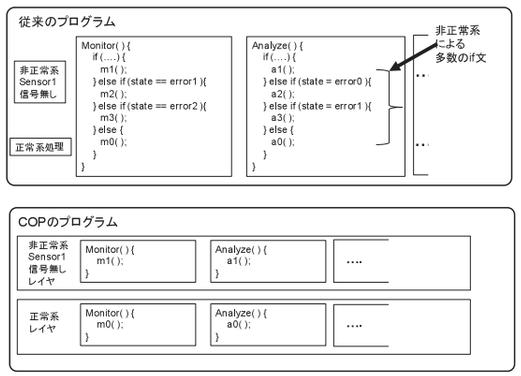


図3　従来のプログラムとCOP

3. コンテキスト指向技術

コンテキスト指向技術とは、コンテキスト指向プログラミング技術 (Context-Oriented Programming: COP) の概念を取り入れたソフトウェア開発技術である。COP の代表的な研究に R. Hirschfeld らをはじめとした研究がある［1-4］。組込みシステムに対しては ROS 対応の研究がある［5］。COP の多くは、環境の変化に応じて、振る舞いを変化させるために、レイヤと呼ぶ概念を取り入れている。図 1 に示すとおり、レイヤは、環境に応じたサービスや非正常系をレイヤで管理する。レイヤは複数のオブジェクトから構成される。図 2 は自動掃除機の例である。自動掃除機は、基盤レイヤにあるとおり、走行、位置獲得、掃除の 3 種類のオブジェクトからなり、複数台で掃除を行う場合は、協調レイヤを活性化し、基盤レイヤに上書きする。この図では、走行と掃除が、協調走行と協調掃除に書き換わる。このように、環境（コンテキスト）に応じて、複数のオブジェクトに対し横断的に振る舞いを変更することが可能である。

図 3 に従来のプログラムと COP のプログラムを比較した例を示す。従来のプログラムでは、非正常系による多数の if 文があるのが通常である。これに対し、COP では、レイヤにより、モジュールごとに散在していた分岐を減らすことができる。すなわち、レイヤごとの各モジュール内は整理され、レイヤで着目した関心事と関係のあるコードのみになる。カメラが故障した場合等のシステムに横断的なテストと一致したコードを表現できる。従って、前述したテスト・保守にも貢献できると考える。

レイヤについて、プロセスやタスクと似ているという誤解が時々見受けられる。レイヤは、プロセスやタスクというよりも、オブジェクト指向のオーバーライドやポリモフィズムの集合体に近い。ポリモフィズムはメソッドを呼び出すときに、オブジェクトに応じて単体のメソッドのみが変わるが、コンテキスト指向技術では、レイヤで指定されたメソッド、あるいはオブジェクトに属するメソッドは全て変わることになる。また、レイヤは必ずしもシステム全体を変更するのではなく部分変更可能である。関連したオブジェクトをレイヤとしてまとめることもできる。すなわち、プロセスやタスクは、並行処理が目的であるのに対し、レイヤは、理解性の向上が目的であると言える。

ZIPC WATCHERS Vol.20

4. フレームワーク

前章で、コンテキスト指向技術はオブジェクト指向技術のオーバーライドやポリモフィズムの集合体であるということを述べた。こうした技術は、組込みソフトウェアに不可欠なリアルタイム性やリアクティブ性を損ない、結局、テストケースの数を増やし、保守を複雑にするのではないかという疑念を持たれるであろう。実際、現在提案されているプログラミング言語の多くは、レイヤの機能があるのみである。従って、熟考せずにプログラミングすれば、こうした問題は起こりうる。

また、全く新しい考え方でシステム全体を作ることは、レガシーソフトウェアを多く抱える組込みシステムでは現実的ではない。そこで、我々は、下記で述べる3つの特徴を持つフレームワークを提案している［6-8］。このようなフレームワークについて［9］も取り組んでいる。

（1）非正常系分離と導入容易性：これまで述べてきた非正常系の問題、既存資産の問題、新規パラダイムの学習コストを軽減するために、図4に示すとおり、レイヤ・マネージャ部分とレイヤ部分に分離している。レイヤ部分では、COP特有のコードを排し、従来のソースコードをそのまま利用できるように工夫している。レイヤ・マネージャ、すなわちレイヤ切り替えを行う部分を分離することにより、図3に示した理解性の高いコードが実現できる。一方、COP特有のコードを排することで、既存資産を利用可能にすることができる。各レイヤにオブザーバとよぶ監視役を配置し、マネージャ側とやりとりすることで、実現している。

（2）安全なサービスの提供：レイヤの切り替えは、前述の通り、資源競合、実行順序、リアルタイム性などを考慮した設計実装が必要となる。従ってこの部分をオペレーティングシステムのように分離することを目指している。図4のレイヤ・マネージャ内をさらに、スケジューラとリスナーに分離し、リスナー部分を開発者は設計する。

（3）新規サービスの追加：保守性を高めるために、稼働を止めずに新規サービスを追加可能にする。具体的には、サービスに対応するレイヤを実行時に追加可能にする。

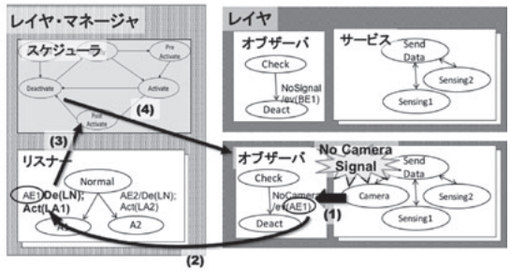


図4　RT-COAフレームワーク

5. 振る舞いモデル

コンテキスト指向技術を導入するようなシステムでは、状態遷移図に基づいた振る舞いモデルとテストは極めて重要である。コンテキスト指向技術と関連した、振る舞いモデルとテストに着目した研究として［10］が知られており、我々も取り組んでいる［8］。特に重要になる箇所について以下で述べる。

（1）環境の判断：環境の変化を判断する過程の多くは、単純ではない。すなわち、単一のスイッチ切り替えや、センサの閾値よりも複雑であると考える。従来も、センサが閾値を一度超えただけでは、エラーと判断せずに、様々な要件を考慮している。イベント駆動の一つのイベントというよりも、環境を変化させるイベントは、それ自体に状態変化を持つ。未来型システムでは、この部分に、ディープラーニングなどの技術が入ってくる重要な箇所と考える。

（2）レイヤの切り替え：組込みソフトウェアでは、突然のレイヤ切り替えはできない。スマートフォンでは、ゲームの最中に電話を受けることは可能かもしれないが、ロボットの場合、切り替え時に、ロボットアームは振り上げられているかもしれないし、停止しているかもしれない。さらに、これらが動作を変える時には、切り替えるためのアルゴリズムが必要となり、実行順序が保たれている必要がある。

従って、レイヤ切り替え時には、関連するモジュールの状態把握と実行順序の保持が重要であり、これらについての振る舞いモデルとテストが重要となる。

6. おわりに

本稿では、IoT時代のソフトウェアは、非正常系の問題をより複雑にすることについて再考し、こうした問題を解決する萌芽的な技術として、コンテキスト指向技術について紹介した。コンテキスト指向技術は萌芽的ではあるが、考え方は、現在の開発にも参考になる考え方である。例示したカメラのトラブル問題において、設計時に影響範囲を把握し、コードで明示すること、また、4章のフレームワークで紹介したように、横断的関心事の分岐をサービスの外側に持っていく考え方等は、COP言語でなくても、C言語やC++言語等でも十分に実現可能である。

今後、コンテキスト指向技術は、メジャー言語に取り入れられ、開発方法論が整理されていくことが期待できる。

参考文献

［1］ R. Hirschfeld, P. Costanza, and O. Nierstrasz: Context-oriented Programming, Journal of Object Technology, Vol. 7, No. 3, pp. 125-151, 2008.

［2］ G. Salvaneschia, C. Ghezzia, M. Pradellaa: Context-oriented Programming: A Software Engineering Perspective, Journal of Systems and Software archive, Vol.85 Issue 8, pp. 1801-1817, 2012.

［3］ T. Kamina, T. Aotani, and H. Masuhara, Generalized Layer Activation Mechanism Through Contexts and Subscribers, In Proceedings of the 14th International Conference on Modularity (MODULARITY'15), pages 14-28, 2015.

［4］ Y. Saeki, I. Tanigawa, K. Hisazumi, A. Fukuda: ContextROS: Context-Oriented Programming for the Robot Operating System, Proceedings of the Workshop on Context-oriented Programming (COP) 2017, ECOOP 2017,2017.

［5］ J. O. Kephart and D. M. Chess: The vision of autonomic computing, in Computer, vol. 36, no. 1, pp. 41-50, Jan 2003.

［6］ H. Watanabe, M. Sugaya, I. Tanigawa, N. Ogura, and K. Hisazumi: A Study of Context-Oriented Programming for Applying to Robot Development, Proceedings of the Workshop on Context-oriented Programming (COP) 2015, ECOOP 2015,2015.

［7］ H. Watanabe, I, Tanigawa, M. Sugaya, N. Ogura, K. Hisazumi: A Layer Structure Diagram and a Layer Interaction Diagram towards a Context-Oriented Development Methodology for Embedded System, The Workshop on Live Adaptation of Software Systems (LASSY' 16), 2016.

［8］ H. Watanabe, I. Tanigawa, N. Ogura, M. Sugaya, K. Hisazumi and A. Fukuda: Coloured Petri-Nets Framework for Simulating Method Invocations on Context-Oriented Software, Proceedings of the Workshop on Meta-Programming Techniques and Reflection (META) 2016, SPLASH 2016,2016.

［9］ Kim Mens, Nicolás Cardozo, and Benoît Duhoux. 2016. A Context-Oriented Software Architecture. In Proceedings of the 8th International Workshop on Context-Oriented Programming (COP'16).

［10］ N. Cardozo, S. González, K. Mens, R.Van Der Straeten, J. Vallejos, T. D' Hondt: Consistent Activation Semantics of Context-Oriented Systems, Journal of Information and Software Technology (JIST). Else-vier, 58 - 2015, pp. 71--94, 2015.

コードから STM へリバース RexSTM for C ツールのこれまでとこれから

JASA 状態遷移設計 WG
主査
青木　奈央

名古屋大学　大学院情報学研究科
情報システム学専攻
吉田　則裕

1. はじめに

昨年、本稿に「コードから STM へリバース RE x STM for C ツールについて」と題して JASA（組込みシステム技術協会）の状態遷移設計研究 WG で取り組んでいるリバースエンジニアリングを補助する手法やツール開発について紹介した。今回は、本年の組込み総合技術展（Embedded Technology 2017: ET2017）以降で JASA 会員に公開されることが決まったので、いままでの経緯と、今後の活動、また、ツールの使い方等について紹介する。

2. これまでの経緯

JASA の状態遷移設計研究 WG では、近年、組込みソフトウエア開発の現場で懸念されているレガシーコードの複雑化・肥大化やドキュメントの陳腐化に着目し、その現状を踏まえて議論を重ねてきた。特に、設計資料がソースコードのみである場合や、担当者が不在でソースコードや設計書を修正すると他の不具合が発生してしまう場合などの問題については、WG 内で多くの時間議論を行った。その結果、これらの問題を解決すべく、リバースエンジニアリングの手法に着目し、補助するツールの開発を検討してきた。

世の中のリバースエンジニアに関するツールを調べてみると、いくつかあるが、ソースコードからコードのフロー、いわゆる構造やロジックをビジュアル化するものはあるが、ソースコードから状態遷移表を生成するツールは見受けられなかった。組込みシステムの分野では、状態遷移の考え方が一般的であり、ブラックボックス化したソースコードから状態遷移表を生成するツールは企業からの需要が高いと考えられる。

今回の RE x STM for C での使用言語を C 言語に選定した理由としては、組込みシステムのソフトウエア開発言語では、C++、C#、JAVA、PHP などがあり、一昔前ではアセンブラという時代もあったが、現在では、まだまだ多くの組込みソフトウエアは、C 言語で開発されているものも多く、状態遷移設計研究 WG 内で議論にあがったレガシーコードと呼ばれているものの多くは C 言語であった。このような理由から今回の対象言語を C 言語と決定した。

C 言語のソースコードから状態遷移表を生成するツールを開発するに際して、ソースコードの解析は必須になるのだが、これをリバースエンジニアリングするときに、C 言語の構文要素に非常に依存することが判明した。ヘッダーファイルに書かれている情報、例えばコメントなどに関しては、これらをリバースするには非常に困難である。そこで、リバース可能な情報を精査し、コメント等はリバースする対象とはせず、限定的なソースコードを状態遷移表生成の対象とすることにした。

3. RE x STM for C

RE x STM for C とは、レガシーコードから状態遷移表を自動で生成するツールである。手順は、大きくわけて 3 つのステップに分けられる。1. レガシーコードの整形（図 1）、2. 中間ファイルの作成（図 2）、3. 状態遷移表の生成（図 3）となる。これらの手順を進めるにあたり、図 1 から図 3 のように進めていく。

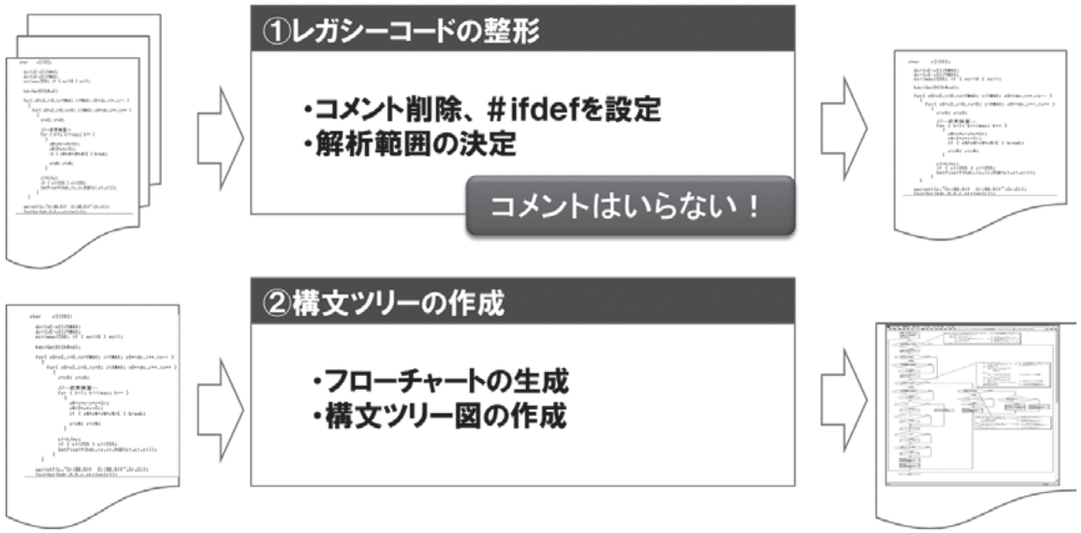


図1　レガシーコードの整形

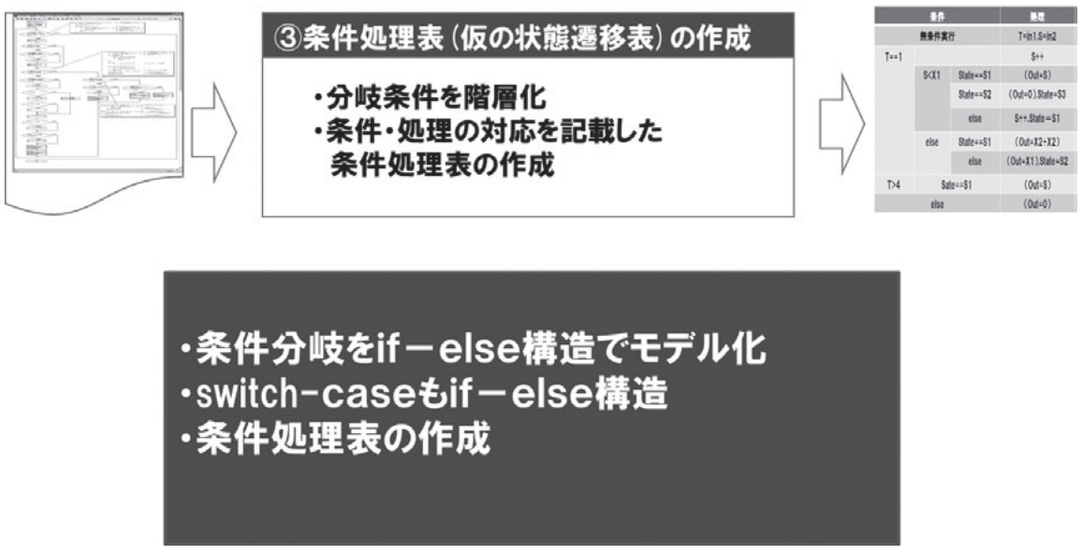


図2　中間ファイルの作成

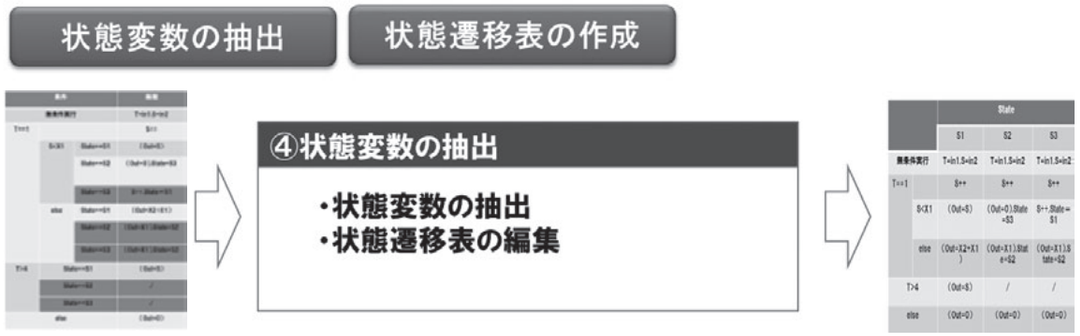


図3　状態遷移表の生成

最初に、レガシーコードの整形を行う。これは、既存のコードに記載のあるコメントなどを削除し、#ifdef を設定し、コード解析の範囲を設定する。次に構文ツリーの作成を行うのだが、フローチャートを生成後、構文ツリー図の作成を行う。その後、分岐条件を階層化し、条件・処理の対応を記載した仮の状態遷移表を作成する。これが中間ファイルにあたる部分となる。

仮の状態遷移表を作成後、状態変数を抽出し、状態遷移表を作成する。この時のポイントとしては、状態変数を見つけることや、分岐条件の変数は、状態変数かを見極めることである。状態変数の定義としては、状態変数は、有限個であり、内部で変更されることである。状態遷移表が生成されたら、レビューができる状態となる。

4. 考察

このツールの適用効果として4つがあげられる。1. 機械的にソースコードから状態遷移表を作成するプロセスがみえてきた。2. 振舞いの可視化、モデル化によりレビューがしやすくなった。振舞いの漏れ抜けが発見できるようになった。3. 状態変数名の変更など、リファクタリングの要素が抽出可能になった。4. 言語に依存せず、すべてのコードに適用可能であることがわかった。今回のツールはC言語対応としているが、他のJAVAなどの言語でも十分適用可能であると考える。

現状のツールで状態遷移表が生成できることにより、第三者でも容易に仕様が理解できるようになる。また、状態遷移表を主要に扱うツール（ZIPC）などと連携をはかることで、仕様の変更だけでなく高品質なコード、製品を顧客へ提供可能になり、より有効に活用できると考える。

5. 今後の展開

RE x STM for Cでは、仮説の裏付け、手順が適用できることは証明されたが、様々なパターンに適用できるかの検証は不十分である。実際に変換が困難なパターンとして、「状態の階層化パターン」if-else で条件が階層化されている場合、状態遷移表も階層構造となる場合、「状態変数の演算パターン」and や or で演算してから判定している場合、演算式を状態変数として判定する場合、「状態変数の置換えパターン」A&B をCに代入して、Cで判定する場合に、変数の変更は A,B で行っている場合、「状態変数の関数渡しパターン」A&B をCに代入して、Cで判定しているとき、変数の変更は関数内で行っている場合、「状態の並列化パターン」状態が並列である場合の優先順位、「イベント条件と内部条件」状態変数以下の層の分岐条件をイベントでなく処理する場合が考えられる。今後は、自動化できる部分はツール内で実行できるように進める。例えば、ソースから条件・処理対応表を自動生成や状態変数候補自動抽出、選択後、状態遷移表の自動生成をめざす。

また、RE x STM for Cのプログラム解析部分において、記号実行ツールの利用を考えている。記号実行とは、静的解析技術の一種であり、プログラムを実行させることなく、入力値の値域と実行経路のペアを列挙する技術である。記号実行では、入力として具体的な値（例：1、2、3などの整数値）を与えず、記号（例：x、 y、z）が与えられたとし、各記号の値域（例：0＜x≦5、かつ5＜y≦7）に対応する実行経路を列挙する技術のことである。テストケースを十分に用意することができれば、プログラムを動作させることで、記号実行と同様に各記号の値域と対応する実行経路を列挙することができるが、レガシーコードではテストケースが十分に用意されていないことが多いため、プログラムを動作させることによる解析は困難である。記号実行はテストケースを用意する必要がないため、レガシーコードの分析に向いていると考えられる。

これまでのところ、RE x STM for Cでは構文解析技術のみを使用し、記号実行等の高度な解析技術を使わずに実装を行ってきた。近年、コンコリックテストが注目されており、その影響で記号実行技術についても様々な研究が行われている。これら研究成果を活用し、RE x STM for Cのプログラム解析部分を洗練されれば、より多くのソースコードに対して正確な状態遷移表を抽出することができると期待している。

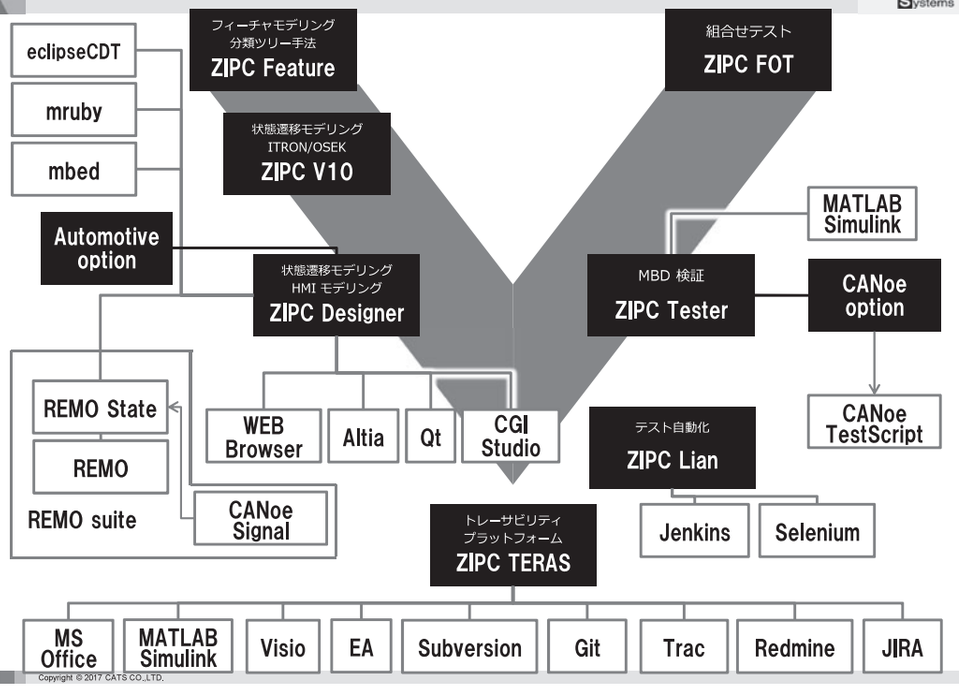
具体的な記号実行技術の活用方法について述べる。記号実行ツールでは、ソースコード中の各ブロックについて、到達可能な条件を記号の値域として求められる（例:0＜x≦5,かつ5＜y≦7)。これを利用し、状態遷移表における遷移条件と遷移後に実行されるプログラムのペアを求めることができるのではないかと考えている。RE x STM for Cでは、状態変数はユーザが与える前提であるため、イベントの発生を判定する条件式と状態変数を判定する条件式の区別は自動的に行うことができると考えている。

現在、KLEE（https://klee.github.io/）等の記号実行ツールを調査することで、状態遷移表の抽出に有効な記号実行ツールを調査しているところである。一部の記号実行ツールは、コンパイル可能であることを前提としているため、このような前提で実際の開発現場で受け入れられるかについても、状態遷移設計WGの活動を通して確認していきたいと考えている。

青木奈央：nao@zipc.com
1998年BS in Mathematics, American University, Washington D.C. IPA研究員、北陸先端科学技術大学院大学 研究員などを経て現在、キャッツ株式会社プロダクト事業本部に所属

吉田則裕：yoshida@ertl.jp
2004年九州工業大学情報工学部知能情報工学科卒業。2009年大阪大学大学院情報科学研究科博士後期課程修了。博士（情報科学)。現在、名古屋大学大学院情報学研究科准教授。

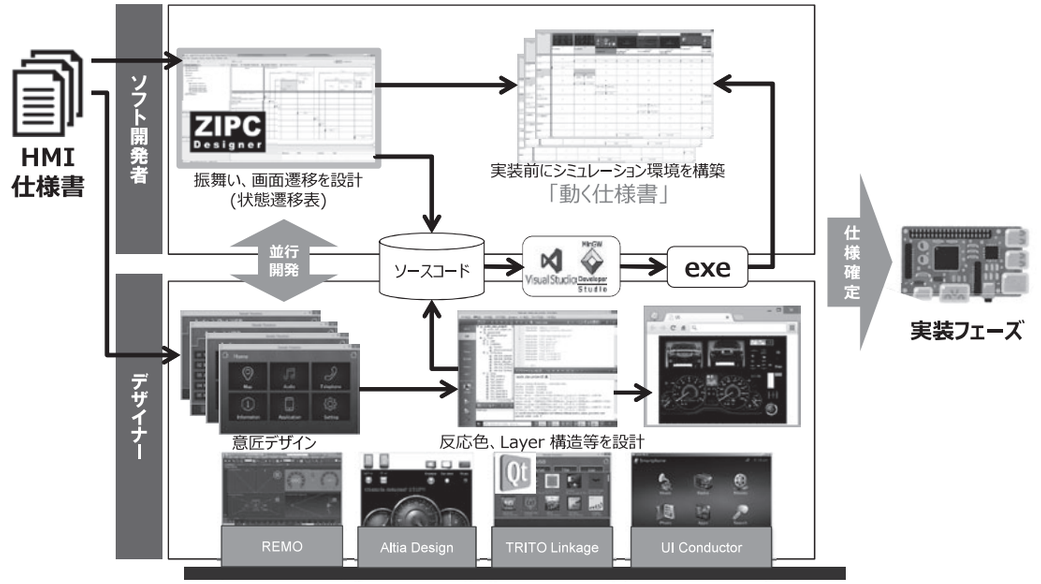
MBDを支援するZIPCファミリー



e-コックピット時代の高度HMI開発支援ツール

ZIPC Designer

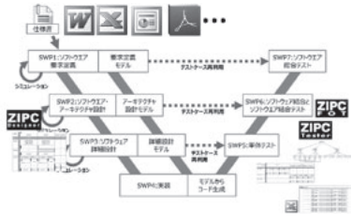
様々なオーサリングツールと連携して高度な HMI 開発の品質生産性を向上させます。



トレーサビリティ・ツールプラットフォーム

ZIPC TERAS

ソフトウェア開発で作成される成果物・文書間のトレーサビリティ(追跡可能性) を確保・支援
→ 文書であれば何でもトレーサビリティ可能! Subversion や Redmine 等と連携!

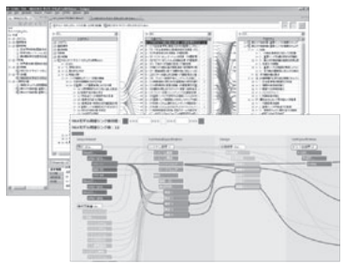


既存資産・既存環境をそのまま活用
成果物を登録
(仕様書、モデル等)

連携ツール

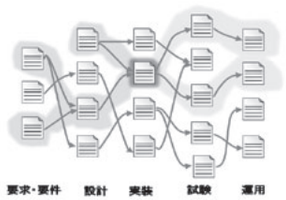
- Subversion, Git
- Trac, Redmine
- JIRA
- MS Office
- PDF
- テキストファイル
- ZIPC ファミリ
- EA
- MATLAB/Simulink

リンクエディタ



影響範囲検索

変更時の影響範囲を上流と下流それぞれに分析



カバレッジ確認

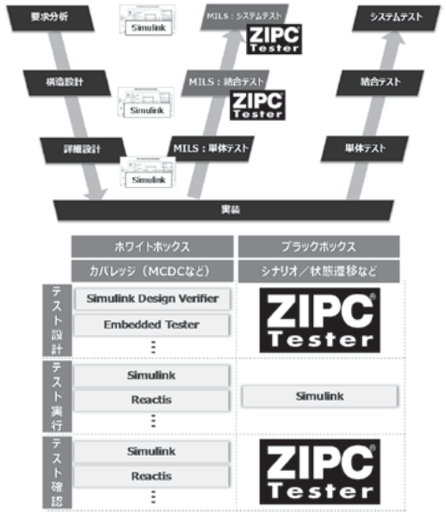
上流と下流それぞれに対応関係を確認・判定し、設計のモレヌケを防止

MBD検証支援テストケース生成ツール

ZIPC Tester

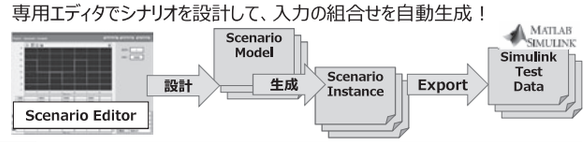
MBD における組合せテスト／システムテストのテスト設計及びテスト結果確認を支援します。
ツール支援により、担当者依存の軽減、工数削減につながります。

適用領域



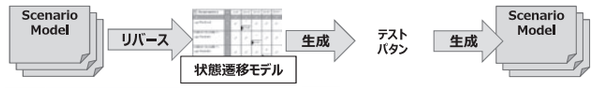
Scenario設計支援

専用エディタでシナリオを設計して、入力の組合せを自動生成！



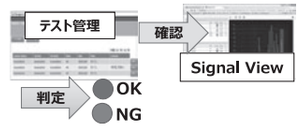
状態遷移テスト

状態遷移モデルから網羅的なシナリオを設計！
シナリオから状態遷移モデルへリバースして網羅度を確認可能！



自動判定

期待値と結果を自動判定！



Simulink連携

Simulinkを表示して選択したサブシステムからシナリオ雛形生成

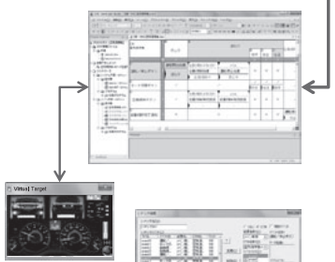
状態遷移MBDツール

ZIPC V10

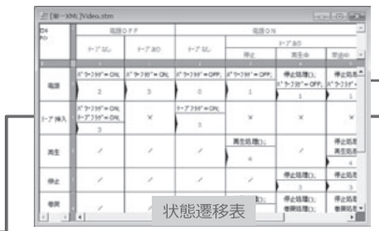
システムの振る舞いを状態遷移モデルに記述して、シミュレーションで動作検証し C コード生成まで支援するモデルベース開発ツールです。

シミュレーション

- 抽象度の高いモデルを使い、簡単にシミュレーションが可能。
- コード化の前に、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。
- 実機の完成前に、バーチャルターゲットを使いシミュレーション可能。
- 多彩なシミュレーション/デバッグ機能を搭載



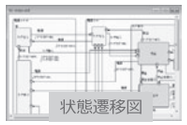
専用エディタ



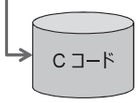
状態遷移表の差分検出機能により、派生開発や流用開発時の効率 UP

ドキュメントチェック

- 静的な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



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
- ドキュメントとコードが常に一致

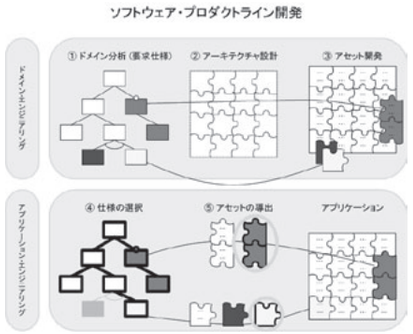
ソフトウェアプロダクトライン・モデリングツール

ZIPC Feature

**ソフトウェア・プロダクトラインは多品種開発向けのソフトウェア開発手法です。
Feature図のモデリング、部品化、その再利用により個々の製品を構成し、品質・生産性を向上させます。**

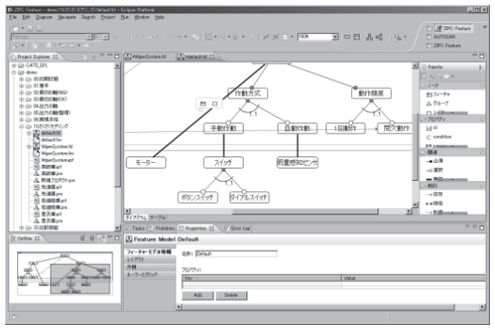
ソフトウェア・プロダクトライン開発とは

アーキテクチャが備えるべき拡張性(変動性)、又は備えなくても良い拡張性（変動性）を明らかにすることで、戦略的な再利用計画に基づく部品開発を行い、再利用によるプロダクトを導出することで多品種開発の品質・生産性を向上させます。



専用エディタでFeature図モデリングを効率的に

専用エディタで効率的なFeatureモデリングを実現し、ソフトウェア・プロダクトライン開発を強力にサポートします。

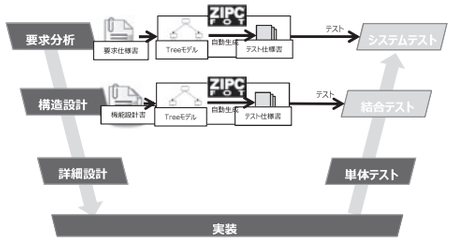


パラメータ組合せテストケース生成ツール

ZIPC FOT

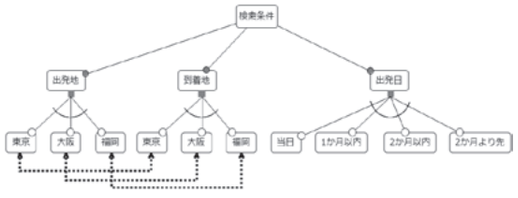
**膨大なパラメータの組合せテストケースをCTM(Classification Tree Method)を活用して
N-Wise網羅度のテストケースを自動生成する、テスト設計支援ツールです。**

適用領域



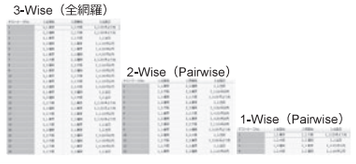
CTM(Classification Tree Method)

システムに対するパラメータとパラメータ値をツリー上に設計！
ツリーに設計するため、テスト設計のレビューが容易です。



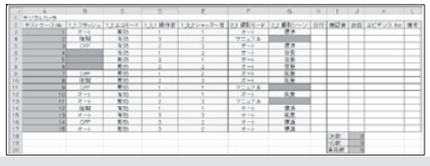
N-Wiseの網羅度テストケース生成

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る！
様々な網羅基準で、絞り込みが可能です。



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に出力！
Excelに出力することにより、多くのツールと連携可能です。



スマートフォン自動テストツール

ZIPC Lian

**ZIPC Lian は、スマートフォンのNativeアプリケーションのテスト作業を効率化するツールです。
プログラムレスで自動化シナリオの作成が可能です。**

特長1 シナリオ簡易作成(コード作成不要)

Excel ファイル上から自動実行用のテストシナリオの作成が可能です。

※ 実行コマンドとUI部品名やテキスト名を入力するだけで作成できます。
※ iOSはリプレイキャプチャ機能もございます。

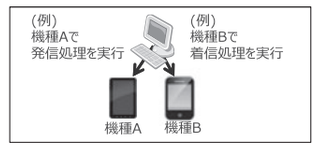
特長3 実行結果HTML出力

テスト自動化による実行結果はHTMLファイルとして自動生成されます。
実行時の各コマンドの実行結果及び画面キャプチャがHTMLファイル上から確認可能です。
結果出力ファイルは全体結果、端末毎結果、端末比較結果の複数の結果が出力されます。

特長2 対向試験実行

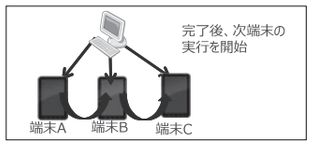
対向試験の実施が可能です。

※ 端末毎に実行するシナリオを作成して別々の処理実行を行うことができます。
(通話試験等にも使用可能です)



特長4 複数台連続実行

複数台連続で同一シナリオの実行が可能です。
また、NGの際はリトライ処理で再実行を行うこともできます。



実行方法

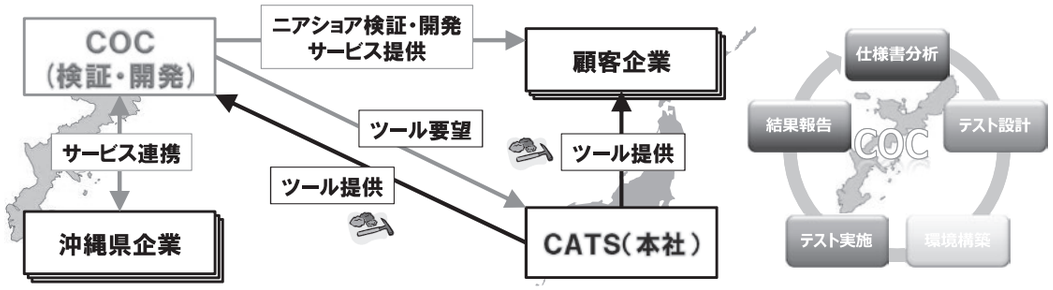
① エクセルファイル上で自動実行用のシナリオを作成※実行コマンドを入力
② 自動でビルドを行い、Excel ファイルからJavaコードに変換
③ 作成したシナリオ通りにテストを実行
④ テスト完了後、実行結果及び画面キャプチャをHTML出力

ニアショア検証・開発サービス

キャッツ沖縄センタ（COC）

Communication Art Technology Systems

キャッツが保有する技術（ツール）を利用したソリューションサービスを提供しております。
高度な技術（ツール）とニアショア環境が融合した検証事業(Webアプリ・組込み検証)及び開発事業（Webアプリ開発・Androidアプリ開発・ツール開発）を提供しております。



キャッツ沖縄センタのサービスメニュー

モバイル／Webアプリ検証	組込み検証	開発業務	ITプロセスアウトソーシング
スマートデバイス及びPCアプリケーションのテスト設計からテスト実施まで対応するテスティングサービスです。キャッツテスト自動化ツールによる効率化もご提案致します。	車載組込みシステム、医療組込みシステム製品のテスト設計からテスト実施まで対応するテスティングサービスです。キャッツ独自の組込み向けツールの適用も可能です。（ZIPC Tester/FOT）	Webアプリケーション、Androidアプリケーション、ツールを含むアプリケーションの基本設計～製造～テストまで対応する開発サービスです。主にJava開発をメインで実施しております。	非技術系の日々のルーチンワーク業務（品質監査、仕様書の差分検出、テスト結果等のデータ収集、整理、入力支援、トレーサビリティー支援）を実施するサービスです。

ルールベースシステム

ZIPC R&B

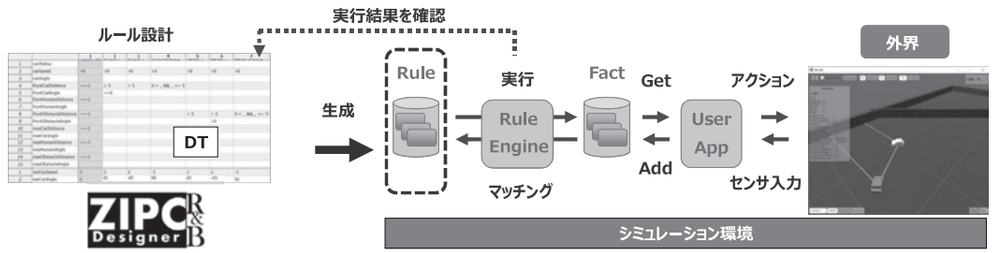
ZIPC Designer R&B

Communication Art Technology Systems

ルールベースシステムは AI 技術の一つであり、自動運転技術の一つとして注目を集めています。

ディシジョンテーブルでルール設計

ルールはディシジョンテーブル(DT)エディタを使って設計が可能です。ツール支援によってルールの設計に注力でき繰り返しの試行錯誤が容易となります。
デバッグ時にはハイライトでルールの実行を視覚的に確認できます。



ディープラーニングの課題解決に

- 判断の過程がブラックボックス
- レアケースの学習不足

- 交通ルール
- マナー
- バリエーション切替
  - ✓ 国別
  - ✓ ユーザー別

組込みデバイスのサポート

C言語をサポート、省メモリ環境でも高速動作
様々な組込環境へのポーティングが可能

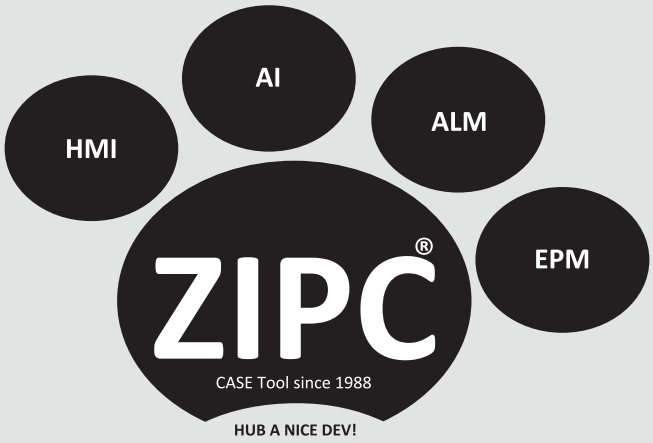
COVER IMAGE

想像力をオン！

豊かな発想と先端技術を使用して、新しい技術を創造し人、モノ、環境の利便性を高め、世界中の人々の幸福な生活へつなげていこう。

ZIPC WATCHERS Vol.20

2017 年 11 月 9 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



Communication
Art
Technology
Systems

www.ZIPC.com

最先端技術で
世界中を幸せに！

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